

Cisco Prime Optical 9.6.3 Database Schema

March 08, 2013

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: OL- 28045-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING,WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Prime Optical 9.6.3 Database Schema
© 1999–2013 Cisco Systems, Inc. All rights reserved.

Preface 18
Objectives 18
Organization 18
Audience 19
Conventions 19
Related Documentation 20
Obtaining Documentation and Submitting a Service Request 20
1 Database Schema Tables 22
1.1 Prime Optical Tables 22
1.1.1 Ack_Table 22
1.1.2 Active_Alarm_Table 23
1.1.3 Additional_Device_Info 24
1.1.4 Admin_Job_Table 25
1.1.5 Alarm_Ack_Config_Table 27
1.1.6 Alarm_Count_Table 27
1.1.7 Alarm_Event_Cause_Table 28
1.1.8 Alarm_Event_Table 28
1.1.9 Alarm_Event_Type_ENUM_Table 29
1.1.10 APC_Domain_Table 30
1.1.11 APS_Group_Table 31
1.1.12 Audit_Log_Mapping_Table 32
1.1.13 Audit_Log_Table 32
1.1.14 BLSR_Node 32
1.1.15 BLSR_Span 32
1.1.16 BLSR_Table 33
1.1.17 Bridge_Group_Table 33
1.1.18 Cerent_NE_Group_Table 34
1.1.19 Circuit_BS_Tbl 34
1.1.20 Circuit_Ctp_Tbl 40
1.1.21 Circuit_Dest_Tbl 42
1.1.22 Circuit_Size_Table 43
1.1.23 Circuit_Span_BS_Tbl 44
1.1.24 Circuit_Span_Tbl 46
1.1.25 Circuit_Tbl 48
1.1.26 Circuit_UPSR_BS_Tbl 56
1.1.27 Circuit_VCG_Tbl 56
1.1.28 Circuit_VLAN_Tbl 57
1.1.29 CLISession_Command_History_Tbl 58
1.1.30 Connless_TP_Table 58
1.1.31 CORBA_Functions_Table 59
1.1.32 CORBA_GW_Active_User_Table 60
1.1.33 CORBA_Interface_Table 60
1.1.34 CORBA_Permission_Table 61

1.1.35 CTC_Model_Const_Table....61
1.1.36 CTC_UserProfile_Table....62
1.1.37 CTM_Config_Table....62
1.1.38 CTM_Unique_ObjectType_Table....62
1.1.39 CTM_Unknown_User_Table....63
1.1.40 Domain_Table....63
1.1.41 Ems_Alarm_Table....64
1.1.42 Enum_Table....65
1.1.43 EQPT_Info_Table....65
1.1.44 Eqpt_Relationship_Table....66
1.1.45 Error_Log_Table....67
1.1.46 EVC_EP....67
1.1.47 EVC_Node....68
1.1.48 EVC_SRV....69
1.1.49 Filter_Trial....69
1.1.50 Flow_Domain_Table....70
1.1.51 Forwarding_Table....70
1.1.52 GMPLS_VALUE_TABLE....70
1.1.53 GNE_Table....72
1.1.54 Group_Info_Table....73
1.1.55 GW_SNMP_OSS_Info_Table....73
1.1.56 IF_Cross_Connect....74
1.1.57 IF_DWDM_Channel_Group....75
1.1.58 IF_Optical_Phy_Config_Table....76
1.1.59 IFIndex_Rule_Table....77
1.1.60 Interface_APS_Info....78
1.1.61 Interface_DWDM_Frequency....79
1.1.62 Interface_Stack_Table....79
1.1.63 Interface_Table....80
1.1.64 IOS_Parser_Static_Tree_Table....80
1.1.65 L2_FrontPort_Table....81
1.1.66 L2_Interface_QoSClass_Table....82
1.1.67 L2_Interface_Table....84
1.1.68 L2_Ipsla_Attribute_Table....85
1.1.69 L2_Ipsla_Table....86
1.1.70 L2QoSTemplate_QoSClass_Table....86
1.1.71 L2QoSTemplate_Table....88
1.1.72 L2Topology_Circuits_Table....89
1.1.73 L2Topology_Table....89
1.1.74 L2_Vlan_Table....92
1.1.75 Link_Layer_Mapping_Table....92
1.1.76 Link_Table....92
1.1.77 LinkId_To_UniqueId....97

1.1.78 Managed_Et_Table 98
1.1.79 Matrix_Flow_Domain_Table....98
1.1.80 MIB2_Interface_Table 99
1.1.81 MLPPP_Bundle_Data 100
1.1.82 MLPPP_Link_Data 101
1.1.83 MLPPP_Mux_Data 101
1.1.84 Model_Type_Table 102
1.1.85 Module155xx_Community_String 103
1.1.86 Module155xx_VTOID_Table 103
1.1.87 Module_Type_Mll_Info 104
1.1.88 Module_Type_Table 104
1.1.89 MPLS_Data_Service_Table 105
1.1.90 MPLS_LSP_CTP 105
1.1.91 MPLS_LSP_XC....107
1.1.92 MPLS_NE_Table 107
1.1.93 MPLS_QoS_Action_Table....108
1.1.94 MPLS_QoS_AM_Table 109
1.1.95 MPLS_QoS_Class_Map_Table 109
1.1.96 MPLS_QoS_Conform_PA_Table 110
1.1.97 MPLS_QoS_Exceed_PA_Table 110
1.1.98 MPLS_QoS_Mapper_Table 111
1.1.99 MPLS_QoS_Pmap_Table....111
1.1.100 MPLS_QoS_Policy_Table....111
1.1.101 MPLS_QoS_Tablemap_Table 112
1.1.102 MPLS_QoS_Violate_PA_Table 112
1.1.103 MPLS_Service_Table 113
1.1.104 MPLS_TP_EP 113
1.1.105 NE_Audit_Trail_Table 115
1.1.106 NE_INFO_TABLE....115
1.1.107 NE_Inventory_Table 123
1.1.108 NE_Model_State 124
1.1.109 NE_Model_Type....124
1.1.110 NE_Port_Status_Table 124
1.1.111 NE_SNMPv3_Groups_Table....124
1.1.112 NE_SNMPv3_NFILTER_TABLE 125
1.1.113 NE_SNMPv3_PROXY_FWD_TABLE 126
1.1.114 NE_SNMPv3_PROXY_TRAP_FWD_TABLE 126
1.1.115 NE_SNMPv3_Users_Table 127
1.1.116 NE_SNMPv3_Views_Table....128
1.1.117 NE_SNMPv3_TRAP_DEST_TABLE....128
1.1.118 NE_Swimage_Table....129
1.1.119 Network_Partition_Table 130
1.1.120 NGXP_Bet_Gts_Table 130

1.1.121 NGXP_Bet_Table 131
1.1.122 Object_Relationship_Table 132
1.1.123 ObjectAttributeTable 132
1.1.124 ObjectTypeTable 134
1.1.125 ONS15216_Active_User_Table 135
1.1.126 ONS15216_NE_Inventory_Table 135
1.1.127 ONS15216_Privilege_Table 136
1.1.128 ONS15216_User_Table 136
1.1.129 ONS1530x_HO_PM_Table 137
1.1.130 ONS1530x_IP_PM_Table 137
1.1.131 ONS1530x_LO_PM_Table 139
1.1.132 ONS1530x_MS_PM_Table 140
1.1.133 ONS1530x_NE_Inventory_Table 141
1.1.134 ONS1530x_RS_PM_Table 142
1.1.135 ONS15454_327_NE_Table 142
1.1.136 ONS15454_8B10B_PM_Table 143
1.1.137 ONS15454_Active_User_Table 145
1.1.138 ONS15454_CBQoS_Info_Table 145
1.1.139 ONS15454_CLASSMAP_PM_Table 146
1.1.140 ONS15454_CMS_Storage_Table 147
1.1.141 ONS15454_COS_PM_Table 148
1.1.142 ONS15454_DS1_PM_Table 149
1.1.143 ONS15454_DS3_PM_Table 154
1.1.144 ONS15454_ENET_PM_Table 158
1.1.145 ONS15454_FC_PM_Table 167
1.1.146 ONS15454_GFP_PM_Table 171
1.1.147 ONS15454_IFX_PM_Table 173
1.1.148 ONS15454_INVENTORY_NOTES_Table 174
1.1.149 ONS15454_IPSLA_PM_Table 175
1.1.150 ONS15454_LAG_PM_Table 179
1.1.151 ONS15454_Match_STMT_PM_Table 181
1.1.152 ONS15454_NE_Inventory_Table 182
1.1.153 ONS15454_OTN_Fec_PM_Table 183
1.1.154 ONS15454_OTN_Path_PM_Table 185
1.1.155 ONS15454_OTN_Sec_PM_Table 187
1.1.156 ONS15454_PHY_LAYER_PM_Table 189
1.1.157 ONS15454_Police_PM_Table 193
1.1.158 ONS15454_POS_PM_Table 194
1.1.159 ONS15454_Privilege_Table 198
1.1.160 ONS15454_RPR_PM_Table 198
1.1.161 ONS15454_SONET_Line_PM_Table 205
1.1.162 ONS15454_SONET_Path_PM_Table 209
1.1.163 ONS15454_SONET_Sec_PM_Table 212

1.1.164 ONS15454_SONET_VT_PM_Table....213
1.1.165 ONS15454_User_Table....215
1.1.166 ONS15454SDH_E1_PM_Table....216
1.1.167 ONS15454SDH_E3_PM_Table....221
1.1.168 ONS15454SDH_E4_PM_Table....223
1.1.169 ONS15454SDH_HO_PM_Table....224
1.1.170 ONS15454SDH_LO_PM_Table....228
1.1.171 ONS15454SDH_MS_PM_Table....231
1.1.172 ONS15454SDH_RS_PM_Table....236
1.1.173 ONS155xx_CDL_Flow_Config....237
1.1.174 ONS155xx_CDL_IF_Config....238
1.1.175 ONS155xx_CDL_PM_Table....238
1.1.176 ONS155xx_Circuit_CC_Tbl....240
1.1.177 ONS155xx_Circuit_Path_Tbl....242
1.1.178 ONS155xx_Circuit_Stack_Tbl....242
1.1.179 ONS155xx_Condtype_Map....243
1.1.180 ONS155xx_DCRP_Definition_Table....244
1.1.181 ONS155xx_DCRP_Table....244
1.1.182 ONS155xx_Discovery_Info_Table....245
1.1.183 ONS155xx_Ether_Hist_PM_Table....246
1.1.184 ONS155xx_FCM_PE_PM_Table....247
1.1.185 ONS155xx_Flash_Device_Table....249
1.1.186 ONS155xx_Flash_Partition_Table....250
1.1.187 ONS155xx_GRA_CC_To_CKT_CC_Tbl....251
1.1.188 ONS155xx_Granular_CC_Tbl....251
1.1.189 ONS155xx_IF_Circuit_Params_Tbl....254
1.1.190 ONS155xx_IF_Optical_Power....254
1.1.191 ONS155xx_IF_Phy_Info_Table....255
1.1.192 ONS155xx_NE_Info_Table....257
1.1.193 ONS155xx_NE_Type_Table....257
1.1.194 ONS155xx_Opt_Power_PM_Table....258
1.1.195 ONS155xx_Path_Hop_Tbl....261
1.1.196 ONS155xx_Path_Tbl....262
1.1.197 ONS155xx_Phy_PM_Table....263
1.1.198 ONS155xx_SONET_Sec_PM_Table....265
1.1.199 ONS15600_ENET_PM_Table....266
1.1.200 ONS15600_POS_PM_Table....269
1.1.201 ONS15600_Sonet_Line_PM_Table....271
1.1.202 ONS15600_Sonet_Path_PM_Table....271
1.1.203 ONS15600_Sonet_Sec_PM_Table....271
1.1.204 ONS15600SDH_HO_PM_Table....272
1.1.205 ONS15600SDH_MS_PM_Table....272
1.1.206 ONS15600SDH_RS_PM_Table....272

1.1.207 Operation_Permissions_Table 272
1.1.208 Operations_Table 272
1.1.209 OSS_Active_User_Table 276
1.1.210 OSS_AID_ENUM_Table 277
1.1.211 OSS_Alarm_Filter_Table 277
1.1.212 OSS_Alarm_Fltr_AID_Deny_Tbl 278
1.1.213 OSS_Alarm_Fltr_TID_Deny_Tbl 279
1.1.214 OSS_CORBA_User_Table 280
1.1.215 OSS_User_Table 280
1.1.216 Password_History_Table 281
1.1.217 PatchCord_Table 281
1.1.218 PM_ATTR_Value_Mapping_Table 282
1.1.219 PM_ATTRID_Mapping_Table 282
1.1.220 PM_Category_View_Table 283
1.1.221 PM_Cerent_Info_Table 283
1.1.222 PM_Historical_Appointments 284
1.1.223 PM_Metadata_Info_Table 285
1.1.224 PM_Metadata_Table 285
1.1.225 Port_Channel_Table 286
1.1.226 Port_Channel_Members_Table 287
1.1.227 Proxy_Server_Table 288
1.1.228 PW_EFP 288
1.1.229 PW_EP 289
1.1.230 PW_PP 290
1.1.231 PW_SRV 291
1.1.232 Roll_Table 292
1.1.233 Schedule_Info_Table 294
1.1.234 Server_Monitor_Table 295
1.1.235 Server_Parameter_Table 296
1.1.236 Service_Availability_Table 296
1.1.237 Subnetwork_Table 297
1.1.238 SVC_Config_Table 298
1.1.239 SVC_Info_Table 299
1.1.240 SVC_NEModel_Table 299
1.1.241 SVC_Package_Table 300
1.1.242 SVC_Property_Table 300
1.1.243 Sway_User_Port_Table 301
1.1.244 Table_Info 301
1.1.245 TCA_Event_Table 301
1.1.246 TCA_Event_Type_Table 302
1.1.247 Threshold_Table 303
1.1.248 TL1_Encoding_Table 303
1.1.249 TL1_Tunnel_Gateway_Table 304

1.1.250 TL1_Tunnel_Info_Table 304
1.1.251 Transaction_Log_Table 305
1.1.252 TP_SRV 306
1.1.253 Unmanaged_NE_Inventory_Table 307
1.1.254 User_Map_Table 307
1.1.255 User_NE_Table 307
1.1.256 User_Preference_Table 308
1.1.257 User_Table 308
1.1.258 User_Type_Table 310
1.1.259 Version_Table 311
1.1.260 Wavelength_Table 315
1.1.261 Wavemux_NE 316
1.1.262 1.1.264 XML_Parser_Static_Tree_Table 316
2 Database Schema Views 317
2.1 Common PM Views 317
2.1.1 NE_Module_Level_Debug_View 317
2.1.2 SDH_HO_FarEnd_PM_View 317
2.1.3 SDH_HO_NearEnd_PM_View 318
2.1.4 SDH_MS_FarEnd_PM_View 319
2.1.5 SDH_MS_NearEnd_PM_View 319
2.1.6 SDH_RS_NearEnd_PM_View 320
2.1.7 SONET_ENET_PM_View 320
2.1.8 SONET_Line_FarEnd_PM_View 324
2.1.9 SONET_Line_NearEnd_PM_View 325
2.1.10 SONET_Physical_PM_View 325
2.1.11 SONET_POS_PM_View 326
2.1.12 SONET_Section_NearEnd_PM_View 328
2.1.13 SONET_STSPath_FarEnd_PM_View 329
2.1.14 SONET_STSPath_NearEnd_PM_View 329
2.2 Prime Optical Base Views 330
2.2.1 Active_Alarm_View_With_Ack 330
2.2.2 Admin_Job_View 331
2.2.3 Alarm_Count_View (Deprecated) 332
2.2.4 Alarm_Log_View 332
2.2.5 Audit_Log_View 334
2.2.6 BLSR_Table_View 334
2.2.7 Children_View 334
2.2.8 Circuit_Ctpspan_View 334
2.2.9 Circuit_Ctp_Drop_View 335
2.2.10 Circuit_Ctp_Sec_View 335
2.2.11 Circuit_Ctp_Src_View 335
2.2.12 Circuit_Ctp_View 336
2.2.13 Circuit_Ctp_XC_View 336

2.2.14 Circuit_Node_GUI_View 336
2.2.15 Circuit_Node_GUI_View_CTM46....338
2.2.16 Circuit_Node_View 339
2.2.17 Circuit_Span_View 340
2.2.18 Circuit_Span_View_CTM46....341
2.2.19 Circuit_UPSR_View 341
2.2.20 Circuit_View 342
2.2.21 Circuit_VLAN_View 343
2.2.22 Ckt_Span_Dest_Mt_View 343
2.2.23 Ckt_Span_Src_Mt_View....344
2.2.24 Client_User_View....344
2.2.25 CTM_Unknown_User_View 344
2.2.26 Data_Card_View 344
2.2.27 Domain_NE_Inventory_View....345
2.2.28 DSP_Circuit_View....345
2.2.29 DSP_Eqpt_View....346
2.2.30 GWCORBA_PM_Mapping_View 346
2.2.31 INTF_Sub_Intf_View 346
2.2.32 Inventory_Table_View 347
2.2.33 IOS_Active_Cards 347
2.2.34 IOS_Users_View 347
2.2.35 Link_Utilization_View 348
2.2.36 Mcard_L2topology_Map_View 348
2.2.37 MLBCC_View 349
2.2.38 MLL_Inv_View 350
2.2.39 NE_Alarm_View 351
2.2.40 NE_Circuit_NodeID_View 351
2.2.41 NE_Circuit_View 351
2.2.42 NE_Circuit_Vlan_View 352
2.2.43 NE_Info_NP_View 352
2.2.44 NE_Inventory_View 353
2.2.45 NE_SNMPV3_FILTERS_VIEW....353
2.2.46 NE_SNMPV3_GROUPS_VIEW 353
2.2.47 NE_SNMPV3_LOCAL_USER_VIEW 354
2.2.48 NE_SNMPV3_PROXY_FWD_VIEW 354
2.2.49 NE_SNMPV3_PROXY_TRAP_FWD_VIEW....354
2.2.50 NE_SNMPV3_REMOTE_USER_VIEW 355
2.2.51 NE_SNMPV3_TRAP_DEST_VIEW 356
2.2.52 NE_SNMPV3_VIEWS_VIEW 356
2.2.53 NE_Swimage_View 356
2.2.54 NGNE_Info_View 357
2.2.55 OSS_Alarm_Fltr_AidDeny_View 357
2.2.56 OSS_Alarm_Fltr_TidDeny_VIEW....357

2.2.57 OSS_Corba_User_View 358
2.2.58 OSS_User_View 358
2.2.59 POS_Ports_View 358
2.2.60 QoSTemplate_General_View 358
2.2.61 QoS_View 358
2.2.62 Roll_View 359
2.2.63 Self_Monitoring_View 359
2.2.64 Service_Availability_Log_View 360
2.2.65 SPAN_LINK_View 360
2.2.66 SPR_View 361
2.2.67 SW_Active_Alarm_View 361
2.2.68 SW_User_Port_View 361
2.2.69 TL1_Active_Alarm_View 362
2.2.70 TL1_NE_View 362
2.2.71 Topology_Circuit_View 363
2.2.72 Topology_Circuit_View_CTM5_0 365
2.2.73 Topology_GUI_View 366
2.2.74 Trace_Vlan_View 367
2.2.75 Unmanaged_NE_Inventory_View 367
2.2.76 User_NE_View 368
2.2.77 User_View 368
2.2.78 Vlan_Drop_Ports_QoS_View 368
2.2.79 Vlan_Drop_Ports_View 369
2.2.80 Vlan_Service_View 370
2.3 Cisco ONS 152xx Views 370
2.4 Cisco ONS 15216 Views 371
2.4.1 ONS15216_Active_User_View 371
2.4.2 ONS15216_Inventory_View 371
2.4.3 ONS15216_User_View 371
2.5 Cisco ONS 153xx Views 372
2.5.1 ONS1530x_Dcc_PM_View 372
2.5.2 ONS1530x_HO_FE_PM_View 372
2.5.3 ONS1530x_HO_PM_View 373
2.5.4 ONS1530x_Inventory_View 373
2.5.5 ONS1530x_Lan_PM_View 373
2.5.6 ONS1530x_LO_FE_PM_View 374
2.5.7 ONS1530x_LO_PM_View 374
2.5.8 ONS1530x_MS_FE_PM_View 375
2.5.9 ONS1530x_MS_PM_View 375
2.5.10 ONS1530x_RS_PM_View 376
2.5.11 ONS1530x_Wan_PM_View 376
2.6 Cisco ONS 15454 Views 377
2.6.1 APC_DOMAIN_NE_VIEW 377

2.6.2 Ctc_User_Profiles_View....377
2.6.3 L2_IPSLA_View....377
2.6.4 L2_Topology_Cards_View....377
2.6.5 NE_Audit_Trail_View....378
2.6.6 ONS15454SDH_E1_PM_FE_View....378
2.6.7 ONS15454SDH_E1_PM_View....379
2.6.8 ONS15454SDH_E3_PM_View....380
2.6.9 ONS15454SDH_E4_PM_View....380
2.6.10 ONS15454SDH_HO_PM_FE_View....381
2.6.11 ONS15454SDH_HO_PM_View....381
2.6.12 ONS15454SDH_LO_VC11_PM_FE_View....382
2.6.13 ONS15454SDH_LO_VC11_PM_View....382
2.6.14 ONS15454SDH_LO_VC12_PM_FE_View....383
2.6.15 ONS15454SDH_LO_VC12_PM_View....384
2.6.16 ONS15454SDH_LO_VC3_PM_FE_View....384
2.6.17 ONS15454SDH_LO_VC3_PM_View....385
2.6.18 ONS15454SDH_MS_PM_FE_View....385
2.6.19 ONS15454SDH_MS_PM_View....386
2.6.20 ONS15454SDH_RS_PM_View....387
2.6.21 APC_DOMAIN_VIEW....387
2.6.22 ONS15454_8B10B_PM_View....387
2.6.23 ONS15454_Active_User_View....388
2.6.24 ONS15454_APC_Result_View....388
2.6.25 ONS15454_BLSR_Config_View....389
2.6.26 ONS15454_BLSR_Squelch_View....389
2.6.27 ONS15454_CMS_Storage_View....389
2.6.28 ONS15454_COS_PM_View....390
2.6.29 ONS15454_DS1_PM_FE_View....390
2.6.30 ONS15454_DS1_PM_View....391
2.6.31 ONS15454_DS3_PM_FE_View....392
2.6.32 ONS15454_DS3_PM_View....392
2.6.33 ONS15454_DWDM_Ring_View....393
2.6.34 ONS15454_Enet_PM_View....393
2.6.35 ONS15454_ESCON_PM_View....396
2.6.36 ONS15454_FC_PM_View....396
2.6.37 ONS15454_GFP_PM_View....397
2.6.38 ONS15454_Inventory_View....398
2.6.39 ONS15454_IPSLA_PM_View....398
2.6.40 ONS15454_LEX_PM_View....400
2.6.41 ONS15454_ISC_PM_View....400
2.6.42 ONS15454_OPTICS_PM_VIEW....401
2.6.43 ONS15454_OSPF_Area_Range_View....401
2.6.44 ONS15454_OTN_Fec_PM_View....402

2.6.45 ONS15454_OTN_Path_PM_FE_View 402
2.6.46 ONS15454_OTN_Path_PM_View 403
2.6.47 ONS15454_OTN_Sec_PM_FE_View 403
2.6.48 ONS15454_OTN_Sec_PM_View 404
2.6.49 ONS15454_Phy_Layer_PM_View 404
2.6.50 ONS15454_POS_PM_View 405
2.6.51 ONS15454_Ring_Map_View 406
2.6.52 ONS15454_RPR_Client_In_PM_View 406
2.6.53 ONS15454_RPR_ClientOut_PM_View 407
2.6.54 ONS15454_RPR_Error_PM_View 408
2.6.55 ONS15454_RPR_PM_View 408
2.6.56 ONS15454_RPR_Span_In_PM_View 410
2.6.57 ONS15454_RPR_Span_Out_PM_View 411
2.6.58 ONS15454_Security_View 411
2.6.59 ONS15454_SNMP_Trap_Dest_View 412
2.6.60 ONS15454_SONET_Line_PM_FE_View 412
2.6.61 ONS15454_SONET_Line_PM_View 413
2.6.62 ONS15454_SONET_Path_PM_FE_View 413
2.6.63 ONS15454_SONET_Path_PM_View 414
2.6.64 ONS15454_SONET_Section_PM_View 415
2.6.65 ONS15454_SONET_VT2_PM_FE_View 415
2.6.66 ONS15454_SONET_VT2_PM_View 416
2.6.67 ONS15454_SONET_VT_PM_FE_View 416
2.6.68 ONS15454_SONET_VT_PM_View 417
2.6.69 ONS15454_Spanning_Trees_View 417
2.6.70 ONS15454_Static_Route_View 417
2.6.71 ONS15454_Threshold_View 418
2.6.72 ONS15454_User_View 418
2.6.73 ONS15454_Virtual_Link_View 419
2.6.74 PM_Param_ColNo_View 419
2.6.75 PM_Metadata_Attrid_View 419
2.6.76 PM_Metadata_Iinfo_View 419
2.6.77 PM_Metadata_View 420
2.7 Cisco ONS 155xx Views 420
2.7.1 ONS155xx_CDL_Flow_Config_View 420
2.7.2 ONS155xx_CDL_PM_View 421
2.7.3 ONS155xx_Circuit_CC_View 421
2.7.4 ONS155xx_Circuit_Dest_View 422
2.7.5 ONS155xx_Circuit_Final_View 422
2.7.6 ONS155xx_Circuit_Path_View 423
2.7.7 ONS155xx_Circuit_Segment_View 424
2.7.8 ONS155xx_Circuit_Span_View 424
2.7.9 ONS155xx_Circuit_SrcDest_View 425

2.7.10 ONS155xx_Circuit_Stack_View 426
2.7.11 ONS155xx_Circuit_View 426
2.7.12 ONS155xx_CKT_CC_Dest_MT_View 427
2.7.13 ONS155xx_CKT_CC_Detailed_View 427
2.7.14 ONS155xx_CKT_CC_Link_View 427
2.7.15 ONS155xx_CKT_CC_Path_View 428
2.7.16 ONS155XX_CKT_CC_Src_Dest_View 428
2.7.17 ONS155xx_CKT_CC_Src_MT_View 429
2.7.18 ONS155xx_CKT_Endpoint_View 429
2.7.19 ONS155XX_CKT_Link_Final_View 430
2.7.20 ONS155xx_CKT_Link_View 430
2.7.21 ONS155XX_CKT_NE_Final_View 431
2.7.22 ONS155xx_CKT_OPT_Pwr_View 431
2.7.23 ONS155xx_CKT_Path_Final_View 431
2.7.24 ONS155xx_CKT_Path_Join_View 432
2.7.25 ONS155xx_CKT_Path_Stack_View 433
2.7.26 ONS155xx_CKT_Segment_Path_View 433
2.7.27 ONS155xx_CKT_SEG_Detailed_View 434
2.7.28 ONS155xx_CKT_Span_Dest_MT_View 434
2.7.29 ONS155xx_CKT_Span_Det_View 435
2.7.30 ONS155xx_CKT_Span_Join_View 435
2.7.31 ONS155xx_CKT_Span_MT_View 436
2.7.32 ONS155xx_CKT_Span_Src_MT_View 437
2.7.33 ONS155xx_CKT_Span_View 437
2.7.34 ONS155xx_CKT_Srcdest_MT_View 438
2.7.35 ONS155xx_CKT_SRC_CDL_View 439
2.7.36 ONS155XX_CKT_Stack_Final_View 439
2.7.37 ONS155xx_Discovery_Info_View 440
2.7.38 ONS155xx_Dcrp_View 440
2.7.39 ONS155xx_Ether_Hist_PM_View 441
2.7.40 ONS155xx_FCM_PE_PM_View 441
2.7.41 Ons155xx_Flash_Part_View 442
2.7.42 ONS155xx_Inventory_View 442
2.7.43 ONS155XX_Link_Correlator 443
2.7.44 ONS155xx_Module_Type_View 444
2.7.45 ONS155xx_OPT_Power_PM_View 444
2.7.46 ONS155XX_Path_CKT_Final_View 445
2.7.47 ONS155xx_Path_Dest_Pwr_View 445
2.7.48 ONS155XX_Path_Link_Final_View 446
2.7.49 ONS155xx_Path_Link_View 447
2.7.50 ONS155xx_Path_Pwr_Monitor_View 447
2.7.51 ONS155xx_Path_Pwr_No_Ava_View 447
2.7.52 ONS155xx_Path_Pwr_No_Mon_View 448

2.7.53 ONS155xx_Path_Stack_Final_View 448
2.7.54 ONS155xx_Phy_PM_View 449
2.7.55 ONS155xx_SONET_Section_PM_View 449
2.7.56 ONS155xx_Tca_PM_View 450
2.8 Cisco ONS 156xx Views 450
2.8.1 ONS15600sdh_HO_PM_FE_View 450
2.8.2 ONS15600sdh_HO_PM_View 451
2.8.3 ONS15600sdh_MS_PM_FE_View 451
2.8.4 ONS15600sdh_MS_PM_View 452
2.8.5 ONS15600sdh_RS_PM_View 452
2.8.6 ONS15600_Enet_PM_View 453
2.8.7 ONS15600_Pos_PM_View 454
2.8.8 ONS15600_SONET_Line_FE_PM_View 454
2.8.9 ONS15600_SONET_Line_PM_View 455
2.8.10 ONS15600_SONET_Path_FE_PM_View 456
2.8.11 ONS15600_SONET_Path_PM_View 456
2.8.12 ONS15600_SONET_Sec_PM_View 457
3 Database Schema Functions 458
3.1 Prime Optical Functions 458
3.1.1 CollapsedDestNodeName 458
3.1.2 CollapsedDestNodeName 458
3.1.3 CollapsedEqptPortName 459
3.1.4 CollapsedIfIndex 459
3.1.5 CollapsedIfIndex 460
3.1.6 CollapsedModelType 460
3.1.7 CollapsedModuleName 460
3.1.8 CollapsedObjectIndex 461
3.1.9 CollapsedPhy 461
3.1.10 CollapsedPort 462
3.1.11 CollapsedPortModuleName 462
3.1.12 ConvertAxxIfIndex 462
3.1.13 ConvertLinkModuleType 463
3.1.14 CountCoSPh 463
3.1.15 CTMDuration 463
3.1.16 DisplayIf 464
3.1.17 DisplayIPAll 466
3.1.18 DisplayIP 466
3.1.19 DisplayPhyLoc 467
3.1.20 EqptInfoPort 467
3.1.21 Get_Enet_Value 469
3.1.22 Get_Enet_Value2 470
3.1.23 GetBccBwInfo 472
3.1.24 GetBccClassInfo 473

3.1.25 GetBccClassRPR80217 473
3.1.26 GetBccCosCommitGrp 473
3.1.27 GetBccCosInfo 474
3.1.28 GetBGFromBGObj 474
3.1.29 GetBGFromQoSObj 474
3.1.30 GetCktDestNodeId 475
3.1.31 GetCktSecSrcIfIndex 475
3.1.32 GetCktSecSrcPhysicalLoc 475
3.1.33 GetCktSecSrcPortName 475
3.1.34 GetCktSrcIfIndex 476
3.1.35 GetCktSrcPhysicalLoc 476
3.1.36 GetCktSrcPortName 476
3.1.37 GetCollapsedPortname 476
3.1.38 GetEqptInfoPhysicalLoc 477
3.1.39 GetIfCosGrp 477
3.1.40 GetInterfaceId 478
3.1.41 Get_Inventory_Note 479
3.1.42 GetLinkType 479
3.1.43 GetLowBits 481
3.1.44 GetMgx88xxMgmtState 481
3.1.45 GetMgx88xxSyncMode 481
3.1.46 GetMLcardAddnlInfo 481
3.1.47 GetModelTypeFromPhysicalLoc 482
3.1.48 GetModuleName 482
3.1.49 GetModuleName_PM 482
3.1.50 GetModuleName1 483
3.1.51 GetModuleName2 483
3.1.52 GetgetProtectMLcardConfigState 484
3.1.53 GetNEDBAccessID 484
3.1.54 GetNEIPAddr 484
3.1.55 GetObjectTypeFromIfIndex 485
3.1.56 GetObjectTypeFromIfIndex2 485
3.1.57 GetOutIPAddr 485
3.1.58 GetPhysicalLoc 486
3.1.59 GetPhysicalLocWithPIMPPM 486
3.1.60 GetPIM 486
3.1.61 GetPMParameterValue 486
3.1.62 GetPolicyMapType 487
3.1.63 GetPort 487
3.1.64 GetPort_XTC 487
3.1.65 GetPortModuleName 488
3.1.66 GetPortModuleName2 488
3.1.67 GetPortName 489

3.1.68 GetPortWithPIMPPM 490
3.1.69 GetPosPortState 490
3.1.70 GetPosPortStateForRPR802 490
3.1.71 GetPPM 491
3.1.72 GetProtectMLCardInfo 491
3.1.73 GetQoSPhLoc 491
3.1.74 GetShelf 492
3.1.75 GetSlot 492
3.1.76 GetSlotModuleName 492
3.1.77 GetSlotModuleType 492
3.1.78 GetSpanTblCktSpanDestNodeId 493
3.1.79 GetSpanTblCktSpanSrcNodeId 493
3.1.80 GetSubIf 493
3.1.81 GetValidCktSecSrcNodeId 494
3.1.82 GetValidCktSrcNodeId 494
3.1.83 GetValidModelType 494
3.1.84 GetValidModuleType 494
3.1.85 GetValidObjectType 495
3.1.86 Has_Next 495
3.1.87 IS_NUMBER 495
3.1.88 IsIPv6 495
3.1.89 MapObjectIndex 496
3.1.90 MergedInvalidityList 499
3.1.91 NE_Belongs_to_APC_Domain 500
3.1.92 RemoveFirstByteIf 501
3.1.93 RemoveLowBits 501
3.1.94 Time_Diff 501
3.1.95 TYPE AlarmInfoListType 502
3.1.96 TYPE AlarmInfoType 502
3.1.97 TYPE Link_Id_Tabtype 502
A. Oracle Built-In Data Types 503
B. Config_Table Parameters 504

Preface

Objectives

This document describes the database schema that Cisco Prime Optical uses to store information in a Structured Query Language (SQL) database such as the Oracle database. The document is designed for users who need to create their own reports without using Prime Optical.

The Prime Optical schema is subject to change.

Various schema objects (tables, views, and functions) form the schema for Prime Optical in the Oracle database. The following information is stored in the Oracle database:

- Network element (NE) information
- Gateway network element (GNE) information
- Topology information
- Interface tables
- Performance monitoring (PM) data
- SONET and SDH PM data
- Active alarm information
- Alarm and event logs
- Log tables
- Administrative job and task information
- Port status
- NE service-related configuration information

Organization

This document contains the following information:

- Database Schema Tables
  - Table descriptions.
- Database Schema Views
  - Descriptions of the database views created and used by Prime Optical.
- Database Schema Functions
  - Functions defined in Prime Optical tables.
- Oracle Built-In Data Types
  - List of Oracle data types.
- Config_Table Parameters
  - Configuration information.

Audience

The primary audience for this guide includes network operations personnel and system administrators. This guide assumes the reader is familiar with the following products and topics:

- Basic internetworking terminology and concepts
- Network topology and protocols
- Microsoft Windows 7, Windows Vista, Windows 2000 Professional, Windows XP Professional, and Windows Server 2003 Terminal Server
- Sun Microsystems Solaris or Linux Red Hat administration
- Oracle database administration

Conventions

This document uses the conventions listed in Table 1.

Table 1: Document Conventions

Item	Convention
Commands and keywords	**boldface** font
Variables for which you supply values	*italic* font
Displayed session and system information	screen font
Information you enter	**boldface screen** font
Variables you enter	*italic screen* font
Menu items and button names	**boldface** font
Selecting a menu item	**Option > Network Preferences**

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

! Means the following information will help you solve a problem.

Related Documentation

You can access the most current Prime Optical documentation online at http://www.cisco.com/en/US/products/ps11670/tsd_products_support_series_home.html.

The Prime Optical documentation set comprises the following guides:

- Release Notes for Cisco Prime Optical 9.6.3—Describes the caveats for Prime Optical.
- Cisco Prime Optical 9.6.3 Installation Guide—Explains how to install Prime Optical and how to upgrade from previous releases.
- Cisco Prime Optical 9.6.3 User Guide—Describes how to use the Prime Optical software, which consists of user applications and tools for network discovery, network configuration, connection management, fault management, system administration, and security management.
- Cisco Prime Optical 9.6.3 GateWay/CORBA User Guide and Programmer Manual—Describes the GateWay/CORBA northbound interface product that is available for Prime Optical. This document serves as a reference for developers of OSS applications that work with the GateWay/CORBA interface.
- Cisco Prime Optical 9.6.3 Database Schema—This guide.
- Cisco Prime Optical 9.6.3 High Availability Installation Guide—Explains how to install Prime Optical in a high availability (HA) environment.

  To obtain the *Cisco Prime Optical 9.6.3 High Availability Installation Guide*, contact your Cisco account representative.

- Cisco Prime Optical 9.6.3 ML Provisioning Methodology—Describes the methodology that Prime Optical uses to provision ML-series cards.
- Cisco Prime Optical 9.6.3 Basic External Authentication—Describes how Prime Optical supports basic external authentication.
- Migration Matrix for Service Pack Releases—Describes the migration matrix for service pack releases.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

.

1 Database Schema Tables

This chapter describes the database schema tables. It contains the following section:

- Prime Optical Tables

1.1 Prime Optical Tables

This section describes the database tables created by Prime Optical, listed in alphabetical order.

The following tables are partitioned in medium, large, and high-end configurations:

- ONS 1530x PM tables
- ONS 15454 PM tables
- ONS 155xx PM tables
- ONS 15600 PM tables

1.1.1 Ack_Table

The Ack_Table provides information about the user who acknowledged an alarm and the time that the alarm was acknowledged.

Primary key: ALARM_ACK_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-1: Ack_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
AckIndex	NUMBER(10)	—	Index of the alarm acknowledgement. A unique sequence number whose value is saved in the database and incremented in memory. There is a single value of 0 for all unacknowledged alarms. Acknowledged alarms have an entry in this table and the AckIndex is stored in the AckIndex of the Active_Alarm_Table. Values are: 0: Unacknowledged Any other number: Acknowledged –1: Automatic Acknowledge
AckUserName	VARCHAR2(64)	—	Name of the user.

Column Name	Oracle Data Type	Integrity Constraints	Description
AckTimeStamp	DATE	—	Date and time of the acknowledgement.
IsAutoAck	NUMBER(1)	—	Whether acknowledgment is manual or automatic: 0: Manual 1: Automatic

1.1.2 Active_Alarm_Table

The Active_Alarm_Table contains information about all standing conditions on NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ACTIVE_ALARM_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-2: Active_Alarm_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
AlarmSeqNum	NUMBER(20)	NOT NULL	Unique sequence number of the alarm as referenced by the NE.
ModuleOrIfIndex	NUMBER(20)	NOT NULL	Unique long data type that identifies an interface in the NE; same as ifStaticIndex in the Interface_Table.
ActiveAlarmTimeStamp	DATE	NOT NULL	Date and time when the active alarm was generated.
ActiveAlarmType	NUMBER(5)	NOT NULL	Alarm event type. Values are rows of the Alarm_Event_Type_Enum_Table.
ActiveAlarmSeverity	NUMBER(1)	NOT NULL	Severity level of the alarm. Values are: 3: Not alarmed 4: Warning 5: Minor 6: Major 7: Critical
ActiveAlarmServEff	NUMBER(1)	NOT NULL	Values are: 0: Alarm does not affect service 1: Alarm affects service 2: Unknown
ActiveAlarmAdditionalInfo	VARCHAR2(512)	—	Additional information about the current alarm.

Column Name	Oracle Data Type	Integrity Constraints	Description
AckIndex	NUMBER(10)	NOT NULL	Index of the alarm acknowledgement.
ActiveAlarmFlag	NUMBER(1)	DEFAULT 0	Active alarm flag, which the server uses to synchronize the database to the NE information.
ActiveAlarmComment	VARCHAR2(2014)	—	Alarm comments.
ClearAlarmTimeStamp	DATE	DEFAULT NULL	Date and time when the alarm was cleared.
NEAlarmTimeStamp	DATE	NOT NULL	Date and time when the alarm was generated as viewed by the NE.
ActiveAlarmIndex	NUMBER(16)	NOT NULL	Unique alarm sequence number as created by the database.
ModuleType	NUMBER(8)	DEFAULT 0	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
Alarmstatus	NUMBER(1)	NOT NULL	Indicates whether an alarm is active or clear. Values are: 0: Active 1: Clear
NEAlarmClearTimeStamp	DATE	—	When the NE alarm was cleared.
ExternalCondition	VARCHAR2(1024)	—	External environment condition for the NEs.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjectInstance	VARCHAR2(256)	—	StrObjInstance name.
Direction	VARCHAR2(8)	DEFAULT NULL	TL1 direction.
Location	VARCHAR2(4)	DEFAULT NULL	TL1 location.

1.1.3 Additional_Device_Info

The Additional_Device_Info contains information on additional devices.

Primary key: ADDITIONAL_DEVICE_INFO

Tablespace: BASE_DATA_TABLESPACE

Table 1-3: Additional_Device_Info

Column Name	Oracle Data Type	Integrity Constraints	Description
DeviceID	NUMBER(10)	NOT NULL	Unique ID generated by the database.

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access the Additional_Device_Info table; same as NEDBAccessID in the NE_INFO_TABLE.
PhyLoc	NUMBER(20)	NOT NULL	Physical location of the card.
ModuleType	NUMBER(8)	NOT NULL	Module type of the additional device.
State	NUMBER(1)	NOT NULL	State of the additional device. Values are: 1: Device In Service 2: Device Out of Service 3: Device Preprovisioned 4: Device Under Maintenance
isConnected	NUMBER(1)	NOT NULL	Values are: 0: Unreachable 1: Device is reachable from Prime Optical server
DiscoveryState	NUMBER(1)	—	Device discovery state. Values are: 1: Device undiscovered 2: Device Initializing 3: Device synchronization configuration 4: Device discovered

1.1.4 Admin_Job_Table

The Admin_Job_Table contains information about all administrative tasks and their status.

Primary key: ADMIN_JOB_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-4: Admin_Job_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
JobID	NUMBER(12)	NOT NULL	Unique job identifier.
TaskID	NUMBER(12)	NOT NULL	Unique identifier for the task within a job.

Column Name	Oracle Data Type	Integrity Constraints	Description
JobType	NUMBER(2)	NOT NULL	Type of job. Valid types are: 0: NE User Add 1: NE User Modify 2: NE User Delete 3: NE User Logout 4: NE User Conn Delete 5: NE User Enable Log 6: NE User Disable Log 7: NE User Delete Log 8: Image Activate 9: Image Revert 10: Image Download 11: Memory Auto Backup 12: Memory Backup 13: Memory Restore 14: Image Download and Activate 15: SNTP Setup Job 22: BLSR Exercise Ring Backup
JobOwner	VARCHAR2(64)	—	Owner of job.
NEDBAccessID	NUMBER(10)	—	NEDBAccessID of the NE for which the job was scheduled; same as NEDBAccessID in the NE_INFO_TABLE.
JobStatus	NUMBER(1)	—	Status of the job. Values are: 0: Uninitialized 1: Queued 2: Running 3: Cancelled 4: Failed 5: Succeeded 6: Unknown 7: Waiting
CreationTime	DATE	—	Time stamp when the job was created.
ScheduledTime	DATE	—	Time stamp when the job was scheduled to run.
StartTime	DATE	—	Time stamp when the job was actually started.
EndTime	DATE	—	Time stamp when the job was completed.
AdditionalInfo	VARCHAR2(4000)	—	Any additional information.
UserComments	VARCHAR2(2048)	—	Any comments entered by the user.
operation_info	VARCHAR2(4000)	—	List of parameters specific to a job.
display_info	VARCHAR2(4000)	—	String that displays job-specific information according to specific

Column Name	Oracle Data Type	Integrity Constraints	Description
			criteria.

1.1.5 Alarm_Ack_Config_Table

The Alarm_Ack_Config_Table stores information about whether or not automatic alarm acknowledgement is enabled.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-5: Alarm_Ack_Config_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ConfigAlarmAck	NUMBER(1)	NOT NULL	A global, C-style, Boolean variable that represents whether alarm acknowledgement is on for all alarms. Values are: 0: Automatic Acknowledge 1: Manual Acknowledge

1.1.6 Alarm_Count_Table

The Alarm_Count_Table stores the alarm count for each level of severity for every NE.

Primary key: ALARM_COUNT_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-6: Alarm_Count_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
nedbaccessid	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ackalarmcount	NUMBER(20)	DEFAULT 0	Acknowledged alarm count.
clearedalarmcount	NUMBER(20)	DEFAULT 0	Cleared alarm count.
minoralarmcount	NUMBER(20)	DEFAULT 0	Minor alarm count.

Column Name	Oracle Data Type	Integrity Constraints	Description
majoralarmcount	NUMBER(20)	DEFAULT 0	Major alarm count.
criticalalarmcount	NUMBER(20)	DEFAULT 0	Critical alarm count.
warningalarmcount	NUMBER(20)	DEFAULT 0	Warning alarm count.

1.1.7 Alarm_Event_Cause_Table

The Alarm_Event_Cause_Table statically maps alarm and event cause indexes to strings.

Primary key: ALARM_EVENT_CAUSE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-7: Alarm_Event_Cause_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
EnumIndex	NUMBER(4)	—	Index of alarm or event cause.
EnumString	VARCHAR2(64)	—	Alarm or event cause string for the EnumIndex.

1.1.8 Alarm_Event_Table

The Alarm_Event_Table contains information about all events reported by NEs.

Primary key: None

Tablespace: EVENT_DATA_TABLESPACE

Table 1-8: Alarm_Event_Cause_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
EventIndex	NUMBER(10)	NOT NULL	Index of alarm or event. Unique sequence number generated by the database.
ModuleOrIfIndex	NUMBER(20)	NOT NULL	Unique integer that identifies an interface in the NE; same as ifStaticIndex in the Interface_Table.

Column Name	Oracle Data Type	Integrity Constraints	Description
AlarmEventType	NUMBER(5)	NOT NULL	Type of alarm or event. The Alarm_Event_Type_ENUM_Table lists valid types. The range is from 1 to n.
AlarmEventTimeStamp	DATE	NOT NULL	Date and time when the alarm or event was generated.
AlarmEventCause	NUMBER(4)	NOT NULL	Cause of alarm or event. The Alarm_Event_Cause_Table lists valid types. The range is from 1 to n.
AlarmEventSeverity	NUMBER(1)	NOT NULL	Severity level of the alarm.
ServiceEffecting	NUMBER(1)	NOT NULL	Values are: 0: Alarm does not affect service 1: Alarm affects service 2: Unknown
OtherInfo	VARCHAR2(512)	DEFAULT Not applicable	Display string of alarm or event.
ModuleType	NUMBER(8)	DEFAULT 0	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEEventTimeStamp	DATE	NOT NULL	Time stamp of the NE event.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Model type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
Direction	VARCHAR2(8)	DEFAULT NULL	TL1 direction.
Location	VARCHAR2(4)	DEFAULT NULL	TL1 location.

1.1.9 Alarm_Event_Type_ENUM_Table

The Alarm_Event_Type_ENUM_Table statically maps alarm and event type indexes to strings.

Primary key: ALARM_EVENT_TYPE_ENUM_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-9: Alarm_Event_Type_ENUM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
EnumIndex	NUMBER(5)	—	Index of alarm or event type.
EnumString	VARCHAR2(128)	—	Alarm or event type string for the EnumIndex.
TL1String	VARCHAR2(64)	—	Transaction language 1 (TL1) string for the alarm.
TMFString	VARCHAR2(64)	DEFAULT UNIDENTIFIED	TMF mapping string for the alarm.
EventType	NUMBER(1)	DEFAULT 1	Type of event. Values are: 1: NE_ALARM 2: NE_EVENT 3: EMS_EVENT 4: EMS_ALARM

1.1.10 APC_Domain_Table

The APC_Domain_Table contains information about the adjacent point code (APC) domains.

Primary key: APC_DOMAIN_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-10: APC_Domain_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NEDBAccessID in NE_INFO_TABLE.
SideOut	NUMBER(10)	DEFAULT 0	ID of the outgoing side.
ApcState	NUMBER(10)	DEFAULT 0	APC state.
ApcAdminState	NUMBER(10)	DEFAULT 0	APC admin state.
Sides	VARCHAR2(1000)	—	All sides of the domain. This is a string of separated sides. Each side is defined by NEDBAccessID, SideIn, SideOut. SideOut is not defined for the last side of a domain. NEDBAccessID and SideOut of the first side are not stored in this column; so this column may contain nothing for a domain which does not have any sides; for example, “2 2 3,4 3 2,3 4.”

1.1.11 APS_Group_Table

The APS_Group_Table contains information about the automatic protection switching (APS) groups configured in each device.

Primary key: APS_GROUP_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-11: APS_Group_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	Specific NE instance; same as NEDBAccessID in the NE_INFO_TABLE.
GroupName	VARCHAR2(64)	NOT NULL	Group name for APS.
ConfigMode	NUMBER(2)	DEFAULT 1	APS group architecture. Values are: 1: 1+1 2: 1:N 3: 1+1 compatible 4: 1+1 optimized
ConfigRevert	NUMBER(2)	DEFAULT 1	Revertive mode. Values are: 1: Nonrevertive 2: Revertive
ConfigDirection	NUMBER(2)	DEFAULT 1	Direction. Values are: 1: Unidirectional 2: Bidirectional
ConfigCreationTime	DATE	—	Time when this group was created.
ConfigSpan	NUMBER(2)	DEFAULT 1	Type of span. Values are: 1: Hop-by-hop 2: End-to-end
ProtectionType	NUMBER(2)	DEFAULT 1	Type of protection. Values are: 1: None 2: Splitter 3: Y-cable 4: Trunk
YCableConfig	NUMBER(2)	DEFAULT 1	Y-cable configuration if y-cable protection is used. Values are: 1: Not y-cable 2: Y-cable 3: Y-cable cross-connect common

Column Name	Oracle Data Type	Integrity Constraints	Description
SwitchedChannelStatus	NUMBER(2)	DEFAULT 0	Protection channel number.

1.1.12 Audit_Log_Mapping_Table

The Audit_Log_Mapping_Table contains information about audit log mapping.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-12: Audit_Log_Mapping_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserType	NUMBER(10)	—	Defines whether user type is Service or User.
ActualName	VARCHAR(256)	—	Name agreed upon by client and server.
CurrentName	VARCHAR(256)	—	Name currently used by the server.

1.1.13 Audit_Log_Table

The Audit_Log_Table is obsolete.

1.1.14 BLSR_Node

The BLSR_Node contains information about the BLSR node.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-13: BLSR_Node

Column Name	Oracle Data Type	Integrity Constraints	Description
BLSRID	NUMBER(8)	NOT NULL	BLSR ID.
RINGID	VARCHAR(8)	NOT NULL	Ring ID.
NODEID	NUMBER(8)	NOT NULL	Node ID.
RINGNODEID	NUMBER(8)	NOT NULL	Ring node ID.

1.1.15 BLSR_Span

The BLSR_Span contains information about the BLSR span.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-14: BLSR_Span

Column Name	Oracle Data Type	Integrity Constraints	Description
BLSRID	NUMBER(8)	NOT NULL	BLSR ID.
RINGID	VARCHAR(8)	NOT NULL	Ring ID.
LINKID	NUMBER(8)	NOT NULL	Link ID.
WORKINGSPAN	NUMBER(1)	NOT NULL	Working span.
WESTNODEID	NUMBER(10)	NOT NULL	Node ID of the BLSR (west line).
EASTNODEID	NUMBER(10)	NOT NULL	Node ID of the BLSR (east line).

1.1.16 BLSR_Table

The BLSR_Table contains information about BLSRs.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-15: BLSR_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
BLSRID	NUMBER(8)	NOT NULL	BLSR ID.
RINGID	VARCHAR(8)	NOT NULL	Ring ID.
RINGTYPE	NUMBER(2)	NOT NULL	Ring type.
LINERATE	NUMBER(2)	NOT NULL	Line rate.
STATUS	VARCHAR2(12)	NOT NULL	Ring status.
RINGREVERSION	NUMBER(8, 2)	NOT NULL	Ring reversion time.
SPANREVERSION	NUMBER(8, 2)	NOT NULL	Span reversion time.
NODES	VARCHAR2(512)	NOT NULL	Nodes associated with the ring.

1.1.17 Bridge_Group_Table

The Bridge_Group_Table contains information about the bridge group table.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-16: Bridge_Group_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoNodeID	NUMBER(10)	NOT NULL	Node ID of the circuit's source node.
TopoUniqueId	NUMBER(10)	NOT NULL	Unique number within the node.
Vlan_Number	NUMBER(10)	NOT NULL	Unique number to identify the VLAN.
NEID	NUMBER(10)	NOT NULL	NE ID.
Slot_Number	NUMBER(3)	NOT NULL	Slot number.
Bridge_Group_Number	NUMBER(3)	NOT NULL	Bridge group number.
BRIDGE_PROTOCOL	VARCHAR2(30)	—	—

1.1.18 Cerent_NE_Group_Table

The Cerent_NE_Group_Table indicates how the CTC-based NEs' discovered NEs are grouped based on the grouping rule.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-17: Cerent_NE_Group_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
GroupOption	NUMBER(4)	DEFAULT 0	Values are: 0: Default grouping 1: Group by discovered NE 2: Group by subnet

1.1.19 Circuit_BS_Tbl

The Circuit_BS_Tbl stores basic information about the circuit. .

Primary key: CIRCUIT_BS_TBL_PK

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-18: Circuit_BS_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktName	VARCHAR2(64)	—	Name of the circuit.
CktDescription	VARCHAR2(256)	—	Description of the circuit.
CustomerId	VARCHAR2(256)	—	Customer ID.
ServiceId	VARCHAR2(256)	—	Service ID.
CktType	NUMBER(2)	—	Type of circuit. Values are: 0: ETHERNET 1: STS_CIRCUIT 2: VT_CIRCUIT 3: VT_TUNNEL 4: VT_AGGREGATION 5: STS_VCAT_CIRCUIT 6: VT_VCAT_CIRCUIT 7: GIGABIT_ETHERNET 8: TENGIGABIT_ETHERNET 9: FIBERCHANNEL 10: FICON 11: ESCON 12: SDH_CIRCUIT 13: ISC_COMPATIBLE 14: ISC_PEER 15: SYSPLEX 16: DWDM_TRANSPARENT_ANY 17: VC_HO_PATH_CIRCUIT 18: VC_LO_PATH_CIRCUIT 19: VC_LO_PATH_TUNNEL 20: VC_LO_PATH_AGGREGATION 21: VC_HO_VCAT_CIRCUIT 22: VC_LO_VCAT_CIRCUIT 23: DWDM_WAVELENGTH 24: FAST_ETHERNET 25: FDDI 26: SYSPLEX_TIMER_ETR 27: SYSPLEX_TIMER_CLO 28:

Column Name	Oracle Data Type	Integrity Constraints	Description
			TENGIGABIT_ETHERNET_WAVELENGTH 29: POINT5GIGABIT_ETHERNET_WAVELENGTH 30: TENGIGABIT_ETHERNET_TRUNK 31: SONET_CIRCUIT 32: OCHNC_CIRCUIT 33: TWO_GIGABIT_ETHERNET 34: E1_CIRCUIT 35: T1 CIRCUIT 36: DVB_CIRCUIT 37: SDI_CIRCUIT 38: OCHCC_CIRCUIT 39: OCHTRAIL_TUNNEL_CIRCUIT
CktSize	NUMBER(2)	—	Size of circuit. Values are: 0: VT1_5 1: 1 = STS1 2: STS3C 3: STS6C 4: STS9C 5: STS12C 6: STS24C 7: STS48C 8: STS192C 9: TS768C 10: DWDM_2POINT5G 11: DWDM_10G 12: VC_11 13: VC_12 14: VC_3 15: VC_4 16: VC_4_2C 17: VC_4_3C 18: VC_4_4C 19: VC_4_8C 20: VC_4_16C 21: VC_4_64C 22: FIBERCHANNEL1 23: FIBERCHANNEL2 24: FICON 25: ESCON 26: SYSPLEX 27: FAST_ETHERNET 28: GIGABIT_ETHERNET 29: TENGIGABIT_ETHERNET

Column Name	Oracle Data Type	Integrity Constraints	Description
			30: VT_2 31: FDDI 32: OCHNC_NOTEQUIPPED 33: OCHNC_MULTIRATE 34: OCHNC_2_5G_FEC 35: OCHNC_2_5G_NOFEC 36: OCHNC_10G_FEC 37: OCHNC_10G_NOFEC 38: STS36C 39: VC_4_12C 40: STS18C 41: VC_4_6C 42: STM1C 43: STM4C 44: STM16C 45: STM64C 46: ISC_COMPATIBLE 47: ISC_PEER 48: SYSPLEX_TIMER_ETR 49: SYSPLEX_TIMER_CLO 50: TWO_GIGABIT_ETHERNET 51: E1 52: T1 53: DVB 54: SDI 55: OCHCC_OC_192 56: OCHCC_OC_48 57: OCHCC_OC_12 58: OCHCC_OC_3 59: OCHCC_10_GE 60: OCHCC_1_GE 61: OCHCC_10_GFC 62: OCHCC_4_GFC 63: OCHCC_2_GFC 64: OCHCC_1_GFC 65: OCHCC_4_GFICON 66: OCHCC_2_GFICON 67: OCHCC_1_GFICON 68: OCHCC_ESCON 69: OCHCC_ETR_CLO 70: OCHCC_ISC_PEER 71: OCHCC_ISC3_PEER_1G 72: OCHCC_ISC3_PEER2G 73: OCHCC_ISC_COMPAT 74: OCHCC_HVDT

Column Name	Oracle Data Type	Integrity Constraints	Description
			75: OCHCC_SD1_D1 76: OCHCC_DV_6000 77: OCHCC_PASS_THROUGH 78: OCHCC_OC_768 79: STS96C 80: VC_4_32C 81: OCHCC_STM_256 82: OCHCC_ISC1 83: OCHCC_DVBASI 84: OCHCC_IB 85: OCHCC_FE
CktDirection	NUMBER(1)	—	Type of traffic that the circuit carries. Values are: 0: Unidirectional 1: Bidirectional
CktIsMonitor	NUMBER(1)	—	Whether the circuit is a monitor circuit. Values are: 0: False 1: True
CktState	NUMBER(2)	—	State of the circuit. Values are: 0: CREATING 1: ACTIVE 2: DELETING 3: DROP_PENDING 4: INCOMPLETE 5: UPGRADABLE 6: INCOMPLETE_UPGRADABLE 7: ERRORED 8: NOT_UPGRADABLE 9: INCOMPLETE_NOT_UPGRADABLE 10: ROLL_PENDING 11: NOT_ACTIVATED 12: NOT_ACTIVE (A circuit assumes this temporary state during an NE service restart, or after a circuit removal on an entry that will be removed later. This state also applies to circuits that no longer exist.)
CktEnhancedState	NUMBER(2)	—	Administrative state of the circuit.
CktProtectionType	NUMBER(2)	—	Circuit protection type.
CktAdditionalInfo	VARCHAR2(64)	—	Additional information about the circuits.

Column Name	Oracle Data Type	Integrity Constraints	Description
cktComment	VARCHAR2(2014)	—	—
CktUseLap	NUMBER(1)	—	Whether the circuit uses LAP. Values are: 0: False 1: True
CktOchncChannel	NUMBER(3)	—	134: OCHCC_72_48 135: OCHCC_72_89 136: OCHCC_73_30 137: OCHCC_73_71 138: OCHCC_74_13 139: OCHCC_74_54 140: OCHCC_74_95 141: OCHCC_75_37 142: OCHCC_75_78 143: OCHCC_76_20 144: OCHCC_76_61 145: OCHCC_77_03 146: OCHCC_94_22 147: OCHCC_94_64 148: OCHCC_95_06 149: OCHCC_95_49 150: OCHCC_95_91 151: OCHCC_96_34 152: OCHCC_96_76 153: OCHCC_97_19 154: OCHCC_97_62 155: OCHCC_98_04 156: OCHCC_98_47 157: OCHCC_98_89 158: OCHCC_99_32 159: OCHCC_99_75 160: OCHCC_00_17 161: OCHCC_00_60 162: OCHCC_01_03 163: OCHCC_01_46 164: OCHCC_01_88 165: OCHCC_02_31 166: OCHCC_02_74 167: OCHCC_03_17 168: OCHCC_03_60 169: OCHCC_04_03 170: OCHCC_61_42 171: OCHCC_61_83 172: WV_850_00

Column Name	Oracle Data Type	Integrity Constraints	Description
CktOchncDir	NUMBER(1)	—	Direction of the OCHNC circuit. Values are: 0: East-to-west 1: West-to-east 2: Not applicable
isDuplicateName	NUMBER(1)	DEFAULT 1	Whether or not the circuit has a duplicate name. Values are: 0: Duplicate 1: Not duplicate
NPID	NUMBER(10)	—	Network partition ID to which the subnetwork belongs.
CktAliasName	VARCHAR2(80)	—	Circuit alias name.
isOpenVcat	NUMBER(1)	DEFAULT 0	Indicates whether or not the circuit is an open VCAT. Values are: 0: False 1: True
isOverlay	NUMBER(1)	DEFAULT 0	Indicates whether or not the circuit is an overlay. Values are: 0: False 1: True
isForPortGrouping	NUMBER(1)	—	Indicates the Port Grouping Only check box in the circuit wizard.
isGmpls	NUMBER(1)	—	Whether or not the service is GMPLS. Values are: 0: False 1: True

1.1.20 Circuit_Ctp_Tbl

The Circuit_Ctp_Tbl stores the circuit termination points.

Foreign key: FK_CIRCUIT_CTPNPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-19: Circuit_Ctp_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
CTPAccessID	NUMBER(10)	NOT NULL	Unique sequence number generated by the database. This is the link between the table and the Circuit_Span_BS_Tbl that contains CTPs. Values are: 1: Circuit termination point CTP 2: L2 VLAN
CTP_Type	NUMBER(5)	NOT NULL	CTP entity type. Values are: 0: SRC 1: SEC_SRC 2: DROP 3: SPAN_SRC 4: SPAN_DEST 5: SPAN_SRC_THRU_TUNNEL 6: SPAN_DEST_THRU_TUNNEL 7: SPAN_SRC_VTT_EXPANDED 8: SPAN_DEST_VTT_EXPANDED
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktCtpNodeId	NUMBER(10)	—	CTP circuit node ID.
CKTCTPNeDBAccessId	NUMBER(10)	NOT NULL	NE database ID.
CktCtpModuleType	NUMBER(8)	—	Module type of the CTP.
CktCtpIfIndex	NUMBER(20)	—	If_Static_Index at the destination.
CktCtpObjectType	NUMBER(10)	—	Object type at the destination.
CktCtpPhysicalLoc	NUMBER(20)	—	Physical location of the destination node.
CktCtpModelType	NUMBER(4)	—	Model type of the destination node.
CktCtpDropProt	NUMBER(1)	—	Whether the endpoint is protected. Values are: 0: False 1: True
CktCtpPathProt	NUMBER(1)	—	Path protection for this drop. Values are: 0: PATH FULLY PROTECTED 1: PATH PROTECTED 2: PATH UNPROTECTED 3: PATH FULLY UNPROTECTED

Column Name	Oracle Data Type	Integrity Constraints	Description
CktCtpDropProtType	NUMBER(1)	—	Protection type. Values are: 0: NO_PROTECTION 1: UNKNOWN_PROTECTION 2: SPLITTER_PROTECTION 3: Y_CABLE_OR_LINE_CARD_PROTECTION 4: TRUNK_CARD_PROTECTION
CktCtpPortName	VARCHAR2(64)	—	Destination port name of the circuit.
CktCtpAdditionalInfo	VARCHAR2(64)	DEFAULT NULL	Additional information about the circuit destination.
NPID	NUMBER(10)	DEFAULT 0	Network partition ID to which the subnetwork belongs.
EqptInfoObjIndex	NUMBER(6)	DEFAULT 0	ObjectIndex of reference in EqptInfoTable.
PayLoadType	NUMBER(6)	—	Indicates the payload type of the Circuit Termination Point (CTP).

1.1.21 Circuit_Dest_Tbl

The Circuit_Dest_Tbl shows the end and drop points for network circuits.

Primary key: None

Foreign key: FK_CIRCUIT_DEST_NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-20: Circuit_Dest_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktDestNodeId	NUMBER(10)	—	Destination node ID.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktDestModuleType	NUMBER(8)	—	Module type at destination.
CktDestIfIndex	NUMBER(20)	—	Static index of the interface at the destination.
CktDestObjectType	NUMBER(10)	—	Object type at the destination.
CktDestPhysicalLoc	NUMBER(20)	—	Physical location of the destination node.
CktDestModelType	NUMBER(4)	—	Model type of the destination node.
CktDestDropProt	NUMBER(1)	—	Whether the endpoint is protected. Values are: 0: False 1: True
CktDestPathProt	NUMBER(1)	—	Path protection for this drop. Values are: 0: Path fully protected 1: Path protected 2: Path unprotected 3: Path fully unprotected
CktDestDropProtType	NUMBER(1)	—	Protection type for ONS 15540: 0: No protection 1: Unknown protection 2: Splitter protection 3: Y-cable or line-card protection 4: Trunk card protection
CktDestPortName	VARCHAR2(64)	—	Destination port name of the circuit.
CktDestAdditionalInfo	VARCHAR2(64)	DEFAULT NULL	Additional information about the circuit destination.
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.

1.1.22 Circuit_Size_Table

The Circuit_Size_Table contains information about the circuit sizes.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-21: Circuit_Size_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserId	NUMBER(4)	—	User ID.
CircuitSizeType	NUMBER(2)	—	Circuit size.
CircuitSizeId	NUMBER(2)	—	Circuit size ID.
CircuitSizeName	VARCHAR2(32)	—	Name of the circuit size.

1.1.23 Circuit_Span_BS_Tbl

The Circuit_Span_BS_Tbl stores circuit span information. Each span consists of the span base information, span source circuit termination point, and span destination circuit termination point. The span base information is stored in the Circuit_Span_BS_Tbl. The span source circuit termination point is stored in the Circuit_CTP_Tbl and a reference to the CIRCUIT_CTP_TBL row (that is, database row ID) is stored in the Circuit_Span_BS_Tbl as CIRCUIT_SPAN_SRC_CTP_ACCESSID. Similarly, the span destination circuit termination point is stored in the Circuit_CTP_Tbl and a reference to the CIRCUIT_CTP_TBL row (database row ID) is stored in the Circuit_Span_BS_Tbl as CIRCUIT_SPAN_DEST_CTP_ACCESSID.

Foreign key: FK_CIRCUIT_SPAN_BS_NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-22: Circuit_Span_BS_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSpanBSId	NUMBER(10)	NOT NULL	Unique sequence number generated by the database.
CktNodeId	NUMBER(10)	—	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	—	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktLinkId	NUMBER(10)	—	ID of the link to which this span belongs.
Span_Type	NUMBER(5)	NOT NULL	Type of span.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSpanSrcCtpAccessId	NUMBER(10)	—	CTP ID of the span source.
CktSpanDstCtpAccessId	NUMBER(10)	—	CTP ID of the span destination.
CktSpanSrcState	NUMBER(1)	—	State of the circuit span from the source circuit node's point of view. Values are: 0: INACTIVE 1: STRANDED 2: ROUTED 3: TENTATIVE 4: ACTIVE
CktSpanDestState	NUMBER(1)	—	State of the circuit span from the destination circuit node's point of view. Values are: 0: INACTIVE 1: STRANDED 2: ROUTED 3: TENTATIVE 4: ACTIVE
IsCktForwarding	NUMBER(1)	—	Whether the circuit span is forwarding; valid for Ethernet circuit spans only. Values are: 0: False 1: True
IsCktSpanInUPSR	NUMBER(1)	—	Whether the circuit span is part of a UPSR ring. Values are: 0: False 1: True
IsCktSpanWorking	NUMBER(1)	—	Whether the circuit span is a working span. Values are: 0: False 1: True
IsCktSpanActive	NUMBER(1)	—	Whether the circuit span carries active traffic. Values are: 0: False 1: True
CktSpanProtOp	NUMBER(1)	—	Last successful span protection operation performed on this span. Values are: 1: Manual 2: Force 3: Lockout

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSpanCDLFlowID	NUMBER(3)	DEFAULT –1	(Applicable only to ONS 155xx circuits) Source CDL flow identifier. The valid range is from 0 to 254.
CktSpanSrcPortName	VARCHAR2(64)	—	Span source port name of the circuit.
CktSpanDestPortName	VARCHAR2(64)	—	Span destination name of the circuit.
CktUpsrSpanSrcState	NUMBER(2)	—	UPSR (SNCP) status selector at the edge of the span at the source side.
CktUpsrSpanDstState	NUMBER(2)	—	UPSR (SNCP) status selector at the edge of the span at the destination side.
NPID	NUMBER(10)	DEFAULT 0	Network partition ID to which the subnetwork belongs.

1.1.24 Circuit_Span_Tbl

The Circut_Span_Tbl contains information about spans associated with circuits.

Primary key: None

Foreign key: FK_CIRCUIT_SPAN_NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-23: Circuit_Span_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSpanId	NUMBER(10)	NOT NULL	Unique sequence number generated by the database.
CktNodeId	NUMBER(10)	—	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	—	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktLinkId	NUMBER(10)	—	ID of the link to which this span belongs.
CktSpanSrcNodeId	NUMBER(10)	—	Node ID of the span source.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSpanSrcModuleType	NUMBER(8)	—	Module type at the span source.
CktSpanSrcIfIndex	NUMBER(20)	—	Static index of the interface at the span source.
CktSpanSrcObjectType	NUMBER(10)	—	Object type at span source.
CktSpanSrcPhysicalLoc	NUMBER(20)	—	Physical location of the source node of the span.
CktSpanSrcModelType	NUMBER(4)	—	Model type for source node of span.
CktSpanDestNodeId	NUMBER(10)	—	Node ID of the span destination.
CktSpanDestModuleType	NUMBER(8)	—	Module type at the span destination.
CktSpanDestIfIndex	NUMBER(20)	—	Static index of the interface at the span destination.
CktSpanDestObjectType	NUMBER(10)	—	Object type at span destination.
CktSpanDestPhysicalLoc	NUMBER(20)	—	Physical location of the destination node of the span.
CktSpanDestModelType	NUMBER(4)	—	Model type for destination node of span.
CktSpanSrcState	NUMBER(1)	—	State of the circuit span from the source circuit node. Values are: 0: Inactive 1: Stranded 2: Routed 3: Tentative 4: Active
CktSpanDestState	NUMBER(1)	—	State of the circuit span from the destination circuit. Values are: 0: Inactive 1: Stranded 2: Routed 3: Tentative 4: Active
IsCktForwarding	NUMBER(1)	—	Whether the circuit span is forwarding (for Ethernet circuit spans only): 0: False 1: True

Column Name	Oracle Data Type	Integrity Constraints	Description
IsCktSpanInUPSR	NUMBER(1)	—	Whether the circuit span is part of a unidirectional path switched ring (UPSR) ring. Values are: 0: False 1: True
IsCktSpanWorking	NUMBER(1)	—	Whether the circuit span is a working span. Values are: 0: False 1: True
IsCktSpanActive	NUMBER(1)	—	Whether the circuit span carries active traffic. Values are: 0: False 1: True
CktSpanProtOp	NUMBER(1)	—	Last successful span protection operation performed on this span. Values are: 0: Clear 1: Manual 2: Force 3: Lockout
CktSpanAdditionalInfo	VARCHAR2(64)	DEFAULT NULL	Additional information about the circuit span.
CktSpanCDLFlowID	NUMBER(3)	DEFAULT –1	Source converged data link (CDL) flow identifier; applies only to ONS 15530 and ONS 15540 circuits. Valid range is 0 to 254.
CktSpanSrcPortName	VARCHAR2(64)	—	Span source port name of the circuit.
CktSpanDestPortName	VARCHAR2(64)	—	Span destination port name of the circuit.
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.

1.1.25 Circuit_Tbl

The Circuit_Tbl shows circuit information about an entire network.

Primary key: CIRCUIT_TBL_PK

Foreign key: FK_CIRCUIT_NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-24: Circuit_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktName	VARCHAR2(80)	—	Name of the circuit.
CktDescription	VARCHAR2(256)	—	Description of the circuit.
CustomerID	VARCHAR2(256)	—	Customer ID.
ServiceID	VARCHAR2(256)	—	Service ID.
CktType	NUMBER(2)	—	Circuit type. Values are: 0: Ethernet 1: Synchronous transport signal (STS) circuit 2: Virtual Tributary (VT) circuit 3: VT tunnel 4: VT aggregation 5: STS VCAT circuit 6: VT VCAT circuit 7: Gigabit Ethernet 8: 10-Gigabit Ethernet 9: Fiber channel 10: Fiber connectivity (FICON) 11: Enterprise System Connection (ESCON) 12: SDH circuit 13: Intersystem channel (ISC)-compatible 14: ISC peer 15: SYSPLEX 16: Dense wavelength division multiplexing (DWDM) transparent any 17: Virtual channel (VC) high order path circuit 18: VC low order path circuit 19: VC low order path tunnel 20: VC low order path aggregation 21: VC high order VCAT circuit 22: VC low order VCAT circuit 23: DWDM wavelength 24: Fast Ethernet

Column Name	Oracle Data Type	Integrity Constraints	Description
			25: FDDI 26: SYSPLEX_TIMER_ETR 27: SYSPLEX_TIMER_CLO 28: 10-Gigabit Ethernet wavelength 29: 2.5-Gigabit Ethernet wavelength 30: 10-Gigabit Ethernet trunk 31: SONET circuit 32: OCHNC circuit 33: Two_Gigabit_Ethernet 34: E1 circuit 35: T1 circuit 36: DVB_CIRCUIT 37: SDI_CIRCUIT 38: OCHCC_CIRCUIT 39: OCHTRAIL_TUNNEL_CIRCUIT
CktSize	NUMBER(2)	—	Size of the circuit. Values are: 0: VT 1.5 1: STS1 2: STS3C 3: STS6C 4: STS9C 5: STS12C 6: STS24C 7: STS48C 8: STS192C 9: STS768C 10: DWDM 2.5 G 11: DWDM 10 G 12: VC_11 13: VC_12 14: VC_3 15: VC_4 16: VC_4_2C 17: VC_4_3C 18: VC_4_4C 19: VC_4_8C 20: VC_4_16C 21: VC_4_64C 22: Fiber channel 1 23: Fiber channel 2 24: FICON 25: ESCON 26: SYSPLEX

Column Name	Oracle Data Type	Integrity Constraints	Description
			27: Fast Ethernet 28: Gigabit Ethernet 29: 10-Gigabit Ethernet 30: VT_2 31: FDDI 32: Optical Channel Network Connection (OCHNC) not equipped 33: OCHNC multirate 34: OCHNC 2.5G FEC 35: OCHNC 2.5G NOFEC 36: OCHNC 10G FEC 37: OCHNC 10G NOFEC 38: STM 1C 39: STM 4C 40: STM 16C 41: STM 64C 42: ISC compatible 43: ISC peer 44: SYSPLEX_TIMER_ETR 45: SYSPLEX_TIMER_CLO 46: TWO_GIGABIT_ETHERNET 47: E1 48: T1
CktDirection	NUMBER(1)	—	Type of traffic that the circuit carries. Values are: 0: Unidirectional 1: Bidirectional
CktIsMonitor	NUMBER(1)	—	Whether the circuit is a monitor circuit. Values are: 0: False 1: True
CktState	NUMBER(2)	—	State of the circuit. Values are: 0: Creating 1: Active 2: Deleting 3: Drop-pending 4: Incomplete 5: TL1 active 6: TL1 incomplete 7: Errored 8: Roll-pending 9. Sig-pending 10: Sig-failed 11: Conversion-pending 12: Not activated

Column Name	Oracle Data Type	Integrity Constraints	Description
CktEnhancedState	NUMBER(2)	—	Administrative state of the circuit. Values are: 0: IS 1: OOS 2: OOS_MT 3: OOS_AINS 4: CES_NOT_DEFINED 5: OOS_PARTIAL 6: OOS_MT_PARTIAL 7: OOS_AINS_PARTIAL 8: IS_PARTIAL 9: UNKNOWN_ADMIN_STATE
CktProtectiontype	NUMBER(2)	—	Circuit protection type. Values are: 0: BLSR2F 1: OnePlusOne 2: NotDefined 3: UnProt 4: BLSR4F 5: PCA2F 6: PCA4F 7: NotDefined1 8: Protected 9: BLSR 10: UPSR 11: PCA 12: Lost 13: Unknown 14: None 15: UPSRDRI
CktSrcNodeId	NUMBER(10)	—	Source node ID.
CktSrcModuleType	NUMBER(8)	—	Module type at source.
CktSrcIfIndex	NUMBER(20)	—	Static index of the interface at the source.
CktObjectType	NUMBER(10)	—	Source object type.
CktSrcPhysicalLoc	NUMBER(20)	—	Source node physical location.
CktModelType	NUMBER(4)	—	Source node model type.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktSrcDropProt	NUMBER(1)	—	Whether the endpoint is protected. Values are: 0: False 1: True
CktSrcDropProtType	NUMBER(1)	—	Protection type for ONS 15540: 0: No protection 1: Unknown protection 2: Splitter protection 3: Y-cable or line-card protection 4: Trunk card protection
CktSecSrcNodeId	NUMBER(10)	—	Secondary source node ID.
CktSecSrcModuleType	NUMBER(8)	—	Module type at secondary source node.
CktSecSrcIfIndex	NUMBER(20)	—	Static index of the interface at the secondary source node.
CktSecSrcObjectType	NUMBER(10)	—	Secondary source object type.
CktSecSrcPhysicalLoc	NUMBER(20)	—	Secondary source node physical location.
CktSecSrcModelType	NUMBER(4)	—	Model type for the secondary source node.
CktSecSrcDropProt	NUMBER(1)	—	Whether the secondary endpoint is protected. Values are: 0: False 1: True
CktSecSrcDropProtType	NUMBER(1)	—	Protection type for ONS 15540: 0: No protection 1: Unknown protection 2: Splitter protection 3: Y-cable or line-card protection 4: Trunk card protection
CktAdditionalInfo	VARCHAR2(64)	—	Additional information about the circuits.
CktDesNodeName	VARCHAR2(4000)	—	Node name of the circuit destination.
CktDestModuleName	VARCHAR2(4000)	—	Module name of the circuit destination.
CktDestPhysicalLoc	VARCHAR2(4000)	—	Physical location of the circuit destination.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktDestModelType	VARCHAR2(4000)	—	Model type of the circuit destination.
CktDestIfIndex	VARCHAR2(4000)	—	Circuit destination node ifindex.
CktDestObjectType	VARCHAR2(4000)	—	Object type.
CktComment	VARCHAR2(2014)	—	Circuit comment.
CktSrcPortName	VARCHAR2(64)	—	Port name of the circuit source.
CktSecSrcPortName	VARCHAR2(64)	—	Port name of the circuit secondary circuit source.
CktDestPortName	VARCHAR2(4000)	—	Port name of the circuit destination.
CktUseLap	NUMBER(1)	—	Indicates if the circuit uses a lap. Values are: 0: False 1: True

Column Name	Oracle Data Type	Integrity Constraints	Description
CktOchncChannel	NUMBER(2)	—	Circuit OCHNC channel number. Values are: 0: Not applicable 1: OCHNC_30_33 2: OCHNC_31_12 3: OCHNC_31_90 4: OCHNC_32_68 5: OCHNC_34_25 6: OCHNC_35_04 7: OCHNC_35_82 8: OCHNC_36_61 9: OCHNC_38_19 10: OCHNC_38_98 11: OCHNC_39_77 12: OCHNC_40_56 13: OCHNC_42_14 14: OCHNC_42_94 15: OCHNC_43_73 16: OCHNC_44_53 17: OCHNC_46_12 18: OCHNC_46_92 19: OCHNC_47_72 20: OCHNC_48_51 21: OCHNC_50_12 22: OCHNC_50_92 23: OCHNC_51_72 24: OCHNC_52_52 25: OCHNC_54_13 26: OCHNC_54_94 27: OCHNC_55_75 28: OCHNC_56_55 29: OCHNC_58_17 30: OCHNC_58_98 31: OCHNC_59_79 32: OCHNC_60_61
CktOchncDir	NUMBER(1)	—	Direction of the OCHNC circuit. Values are: 0: East to west 1: West to east 2: Not applicable
isDuplicateName	NUMBER(1)	DEFAULT 1	Values are: 0: DUPLICATE 1: NOT_DUPLICATE
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.
CktAliasName	VARCHAR2(64)	—	Circuit alias name.

1.1.26 Circuit_UPSR_BS_Tbl

The Circuit_UPSR_BS_Tbl shows information about the UPSR and SNCP selectors in the Prime Optical database.

Primary key: None

Foreign key: FK_CIRCUIT_UPSR_BS_NPID

Tablespace: BASE_DATA_TABLESPACE

> In CTM R9.2, the Circuit_UPSR_BS_Tbl is present in the database, but it is blank and contains no data. When a future Prime Optical patch is applied, the Circuit_UPSR_BS_Tbl will populate during the discovery of NEs and circuits.

Table 1-25: Circuit_UPSR_BS_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
NodeId	NUMBER(10)	NOT NULL	The status of the device.
WorkingCtpIfindex	NUMBER(20)	NOT NULL	—
WorkingCtpPhysicalLoc	NUMBER(20)	NOT NULL	—
ProtectCtpIfindex	NUMBER(20)	NOT NULL	—
ProtectCtpPhysicalLoc	NUMBER(20)	NOT NULL	—
CurrentIsWorking	NUMBER(2)	NOT NULL	—
SwitchState	NUMBER(2)	NOT NULL	—
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.

1.1.27 Circuit_VCG_Tbl

The Circuit_VCG_Tbl contains the inventory of virtual concatenation group circuits.

Primary key: none

Foreign key: FK_CIRCUIT_VCG_ NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-26: Circuit_VCG_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
ParentCktNodeId	NUMBER(10)	NOT NULL	Node ID of the source node of the parent VCAT circuit. This is an internal field.
ParentCktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The combination of ParentCktNodeId and ParentCktUniqueId results in a unique circuit ID in the whole network.
VCGDirection	NUMBER(1)	—	VCG direction. Values are: 0: None 1: Forward 2: Backward 3: Bidirectional
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.
VCATSTATE	NUMBER(1)	DEFAULT 0	VCAT member state.

1.1.28 Circuit_VLAN_Tbl

The Circuit_VLAN_Tbl contains information about VLANs associated with circuits.

Primary key: CIRCUIT_VLAN_PK

Foreign key: FK_CIRCUIT_VLAN_NPID

Tablespace: BASE_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-27 Circuit_VLAN_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktVlanId	NUMBER(10)	—	VLAN ID.
CktVlanName	VARCHAR2(64)	—	VLAN name.
NPID	NUMBER(10)	DEFAULT 0	Network partition ID.

1.1.29 CLISession_Command_History_Tbl

The CLISession_Command_History_Tbl provides a list of commands used by a specific user in a CLI session.

Primary key: CLISESSION_COMMAND_HISTORY_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-28 CLISession_Command_History_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
SessionTag	VARCHAR2(32)	NOT NULL	Session tag.
UserName	VARCHAR2(64)	NOT NULL	Name of the user.
Command_History	VARCHAR2(256)	NOT NULL	Command history.

1.1.30 Connless_TP_Table

The Connless_TP_Table contains the connectionless termination points (TPs) along with their associated information.

Primary key: CONNLESS_TP_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-29 Connless_TP_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
CLTP_NE_DB_ID	NUMBER(10)	NOT NULL	NEDB ID of the connectionless TP.
CLTP_PHYS_LOC	NUMBER(20)	NOT NULL	Physical location of the connectionless TP.
CLTP_IF_INDEX	NUMBER(20)	NOT NULL	IF index of the connectionless TP for a Circuit Termination Point (CTP).
CLTP_TRANSM_PARAMS	VARCHAR2(512)	—	Not used (reserved for future use).
MFD_NATIVE_NAME	VARCHAR2(64)	—	Transmission parameters associated to the connectionless TP.
CLTP_PORT_TP_ROLE_STATE	NUMBER(1)	—	Matrix flow domain name associated to the connectionless TP, if any.
CLTP_TYPE	NUMBER(1)	—	Connectionless TP type. Options are: Physical Termination Point (PTP) Floating Termination Point (FTP) Circuit Termination Point (CTP)

1.1.31 CORBA_Functions_Table

The CORBA_Functions_Table contains information about the different GateWay/CORBA function name and types.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-30 CORBA_Functions_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
InterfaceIndex	NUMBER(10)	NOT NULL	Interface to which the function belongs.
FunctionName	VARCHAR(32)	NOT NULL	Name of function.
FunctionType	NUMBER(10)	NOT NULL	Type of the function. Values are:

1.1.32 CORBA_GW_Active_User_Table

The CORBA_GW_Active_User_Table contains information about the active Prime Optical GateWay/CORBA users.

Primary key: CORBA_GW_ACTIVE_USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-31 CORBA_GW_Active_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
active_session_id	VARCHAR2(128)	NOT NULL	Session ID.
user_name	VARCHAR2(128)	NOT NULL	Username of the OSS user.
client_end_ip	NUMBER(10)	NOT NULL	IP address of the OSS user.
LastLoginTime	DATE	—	Most recent user login date and time.
client_end_ipv6	RAW(16)	—	IPv6 address.
is_ipv6	NUMBER(1)	—	Flag indicating the IP version. Values are: 0: IPv4 1: IPv6

1.1.33 CORBA_Interface_Table

The CORBA_Interface_Table contains information about the GateWay/CORBA interface.

Primary key: CORBA_INTERFACE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-32: CORBA_Interface_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
InterfaceIndex	NUMBER(10)	NOT NULL	Index used for mapping of operations.
InterfaceName	VARCHAR(32)	NOT NULL	Name of the interface.
AdditionalPermissions	NUMBER(1)	NOT NULL	Type of interface. Values are: 0: No permissions 1: All read 2: All write
HelperClass	VARCHAR(128)	NOT NULL	Helper class name used to get the

Column Name	Oracle Data Type	Integrity Constraints	Description
			repository ID.
BaseInterface	NUMBER(10	NOT NULL	Base interface.

1.1.34 CORBA_Permission_Table

The CORBA_Permission_Table contains information about the GateWay/CORBA interface operation permissions.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-33: CORBA_Permission_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
InterfaceIndex	NUMBER(10)	NOT NULL	Index of the CORBA interface.
OperationID	NUMBER(10)	NOT NULL	Operation ID from the Operations_Table.

1.1.35 CTC_Model_Const_Table

The CTC_Model_Const_Table contains information about Cisco Transport Controller (CTC) models.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-34: CTC_Model_Const_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
CTCPackageName	VARCHAR2(40)	—	CTC package name.
CTCModel	VARCHAR2(40)	—	CTC model name.
CTMModel	VARCHAR2(128)	—	Corresponding Prime Optical model name.
CTMObjectIndex	NUMBER(10)	—	Metadata object Index as defined in ObjectAttributeTable.
ModelType	NUMBER(10)	—	Model type as defined in Model_Type_Table.

1.1.36 CTC_UserProfile_Table

The CTC_UserProfile_Table contains information about CTC user profiles.

Primary key: CTC_USR_PROF_TABLE_PK

Tablespace: None

Table 1-35: CTC_UserProfile_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserID	VARCHAR2(64)	—	User ID of the CTC user type.
UserPassword	VARCHAR2(64)	NOT NULL	Password of the CTC user type.
PrivilegeID	NUMBER(10)	NOT NULL	Privilege ID of the CTC user type.
UserDesc	VARCHAR2(256)	NOT NULL	Description of the CTC user type.

1.1.37 CTM_Config_Table

The CTM_Config_Table stores Prime Optical configuration information. For detailed definitions of the parameters in this table, see Config_Table Parameters.

Primary key: ctm_config_table_pk

Tablespace: BASE_DATA_TABLESPACE

Table 1-36: CTM_Config_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SectionName	VARCHAR2(40)	NOT NULL	Name of the section.
PropertyName	VARCHAR2(40)	NOT NULL	Name of the parameter.
ActiveValue	VARCHAR2(128)	—	Active value of the parameter.
IsPermanent	NUMBER(1)	NOT NULL	Whether or not the value can be changed. Values are: 0: False 1: True
PermanentValue	VARCHAR2(128)	—	Permanent value of the parameter.

1.1.38 CTM_Unique_ObjectType_Table

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-37: CTM_Unique_ObjectType_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ObjectType	NUMBER(10)	NOT NULL UNIQUE	Object type.
ObjectName	VARCHAR2(256)	NOT NULL	Object name of the object type.

1.1.39 CTM_Unknown_User_Table

The CTM_Unknown_User_Table contains information about login attempts by unknown Prime Optical users.

Primary key: CTM_Unknown_USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-38: CTM_Unknown_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IPAddress	NUMBER(10)	NOT NULL	IP address of the unknown user.
FailedAttempts	NUMBER(2)	DEFAULT 0	Number of login attempts from the unknown user associated with a specific IP address.
LastLoginFailTime	DATE	—	Most recent time when the unknown user attempted to log into the Prime Optical client.
LockedState	NUMBER(1)	DEFAULT 0	Status of the unknown user's account (locked or unlocked). Values are: 0: Not locked 1: Locked

1.1.40 Domain_Table

The Domain_Table contains information about the hierarchy of group and NE nodes in the domain.

Primary key: DOMAIN_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-39: Domain_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
TreeNodeID	NUMBER(10)	NOT NULL	Unique sequence number generated by the database. Each node in the Domain Explorer tree is assigned a unique number that is used as an index to access group information in other tables.
ParentType	NUMBER(2)	NOT NULL	Indicates that the parent is a group.
ParentId	NUMBER(10)	NOT NULL	ID of the group.
ChildType	NUMBER(2)	NOT NULL	Type of tree node. Values are: 1: Root node 2: Group node 3: NE node
ChildId	NUMBER(10)	NOT NULL	ID of the group or NE.

1.1.41 Ems_Alarm_Table

The Ems_Alarm_Table shows information about EMS alarms.

Primary key: EMS_ALARM_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-40: Ems_Alarm_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
Name	VARCHAR2(128)	—	Name of the alarm displayed to client.
alarmIndex	NUMBER(5)	—	EMS index of alarm or group.
isGroup	NUMBER(1)	—	Whether or not this alarm belongs to a group. Values are: 0: FALSE 1: TRUE
Severity	NUMBER(1)	DEFAULT 4	Alarm severity. Values are: –1: N/A 4: Warning 5: Minor 6: Major 7: Critical
isEnabled	NUMBER(1)	DEFAULT 1	Whether or not the alarm is enabled. Values are: 0: False 1: True

1.1.42 Enum_Table

The Enum_Table is used by link management to display link information in the link table.

Primary key: ENUM_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-41: Enum_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
EnumBase	NUMBER(4)	—	Value of attribute base.
EnumInt	NUMBER(3)	—	Possible values of the attribute base.
UIValue	VARCHAR2(128)	—	String representation of attribute base.

1.1.43 EQPT_Info_Table

The EQPT_Info_Table stores equipment provisioning information.

Primary key: EQPT_INFO_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-42: EQPT_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	NOT NULL	Database access ID; same as NEDBAccessID in the NE_INFO_TABLE.
PHYSICALLOC	NUMBER(20)	NOT NULL	Physical location.
IFINDEX	NUMBER(20)	NOT NULL	Interface index.
MODULETYPE	NUMBER(8)	—	Module type.
OBJECTINDEX	NUMBER(10)	NOT NULL	Metadata object index.
MODELTYPE	NUMBER(4)	NOT NULL	Model type.
COL1	VARCHAR2(256)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL2	VARCHAR2(256)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL3	VARCHAR2(256)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL4	VARCHAR2(256)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL5	VARCHAR2(256)	—	Columns 1 to 250 store attribute values as defined by the metadata.

Column Name	Oracle Data Type	Integrity Constraints	Description
...			
COL248	VARCHAR2(4000)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL249	VARCHAR2(4000)	—	Columns 1 to 250 store attribute values as defined by the metadata.
COL250	VARCHAR2(4000)	—	Columns 1 to 250 store attribute values as defined by the metadata.

1.1.44 Eqpt_Relationship_Table

The Eqpt_Relationship_Table contains information about the equipment relationship table.

Primary key: EQPT_RELATIONSHIP_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-43: Eqpt_Relationship_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	NOT NULL	NE database access ID.
SELFPHYSICALLOC	NUMBER(20)	NOT NULL	Physical location (equipment 1).
SELFIFINDEX	NUMBER(20)	NOT NULL	If index (equipment 1).
SELFMODULETYPE	NUMBER(10)	NOT NULL	Module type (equipment 1).
SELFMODELTYPE	NUMBER(10)	NOT NULL	Model type.
SELFOBJECTINDEX	NUMBER(10)	NOT NULL	Object index.
SELFSTROBJINST	VARCHAR2(4000)	NOT NULL	—
RELATEDPHYSICALLOC	NUMBER(20)	NOT NULL	Physical location (equipment 2).
RELATEDIFINDEX	NUMBER(20)	NOT NULL	If index (equipment 2).
RELATEDMODULETYPE	NUMBER(10)	NOT NULL	Module type (equipment 2).
RELATEDMODELTYPE	NUMBER(10)	NOT NULL	Model type.

Column Name	Oracle Data Type	Integrity Constraints	Description
RELATEDOBJECTINDEX	NUMBER(10)	NOT NULL	Object index.
RELATEDSTROBJINST	VARCHAR2(4000)	NOT NULL	—

1.1.45 Error_Log_Table

The Error_Log_Table contains information about all errors reported by Prime Optical.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-44: Error_Log_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TimeStamp	DATE	—	Time stamp of log message.
Module	NUMBER(2)	NOT NULL	One of the predefined modules. Values are: 1: ServiceWay 2: TL1 3: ThruWay 4: Transport Manager
Severity	NUMBER(2)	NOT NULL	Severity of error message. Values are: 1: Critical 2: Major 3: Minor 4: Informational
SubModule	NUMBER(2)	—	Submodules for modules.
FileName	VARCHAR2(256)	—	Name of the file from which the message was printed.
LineNumber	NUMBER(4)	—	Line number from which the message was printed.
LogMessage	VARCHAR2(1024)	—	Text of the error message.

1.1.46 EVC_EP

The EVC_EP contains information related to the endpoints of the Ethernet Virtual Circuit (EVC) services.

Primary key: EVC_EP_PK

Table 1-45: EVC_EP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the NE in the NE_INFO_TABLE. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	CTC index related to the packet transport switch.
SERVICEID	NUMBER(10)	NOT NULL	Identifier of the service.
CTCINDEX	NUMBER(10)	NOT NULL	CTC identifier of the physical object involved in the service.
PHYLOC	NUMBER(20)	NOT NULL	Prime Optical identifier of the physical object involved in the service.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical identifier used by the splice algorithm. This is an internal field.
SERVICENAME	VARCHAR2(256)	NOT NULL	Service name.
OPERSTATE	NUMBER(1)	NOT NULL	Operational status of the service endpoint.
ADMINSTATE	NUMBER(1)	NOT NULL	Administrative status of the service endpoint.
LINKID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the link in the LINK_TABLE. This is an internal field.
SERVICETYPE	NUMBER(10)	NOT NULL	Type of service.
DESCRIPTION	VARCHAR2(256)	—	Description of the service.
SLOTMODULETYPE	NUMBER(10)	NOT NULL	Slot module type. This is an internal field.
PORTMODULETYPE	NUMBER(10)	NOT NULL	Port module type. This is an internal field.

1.1.47 EVC_Node

The EVC_Node contains information about the NE for each EVC.

Primary key: EVC_NODE_PK

Table 1-46: EVC_Node

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the NE in the NE_INFO_TABLE. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	CTC index related to the packet transport switch.
SERVICEID	NUMBER(10)	NOT NULL	Identifier of the service.

Column Name	Oracle Data Type	Integrity Constraints	Description
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical identifier used by the splice algorithm. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
OPERSTATE	NUMBER(1)	NOT NULL	Operational status of the service endpoint.
ADMINSTATE	NUMBER(1)	NOT NULL	Administrative status of the service endpoint.
SERVICETYPE	NUMBER(10)	NOT NULL	Type of the service.
DESCRIPTION	VARCHAR2(256)	—	Description of the service.

1.1.48 EVC_SRV

The EVC_SRV stores the end-to-end information of a specific EVC service.

Primary key: EVC_SRV_PK

Table 1-47: EVC_SRV

Column Name	Oracle Data Type	Integrity Constraints	Description
SERVICEID	NUMBER(10)	NOT NULL	Identifier of the service.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical identifier used by the splice algorithm. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
DESCRIPTION	VARCHAR2(256)	—	Description of the service.
DISCOVERYSTATE	NUMBER(1)	NOT NULL	Discovery status of the service. Values are: 0: Partial 1: Completed
OPERSTATE	NUMBER(1)	NOT NULL	Operational status of the service.
EPCOUNT	NUMBER(4)	NOT NULL	Number of endpoints involved in the service.
SERVICETYPE	NUMBER(10)	NOT NULL	Type of service.
RXBW	NUMBER(20)	NOT NULL	Not used (reserved for future use).
TXBW	NUMBER(20)	NOT NULL	Overall bandwidth value used by the service.

1.1.49 Filter_Trial

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-48: Filter_Trial

Column Name	Oracle Data Type	Integrity Constraints	Description
TABLE_NAME	VARCHAR2(128)	NOT NULL	Table name.
Filter	BLOB	—	—

1.1.50 Flow_Domain_Table

The Flow_Domain_Table contains flow domains (FDs) and their associated information.

Primary key: FD_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-49: Flow_Domain_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
FD_USER_LABEL	VARCHAR2(64)	—	Description of the flow domain.
FD_NATIVE_NAME	VARCHAR2(64)	NOT NULL	Name of the flow domain.
FD_OWNER	VARCHAR2(64)	—	Owner of the flow domain.
FD_TRANSM_PARAMS	VARCHAR2(512)	—	Transmission parameters of the flow domain.
FD_NET_ACCESS_DOMAIN	VARCHAR2(64)	—	Network access of the flow domain.
FD_CONN_STATE	NUMBER(1)	—	Connection state of the flow domain.
FD_TYPE	NUMBER(1)	—	Type of flow domain.
FD_ADDITIONAL_INFO	VARCHAR2(512)	—	Additional information about the flow domain.

1.1.51 Forwarding_Table

The Forwarding_Table is obsolete.

1.1.52 GMPLS_VALUE_TABLE

The GMPLS_VALUE_TABLE contains information about the threshold and optical values.

Primary key: CKTNODEID_TO_CKTUNIQUEID_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-50: GMPLS_VALUE_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
CKTNODEID	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CKTUNIQUEID	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
ACTP_THRESHOLD	NUMBER(2)	NOT NULL	Accepted threshold value. Values are: 0: Green 1: Yellow 2: Orange 3: Red 4: Not applicable
OPT_VALUE	NUMBER(2)	NOT NULL	Calculated optical value. Values are: 0: Green 1: Yellow 2: Orange 3: Red 4: Not applicable
RESTORATIONSTATE_VALUE	NUMBER (2)	NOT NULL	Indicates the restoration status value related to a specific circuit. Values are: 0: None 1: Revertible 2: Restoring 3: Restored 4: Failed
RESTORATION_VALUE	NUMBER (2)	NOT NULL	Indicates whether the restoration facility is enabled. Values are: 0: False 1: True
REVERTIVE_VALUE	NUMBER (2)	NOT NULL	Indicates the type of revertive mode. Values are: Values are: 0: None 1: Manual 2: Automatic

Column Name	Oracle Data Type	Integrity Constraints	Description
SOAKTIME_VALUE	NUMBER (10)	NOT NULL	Indicates the soak time related to the revertive option.
IGNOREPATHALARM_VALUE	NUMBER (1)	NOT NULL	Indicates whether the circuit has been provisioned using the ignore path alarm option. Values are: 0: False 1: True

1.1.53 GNE_Table

The GNE_Table contains information about the various GNEs in the network.

Primary key: GNE_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-51: GNE_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
GNEID	NUMBER(8)	NOT NULL	GNE identifier. Unique sequence number generated by the database. This is the link between this table and GNEID in the NE_INFO_TABLE.
GNEType	NUMBER(4)	DEFAULT 1	Values are: 1: Unknown 2: Self 3: ISR_as_GNE 4: Foreign_GNE_TL1 5: Foreign_GNE_CLNP 6: Foreign_GNE_IP
RINGID	NUMBER(8)	—	Identifies the ring to which the NE is connected. NEs with similar RING_IDs are in the same ring.
GNESysId	VARCHAR2(128)	NOT NULL	Name of the GNE.
NextHopIPAddr	NUMBER(10)	NOT NULL	IP address of the next hop used to reach the GNE.
GNEComment	VARCHAR2(512)	—	Comment for the GNE.

Column Name	Oracle Data Type	Integrity Constraints	Description
GroupOption	NUMBER(1)	DEFAULT 0	Values are: 0: Default grouping 1: Group by GNE 2: Group by Subnet
GroupPrefix	VARCHAR2(12)	—	Group prefix that may be used for group names.
UserID	NUMBER(4)	DEFAULT 0	Value of 0 means no owner.

1.1.54 Group_Info_Table

The Group_Info_Table contains information about all groups defined in the Domain Explorer tree.

Primary key: GROUP_INFO_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-52: Group_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
GroupID	NUMBER(4)	NOT NULL	ID of the group; a unique sequence number generated by the database.
GroupName	VARCHAR2(64)	UNIQUE NOT NULL	Name of the group.
GroupDescription	VARCHAR2(256)	—	Description field.
GroupUserLabel	VARCHAR2(64)	—	Users for this group.
GroupLocationName	VARCHAR2(64)	—	Location of this group.

1.1.55 GW_SNMP_OSS_Info_Table

The GW_SNMP_OSS_Info_Table contains information about the Prime Optical GateWay/Simple Network Management Protocol (SNMP) Operations Support System (OSS). The Prime Optical GateWay/SNMP feature provides an SNMP trap forwarding service, where any trap received by Prime Optical is forwarded to the set of defined trap destinations.

Primary key: GW_SNMP_OSS_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-53: GW_SNMP_OSS_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SNMPV3_USERNAME	VARCHAR2(64)	—	Name of the user who authenticates the SNMPv3 trap.
SNMPV3_COMMUNITY	VARCHAR2(64)	—	The SNMPv3 community name.
SNMPV3_AUTH_PROT	VARCHAR2(64)	—	The SNMPv3 authentication protocol.
SNMPV3_AUTH_PWD	VARCHAR2(64)	—	The SNMPv3 authentication password.
SNMPV3_PRIV_PROT	VARCHAR2(64)	—	The SNMPv3 privacy protocol.
SNMPV3_PRIV_PWD	VARCHAR2(64)	—	The SNMPv3 privacy password.
SNMP_VERSION	NUMBER(4)	—	The SNMP version: 0: SNMPv1 1: SNMPv2 2: SNMPv3
SNMP_COMMUNITY	VARCHAR2(64)	—	The SNMP community name.
SNMP_PORT	NUMBER(10)	—	The SNMP OSS destination port.
SNMPV3_EID	VARCHAR2(64)	—	The SNMPv3 engine identification.
SNMPV3_IP	RAW(16)	NOT NULL	The SNMPv3 IP address.
SNMPV3_IS_IP6	NUMBER(1)	—	The SNMPv3 IPv6 flag.

1.1.56 IF_Cross_Connect

The IF_Cross_Connect table contains information about cross-connects between interfaces on the same NE. Point-to-point (p2p) and point-to-multipoint (p2mp) are supported.

Primary key: None

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-54: IF_Cross_Connect

Column Name	Oracle Data Type	Integrity Constraints	Description
XCDBAccessID	NUMBER(10)	NOT NULL	Unique sequence number (can be a foreign key).

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessId	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
SrcIFDBAccessID	NUMBER(10)	NOT NULL	Source interface endpoint.
DestIFDBAccessID	NUMBER(10)	NOT NULL	Destination interface endpoint.
XCType	NUMBER(2)	DEFAULT 4	Type of switching element used: 1: Unknown 2: Electrical cross-connect 3: Optical cross-connect 4: Autoselect
XCKind	NUMBER(2)	DEFAULT 5	Type of call control to establish cross-connect: 1: Provisioned 2: Automatic 3: Dynamic 4: Protection 5: Other
CreationTime	DATE	—	Time when this cross-connect was created.
Src2DestOperStatus	NUMBER(2)	NOT NULL	Operational state from source to destination: 1: Up 2: Down 3: Dormant 4: Unknown
Dest2SrcOperStatus	NUMBER(2)	NOT NULL	Operational state from destination to source: 1: Up 2: Down 3: Dormant 4: Unknown
Src2DestLastChange	DATE	—	Sysuptime value when the cross-connect in this direction changed its operational state.
Dest2SrcLastChange	DATE	—	Sysuptime value when the cross-connect in this direction changed its operational state.

1.1.57 IF_DWDM_Channel_Group

The IF_DWDM_Channel_Group table contains information about interfaces representing groups of DWDM channels.

Primary key: IF_DWDM_CHAN_FREQ_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-55: IF_DWDM_Channel_Group

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	Reference to IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
MinFrequency	NUMBER(10)	NOT NULL	Lowest frequency in the bitmap for bit 0.
Spacing	NUMBER(4)	NOT NULL	Frequency spacing between successive bits.
BitmapLogic	NUMBER(2)	DEFAULT 1	Whether frequencies in the bitmap are carried or blocked. Values are: 1: Carried 2: Blocked
Bitmap	VARCHAR2(64)	NOT NULL	List of frequencies as a bitmap.

1.1.58 IF_Optical_Phy_Config_Table

The IF_Optical_Phy_Config_Table describes extended physical properties for certain interfaces.

Primary key: IF_OPTICAL_PHY_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-56: IF_Optical_Phy_Config_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	Reference to IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
IfMode	NUMBER(2)	DEFAULT 1	Mode of operation. Values are: 1: mode2R 2: mode3R

Column Name	Oracle Data Type	Integrity Constraints	Description
IfProtocol	NUMBER(2)	DEFAULT 1	Client-side protocol to be monitored. Values are: 1: Other 2: Unknown 3: Gigabit Ethernet 4: 10-Gigabit Ethernet 5: Fiber channel 6: FICON 7: ESCON 8: SONET
IfClockRate	NUMBER(4)	—	Clock rate for client interface.
IfMonitorState	NUMBER(1)	—	Monitor state. Values are: 1: Enabled 2: Disabled
IfLoopback	NUMBER(2)	DEFAULT 1	Loopback status. Values are: 1: No loop 2: Diagnostic loop 3: Line loop 4: Other loop
IfOFC	NUMBER(1)	—	OFC state. Values are: 1: Enabled 2: Disabled
IfLaserSafteyControl	NUMBER(1)	—	Trunk-side laser safety control (LSC) state. Values are: 1: Enabled 2: Disabled
IfFLC	NUMBER(1)	—	State of forward laser control (FLC). Values are: 1: Enabled 2: Disabled

1.1.59 IFIndex_Rule_Table

The IFIndex_Rule_Table contains information on the rules to construct the IfIndex for each object type of an NE model.

Primary key: IFINDEX_RULE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-57: IFIndex_Rule_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelType	NUMBER(10)	NOT NULL	Model type.
ObjectType	NUMBER(10)	NOT NULL	Object type.
AttributeNameConst	NUMBER(10)	NOT NULL	Integer constant of the attribute.
AttrOrder	NUMBER(10)	NOT NULL	Sequential number.
AttrSize	NUMBER(10)	NOT NULL	Size of the attribute in bytes.
AttrType	NUMBER(10)	NOT NULL	Type of the attribute.

1.1.60 Interface_APS_Info

The Interface_APS_Info table contains APS information about interfaces that are members of an APS group.

Primary key: IF_APS_INFO_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-58: Interface_APS_Info

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	Reference to IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ApsGroupName	VARCHAR2(64)	—	APS group name.
ApsChanNumber	NUMBER(2)	DEFAULT –1	Working or protected channel.
ApsChanStatusCurrent	NUMBER(2)	DEFAULT –1	Current state of interface in the group: 1: Locked out 2: Signal degrade 3: Signal failure 4: Switched
ApsChanLastSwitchover	DATE	—	Time since switch to protection channel.
ApsInterfaceState	NUMBER(2)	DEFAULT –1	Active or standby: –1: Unknown 1: Active 2: Standby

1.1.61 Interface_DWDM_Frequency

The Interface_DWDM_Frequency table contains wavelength information about certain optical interfaces carrying DWDM frequencies.

Primary key: IF_DWDM_FREQ_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-59: Interface_DWDM_Frequency

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	Reference to IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
Frequency	NUMBER(10)	NOT NULL	DWDM frequency.

1.1.62 Interface_Stack_Table

The Interface_Stack_Table provides information about underlying and overlaying layer relationships between interfaces.

Primary key: IFSTACK_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-60: Interface_Stack_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UniqueID	NUMBER(10)	NOT NULL	Unique sequence number. Acts as foreign key.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
HighIfDBAccessID	NUMBER(10)	NOT NULL	IFDBAccessID at the higher layer in the stack.
LowIfDBAccessID	NUMBER(10)	NOT NULL	IFDBAccessID at the lower layer in the stack.

Column Name	Oracle Data Type	Integrity Constraints	Description
Status	NUMBER(2)	NOT NULL	Operational state of lower layer. Values are: 1: Up 2: Down 3: Testing 4: Unknown 5: Dormant 6: Not present 7: Lower layer down

1.1.63 Interface_Table

The Interface_Table contains static interface indexes and names for an NE managed by Prime Optical.

Primary key: INTERFACE_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-61: Interface_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEType	NUMBER(1)	NOT NULL	Type of NE. Values are: –1: Common ifNames Any-other-number: NEModelIndex of the NE_INFO_TABLE
ifStaticIndex	NUMBER(4)	NOT NULL	Interface table static index; unique value for each interface on the NE.
ifName	VARCHAR2(64)	NOT NULL	Name of the interface.

1.1.64 IOS_Parser_Static_Tree_Table

The IOS_Paerser_Static_Tree_Table contains information about the IOS parser static tree table.

Primary key: IOS_PARSER_STATIC_TREE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-62: IOS_Parser_Static_Tree_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelIndex	NUMBER(8)	NOT NULL	Model index for the static tree.

Column Name	Oracle Data Type	Integrity Constraints	Description
IOSModelName	VARCHAR(128)	NOT NULL	Index of the static tree.
NodeIndex	NUMBER(8)	NOT NULL	Index of the node.
ParentIndex	NUMBER(8)	NOT NULL	Parent node index: –1 for the root node.
NodeName	VARCHAR(128)	NOT NULL	Name of the IOS node.
NodeValue	VARCHAR(128)	—	Value of the nodes.
CTMAttributeName	VARCHAR(128)	—	Corresponding Prime Optical attribute name.
AttributeLocation	NUMBER(8)	NOT NULL	Position of the attribute in the object index.

1.1.65 L2_FrontPort_Table

The L2_FrontPort_Table contains information about the L2 front port table.

Primary key: L2_FRONTPORT_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-63: L2_FrontPort_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEID	NUMBER(10)	NOT NULL	NE ID.
SLOT_NUMBER	NUMBER(2)	NOT NULL	Slot number.
PORT_NUMBER	NUMBER(2)	NOT NULL	Port number.
NNI	NUMBER(1)	—	Port type.
UNI	NUMBER(1)	—	Port type.
DOT1Q	NUMBER(1)	—	Connection type.
QINQ	NUMBER(1)	—	Connection type.
UNTAGGED	NUMBER(1)	—	Connection type.
BG_COUNT	NUMBER(3)	—	—
SUB_IF_COUNT	NUMBER(3)	—	—
PORT_STATE	NUMBER(1)	—	Port state. Values are: 0: PORT_STATE_DISABLED 1: PORT_STATE_ENABLED
PM_STATUS	NUMBER(1)	—	—
MTU_SIZE	VARCHAR(6)	—	—

Column Name	Oracle Data Type	Integrity Constraints	Description
SPEED	VARCHAR(6)	—	—
DUPLEX	VARCHAR(6)	—	—
FLOWCONTROL_SEND	VARCHAR(30)	—	—
FLOWCONTROL_RECEIVE	VARCHAR(30)	—	—
CDP	VARCHAR(30)	—	Cisco Discovery Protocol.

1.1.66 L2_Interface_QoSClass_Table

The L2_Interface_QoSClass_Table stores all the QoS parameters that are configured on a Layer 2 service drop. The Layer 2 services are configured on a Layer 2 topology. The table is populated when a new Layer 2 service is created or discovered in Prime Optical.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-64: L2_Interface_QoSClass_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEID	NUMBER(10)	NOT NULL	NE ID.
Slot_Number	NUMBER(3)	NOT NULL	Slot number.
Bridge_Group_Number	NUMBER(3)	NOT NULL	Bridge group number.
Port_Number	NUMBER(2)	NOT NULL	Port number.
Port_Sub_Interface	NUMBER(4)	—	Subinterface number on the port.
MATCH_ANY	NUMBER(1)	—	Matches any traffic.
MATCH_DSCP	NUMBER(1)	—	Matches the Differentiated Services Code Point (DSCP). The DSCP value needs to be specified.
MATCH_COS	NUMBER(1)	—	Matches the class of service.
MATCH_IP_PRECEDENCE	NUMBER(1)	—	Matches the IP precedence value.
MATCH_DSCP_VALUE	VARCHAR2(64)	—	DSCP value is used with Match_DSCP.
MATCH_COS_VALUE	VARCHAR2(32)	—	COS value is used with Match_COS.
MATCH_IP_PRECEDENCE_VALUE	VARCHAR2(32)	—	IP precedence value is used with Match_IP_Precedence.

Column Name	Oracle Data Type	Integrity Constraints	Description
ANDOR	NUMBER(1)	—	And/or condition.
CLASS_NAME	VARCHAR2(60)	—	Name of class map.
CIR_TYPE	NUMBER(2)	—	CIR type. Values are: 0: Line rate CIR 1: Rate-limited CIR
MAX_RATE_KBPS	NUMBER(20)	—	Maximum allowed rate in kbps.
BURST_SIZE_BPS	NUMBER(20)	—	Size of the burst in bps.
CIR_COS_TYPE	NUMBER(2)	—	Committed Class of Service (CoS) mark type. Values are: 0: Trust 1: Mark CoS
CIR_COS_VALUE	NUMBER(2)	—	If CIR_COS_TYPE is 1, this is the CoS value.
EXCESS_ACTION	NUMBER(2)	—	Excess traffic action. Values are: 0: Discard 1: Allow
PIR_BURST_BPS	NUMBER(20)	—	Peak information rate burst value in bps.
PIR_KBPS	NUMBER(20)	—	Peak information rate in kbps.
PIR_COS_TYPE	NUMBER(2)	—	PIR CoS type. Values are: 0: Trust 1: Mark CoS
PIR_COS_VALUE	NUMBER(2)	—	PIR CoS marking value.
VIOLATE_ACTION	NUMBER(2)	—	Violate action. Values are: 0: Discard 1: Allow
VIOLATE_COS_VALUE	NUMBER(2)	—	CoS marking for violation traffic.
BEFF_TYPE	NUMBER(2)	—	Best effort type. Values are: 0: Line rate 1: Rate limited
BEFF_RATE_KBPS	NUMBER(20)	—	Best effort maximum rate in kbps.
BEFF_BURST_SIZE_BPS	NUMBER(20	—	Best effort burst size in bps.

1.1.67 L2_Interface_Table

The L2_Interface_Table contains information about L2 interfaces.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-65: L2_Interface_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEID	NUMBER(10)	NOT NULL	NE ID.
SLOT_NUMBER	NUMBER(3)	NOT NULL	Slot number.
BRIDGE_GROUP_NUMBER	NUMBER(3)	NOT NULL	Bridge group number.
PORT_NUMBER	NUMBER(2)	NOT NULL	Port number.
PORT_SUB_INTERFACE	NUMBER(4)	—	Port subinterface.
INTERFACE_NAME	VARCHAR2(30)	—	Interface name.
INTERFACE_TYPE	NUMBER(3)	—	Interface type: 1: POS 2: FAST ETHERNET 3: GIGABIT ETHERNET
PORT_TYPE	NUMBER(2)	—	Port types: 1: UNI 2: NNI
CONNECTION_TYPE	NUMBER(2)	—	Connection types: 1: QinQ 2: Dot1Q 3: Untagged
PORT_STATE	NUMBER(2)	—	Port state. Values are: 0: PORT_STATE_DISABLED 1: PORT_STATE_ENABLED
POLICING_TYPE	NUMBER(2)	—	Policing type.
QOSTEMPLATE_NAME	VARCHAR2(30	—	Name of the QoS template.
MTU_SIZE	VARCHAR2(30)	—	Maximum transmission unit (MTU) size.

Column Name	Oracle Data Type	Integrity Constraints	Description
SPEED	VARCHAR2(30)	—	Speed: 10: MIN_SPEED 100: MAX_SPEED auto: AUTO_SPEED
DUPLEX	VARCHAR2(30)	—	Duplex value: full: FULL_DUPLEX half: HALF_DUPLEX auto: AUTO_DUPLEX
FLOWCONTROL_SEND	VARCHAR2(30)	—	Flow Control (send) value.
FLOWCONTROL_RECEIVE	VARCHAR2(30)	—	Flow Control (receive) value.
RSTP_STATUS	NUMBER(2)	—	RSTP_STATUS has the following values: 0: DISABLE 1: ENABLE
CDP	VARCHAR2(30)	—	Cisco Discovery Protocol.
IP_ADDRESS	VARCHAR2(50)	—	IP address.
SUBNET_MASK	VARCHAR2(50)	—	Subnet mask.

1.1.68 L2_Ipsla_Attribute_Table

The L2_Ipsla_Attribute_Table contains information about the attributes of IP service-level agreements (SLAs) that were created for a specific Layer 2 service.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-66: L2_Ipsla_Attribute_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
Source_NeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node.
Source_PhyLoc	NUMBER(20)	NOT NULL	Source physical location.
Ipsla_Number	NUMBER(10)	NOT NULL	Unique number to identify the IP SLA.
AttributeName	VARCHAR2(50)	NOT NULL	Name of the IP SLA attribute.
AttributeValue	VARCHAR2(50)	—	Value of the IP SLA attribute.

1.1.69 L2_Ipsla_Table

The L2_Ipsla_Table contains information about the IP SLAs that were created for a specific Layer 2 service.

Primary key: IPSLA_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-67: L2_Ipsla_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoNodeID	NUMBER(10)	NOT NULL	Node ID of the circuit's source node.
TopoUniqueId	NUMBER(10)	NOT NULL	Unique number within the node.
Vlan_Number	NUMBER(10)	NOT NULL	Unique number to identify the VLAN.
Ipsla_Number	NUMBER(10)	NOT NULL	Unique number to identify the IP SLA.
Destination_NeId	NUMBER(10)	—	ID of the destination NE.
Destination_PhyLoc	NUMBER(20)	—	Destination physical location.
Destination_Addr	VARCHAR2(50)	—	Destination IP address.
Source_NeId	NUMBER(10)	—	Source NE ID.
Source_PhyLoc	NUMBER(20)	—	Source physical location.
Source_Addr	VARCHAR2(50)	—	Source IP address.
Operation_Type	VARCHAR2(50)	—	Operation type. The possible values are: jitter: JITTER echo: ECHO
Operation_status	NUMBER(10)	—	Operation status. The possible values are: 0: NOT_SCHEDULED 1: SCHEDULED_RUNNING 2: SCHEDULED_NOT_RUNNING

1.1.70 L2QoSTemplate_QoSClass_Table

The L2QoSTemplate_QoSClass_Table is used to store all the Quality of Service (QoS) parameters defined in a QoS profile. QoS profiles are used during Layer 2 services creation to specify the QoS parameters for the service drops.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-68: L2QoSTemplate_QoSClass_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
QoSTemplateID	NUMBER(6)	NOT NULL	QoS template ID.
MATCH_ANY	NUMBER(1)	—	Matches any traffic.
MATCH_DSCP	NUMBER(1)	—	Match DSCP. DSCP value needs to be specified.
MATCH_COS	NUMBER(1)	—	Match class of service.
MATCH_IP_PRECEDENCE	NUMBER(1)	—	Match IP precedence value.
MATCH_DSCP_VALUE	VARCHAR2(64)	—	DSCP value is used along with Match_DSCP.
MATCH_COS_VALUE	VARCHAR2(32)	—	COS value is used along with Match_COS.
MATCH_IP_PRECEDENCE_VALUE	VARCHAR2(32)	—	IP precedence value is used along with Match_IP_Precedence.
ANDOR	NUMBER(1)	—	And/or condition.
CLASS_NAME	VARCHAR2(60)	—	Name of class map.
CIR_TYPE	NUMBER(2)	—	CIR type. Values are: 0: Line rate CIR 1: Rate-limited CIR
MAX_RATE_KBPS	NUMBER(20)	—	Maximum allowed rate in kbps.
BURST_SIZE_BPS	NUMBER(20)	—	Size of the burst in bps.
CIR_COS_TYPE	NUMBER(2)	—	Committed CoS mark type. Values are: 0: Trust 1: Mark CoS
CIR_COS_VALUE	NUMBER(2)	—	If CIR_COS_TYPE is 1, this is the CoS value.
EXCESS_ACTION	NUMBER(2)	—	Excess traffic action. Values are: 0: Discard 1: Allow
PIR_BURST_BPS	NUMBER(20)	—	Peak information rate burst value in bps.
PIR_KBPS	NUMBER(20)	—	Peak information rate in kbps.
PIR_COS_TYPE	NUMBER(2)	—	PIR CoS types. Values are: 0: Trust 1: Mark CoS

Column Name	Oracle Data Type	Integrity Constraints	Description
PIR_COS_VALUE	NUMBER(2)	—	PIR CoS marking value.
VIOLATE_ACTION	NUMBER(2)	—	Violate action. Values are: 0: Discard 1: Allow
VIOLATE_COS_VALUE	NUMBER(2)	—	CoS marking for violation traffic.
BEFF_TYPE	NUMBER(2)	—	Best effort type. Values are: 0: Line rate 1: Rate limited
BEFF_RATE_KBPS	NUMBER(20)	—	Best effort maximum rate in kbps.
BEFF_BURST_SIZE_BPS	NUMBER(20)	—	Best effort burst size in bps.
CLASSORDER	NUMBER(20)	DEFAULT 1, NOT NULL	Position of the class policy within the QoS template.

1.1.71 L2QoSTemplate_Table

The L2QoSTemplate_Table contains information about the L2 QoS template.

Primary key: L2QOSTEMPLATE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-69: L2QoSTemplate_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
QoSTemplateID	NUMBER(6)	NOT NULL	QoS template ID.
QoSTemplateName	VARCHAR2(50)	NOT NULL	Unique name of the QoS template.
QoSTemplateType	NUMBER(2)	—	QoS template types: 1: BEST_EFFORT 2: CIR_PIR 3: ADVANCED
QoSTemplateDescription	VARCHAR2(257)	—	QoS template description.
ClassOrder	NUMBER(20)	—	Position of the class policy within the QoS template.

1.1.72 L2Topology_Circuits_Table

The L2Topology_Circuits_Table contains information about the L2 topology.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-70: L2Topology_Circuits_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoNodeID	NUMBER(10)	NOT NULL	ID of the circuit's source node where Point-to-Point topology is used. Where Shared Packet Ring (SPR) is used, it is the node ID of the node that is equal to the maximum of all node IDs in the SPR.
TopoUniqueId	NUMBER(10)	NOT NULL	Unique number within the node.
NodeID	NUMBER(10)	NOT NULL	Node ID of the circuit's source node.
UniqueId	NUMBER(10)	NOT NULL	Unique number within the node.
LinkId	NUMBER(10)	NOT NULL	Unique number within the link.

1.1.73 L2Topology_Table

The L2Topology_Table contains information about L2 topologies.

Primary key: L2TOPOLOGY_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-71: L2Topology_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoNodeID	NUMBER(10)	NOT NULL	Node ID of the circuit’s source node.
TopoUniqueID	NUMBER(10)	NOT NULL	Unique number within the node.
TopoType	NUMBER(2)	—	Topology types. Values are: 1: Point to point 2: SPR 3: Hub and spoke
Topology_Name	VARCHAR2(50)	—	Prime Optical name for the topology.
TopoDescription	VARCHAR2(256)	—	Description of topology.
AlarmSeverity	NUMBER(2)	—	Alarm severity.

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoStatus	NUMBER(2)	—	Status of the topology.
TopoSize	NUMBER(10)	—	Size of the topology.
TopoProtection	NUMBER(2)	—	Topology protection.
TopoResyncStatus	NUMBER(2)	—	Topology resynchronization states. Values are: 0: TOPO_SYNC_NEEDED 1: TOPO_SYNC_IN_PROGRESS 2: TOPO_SYNC_COMPLETED
TopoOperStatus	NUMBER(2)	—	Operational status of the topology.
BW_SP_Management	NUMBER(3)	—	Displays the bandwidth percentage used for the service provider (SP) management traffic class.
BW_Committed	NUMBER(3)	—	Displays the bandwidth percentage used for the committed information rate (CIR) traffic class.
BW_Best_Effort	NUMBER(3)	—	Displays the bandwidth percentage used for the best-effort traffic class.
TOPOCONSTATE	NUMBER(2)	—	Displays Prime Optical status after trying to synchronize the configuration of all the multilayer cards in the topology.
COS_COMMIT	NUMBER(3)	—	Committed CoS value.
BW_Utilized	NUMBER(20)	—	Percentage of utilized bandwidth.
BW_Available	NUMBER(20)	—	Percentage of available bandwidth.
BW_AVVID_CONTROL	NUMBER(3)	—	Percentage of bandwidth for AVVID call control.
COS_SP_MANAGEMENT	VARCHAR2(128)	—	CoS for SP management.
COS_COMMITTED	VARCHAR2(128)	—	CoS for the committed rate.
COS_AVVID_VOICEVIDEO	VARCHAR2(128)	—	CoS for AVVID voice video.
COS_AVVID_CONTROL	VARCHAR2(128)	—	CoS for AVVID control.
BW_Group1	NUMBER(3)	—	First bandwidth group.

Column Name	Oracle Data Type	Integrity Constraints	Description
BW_Group2	NUMBER(3)	—	Second bandwidth group.
COS_GROUP1	VARCHAR2(128)	—	First CoS group.
COS_GROUP2	VARCHAR2(128)	—	Second CoS group.
TopoAliasName	VARCHAR2(64)	—	Circuit alias name.
BW_A0	NUMBER(4)	—	Absolute bandwidth value in megabits per second (Mbps) for Class A0, which conforms to the reserved rate on the ring. The range is from 0 to 48 Mbps.
BW_A1	NUMBER(4)	—	Absolute bandwidth value in Mbps for Class A1. The range is from 1 to 48 Mbps.
BW_B_CIR	NUMBER(4)	—	Absolute bandwidth value in Mbps for Class B-CIR. The range is from 1 to 48 Mbps.
CLASS_SP_MGMT	VARCHAR2(128)	—	Specifies that the user-defined class SP_MANAGEMENT is mapped to a specific service class a, b, or c as defined by the 802.17 RPR standard.
CLASS_CIR	VARCHAR2(128)	—	Specifies that the user-defined class CLASS_CIR is mapped to a specific service class a, b, or c as defined by the 802.17 RPR standard.
CLASS_BEST_EFFORT	VARCHAR2(128)	—	Specifies that the user-defined class BEST_EFFORT is mapped to a specific service class a, b, or c as defined by the 802.17 RPR standard.
CLASS_AVVID_CTRL	VARCHAR2(128)	—	Specifies that the user-defined class AVVID_CTRL is mapped to a specific service class a, b, or c as defined by the 802.17 RPR standard.
CLASS_AVVID_VOICEVIDEO	VARCHAR2(128)	—	Specifies that the user-defined class AVVIDVOICEVIDEO is mapped to a specific service class a, b, or c as defined by the 802.17 RPR standard.

1.1.74 L2_Vlan_Table

The L2_Vlan_Table contains information about the VLANs in an L2 topology.

Primary key: VLAN_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-72: L2_Vlan_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TopoNodeID	NUMBER(10)	NOT NULL	Node ID of the circuit's source node where Point-to-Point topology is used. Where SPR is used, it is the node ID of the node that is equal to the maximum of all node IDs in the SPR.
TopoUniqueId	NUMBER(10)	NOT NULL	Unique number within the node.
Vlan_Number	NUMBER(10)	NOT NULL	Unique number to identify the VLAN.
Service_Id	VARCHAR2(50)	—	Prime Optical service ID.
Customer_Id	VARCHAR2(50)	—	Prime Optical customer ID.
VlanStatus	NUMBER(2)	—	VLAN status.
ManagedVlan	NUMBER(1)	—	VLAN in the L2 topology.

1.1.75 Link_Layer_Mapping_Table

The Link_Layer_Mapping_Table maps information between link layer integer values and strings. Integer values are stored in the database and are converted to string values for user display. The Link_Layer_Mapping_Table is used to convert the integer values to string values.

Primary key: LINK_LAYER_MAPPING_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-73: Link_Layer_Mapping_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkLayerId	NUMBER(2)	—	Integer representing the layer ENUM.
LinkLayerString	VARCHAR2(30)	—	Corresponding string value of LinkLayerId.

1.1.76 Link_Table

The Link_Table contains information about the links on the Prime Optical Network Map.

Primary key: LINK_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-74 : Link_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkID	NUMBER(8)	NOT NULL	Unique number created by the database.
LinkSrcNode	NUMBER(10)	NOT NULL	Source node.
LinkSrcModuleType	NUMBER(8)	NOT NULL	Module type at the source.
LinkSrcPhysicalLoc	NUMBER(20)	NOT NULL	Source node physical location.
LinkSrcIfIndex	NUMBER(10)	NOT NULL	Whether there is a static index at the source. This is not applicable to the Cisco ONS 15327 or Cisco ONS 15454.
LinkSrcObjectType	NUMBER(10)	—	Type from object ID.
LinkDstNode	NUMBER(10)	NOT NULL	Source node (node network element database ID [NEDBID]).
LinkDstModuleType	NUMBER(8)	NOT NULL	Module type at the source.
LinkDstPhysicalLoc	NUMBER(20)	NOT NULL	Source node physical location.
LinkDstModelType	NUMBER(4)	NOT NULL	Destination node physical location.
LinkDstIfIndex	NUMBER(10)	NOT NULL	Whether there is a static index at the destination. This is not applicable to the ONS 15327 or ONS 15454.
LinkDstObjectType	NUMBER(10)	—	Object type from object ID.

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkLayer	NUMBER(2)	NOT NULL	Type of link layer. Values are: 0: Link Layer Physical 1: Link Layer OTS 2: Link Layer OMS 3: Link Layer OCH 4: Link Layer SONET_SDH 5: Link Layer Layer2 6: Link Layer Invalid 7: Link Layer Layer3 8: Link Layer OTUk 9: Link Layer Composite 10: Link Layer OTT Tunnel
LinkSize	NUMBER(2)	NOT NULL	Size of the link. Values are: 0: OC-3 1: OC-12 2: OC-48 3: OC-192 4: OC-768 5: DWDM 6: DWDM 1CH 7: DWDM 8CH 8: DWDM 16CH 9: DWDM 32CH 10: OTS 11: VT tunnel 12: 10BaseT 13:100BaseT 14: GIGE 15: Tunnel 16: Unknown 17: Not applicable 18: STM 1 19: STM 4 20: STM 16 21: STM 64 22: STM 256 23: OMS 24: 10GIG ITU 25: 2_5GIG ITU 26: LAYER2 27: 10 GIGE 28: ROUTING PROTOCOL

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkValid	NUMBER(1)	NOT NULL	Link validity. Values are: 0: Valid 1: Invalid
LegIndex	NUMBER(1)	DEFAULT 1	Link sequence number of the leg.
CableGroupId	NUMBER(8)	DEFAULT 0	Same for all legs of a link.
LinkProvType	NUMBER(1)	NOT NULL	Type of link provisioning. Values are: 0: Link autodiscovered 1: Link manual 2: Link virtual 3: Link unmanaged 4: Link server trail
LinkConnectionType	NUMBER(1)	NOT NULL DEFAULT 0	Connection type of the link. Values are: 0: Inter-NE 1: Intra-NE
LinkType	NUMBER(2)	NOT NULL	Type of link. Values are: 0: OC-N 1: VT tunnel 2: DWDM 3: CDL 4: PCA 5: Ethernet 6: POS 7: DPT 8: MPLS Tunnel 9: IPSEC Tunnel 10: Not applicable 11: Y-cable 12: LAP 13: Virtual 14: BGP CONFEDERATE 15: BGP EXTERNAL 16: BGP INTERNAL 17: SERVERTRAIL 18: OCHTRAIL TUNNEL

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkProtType	NUMBER(2)	NOT NULL	Protection scheme of the link. Values are: 0: BLSR two fiber (2F) 1: 1+1 2: VT tunnel 3: Unprotected 4: BLSR four fiber (4F) 5: BLSR two fiber PCA 6: BLSR four fiber PCA 7: BLSR 8: Unprotected 9: PCA 10: Unknown 11: Not applicable 12: Partially protected 13: Y-cable
LinkProtRole	NUMBER(1)	NOT NULL	Protection role of the link. Values are: 0: Link protect 1: Link working 2: Link protection not applicable
LinkDirection	NUMBER(1)	NOT NULL	Direction of the link. Values are: 0: Bidirectional 1: Unidirectional
LinkState	NUMBER(1)	NOT NULL	State of the link. Values are: 0: Link state unknown 1: Link state up 2: Link state down
useForRouting	NUMBER(1)	DEFAULT 1 NOT NULL	Whether the link should be used for routing. Values are: 0: False 1: True 2: Unknown
LinkName	VARCHAR2(256)	NOT NULL	User-defined name of the link.
LinkDescription	VARCHAR2(256)	—	Any user-defined notes for the link.
NumCriticalAlarms	NUMBER(8)	DEFAULT 0	Number of critical alarms on the link.
NumMajorAlarms	NUMBER(8)	DEFAULT 0	Number of major alarms on the link.
NumMinorAlarms	NUMBER(8)	DEFAULT 0	Number of minor alarms on the link.

Column Name	Oracle Data Type	Integrity Constraints	Description
NumWarningAlarms	NUMBER(8)	DEFAULT 0	Number of warning alarms on the link.
LinkSrcPhyLocUnManaged	VARCHAR2(30)	DEFAULT NULL	Source physical location if unmanaged NE is a termination point.
LinkDestPhyLocUnmanaged	VARCHAR2(30)	DEFAULT NULL	Destination physical location if unmanaged NE is a termination point.
LinkUnmanaged	NUMBER(1)	DEFAULT 0	Whether a link has at least one unmanaged NE. Values are: 0: False 1: True
LinkCost	NUMBER(10)	DEFAULT 1	Link weight.
LinkSrlgList	VARCHAR2(256)	DEFAULT NULL	User-defined SRL groups for the link.
STSize	NUMBER(8)	DEFAULT –1	Size of a server trail.
STType	NUMBER(8)	DEFAULT –1	Type of a server trail.
STCount	NUMBER(8)	DEFAULT –1	Number of server trails created.
STNodeId	NUMBER(10)	DEFAULT –1	Node ID of a server trail.
STUniqueId	NUMBER(10)	DEFAULT –1	Unique ID of a server trail.
LinkAliasName	VARCHAR2(256)	—	User-defined alias for the link.
LinkBlsrRingId	VARCHAR(8)	—	BLSR ring ID for the link.
LinkDstCTCIndex	NUMBER(8)	—	Static CTC index at the destination. This is an internal field.
LinkSrcCTCIndex	NUMBER(8)	—	Static CTC index at the source. This is an internal field.
CktNodeID	NUMBER(10)	—	Node ID of the circuit's source node. This is an internal field.
CktUniqueID	NUMBER(10)	—	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.

1.1.77 LinkId_To_UniqueId

The LinkId_To_UniqueId table stores the relation among the Transport Profile (TP) tunnel links and the TP services.

Primary key: LINKID_TO_UNIQUEID_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-75: LinkId_To_UniqueId

Column Name	Oracle Data Type	Integrity Constraints	Description
LINKID	NUMBER(10)	NOT NULL	Key of the link in the LINK_TABLE table. This is an internal field.
UNIQUEID	NUMBER(10)	NOT NULL	Key of the TP service in the MPLS_TP_EP table. This is an internal field.

1.1.78 Managed_Et_Table

The Managed_Et_Table contains information about the managed entity table.

Primary key: MANAGED_ET_TAB_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-76: Managed_Et_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
Mng_Entity_Type	NUMBER(5)	NOT NULL	Managed entity type. Values are: 1: Circuit termination point (CTP) 2: VLAN
Mng_Entity_Id_Num	NUMBER(10)	NOT NULL	Managed entity ID number.
Mng_Entity_Id2_Num	NUMBER(10)	NOT NULL	Managed entity ID 2 number.
NeDBAccessId	NUMBER(10)	NOT NULL	NE database ID.
FIELD1_Num	NUMBER(10)	NOT NULL	Polymorphic field module type.
FIELD2_Num	NUMBER(20)	NOT NULL	Polymorphic field physical location.
FIELD3_Num	NUMBER(20)	NOT NULL	Polymorphic field If index.
FIELD4_Num	NUMBER(10)	NOT NULL	Polymorphic field object index.
Customer_Id_Str	VARCHAR2(256)	—	Customer ID string representation.
Service_Id_Str	VARCHAR2(256)	—	Service ID string representation.

1.1.79 Matrix_Flow_Domain_Table

The Matrix_Flow_Domain_Table contains matrix flow domains (MFDs) and their associated information.

Primary key: MFD_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-77: Matrix_Flow_Domain_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
MFD_NE_DB_ID	NUMBER(10)	NOT NULL	MFD ID.
FD_NAME	VARCHAR2(128)	—	Flow domain to which this MFD is associated, if any.
MFD_USER_LABEL	VARCHAR2(64)	—	Description of the MFD.
MFD_NATIVE_NAME	VARCHAR2(64)	NOT NULL	Native name of the MFD.
MFD_OWNER	VARCHAR2(64)	—	Owner of the MFD.
MFD_TRANSM_PARAMS	VARCHAR2(512)	—	Transmission parameters of the MFD.
MFD_NET_ACCESS_DOMAIN	VARCHAR2(64)	—	Network access domain of the MFD.
MFD_FLEXIBLE	NUMBER(1)	—	Flexibility of the MFD.
MFD_ADDITIONAL_INFO	VARCHAR2(512)	—	Additional information of the MFD.

1.1.80 MIB2_Interface_Table

The MIB2_Interface_Table corresponds to the MIB2 table in IF-MIB.

Primary key: MIB2_INTERFACE_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-78: MIB2_Interface_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	Unique sequence number generated by the database.
NEDBAccessID	NUMBER(10)	NOT NULL	Refers to the NEDBAccessID in the NE_INFO_TABLE.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location.
ModuleType	NUMBER(8)	DEFAULT 0	Module type.
NEIfIndex	NUMBER(20)	NOT NULL	Encoded value for name derivation.
ObjectType	NUMBER(10)	NOT NULL	Object type.
ModelType	NUMBER(5)	NOT NULL	Model type.

Column Name	Oracle Data Type	Integrity Constraints	Description
PhyIfIndex	NUMBER(10)	DEFAULT 0	Entity PhyIndex that realizes this interface.
IfIndex	NUMBER(10)	NOT NULL	ifindex in mib-2.
IfType	NUMBER(4)	NOT NULL	ifType as described in IF-MIB:ifTable MIB.
IfExtType	NUMBER(4)	DEFAULT 0	Extended type (if any).
IfSpecificType	NUMBER(4)	DEFAULT 0	Any device-specific type.
IfDescr	VARCHAR2(255)	NOT NULL	ifDescr as described in IF-MIB:ifTable MIB.
IfName	VARCHAR2(50)	NOT NULL	ifName as described in IF-MIB:ifTable MIB.
IfSpeed	VARCHAR2(15)	—	ifSpeed as described in IF-MIB:ifTable MIB.
IfAdminStatus	NUMBER(2)	NOT NULL	Administrative state of the interface. Values are: 1: Up 2: Down 3: Testing
IfOperStatus	NUMBER(2)	NOT NULL	Operational state of the interface. Values are: 1: Up 2: Down 3: Testing 4: Unknown 5: Dormant 6: Not present 7: Lower layer down
IfHighSpeed	VARCHAR2(15)	—	High speed that is capable on the interface, in 1,000,000 bits per second. The value is derived from the IF-MIB.

1.1.81 MLPPP_Bundle_Data

Primary key: None

Table 1-79: MLPPP_Bundle_Data

Column Name	Oracle Data Type
timestamp	NUMBER(10)
l_node_id	NUMBER(10)
l_slot	NUMBER(5)
l_bay	NUMBER(5)
mlppp_bundleID	NUMBER(5)

Column Name	Oracle Data Type
subobject_type	NUMBER(5)
stat_type	NUMBER(5)
bucket_type	NUMBER(5)
totalId	NUMBER(22,2)
peak	NUMBER(22,2)

1.1.82 MLPPP_Link_Data

Primary key: None

Table 1-80: MLPPP_Link_Data

Column Name	Oracle Data Type
timestamp	NUMBER(10)
l_node_id	NUMBER(10)
l_slot	NUMBER(5)
l_bay	NUMBER(5)
mlppp_linkID	NUMBER(5)
subobject_type	NUMBER(5)
stat_type	NUMBER(5)
bucket_type	NUMBER(5)
totalId	NUMBER(22,2)
peak	NUMBER(22,2)

1.1.83 MLPPP_Mux_Data

Primary key: None

Table 1-81: MLPPP_Mux_Data

Column Name	Oracle Data Type
timestamp	NUMBER(10)
l_node_id	NUMBER(10)
l_slot	NUMBER(5)
l_bay	NUMBER(5)
mlppp_bundleID	NUMBER(5)
subobject_type	NUMBER(5)
stat_type	NUMBER(5)
bucket_type	NUMBER(5)

Column Name	Oracle Data Type
totald	NUMBER(22,2)
peak	NUMBER(22,2)

1.1.84 Model_Type_Table

The Model_Type_Table contains a list of NE models supported by Prime Optical.

Primary key: MODEL_TYPE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-82: Model_Type_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelType	NUMBER(4)	NOT NULL	Model type. Values are: 0: Unknown 1: Cisco ONS 15303 2: Cisco ONS 15304 3: Cisco ONS 15454 5: Cisco ONS 15327 7: Cisco ONS 15200 9: Cisco ONS 15540 11: Cisco ONS 15540 XPS 12: Cisco ONS 15530 14: Cisco ONS 15454 SDH 15: Cisco ONS 15216 EDFA2 16: Cisco ONS 15216 Optical Add/Drop Multiplexer (OADM) 17: Cisco ONS 15216 DWDM 18: Cisco ONS 15600 19: Cisco ONS 15216 OSC 20: Cisco ONS 15216 DCU 22: Cisco 1530523: Cisco ONS 15216 UNIDIRFILTER 24: Unmanaged NE 25: Cisco ONS 15216 EDFA3 26: Cisco HFR 27: GSR IOX 28: Cisco 6509 29: Cisco ONS 15310 CL 30: Cisco ONS 15600 SDH 31: Cisco ONS 15216 EDFA1 32: Cisco 15305 CTC 33: Cisco MDS 9000 35: Cisco 15365

Column Name	Oracle Data Type	Integrity Constraints	Description
			37: Cisco 7600 38: Cisco 15310MA 39: Cisco 15310MASDH 40: Cisco CRS 41: Cisco ASR9K 42: Carrier Packet Transport 600 43: Carrier Packet Transport 600 SDH 44: Carrier Packet Transport 200 45: Carrier Packet Transport 200 SDH
ModelTypeName	VARCHAR2(64)	NOT NULL	Name of the model.
ModelSortOrder	NUMBER(2)	—	—

1.1.85 Module155xx_Community_String

The Module155xx_Community_String table contains the community strings for ONS 15530 and ONS 15540 devices.

Primary key: MODULE155XX_TARGET_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-83: Module155xx_Community_String

Column Name	Oracle Data Type	Integrity Constraints	Description
Target	VARCHAR2(64)	NOT NULL	Target IP address.
ReadString	VARCHAR2(64)	DEFAULT 'public'	Read community string.
Retry	NUMBER(2)	DEFAULT 1	Number of times to retry a connection.
Timeout	NUMBER(2)	DEFAULT 6	Time to wait before timing out a connection attempt.
WriteString	VARCHAR2(64)	NOT NULL	Write community string.

1.1.86 Module155xx_VTOID_Table

The Module155xx_VTOID_Table contains relating the module type to the vendor object ID.

Primary key: MODULE155XX_TYPE_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-84: Module155xx_VTOID_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleType	NUMBER(8)	NOT NULL	Module type.
VendorOID	VARCHAR2(64)	NOT NULL	Vendor object ID.

1.1.87 Module_Type_Mll_Info

The Module_Type_Mll_Info table defines the possible layer rates for each type of module.

Primary key: MLL_TYPE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-85: Module_Type_Mll_Info

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleType	NUMBER(8)	NOT NULL	Type of module, filled in at installation time.
LayerRate	NUMBER(2)	—	Link layers.
ModelType	NUMBER(4)	—	Type of model.

1.1.88 Module_Type_Table

The Module_Type_Table contains inventory information about the NE module types.

Primary key: MODULE_TYPE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-86 : Module_Type_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleType	NUMBER(8)	NOT NULL	Type of module, filled in at installation time. Possible values are: 1: ISR A 2: ISR E 3: Cerent DS1 4: Cerent DS3 5: Cerent DS3XM 6: Cerent EC1 7: Cerent EPOS 100 8: Cerent EPOS 1000 9: Cerent OC3 10: Cerent OC12 11: Cerent OC48 60: Cerent OC192 86: Cerent STM64_64 87: Cerent STM16_16
ModuleName	VARCHAR2(64)	—	Name of the equipment.
ModelType	NUMBER(4)	—	Type of model.

1.1.89 MPLS_Data_Service_Table

The MPLS_Data_Service_Table stores information about a single cross-connect contributor for each service.

Primary key: None

Tablespace: NGXP_DATA_TABLESPACE

Table 1-87: MPLS_Data_Service_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	Packet Transport Switch (PTS) ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
SERVICEDATA	BLOB	NOT NULL	Data packet that describes the cross-connect information. This is an internal field.

1.1.90 MPLS_LSP_CTP

The MPLS_LSP_CTP contains the information about Label Switched Path (LSP) contributors for each TP service.

Primary key: MPLS_LSP_CTP_PK

Table 1-88: MPLS_LSP_CTP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
WORKPROT	NUMBER(10)	NOT NULL	Whether path is working or protected: 0: Working path 1: Protected path
CTCINDEX	NUMBER(10)	NOT NULL	Physical object ID managed by the NE. This is an internal field.
PHYLOC	NUMBER(10)	NOT NULL	Physical object ID managed by the Prime Optical system. This is an internal field.
LSPNUM	NUMBER(10)	NOT NULL	LSP number has been provisioned.
LINKNUM	NUMBER(10)	NOT NULL	Link number has been provisioned.
LOCALLABEL	NUMBER(10)	NOT NULL	Local label number has been provisioned.
OUTLABEL	NUMBER(10)	NOT NULL	Out label number has been provisioned.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical ID used to identify the contributor in the service. This is an internal field.
LINKID	NUMBER(10)	NOT NULL	Link ID used for routing. This is an internal field.
CTPTYPE	NUMBER(1)	NOT NULL	Type of circuit termination point. Values are: endpoint middle point
SERVICENAME	VARCHAR(256)	—	Service name.
ISLOCKOUT	NUMBER(1)	NOT NULL	Unidirectional Path Switched Ring (UPSR) protection status. Values are: 0: unlock 1: lockout
ISACTIVE	NUMBER(1)	NOT NULL	UPSR protection status. Values are: 0: inactive 1: active
LSPPATHDEFDIRECTION	NUMBER(1)	NOT NULL	Direction of the LSP path.

Column Name	Oracle Data Type	Integrity Constraints	Description
SLOTMODULETYPE	NUMBER(10)	NOT NULL	Slot module type. This is an internal field.
PORTMODULETYPE	NUMBER(10)	NOT NULL	Port module type. This is an internal field.

1.1.91 MPLS_LSP_XC

The MPLS_LSP_XC stores LSP services cross-connections.

Primary key: MPLS_LSP_XC_PK

Table 1-89: MPLS_LSP_XC

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the NE in the NE_INFO_TABLE. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	CTC index related to the packet transport switch.
SERVICEID	NUMBER(10)	NOT NULL	Identifier of the service.
WFWDCTCINDEX	NUMBER(10)	NOT NULL	CTC index of the physical object on the working forward direction.
WREVCTCINDEX	NUMBER(10)	NOT NULL	CTC index of the physical object on the working reverse direction.
PFWDCTCINDEX	NUMBER(10)	NOT NULL	CTC index of the physical object on the protected forward direction.
PREVCTCINDEX	NUMBER(10)	NOT NULL	CTC index of the physical object on the protected reverse direction.
WLSPNUM	NUMBER(10)	NOT NULL	Working LSP number.
PLSPNUM	NUMBER(10)	NOT NULL	Protected LSP number.
XCTYPE	NUMBER(1)	NOT NULL	Cross-connection type.
SERVICENAME	VARCHAR2(256)	—	Service name.

1.1.92 MPLS_NE_Table

The MPLS_NE_Table contains the list of network elements equipped with PTF_10GE_4 and PT_10GE_4 cards.

Primary key: MPLS_NE_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-90: MPLS_NE_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.

1.1.93 MPLS_QoS_Action_Table

The MPLS_QoS_Action_Table describes the QoS action of the CPT QoS Editor.

Primary key: ACTION_PK

Table 1-91: MPLS_QoS_Action_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	Primary key.
NAME	VARCHAR(255)	NOT NULL	Action name.
CIRVALUE	FLOAT(126)	—	Value for committed information rate (CIR) QoS parameter.
CIRUNIT	NUMBER(5)	—	Unit of measure for CIR QoS value.
PIRVALUE	FLOAT(126)	—	Value for peak information rate (PIR) QoS parameter.
PIRUNIT	NUMBER(5)	—	Unit of measure for PIR QoS value.
PEAKBURSTSIZEVALUE	FLOAT(126)	—	Value for peak burst size QoS parameter.
PEAKBURSTSIZEUNIT	NUMBER(5)	—	Unit of measure for peak burst size QoS value.
BURSTSIZEVALUE	FLOAT(126)	—	Value for burst size QoS parameter.
BURSTSIZEUNIT	NUMBER(5)	—	Unit of measure for burst size QoS value.
AVERAGERATEVALUE	FLOAT(126)	—	Value for average rate QoS parameter.
AVERAGERATEUNIT	NUMBER(5)	—	Unit of measure for average rate QoS value.
PRIORITYVALUE	FLOAT(126)	—	Value for priority QoS parameter.
PRIORITYUNIT	NUMBER(5)	—	Unit of measure for priority unit QoS value.
MINIMUMBANDWIDTHVALUE	FLOAT(126)	—	Value for minimum bandwidth QoS parameter.

Column Name	Oracle Data Type	Integrity Constraints	Description
MINIMUMBANDWIDTHUNIT	NUMBER(5)	—	Unit of measure for minimum bandwidth QoS value.
REMAININGBANDWIDTHVALUE	FLOAT(126)	—	Value for remaining bandwidth QoS parameter.
REMAININGBANDWIDTHUNIT	NUMBER(5)	—	Unit of measure for remaining bandwidth QoS value.
RATECODE	NUMBER(5)	—	Code for rate value of QoS action.

1.1.94 MPLS_QoS_AM_Table

The MPLS_QoS_AM_Table describes the QoS markers of a QoS action. A marker contains information to mark a traffic packet as per the QoS model. A marker is associated to one QoS Action.

Primary key: None

Table 1-92: MPLS_QoS_AM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	QoS action to which a marker is associated. Acts as foreign key.
ATTRIBUTENAMECODE	NUMBER(5)	—	Internal Code that identifies the attribute to be marked. Options are: Class of Service (CoS) IP precedence Differentiated Services Code Point (DSCP) QoS group
VALUE	NUMBER(5)	—	Value of the marking.

1.1.95 MPLS_QoS_Class_Map_Table

The MPLS_QoS_Class_Map_Table describes the QoS class maps of the CPT QoS Editor.

Primary key: CLASS_MAP_PK

Table 1-93: MPLS_QoS_Class_Map_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	Primary Key.

Column Name	Oracle Data Type	Integrity Constraints	Description
NAME	VARCHAR(255)	NOT NULL	Name of the class map.
COS	VARCHAR(255)	—	CoS values of the class map.
PRECEDENCE	VARCHAR(255)	—	Precedence value of the class map.
DSCP	VARCHAR(255)	—	DSCP values of the class map.
MPLSEXPERIMENTAL	VARCHAR(255)	—	MPLS experimental value of the class map.
QOSGROUP	VARCHAR(255)	—	QoS group value of the class map.
MATCHALL	NUMBER(1)	—	Matching criteria for the class map. Values are: True: matchAll False: matchAny

1.1.96 MPLS_QoS_Conform_PA_Table

The MPLS_QoS_Conform_PA_Table describes the conform actions for a QoS action. A QoS action can have zero or more conform actions.

Primary key: None

Table 1-94: MPLS_QoS_Conform_PA_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	QoS action to which a conform action is associated. Acts as foreign key.
NAMECODE	NUMBER(5)	—	Internal code that identifies the name of the conform action.
VALUE	FLOAT(126)	—	Value of the conform action.
UNIT	NUMBER(5)	—	Unit of measure for the value.

1.1.97 MPLS_QoS_Exceed_PA_Table

The MPLS_QoS_Exceed_PA_Table describes the exceed actions for a QoS action. A QoS action can have zero or more exceed actions.

Primary key: None

Table 1-95: MPLS_QoS_Exceed_PA_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	QoS action to which an exceed action is associated. Acts as foreign key.
NAMECODE	NUMBER(5)	—	Internal code that identifies the name of

Column Name	Oracle Data Type	Integrity Constraints	Description
			the exceed action.
VALUE	FLOAT(126)	—	Value of the exceed action.
UNIT	NUMBER(5)	—	Unit of measure for the value.

1.1.98 MPLS_QoS_Mapper_Table

The MPLS_QoS_Mapper_Table describes the mappers for the QoS table map. A table map can have one or more mappers.

Primary key: None

Table 1-96: MPLS_QoS_Mapper_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	QoS action to which a marker is associated. Acts as foreign key.
QOSGROUP	NUMBER(5)	—	QoS group field of the mapper.
DISCARDCLASS	NUMBER(5)	—	Discard class field of the mapper.
MPLSORCOS	NUMBER(5)	—	MPLS or CoS field of the mapper.

1.1.99 MPLS_QoS_Pmap_Table

The MPLS_QoS_Pmap_Table describes the QoS policy map of the CPT QoS Editor.

Primary key: PMAP_PK

Table 1-97: MPLS_QoS_Pmap_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	Primary key.
NAME	VARCHAR(255)	—	Policy map name.

1.1.100 MPLS_QoS_Policy_Table

The MPLS_QoS_Policy_Table describes the QoS policy maps association between class map, action, and child policy.

Primary key: POLICY_PK

Table 1-98 : MPLS_QoS_Policy_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	Primary key.
CLASSMAP	NUMBER(19)	—	ID of the class map.
ACTION	NUMBER(19)	—	ID of the action.
CHILDPOLICY	NUMBER(19)	—	ID of the child policy.
POLICYMAPID	NUMBER(19)	—	ID of the policy map to which this policy is associated.

1.1.101 MPLS_QoS_Tablemap_Table

The MPLS_QoS_Tablemap_Table describes the QoS table maps of the CPT QoS Editor.

Primary key: TABLEMAP_PK

Table 1-99: MPLS_QoS_Tablemap_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ID	NUMBER(19)	NOT NULL	Primary key.
NAME	VARCHAR(255)	NOT NULL	Name of the table map.
DEFAULTCOPY	NUMBER(1)	—	Default copy value.
DEFAULTVALUE	NUMBER(5)	—	Default value of the class map.

1.1.102 MPLS_QoS_Violate_PA_Table

The MPLS_QoS_Violate_PA_Table describes the violate actions for a QoS action. A QoS action can have zero or more violate actions.

Primary key: None

Table 1-100: MPLS_QoS_Violate_PA_Table

Column Name	Oracle Data Type	Constraints	Description
ID	NUMBER(19)	NOT NULL	QoS action to which a violate action is associated. Acts as foreign key.
NAMECODE	NUMBER(5)	—	Internal code that identifies the name of the violate action.
VALUE	FLOAT(126)	—	Value of the violate action.
UNIT	NUMBER(5)	—	Unit of measure for the value.

1.1.103 MPLS_Service_Table

The MPLS_Service_Table contains information about the cross-connect contributors for each service.

Primary key: MPLS_SERVICE_TABLE_PK

Table 1-101: MPLS_Service_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
SERVICETYPE	NUMBER(10)	NOT NULL	Type of service. Options are: Pseudowire (PW) Transport Profile (TP) Ethernet Virtual Circuit (EVC)
SERVICENAME	VARCHAR2(256)	—	Service name.
SERVICEDESCRIPTION	VARCHAR2(256)	—	Description of the service.
LASTTIME	NUMBER(20)	NOT NULL	Time stamp that indicates when the last operation was updated. This is an internal field.
ADMINSTATE	NUMBER(4)	NOT NULL	Administrative status of the cross-connect contributor.
OPERSTATE	NUMBER(4)	NOT NULL	Operational status of the cross-connect contributor.
MARKBAD	NUMBER(1)	NOT NULL	Consistency of the cross-connect contributor. This is an internal field. Values are: 0: Consistent 1: Inconsistent

1.1.104 MPLS_TP_EP

The MPLS_TP_EP describes endpoints on the TP service.

Primary key: MPLS_TP_EP_PK

Table 1-102: MPLS_TP_EP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.

Column Name	Oracle Data Type	Integrity Constraints	Description
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
CTCINDEX	NUMBER(10)	NOT NULL	Physical object ID managed by the NE. This is an internal field.
PHYLOC	NUMBER(20)	NOT NULL	Physical object ID managed by the Prime Optical system. This is an internal field.
BDFNAME	VARCHAR2(256)	—	Bidirectional Forwarding Detection (BFD) template name has been provisioned.
SRCTUNNELNUM	NUMBER(6)	NOT NULL	Source tunnel number has been provisioned.
DSTTUNNELNUM	NUMBER(6)	NOT NULL	Destination tunnel number has been provisioned.
CTMWORKINGUNIQID	NUMBER(10)	NOT NULL	Unique Prime Optical ID used to reference the LSP contributors for the working path. This is an internal field.
CTMPROTECTEDUNIQID	NUMBER(10)	NOT NULL	Unique Prime Optical ID used to reference the LSP contributors for the protected path. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
TXBANDWIDTH	NUMBER(20)	NOT NULL	Transmission bandwidth value has been provisioned.
RXBANDWIDTH	NUMBER(20)	NOT NULL	Receiver bandwidth value has been provisioned.
OPERSTATE	NUMBER(1)	NOT NULL	Operation status of the endpoint.
ADMINSTATE	NUMBER(1)	NOT NULL	Administrative status of the endpoint.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical ID used to identify the contributor in the service. This is an internal field.
ISPROTECTED	NUMBER(1)	NOT NULL	Endpoint protection. Values are: 0: Unprotected 1: Protected
SRCNODEIP	VARCHAR2(256)	—	IP address of the source node.
DSTNODEIP	VARCHAR2(256)	—	IP address of the destination node.
WLSPNUM	NUMBER(10)	NOT NULL	LSP number used for the working path.
PLSPNUM	NUMBER(10)	NOT NULL	LSP number used for the protected path.

Column Name	Oracle Data Type	Integrity Constraints	Description
DESCRIPTION	VARCHAR2(256)	—	Description of the service has been provisioned.
SLOTMODULETYPE	NUMBER(10)	NOT NULL	Slot module type. This is an internal field.
PORTMODULETYPE	NUMBER(10)	NOT NULL	Port module type. This is an internal field.

1.1.105 NE_Audit_Trail_Table

The NE_Audit_Trail_Table shows ONS 15327 and ONS 15454 audit trail information.

Primary key: NE_AUDIT_TRAIL_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

For medium, large, and high-end configurations, this table is partitioned.

Table 1-103: NE_Audit_Trail_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE table.
SequenceNumber	NUMBER(6)	NOT NULL	Sequence number of the entry. Possible values are 1 to 59,999.
Timestamp	DATE	NOT NULL	Time stamp in Greenwich Mean Time (GMT).
OperationDescription	VARCHAR2(256)	NOT NULL	Server combines information obtained from multiple sources to create the description.
OperationStatus	VARCHAR2(1)	—	Operational status. Values are: P: Passed F: Failed X: Aborted
NEUserId	VARCHAR2(64)	—	CTC user ID.

1.1.106 NE_INFO_TABLE

The NE_INFO_TABLE contains information about all NEs managed by Prime Optical.

Primary key: NE_INFO_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-104: NE_INFO_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	Unique sequence number generated by the database. NEDBAccessID is the link between the NE_Info_Table and other tables that contain NE information such as inventory, alarms, PM data, and so on.
NESysID	VARCHAR2(128)	NOT NULL	NE system ID.
NEIPAddr	NUMBER(10)	—	NE IP address. The assumption is that the NE IP address is unique across the domains. The Prime Optical GUI and server convert the IP address to a number before storing the address in the database.
NENSAPAddr	VARCHAR2(60)	UNIQUE	NE OSI network service access point (NSAP) address.
NEState	NUMBER(1)	NOT NULL	State of the NE. Values are: 1: In service 2: Out of service 3: Preprovisioned 4: Under maintenance
isNEConnected	NUMBER(1)	NOT NULL	Whether the NE is reachable or unreachable from the Prime Optical server. Values are: 0: Unreachable 1: Reachable 2: Initialization Failed 3: Network Only Reachable
GNEID	NUMBER(8)	—	Index of the GNE in the GNE_Table that is used to route messages to the NE.
NEDescription	VARCHAR2(256)	—	Description of the NE.
NEConfigMode	NUMBER(2)	—	Configuration mode of the NE. Values are: –1: Undefined 0: Back-to-Back Terminal 1: BLSR add/drop multiplexer (ADM) 2: Hybrid 3: Line Extender Module 4: MS-SPRing ADM 5: Optical ADM 6: Optical Line Amplifier 7: Subnetwork connection protection

Column Name	Oracle Data Type	Integrity Constraints	Description
			(SNCP) ring ADM 8: Terminal 9: UPSR ADM 10: Other 11: Unknown 12: East Terminal 13: West Terminal 14: Not applicable 15: FULL-BLUE-OADM 16: FULL-RED-OADM 17: FULL-RED-BLUE-OADM 18: Extended Long Haul (ELH) West Terminal 19: ELH East Terminal 20: ELH Optical Line Amplifier 21: ELH Line Extender Module 22: ELH Optical ADM 23: ELH Terminal 25: ONS 15216 200-GHz OADM1 26: ONS 15216 200-GHz OADM2 27: ONS 15216 100-GHz OADM1 28: ONS 15216 100-GHz OADM2 29: ONS 15216 100-GHz OADM4 30: ONS 15216 EDFA1 31: ONS 15216 EDFA2 32: ONS 15216 Red 200-GHz DWDM Filter 33: ONS 15216 Blue 200-GHz DWDM Filter 34: ONS 15216 Red 100-GHz DWDM Filter 35: ONS 15216 Blue 100-GHz DWDM Filter 36: DCS 37: Linear ADM 38: Regeneration opt ADM 39: ELH Regeneration opt ADM 40: ONS 15216 OSC 1510 nm 41: ONS 15216 DCU 44: ONS 15216 UNIDIRFILTER 45: Unmanaged NE 46: Line 47: Hub 48: ONS 15216 EDFA3 49: Terminal Hub East 50: Terminal Hub West 51: Terminal OADM East 52: Terminal OADM West 53: Active TDM Hybrid Site

Column Name	Oracle Data Type	Integrity Constraints	Description
			54: Active Line Site 55: Passive Line Site 56: DWDM Hub Node 57: Amplified OADM 58: Passive OADM 59: Line Amplifier 60: DWDM Terminal 61: OSC Regeneration Site 62: hfr 63: hsr 64: miniq 65: gsriox 66: FULLQ 67: HALFQ 68: CAT6509 69: ROADM 70: Postamplifier 71: Preamplifier 72: MDS9100 73: MDS9200 74: MDS9500 75: MDSPAA 76: MDSSNM 77: CRS1MC 78: SNCP 79: GSR 12006 80: GSR 12008 81: GSR 12010 82: GSR 12012 83: GSR 12016 84: GSR 12404 85: GSR 12406 86: GSR 12410 87: GSR 12416 88: GSR 12810 89: GSR 12816 90: CISCO 7600 91: L_Line Site 92: L_ROADM 93: L_Terminal 94: L_Hub
NESnmpCommString	VARCHAR2(256)	—	SNMP community string of the NE.
NECLLICode	VARCHAR2(64)	—	CLLI code of the NE.

Column Name	Oracle Data Type	Integrity Constraints	Description
NEStateChangeTime	DATE	—	Time stamp of last state change of the NE.
NEModelType	NUMBER(4)	DEFAULT 1	Type of NE.
NEModelIndex	NUMBER(4)	DEFAULT 0	Index of the appropriate database record for the version. The value 0 represents an unknown model index.
NEModelStatus	NUMBER(4)	DEFAULT 0	Operational state of the NE model index. This is the server status view of the index. The NEModelStatus variable should not be set by the client. Values are: 0: Unknown model (probably being fetched) 1: Valid model 2: Mismatched model (NE type does not match) 3: Unknown model type 4: Unknown model software version
NESubNetworkID	NUMBER(8)	DEFAULT 0	Reference to the subnetwork in which this NE belongs.
NETimeOffset	NUMBER(10)	DEFAULT 0	Time offset between UNIX and the NE.
NELEDStatus	RAW(8)	DEFAULT '0000000000000000'	Reflects all LEDs on the NE front panel.
NESystemTitle	VARCHAR2(64)	DEFAULT NULL	Title for this NE.
NEUserLabel	VARCHAR2(64)	DEFAULT NULL	Name of the user of this NE.
NEVendorName	VARCHAR2(32)	DEFAULT NULL	NE vendor name.
NEVersion	VARCHAR2(128)	DEFAULT NULL	Version string for the NE.
NENumActiveCriticalErrors	NUMBER(8)	DEFAULT 0	Current number of active critical errors for the NE.
NENumActiveMajorErrors	NUMBER(8)	DEFAULT 0	Current number of active major errors for the NE.
NENumActiveMinorErrors	NUMBER(8)	DEFAULT 0	Current number of active minor errors for the NE.

Column Name	Oracle Data Type	Integrity Constraints	Description
NEPmEnabled	NUMBER(1)	DEFAULT 1	Whether PM data collection is enabled or disabled for PM data other than robust PM. Values are: 0: PM collection is disabled 1: PM collection is enabled 2: 15-minute PM collection is enabled 3: 1-day PM collection is enabled 4: 10-minute PM collection is enabled 5: 5-minute PM collection is enabled
NENodeID	NUMBER(10)	—	Used to specify an ONS 15327 or ONS 15454 NE from other NEs.
NEInventoryLastChanged	NUMBER(20)	—	Used to specify the last time an ONS 15327 or ONS 15454 NE changed its inventory.
NESubnetMask	NUMBER(10)	—	Subnet mask.
NEDefaultGateway	NUMBER(10)	—	Default gateway.
NENumUnActiveCriticalErrors	NUMBER(8)	DEFAULT 0	Current number of inactive critical errors for the NE.
NENumUnActiveMajorErrors	NUMBER(8)	DEFAULT 0	Current number of inactive major errors for the NE.
NENumUnActiveMinorErrors	NUMBER(8)	DEFAULT 0	Current number of inactive minor errors for the NE.
NENumCleared	NUMBER(8)	DEFAULT 0	Current number of cleared errors for the NE.
DisplayModelName	VARCHAR2(15)	—	Display string for the NE.
GWTL1USERNAME	VARCHAR2(20)	DEFAULT NULL	Default TL1 username.
GWTL1PASSWD	VARCHAR2(20)	DEFAULT NULL	Default TL1 password.
ServiceID	NUMBER(20)	DEFAULT 0	Which instances of NE and PM service are currently managing this NE.
DeleteState	NUMBER(1)	DEFAULT 0	State of deletion. Values are: 0: Working 1: Deleted 2: Purge in progress 3: Purge failed

Column Name	Oracle Data Type	Integrity Constraints	Description
ConfigSyncStatus	NUMBER(1)	DEFAULT 0	Configuration resynchronization status. Values are: 0: Initial 1: Queued 2: In Progress 3: Successful
NEUSERNAME	VARCHAR2(64)	DEFAULT NULL	Default NE username.
NEUSERPASSWORD	VARCHAR2(64)	DEFAULT NULL	Default NE password.
NEDiscoveryState	NUMBER(1)	NOT NULL	State of the NE under discovery. Values are: 1: NE undiscovered 2: NE initializing 3: NE synchronization configuration 4: NE discovered
NERobustPmEnabled	NUMBER(1)	DEFAULT 0	Whether robust PM data collection is enabled or disabled. Values are: 0: Robust PM data collection is disabled. 1: Robust PM data collection is enabled. 2: 15-minute Robust PM data collection is enabled. 3: 1-day Robust PM data collection is enabled.
SecurityTicket	VARCHAR2(32)	DEFAULT NULL	Default ticket.
RobustPMEnableTime	DATE	—	Time stamp of 15-minute Robust PM data collection is enabled.
DebugLevel	NUMBER(1)	DEFAULT –1	Debug level for the NE. Values are: –1: Service 0: Fatal 1: Major 2: Minor 3: Info 4: Debug 5: Trace
Robust1DayPMEnableTime	DATE	—	Time stamp of 1-day Robust PM data collection is enabled.
SecondsBehindUTC	NUMBER(20)	DEFAULT 0	Offset—Number of seconds behind Coordinated Universal Time (UTC).
SecondsDstBehindUTC	NUMBER(20)	DEFAULT 0	Offset—Number of seconds Daylight Saving Time (DST) is behind UTC.
NESNGRPID	NUMBER(8)	DEFAULT 0	A reference to the group of connected subnetworks to which this NE belongs.

Column Name	Oracle Data Type	Integrity Constraints	Description
NEAliasID	VARCHAR2(128)	NOT NULL	NE alias ID.
IsMultiShelf	NUMBER(1)	DEFAULT 0	Values are: 0: Single-shelf 1: Multishelf
RoleType	VARCHAR2(30)	—	NE role type.
NEVirtualIpAddr	NUMBER(10)	DEFAULT 0	Values are: NE Virtual IP Addr 0: Unique IP address non-zero
IsTunnelRequired	NUMBER(1)	DEFAULT 0	Values are: 0: Tunnel is not required 1: Tunnel is required
NEIPV6Addr	RAW(16)	—	NE IP address in IPv6 format.
IS_IPV6	NUMBER(1)	—	Flag indicating the IP version. Values are: 0: IPv4 1: IPv6
PREPROVISIONED_IP	NUMBER(1)	—	Flag to indicate the preprovisioned type. Values are: 4: IPv4 6: IPv6
IsDSS	NUMBER(1)	DEFAULT 0	Indicates whether the selected CTC-based SONET or CTC-based SDH NE is a designated SOCKS server (DSS). If so, the NE manages connectivity among other NEs through firewalls. Values are: 0: False 1: True
NE_SNMP_AUTH_PROT	VARCHAR2(10)	DEFAULT NULL	The SNMP authentication protocol.
NE_SNMP_AUTH_PWD	VARCHAR2(32)	DEFAULT NULL	The SNMP authentication password.
NE_SNMP_EID	VARCHAR2(64)	DEFAULT NULL	The SNMP engine identification.
NE_SNMP_PRIV_PWD	VARCHAR2(32)	DEFAULT NULL	The SNMP privacy password.
NE_SNMP_UNAME	VARCHAR2(20)	DEFAULT NULL	Default NE username.

Column Name	Oracle Data Type	Integrity Constraints	Description
NE_SNMP_VERSION	NUMBER(1)	DEFAULT –1	Default NE SNMP protection version. Values are: –1: Unknown 0: v1 1: v2c 2: v2 3: v3

1.1.107 NE_Inventory_Table

The NE_Inventory_Table contains inventory information about each NE managed by Prime Optical.

Primary key: NE_INVENTORY_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-105: NE_Inventory_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE Table.
ModuleIndex	NUMBER(4)	NOT NULL	Unique integer that identifies a subcomponent in a chassis; same as ifStaticIndex in the Interface_Table.
ModuleName	VARCHAR2(32)	—	Name of the module on the NE whose inventory information is stored in this row.
ModulePartNum	VARCHAR2(32)	—	Module part number.
ModuleCLEICode	VARCHAR2(32)	—	Module CLEI code.
ModuleHWVersion	VARCHAR2(32)	—	Module hardware version.
ModuleSWVersion	VARCHAR2(32)	—	Module software version.
ModuleSerialNumber	VARCHAR2(32)	—	Module serial number.
ModuleDATEOfManuf	DATE	—	Module date of manufacture.
ModuleDateOfEntry	DATE	—	Time stamp when this entry was made.

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleHoursInUse	NUMBER(10)	—	Number of hours the module was in use when the entry was made. Current hours in use = (Current Time – ModuleDateOfEntry) + ModuleHoursInUse.

1.1.108 NE_Model_State

The NE_Model_State table is not in use in Prime Optical R9.6.

1.1.109 NE_Model_Type

The NE_Model_Type table is not in use in Prime Optical R9.6.

1.1.110 NE_Port_Status_Table

The NE_Port_Status_Table contains information about the tagged status of all ports on an NE.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-106: NE_Port_Status_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NEDBAccessID of the NE for which the port status information is stored; same as NEDBAccessID in the NE_Info_Table.
PortIndex	NUMBER(4)	NOT NULL	Static index of the port.
PortOperState	NUMBER(2)	NOT NULL	State of the port. Values are: 1: Preprofessional 2: Under maintenance 3: In service 4: Out of service

1.1.111 NE_SNMPv3_Groups_Table

The NE_SNMPv3_Groups_Table contains information about SNMPv3 groups including name, security level, and read, write, and notification access settings.

Primary key: NE_SNMPv3_GROUPS_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-107: NE_SNMPv3_Groups_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Group_Name	VARCHAR2(32)	—	The group name. The CTC maximum is 32.
Security_Level	VARCHAR2(30)	—	The security level of the group. Values are: AuthPriv AuthNoPriv NoAuthNoPriv
Read_View_Access	VARCHAR2(32)	—	Displays the group's view-for-read access name.
Write_View_Access	VARCHAR2(32)	—	Displays the group's view-for-write access name.
Notify_View_Access	VARCHAR2(32)	—	Displays the group's view-for-notification access name.

1.1.112 NE_SNMPv3_NFILTER_TABLE

The NE_SNMPV3_NFILTER_TABLE displays information on the notification filters provisioned on a selected NE.

Primary key: NE_SNMPv3_NFILTER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-108: NE_SNMPv3_NFILTER_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Filter_Profile_Name	VARCHAR2(32)	—	Filter profile name that identifies the table entry.
SubTreeOID	VARCHAR2(200)	—	Object identifier that designates a subtree element in the MIB hierarchy.
Bit_Mask	VARCHAR2(32)	—	Bit mask that identifies the objects in the subtree.

Column Name	Oracle Data Type	Integrity Constraints	Description
Filter_Type	VARCHAR2(10)	—	Defines the filter type with respect to the subset of the MIB object that is identified by the subtree OID and bitmask fields. Values are: Included Excluded

1.1.113 NE_SNMPv3_PROXY_FWD_TABLE

The NE_SNMPV3_PROXY_FWD_TABLE displays information on the proxy forwarding entries. The proxy forwarding entries enable SNMPv3 proxying by selected remote NE proxies for GET and SET operations.

Primary key: NE_SNMPv3_PROXY_FWD_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-109: NE_SNMPv3_PROXY_FWD_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Row_ID	VARCHAR2(10)	—	Row ID.
Context_Engine_ID	VARCHAR2(32)	—	Engine ID of the remote NE.
Target_IP	VARCHAR2(50)	—	Target IP address of the SNMPv3 NE.
Local_User	VARCHAR2(10)	—	Local SNMPv3 username.
Remote_User	VARCHAR2(32)		Remote SNMPv3 username.
Proxy_Type	VARCHAR2(5)		Identifies the proxy action. Values are: Read Write
Security_Level	VARCHAR2(30)		Security level that the SNMPv3 NE uses. Values are: AuthPriv AuthNoPriv NoAuthNoPriv

1.1.114 NE_SNMPv3_PROXY_TRAP_FWD_TABLE

The NE_SNMPv3_PROXY_TRAP_FWD_TABLE displays information on trap forwarder entries. The trap forwarder entries enable SNMPv3 proxying by selected remote NE proxies for the TRAP operation.

Primary key: NE_SNMPv3_PROXY_FTABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-110: NE_SNMPv3_PROXY_TRAP_FWD_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Row_ID	VARCHAR2(40)	—	The row ID.
Context_Engine_ID	VARCHAR2(32)	—	Engine ID of the remote NE.
Tag	VARCHAR2(32)	—	A string that identifies one or more rows in the SNMPv3 NE Trap Destinations table. The Tag value references the Proxy Tags column in the SNMPv3 NE Trap Destinations table.
Incoming_User	VARCHAR2(10)	—	Name of the remote NE user.
Security_Level	VARCHAR2(12)	—	Security level that the SNMPv3 NE uses. Values are: AuthPriv AuthNoPriv NoAuthNoPriv

1.1.115 NE_SNMPv3_Users_Table

The NE_SNMPv3_Users_Table displays information about the SNMPv3 users defined on a selected NE. Local users are the ones for which the engine ID is the same as the one on the given NE.

Primary key: NE_SNMPv3_USERS_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-111 : NE_SNMPv3_Users_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
User_Name	VARCHAR2(32)	—	The username. The CTC maximum is 32.

Column Name	Oracle Data Type	Integrity Constraints	Description
Group_Name	VARCHAR2(32)	—	The group name. The CTC maximum is 32.
Authentication_Protocol	VARCHAR2(6)	—	The authentication protocol (MD5, SH1).
Privacy_Protocol	VARCHAR2(6)	—	The privacy protocol (DES).
Engine_Id	VARCHAR2(40)	—	The engine identification.
Engine_Id_Len	VARCHAR2(4)	—	The engine identification length.
Boots	VARCHAR2(50)	—	—

1.1.116 NE_SNMPv3_Views_Table

The NE_SNMPv3_Views_Table displays information on the views defined on a selected NE. A view identifies the subset of an object in the MIB.

Primary key: NE_SNMPv3_VIEWS_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-112: NE_SNMPv3_Views_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
View_Name	VARCHAR2(32)	—	Name of the MIB view.
SubTreeOID	VARCHAR2(200)	—	Object identifier that designates a subtree element in the MIB hierarchy.
Mask	VARCHAR2(32)	—	Bit mask that identifies the objects in the subtree.
Type	VARCHAR2(10)	—	Specifies the status of the view. Values are: Included Excluded

1.1.117 NE_SNMPv3_TRAP_DEST_TABLE

The NE_SNMPv3_TRAP_DEST_TABLE displays information on the trap destinations provisioned on a selected NE.

Primary key: NE_SNMPv3_TRAP_DEST_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-113: NE_SNMPv3_TRAP_DEST_TABLE

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Target_Address	VARCHAR2(50)	—	IP address of the EMS to which the NE forwards the SNMPv3 traps.
UDP_Port	VARCHAR2(10)	—	UDP port number to which the NE forwards the SNMPv3 traps.
User_Name	VARCHAR2(32)	—	The SNMPv3 username. The CTC maximum is 32.
Security_Level	VARCHAR2(30)	—	Security level that the SNMPv3 NE uses. Valid values are AuthPriv, AuthNoPriv, and NoAuthNoPriv.
Filter_Profile_Name	VARCHAR2(32)	—	Filter profile name associated with the trap destination. The filter name must be present in the SNMPv3 NE Notification Filters table.
Proxy_Traps_Only	VARCHAR2(6)	—	If true, the NE that acts as the SNMPv3 proxy uses the entries in the SNMPv3 NE Trap Destinations table to forward traps from other NEs to the target address. The SNMPv3 proxy NE does not send its own traps to the target address.
Proxy_Tag	VARCHAR2(32)	—	A string used to identify a set of entries in this table. This field is required for SNMPv3 proxy configuration and referenced by the SNMPv3 NE Proxy Trap Forwarder table.

1.1.118 NE_Swimage_Table

The NE_Swimage_Table stores information about the active and standby software images on the NE.

Primary key: NE_SWIMAGE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-114: NE_Swimage_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	—	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
ACTIVEIMAGENAME	VARCHAR2(64)	—	Active image name.
STANDBYIMAGENAME	VARCHAR2(64)	—	Standby image name.
DEFAULTIMAGENAME	VARCHAR2(64)	—	Default image name.
INPROGRESS	NUMBER(1)	DEFAULT 0	Whether there is an activate or revert procedure in progress. Values are: 0: False 1: True
PHYSICALLOC	NUMBER(20)	DEFAULT –1	Physical location.
PARTIALUPGRADE	NUMBER(1)	DEFAULT –1	Values are: –1: Not applicable 0: False 1: True
TYPE	VARCHAR2(256)	DEFAULT 'N/A'	—

1.1.119 Network_Partition_Table

The Network_Partition_Table stores network partition information.

Primary key: NETWORK_PARTITION_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-115: Network_Partition_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NPId	NUMBER(10)	—	Network partition ID.
NPName	VARCHAR2(128)	UNIQUE, NOT NULL	String name of the network partition.
NPDescription	VARCHAR2(256)	—	Description of the network partition.

1.1.120 NGXP_Bet_Gts_Table

The NGXP_Bet_Gts_Table is used by the Prime Optical service to synchronize the operation among the Prime Optical system and the NGXP network elements to keep the

NGXP services provisioned in alignment. The table stores a global time stamp indicating the changes that are made on the node.

Primary key: NGXP_BET_GTS_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-116: NGXP_Bet_Gts_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the NE in the NE_INFO_TABLE. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	CTC index related to the packet transport switch.
GTS	NUMBER(20)	NOT NULL	Global time stamp related to the last change on the NE.

1.1.121 NGXP_Bet_Table

The NGXP_Bet_Table is used by the Prime Optical service to synchronize the operation among the Prime Optical system and the NGXP network elements to keep the NGXP services provisioned in alignment. The table stores more detailed information about the changes involved in the NGXP services.

Primary key: NGXP_BET_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-117 : NGXP_Bet_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	Prime Optical index used to identify the NE in the NE_INFO_TABLE. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	CTC index related to the packet transport switch.
RANGEID	NUMBER(4)	NOT NULL	Range of the service identifiers that are changed.
CTS	NUMBER(20)	NOT NULL	Change time stamp value related to the range identifier.
DTS	NUMBER(20)	NOT NULL	Delete time stamp value related to the range identifier.

1.1.122 Object_Relationship_Table

The Object_Relationship_Table contains information about Prime Optical object relationships.

Primary key: OBJECT_RELATIONSHIP_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-118 : Object_Relationship_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
MODELTYPE	NUMBER(4)	NOT NULL	Model type.
RELATIONSHIPTYPE	NUMBER(1)	NOT NULL	Relationship type. Values are: 0: Self 1: Association 2: Parent
SELFOBJECTINDEX	NUMBER(10)	NOT NULL	—
RELATEDOBJECTINDEX	NUMBER(10)	NOT NULL	—

1.1.123 ObjectAttributeTable

The ObjectAttributeTable stores metadata attribute information about all attributes of objects in the ObjectTypeTable.

Primary key: OBJECTATTRIBUTETABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-119: ObjectAttributeTable

Column Name	Oracle Data Type	Integrity Constraints	Description
METADATAINDEX	NUMBER(5)	NOT NULL	Metadata index of the object.
ATTRINDEX	NUMBER(5)	—	Index of the attribute.
PARENTINDEX	NUMBER(5)	—	Index of the parent attribute.
COLID	NUMBER(4)	NOT NULL	Column ID used to store the attribute value in the EQPT_Info_Table.
NAME	VARCHAR2(64)	—	Name of the attribute.
DISPLAYTEXT	VARCHAR2(256)	—	Display name of the attribute on Prime Optical UI client.

Column Name	Oracle Data Type	Integrity Constraints	Description
ATTRCLASS	NUMBER(4)	—	Attribute class. Values are: 1: Equipment 2: Analog input/output 3: Digital input/output 4: Threshold 5: Severity 6: Action 7: WRKP 8: Real time
ATTRTYPE	NUMBER(4)	—	Attribute type. Values are: 1: Integer 2: Float 3: ENUM 4: Boolean 5: String 6: Exponential 7: Float string 10: Group SONET PM Threshold 11: Group DWDM PM Threshold
EXPBASE	NUMBER(10)	—	Base of the exponential for exponential attributes.
EXPONENT	NUMBER(10)	—	Exponent of the exponential for exponential attributes.
FACTORYDEFAULT	VARCHAR2(32)	—	Factory default value.
CUSTOMERDEFAULT	VARCHAR2(32)	—	Customer default value.
MINVAL	VARCHAR2(32)	—	Minimum allowable value for the attribute.
MAXVAL	VARCHAR2(32)	—	Maximum allowable value for the attribute.
ENUMVAL	VARCHAR2(1024)	—	ENUM values for the attribute.
INCRVAL	NUMBER(7,3)	—	Increments allowed for the attribute.
UNIT	VARCHAR2(60)	—	Unit of the attribute.
ATTRACCESS	NUMBER(3)	—	Type of access. Values are: 1: Read-write 2: Read-only 3: Read-create
MANDATORY	NUMBER(1)	—	Whether the attribute is mandatory. Values are: 0: Not mandatory

Column Name	Oracle Data Type	Integrity Constraints	Description
			1: Mandatory
ATTRGROUP	NUMBER(2)	—	Type of group. Values are: 1: Temperature 2: Current 3: Power 4: Input power 5: Output power 6: Analog 7: Digital 8: Info 9: Action 10: Current value 11: Extra pump (EP) power 12: Analog battery management module (BAT) 13: Analog data communications channel (DCC) 14: Analog fan 15: Digital in 16: Digital out 17: Config 18: Optical power 19: Performance 20: QoS 21: Security 22: Insertion 23: Attenuation 24: Pump power 25: Collision 26: Widget action
MAXLEN	NUMBER(10)	—	Maximum length of attribute.

1.1.124 ObjectTypeTable

The ObjectTypeTable stores metadata information about the various object types that Prime Optical manages.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-120: ObjectTypeTable

Column Name	Oracle Data Type	Integrity Constraints	Description
MODELTYPE	NUMBER(4)	NOT NULL	Model type.
OBJECTINDEX	NUMBER(10)	NOT NULL	Metadata object index.
METADATAINDEX	NUMBER(5)	—	Index in the ObjectAttributeTable.
OBJSHORTNAME	VARCHAR2(32)	NOT NULL	Short name for the object.
OBJLONGNAME	VARCHAR2(128)	—	Long name for the object.
PARENTINDEX	NUMBER(5)	—	Index of the parent object.
BOARDNAME	VARCHAR2(32)	—	Name of the object (module).

1.1.125 ONS15216_Active_User_Table

Primary key: ONS15216_ACTIVE_USER_TABLE_PK

Tablespace: ONS152xx_DATA_TABLESPACE

Table 1-121: ONS15216_Active_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
UserID	VARCHAR2(64)	NOT NULL	ONS 15216 username.

1.1.126 ONS15216_NE_Inventory_Table

The ONS15216_NE_Inventory_Table stores ONS 15216 inventory information.

Primary key: ONS15216_NE_INVENTORY_PK

Tablespace: ONS152XX_DATA_TABLESPACE

Table 1-122: ONS15216_NE_Inventory_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
Cleicode	VARCHAR2(32)	—	CLEI code for the card.
PartNumber	VARCHAR2(32)	—	Part number of the module.
SerialNumber	VARCHAR2(32)	—	Serial number of the module.

Column Name	Oracle Data Type	Integrity Constraints	Description
FWVersion	VARCHAR2(32)	—	Firmware version.
FWUpdateDate	VARCHAR2(32)	—	Date of firmware update.
WaveLength	VARCHAR2(64)	—	Wavelength.
ModuleNumber	NUMBER(20)	—	Module number.
MacAddress	VARCHAR2(17)	—	MAC address of the module.

1.1.127 ONS15216_Privilege_Table

The ONS15216_Privilege_Table contains information on privilege levels allowed for the ONS 15216 NE.

Primary key: PK_ONS15216_PRIV_ID

Tablespace: ONS152xx_DATA_TABLESPACE

Table 1-123 : ONS15216_Privilege_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
PrivilegeID	NUMBER(10)	NOT NULL	ID to access this table.
Privilege	VARCHAR2(30)	NOT NULL	Privilege level.
Description	VARCHAR(64)	—	Description.

1.1.128 ONS15216_User_Table

The ONS15216_User_Table contains information about ONS 15216 users.

Primary key: ONS15216_USER_TABLE_PK

Tablespace: ONS152xx_DATA_TABLESPACE

Table 1-124: ONS15216_User Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access an NE in this table.
UserID	VARCHAR2(64)	NOT NULL	ONS 15216 username.
Password	VARCHAR2(64)	—	ONS 15216 password.
PrivilegeID	NUMBER(10)	—	Privilege level.
LastLoginTime	DATE	—	Time stamp of the last login.
Timeout	NUMBER(3)	DEFAULT 0	Change password on next login status.

1.1.129 ONS1530x_HO_PM_Table

The ONS1530x_HO_PM_Table contains information about the ONS 1530x high-order PM parameters.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS1530x_HO_PM_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-125: ONS1530x_HO_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
ModuleType	NUMBER(8)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location.
NEIFINDEX	NUMBER(20)	NOT NULL	—
TimeStamp	DATE	NOT NULL	—
NEBBE	VARCHAR2(40)	—	Background block error.
NEES	VARCHAR2(40)	—	Errored seconds.
NESES	VARCHAR2(40)	—	Severely errored seconds.
NEUAS	VARCHAR2(40)	—	Unavailable seconds.
FEBBE	VARCHAR2(40)	—	Background block error (far end).
FEES	VARCHAR2(40)	—	Errored seconds (far end).
FESES	VARCHAR2(40)	—	Severely errored seconds (far end).
FEUAS	VARCHAR2(40)	—	Unavailable seconds (far end).
OBJECTTYPE	NUMBER(10)	—	Object type.
STROBJINSTANCE	VARCHAR2(256)	—	—
is24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

1.1.130 ONS1530x_IP_PM_Table

The ONS1530x_IP_PM_Table contains information about the ONS1530x IP PM table.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS1530x_IP_PM_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-126: ONS1530x_IP_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access the ONS1530x_NE_Inventory_Table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(8)	—	Module type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location of the card.
NEIFINDEX	NUMBER(20)	NOT NULL	—
TimeStamp	DATE	NOT NULL	Date and time when statistics were stored.
IFINUCASTPKTS	VARCHAR2(40)	—	—
IFINDISCARDS	VARCHAR2(40)	—	—
IFINERRORS	VARCHAR2(40)	—	—
IFINOCTETS	VARCHAR2(40)	—	—
IFOUTOCTETS	VARCHAR2(40)	—	—
IFINNUCASTPKTS	VARCHAR2(40)	—	—
IFINUNKNOWNPROTOS	VARCHAR2(40)	—	—
IFOUTUCASTPKTS	VARCHAR2(40)	—	—
IFOUTNUCASTPKTS	VARCHAR2(40)	—	—
IFOUTDISCARDS	VARCHAR2(40)	—	—
IFOUTERRORS	VARCHAR2(40)	—	—
IFINMULTICASTPKTS	VARCHAR2(40)	—	—
IFINBROADCASTPKTS	VARCHAR2(40)	—	—
IFOUTMULTICASTPKTS	VARCHAR2(40)	—	—
IFOUTBROADCASTPKTS	VARCHAR2(40)	—	—
IFTYPE	NUMBER(1)	NOT NULL	—
OBJECTTYPE	NUMBER(10)	NOT NULL	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
STROBJINSTANCE	VARCHAR2(256)	—	—
is24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

1.1.131 ONS1530x_LO_PM_Table

The ONS1530x_LO_PM_Table contains information about the ONS 1530x low-order PM parameters.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS1530x_LO_PM_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-127: ONS1530x_LO_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
ModuleType	NUMBER(8)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location.
NEIFINDEX	NUMBER(20)	NOT NULL	—
TimeStamp	DATE	NOT NULL	—
NEBBE	VARCHAR2(40)	—	Background block error.
NEES	VARCHAR2(40)	—	Errored seconds.
NESES	VARCHAR2(40)	—	Severely errored seconds.
NEUAS	VARCHAR2(40)	—	Unavailable seconds.
FEBBE	VARCHAR2(40)	—	Background block error (far end).
FEES	VARCHAR2(40)	—	Errored seconds (far end).
FESES	VARCHAR2(40)	—	Severely errored seconds (far end).
FEUAS	VARCHAR2(40)	—	Unavailable seconds (far end).
OBJECTTYPE	NUMBER(10)	—	Object type.
STROBJINSTANCE	VARCHAR2(256)	—	—

Column Name	Oracle Data Type	Integrity Constraints	Description
is24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

1.1.132 ONS1530x_MS_PM_Table

The ONS1530x_MS_PM_Table contains information about the ONS 1530x multiplex section PM parameters.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS1530x_MS_PM_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-128 : ONS1530x_MS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
ModuleType	NUMBER(8)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location.
NEIFINDEX	NUMBER(20)	NOT NULL	—
TimeStamp	DATE	NOT NULL	—
NEBBE	VARCHAR2(40)	—	Background block error.
NEES	VARCHAR2(40)	—	Errored seconds.
NESES	VARCHAR2(40)	—	Severely errored seconds.
NEUAS	VARCHAR2(40)	—	Unavailable seconds.
FEBBE	VARCHAR2(40)	—	Background block error (far end).
FEES	VARCHAR2(40)	—	Errored seconds (far end).
FESES	VARCHAR2(40)	—	Severely errored seconds (far end).
FEUAS	VARCHAR2(40)	—	Unavailable seconds (far end).
OBJECTTYPE	NUMBER(10)	—	Object type.
STROBJINSTANCE	VARCHAR2(256)	—	—
is24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

1.1.133 ONS1530x_NE_Inventory_Table

The ONS1530x_NE_Inventory_Table contains information about the ONS 1530x equipment inventory table.

Primary key: ONS1530x_NE_INVENTORY_TABLE_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-129: ONS1530x_NE_Inventory_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access the ONS1530x_NE_Inventory_Table; same as NEDBAccessID in the NE_Info_Table.
Type	NUMBER(10)	—	Equipment type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location of the card.
ModuleName	NUMBER(10)	—	Equipment name.
ServiceState	VARCHAR2(40)	—	State of the module. Values are: 0: In Service 1: Out of Service 2: Out of Service By Maintenance 3: Service N/A 4: Out of Service Pending
SoftwareVersion	VARCHAR2(16)	—	Software version of the module.
SerialNumber	VARCHAR2(16)	—	Serial number of the module.
HardwareVersion	VARCHAR2(16)	—	Hardware version of the module.
InstallState	VARCHAR2(40)	—	State of the slot. Values are: 0: Empty 1: Installed and Expected 2: Expected and Not Installed 3: Installed and Not Expected 4: Mismatch of Installed and Expected 5: Unavailable 6: Unknown
ExpectedModule	NUMBER(10)	—	Expected module in the slot.

1.1.134 ONS1530x_RS_PM_Table

The ONS1530x_RS_PM_Table contains information about the ONS 1530x regeneration section PM parameters.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS1530x_RS_PM_PK

Tablespace: ONS153xx_DATA_TABLESPACE

Table 1-130: ONS1530x_RS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
ModuleType	NUMBER(8)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location.
NEIFINDEX	NUMBER(20)	NOT NULL	—
TimeStamp	DATE	NOT NULL	—
NEBBE	VARCHAR2(40)	—	Background block error.
NEES	VARCHAR2(40)	—	Errored seconds.
NESES	VARCHAR2(40)	—	Severely errored seconds.
NEUAS	VARCHAR2(40)	—	Unavailable seconds.
OBJECTTYPE	NUMBER(10)	—	Object type.
STROBJINSTANCE	VARCHAR2(256)	—	—
is24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

1.1.135 ONS15454_327_NE_Table

The ONS15454_327_NE_Table shows audit configuration information about the ONS 15327 and ONS 15454 SONET.

Primary key: ONS15454_327_NE_Table_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-131: ONS15454_327_NE_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE table.
NEAuditTrailStatus	NUMBER(1)	DEFAULT 0	Whether the audit trail is enabled. Values are: 0: Disabled 1: Enabled
NEAuditTrailLastSequenceNumber	NUMBER(8)	DEFAULT 0	Maximum sequence number for which audit trail information has already been collected.
NELastAuditTrailCollectedTime	DATE	DEFAULT NULL	Last time the audit trail was collected.
NEDwdmConfigMode	NUMBER(2)	DEFAULT –1	Stores DWDM NE types. A list of possible DWDM NE types can also be found in the NEConfigMode column in the NE_INFO_TABLE.
Last15MinPMCollectedTime	DATE	DEFAULT NULL	The last time the 15-minute PM data was collected.
Last1DayPMCollectedTime	DATE	DEFAULT NULL	The last time the 1-day PM data was collected.

1.1.136 ONS15454_8B10B_PM_Table

The ONS15454_8B10B_PM table is used for fiber channel transport PM data accumulated on the FC_MR-4 card.

Primary key: ONS15454_8B10B_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-132: ONS15454_8B10B_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE table.
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates the status of the PM bucket. Values are: 0: Valid period Other: Bit mask value

Column Name	Oracle Data Type	Integrity Constraints	Description
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold values for the PM bucket for the 15-minute interval.
ThresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
validPackets	NUMBER(20)	DEFAULT –1	8b10b valid packets.
invalidPackets	NUMBER(20)	DEFAULT –1	8b10b invalid packets.
codeGroupViolations	NUMBER(20)	DEFAULT –1	8b10b code group violations.
idleOrderedSets	NUMBER(20)	DEFAULT –1	8b10b idle ordered sets.
nonIdleOrderedSets	NUMBER(20)	DEFAULT –1	8b10b nonidle ordered sets.
dataCodeGroups	NUMBER(20)	DEFAULT –1	8b10b data code groups.
rxTotalPackets	NUMBER(20)	DEFAULT –1	8b10b total packets received.
ifInErrors	NUMBER(20)	DEFAULT –1	8b10b errors in received packets.
statsEncodingDispErrors	NUMBER(20)	DEFAULT –1	8b10b encoding errors.
dataOrderedSets	NUMBER(20)	DEFAULT –1	8b10b data ordered sets.
dataPayLoad	VARCHAR2(1024)	—	8b10 data payload.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15 minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
invalidOrderedSets	NUMBER(20)	DEFAULT –1	8b10b invalid ordered sets.
InvldOrdrdSetDispErrorSum	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.137 ONS15454_Active_User_Table

The ONS15454_Active_User_Table contains information about the ONS 15454 SONET and ONS 15454 SDH Active Users Table.

Primary key: ONS15454_ACTIVE_USER_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-133 : ONS15454_Active_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE.
UserID	VARCHAR2(64)	NOT NULL	Cisco ONS 15454 username.
ClientIp	VARCHAR2(64)	NOT NULL	Client IP address.
SessionType	VARCHAR2(20)	DEFAULT NULL	Type of session.
LastLoginTime	DATE	DEFAULT NULL	Date and time of most recent user login.
LastActivityTime	DATE	DEFAULT NULL	Date and time of most recent user activity.
SessionTicket	NUMBER(10)	DEFAULT 0	Ticket for the session.

1.1.138 ONS15454_CBQoS_Info_Table

Primary key: ONS15454_CBQOS_INFO_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-134: ONS15454_CBQoS_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE table.
PhysicalLoc	NUMBER(10)	NOT NULL	Physical location of the card.
ModuleType	NUMBER(8)	—	Card type.
CbQosPolicyindex	NUMBER(10)	NOT NULL	—
CbQosObjectIndex	NUMBER(10)	NOT NULL	—
CbQosConfigIndex	NUMBER(10)	NOT NULL	—
CbQosObjectType	NUMBER(2)	—	Values are: 1: Policymap 2: Classmap 3: MatchStatement 4: Queueing 5: RandomDetect 6: TrafficShaping 7: Police 8: Set
CbQosParentindex	NUMBER(10)	NOT NULL	—
ObjectName	VARCHAR2(4000)	—	Name of the object.
DisplayString	VARCHAR2(4000)	—	Complete name of the object.
StatsEnabled	NUMBER(1)	DEFAULT 0	Values are: 0: False 1: True

1.1.139 ONS15454_CLASSMAP_PM_Table

The ONS15454_CLASSMAP_PM_Table is used for CBQOS PM data where the module type is an ML-series card.

Primary key: ONS15454_QOSCM_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-135: ONS15454_CLASSMAP_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE.

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit Mask value
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
CBQosCMPrePolicyPkt64	NUMBER(20)	DEFAULT 0	—
CBQosCMPrePolicyByte64	NUMBER(20)	DEFAULT 0	—
CBQosCMPostPolicyByte64	NUMBER(20)	DEFAULT 0	—
CBQosCMDropPkt64	NUMBER(20)	DEFAULT 0	—
CBQosCMDropByte64	NUMBER(20)	DEFAULT 0	—
CBQosCMNoBufDropPkt64	NUMBER(20)	DEFAULT 0	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.140 ONS15454_CMS_Storage_Table

The ONS15454_CMS_Storage_Table stores information about the CTC binary. The table can have multiple entries; each entry indicates a different CTC package that was manually uploaded by the user. At any one time, only one of the packages can be active.

The active package is used by the Prime Optical client to launch CTC. The Preferred-jar-path value in the CTM_Config_Table points to the entry in the ONS15454_CMS_Storage_Table that is active at that time.

Primary key: ONS15454_CMS_STORAGE_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-136: ONS15454_CMS_Storage_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
JarID	NUMBER(4)	NOT NULL	Running number that is incremented by one for each insertion into the table.
FileName	VARCHAR2(128)	NOT NULL	Used to store the system recommended JAR file version number.
JarTime	DATE	NOT NULL	Date of the JAR.
UserName	VARCHAR2(128)	NOT NULL	User profile when JAR file is active.

1.1.141 ONS15454_COS_PM_Table

The ONS15454_COS_PM_Table contains historical PM data collected in each interface and class of service level for the ML-series cards of CTC-based NEs.

Primary key: ONS15454_COS_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-137 : ONS15454_COS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE table.
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit mask value
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.

Column Name	Oracle Data Type	Integrity Constraints	Description
InterfaceName	VARCHAR2(40)	NOT NULL	—
InterfaceDirection	VARCHAR2(40)	NOT NULL	—
ClassOfServiceLevel	NUMBER(1)	NOT NULL	—
PostPolicyPackets	NUMBER(20)	DEFAULT –1	—
PostPolicyBytes	NUMBER(20)	DEFAULT –1	—
DropPackets	NUMBER(20)	DEFAULT –1	—
DropBytes	NUMBER(20)	DEFAULT –1	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15 minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.142 ONS15454_DS1_PM_Table

The ONS15454_DS1_PM_Table stores DS-1 PM information about ONS 15327 and ONS 15454 NEs. This table is used when the module type is a DS1, DS1N, or DS3XM.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_DS1_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-138: ONS15454_DS1_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM

Column Name	Oracle Data Type	Integrity Constraints	Description
			pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
DS1LineCodeViolations	NUMBER(10)	DEFAULT –1	Number of line code violations (LCVs) encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1LineErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1LineSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1RxPathAIS	NUMBER(10)	DEFAULT –1	Number of alarm indication signal seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.

Column Name	Oracle Data Type	Integrity Constraints	Description
DS1RxPathErrSecs	NUMBER(10)	DEFAULT – 1	Number of errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.
DS1RxPathSAS	NUMBER(10)	DEFAULT – 1	Number of severely alarmed seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.
DS1RxPathSevErrSecs	NUMBER(10)	DEFAULT – 1	Number of severely errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.
DS1RxPathUnavailableSecs	NUMBER(10)	DEFAULT – 1	Number of unavailable seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.
DS1RxPathCodeViolations	NUMBER(10)	DEFAULT – 1	Number of LCVs encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS3XM card and the DS1 line card.
DS1TxPathAIS	NUMBER(10)	DEFAULT – 1	Number of alarm indication signal seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1TxPathErrSecs	NUMBER(10)	DEFAULT – 1	Number of errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1TxPathSAS	NUMBER(10)	DEFAULT – 1	Number of severely alarmed seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.

Column Name	Oracle Data Type	Integrity Constraints	Description
DS1TxPathSevErrSecs	NUMBER(10)	DEFAULT – 1	Number of severely errored seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1TxPathUnavailableSecs	NUMBER(10)	DEFAULT – 1	Number of unavailable seconds encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1TxPathCodeViolations	NUMBER(10)	DEFAULT – 1	Number of line code violations encountered by a DS1 interface in the interval. This field is valid for the DS1 interface on the DS1 line card only.
DS1LineLOSS	NUMBER(10)	DEFAULT – 1	Loss of signal seconds line. Count of 1-second intervals containing one or more LOS defects.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeDS1Fd1PathCV	NUMBER(10)	DEFAULT – 1	Number of far-end path CV (ESF).
FeDS1Fd1PathES	NUMBER(10)	DEFAULT – 1	Number of far-end path ES (ESF).
FeDS1Fd1PathESA	NUMBER(10)	DEFAULT – 1	Number of far-end path ES-Type A(ESF).
FeDS1Fd1PathESB	NUMBER(10)	DEFAULT – 1	Number of far-end path ES-Type B (ESF).
FeDS1Fd1PathSES	NUMBER(10)	DEFAULT – 1	Number of far-end path SES (ESF).
FeDS1Fd1PathSEFS	NUMBER(10)	DEFAULT – 1	Number of far-end path SEFS (ESF).
FeDS1Fd1PathCSS	NUMBER(10)	DEFAULT – 1	Number of far-end path Controlled Slip Seconds (ESF).
FeDS1Fd1PathUAS	NUMBER(10)	DEFAULT – 1	Number of far-end path UAS (ESF).
FeDS1Fd1PathFC	NUMBER(10)	DEFAULT – 1	Number of far-end path FC (ESF).
FeDS1Fd1LineES	NUMBER(10)	DEFAULT – 1	Number of far-end line ES (ESF).

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
FeDS1FdlEsNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance errored seconds.
FeDS1FdlSesNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance severely errored seconds.
FeDS1FdlUasNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance unavailable seconds.
DS1EsNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance errored seconds.
DS1SesNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance severely errored seconds.
DS1UasNP	NUMBER(10)	DEFAULT –1	Number of far-end network performance unavailable seconds.
DS1RxPathBBE	NUMBER(10)	DEFAULT –1	Number of receive path background block errors.
DS1RxPathESR	NUMBER(10)	DEFAULT –1	Receive path errored seconds ratio.
DS1RxPathSESR	NUMBER(10)	DEFAULT –1	Receive path severely errored seconds ratio.
DS1RxPathBBER	NUMBER(10)	DEFAULT –1	Receive path background block error ratio.
DS1TxPathBBE	NUMBER(10)	DEFAULT –1	Number of transmit path background block errors.
DS1TxPathESR	NUMBER(10)	DEFAULT –1	Transmit path errored seconds ratio.
DS1TxPathSESR	NUMBER(10)	DEFAULT –1	Transmit path severely errored seconds ratio.
DS1TxPathBBER	NUMBER(10)	DEFAULT –1	Transmit path background block error ratio.
DS1RxPathFC	NUMBER(10)	DEFAULT –1	Receive path failure count.
FeDS1FdlPathESR	NUMBER(10)	DEFAULT –1	Far-end path errored seconds ratio.
FeDS1FdlPathSESR	NUMBER(10)	DEFAULT –1	Far-end path severely errored seconds ratio.

Column Name	Oracle Data Type	Integrity Constraints	Description
DS1TxPathFC	NUMBER(10)	DEFAULT – 1	Transmit path failure count.
StrCorbaTIDAID	VARCHAR(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.143 ONS15454_DS3_PM_Table

The ONS15454_DS3_PM_Table stores DS-3 PM information about ONS 15327 and ONS 15454 NEs. This table is used for DS3 PM data where the module type is a DS3, DS3E, DS3N, DS3NE, or DS3XM line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_DS3_LINE_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-139 : ONS15454_DS3_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.

Column Name	Oracle Data Type	Integrity Constraints	Description
TimeStamp	DATE	DEFAULT 0	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
DS3LineCodeViolations	NUMBER(10)	DEFAULT –1	Number of LCVs encountered by a DS3 interface in the interval. This is valid for the DS3XM and DS3 line cards.
DS3LineErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a DS3 interface in the interval. This is valid for the DS3XM and DS3 line cards.
DS3LineSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a DS3 interface in the interval. This is valid for the DS3XM and DS3 line cards.
DS3LineLOSS	NUMBER(10)	DEFAULT –1	Number of loss of signal seconds encountered by a DS3 interface in the interval.
DS3PBitAIS	NUMBER(10)	DEFAULT –1	Number of alarm indication signal seconds encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3PBitCodeViolations	NUMBER(10)	DEFAULT –1	Number of LCVs encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is

Column Name	Oracle Data Type	Integrity Constraints	Description
			available on the DS3XM card only.
DS3PBitErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3PBitSAS	NUMBER(10)	DEFAULT –1	Number of severely alarmed seconds encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3PBitSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3PBitUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a DS3 interface in the P-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3CpBitCodeViolations	NUMBER(10)	DEFAULT –1	Number of LCVs encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3CpBitErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3CpBitSAS	NUMBER(10)	DEFAULT –1	Number of severely alarmed seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.

Column Name	Oracle Data Type	Integrity Constraints	Description
DS3CpBitSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
DS3CpBitUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeDS3CpBitCodeViolations	NUMBER(10)	DEFAULT –1	Number of far end line code violations encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
FeDS3CpBitErrSecs	NUMBER(10)	DEFAULT –1	Number of far end errored seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
FeDS3CpBitSAS	NUMBER(10)	DEFAULT –1	Number of far end errored severely alarmed seconds encountered by a DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
FeDS3CpBitSevErrSecs	NUMBER(10)	DEFAULT –1	Number of far end severely errored seconds encountered by an DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.

Column Name	Oracle Data Type	Integrity Constraints	Description
FeDS3CpBitUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of far end unavailable seconds encountered by an DS3 interface in the CP-bit of the interface in the interval. This is a path variable and is available on the DS3XM card only.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for the PM data.
StrCorbaTIDAID	VARCHAR(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.144 ONS15454_ENET_PM_Table

The ONS15454_ENET_PM_Table contains Ethernet PM data for ONS 15327 and ONS 15454 SONET NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_ENET_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-140: ONS15454_ENET_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type (Ethernet line card) to which this PM data pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	DEFAULT 0	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
IfInOctets	NUMBER(20)	DEFAULT –1	Total number of octets of data (including those in bad packets) received on the network (excluding framing bits but including frame check sequence [FCS] octets).
IfInUcastPkts	NUMBER(20)	DEFAULT –1	Number of packets, delivered by this sublayer to a higher layer, which were not addressed to a multicast or broadcast address at this sublayer.
IfInMulticastPkts	NUMBER(20)	DEFAULT –1	Total number of good packets received that were directed to a multicast address. This number does not include packets directed to the broadcast address.
IfInBroadcastPkts	NUMBER(20)	DEFAULT –1	Total number of good packets received that were directed to the broadcast address. This number does not include multicast packets.
IfInErrors	NUMBER(20)	DEFAULT –1	Total number of errors received.
IfOutOctets	NUMBER(20)	DEFAULT –1	Total number of octets transmitted out of the interface, including framing characters.

Column Name	Oracle Data Type	Integrity Constraints	Description
IfOutUcastPkts	NUMBER(20)	DEFAULT –1	Total number of packets that higher-level protocols requested to be transmitted that were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent.
IfOutMulticastPkts	NUMBER(20)	DEFAULT –1	Number of good packets transmitted by this address that were directed to a multicast address. This number does not include packets directed to the broadcast address.
IfOutBroadcastPkts	NUMBER(20)	DEFAULT –1	Number of good packets transmitted by this address that were directed to the broadcast address.
Dot3StatsAlignmentErrors	NUMBER(20)	DEFAULT –1	Total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, but had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a nonintegral number of octets (alignment error).
Dot3StatsFCSErrors	NUMBER(20)	DEFAULT –1	Count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check.
Dot3StatsSingleCollisionFrames	NUMBER(20)	DEFAULT –1	Count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision.
Dot3StatsMultiCollisionFrames	NUMBER(20)	DEFAULT –1	Count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision.
Dot3StatsDeferredTransmissions	NUMBER(20)	DEFAULT –1	Count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions.
Dot3StatsLateCollisions	NUMBER(20)	DEFAULT –1	Number of times a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet. 512 bit-times corresponds to 51.2 microseconds on a 10-Mbps system. A late collision included in a count

Column Name	Oracle Data Type	Integrity Constraints	Description
			represented by an instance of this object is also considered a generic collision for purposes of other collision-related statistics.
Dot3StatsExcessiveCollisions	NUMBER(20)	DEFAULT –1	Count of frames for which transmission on a particular interface fails due to excessive collisions.
EtherStatsUndersizePkts	NUMBER(20)	DEFAULT –1	Total number of packets received that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed.
EtherStatsFragments	NUMBER(20)	DEFAULT –1	Total number of packets received that were less than 64 octets in length (excluding framing bits but including FCS octets) and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a nonmanagerial number of octets (alignment error). Note that it is normal for Ethernet fragments to increment, because the count includes runts (which are normal occurrences due to collisions) and noise hits.
EtherStatsOversizePkts	NUMBER(20)	DEFAULT –1	Total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.
EtherStatsJabbers	NUMBER(20)	DEFAULT –1	Total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a nonmanagerial number of octets (alignment error). Note that this definition of jabber is different than the definition in IEEE-802.3 section 8.2.1.5 (10BASE-5) and section 10.3.1.4 (10BASE-2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.
RxPauseFrames	NUMBER(20)	DEFAULT –1	Number of received Ethernet 802.3x pause frames.

Column Name	Oracle Data Type	Integrity Constraints	Description
TxPauseFrames	NUMBER(20)	DEFAULT –1	Number of transmitted 802.3x pause frames.
RxPktsDropInternalCongestion	NUMBER(20)	DEFAULT –1	Number of received packets dropped due to overflow in G1000-4 frame buffer.
TxPktsDropInternalCongestion	NUMBER(20)	DEFAULT –1	Number of transmit queue drops due to drops in the G1000-4 frame buffer.
RxPackets	NUMBER(20)	DEFAULT –1	Number of packets received since the last counter reset.
TxPackets	NUMBER(20)	DEFAULT –1	Number of packets transmitted since the last counter reset.
RxTotalErrors	NUMBER(20)	DEFAULT –1	Total number of receive errors.
RxRunts	NUMBER(20)	DEFAULT –1	Total number of frames received that are less than 64 bytes in length and have a cyclic redundancy check (CRC) error.
RxGiants	NUMBER(20)	DEFAULT –1	Number of packets received that are greater than 1548 bytes in length.
TxCollisions	NUMBER(20)	DEFAULT –1	Number of transmit packets that are collisions; the port and the attached device transmitting at the same time caused collisions.
RxGMACDropCounts	NUMBER(20)	DEFAULT –1	Gigabit Ethernet MAC drop counts.
RxThresholdOversizes	NUMBER(20)	DEFAULT –1	Number of packets received with a length longer than 1518 octets.
PortDropCounts	NUMBER(20)	DEFAULT –1	Drop counts per port.
TxGiants	NUMBER(20)	DEFAULT –1	Number of packets transmitted that are greater than 1548 bytes in length.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
EtherUtilizationStats	NUMBER(10)	DEFAULT –1	The line utilization number expresses the average of ingress and egress traffic as a percentage of the total capacity.
HDLCPktDrops	NUMBER(20)	DEFAULT –1	HDLC packet drop.
RxControlFrames	NUMBER(20)	DEFAULT –1	Receive control frames.
RxUnknownOpCodeFrames	NUMBER(20)	DEFAULT –1	Receive unknown opcode frames.

Column Name	Oracle Data Type	Integrity Constraints	Description
IfInErrorBytePkts	NUMBER(20)	DEFAULT –1	—
IfInFramingErrorPkts	NUMBER(20)	DEFAULT –1	—
IfInJunkInterPkts	NUMBER(20)	DEFAULT –1	—
IfInDiscards	NUMBER(20)	DEFAULT –1	—
IfOutDiscards	NUMBER(20)	DEFAULT –1	—
Dot3StatsFrameTooLong	NUMBER(20)	DEFAULT –1	—
Dot3StatsCarrierSenseErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsSOETestErrors	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts64Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts65to127Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts128to255Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts256to511Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts512to1023Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts1024to1518Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsBroadcastPkts	NUMBER(20)	DEFAULT –1	—
EtherStatsMulticastPkts	NUMBER(20)	DEFAULT –1	—
EtherStatsOctets	NUMBER(20)	DEFAULT –1	—
EtherStatsCollisions	NUMBER(20)	DEFAULT –1	—
EtherStatsCollisionFrames	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
EtherStatsCRCAlignErrors	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesTruncated	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesTooLong	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesBadCRC	NUMBER(20)	DEFAULT –1	—
MediaIndStatsTxFramesBadCRC	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxShortPkts	NUMBER(20)	DEFAULT –1	—
MediaIndStatsOversizeDropped	NUMBER(20)	DEFAULT –1	—
IfOutErrors	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts	NUMBER(20)	DEFAULT –1	—
Dot3StatsInternalMacTxErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsInternalMacRxErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsSymbolErrors	NUMBER(20)	DEFAULT –1	—
MediaIndStatsTxFramesTooLong	NUMBER(20)	DEFAULT –1	—
RxEtherUtilizationStats	NUMBER(10)	DEFAULT –1	—
TxEtherUtilizationStats	NUMBER(10)	DEFAULT –1	—
StatsLineLastClearedTime	DATE	—	Time stamp when Ethernet statistics were last cleared.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
Dot3StatsCtrInUnknownOpCodes	NUMBER(20)	DEFAULT –1	—
Dot3StatsInPauseFrames	NUMBER(20)	DEFAULT –1	—
Dot3StatsOutPauseFrames	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts1519to1522Octets	NUMBER(20)	DEFAULT –1	—
StrCorbaTIDAID	NUMBER(512)	DEFAULT –1	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.
EtherStatsTxFifoOverflowEvents	NUMBER(20)	DEFAULT –1	—
IFHCINOCTETS	NUMBER(20)	DEFAULT –1	—
IFHCINUCASTPKTS	NUMBER(20)	DEFAULT –1	—
IFHCINMULTICASTPKTS	NUMBER(20)	DEFAULT –1	—
IFHCINBROADCASTPKTS	NUMBER(20)	DEFAULT –1	—
IFHCOUTOCTETS	NUMBER(20)	DEFAULT –1	—
IFHCOUTMULTICASTPKTS	NUMBER(20)	DEFAULT –1	—
IFHCOUTBROADCASTPKTS	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
ETHERSTATSHIGHCAPACITYPKTS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHIGHCAPACITYOCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC64OCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC65TO127OCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC128TO255OCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC256TO511OCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC512TO1023OCTETS	NUMBER(20)	DEFAULT –1	—
ETHERSTATSHC1024TO1518OCTETS	NUMBER(20)	DEFAULT –1	—
CISRXREPORTS	NUMBER(20)	DEFAULT –1	—
CISRXLEAVES	NUMBER(20)	DEFAULT –1	—
CISTXREPORTS	NUMBER(20)	DEFAULT –1	—
CISTXLEAVES	NUMBER(20)	DEFAULT –1	—
CISTXGENERALQUERIES	NUMBER(20)	DEFAULT –1	—
CISTXGROUPSPECIFICQUERIES	NUMBER(20)	DEFAULT –1	—
CISRXGENERALQUERIES	NUMBER(20)	DEFAULT –1	—
CISRXGROUPSPECIFICQUERIES	NUMBER(20)	DEFAULT –1	—
CISRXVALIDPACKETS	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
CISRXINVALIDPACKETS	NUMBER(20)	DEFAULT –1	—
CREPHFLRXPDUS	NUMBER(20)	—	PM counter specific to Resilient Ethernet Protocol (REP); indicates the number of hardware flood layer Protocol Data Units (PDUs) received on the port.
CREPHFLTXPDUS	NUMBER(20)	—	PM counter specific to REP; indicates the number of hardware flood layer PDUs transmitted on the port.
CREPLSLRXPDUS	NUMBER(20)	—	PM counter specific to REP; indicates the number of link status layer PDUs received on the port.
CREPLSLTXPDUS	NUMBER(20)	—	PM counter specific to REP; indicates the number of link status layer PDUs transmitted on the port.
DOT3ADAGGPORTSTATSLACPDUSRX	NUMBER(20)	—	PM counter specific to aggregated ports; indicates the number of valid Link Aggregate Control Protocol Data Units (LACPDUs) received on the aggregation port.
DOT3ADAGGPORTSTATSLACPDUSTX	NUMBER(20)	—	PM counter specific to aggregated ports; indicates the number of LACPDUs transmitted on the aggregation port.

1.1.145 ONS15454_FC_PM_Table

The ONS15454_FC_PM_Table is used for fiber channel transport PM data accumulated on FCMR_4 cards.

Primary key: ONS15454_FC_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-141: ONS15454_FC_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.

Column Name	Oracle Data Type	Integrity Constraints	Description
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Fifteen-minute interval threshold value for the PM bucket.
thresholdList	VARCHAR2(1024)	—	Daily threshold value for the PM bucket.
fibreStatsInvalidOrderedSets	NUMBER(20)	DEFAULT –1	Number of invalid ordered sets received.
fibreStatsEncodingDispErrors	NUMBER(20)	DEFAULT –1	A count of the disparity errors received by the serial/deserializer.
fibreStatsRxTotalErrors	NUMBER(20)	DEFAULT –1	A count of 10b errors received by the serial/deserializer.
fibreStatsRxFramesTruncated	NUMBER(20)	DEFAULT –1	A count of the received fiber channel frames < 36 bytes including CRC, start of frame (SOF), and end of frame (EOF).
fibreStatsRxFramesTooLong	NUMBER(20)	DEFAULT –1	A count of the received fiber channel frames > 2116 bytes of the payload. Four bytes are allowed for supporting VSAN tags sent.
fibreStatsRxFramesBadCRC	NUMBER(20)	DEFAULT –1	A count of the received fiber channel frames with errored CRCs.
fibreStatsRxFrames	NUMBER(20)	DEFAULT –1	A count of the number of fiber channel frames received without errors.
fibreStatsRxOctets	NUMBER(20)	DEFAULT –1	A count of the number of bytes received without error for the fiber channel payload.
fibreStatsRxDiscards	NUMBER(20)	DEFAULT –1	Number of inbound discarded packets.
fibreStatsTxFramesBadCRC	NUMBER(20)	DEFAULT –1	A count of the transmitted fiber channel frames with errored CRCs.
fibreStatsTxFrames	NUMBER(20)	DEFAULT –1	A count of the number of transmitted fiber channel frames.

Column Name	Oracle Data Type	Integrity Constraints	Description
fibreStatsTxOctets	NUMBER(20)	DEFAULT –1	A count of the number of bytes transmitted from the fiber channel frame.
fibreStatsTxDiscards	NUMBER(20)	DEFAULT –1	Number of discarded packets.
fibreStatsLinkResets	NUMBER(20)	DEFAULT –1	A count of the FC-MR software initiated link recovery attempts toward the FC line side because of SONET protection switches.
gfpStatsRxSBitErrors	NUMBER(20)	DEFAULT –1	A count of generic framing procedure (GFP) single bit errors in the core header error check (CHEC).
gfpStatsRxMBitErrors	NUMBER(20)	DEFAULT –1	A count of GFP multibit errors in the CHEC.
gfpStatsRxTypeInvalid	NUMBER(20)	DEFAULT –1	A count of GFP invalid user payload identifier (UPI) field in the type field.
gfpStatsRxSblkCRCErrors	NUMBER(20)	DEFAULT –1	A count of superblock CRC errors in the transparent GFP frame.
RxFCUtilizationStats	NUMBER(10)	DEFAULT –2	Percentage of received line bandwidth used by the ports during consecutive time segments.
TxFCUtilizationStats	NUMBER(10)	DEFAULT –2	Percentage of transmitted line bandwidth used by the ports during consecutive time segments.
StatsLineLastClearedTime	DATE	—	Time stamp indicating the last time statistics were reset.
gfpStatsCSFRaised	NUMBER(20)	DEFAULT –1	—
gfpStatsRoundTripLatency	NUMBER(20)	DEFAULT –1	—
IfInErrors	NUMBER(20)	DEFAULT –1	—
FibreStatsLinkRecoveries	NUMBER(20)	DEFAULT –1	—
FCIngressRxDistanceExtBuffers	NUMBER(20)	DEFAULT –1	—
FCEgressTTxDistanceExtBuffers	NUMBER(20)	DEFAULT –1	—
FibreStatsRxCredits	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
FibreStatsTxCredits	NUMBER(20)	DEFAULT –1	—
FibreStatsZeroTxCredits	NUMBER(20)	DEFAULT –1	—
TxPackets	NUMBER(20)	DEFAULT –1	—
RxPackets	NUMBER(20)	DEFAULT –1	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15 minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
FibreStatsRxRecvrReady	NUMBER(20)	DEFAULT –1	—
FibreStatsTxRecvrReady	NUMBER(20)	DEFAULT –1	—
InvldOrdrdSetDispErrorSum	NUMBER(20)	DEFAULT –1	—
IfOutOversizePkts	NUMBER(20)	DEFAULT –1	—
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.
fibreStatsTxFramesTruncated	NUMBER(20)	DEFAULT –1	—
fibreStatsTxFramesTooLong	NUMBER(20)	DEFAULT –1	—
IfOutErrors	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
PCS49ERRDEC	NUMBER(22)	—	A counter related to Physical Coding Sublayer (PCS) that supports Cervino cards in Ethernet and FC PM tables.
PCS49RXERRBER	NUMBER(22)	—	A counter related to Physical Coding Sublayer (PCS) that supports Cervino cards in Ethernet and FC PM tables.

1.1.146 ONS15454_GFP_PM_Table

The ONS15454_GFP_PM_Table provides monitoring capability to allow operators to identify performance degradation in the GFP multiplexing and framing. GFP describes a framing and encapsulation format for transporting L2/L3 data over SONET, SDH, or OTN transport networks.

Primary key: ONS15454_GFP_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-142: *ONS15454_GFP_PM_Table*

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Fifteen-minute interval threshold value for the PM bucket.
thresholdList	VARCHAR2(1024)	—	Daily threshold value for the PM bucket.
GFPSTATSCSFRAISED	NUMBER(20)	DEFAULT –1	Number of GFP client signal fail frames.

Column Name	Oracle Data Type	Integrity Constraints	Description
GFPSTATSLFDRAISED	NUMBER(20)	DEFAULT –1	Number of GFP LFD frames.
GFPSTATSROUNDTRIPLATENCY	NUMBER(20)	DEFAULT –1	Round-trip delay (in milliseconds [ms]) for the end-to-end FC transport.
GFPSTATSRXCIDINVALID	NUMBER(20)	DEFAULT –1	Number of received data frames with an invalid CID.
GFPSTATSRXCRCERRORS	NUMBER(20)	DEFAULT –1	Number of superblock CRC errors in the transparent GFP frame.
GFPSTATSRXFRAME	NUMBER(20)	DEFAULT –1	Number of received data frames.
GFPSTATSRXMBITERRORS	NUMBER(20)	DEFAULT –1	Number of GFP multibit errors in the CHEC.
GFPSTATSRXOCTETS	NUMBER(20)	DEFAULT –1	Number of received data octets.
GFPSTATSRXSBITERRORS	NUMBER(20)	DEFAULT –1	Number of GFP single-bit errors in the CHEC.
GFPSTATSRXSBLKCRCERRORS	NUMBER(20)	DEFAULT –1	Number of super block CRC errors detected in the received GFP-T frames.
GFPSTATSRXTYPEINVALID	NUMBER(20)	DEFAULT –1	Number of GFP invalid UPI fields in the type field.
GFPSTATSTXFRAME	NUMBER(20)	DEFAULT –1	Number of transmitted data frames.
GFPSTATSTXOCTETS	NUMBER(20)	DEFAULT –1	Number of transmitted data octets.
GFPSTATSRXDISTANCEEXTBUFFERS	NUMBER(20)	DEFAULT –1	Number of receive buffer credits for GFP-T (valid only if distance extension is enabled).
GFPSTATSTXDISTANCEEXTBUFFERS	NUMBER(20)	DEFAULT –1	Number of transmit buffer credits for GFP-T (valid only if distance extension is enabled).
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15 minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.

Column Name	Oracle Data Type	Integrity Constraints	Description
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.
GFPRXCMFFRAME	NUMBER(20)	DEFAULT –1	—
GFPTXCMFFRAME	NUMBER(20)	DEFAULT –1	—
IfInPayloadCrcErrors	NUMBER(20)	DEFAULT –1	—

1.1.147 ONS15454_IFX_PM_Table

The ONS15454_IFX_PM_Table is used for IFX PM data where the module type is an ML-series card.

Primary key: ONS15454_IFX_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-143: ONS15454_IFX_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was

Column Name	Oracle Data Type	Integrity Constraints	Description
			stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
IfName	VARCHAR2(40	—	—
IfHCInOctets	NUMBER(20)	DEFAULT 0	—
IfHCOutOctets	NUMBER(20)	DEFAULT 0	—
IfHCInUcasrPkts	NUMBER(20)	DEFAULT 0	—
IfHCOutUcastPkts	NUMBER(20)	DEFAULT 0	—
IfHCInBroadcastPkts	NUMBER(20)	DEFAULT 0	—
IfHCOutBroadcastPkts	NUMBER(20)	DEFAULT 0	—
IfHCInMulticastPkts	NUMBER(20)	DEFAULT 0	—
IfHCOutMulticastPkts	NUMBER(20)	DEFAULT 0	—
IfInDiscards	NUMBER(20)	DEFAULT 0	—
IfOutDiscards	NUMBER(20)	DEFAULT 0	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.148 ONS15454_INVENTORY_NOTES_Table

Primary key: ONS15454_INVENTORY_NOTES_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-144: ONS15454_INVENTORY_NOTES_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SERIALNUMBER	VARCHAR2(16)	NOT NULL	—
NOTE	VARCHAR2(4000)	—	—
INSERTED DATE	—	—	—

1.1.149 ONS15454_IPSLA_PM_Table

Primary key: ONS15454_IPSLA_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-145: ONS15454_IPSLA_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface table index for the interface. Same as IfStaticIndex of Interface_Table.
TimeStamp	DATE	NOT NULL	Interval timestamp for which the set of statistics is stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit mask value. This information is about PM statistics.
inMaintenance	NUMBER(1)	—	Indicates whether the NE is in Maintenance state when the PM bucket is collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for 15-minute PM bucket.
thresholdList	VARCHAR2(1024)	—	Threshold value for daily PM bucket.
rttMonStatsStartTimeIndex	NUMBER(20)	DEFAULT –1	Received invalid word error count.
rttMonStatsCompletions	NUMBER(20)	DEFAULT –1	Received disparity error count.

Column Name	Oracle Data Type	Integrity Constraints	Description
rttMonStatsOverThresholds	NUMBER(20)	DEFAULT –1	Received total errored frames.
rttMonStatsNumOfRtt	NUMBER(20)	DEFAULT –1	Received frames too small count.
rttMonStatsRttSum	NUMBER(20)	DEFAULT –1	Received frames too large count.
rttMonStatsRttSum2Low	NUMBER(20)	DEFAULT –1	Received frames with CRC error count.
rttMonStatsRttSum2High	NUMBER(20)	DEFAULT –1	Received total frames count.
rttMonStatsRTTMin	NUMBER(20)	DEFAULT –1	Received total bytes count.
rttMonStatsRttMax	NUMBER(20)	DEFAULT –1	Received frame overflow count.
rttMonStatsMinOfPositivesSD	NUMBER(20)	DEFAULT –1	Tx frames with CRC error count.
rttMonStatsMaxOfPositivesSD	NUMBER(20)	DEFAULT –1	Tx total frames count.
rttMonStatsNumOfPositivesSD	NUMBER(20)	DEFAULT –1	Tx total bytes count.
rttMonStatsSumOfPositivesSD	NUMBER(20)	DEFAULT –1	Tx frame overflow count.
rttMonStatsSum2PositivesSDLow	NUMBER(20)	DEFAULT –1	Approximate count of link resets.
rttMonStatsSum2PositivesSDHigh	NUMBER(20)	DEFAULT –1	GFP received single bit errors in core hdr count.
rttMonStatsMinOfNegativesSD	NUMBER(20)	DEFAULT –1	GFP received multibit errors in core hdr count.
rttMonStatsMaxOfNegativesSD	NUMBER(20)	DEFAULT –1	GFP received frames with invalid type.

Column Name	Oracle Data Type	Integrity Constraints	Description
rttMonStatsNumOfNegatives	NUMBER(20)	DEFAULT –1	GFP received superblock CRC error count.
rttMonStatsSumOfNegativesSD	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2NegativesSDLow	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2NegativesSDHigh	NUMBER(20)	DEFAULT –1	—
rttMonStatsMinOfPositivesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsMaxOfPositivesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsNumOfPositivesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsSumOfPositivesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2PositivesDSLow	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2PositivesDSHigh	NUMBER(20)	DEFAULT –1	—
rttMonStatsMinOfNegativesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsMaxOfNegativesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsNumOfNegativesDS	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
rttMonStatsSumOfNegativesDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2NegativesDSLow	NUMBER(20)	DEFAULT –1	—
rttMonStatsSum2NegativesDSHigh	NUMBER(20)	DEFAULT –1	—
rttMonStatsPacketLossSD	NUMBER(20)	DEFAULT –1	—
rttMonStatsPacketLossDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsPacketOutOfSequence	NUMBER(20)	DEFAULT –1	—
rttMonStatsPacketMIA	NUMBER(20)	DEFAULT –1	—
rttMonStatsPacketLateArrival	NUMBER(20)	DEFAULT –1	—
rttMonStatsError	NUMBER(20)	DEFAULT –1	—
rttMonStatsBusies	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWSumSD	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWSum2SDLow	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWSum2SDHigh	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWMinSD	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWMaxSD	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWSumDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWSum2DSLow	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
rttMonStatsOWSum2DSHigh	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWMinDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWMaxDS	NUMBER(20)	DEFAULT –1	—
rttMonStatsNumOfOW	NUMBER(20)	DEFAULT –1	—
rttMonStatsOWMinSDNew	NUMBER(20)	DEFAULT –1	—
StatsLineLastClearedTime	DATE	—	
IS24H	NUMBER(1)	NOT Null	Specifies whether the PM data is a 24-hour or 15-minute PM data collection. Values are: 1: 24 hour 0: 15 minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.150 ONS15454_LAG_PM_Table

The ONS15454_LAG_PM_Table is used for is used for ML card CB QOS PM data.

Primary key: ONS15454_LAG_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-146: ONS15454_LAG_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ModuleType	NUMBER(4)	NOT NULL	The module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface. Same as the IfStaticIndex on the Interface Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit Mask Value
inMaintenance	NUMBER(1)	—	Indicates whether the NE was under maintenance when the PM bucket was collected.
InterfaceName	NUMBER(20)	DEFAULT 0	—
LACPDUsRx	NUMBER(20)	DEFAULT 0	—
MarkerPDUsRx	NUMBER(20)	DEFAULT 0	—
MarkerResponsePDUsRx	NUMBER(20)	DEFAULT 0	—
UnknownRx	NUMBER(20)	DEFAULT 0	—
IllegalRx	NUMBER(20)	DEFAULT 0	—
LACPDUsTx	NUMBER(20)	DEFAULT 0	—
MarkerPDUsTx	NUMBER(20)	DEFAULT 0	—
MarkerResponsePDUsTx	NUMBER(20)	DEFAULT 0	—
IS24H	NUMBER(1)	NOT NULL	Indicates whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.151 ONS15454_Match_STMT_PM_Table

The ONS15454_Match_STMT_PM_Table is used for PM data where the module type is an ML-series card.

Primary key: None

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-147: ONS15454_Match_STMT_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit Mask Value
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
CBQosMatchPrePolicyPkt64	NUMBER(20)	DEFAULT 0	—
CBQosMatchPrePolicyByte64	NUMBER(20)	DEFAULT 0	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.152 ONS15454_NE_Inventory_Table

The ONS15454_NE_Inventory_Table stores inventory information about ONS 15454 SONET and ONS 15454 SDH NEs.

Primary key: ONS15454_NE_INVENTORY_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-148: ONS15454_NE_Inventory_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
PhysicalLoc	NUMBER(20)	NOT NULL	Physical location of the card.
EquipmentType	NUMBER(10)	—	Equipment type index in the Module_Type_Table. This value represents the configured value of the slot.
AdminState	VARCHAR2(32)	—	Admin state of the equipment.
ServiceState	VARCHAR2(150)	—	Service state of the equipment.
ActualEquipmentType	NUMBER(10)	—	Equipment type index in the Module_Type_Table. This value represents the actual card in the slot.
HWPartNumber	VARCHAR2(16)	—	Hardware version of the module.
SerialNumber	VARCHAR2(16)	—	Serial number of the module.
Name	VARCHAR2(100)	—	Name of the module.
ApplicationFileName	VARCHAR2(30)	—	Name of the software binary file for the module.
ExtraInfo	VARCHAR2(4000)	—	Extra information about the module.
FirmwareVersion	VARCHAR2(16)	—	Firmware version of the module.

Column Name	Oracle Data Type	Integrity Constraints	Description
EquipmentState	NUMBER(1)	DEFAULT 1	State of the module. Values are: 0: Not Present 1: Mismatch 2: Loading 3: Failed 4: Standby 5: Active 6: Empty 7: Deleted 8: Present
EntityID	NUMBER(10)	—	Entity ID assigned to the module.
HardwareRevision	VARCHAR2(7)	—	Manufacturer/hardware revision.
CleiCode	VARCHAR2(20)	—	CLEI code for the card.
InventoryCode	VARCHAR2(40)	DEFAULT NULL	Provisionable inventory code.
ProductId	VARCHAR2(20)	DEFAULT NULL	Product ID.
VersionId	VARCHAR2(20)	DEFAULT NULL	Version ID.
LastInterfaceChange	NUMBER(20)	DEFAULT 0	Last interface change time stamp.

1.1.153 ONS15454_OTN_Fec_PM_Table

The ONS15454_OTN_Fec_PM_Table is used for OC-N PM data where the module type is an OC-N line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_OTNFEC_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-149 : ONS15454_OTN_Fec_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access the ONS15454_OTN_Fec_PM_Table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type.

Column Name	Oracle Data Type	Integrity Constraints	Description
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NetIfIndex	NUMBER(20)	NOT NULL	Table index for the interface. Same as IfStaticIndex of INTERFACE_TABLE.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid Period Other: Bit Mask Value (info about PM stats)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
BitErrsCorrected	NUMBER(20)	DEFAULT –1	Number of bit errors corrected.
ByteErrsCorrected	NUMBER(20)	DEFAULT –1	Number of byte errors corrected.
ZeroBitErrsDetected	NUMBER(20)	DEFAULT –1	Number of zero bit errors detected.
OneBitErrsDetected	NUMBER(20)	DEFAULT –1	Number of one-bit errors detected.
UnCorrectableWord	NUMBER(20)	DEFAULT –1	Number of uncorrectable words.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.154 ONS15454_OTN_Path_PM_Table

The ONS15454_OTN_Path_PM_Table is used for OCN PM data where the module type is an OCN Line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_OTNPATH_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-150: ONS15454_OTN_Path_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access the ONS15454_OTN_Path_PM_Table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface. Same as IfStaticIndex of INTERFACE_TABLE.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid Period Other: Bit Mask Value (info about PM stats)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
ErrSecs	NUMBER(10)	DEFAULT –1	Number of seconds when at least one path layer BIP error is detected or an SEF or LOS defect is present.
SevErrSecs	NUMBER(10)	DEFAULT –1	Count when K or more path layer BIP errors are detected. (Refer to Telcordia GR-253-CORE for the value of K.)
UnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a path interface in the interval.
BackGroundBlockErr	NUMBER(10)	DEFAULT –1	Number of background block errors in the interval.
FailureCount	NUMBER(10)	DEFAULT –1	Number of failed events.
ErrSecsRatio	NUMBER(10)	DEFAULT –1	Error seconds ratio when at least one path layer BIP error is detected or an SEF or LOS is detected.
SevErrSecsRatio	NUMBER(10)	DEFAULT –1	Severely errored seconds ratio.
BackGroundBlockErrRatio	NUMBER(10)	DEFAULT –1	Number of background block errors ratio in the interval.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeErrSecs	NUMBER(10)	DEFAULT –1	Number of far-end errored seconds encountered by an OCN interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeSevErrSecs	NUMBER(10)	DEFAULT –1	Number of far-end severely errored seconds encountered by an OCN interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of far-end unavailable seconds encountered by an OCN interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeBackGroundBlockErrs	NUMBER(10)	DEFAULT –1	Number of far-end background block errors in the interval.
FeFailureCount	NUMBER(10)	DEFAULT –1	Number of far-end failure count encountered in the interval.
FeErrSecsRatio	NUMBER(10)	DEFAULT –1	Far-end error seconds ratio when at least one path layer BIP error was detected or

Column Name	Oracle Data Type	Integrity Constraints	Description
			an SEF or LOS defect is present.
FeSevErrSecsRatio	NUMBER(10)	DEFAULT –1	Far-end severely errored seconds ratio.
FeBackGroundBlockErrRatio	NUMBER(10)	DEFAULT –1	Number of far-end background block errors ratio in the interval.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.155 ONS15454_OTN_Sec_PM_Table

The ONS15454_OTN_Sec_PM_Table is used for OCN PM data where the module type is an OCN Line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_OTNSEC_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-151: ONS15454_OTN_Sec_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type.

Column Name	Oracle Data Type	Integrity Constraints	Description
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NetIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid Period Other: Bit Mask Value (info about PM stats)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
ErrSecs	NUMBER(10)	DEFAULT –1	Number of seconds when at least one section layer BIP error is detected or an SEF or LOS defect is present.
SevErrSecs	NUMBER(10)	DEFAULT –1	Count when K or more section layer BIP errors are detected. (Refer to Telcordia GR-253-CORE for the value of K.)
UnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a Section interface in the interval.
BackGroundBlockErr	NUMBER(10)	DEFAULT –1	Number of background block error in the interval.
FailureCount	NUMBER(10)	DEFAULT –1	Number of failed events.
ErrSecsRatio	NUMBER(10)	DEFAULT –1	Error seconds ratio when at least one section layer BIP error is detected or an SEF or LOS is detected.
SevErrSecsRatio	NUMBER(10)	DEFAULT –1	Severely errored seconds ratio.
BackGroundBlockErrRatio	NUMBER(10)	DEFAULT –1	Number of background block errors ratio in the interval.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeErrSecs	NUMBER(10)	DEFAULT –1	Number of far-end errored seconds encountered by a CHANNEL interface in the interval. This is valid for the XP_MR_10G and MXP_2.5G_10G cards.

Column Name	Oracle Data Type	Integrity Constraints	Description
FeSevErrSecs	NUMBER(10)	DEFAULT –1	Number of far-end severely errored seconds encountered by a CHANNEL interface in the interval. This is valid for the XP_MR_10G and MXP_2.5G_10G cards.
FeUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of far-end unavailable seconds encountered by a CHANNEL interface in the interval. This is valid for the XP_MR_10G and MXP_2.5G_10G cards.
FeBackGroundBlockErrs	NUMBER(10)	DEFAULT –1	Number of far-end back ground block errors in the interval.
FeFailureCount	NUMBER(10)	DEFAULT –1	Number of far-end failure count encountered in the interval.
FeErrSecsRatio	NUMBER(10)	DEFAULT –1	Far-end error seconds ratio when at least one section layer BIP error was detected or an SEF or LOS defect is present.
FeSevErrSecsRatio	NUMBER(10)	DEFAULT –1	Far-end severely errored seconds ratio.
FeBackGroundBlockErrRatio	NUMBER(10)	DEFAULT –1	Number of far-end background block errors ratio in the interval.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.156 ONS15454_PHY_LAYER_PM_Table

The ONS 15454_PHY_LAYER_PM_Table is used for OC-N/STM Physical PM data of the ONS 15327, ONS 15454 SONET, ONS 15454 SDH, and ONS 15600 NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_PHY_LAYER_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-152: ONS15454_PHY_LAYER_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access this table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NetIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid Period Other: Bit Mask Value (info about PM stats)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
LaserBiasCurrent	NUMBER(10)	DEFAULT –1	Laser bias current (normalized).
OpticalPowerTransmitted	NUMBER(10)	DEFAULT –1	Optical power transmitted (normalized).
OpticalPowerReceived	NUMBER(10)	DEFAULT –1	Optical power received (normalized).
MinLaserBias	NUMBER(10)	DEFAULT –1	Minimum laser bias.
AvgLaserBias	NUMBER(10)	DEFAULT –1	Average laser bias.
MaxLaserBias	NUMBER(10)	DEFAULT –1	Maximum laser bias.
MinLaserTemp	NUMBER(10)	DEFAULT –1	Minimum laser temperature.
AvgLaserTemp	NUMBER(10)	DEFAULT –1	Average laser temperature.
MaxLaserTemp	NUMBER(10)	DEFAULT –1	Maximum laser temperature.

Column Name	Oracle Data Type	Integrity Constraints	Description
MinReceivedLaserTemp	NUMBER(10)	DEFAULT –1	Minimum received laser temperature.
AvgReceivedLaserTemp	NUMBER(10)	DEFAULT –1	Average received laser temperature.
MaxReceivedLaserTemp	NUMBER(10)	DEFAULT –1	Maximum received laser temperature.
MinTransReceiverVoltage	NUMBER(10)	DEFAULT –1	Minimum transreceiver voltage.
AvgTransReceiverVoltage	NUMBER(10)	DEFAULT –1	Average transreceiver voltage.
MaxTransReceiverVoltage	NUMBER(10)	DEFAULT –1	Maximum transreceiver voltage.
MinTransmittedPower	NUMBER(10)	DEFAULT –1	Minimum transmitted power.
AvgTransmittedPower	NUMBER(10)	DEFAULT –1	Average transmitted power.
MaxTransmittedPower	NUMBER(10)	DEFAULT –1	Maximum transmitted power.
MinReceivedPower	NUMBER(10)	DEFAULT –1	Minimum received power.
AvgReceivedPower	NUMBER(10)	DEFAULT –1	Average received power.
MaxReceivedPower	NUMBER(10)	DEFAULT –1	Maximum received power.
MinPassThroughPower	NUMBER(10)	DEFAULT –1	Minimum passthrough power.
AvgPassThroughPower	NUMBER(10)	DEFAULT –1	Average passthrough power.
MaxPassThroughPower	NUMBER(10)	DEFAULT –1	Maximum passthrough power.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.
MinTxOSCPower	NUMBER(10)	DEFAULT –1	Minimum transmitted OSC power.
AvgTxOSCPower	NUMBER(10)	DEFAULT –1	Average transmitted OSC power.
MaxTxOSCPower	NUMBER(10)	DEFAULT –1	Maximum transmitted OSC power.
MinRxOSCPower	NUMBER(10)	DEFAULT –1	Minimum received OSC power.
AvgRxOSCPower	NUMBER(10)	DEFAULT –1	Average received OSC power.
MaxRxOSCPower	NUMBER(10)	DEFAULT –1	Maximum received OSC power.
MinPMD	NUMBER(10)	—	Minimum Polarization Mode Dispersion (PMD) value. The value is expressed in picoseconds (ps).
AvgPMD	NUMBER(10)	—	Average PMD value. The value is expressed in ps.
MaxPMD	NUMBER(10)	—	Maximum PMD value. The value is expressed in ps.
MinOSNR	NUMBER(10)	—	Minimum Optical Signal-to-Noise ratio (OSNR) value. The value is expressed in decibel-milliwatt (dBm).
AvgOSNR	NUMBER(10)	—	Average OSNR value. The value is expressed in dBm.
MaxOSNR	NUMBER(10)	—	Maximum OSNR value. The value is expressed in dBm.
CD	NUMBER(22)	—	Chromatic dispersion (CD) is an optical performance monitoring value that is reported only as the current value. All PM and real-time tables will have the CD value, but a valid value can be found only on real-time tables.

1.1.157 ONS15454_Police_PM_Table

The ONS15454_Police_PM_Table is used for CBQOS PM data where the module type is an ML-series card.

Primary key: ONS15454_QOSPLS_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-153: ONS15454_Police_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access this table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates the status of the PM bucket. Values are: 0: Valid period Other: Bit mask value
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
CBQosPoliceConformedPkt64	NUMBER(20)	DEFAULT 0	—
CBQosPoliceConformedByte64	NUMBER(20)	DEFAULT 0	—
CBQosPoliceExceededPkt64	NUMBER(20)	DEFAULT 0	—
CBQosPoliceExceededByte64	NUMBER(20)	DEFAULT 0	—
CBQosPoliceViolatedPkt64	NUMBER(20)	DEFAULT 0	—
CBQosPoliceViolatedByte64	NUMBER(20)	DEFAULT 0	—

Column Name	Oracle Data Type	Integrity Constraints	Description
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.158 ONS15454_POS_PM_Table

The ONS15454_POS_PM_Table is used for POS PM data where the module type is an ML-series card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_DATA_TABLESPACE

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-154: ONS15454_POS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access this table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface. Same as IfStaticIndex of INTERFACE_TABLE.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid Period Other: Bit Mask Value (info about PM stats)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
POSStatsRxPreHdlcBytes	NUMBER(20)	DEFAULT –1	Number of bytes received prior to the bytes undergoing high-level data link control (HDLC) encapsulation by the policy engine.
POSStatsRxPostHdlcBytes	NUMBER(20)	DEFAULT –1	Number of bytes received after the bytes undergoing HDLC encapsulation by the policy engine.
PosStatsRxPackets	NUMBER(20)	DEFAULT –1	Number of packets received since the last counter reset.
PosStatsRxNormalPackets	NUMBER(20)	DEFAULT –1	Number of normal packets received since the last counter reset.
PosStatsRxShorts	NUMBER(20)	DEFAULT –1	Number of short packets received since the last counter reset.
PosStatsRxRunts	NUMBER(20)	DEFAULT –1	Total number of frames received that are less than 64 bytes in length and have a CRC error.
PosStatsRxLongs	NUMBER(20)	DEFAULT –1	Number of long packets received since the last counter reset.
PosStatsRxTotalErrors	NUMBER(20)	DEFAULT –1	Total number of received errors.
PosStatsRxCrcErrors	NUMBER(20)	DEFAULT –1	CRC errors on received frames.
PosStatsRxInputDropPackets	NUMBER(20)	DEFAULT –1	Number of packets received that have been dropped in input.
PosStatsRxInputAbortPackets	NUMBER(20)	DEFAULT –1	Number of packets received that have been aborted in input.
PosStatsTxPreHdlcBytes	NUMBER(20)	DEFAULT –1	Good pre-HDLC bytes transmitted. Bytes from packets which were aborted due to FIFO underflow or an error from the system interface or are smaller than the programmed minimum number of bytes are not included,. These bytes are from the system interface before flags, stuffing, or FCS.
PosStatsTxPostHdlcBytes	NUMBER(20)	DEFAULT –1	Good post-HDLC bytes transmitted. These bytes are from the system interface after the escaping flags and FCS are added.
PosStatsTxPackets	NUMBER(20)	DEFAULT –1	Number of packets transmitted since the last counter reset.

Column Name	Oracle Data Type	Integrity Constraints	Description
PosStatsDropCounts	NUMBER(20)	DEFAULT –1	Drop counts per port.
EtherStatsDropEvents	NUMBER(20)	DEFAULT –1	—
GfpStatsRxSbitErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxMBitErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxTypeInvalid	NUMBER(20)	DEFAULT –1	—
GfpStatsRxCRCErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxCIDInvalid	NUMBER(20)	DEFAULT –1	—
GfpStatsCSFRaised	NUMBER(20)	DEFAULT –1	—
GfpStatsRxFrame	NUMBER(20)	DEFAULT –1	—
GfsStatsTxFrame	NUMBER(20)	DEFAULT –1	—
GfpStatsRxOctets	NUMBER(20)	DEFAULT –1	—
GfpStatsTxOctets	NUMBER(20)	DEFAULT –1	—
GfpStatsLFDRaised	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesTooLong	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesBadCRC	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxShortPkts	NUMBER(20)	DEFAULT –1	—
HdlcInOctets	NUMBER(20)	DEFAULT –1	—
HdlcRxAborts	NUMBER(20)	DEFAULT –1	—
HdlcOutOctets	NUMBER(20)	DEFAULT –1	—
IfInOctets	NUMBER(20)	DEFAULT –1	—
IfOutOctets	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
RxPktsDropInternalCongestion	NUMBER(20)	DEFAULT –1	—
IfInDiscards	NUMBER(20)	DEFAULT –1	—
RXETHERUTILIZATIONSTATS	NUMBER(20)	DEFAULT –1	—
TXETHERUTILIZATIONSTATS	NUMBER(20)	DEFAULT –1	—
STATSLINELASTCLEAREDTIME	NUMBER(20)	DEFAULT –1	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
IfInPayloadCrcErrors	NUMBER(20)	DEFAULT –1	Number of payload CRC errors received since the last counter reset.
IfOutPayloadCrcErrors	NUMBER(20)	DEFAULT –1	Number of payload CRC errors transmitted since the last counter reset.
IfOutOversizePackets	NUMBER(20)	DEFAULT –1	Number of oversize packets transmitted since the last counter reset.
HDLCPKTDROPS	NUMBER(20)	DEFAULT –1	—
IfOutDiscards	NUMBER(20)	DEFAULT –1	—
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.159 ONS15454_Privilege_Table

The ONS15454_Privilege_Table stores user privilege information about ONS 15454 SONET and ONS 15454 SDH NEs.

Primary key: PK_ONS15454_PRIV_ID

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-155: ONS15454_Privilege_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
PrivilegeId	NUMBER(10)	NOT NULL	Privilege ID. Values are: 0: Retrieve 1: Maintenance 2: Provisioning 3: Superuser
Privilege	VARCHAR2(30)	NOT NULL	Privilege level. Values are: 0: Retrieve 1: Maintenance 2: Provisioning 3: Superuser
Description	VARCHAR2(64)	—	Description.

1.1.160 ONS15454_RPR_PM_Table

Primary key: ONS15454_RPR_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-156 ONS15454_RPR_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID used to access this table; same as NEDBAccessID in the NE_Info_Table.
ModuleType	NUMBER(4)	NOT NULL	Module type to which the PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface table index for the interface. This is the same as the IfStaticIndex of the Interface_Table.
TimeStamp	DATE	NOT NULL	Interval timestamp for which the set of statistics is stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Not used in Prime Optical R9.6.

Column Name	Oracle Data Type	Integrity Constraints	Description
inMaintenance	NUMBER(1)	—	Indicates whether the NE is in Maintenance state when the PM bucket is collected.
fifteenMinThresholdMask	NUMBER(10)	—	Not used in Prime Optical R9.6.
thresholdList	VARCHAR2(1024)	—	Threshold value for the daily PM bucket.
SpanInUcastClassCFrames	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassCOctets	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassCFrames	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassCOctets	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassAFrames	NUMBER(20)	DEFAULT –1	—
SpanInUcastClassAOctets	NUMBER(20)	DEFAULT –1	—
SpanInMcastClassAFrames	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
SpanInMcastClassAOctets	NUMBER(20)	DEFAULT –1	—
SpanInCtrlFrames	NUMBER(20)	DEFAULT –1	—
SpanInOamEchoFrames	NUMBER(20)	DEFAULT –1	—
SpanInOamFlushFrames	NUMBER(20)	DEFAULT –1	—
SpanInOamOrgFrames	NUMBER(20)	DEFAULT –1	—
SpanInTopoAtdFrames	NUMBER(20)	DEFAULT –1	—
SpanInTopoChkSumFrames	NUMBER(20)	DEFAULT –1	—
SpanInTopoTpFrames	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassCFrames	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassCOctets	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassCFrames	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassCOctets	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
SpanOutUcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassAFrames	NUMBER(20)	DEFAULT –1	—
SpanOutUcastClassAOctets	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassAFrames	NUMBER(20)	DEFAULT –1	—
SpanOutMcastClassAOctets	NUMBER(20)	DEFAULT –1	—
SpanOutCtrlFrames	NUMBER(20)	DEFAULT –1	—
SpanOutOamEchoFrames	NUMBER(20)	DEFAULT –1	—
SpanOutOamFlushFrames	NUMBER(20)	DEFAULT –1	—
SpanOutOamOrgFrames	NUMBER(20)	DEFAULT –1	—
SpanOutTopoAtFrames	NUMBER(20)	DEFAULT –1	—
SpanOutTopoChkSumFrames	NUMBER(20)	DEFAULT –1	—
SpanOutTopoTpFrames	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassCFrames	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassCOctets	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassCFrames	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassCOctets	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
ClientInUcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassAFrames	NUMBER(20)	DEFAULT –1	—
ClientInUcastClassAOctets	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassAFrames	NUMBER(20)	DEFAULT –1	—
ClientInMcastClassAOctets	NUMBER(20)	DEFAULT –1	—
ClientInBcastFrames	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassCFrames	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassCOctets	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
ClientOutMcastClassCFrames	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassCOctets	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassBEirFrames	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassBEirOctets	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassBCirFrames	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassBCirOctets	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassAFrames	NUMBER(20)	DEFAULT –1	—
ClientOutUcastClassAOctets	NUMBER(20)	DEFAULT –1	—
ClientOutMcastClassAFrames	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
ClientOutMcastClassAOctets	NUMBER(20)	DEFAULT –1	—
ClientOutBcastFrames	NUMBER(20)	DEFAULT –1	—
ErrorBadParityFrames	NUMBER(20)	DEFAULT –1	—
ErrorBadHecFrames	NUMBER(20)	DEFAULT –1	—
ErrorTt1ExpFrames	NUMBER(20)	DEFAULT –1	—
ErrorTooLongFrames	NUMBER(20)	DEFAULT –1	—
ErorTooShortFrames	NUMBER(20)	DEFAULT –1	—
ErrorBadFcsFrames	NUMBER(20)	DEFAULT –1	—
ErrorSelfSrcUcastFrames	NUMBER(20)	DEFAULT –1	—
ErrorPmdAbortFrames	NUMBER(20)	DEFAULT –1	—
ErrorBadAddrFrames	NUMBER(20)	DEFAULT –1	—
ErrorContainedFrames	NUMBER(20)	DEFAULT –1	—
ErrorScffErrors	NUMBER(20)	DEFAULT –1	—
PortCounterError	NUMBER(20)	DEFAULT –1	—
ErrorOverSizeFrames	NUMBER(20)	DEFAULT –1	—
TimeLastCleared	DATE	—	—
IS24	NUMBER(1)	NOT NULL	Specifies whether the PM data is for a 24-hour or 15-minute PM data collection. Values are: 1: 24 hour 0: 15 minutes
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for the PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.161 ONS15454_SONET_Line_PM_Table

The ONS15454_SONET_Line_PM_Table stores SONET line PM information about ONS 15327 and ONS 15454 SONET NEs. This table is used for OC-N PM data where the module type is an OC-N or EC1 line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_LINE_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-157: ONS15454_SONET_Line_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about

Column Name	Oracle Data Type	Integrity Constraints	Description
			PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
CodeViolations	NUMBER(10)	DEFAULT –1	Number of LCVs encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
ErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
SevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
UnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
PpjcPdet	NUMBER(10)	DEFAULT –1	Positive Pointer Justification Count, Path Detected is a count of the positive pointer justifications detected on a particular path on an incoming signal.
NpjcPdet	NUMBER(10)	DEFAULT –1	Negative Pointer Justification Count, Path Detected is a count of the negative pointer justifications detected on a particular path on an incoming signal.
PpjcPgen	NUMBER(10)	DEFAULT –1	Positive Pointer Justification Count, Path Generated is a count of the positive pointer justifications generated for a particular path.
NpjcPgen	NUMBER(10)	DEFAULT –1	Negative Pointer Justification Count, Path Generated is a count of the negative pointer justifications generated for a particular path.

Column Name	Oracle Data Type	Integrity Constraints	Description
FcL	NUMBER(10)	DEFAULT –1	Line failure count (FC-L) encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards. FC-L is a count of the number of near-end line failure events. A failure event begins when a line alarm indication signal (AIS-L) failure is declared or when a lower-layer, traffic-related, near-end failure is declared. This failure event ends when the failure is cleared. A failure event that begins in one period and ends in another period is counted only in the period where it begins.
Psc	NUMBER(10)	DEFAULT –1	Protection switching count (PSC) encountered by an OC-N interface in the interval. This is valid for the OC12 and OC3 only. In a 1 + 1 protection scheme for a working card, PSC is a count of the number of times service switches from a working card to a protection card plus the number of times service switches back to the working card. For a protection card, PSC is a count of the number of times service switches to a working card from a protection card plus the number of times service switches back to the protection card. The PSC PM is only applicable if revertive line-level protection switching is used.
Psd	NUMBER(10)	DEFAULT –1	Protection switching duration (PSD) count encountered by an OC-N interface in the interval. This is valid for the OC12 and OC3 only. PSD applies to the length of time, in seconds, that service is carried on another line. For a working line, PSD is a count of the number of seconds that service was carried on the protection line. For the protection line, PSD is a count of the seconds that the line was used to carry service. The PSD PM is only applicable if revertive line-level protection switching is used.

Column Name	Oracle Data Type	Integrity Constraints	Description
PjDiff	NUMBER(10)	DEFAULT –1	Displays the sum of the absolute values of differences between positive transmitted and received, and negative transmitted and received. The important metric on pointer justification is not the exact counts, but how many were absorbed.
PjpSec	NUMBER(10)	DEFAULT –1	Indicates the number of Pointer Justification Count Seconds detected on a particular path.
PjnSec	NUMBER(10)	DEFAULT –1	Indicates the number of Pointer Justification Count Seconds generated for a particular path.
Pscw	NUMBER(10)	DEFAULT –1	Protection switching count–working.
Psdw	NUMBER(10)	DEFAULT –1	Protection switching duration–working.
Pscr	NUMBER(10)	DEFAULT –1	Protection switching count–ring.
Psdr	NUMBER(10)	DEFAULT –1	Protection switching duration–ring.
Pscs	NUMBER(10)	DEFAULT –1	Protection switching count–span.
Psds	NUMBER(10)	DEFAULT –1	Protection switching duration–span.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeCodeViolations	NUMBER(10)	DEFAULT –1	Number of line code violations encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
FeFcL	NUMBER(10)	DEFAULT –1	Number of failure count encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.162 ONS15454_SONET_Path_PM_Table

The ONS15454_SONET_Path_PM_Table stores SONET path PM information about ONS 15327 and ONS 15454 SONET NEs. This table is used for path PM data where the module type is a DS1, DS1N, DS3, DS3E, DS3N, DS3NE, DS3XM, OC-N, or EC1 line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_PATH_LINE_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-158: ONS15454_SONET_Path_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	DEFAULT 0	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about

Column Name	Oracle Data Type	Integrity Constraints	Description
			PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
SonetPathCodeViolations	NUMBER(10)	DEFAULT –1	Number of LCVs encountered by a path interface in the interval. This is valid for the OC12, OC3, and DS3XM cards.
SonetPathErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a path interface in the interval. This is valid for the OC12, OC3, and DS3XM cards.
SonetPathSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a path interface in the interval. This is valid for the OC12, OC3, and DS3XM cards.
SonetPathUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a path interface in the interval. This is valid for the OC12, OC3, and DS3XM cards.
SonetPathFC	NUMBER(10)	DEFAULT –1	Failure count. Number of path failure events.
PpjcPdet	NUMBER(10)	DEFAULT –1	Positive Pointer Justification Count, Path Detected is a count of the positive pointer justifications detected on a particular path on an incoming signal.
NpjcPdet	NUMBER(10)	DEFAULT –1	Negative Pointer Justification Count, Path Detected is a count of the negative pointer justifications detected on a particular path on an incoming signal.
PpjcPgen	NUMBER(10)	DEFAULT –1	Positive Pointer Justification Count, Path Generated is a count of the positive pointer justifications generated for a particular path.
NpjcPgen	NUMBER(10)	DEFAULT –1	Negative Pointer Justification Count, Path Generated is a count of the negative pointer justifications generated for a particular path.

Column Name	Oracle Data Type	Integrity Constraints	Description
PjDiff	NUMBER(10)	DEFAULT –1	Displays the sum of the absolute values of differences between positive transmitted and received, and negative transmitted and received. The important metric on pointer justification is not the exact counts, but how many were absorbed.
PjpSec	NUMBER(10)	DEFAULT –1	Indicates the number of Pointer Justification Count Seconds detected on a particular path.
PjnSec	NUMBER(10)	DEFAULT –1	Indicates the number of Pointer Justification Count Seconds generated for a particular path.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeSonetPathCodeViolations	NUMBER(10)	DEFAULT –1	Number of code violations encountered by a path interface in the interval. This is valid for the far-end OC12, OC3, and DS3XM cards.
FeSonetPathErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a path interface in the interval. This is valid for the far-end OC12, OC3, and DS3XM cards.
FeSonetPathSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a path interface in the interval. This is valid for the far-end OC12, OC3, and DS3XM cards.
FeSonetPathUnavaliableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a path interface in the interval. This is valid for the far-end OC12, OC3, and DS3XM cards.
FeSonetPathFC	NUMBER(10)	DEFAULT –1	Number of far-end path failure events.
STSPathWidth	NUMBER(4)	DEFAULT –1	Size of the STS. This is valid for the OC-N/STM-N cards. Possible values are: 1: STS-1 3: STS-3c 6: STS-6c 12: STS-12c
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

1.1.163 ONS15454_SONET_Sec_PM_Table

The ONS15454_SONET_Sec_PM_Table stores OC-N or EC1 line card PM information about ONS 15327 and ONS 15454 SONET NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_SONET_SEC_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-159: ONS15454_SONET_Sec_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
SevErrFrameSecs	NUMBER(10)	DEFAULT –1	Number of section severely errored frame seconds encountered by an OC-N interface in the interval. This is valid for the OC12, OC3, and EC1 cards.

Column Name	Oracle Data Type	Integrity Constraints	Description
CodeViolations	NUMBER(10)	DEFAULT –1	Number of bit interleaved parity (BIP) errors detected at the session layer.
ErrSecs	NUMBER(10)	DEFAULT –1	Number of seconds when at least one section layer BIP error was detected or a severely errored frame (SEF) or LOS defect was present.
SevErrSecs	NUMBER(10)	DEFAULT –1	Severely errored seconds is a count when K or more section-layer BIP errors were detected. (Refer to Telcordia GR-253-CORE for the value of K.)
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.164 ONS15454_SONET_VT_PM_Table

The ONS15454_SONET_VT_PM_Table stores VT line card PM information about ONS 15327 and ONS 15454 SONET NEs. This table is used for VT PM data where the module type is a DS1, DS1N, or DS3XM line card.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454_VT_LINE_PM_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-160: ONS15454_SONET_VT_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	DEFAULT 0	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
SonetVTCodeViolations	NUMBER(10)	DEFAULT –1	Number of coding violations encountered by a SONET VT in the interval.
SonetVTErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a VT interface in the interval. This is valid for the DS3XM card.
SonetVTSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a VT interface in the interval. This is valid for the DS3XM card.
SonetLineUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a VT interface in the interval. This is valid for the DS3XM card.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeSonetVTCodeViolations	NUMBER(10)	DEFAULT –1	Number of coding violations encountered by a SONET VT in the interval (far end).
FeSonetVTErrSecs	NUMBER(10)	DEFAULT –1	Number of errored seconds encountered by a VT interface in the interval (far end). This is valid for the DS3XM card.
FESonetVTSevErrSecs	NUMBER(10)	DEFAULT –1	Number of severely errored seconds encountered by a VT interface in the interval (far end). This is valid for the

Column Name	Oracle Data Type	Integrity Constraints	Description
			DS3XM card.
FeSonetLineUnavailableSecs	NUMBER(10)	DEFAULT –1	Number of unavailable seconds encountered by a VT interface in the interval (far end). This is valid for the DS3XM card.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
VTTYPE	NUMBER(1)	DEFAULT 0	Type of VT. Values are: 0: VT1_5 1: VT2*/
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.165 ONS15454_User_Table

The ONS15454_User_Table stores user information about ONS 15454 SONET and ONS 15454 SDH NEs.

Primary key: ONS15454_USER_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-161: ONS15454_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_TABLE.
UserId	VARCHAR2(64)	NOT NULL	ONS 15454 SONET and ONS 15454 SDH username.

Column Name	Oracle Data Type	Integrity Constraints	Description
PrivilegeId	NUMBER(10)	—	Privilege level.
Lockout	NUMBER(1)	DEFAULT 0	Current lockout status.
LastLoginTime	DATE	—	Time stamp for last login.
FailedLoginCount	NUMBER(10)	—	Count for failed login attempts.
Disabled	NUMBER(1)	DEFAULT 0	Disabled status.
ChangePwdNxtLogin	NUMBER(1)	DEFAULT 0	Change password on next login status.

1.1.166 ONS15454SDH_E1_PM_Table

The ONS15454SDH_E1_PM_Table stores PM information about the E1 card in ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_E1_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-162: ONS15454SDH_E1_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT –1	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(20)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.

Column Name	Oracle Data Type	Integrity Constraints	Description
cvl	NUMBER(10)	DEFAULT –1	Line coding violation (CV-L) indicates the number of coding violations occurring on the line. This parameter is a count of bipolar violations (BPVs) and excessive zeros (EXZs) occurring over the accumulation period.
esl	NUMBER(10)	DEFAULT –1	Line errored seconds (ES-L) is a count of the seconds containing one or more anomalies (BPV + EXZ) and/or defects (loss of signal) on the line.
sesl	NUMBER(10)	DEFAULT –1	Line severely errored seconds (SES-L) is a count of the seconds containing more than a particular quantity of anomalies (BPV + EXZ > 2048) and/or defects on the line.
lossl	NUMBER(10)	DEFAULT –1	—
RxPEb	NUMBER(10)	DEFAULT –1	Receive (Rx) path errored block (EB-P) indicates that one or more bits are in error within a block.
RxPBbe	NUMBER(10)	DEFAULT –1	Rx path background block error (BBE-P) is an errored block not occurring as part of a severely errored second (SES).
RxPes	NUMBER(10)	DEFAULT –1	Rx path errored seconds (ES-P) is a 1-second period with one or more errored blocks or at least one defect.
RxPses	NUMBER(10)	DEFAULT –1	Rx path severely errored seconds (SES-P) is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of errored seconds (ES).
RxPUas	NUMBER(10)	DEFAULT –1	Rx path unavailable seconds (E1 Rx UAS-P) is a count of 1-second intervals when the E1 path is unavailable on the receive end of the signal. The E1 path is unavailable when ten consecutive SESs occur. The ten SESs are included in unavailable time. After it becomes unavailable, the E1 path becomes available when ten consecutive seconds occur with no SESs. The ten seconds with no SESs are excluded from unavailable time.
RxPEsr	NUMBER(10)	DEFAULT –1	Rx path errored second ratio (ESR-P) is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
RxPSesr	NUMBER(10)	DEFAULT –1	Rx path severely errored second ratio (SESR-P) is the ratio of SES to total seconds in available time during a fixed measurement interval.
RxPBber	NUMBER(10)	DEFAULT –1	Rx path background block error ratio (BBER) is the ratio of background block errors (BBE) to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
TxAiss	NUMBER(10)	DEFAULT –1	Tx alarm indication signal seconds (AISS) is a count of 1-second intervals containing one or more AIS defects.
TxPEb	NUMBER(10)	DEFAULT –1	Tx EB-P indicates that one or more bits are in error within a block.
TxPBbe	NUMBER(10)	DEFAULT –1	Tx path background block error (BBE-P) is an errored block not occurring as part of an SES.
TxPes	NUMBER(10)	DEFAULT –1	Tx ES-P is a 1-second period with one or more errored blocks or at least one defect.
TxPses	NUMBER(10)	DEFAULT –1	Tx path severely errored seconds (SES-P) is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
TxPUas	NUMBER(10)	DEFAULT –1	E1 Tx UAS-P is a count of 1-second intervals when the E1 path is unavailable on the transmit end of the signal. The E1 path is unavailable when ten consecutive SESs occur. The ten SESs are included in unavailable time. After it becomes unavailable, the E1 path becomes available when ten consecutive seconds occur with no SESs. The ten seconds with no SESs are excluded from unavailable time.
TxPEsr	NUMBER(10)	DEFAULT –1	Tx ESR-P is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
TxPSesr	NUMBER(10)	DEFAULT –1	Tx SESR-P is the ratio of SES to total seconds in available time during a fixed measurement interval.
TxPBber	NUMBER(10)	DEFAULT –1	Tx BBER-P is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total

Column Name	Oracle Data Type	Integrity Constraints	Description
			blocks excludes all blocks during SESs.
RxAiss	NUMBER(10)	DEFAULT –1	Rx AISS is a count of 1-second intervals containing one or more AIS defects.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeRxPEb	NUMBER(10)	DEFAULT –1	Rx EB-P indicates that one or more bits are in error within a block (far end).
FeRxPEs	NUMBER(10)	DEFAULT –1	Rx ES-P is a 1-second period with one or more errored blocks or at least one defect (far end).
FeRxPSes	NUMBER(10)	DEFAULT –1	Rx path severely errored seconds (SES-P) is a 1-second period containing greater than 30 percent errored blocks or at least one defect (far end). SES is a subset of ES.
FeRxPBbe	NUMBER(10)	DEFAULT –1	Rx path background block error (BBE-P) is an errored block not occurring as part of an SES (far end).
FeRxPUas	NUMBER(10)	DEFAULT –1	Rx UAS-P is a count of 1-second intervals when the E1 path is unavailable on the receive end of the signal (far end). The E1 path is unavailable when ten consecutive SESs occur. The ten SESs are included in unavailable time. After it becomes unavailable, the E1 path becomes available when ten consecutive seconds occur with no SESs. The ten seconds with no SESs are excluded from unavailable time.
FeRxPEsr	NUMBER(10)	DEFAULT –1	Rx ESR-P is the ratio of errored seconds to total seconds in available time during a fixed measurement interval (far end).
FeRxPSesr	NUMBER(10)	DEFAULT –1	Rx SESR-P is the ratio of SES to total seconds in available time during a fixed measurement interval (far end).
FeRxPBber	NUMBER(10)	DEFAULT –1	Rx BBER-P is the ratio of BBE to total blocks in available time during a fixed measurement interval (far end). The count of total blocks excludes all blocks during SESs.

Column Name	Oracle Data Type	Integrity Constraints	Description
FeTxPEb	NUMBER(10)	DEFAULT –1	Tx EB-P indicates that one or more bits are in error within a block (far end).
FeTxPEs	NUMBER(10)	DEFAULT –1	Tx ES-P is a 1-second period with one or more errored blocks or at least one defect (far end).
FeTxPSes	NUMBER(10)	DEFAULT –1	Tx path severely errored seconds (SES-P) is a 1-second period containing greater than 30 percent errored blocks or at least one defect (far end). SES is a subset of ES.
FeTxPBbe	NUMBER(10)	DEFAULT –1	Tx path background block error (BBE-P) is an errored block not occurring as part of an SES (far end).
FeTxPUas	NUMBER(10)	DEFAULT –1	Tx UAS-P is a count of 1-second intervals when the E1 path is unavailable on the transmit end of the signal (far end). The E1 path is unavailable when ten consecutive SESs occur. The ten SESs are included in unavailable time. After it becomes unavailable, the E1 path becomes available when ten consecutive seconds occur with no SESs. The ten seconds with no SESs are excluded from unavailable time.
FeTxPEsr	NUMBER(10)	DEFAULT –1	Tx ESR-P is the ratio of errored seconds to total seconds in available time during a fixed measurement interval (far end).
FeTxPSesr	NUMBER(10)	DEFAULT –1	Tx SESR-P is the ratio of SES to total seconds in available time during a fixed measurement interval (far end).
FeTxPBber	NUMBER(10)	DEFAULT –1	Tx BBER-P is the ratio of BBE to total blocks in available time during a fixed measurement interval (far end). The count of total blocks excludes all blocks during SESs.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.167 ONS15454SDH_E3_PM_Table

The ONS15454SDH_E3_PM_Table stores PM information about the E3 card in ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_E3_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-163: ONS15454SDH_E3_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
cvl	NUMBER(10)	DEFAULT –1	CV-L indicates the number of coding violations occurring on the line. This parameter is a BPVs and EXZs occurring over the accumulation period.
esl	NUMBER(10)	DEFAULT –1	ES-L is a count of the seconds containing one or more anomalies (BPV + EXZ) and/or defects (loss of signal) on the line.
sesl	NUMBER(10)	DEFAULT –1	SES-L is a count of the seconds containing more than a particular quantity of anomalies (BPV + EXZ > 44) and/or defects on the line.
lossl	NUMBER(10)	DEFAULT –1	Line loss-of-signal seconds (LOSS-L) is a count of 1-second intervals containing one or more LOS defects.
RxPes	NUMBER(10)	DEFAULT –1	Rx ES-P is a 1-second period with at least one defect.
RxPses	NUMBER(10)	DEFAULT –1	Rx SES-P is a 1-second period containing at least one defect. SES is a subset of ES.
RxPUas	NUMBER(10)	DEFAULT –1	Rx UAS-P is a count of the seconds when the path was unavailable. A path becomes unavailable when ten consecutive seconds occur that qualify as SES-Ps, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as SES-Ps.
RxPEsr	NUMBER(10)	DEFAULT –1	Rx ESR-P is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
RxPSesr	NUMBER(10)	DEFAULT –1	Rx SESR-P is the ratio of SES to total seconds in available time during a fixed measurement interval.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.168 ONS15454SDH_E4_PM_Table

The ONS15454SDH_E4_PM_Table stores PM information about the STM1E card that is configured as E4 in ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_E4_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-164: ONS15454SDH_E4_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
CodeViolations	NUMBER(10)	DEFAULT –1	Coding violation line.
ErrSecs	NUMBER(10)	DEFAULT –1	Errored seconds line.
SevErrSecs	NUMBER(10)	DEFAULT –1	Severely errored seconds line.
UnavailableSecs	NUMBER(10)	DEFAULT –1	Unavailable seconds line.
BackGroundBlockErrors	NUMBER(10)	DEFAULT –1	Background block errors line.
ESR	NUMBER(10)	DEFAULT –1	Errored seconds ratio line.
SESR	NUMBER(10)	DEFAULT –1	Severely errored seconds ratio line.
BBER	NUMBER(10)	DEFAULT –1	Background block errors ratio.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.169 ONS15454SDH_HO_PM_Table

The ONS15454SDH_HO_PM_Table stores high-order (HO) PM information about ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_HO_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-165: ONS15454SDH_HO_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
HpEb	NUMBER(10)	DEFAULT –1	HO path EB indicates that one or more bits are in error within a block.
HpEs	NUMBER(10)	DEFAULT –1	HO path ES is a 1-second period with one or more errored blocks or at least one defect.
HpSes	NUMBER(10)	DEFAULT –1	HO path SES is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
HpUas	NUMBER(10)	DEFAULT –1	HO path UAS is a count of the seconds when the VC path was unavailable. A low-order path becomes unavailable when ten consecutive seconds occur that qualify as HP-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as HP-SESs.
HpBbe	NUMBER(10)	DEFAULT –1	HO path BBE is an errored block not occurring as part of an SES.

Column Name	Oracle Data Type	Integrity Constraints	Description
HpEsr	NUMBER(10)	DEFAULT –1	HO path ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
HpSesr	NUMBER(10)	DEFAULT –1	HO path SESR is the ratio of SES to total seconds in available time during a fixed measurement interval.
HpBber	NUMBER(10)	DEFAULT –1	HO path BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
HpPpjcPdet	NUMBER(10)	DEFAULT –1	HP-PPJC-Pdet Multiplex Section, Positive Pointer Justification Count, Path Detected (MS-PPJC-Pdet) is a count of the positive pointer justifications detected on a particular path on an incoming SDH signal.
HpNpjcPdet	NUMBER(10)	DEFAULT –1	HP-NPJC-Pdet Multiplex Section, Negative Pointer Justification Count, Path Detected (MS-NPJC-Pdet) is a count of the negative pointer justifications detected on a particular path on an incoming SDH signal.
HpPpjcPGen	NUMBER(10)	DEFAULT –1	HP-PPJC-Pgen Multiplex Section, Positive Pointer Justification Count, Path Generated (MS-PPJC-Pgen) is a count of the positive pointer justifications generated for a particular path.
HpNpjcPGen	NUMBER(10)	DEFAULT –1	HP-NPJC-Pgen Multiplex Section, Negative Pointer Justification Count, Path Generated (MS-NPJC-Pgen) is a count of the negative pointer justifications generated for a particular path.
HpPjDiff	NUMBER(10)	DEFAULT –1	Indicates the sum of the absolute values of differences between positive transmitted and received, and negative transmitted and received. The important metric on pointer justification is not the exact counts, but how many were absorbed.
HpPjpSec	NUMBER(10)	DEFAULT –1	Indicates the number of Higher Order Path Pointer Justification Count Seconds detected on a particular path.
HpPjnSec	NUMBER(10)	DEFAULT –1	Indicates the number of Higher Order Path Pointer Justification Count Seconds generated for a particular path.

Column Name	Oracle Data Type	Integrity Constraints	Description
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeHpEb	NUMBER(10)	DEFAULT –1	Far end HO path EB indicates that one or more bits are in error within a block.
FeHpEs	NUMBER(10)	DEFAULT –1	Far end HO path ES is a 1-second period with one or more errored blocks or at least one defect.
FeHpSes	NUMBER(10)	DEFAULT –1	Far end HO path SES is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
FeHpUas	NUMBER(10)	DEFAULT –1	Far end HO path UAS is a count of the seconds when the VC path was unavailable. A low-order path becomes unavailable when ten consecutive seconds occur that qualify as HP-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as HP-SESs.
FeHpBbe	NUMBER(10)	DEFAULT –1	Far end HO path BBE is an errored block not occurring as part of an SES.
FeHpEsr	NUMBER(10)	DEFAULT –1	Far end HO path ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
FeHpSesr	NUMBER(10)	DEFAULT –1	Far end HO path SESR is the ratio of SES to total seconds in available time during a fixed measurement interval.
FeHpBber	NUMBER(10)	DEFAULT –1	Far end HO path BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.

Column Name	Oracle Data Type	Integrity Constraints	Description
StsPathWidth	NUMBER(4)	DEFAULT –1	Size of the STS. This is valid for the OC-N/STM-N cards. Possible values are: 1: VC4 2: VC4-2c 3: VC4-3c 4: VC4-4c 8: VC4-8c 16: VC4-16c 64: VC4-64c
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
HpFc	NUMBER(20)	DEFAULT –1	High-Order Path Failure Count.
FeHpfc	NUMBER(20)	DEFAULT –1	High-Order Path Failure Count (Far End).
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.170 ONS15454SDH_LO_PM_Table

The ONS15454SDH_LO_PM_Table stores low-order (LO) PM information about ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_LO_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-166: ONS15454SDH_LO_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
LoType	NUMBER(1)	—	LO type. Values are: 0: lo_vc3 1: lo_vc12 2: lo_vc11
LoEb	NUMBER(10)	DEFAULT –1	LO EB indicates that one or more bits are in error within a block.
LoBbe	NUMBER(10)	DEFAULT –1	LO BBE is an errored block not occurring as part of an SES.
LoEs	NUMBER(10)	DEFAULT –1	LO ES is a 1-second period with one or more errored blocks or at least one defect.
LoSes	NUMBER(10)	DEFAULT –1	LO SES is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
LoUas	NUMBER(10)	DEFAULT –1	LO UAS is a count of the seconds when the VC path was unavailable. A low-order path becomes unavailable when ten consecutive seconds occur that qualify as LP-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as LP-SESs.
LoEsr	NUMBER(10)	DEFAULT –1	LO ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
LoSesr	NUMBER(10)	DEFAULT –1	LO SESR is the ratio of SES to total seconds in available time during a fixed measurement interval.
LoBber	NUMBER(10)	DEFAULT –1	LO BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeLoEb	NUMBER(10)	DEFAULT –1	Far end LO EB indicates that one or more bits are in error within a block.
FeLoBbe	NUMBER(10)	DEFAULT –1	Far end LO BBE is an errored block not occurring as part of an SES.
FeLoEs	NUMBER(10)	DEFAULT –1	Far end LO ES is a 1-second period with one or more errored blocks or at least one defect.
FeLoSes	NUMBER(10)	DEFAULT –1	Far end LO SES is a 1-second period containing greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
FeLoUas	NUMBER(10)	DEFAULT –1	Far end LO UAS is a count of the seconds when the VC path was unavailable. A low-order path becomes unavailable when ten consecutive seconds occur that qualify as LP-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as LP-SESs.
FeLoEsr	NUMBER(10)	DEFAULT –1	Far end LO ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
FeLoSesr	NUMBER(10)	DEFAULT –1	Far end LO Severely Errored Second Ratio (SESR) is the ratio of SES to total seconds in available time during a fixed measurement interval.
FeLoBber	NUMBER(10)	DEFAULT –1	Far end LO Background Block Error Ratio (BBER) is the ratio of Background Block Errors (BBE) to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.

Column Name	Oracle Data Type	Integrity Constraints	Description
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.171 ONS15454SDH_MS_PM_Table

The ONS15454SDH_MS_PM_Table stores multiplex section (MS) PM information about ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_MS_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-167: ONS15454SDH_MS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)

Column Name	Oracle Data Type	Integrity Constraints	Description
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
MsEb	NUMBER(10)	DEFAULT –1	MS EB indicates that one or more bits are in error within a block.
MsEs	NUMBER(10)	DEFAULT –1	MS ES is a 1-second period with one or more errored blocks or at least one defect.
MsSes	NUMBER(10)	DEFAULT –1	MS SES is a 1-second period which contains greater than a specified percentage of errored blocks or at least one defect. SES is a subset of ES. For more information, refer to ITU-T G.829 Section 5.1.3.
MsUas	NUMBER(10)	DEFAULT –1	MS UAS is a count of the seconds when the section was unavailable. A section becomes unavailable when ten consecutive seconds occur that qualify as MS-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as MS-SESs. When the condition is entered, MS-SESs decrement and then count toward MS-UAS.
MsPpjcPdet	NUMBER(10)	DEFAULT –1	MS PPJC-Pdet is a count of the positive pointer justifications detected on a particular path on an incoming SDH signal.
MsNpjcPdet	NUMBER(10)	DEFAULT –1	MS NPJC-Pdet is a count of the negative pointer justifications detected on a particular path on an incoming SDH signal.
MsPpjcPGen	NUMBER(10)	DEFAULT –1	MS PPJC-Pgen is a count of the positive pointer justifications generated for a particular path.
MsNpjcPGen	NUMBER(10)	DEFAULT –1	MS NPJC-Pgen is a count of the negative pointer justifications generated for a particular path.

Column Name	Oracle Data Type	Integrity Constraints	Description
MsPsc	NUMBER(10)	DEFAULT –1	In a 1 + 1 protection scheme for a working card, MS PSC is a count of the number of times service switches from a working card to a protection card plus the number of times service switches back to the working card. For a protection card, MS-PSC is a count of the number of times service switches to a working card from a protection card plus the number of times service switches back to the protection card. The MS-PSC PM is only applicable if revertive line-level protection switching is used.
MsPsd	NUMBER(10)	DEFAULT –1	MS PSD applies to the length of time, in seconds, that service is carried on another line. For a working line, MS-PSD is a count of the number of seconds that service was carried on the protection line. For the protection line, MS-PSD is a count of the seconds that the line was used to carry service. The MS-PSD PM is only applicable if revertive line-level protection switching is used.
MSPscW	NUMBER(10)	DEFAULT –1	For a working line in a 2-fiber MS-SPRing, MS PSC-W is a count of the number of times traffic switches away from the working capacity in the failed line and back to the working capacity after the failure is cleared. PSC-W increments on the failed working line and PSC increments on the active protect line.
MSPsdW	NUMBER(10)	DEFAULT –1	For a working line in a 2-fiber MS-SPRing, MS PSD-W is a count of the number of seconds that service was carried on the protection line. MS-PSD-W increments on the failed working line and PSD increments on the active protect line.
MSPscS	NUMBER(10)	DEFAULT –1	In a 4-fiber MS-SPRing, MS PSC-S is a count of the number of times service switches from a working line to a protection line plus the number of times it switches back to the working line. A count is only incremented if span switching is used.
MSPsdS	NUMBER(10)	DEFAULT –1	In a 4-fiber MS-SPRing, MS PSD-S is a

Column Name	Oracle Data Type	Integrity Constraints	Description
			count of the seconds that the protection line was used to carry service. A count is only incremented if span switching is used.
MSPscR	NUMBER(10)	DEFAULT –1	In a 4-fiber MS-SPRing, MS PSC-R is a count of the number of times service switches from a working line to a protection line plus the number of times it switches back to a working line. A count is only incremented if ring switching is used.
MSPsdR	NUMBER(10)	DEFAULT –1	In a 4-fiber MS-SPRing, MS PSD-R is a count of the seconds that the protection line was used to carry service. A count is only incremented if ring switching is used.
MsBbe	NUMBER(10)	DEFAULT –1	MS BBE is a count of errored blocks not occurring as part of an SES.
MsFc	NUMBER(10)	DEFAULT –1	MS FC is a count of the number of failure events.
MsEsr	NUMBER(10)	DEFAULT –1	MS ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
MsSesr	NUMBER(10)	DEFAULT –1	MS SESR is the ratio of SES to total seconds in available time during a fixed measurement interval.
MsBber	NUMBER(10)	DEFAULT –1	MS BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
FeMsEb	NUMBER(10)	DEFAULT –1	Indicates that one or more bits are in error within a block (far end).
FeMsEs	NUMBER(10	DEFAULT –1	FE MS ES is a 1-second period with one or more errored blocks or at least one defect (far end).
FeMsSes	NUMBER(10)	DEFAULT –1	FE MS SES is a 1-second period which contains greater than a specified percentage of errored blocks or at least one defect (far end).

Column Name	Oracle Data Type	Integrity Constraints	Description
FeMsUas	NUMBER(10)	DEFAULT –1	FE MS UAS is a count of the seconds when the section was unavailable (far end). A section becomes unavailable when ten consecutive seconds occur that qualify as FE MS-SESs, and it continues to be unavailable until ten consecutive seconds occur that do not qualify as FE MS-SESs. When the condition is entered, FE MS-SESs decrement and then count toward FE MS-UAS.
FeMsBbe	NUMBER(10)	DEFAULT –1	FE MS BBE is a count of errored blocks not occurring as part of an SES (far end).
FeMsEsr	NUMBER(10)	DEFAULT –1	FE MS ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval (far end).
FeMsSesr	NUMBER(10)	DEFAULT –1	FE MS SESR is the ratio of SES to total seconds in available time during a fixed measurement interval (far end).
FeMsBber	NUMBER(10)	DEFAULT –1	FE MS BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval (far end). The count of total blocks excludes all blocks during SESs.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
FeMsfc	NUMBER(20)	DEFAULT –1	
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.172 ONS15454SDH_RS_PM_Table

The ONS15454SDH_RS_PM_Table stores regenerator section (RS) PM information about ONS 15454 SDH NEs.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS15454SDH_RS_PM_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-168: ONS15454SDH_RS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface; same as IfStaticIndex in the Interface_Table.
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Status of the PM bucket. Values are: 0: Valid period Other: Bit mask value (information about PM statistics)
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the 15-minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
RsEb	NUMBER(10)	DEFAULT –1	RS EB indicates that one or more bits are in error within a block.
RsEs	NUMBER(10)	DEFAULT –1	RS ES is a 1-second period with one or more errored blocks or at least one defect.
RsSes	NUMBER(10)	DEFAULT –1	RS SES is a 1-second period which contains greater than 30 percent errored blocks or at least one defect. SES is a subset of ES.
RsBbe	NUMBER(10)	DEFAULT –1	RS BBE is an errored block not occurring as part of an SES.
RsSefs	NUMBER(10)	DEFAULT –1	RS SEFS is a count of the number of SEFSs

Column Name	Oracle Data Type	Integrity Constraints	Description
			occurring during the interval.
RsEsr	NUMBER(10)	DEFAULT –1	RS ESR is the ratio of errored seconds to total seconds in available time during a fixed measurement interval.
RsSesr	NUMBER(10)	DEFAULT –1	RS SESR is the ratio of SES to total seconds in available time during a fixed measurement interval.
RsBber	NUMBER(10)	DEFAULT –1	RS BBER is the ratio of BBE to total blocks in available time during a fixed measurement interval. The count of total blocks excludes all blocks during SESs.
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
RsOfs	NUMBER(10)	DEFAULT –1	Regenerator section out-of-frame seconds.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.173 ONS155xx_CDL_Flow_Config

The ONS155xx_CDL_Flow_Config table provides information about the flows within a CDL-enabled interface for the ONS 155xx.

Primary key: CDLFLOW_CONFIG_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-169: ONS155xx_CDL_Flow_Config

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
CdlXmitFlowId	NUMBER(5)	—	Flow ID assigned to packet to transmit.
CdlRcvFlowId	NUMBER(5)	—	Flow ID assigned to packet to receive.

1.1.174 ONS155xx_CDL_IF_Config

The ONS155xx_CDL_IF_Config table provides information about the configuration of CDL on an ONS 15530 or ONS 15540 interface.

Primary key: CDLPE_CONFIG_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-170 : ONS155xx_CDL_IF_Config

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	IFDBAccessID in the MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
CdlAdminStatus	NUMBER(2)	—	Whether CDL is enabled or disabled. Values are: 1: Enabled 2: Disabled
CdlForceEndOfHop	NUMBER(2)	—	Whether end-of-hop is enabled. Values are: 1: Enabled 2: Disabled
CdlNodeBehavior	NUMBER(2)	—	Currently configured node behavior. Values are: 1: End of path 2: End of hop 3: CDL regenerator

1.1.175 ONS155xx_CDL_PM_Table

The ONS155xx_CDL_PM_Table contains CDL monitoring PM data.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155xx_CDL_PM_PK

Tablespace: ONS155xx_DATA_TABLESPACE

Table 1-171 : ONS155xx_CDL_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	—	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Indicates status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid PM disabled
InMaintenance	NUMBER(10)	—	Whether the NE was under maintenance when the PM bucket was collected.
ThresholdMask	NUMBER(20)	—	Threshold mask for the PM bucket in the interval. This does not apply to rxPower. Values are: 0: RxHEC 1: RxNonCDL 2: RxInvalidFlowID 3: NetEthernetCRC
RxHEC	NUMBER(20)	—	CDL header error count encountered in the receive direction. Applicable only to the Gigabit Ethernet or 10G interfaces that support CDL. Null for others.
RxNonCDL	NUMBER(20)	—	Counter associated with the number of packets received without a CDL header on this interface.

Column Name	Oracle Data Type	Integrity Constraints	Description
RxInvalidFlowID	NUMBER(20)	—	Counter associated with the number of CDL flow identifier lookup errors encountered in the receive direction.
NetEthernetCRC	NUMBER(20)	—	Counter provides the number of Ethernet packets with CRC errors.
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.

1.1.176 ONS155xx_Circuit_CC_Tbl

The ONS155xx_Circuit_CC_Tbl contains circuit-specific cross-connects for the ONS 155xx. For a given circuit, uniquely identified by CktCCDBId, there are one or more circuit cross-connects on a single NE. The CktCcNodeId gives information about the NE on which this cross-connects resides. When 1+1 protection is enabled on an NE, using either splitter or Y-cable, two cross-connects are typically associated with the circuit on the NE. In this case, two rows contain the same circuit and CktCcNodeId information. Each row in this table represents the circuit cross-connects in a single direction (that is, from source to destination). The isWorking and isActive parameters make sense in only one direction.

Primary key: ONS155XX_CIRCUIT_CC_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-172: ONS155xx_Circuit_CC_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktCCDBId	NUMBER(10)	NOT NULL	Unique database ID for each circuit cross-connect.
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
CktCCNodeId	NUMBER(10)	NOT NULL	Node ID of the cross-connect that is associated with the circuit.
CktCCSrcPhysicalLoc	NUMBER(20)	—	Physical location of the source endpoint of the cross-connect.
CktCCSrcIfIndex	NUMBER(10)	—	Static Prime Optical IfIndex of the source endpoint of the cross-connect.

Column Name	Oracle Data Type	Integrity Constraints	Description
CktCCSrcModuleType	NUMBER(8)	—	Prime Optical module type of the source endpoint of the cross-connect.
CktCCSrcObjectType	NUMBER(8)	—	Prime Optical object type of the source endpoint of the cross-connect.
CktCCSrcModelType	NUMBER(8)	—	Prime Optical model type.
CktCCDestPhysicalLoc	NUMBER(20)	—	Physical location of the destination endpoint of the cross-connect.
CktCCDestIfIndex	NUMBER(10)	—	Static Prime Optical IfIndex of the destination endpoint of the cross-connect.
CktCCDestModuleType	NUMBER(8)	—	Prime Optical module type of the destination endpoint of the cross-connect.
CktCCDestObjectType	NUMBER(8)	—	Object type of the destination endpoint of the cross-connect.
CktCCDestModelType	NUMBER(8)	—	Model type of the destination endpoint of the cross-connect.
CktCCSrcWavelength	VARCHAR2(64)	DEFAULT NULL	Wavelength on the source cross-connect side in nanometers (nm).
CktCCDestWavelength	VARCHAR2(64)	DEFAULT NULL	Wavelength on the destination cross-connect side in nm.
CktCCTechnology	NUMBER(1)	DEFAULT 0	Technology involved in this cross-connect. Values are: 0: UNKNOWN 1: DWDM_MUX_DEMUX_TRANSPARENT 2: DWDM_LAMBDA_SWITCHING 3: CISCO_CDL_SUBPORT_SWITCHING 4: DWDM_CLIENT_TRANSPARENT
CktCCSrcState	NUMBER(1)	DEFAULT 4	State of the source side of this cross-connect. Values are: 0: Inactive 1: Stranded 2: Routed 3: Tentative 4: Active
CktCCDestState	NUMBER(1)	DEFAULT 4	State of the destination side of this cross-connect. Values are: 0: Inactive 1: Stranded 2: Routed 3: Tentative 4: Active

Column Name	Oracle Data Type	Integrity Constraints	Description
IsCktCCWorking	NUMBER(1)	DEFAULT 1	Whether the circuit cross-connect is a working span. Values are: 0: False 1: True
IsCktCCActive	NUMBER(1)	DEFAULT 1	Whether the circuit span carries active traffic. Values are: 0: False 1: True
CktCCDescr	VARCHAR2(256)	—	Description associated with the cross-connect.
CktCCSrcCDLFlowID	NUMBER(3)	DEFAULT –1	CDL flow ID associated with the cross-connect source. Values are 0 to 254.
CktCCDestCDLFlowID	NUMBER(3)	DEFAULT –1	CDL flow ID associated with the cross-connect destination. Values are 0 to 254.

1.1.177 ONS155xx_Circuit_Path_Tbl

The ONS155xx_Circuit_Path_Tbl provides the association between an ONS 15501, ONS 15530, or ONS 15540 circuit and its individual paths.

Primary key: CIRCUIT_PATH_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-173: ONS155xx_Circuit_Path_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
CktNodeId	NUMBER(10)	NOT NULL	Node ID of the circuit's source node. This is an internal field.
CktUniqueId	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The CktNodeId and CktUniqueId combine to form a circuit ID that is unique in the network.
PATHDBID	NUMBER(10)	NOT NULL	Unique database ID for each unique path.
PathSeqNum	NUMBER(8)	NOT NULL	Sequence number associated with this circuit to path map.

1.1.178 ONS155xx_Circuit_Stack_Tbl

The ONS155xx_Circuit_Stack_Tbl contains circuit-to-circuit hierarchy and association for the ONS 155xx.

Primary key: ONS155XX_CIRCUIT_STACK_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-174: ONS155xx_Circuit_Stack_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
LLCktNodeId	NUMBER(10)	NOT NULL	Lower layer circuit node ID.
LLCktUniqueId	NUMBER(10)	NOT NULL	Lower layer circuit unique ID on the specified node ID.
HLCktNodeId	NUMBER(10)	NOT NULL	Higher layer circuit node ID.
HLCktUniqueId	NUMBER(10)	NOT NULL	Higher layer circuit unique ID on the CktNodeId.
LLCktPathDBID	NUMBER(10)	NOT NULL	Path database ID associated with the lower-layer circuit.
HLCktPathDBID	NUMBER(10)	NOT NULL	Path database ID associated with the higher-layer circuit.
LLToHLCktAssocType	NUMBER(1)	DEFAULT 0	Lower layer to higher layer circuit association type. Values are: 0: None 1: 1-to-1 2: Cisco CDL
LLToHLCktAssocTechInfo	VARCHAR2(64)	DEFAULT NULL	Lower layer to higher layer circuit association technology information.

1.1.179 ONS155xx_Condtype_Map

The ONS155xx_Condtype_Map table provides information about the mapping between vendor or notification type to Prime Optical condition type for the ONS 155xx.

Primary key: CONDTYPEMAP_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-175 : ONS155xx_Condtype_Map

Column Name	Oracle Data Type	Integrity Constraints	Description
NEModelType	NUMBER(10)	NOT NULL	Model type (not used currently).
DescrId	VARCHAR2(255)	NOT NULL	Vendor or notification ID.
CondType	NUMBER(5)	NOT NULL	Numerical condition type from notification.

1.1.180 ONS155xx_DCRP_Definition_Table

The ONS155xx_DCRP_Definition_Table stores the definition of all discrepancies supported within ONS15501, ONS 15530, or ONS 15540 management. The table is statically populated at the install time.

Primary key: ONS155XX_DCRP_DEF_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-176: ONS155xx_DCRP_Definition_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
DcrpTypeDBID	NUMBER(4)	NOT NULL	Unique sequence number generated by database; the link between this table and other tables that contain discrepancy data.
DcrpName	VARCHAR2(64)	NOT NULL	Unique name for the discrepancy.
DcrpSeverity	NUMBER(1)	DEFAULT 1	Severity for the discrepancy. Values are: 1: High 2: Medium 3: Low
DcrpProbCause	VARCHAR2(256)	DEFAULT NULL	Probable cause of the discrepancy.
DcrpRecAction	VARCHAR2(256)	DEFAULT NULL	Recommended action.

1.1.181 ONS155xx_DCRP_Table

The ONS155xx_DCRP_Table contains the discrepancy instances of each type for the ONS 155xx.

Primary key: ONS155XX_DCRP_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-177 : ONS155xx_DCRP_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
DcrpDBID	NUMBER(10)	NOT NULL	Unique ID generated by the database to identify a discrepancy.
DcrpTypeDBID	NUMBER(4)	NOT NULL	Discrepancy TypeID from Dcrp_Definition.
DcrpIdentifier	VARCHAR2(64)	DEFAULT	Identifier for a discrepancy of a

Column Name	Oracle Data Type	Integrity Constraints	Description
		NULL	particular type.
NEDBAccessID1	NUMBER(10)	—	ID of the NE where the discrepancy is detected.
NEDBAccessID2	NUMBER(10)	—	ID of the second NE involved in the discrepancy.
DcrpTimeStamp	DATE	—	Time when discrepancy was detected.
DcrpDescr	VARCHAR2(256)	DEFAULT NULL	Description of the discrepancy.
DcrpDetails	VARCHAR2(1024)	DEFAULT NULL	Any details about the discrepancy.

1.1.182 ONS155xx_Discovery_Info_Table

The ONS155xx_Discovery_Info_Table contains information about the main discovery and NE discovery for the ONS 155xx.

Primary key: ONS155XX_DISCOVERY_INFO_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-178: ONS155xx_Discovery_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
startTime	DATE	—	Starting time of the discovery.
endTime	DATE	—	Finishing time of the discovery.
status	NUMBER(1)	NOT NULL	Discovery status. Values are: 0: Idle 1: In progress 2: Completed 3: Failed
startMode	NUMBER(1)	NOT NULL	Discovery mode. Values are: 1: Scheduled 2: User-requested 3: Update-triggered
descr	VARCHAR2(256)	—	Description.

1.1.183 ONS155xx_Ether_Hist_PM_Table

The ONS155xx_Ether_Hist_PM_Table contains Ethernet RMON history PM data for the ONS 15501, ONS 15530, and ONS 15540.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155xx_Ether_Hist_PK

Tablespace: ONS155xx_DATA_TABLESPACE

Table 1-179: ONS155xx_Ether_Hist_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	NOT NULL	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Indicates status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid PM disabled
InMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
ThresholdMask	NUMBER(20)	—	Threshold mask for the PM bucket in the interval.
Octets	NUMBER(20)	—	Number of data octets received (including bad packets and FCS octets, but excluding framing bits).
Pkts	NUMBER(20)	—	Number of packets (including bad packets) received during this sampling interval.

Column Name	Oracle Data Type	Integrity Constraints	Description
CRCAlignErrors	NUMBER(20)	—	Number of packets received during this sampling interval that had a length (excluding framing FCS octets) between 64 and 1518 octets inclusive, but had either a bad frame check sequence (FCS) with an integral number of octets (FCS error) or a bad FCS with a nonintegral number of octets (alignment error).
UndersizePkts	NUMBER(20)	—	Number of packets received during this sampling interval that were less than 64 octets long (excluding framing bits but including FCS octets) and were otherwise well formed.
OversizePkts	NUMBER(20)	—	Number of packets received during this sampling interval that were longer than 1518 octets (excluding framing bits but including FCS octets) but were otherwise well formed.
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.

1.1.184 ONS155xx_FCM_PE_PM_Table

The ONS155xx_FCM_PE_PM_Table contains FC port error PM data for the ONS 15501, ONS 15530, and ONS 15540.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155xx_FCP_PE_PK

Tablespace: ONS155xx_DATA_TABLESPACE

Table 1-180: ONS155xx_FCM_PE_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.

Column Name	Oracle Data Type	Integrity Constraints	Description
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	—	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Indicates status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid PM disabled
InMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
ThresholdMaskInterval	NUMBER(20)	—	Threshold mask for the PM bucket in the interval.
RxLinkResets	NUMBER(20)	—	Number of link resets (LR primitive) received.
TxLinkResets	NUMBER(20)	—	Number of link resets (LR primitive) transmitted.
LinkResets	NUMBER(20)	—	Number of times the reset link protocol was initiated on this port. This include the number of loop initialization primitive events on an arbitrated loop.
RxOfflineSequences	NUMBER(20)	—	Number of offline primitive OLS received at this port.
TxOfflineSequences	NUMBER(20)	—	Number of offline primitive OLS transmitted by this port.
LinkFailures	NUMBER(20)	—	Number of link failures. This count is part of the link error status block (LESB) of FC-PH.
LossofSynchs	NUMBER(20)	—	Number of instances of synchronization loss detected at this port. This count is part of the LESB of FC-PH.
LossofSignals	NUMBER(20)	—	Number of instances of signal loss detected at this port. This count is part of the LESB of FC-PH.

Column Name	Oracle Data Type	Integrity Constraints	Description
PrimSeqProtocolErrors	NUMBER(20)	—	Number of primitive sequence protocol errors detected at this port. This count is part of the LESB of FC-PH.
InvalidTxWords	NUMBER(20)	—	Number of invalid transmission words received at this port. This count is part of the LESB of FC-PH.
InvalidCRCs	NUMBER(20)	—	Number of frames received with invalid CRC. This count is part of the LESB of FC-PH.
InvalidOrderedSets	NUMBER(20)	—	Number of invalid ordered sets received at this port.
FrameTooLongs	NUMBER(20)	—	Number of frames received at this port where the frame length was greater than what was agreed to in FLOGI/PLOGI.
TruncatedFrames	NUMBER(20)	—	Number of frames received at this port where the frame length was less than the minimum indicated by the frame header.
AddressErrors	NUMBER(20)	—	Number of frames received with unknown addressing.
DelimiterErrors	NUMBER(20)	—	Number of invalid frame delimiters received at this port.
EncodingDisparityErrors	NUMBER(20)	—	Number of encoding disparity errors received at this port.
OtherErrors	NUMBER(20)	—	Number of errors that were detected on this port but were not counted by any other error counter in this row.
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.
IfInOctets	NUMBER(20)	—	Received byte count.
IfInUcastPkts	NUMBER(20)	—	Received frame count.
IfOutOctets	NUMBER(20)	—	Transmitted byte count.
IfOutUcastPkts	NUMBER(20)	—	Transmitted frame count.

1.1.185 ONS155xx_Flash_Device_Table

The ONS155xx_Flash_Device_Table stores all the Flash devices present on an ONS 155xx. Only the devices which are physically present are stored in the table.

Primary key: ONS155XX_FLASH_DEVICE_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-181: ONS155xx_Flash_Device_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
FlashDevDBID	NUMBER(10)	NOT NULL	Unique sequence number generated by database. This is the link between this table and other tables that contain Flash partition information.
NEDBAccessID	NUMBER(10)	NOT NULL	Database ID of the NE containing the Flash device; same as NEDBAccessID in the NE_Info_Table.
FlashDevIndex	NUMBER(2)	NOT NULL	Flash device sequence number to index within the ciscoFlashDeviceTable.
FlashDevName	VARCHAR2(16)	NOT NULL	Flash device name that refers to the device within the NE.
FlashDevSize	NUMBER(10)	DEFAULT 0	Total size of the Flash device.
FlashDevPartitions	NUMBER(2)	DEFAULT 1	Number of Flash device partitions.
CPUSlotNum	NUMBER(2)	DEFAULT –1	Slot number of the CPU where the Flash device exists.
Status	NUMBER(2)	DEFAULT –1	Status of the Flash bank (active or inactive).

1.1.186 ONS155xx_Flash_Partition_Table

The ONS155xx_Flash_Partition_Table stores all the Flash partitions for each Flash device for a given ONS 155xx. Only the read/write Flash partitions are considered.

Primary key: ONS155XX_FLASHPART_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-182: ONS155xx_Flash_Partition_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
FlashPartDBID	NUMBER(10)	NOT NULL	Unique ID generated by database to identify a row in this table.
FlashDevDBID	NUMBER(10)	NOT NULL	ID of the Flash device.
FlashPartIndex	NUMBER(2)	NOT NULL	Flash partition sequence number to index within the ciscoFlashPartitionTable.

Column Name	Oracle Data Type	Integrity Constraints	Description
FlashPartName	VARCHAR(16)	NOT NULL	Flash device name to refer to the device within the NE.
FlashPartSize	NUMBER(10)	DEFAULT 0	Flash partition size.
FlashPartFreeSize	NUMBER(10)	DEFAULT 0	Free space within a Flash partition.
FlashPartStatus	NUMBER(1)	DEFAULT 3	Flag to indicate the read/write status of the partition. Values are: 1: Read-only 2: Run from Flash 3: Read/write
FlashPartFileNameLen	NUMBER(3)	DEFAULT 64	Maximum file name length supported by file system.

1.1.187 ONS155xx_GRA_CC_To_CKT_CC_Tbl

The ONS155xx_GRA_CC_To_CKT_CC_Tbl contains associations between granular cross-connects and circuit cross-connects in the ONS 155xx.

Primary key: ONS155XX_GRACCTOCKTCC_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-183: ONS155xx_GRA_CC_To_CKT_CC_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
GranularCCDBId	NUMBER(10)	NOT NULL	Granular cross-connect database ID.
CktCrossConnectDBId	NUMBER(10)	NOT NULL	Circuit cross-connect database ID.
GranularCCSeqNum	NUMBER(4)	DEFAULT –1	Granular cross-connect sequence number for the associated circuit cross-connect.

1.1.188 ONS155xx_Granular_CC_Tbl

The ONS155xx_Granular_CC_Tbl contains NE-specific granular cross-connects for the ONS 155xx.

Primary key: ONS155XX_GRANULAR_CC_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-184: ONS155xx_Granular_CC_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
GCCDBId	NUMBER(10)	NOT NULL	Unique ID for the internal cross-connect.
GCCSrcNodeId	NUMBER(10)	NOT NULL	Source NE ID on which this cross-connect resides.
GCCSrcPhysicalLoc	NUMBER(20)	—	Physical location of the source endpoint of the cross-connect.
GCCSrcIfIndex	NUMBER(10)	—	Static Prime Optical IfIndex of the source endpoint of the cross-connect.
GCCSrcModuleType	NUMBER(8)	—	Prime Optical module type of the source endpoint of the cross-connect.
GCCSrcObjectType	NUMBER(8)	—	Prime Optical object type of the source endpoint of the cross-connect.
GCCSrcModelType	NUMBER(8)	—	Prime Optical model type of the source endpoint of the cross-connect.
GCCDestNodeId	NUMBER(10)	NOT NULL	ID of the destination NE on which this cross-connect resides.
GCCDestPhysicalLoc	NUMBER(20)	—	Physical location of the destination endpoint of the cross-connect.
GCCDestIfIndex	NUMBER(10)	—	Static Prime Optical IfIndex of the destination endpoint of the cross-connect.
GCCDestModuleType	NUMBER(8)	—	Module type of the destination endpoint of the cross-connect.
GCCDestObjectType	NUMBER(8)	—	Object type of the destination endpoint of the cross-connect.
GCCDestModelType	NUMBER(8)	—	Model type of the destination endpoint of the cross-connect.
GCCProvisionType	NUMBER(1)	—	Type of internal cross-connect. Values are: 0: Unknown 1: Provisioned 2: Automatic 3: Dynamic
GCCTechnology	NUMBER(1)	DEFAULT 0	Internal cross-connect technology. Values are: 0: Unknown 1: Optical backplane cross-connect 2: Electrical backplane cross-

Column Name	Oracle Data Type	Integrity Constraints	Description
			connect 3: Cisco CDL mux/demux 4: Internal optical patch cross-connect 5: External optical patch cross-connect 6: Traffic aggregation 7: Splitter protection cross-connect 8: Optical DWDM mux/demux
GCCProtType	NUMBER(1)	DEFAULT 1	Protection type. Values are: 0: Unknown 1: No protection 2: Splitter protection 3: Y-cable or line card protection 4: Trunk card protection
GCCSrcToDestOperStatus	NUMBER(1)	DEFAULT 1	Status from source to destination. Values are: 0: Unknown 1: Active 2: Inactive
GCCDestToSrcOperStatus	NUMBER(1)	DEFAULT 1	Status from destination to source. Values are: 0: Unknown 1: Active 2: Inactive
GCCAssocLinkType	NUMBER(1)	DEFAULT 0	Associated link types. Values are: 0: Unknown 1: IF cross-connect 2: IF stack 3: Optical patch
GCCAssocLinkDBID	NUMBER(10)	—	Link database ID associated with this granular cross-connect, based on the link type.
GCCDirectionalType	NUMBER(1)	DEFAULT 1	Directional type for the granular cross-connect. Values are: 0: Unidirectional 1: Bidirectional
GCCDescr	VARCHAR2(256)	DEFAULT NULL	Description associated with the granular cross-connect.

1.1.189 ONS155xx_IF_Circuit_Params_Tbl

The ONS155xx_IF_Circuit_Params_Tbl provides the circuit name and description information configured by the user, in terms of circuit end-point interfaces, for the ONS 155xx.

Primary key: ONS155XX_IF_CKTNAME_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-185: ONS155xx_IF_Circuit_Params_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	NOT NULL	Source NE ID on which this cross-connect resides; same as NEDBAccessID in the NE_Info_Table.
IfPhysicalLoc	NUMBER(20)	NOT NULL	Physical location of the endpoint.
IfIndex	NUMBER(20)	NOT NULL	Static Prime Optical IfIndex of the endpoint.
IfModuleType	NUMBER(8)	NOT NULL	Module type of the endpoint.
IfObjectType	NUMBER(8)	NOT NULL	Object type of the endpoint.
IfModelType	NUMBER(8)	NOT NULL	Model type of the endpoint.
CktName	VARCHAR(64)	NOT NULL	Circuit name associated with the endpoint.
CktDescr	VARCHAR(256)	NOT NULL	Circuit description associated with the endpoint.

1.1.190 ONS155xx_IF_Optical_Power

The ONS155xx_IF_Optical_Power table contains the optical power for the ONS 15501, ONS 15530, and ONS 15540 wave (trunk) ports. The table also contains the object ID of the transparent ports corresponding to the wave interface, making it easy to use this information in the circuit path table (which deals with transponder client ports).

Primary key: ONS155XX_IF_OPTICAL_PWR_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-186: ONS155xx_IF_Optical_Power

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBACCESSID	NUMBER(10)	NOT NULL	NEDBACCESSID of the node containing the optical power interface; same as NEDBAccessID in the NE_Info_Table.

Column Name	Oracle Data Type	Integrity Constraints	Description
PwrSrcPhysicalLoc	NUMBER(20)	NOT NULL	Physical location of the optical power wave port.
PwrSrcIfIndex	NUMBER(20)	NOT NULL	Static Prime Optical IfIndex of the wave port.
PwrSrcModuleType	NUMBER(8)	NOT NULL	Module type of the trunk or wave port.
PwrSrcModelType	NUMBER(8)	NOT NULL	Model type of the wave port.
PwrSrcObjectType	NUMBER(10)	NOT NULL	Object type of the wave port
PwrMonDirection	NUMBER(1)	NOT NULL	Direction associated with the power value entry. Values are: 1: Receive 2: Transmit 3: Not applicable
PwrMonLocation	NUMBER(1)	NOT NULL	Location where power is monitored. Values are: 1: Before adjustment 2: After adjustment 3: Not applicable
OpticalPower	VARCHAR(32)	NOT NULL	Received optical power.

1.1.191 ONS155xx_IF_Phy_Info_Table

The ONS155xx_IF_Phy_Info_Table contains physical information about ONS 15501, ONS 15530, and ONS 15540 interfaces from cmPhyIfTable of Metro-PHY-MIB.

Primary key: ONS155XX_IF_PHY_INFO_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-187: ONS155xx_IF_Phy_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IFDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE interface; same as IFDBAccessId in MIB2_Interface_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_Info_Table.
IfMode	NUMBER(1)	DEFAULT –1	Mode of operation for client interfaces. Values are: 1: mode2R 2: mode3R

Column Name	Oracle Data Type	Integrity Constraints	Description
IfProtocol	NUMBER(2)	DEFAULT –1	Protocol to be monitored on client interfaces. Values are: 1: Other 2: Unknown 3: Gigabit Ethernet 4: 10-Gigabit Ethernet 5: Fiber channel 6: FICON 7: ESCON 8: SONET 9: SDH 10: ISC compatibility 11: ISC peer 12: Sysplextimeretr 13: Sysplextimerclo 14: Fast Ethernet 15: FDDI 16: E1 17: T1 18: E3 19: T3 20: DVB 21: SDI
IfClockRate	NUMBER(10)	DEFAULT –1	Clock rate across the client side.
IfMonitor	NUMBER(1)	DEFAULT –1	Whether monitoring is enabled. Values are: 1: True (enabled) 2: False (disabled)
IfLoopback	NUMBER(1)	DEFAULT –1	Loopback on client side interfaces. Values are: 1: No loop 2: Diagnostic loop 3: Line loop 4: Other loop
IfOFC	NUMBER(1)	DEFAULT –1	Whether open fiber control is enabled. Values are: 1: True (enabled) 2: False (disabled)
IfLSC	NUMBER(1)	DEFAULT –1	Whether laser safety control on the trunk side interfaces is enabled. Values are: 1: True (enabled) 2: False (disabled)

Column Name	Oracle Data Type	Integrity Constraints	Description
IfFLC	NUMBER(1)	DEFAULT –1	Enable/disable FLC on client/trunk interfaces. Values are: 1: True 2: False

1.1.192 ONS155xx_NE_Info_Table

The ONS155xx_NE_Info_Table contains additional ONS 15501, ONS 15530, and ONS 15540 information, mostly from the MIB II system group.

Primary key: ONS155XX_NE_INFO_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-188: ONS155xx_NE_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
HostName	VARCHAR2(64)	DEFAULT ''	Domain Name System (DNS) host name.
GlobalId	VARCHAR2(64)	DEFAULT ''	Global unique NE ID, such as chassis serial number.
SysObjectID	VARCHAR2(256)	DEFAULT ''	System object ID.
SysContact	VARCHAR2(256)	DEFAULT ''	System contact.
SysLocation	VARCHAR2(256)	DEFAULT ''	System location.
SysDescr	VARCHAR2(256)	DEFAULT ''	System description.
LastBootTime	DATE	—	Time stamp of the last restart.
LastSwitchoverTime	DATE	—	Time stamp of the last switchover.

1.1.193 ONS155xx_NE_Type_Table

The ONS155xx_NE_Type_Table provides the association between ONS 15501, ONS 15530, and ONS 15540 NE types and system object IDs.

Primary key: None

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-189: ONS155xx_NE_Type_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelType	NUMBER(4)	NOT NULL	ID to access the table for an NE; same as ModelType in the Model_Type_Table.
SysObjectID	VARCHAR2(256)	NOT NULL	System object ID.
imageVersion	VARCHAR2(64)	—	System image version.
SMFFactory	VARCHAR2(256)	—	Service Manager Function (SMF) factory.
Description	VARCHAR2(256)	—	Description.

1.1.194 ONS155xx_Opt_Power_PM_Table

The ONS155xx_Opt_Power_PM_Table provides historical optical power information about the ONS 155xx.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155XX_OPT_POWER_PM_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-190: ONS155xx_Opt_Power_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	—	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid—PM disabled
InMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.

Column Name	Oracle Data Type	Integrity Constraints	Description
ThresholdMask	NUMBER(20)	—	Threshold mask for the PM bucket in the interval. Values are: 1: Maximum power (second data column) 2: Minimum power (third data column) 4: Maximum ambient temperature (fifth data column) 5: Minimum ambient temperature (sixth data column) 7: Maximum laser temperature (eighth data column) 8: Minimum laser temperature (ninth data column) 10: Maximum bias current (eleventh data column) 11: Minimum bias current (twelfth data column)
MeanPower	NUMBER(10)	—	Mean/instance value of the optical power. The instance value is applicable only for ONS 15540 Extended Services Platform (ESP) Phase 0 and is taken from RxPower of Metro-Phy-MIB. For all other devices, only the mean value is applicable. Unit is in tenths of a decibel per milliwatt. The valid value range is –400 to 250. Value for nonapplicable interfaces: –1,000,000: Not applicable
MaxPower	NUMBER(10)	—	Maximum value of the optical power. Unit is in tenths of a decibel per milliwatt. The valid value range is –400 to 250. Value for nonapplicable interfaces: –1,000,000: Not applicable
MinPower	NUMBER(10)	—	Minimum value of the optical power. Unit is in tenths of a decibel per milliwatt. The valid value range is –400 to 250. Value for nonapplicable interfaces: –1,000,000: Not applicable
MeanAmbTemp	NUMBER(10)	—	Mean value of the ambient temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces:

Column Name	Oracle Data Type	Integrity Constraints	Description
			–1,000,000: Not applicable
MaxAmbTemp	NUMBER(10)	—	Maximum value of the ambient temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces: –1,000,000: Not applicable
MinAmbTemp	NUMBER(10)	—	Minimum value of the ambient temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces: –1,000,000: Not applicable
MeanLaserTemp	NUMBER(10)	—	Mean value of the laser temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces: –1,000,000: Not applicable
MaxLaserTemp	NUMBER(10)	—	Maximum value of the laser temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces: –1,000,000: Not applicable
MinLaserTemp	NUMBER(10)	—	Minimum value of the laser temperature. Unit is in tenths of a degree Celsius. The valid value range is –500 to 850. Value for nonapplicable interfaces: –1,000,000: Not applicable
MeanBiasCurrent	NUMBER(10)	—	Mean value of the bias current. Unit is in hundreds of microamperes. The valid value range is 0 to 10000. Value for nonapplicable interfaces: –1: Not applicable
MaxBiasCurrent	NUMBER(10)	—	Maximum value of the bias current. Unit is in hundreds of microamperes. The valid value range is 0 to 10000. Value for nonapplicable interfaces: –1: Not applicable

Column Name	Oracle Data Type	Integrity Constraints	Description
MinBiasCurrent	NUMBER(10)	—	Minimum value of the bias current. Unit is in hundreds of microamperes. The valid value range is 0 to 10000. Value for nonapplicable interfaces: –1: Not applicable
AvailableSecs	NUMBER(10)	—	Number of seconds in the interval when the PM data is valid. Values are 0 to 86400.
Direction	NUMBER(1)	NOT NULL	Direction in which the parameter is applicable. Values are: 1: Receive 2: Transmit 3: Not applicable
Location	NUMBER(1)	NOT NULL	Location of the amplifier or attenuator, if applicable. Values are: 1: Preadjustment 2: Postadjustment 3: Not applicable
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.

1.1.195 ONS155xx_Path_Hop_Tbl

The ONS155xx_Path_Hop_Tbl provides the association between an ONS 15501, ONS 15530, or ONS 15540 path and its individual hops in terms of circuit span and circuit cross-connects.

Primary key: PATH_HOP_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-191 ONS155xx_Path_Hop_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
PATHDBID	NUMBER(10)	NOT NULL	Unique database ID for each unique path.
PATHHOPID	NUMBER(2)	NOT NULL	Hop number for this path.
PATHHOPTYPE	NUMBER(1)	—	Type of hop. Values are: 1: Cross-connect 2: Segment
PATHHOPDBID	NUMBER(10)	NOT NULL	Database ID associated with this hop, based on type.

1.1.196 ONS155xx_Path_Tbl

The ONS155xx_Path_Tbl provides all ONS 15501, ONS 15530, and ONS 15540 paths and their protection information.

Primary key: ONS155XX_PATH_TBL_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-192 : ONS155xx_Path_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
PATHDBID	NUMBER(10)	NOT NULL	Unique database ID for each unique path.
PathSrcNodeId	NUMBER(10)	NOT NULL	Node ID of the source node of the path.
PathSrcPhysicalLoc	NUMBER(20)	—	Physical location of the source endpoint of the path.
PathSrcIfIndex	NUMBER(20)	—	Static Prime Optical IfIndex of the source endpoint of the path.
PathSrcModuleType	NUMBER(8)	—	Module type of the source endpoint of the path.
PathSrcObjectType	NUMBER(8)	—	Prime Optical object type of the source endpoint of the path.
PathSrcModelType	NUMBER(8)	—	Prime Optical model type of the source endpoint of the path.
PathDestNodeId	NUMBER(10)	NOT NULL	Node ID of the destination node of the path.
PathDestPhysicalLoc	NUMBER(20)	—	Physical location of the destination endpoint of the path.
PathDestIfIndex	NUMBER(10)	—	Static Prime Optical IfIndex of the destination endpoint of the path.
PathDestModuleType	NUMBER(8)	—	Module type of the destination endpoint of the path.
PathDestObjectType	NUMBER(8)	—	Prime Optical object type of the destination endpoint of the path.
PathDestModelType	NUMBER(8)	—	Prime Optical model type of the destination endpoint of the path.
PathStatus	NUMBER(1)	DEFAULT 1	Status of the path. Values are: 1: Active 2: Errored 3: Incomplete

Column Name	Oracle Data Type	Integrity Constraints	Description
IsWorking	NUMBER(1)	DEFAULT 1	Whether the entire path is configured to be working. Values are: 1: True 2: False
IsActive	NUMBER(1)	DEFAULT 1	Whether the entire path is configured to be active. Values are: 1: True 2: False
DownInterfaces	VARCHAR2(256)	DEFAULT NULL	List of all interfaces that are down in the path.
PathDescr	VARCHAR2(256)	DEFAULT NULL	Description associated with the path.
PathDestPwrMonNodeId	NUMBER(10)	DEFAULT –1	Node ID where the path destination power is monitored.
PathDestPwrMonPhysicalLoc	NUMBER(20)	DEFAULT –1	Physical location of the power monitor endpoint.
PathDestPwrMonIfIndex	NUMBER(20)	DEFAULT –1	Static Prime Optical IfIndex of the power monitor endpoint.
PathDestPwrMonModuleType	NUMBER(8)	DEFAULT –1	Module type of the power monitor endpoint.
PathDestPwrMonObjectType	NUMBER(8)	DEFAULT –1	Prime Optical object type of the power monitor endpoint.
PathDestPwrMonModelType	NUMBER(8)	—	Prime Optical model type of the power monitor endpoint.

1.1.197 ONS155xx_Phy_PM_Table

The ONS155xx_Phy_PM_Table contains ONS 15501, ONS 15530, and ONS 15540 physical PM data.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155XX_PHY_PM_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-193: ONS155xx_Phy_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	NOT NULL	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid—PM disabled
InMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
ThresholdMask	NUMBER(20)	—	Threshold mask for the PM bucket in the interval. This does not apply to receive power. Values are: 0: RxCVRD (first data column) 1: RxHEC (second data column) 2: RxCRC (third data column) 3: TxEncapFarEndPktErrors (fourth data column)
RxCVRD	NUMBER(20)	—	Code violations and running disparity errors encountered in the receive direction. Applicable to non-SONET client and wave interfaces only. Null for others.
RxCRC	NUMBER(20)	—	Counter associated with CRC errors. Applicable only for interfaces that provide CRC error monitoring. Null for others.
TxEncapFarEnd PktErrors	NUMBER(20)	—	Counter associated with ingress error indications from the far end of an Ethernet network, where an ESCON stream is encapsulated in Ethernet packets. Applicable only for ESCON interfaces where Ethernet encapsulation is performed. Null for others.
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.

Column Name	Oracle Data Type	Integrity Constraints	Description
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.

1.1.198 ONS155xx_SONET_Sec_PM_Table

The ONS155xx_SONET_Sec_PM_Table contains ONS 15530 and ONS 15540 SONET section layer PM data.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: ONS155XX_SONET_SEC_PM_PK

Tablespace: ONS155XX_DATA_TABLESPACE

Table 1-194: ONS155xx_SONET_Sec_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type to which this PM pertains.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Interface index as defined by ObjectMapper.
Interval	NUMBER(1)	—	Interval of the PM bucket. Values are: 0: 15 minutes 1: 1-day
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	NOT NULL	Status of the PM bucket. Values are: 1: Invalid—Not enough elapsed time 2: Valid period 3: Partial PM (enabled/disabled in this period) 4: Invalid—PM disabled
InMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
ThresholdMask	NUMBER(20)	—	Threshold mask for the PM bucket in the interval. This applies to code violations only. Values are: 3: Code violations (fourth data column)

Column Name	Oracle Data Type	Integrity Constraints	Description
SevErrSecs	NUMBER(10)	—	Number of severely errored seconds encountered by SONET/SDH section in the interval.
SevErrFramingSecs	NUMBER(10)	—	Number of severely errored framing seconds encountered by SONET/SDH section in the interval.
ErrSecs	NUMBER(10)	—	Number of errored seconds encountered by SONET/SDH section in the interval.
CodeViolations	NUMBER(12)	—	Number of code violations encountered by SONET/SDH section in the interval.
ModelType	NUMBER(5)	NOT NULL	Model type as defined by the object mapper.
ObjectType	NUMBER(10)	NOT NULL	Object type as defined by the object mapper.

1.1.199 ONS15600_ENET_PM_Table

The ONS 15600_ENET_PM_Table is used for ENET PM data where the module type is an ENET line card.

Primary key: ONS15600_ENET_PM_PK

Tablespace: ONS15600_DATA_TABLESPACE

Table 1-195: ONS15600_ENET_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type for which the TM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
Timestamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit mask Value
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.

Column Name	Oracle Data Type	Integrity Constraints	Description
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the fifteen minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
IfInOctets	NUMBER(20)	DEFAULT –1	—
IfInUcastPkts	NUMBER(20)	DEFAULT –1	—
IfInMulticastPkts	NUMBER(20)	DEFAULT –1	—
IfInBroadcastPkts	NUMBER(20)	DEFAULT –1	—
IfInDiscards	NUMBER(20)	DEFAULT –1	—
ifInErrors	NUMBER(20)	DEFAULT –1	—
IfOutOctets	NUMBER(20)	DEFAULT –1	—
IfOutUcastPkts	NUMBER(20)	DEFAULT –1	—
IfOutMulticastPkts	NUMBER(20)	DEFAULT –1	—
IfOutBroadcastPkts	NUMBER(20)	DEFAULT –1	—
IfOutDiscards	NUMBER(20)	DEFAULT –1	—
Dot3StatsAlignmentErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsFCSErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsFrameTooLong	NUMBER(20)	DEFAULT –1	—
EtherStatsUndersizePkts	NUMBER(20)	DEFAULT –1	—
EtherStatsFragments	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts64Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts65to127Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts128to255Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts256to511Octets	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
EtherStatsPkts512to1023Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts1024to1518Octets	NUMBER(20)	DEFAULT –1	—
EtherStatsBroadcastPkts	NUMBER(20)	DEFAULT –1	—
EtherStatsMulticastPkts	NUMBER(20)	DEFAULT –1	—
EtherStatsOversizePkts	NUMBER(20)	DEFAULT –1	—
EtherStatsJabbers	NUMBER(20)	DEFAULT –1	—
EtherStatsOctets	NUMBER(20)	DEFAULT –1	—
EtherStatsDropEvents	NUMBER(20)	DEFAULT –1	—
RxPauseFrames	NUMBER(20)	DEFAULT –1	—
TxPauseFrames	NUMBER(20)	DEFAULT –1	—
EtherStatsPkts	NUMBER(20)	DEFAULT –1	—
IfOutErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsInternalMacRxErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsInternalMacTxErrors	NUMBER(20)	DEFAULT –1	—
Dot3StatsSymbolErrors	NUMBER(20)	DEFAULT –1	—
RxEtherUtilizationStats	NUMBER(10)	DEFAULT –2	—
TxEtherUtilizationStats	NUMBER(10)	DEFAULT –2	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute

Column Name	Oracle Data Type	Integrity Constraints	Description
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.200 ONS15600_POS_PM_Table

The ONS15600_POS_PM_Table is used for POS PM data where the module type is an ASAP line card.

Primary key: ONS15600_POS_PM_PK

Tablespace: ONS15600_DATA_TABLESPACE

Table 1-196: ONS15600_POS_PM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE; same as NEDBAccessID in the NE_INFO_TABLE.
ModuleType	NUMBER(4)	NOT NULL	Module type for which the TM refers.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
NEIfIndex	NUMBER(20)	NOT NULL	Table index for the interface.
Timestamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.
NEIfPMStatus	NUMBER(20)	DEFAULT 0	Indicates status of the PM bucket. Values are: 0: Valid period Other: Bit mask Value

Column Name	Oracle Data Type	Integrity Constraints	Description
inMaintenance	NUMBER(1)	—	Whether the NE was under maintenance when the PM bucket was collected.
fifteenMinThresholdMask	NUMBER(10)	—	Threshold value for the PM bucket for the fifteen minute interval.
thresholdList	VARCHAR2(1024)	—	Threshold value for the PM bucket for the day.
hdlcInOctets	NUMBER(20)	DEFAULT –1	—
hdlcOutOctets	NUMBER(20)	DEFAULT –1	—
rxTotalPackets	NUMBER(20)	DEFAULT –1	—
txTotalPackets	NUMBER(20)	DEFAULT –1	—
hdlcRxAborts	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesBadCRC	NUMBER(20)	DEFAULT –1	—
GfpStatsRxFrame	NUMBER(20)	DEFAULT –1	—
GfpStatsTxFrame	NUMBER(20)	DEFAULT –1	—
GfpStatsRxOctets	NUMBER(20)	DEFAULT –1	—
GfpStatsTxOctets	NUMBER(20)	DEFAULT –1	—
GfpStatsRxCRCErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxMbitErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxSbitErrors	NUMBER(20)	DEFAULT –1	—
GfpStatsRxTypeInvalid	NUMBER(20)	DEFAULT –1	—
RxPktsDroppedInternalCongestion	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxShortPkts	NUMBER(20)	DEFAULT –1	—
MediaIndStatsRxFramesTruncated	NUMBER(20)	DEFAULT –1	—

Column Name	Oracle Data Type	Integrity Constraints	Description
MediaIndStats RxFramesTooLong	NUMBER(20)	DEFAULT –1	—
RxEtherUtilizationStats	NUMBER(10)	DEFAULT –2	—
TxEtherUtilizationStats	NUMBER(10)	DEFAULT –2	—
IS24H	NUMBER(1)	NOT NULL	Whether the PM data is collected in 24-hour or 15-minute increments. Values are: 1: 24 hour 0: 15-minute
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.
InvalidityList	VARCHAR2(1024)	—	Invalidity list for PM data.
StrCorbaTIDAID	VARCHAR2(512)	—	Stores the TL1 target identifier (TID) value returned by the getCTMHistoryPMData API for each physical termination point (PTP). The getCTMHistoryPMData API stores the historical PM data output in a .csv file. The PTP native name is one of the output parameters displayed in this file. It contains the rack, shelf, slot, and port information for the PTP. It also contains pluggable port module (PPM) information, if applicable.

1.1.201 ONS15600_Sonet_Line_PM_Table

This table has been deprecated.

1.1.202 ONS15600_Sonet_Path_PM_Table

This table has been deprecated.

1.1.203 ONS15600_Sonet_Sec_PM_Table

This table has been deprecated.

1.1.204 ONS15600SDH_HO_PM_Table

This table has been deprecated.

1.1.205 ONS15600SDH_MS_PM_Table

This table has been deprecated.

1.1.206 ONS15600SDH_RS_PM_Table

This table has been deprecated.

1.1.207 Operation_Permissions_Table

The Operation_Permission_Table contains information about the privilege level for each Prime Optical operation.

Primary key: Operation_Permission_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-197: Operation_Permissions_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserTypeID	NUMBER(4)	NOT NULL	Unique ID for each user.
OperationID	NUMBER(4)	NOT NULL	Unique ID for each operation.
Permission	NUMBER(1)	DEFAULT 2 NOT NULL	Permission for the user type for that operation. Values are: 0: No access 1: Read only 2: Read/Write

1.1.208 Operations_Table

The Operations_Table contains information about the different Prime Optical operations.

Primary key: Operations_Table_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-198: Operations_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
OperationID	NUMBER(4)	NOT NULL	Unique ID for each operation. Values are: 0: ADD_DELETE_NE_OR_GROUP 1: AUDIT_ERROR_LOG 2: AUDIT_TRAIL_TABLE 4: CIRCUIT_TABLE 5: CLI_INTERFACE 6: LOGGED_IN_CTM_USERS 7: CONTROL_PANEL 8: CTC_UPGRADE_TABLE 9: CTM_USER_PROFILES 10: CTM_USERS 11: EDIT_NE_OR_GROUP_PROPERTIES 12: EQUIPMENT_INVENTORY 14: FAULT_MGMT 15: JOB_MONITOR 16: LAUNCH_CRAFT_INTERFACE 17: LINK_TABLE 18: MANAGE_VLANS 19: NE_EXPLORER 20: NE_LOGGED_IN_USERS_TABLE 22: NE_SOFTWARE_TABLE 23: NE_SW_MEM_MGMT 24: NE_USER_ACCESS_ADMIN 26: ONS155XX_CM_DISCOVERY 27: ONS155XX_FLASH_FILES_TABLE 28: ONS155XX_SNMP_TABLE 31: PERFORMANCE 32: SUBNETWORK_EXPLORER 33: SUPPORTED_NE_TABLE 34: TL1_INTERFACE 35: TOPOLOGY_MODIFICATION 36: EDIT_DOMAIN_NODE_PROPERTIES 37: NE_SECURITY 38: USER_PREFERENCES 39: L2SERVICES_TABLE 40: L2TOPOLOGY_TABLE 41: LOGGED_IN_GWTL1_USERS 42: GWTL1_USERS 43: BLSR_TABLE 44: CREATE_BLSR 45: ROLL_CM_TABLE 46: CREATE_MSSP_RING 47: MSSP_RING_TABLE 48: NE_EXPLORER_ADMINISTRATION 49: CTC_USER_PROFILES

Column Name	Oracle Data Type	Integrity Constraints	Description
			50: SAVE_MAP_AS_DEFAULT 51: SHOW_CTM_EMS_ALARMS_EVENTS 52: NE_SNMP 53: NE_STATIC_ROUTES 54: NE_OSPF 55: NE_OSPF_AREA_RANGES 56: NE_OSPF_VIRTUAL_LINK 57: NE_FIREWALL 58: NE_IP_ADDRESS 59: NE_RIP 60: NE_DCC 61: SNTP_CONFIGURATION 62: L2DROPPORTS_TABLE 63: DWDM_RING_TABLE 64: MANAGE_IOS_USERS 65: L1_CIRCUIT_MGMT 66: DATA_ADMIN_FUNCTIONS 68: PROFILE_MGMT 69: L2_SERVICE_PROVISIONING 70: MERGE_CIRCUIT 71: NE_PROTECTION 72: COMPARE_CONFIG_FILES 73: TEMPLATE_CONFIG 74: CTM_USER_SET_CTCPWD 75: CONFIGURATION_CENTER 76: AUDIT_LOG_CONFIGURATION_CENTER 77: CHASSIS_VIEW 78: AUDIT_LOG_CHASSIS_VIEW 79: DIAGNOSTICS_CENTER 80: AUDIT_LOG_DIAGNOSTICS_CENTER 81: STATISTICS_REPORT 82: AUDIT_LOG_STATISTICS_REPORT 83: AUDIT_TRAIL 84: TELNET 85: SSH 86: CTM_FAILED_LOGIN_ATTEMPTS 87: LOGGED_IN_CORBA_USERS 88: CORBA_USERS 89: IOX_SOFTWARE_TABLE 90: ROUTER_USER_SETTINGS 91: MDS_FABRIC_MANAGER 92: MDS_CLI 93: IOX_IEP_CONFIG 94: IOX_MPLSTE_CONFIG

Column Name	Oracle Data Type	Integrity Constraints	Description
			95: IOX_IFCOMMON_CONFIG 96: IOX_IFETHER_CONFIG 97: IOX_IFPOS_CONFIG 98: IOX_IFSONET_CONFIG 99: IOX_ACL_CONFIG 100: IOX_PACKET_FILTER 101: IOX_QOS_CONFIG 102: IOX_ROUTINGPOLICY_CONFIG 103: IOX_BGP_CONFIG 104: IOX_OSPF_CONFIG 105: IOX_ISIS_CONFIG 106: IOX_LDP_CONFIG 107: IOX_RSVP_CONFIG 108: IOX_VRF_CONFIG 109: IOX_STATICROUTE_CONFIG 110: IOX_CONFIG_ROLLBACK 111: IOX_AAA_CONFIG 112: IOX_ALARM_ADMIN 113: IOX_USER_ADMIN 114: IOX_PMTHRESHOLD_CONFIG 115: IOX_PMSTATS_CONFIG 116: IOX_RACK_VIEW 117: IOX_TEMPLATE_CREATION 118: IOX_RESYNC 119: IOX_REDISCOVER 120: IOS7600_DOM_TABLE 121: TEMPLATE_TABLE 122: IOS7600_FLASH_FILES_TABLE 123: IOS_CLI_INTERFACE 124: SYSLOG VIEWER 125: IOX_SONETAPS_CONFIG 126: IOX_DWDM_CONFIG 127: APC_DOMAIN_TABLE 128: ALLOW_PARTIAL_UPGRADE 129: RESET_ML_CARDS 130: TNE_DEVICE_TABLE 132: SNMPv3_USERS 133: SNMPv3_MIB_VIEWS 134: SNMPv3_GROUP_ACCESS 135: SNMPv3_TRAP_DESTINATIONS 136: SNMPv3_NOTIFICATION_FILTERS 137: SNMPv3_PROXY_FORWARDER 138: SNMPv3_PROXY_TRAP_FORWARDER 139: SNMPv3_REMOTE_USERS 140: SNMPv3_CONFIGURATION

Column Name	Oracle Data Type	Integrity Constraints	Description
			141: OSS_SNMPV3_USERS 142: CONFIGURATION_CENTER_FORCED
OperationName	VARCHAR2(65)	NOT NULL	Name of the operation.
Category	VARCHAR2(65)	—	Category name corresponding to operation.
Description	VARCHAR2(256)	—	Description of operation.
AllowedPermissions	NUMBER(1)	DEFAULT 2	Permission that is allowed for the user type. Values are: 0: No Read 1: No Write 2: All 3: No Access 4: None
Warning	VARCHAR2(256)	—	Warnings associated with dependencies on the permissions for this operation.

1.1.209 OSS_Active_User_Table

The OSS_Active_User_Table contains information about the active OSS users.

Primary key: OSS_ACTIVE_USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-199: OSS_Active_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SessionID	NUMBER(10)	NOT NULL UNIQUE	Session ID.
OSSName	VARCHAR2(128)	NOT NULL	OSS name.
IS_IPV6	NUMBER(1)	—	Flag indicating IP version. Values are: 0: IPv4 1: IPv6
TimeOfConnection	DATE	—	Date and time when session was started.

Column Name	Oracle Data Type	Integrity Constraints	Description
ListenerModeFlag	NUMBER(1)	DEFAULT 1	Whether autonomous message should be forwarded to the OSS connection. Values are: 0: DO_NOT_FORWARD 1: FORWARD
DomainMgrFlag	NUMBER(1)	DEFAULT 0	Whether the OSS can use the Domain Manager capabilities of Prime Optical. Values are: 0: No 1: Yes
IPV6Addr	RAW(16)	NOT NULL	IPV6 address in IPV6 format.
IS_IPV6	NUMBER(1)	DEFAULT 0	Values are: 0: IPv4 1: IPv6

1.1.210 OSS_AID_ENUM_Table

The OSS_AID_ENUM_Table contains information about the access identifiers (AIDs) that Prime Optical GateWay/TL1 supports for autonomous message filtering. Prime Optical GateWay/TL1 filters autonomous messages based on the AID values in this table.

Primary key: OSS_AID_ENUM_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-200: OSS_AID_ENUM_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
AIDId	NUMBER(4)	NOT NULL	Access identifier of the corresponding AIDString column.
AIDString	VARCHAR2(32)	NOT NULL UNIQUE	Represents the data that makes up the AID deny list for event forwarding discriminator (EFD) profiles.

1.1.211 OSS_Alarm_Filter_Table

The OSS_Alarm_Filter_Table contains information about EFDs and their characteristics (such as whether or not event forwarding is included in the EFD profile). The table contains 12 default entries.

Primary key: OSS_ALARM_FILTER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-201: OSS_Alarm_Filter_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
AlarmFilterId	NUMBER(4)	NOT NULL	Unique alarm filter ID for this table.
FilterName	VARCHAR2(128)	NOT NULL UNIQUE	Filter profile name.
AlarmSeverity	NUMBER(1)	DEFAULT 0	Minimum severity of alarms that are forwarded to the OSS. Values are: 1: Not reported alarm clear 2: Not reported alarm declared 3: Not alarmed 4: Warning 5: Minor 6: Major 7: Critical
EventReportingFlag	NUMBER(1)	DEFAULT 0	Whether events should be forwarded to the Operations Support System (OSS). Values are: 0: No 1: Yes
AlarmReportingFlag	NUMBER(1)	DEFAULT 1	Whether alarms should be forwarded on the OSS connection. Values are: 0: Disabled 1: Enabled
PMReportingFlag	NUMBER(1)	DEFAULT 0	Whether PM data should be forwarded on the OSS connection. Values are: 0: Disabled 1: Enabled

1.1.212 OSS_Alarm_Fltr_AID_Deny_Tbl

The OSS_Alarm_Fltr_AID_Deny_Tbl contains information about which AIDs are denied for specific EFD profiles. If one EFD profile denies more than one AID, the table contains more than one entry for that EFD profile. The AID ID corresponds to the AID stored in the OSS_AID_ENUM_Table.

Primary key: OSS_ALARM_FLTR_AID_DENY_TBL_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-202: OSS_Alarm_Fltr_AID_Deny_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
AlarmFilterId	NUMBER(4)	NOT NULL	ID of the alarm filter that is listed in the OSS_Alarm_Filter_Table.
AidID	NUMBER(4)	NOT NULL	Values are: 0: DS1 1: EC1 2: OC12 3: OC3 4: OC48 5: OC192 6: STS1 7: STS12C 8: STS3C 9: STS48C 10: T1 11: T3 12: VT1 13: CHASSIS 14: ENET 15: BITS 16: ENV 17: SYNCN 18: ALL

1.1.213 OSS_Alarm_Fltr_TID_Deny_Tbl

The OSS_Alarm_Fltr_TID_Deny_Tbl contains information about alarm filters based on NE or target identifier (TID).

Primary key: OSS_ALARM_FLTR_TID_DENY_TBL_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-203: OSS_Alarm_Fltr_TID_Deny_Tbl

Column Name	Oracle Data Type	Integrity Constraints	Description
AlarmFilterId	NUMBER(4)	NOT NULL	ID of the alarm filter that is listed in the OSS_Alarm_Filter_Table.
NEDBAccessID	NUMBER(10)	NOT NULL	NE database ID of the TID that is blocked for the corresponding alarm filter ID column; same as NEDBAccessID in the NE_Info_Table.

1.1.214 OSS_CORBA_User_Table

The OSS_CORBA_User_Table contains information about OSS users configured in Prime Optical and used by Prime Optical GateWay/CORBA.

Primary key: OSS_CORBA_USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-204: OSS_CORBA_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IPAddress	NUMBER(10)	DEFAULT 0	IP address.
OSSName	VARCHAR2(128)	NOT NULL UNIQUE	Name that the OSS uses to log into Prime Optical GateWay/CORBA.
OSSPasswd	VARCHAR2(128)	NOT NULL	Encrypted password that the OSS uses to log into Prime Optical GateWay/CORBA.

1.1.215 OSS_User_Table

The OSS_User_Table contains information about OSS users configured in Prime Optical and used by Prime Optical GateWay/TL1.

Primary key: OSS_USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-205: OSS_User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
IS_IPV6	NUMBER(1)	NOT NULL	Indicates IP version. Values are: 0: IPv4 1: IPv6
OSSName	VARCHAR2(128)	NOT NULL UNIQUE	OSS name.
OSSPasswd	VARCHAR2(128)	NOT NULL	OSS password.
ListenerModeFlag	NUMBER(1)	DEFAULT 1	Whether autonomous messages should be forwarded on the OSS connection. Values are: 0: Do not forward 1: Forward

Column Name	Oracle Data Type	Integrity Constraints	Description
ActUserDenyFlag	NUMBER(1)	DEFAULT 1	Whether ACT_USER failures should be answered with a DENY or silently discarded. Values are: 0: Discard 1: Send deny
AlarmFilterFlag	NUMBER(1)	DEFAULT 0	Whether the filter is in effect. Values are: 0: No 1: Yes
AlarmFilterId	NUMBER(4)	DEFAULT 0	Alarm filter associated with this user.
DomainMgrFlag	NUMBER(1)	DEFAULT 0	Whether the OSS can use the Domain Manager capabilities of the Element Management System (EMS). Values are: 0: No 1: Yes
IPV6Addr	RAW(16)	NOT NULL	IPV6 address in IPV6 format.
IS_IPV6	NUMBER(1)	DEFAULT 0	Values are: 0: IPv4 1: IPv6

1.1.216 Password_History_Table

The Password_History_Table contains a configurable number of previously-used enduser passwords. This is a security precaution intended to limit password reuse.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-206: Password_History_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserID	NUMBER(4)	NOT NULL	User ID.
UserPassword	RAW(256)	—	User password.
Timestamp	DATE	—	Time stamp.

1.1.217 PatchCord_Table

The PatchCord_Table is used to store the virtual link ID of a patchcord. When a new patchcord link is added in the Link_Table, a new row is added in the PatchCord_Table having the same link ID and one virtual link ID for each endpoint. A row in the PatchCord_Table is deleted when the corresponding link is deleted from the Link_Table.

Primary key: LINKID_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-207: PatchCord_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
LinkID	NUMBER(8)	NOT NULL	Unique number created by the database.
VirtualLinkIdSrc	NUMBER(10)	DEFAULT –1	Source virtual link identifier.
VirtualLinkIdDst	NUMBER(10)	DEFAULT –1	Destination virtual link identifier.

1.1.218 PM_ATTR_Value_Mapping_Table

The PM_ATTR_Value_Mapping_Table is used to map real-time polling result values to different display values.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-208: PM_ATTR_Value_Mapping_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
PMViewName	VARCHAR2(64)	NOT NULL	Specifies the PM table.
ATTRIndex	NUMBER(5)	NOT NULL	Index from ObjectAttribute table.
ATTRValue	VARCHAR2(64)	NOT NULL	Returned attribute value from the real-time PM query.
DisplayText	VARCHAR2(64)	NOT NULL	Text displayed in PM columns.

1.1.219 PM_ATTRID_Mapping_Table

The PM_ATTRID_Mapping_Table contains mapping information about the retrieval of real-time PM data from the NE.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-209: PM_ATTRID_Mapping_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
PMViewName	VARCHAR2(64)	NOT NULL	Specifies the PM table.
PMName	VARCHAR2(256)	NOT NULL	PM parameter name.
MetadataIndex	NUMBER(5)	NOT NULL	Index from ObjectAttribute Table.
ATTRIndex	NUMBER(5)	NOT NULL	Index from ObjectAttribute Table.
Validity_AttrIndex	NUMBER(5)	—	Attribute ID for validity attributes.
IS24H	NUMBER(1)	NOT NULL	Values are: –1: N/A 0: 15 minutes 1: 24 hour
DisplayText	VARCHAR2(64)	—	Text displayed in PM columns.
PMPARAMID	NUMBER(5)	—	PM parameter ID.

1.1.220 PM_Category_View_Table

The PM_Category_View_Table contains mapping information between the NE model and the PM category view.

Primary key: PM_Category_View_Table_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-210: PM_Category_View_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
MODELTYPE	NUMBER(4)	—	Model type.
CATEGORYVIEWNAME	VARCHAR(256)	—	PM category view name.
NEMODELVIEWNAME	VARCHAR(256)	—	NE model view name.

1.1.221 PM_Cerent_Info_Table

The PM_Cerent_Info_Table contains mapping information that associates each PM table column name to a unique identifier.

Primary key: PM_Cerent_Info_Table_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-211: PM_Cerent_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
PMParamID	NUMBER(10)	—	—
PMTableName	VARCHAR2(256)	NOT NULL	PM table name.
PMTableParam	VARCHAR2(256)	NOT NULL	PM table parameter name.
PMTableColNo	NUMBER(10)	—	Table column number.
DBColType	VARCHAR(64)	—	Database column type.
DBColSize	NUMBER(10)	—	Database column size.
TMFNAME	VARCHAR2(64)	DEFAULT N/A	TMF PM parameter name specified in the Prime Optical GateWay/CORBA implementation agreement.
LOCATION	NUMBER(2)	DEFAULT 0	Near-end or far-end location information.
DIRECTION	NUMBER(2)	DEFAULT 0	Receive or transmit direction information.

1.1.222 PM_Historical_Appointments

The PM_Historical_Appointments table contains information about the PM data collection schedule.

For medium, large, and high-end configurations, this table is partitioned.

Primary key: PM_HISTORICAL_APPOINTMENTS_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-212: PM_Historical_Appointments

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDbaccessid	NUMBER(10)	NOT NULL	NEDBaccessid of the purged NE; same as NEDBaccessID in the NE_Info_Table.
IntervalType	NUMBER(1)	NOT NULL	Types of intervals. Values are: 0: 15 minutes 1: 24 hours
OriginalScheduledTime	DATE	NOT NULL	Originally scheduled PM collection time.

Column Name	Oracle Data Type	Integrity Constraints	Description
Status	NUMBER(1)	NOT NULL	Values are: 0: Success 1: Failure 2: In Process 3: Pending
TimeStamp	DATE	NOT NULL	Time stamp that shows the interval at which the set of statistics was stored.

1.1.223 PM_Metadata_Info_Table

The PM_Metadata_Info_Table contains metadata for each possible PM category (such as section, line, path, and so on).

Primary key: PM_Metadata_Info_Table_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-213: PM_Metadata_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEModel	NUMBER(10)	NOT NULL	NE model.
NEModelIndex	NUMBER(10)	NOT NULL	Model index of the NE.
ModuleType	NUMBER(10)	NOT NULL	Module type.
InterfaceType	VARCHAR2(64)	NOT NULL	Interface type.
BaseNEModelIndex	NUMBER(10)	NOT NULL	Base NE model index of the inherited PM metadata.
BaseModuleType	NUMBER(10)	NOT NULL	Base module type of the inherited PM metadata.

1.1.224 PM_Metadata_Table

The PM_Metadata_Table contains metadata for each possible PM category (such as section, line, path, and so on).

Primary key: PM_Metadata_Table_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-214: PM_Metadata_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEModel	NUMBER(10)	NOT NULL	NE model.
NEModelIndex	NUMBER(10)	NOT NULL	NE model index.
ModuleType	NUMBER(10)	NOT NULL	Module type.
PMCategory	VARCHAR2(64)	NOT NULL	PM category.
OBJECTTYPEID	NUMBER(10)	NOT NULL	Object type ID.
PMTableColNo	NUMBER(10)	—	PM table column number.
PMParamID	NUMBER(10)	NOT NULL	Sequence ID.
InterfaceType	VARCHAR2(64)	NOT NULL	Interface type.
ParamPosition	NUMBER(10)	NOT NULL	Bulk PM parameter position.
ValidityPosition	NUMBER(10)	NOT NULL	Bulk PM validity position.
TCANumber	NUMBER(10)	—	TCA number.
TCALocation	VARCHAR(10)	—	TCA location.
Direction	VARCHAR(10)	DEFAULT 'N/A'	Line direction.

1.1.225 Port_Channel_Table

The Port_Channel_Table contains information about port channel interfaces for ML-series cards.

Primary key: PORT_CHANNEL_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-215: Port_Channel_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NE_NODE_ID	NUMBER(10,0)	—	The identification of the NE containing the ML card and port channel.
SLOT_NUMBER	NUMBER(2,0)	NOT NULL	The slot number.
PORT_CHANNEL_ID	NUMBER(2,0)	NOT NULL	The port channel ID identifies the port channel interface. Prime Optical generates the ID automatically; the range is from 1 to 64.
ADMIN_STATE	NUMBER(1,0)	—	The status of the port channel interface. Values are: 0: Down 1: Up

Column Name	Oracle Data Type	Integrity Constraints	Description
MTU	NUMBER(4,0)	DEFAULT 1500	The maximum transmission unit (MTU) size, which is the largest acceptable packet size configured for the port channel interface. The range is from 64 to 9000 bytes.
HOLD_QUEUE_IN	NUMBER(4,0)	—	The size of the memory packets input queue. Values are: –1: input queue size determined automatically; or 0 to 4096.
HOLD_QUEUE_OUT	NUMBER(4,0)	—	The size of the memory packets output queue. Values are: –1: output queue size determined automatically; or 0 to 4096.

1.1.226 Port_Channel_Members_Table

The Port_Channel_Members_Table is used to add members to the port channel.

Primary key: PORT_CHANNEL_MEMBERS_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-216: Port_Channel_Members_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NE_NODE_ID	NUMBER(10,0)	—	The identification of the NE containing the ML card and port channel.
SLOT_NUMBER	NUMBER(2,0)	NOT NULL	The slot number.
PORT_CHANNEL_ID	NUMBER(2,0)	NOT NULL	The port channel identification. Values are 1 to 64.
ETH_PORT_NUMBER	NUMBER(2,0)	—	The Ethernet front port containing the port channel.
LACP_MODE	NUMBER(1,0)	—	The mode for Link Aggregation Control Protocol (LACP) packet exchange between ports. Values are: 0: Transparent; LACP is not enabled 1: Active negotiating mode 2: Passive negotiating mode
LACP_PORT_PRIORITY	NUMBER(5,0)	—	The LACP port priority. Values are: –1: Automatic; or 0 to 65535.

1.1.227 Proxy_Server_Table

The Proxy_Server_Table contains information about the various proxy servers in the network.

Primary key: PROXY_SERVER_TABLE_PK

Tablespace: ONS15454_DATA_TABLESPACE

Table 1-217: Proxy_Server_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEIPAddr	NUMBER(10)	NOT NULL	IP address of the NE.
GNEIPAddr	VARCHAR(64)	NOT NULL	IP address of the GNE.
ISPROXYINGGNE	NUMBER(1)	NOT NULL	In the case of multiple GNEs in a ring, acts as the GNE proxy for the ENE(s).
MaskLength	NUMBER(4)	DEFAULT 0	Specifies if this is the host or subnet.
Cost	NUMBER(10)	DEFAULT 0	Routing cost for this client.
Flags	NUMBER(4)	DEFAULT 0	Routing flags for this client.
GNEID	NUMBER(8)	DEFAULT 0	The GNE identification.

1.1.228 PW_EFP

The PW_EFP stores the information that describes the Ethernet Flow Point (EFP) for each pseudowire cross-connect contributor.

Primary key: PW_EFP_PK

Table 1-218: PW_EFP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
SERSTATE	NUMBER(1)	NOT NULL	EFP service status.
ISDOUBLETAGGED	NUMBER(1)	NOT NULL	EFP tagging status. Values are: 0: single-tagged 1: double-tagged
ISEXACT	NUMBER(1)	NOT NULL	Whether the exact EFP option is enabled or disabled. Values are: 0: Disabled 1: Enabled

Column Name	Oracle Data Type	Integrity Constraints	Description
ISSTATSENABLED	NUMBER(1)	NOT NULL	Whether statistics on EFP are enabled or disabled. Values are: 0: Disabled 1: Enabled
PORTDIRECTION	NUMBER(1)	NOT NULL	Direction of the port.
PRIMARYLOADBAL	NUMBER(10)	NOT NULL	Primary load balancing value has been provisioned.
OUTVLANTAG	VARCHAR2(256)	NOT NULL	Outer VLAN ID tag has been provisioned.
OUTVLANTPID	NUMBER(10)	NOT NULL	Outer VLAN TP ID has been provisioned.
INVLANTAG	VARCHAR2(256)	NOT NULL	Inner VLAN ID tag has been provisioned.
INVLANTPID	NUMBER(10)	NOT NULL	Inner VLAN TP ID has been provisioned.
RWOUTVLANTAG	VARCHAR2(256)	NOT NULL	Rewrite outer VLAN ID tag has been provisioned.
RWOUTVLANTPID	NUMBER(10)	NOT NULL	Rewrite outer VLAN TP ID has been provisioned.
RWINVLANTAG	VARCHAR2(256)	NOT NULL	Rewrite inner VLAN ID tag has been provisioned.
RWINVLANTPID	NUMBER(10)	NOT NULL	Rewrite inner VLAN TP ID has been provisioned.
RWOPERATION	NUMBER(10)	NOT NULL	Rewrite operation has been provisioned.
ISSYMMETRICENABLED	NUMBER(1)	NOT NULL	Whether the EFP symmetric option is enabled or disabled. Values are: 0: Disabled 1: Enabled

1.1.229 PW_EP

The PW_EP describes the endpoints related to the pseudowire services.

Primary key: PW_EP_PK

Table 1-219: PW_EP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
CTCINDEX	NUMBER(10)	NOT NULL	Physical object ID managed by the NE.

Column Name	Oracle Data Type	Integrity Constraints	Description
			This is an internal field.
PHYLOC	NUMBER(20)	NOT NULL	Physical object ID managed by the Prime Optical system. This is an internal field.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical ID used to identify the contributor in the service. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
ROUTEID	VARCHAR2(256)	NOT NULL	The route IP address.
ISPROTECTED	NUMBER(1)	NOT NULL	Endpoint protection. Values are: 0: Unprotected 1: Protected
ENABLEDELAY	NUMBER(10)	NOT NULL	Enabled delay value has been provisioned.
ISNEVERDISABLE	NUMBER(1)	NOT NULL	Whether the never disable option is enabled or disabled. Values are: 0: Disabled 1: Enabled
DISABLEDELAY	NUMBER(10)	NOT NULL	Disabled delay value has been provisioned.
TINFOTYPE	NUMBER(10)	NOT NULL	Endpoint type has been provisioned.
OPERSTATE	NUMBER(1)	NOT NULL	Operation status of the endpoint.
IFPORTINDEX	NUMBER(10)	NOT NULL	Physical object ID managed by the NE.
ADMINSTATE	NUMBER(1)	NOT NULL	Administrative status of the endpoint.
RXBW	NUMBER(20)	NOT NULL	Transmission bandwidth value has been provisioned.
TXBW	NUMBER(20)	NOT NULL	Receiver bandwidth value has been provisioned.
DESCRIPTION	VARCHAR2(256)	—	Description of the service has been provisioned.
SLOTMODULETYPE	NUMBER(10)	NOT NULL	Slot module type. This is an internal field.
PORTMODULETYPE	NUMBER(10)	NOT NULL	Port module type. This is an internal field.

1.1.230 PW_PP

The PW_PP stores the information related to the path point contributors for the pseudowire services.

Primary key: PW_PP_PK

Table 1-220: PW_PP

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBID	NUMBER(10)	NOT NULL	NE ID. This is an internal field.
PTSINDEX	NUMBER(10)	NOT NULL	PTS ID. This is an internal field.
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
WORKPROT	NUMBER(10)	NOT NULL	Whether the path is working or protected: 0: Working path 1: Protected path
CTCINDEX	NUMBER(10)	NOT NULL	Physical object ID managed by the NE. This is an internal field.
PHYLOC	NUMBER(20)	NOT NULL	Physical object ID managed by the Prime Optical system. This is an internal field.
TUNNELNUM	NUMBER(6)	NOT NULL	Tunnel number used for routing.
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical ID used to identify the contributor in the service. This is an internal field.
LINKID	NUMBER(10)	NOT NULL	Link ID used for routing.
SERVICENAME	VARCHAR2(256)	—	Service name.
PEERID	VARCHAR2(256)	NOT NULL	IP address of the peer path point.
VCID	NUMBER(10)	NOT NULL	Virtual circuit ID has been provisioned.
LOCALLABEL	NUMBER(10)	NOT NULL	Local label has been provisioned.
REMOTELABEL	NUMBER(10)	NOT NULL	Remote label has been provisioned.
PWCLASSNAME	VARCHAR2(256)	NOT NULL	Pseudowire class has been provisioned.
SLOTMODULETYPE	NUMBER(10)	NOT NULL	Slot module type. This is an internal field.
PORTMODULETYPE	NUMBER(10)	NOT NULL	Port module type. This is an internal field.

1.1.231 PW_SRV

The PW_SRV describes the pseudowire service as an end-to-end circuit.

Primary key: PW_SRV_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-221: PW_SRV

Column Name	Oracle Data Type	Integrity Constraints	Description
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.

Column Name	Oracle Data Type	Integrity Constraints	Description
CTMUNIQID	NUMBER(10)	NOT NULL	Prime Optical ID used to identify the contributor in the service. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
DESCRIPTION	VARCHAR2(256)	—	Description of the service has been provisioned.
DISCOVERYSTATE	NUMBER(1)	NOT NULL	Discovery status of the service. Values are: 0: Partial 1: Discovered
OPERSTATE	NUMBER(1)	NOT NULL	Summary of the operational state of all the contributors.
ISPROTECTED	NUMBER(1)	NOT NULL	Whether the service is protected or unprotected. Values are: 0: Unprotected 1: Protected
EPCOUNT	NUMBER(4)	NOT NULL	Number of endpoint contributors. This is an internal field.

1.1.232 Roll_Table

The Roll_Table contains information about the ONS 15600 Rolls table.

Primary key: ROLL_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-222 : Roll_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
RollNodeId	NUMBER(10)	NOT NULL	Node ID of the source node of the roll-from circuit. This is an internal field.
ConnectionIndex	NUMBER(10)	NOT NULL	Unique number within each source node. This is an internal field. The combination of RollNodeId and ConnectionIndex results in a unique number in the whole network.
RollFromCktNodeId	NUMBER(10)	NOT NULL	Node ID of the roll-from circuit.
RollFromCktUniqueID	NUMBER(10)	NOT NULL	Unique ID of the roll-from circuit.
RollToCktNodeID	NUMBER(10)	—	Node ID of the roll-to circuit.

Column Name	Oracle Data Type	Integrity Constraints	Description
RollToCktUniqueID	NUMBER(10)	—	Unique ID of the roll-to circuit.
RollState	NUMBER(1)	—	Roll state. Values are: 0: Roll pending 1: Roll completed 2: Roll canceled 3: Incomplete
RollValidSignal	NUMBER(1)	—	Valid signal on the roll-to point. Values are: 0: True 1: False
RollMode	NUMBER(1)	—	Mode of roll. Values are: 0: Auto 1: Manual
RollPathNodeID	NUMBER(10)	—	Node ID of the roll path.
RollPathModuleType	NUMBER(8)	—	Module type of the roll path.
RollPathIfIndex	NUMBER(20)	—	If static index at the roll path.
RollPathObjectIndex	NUMBER(10)	—	Object index at the roll path.
RollPathPhysicalLoc	NUMBER(20)	—	Physical location of the roll path.
RollPathModelType	NUMBER(4)	—	Model type of the poll path.
RollFromNodeID	NUMBER(10)	—	Node ID of the roll-from path.
RollFromModuleType	NUMBER(8)	—	Module type of the roll-from path.
RollFromIfIndex	NUMBER(20)	—	If static index at the roll-from path.
RollFromObjectIndex	NUMBER(10)	—	Object index at the roll-from path.
RollFromPhysicalLoc	NUMBER(20)	—	Physical location of the roll-to path.
RollFromModelType	NUMBER(4)	—	Roll-to path model type.
RollToNodeId	NUMBER(10)	—	Node ID of the roll-to path.

Column Name	Oracle Data Type	Integrity Constraints	Description
RollToModuleType	NUMBER(8)	—	Module type of the roll-to path.
RollToIfIndex	NUMBER(20)	—	If Static index at the roll-from path.
RollToObjectIndex	NUMBER(10)	—	Object index at roll-from path.
RollToPhysicalLoc	NUMBER(20)	—	Physical location at the roll-from path.
RollToModelType	NUMBER(4)	—	Model type of the roll-from path.
RollName	VARCHAR2(128)	NOT NULL	Unique roll name for Prime Optical GateWay/CORBA.
canBeCompleted	NUMBER(1)	—	Roll can be completed if the value is true. Values are: 0: True 1: False
canBeFinished	NUMBER(1)	—	Roll can be finished if the value is true. Values are: 0: True 1: False
canBeCancelled	NUMBER(1)	—	Roll can be cancelled if the value is true. Values are: 0: True 1: False
canBeForced	NUMBER(1)	—	Roll can be forced for a valid signal if the value is true. Values are: 0: True 1: False
rollFromCktSize	NUMBER(10)	NOT NULL	Roll from circuit size.

1.1.233 Schedule_Info_Table

Primary key: Schedule_Info_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-223 : Schedule_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ActivityType	VARCHAR2(64)	NOT NULL	Type of scheduled activity.
NEServiceName	VARCHAR2(256)	NOT NULL	NE service name.

Column Name	Oracle Data Type	Integrity Constraints	Description
RunOnMonday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Monday. Values are: 0: False 1: True
RunOnTuesday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Tuesday. Values are: 0: False 1: True
RunOnWednesday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Wednesday. Values are: 0: False 1: True
RunOnThursday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Thursday. Values are: 0: False 1: True
RunOnFriday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Friday. Values are: 0: False 1: True
RunOnSaturday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Saturday. Values are: 0: False 1: True
RunOnSunday	NUMBER(1)	DEFAULT 1 NOT NULL	Run configuration resynchronization on Sunday. Values are: 0: False 1: True
ActivityStartTime	DATE	NOT NULL	Time to start daily resynchronization.
ActivityInterval	NUMBER(4)	DEFAULT 360 NOT NULL	Resynchronization interval in minutes.
LastExecutionTime	DATE	—	Last execution time.

1.1.234 Server_Monitor_Table

The Server_Monitor_Table contains information about the server monitor.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-224: Server_Monitor_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ParameterIndex	NUMBER(10)	NOT NULL	Index used in the alarm.
Value	VARCHAR(32)	—	Value of the parameter.
CollectionTime	DATE	—	Collection time of the parameter.
NEDBAccessID	NUMBER(10)	—	Database ID of the NE; same as NEDBAccessID in the NE_Info_Table.

1.1.235 Server_Parameter_Table

The Server_Parameter_Table contains information about the server.

Primary key: SERVER_PARAMETER_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-225: Server_Parameter_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ParameterIndex	NUMBER(10)	NOT NULL	Index used in the alarm.
ParameterName	VARCHAR(32)	NOT NULL	Name of the parameter.
DisplayName	VARCHAR(64)	NOT NULL	Name of the parameter for display.
ModelType	NUMBER(4)	NOT NULL	Model type. Values are: 0: N/A –1: CTM –2: System –3: Network
Type	NUMBER(1)	NOT NULL	Parameter type. Values are: 1: Application task 2: System task 3: Circuit creation task
ConfigProperty	VARCHAR(32)	NOT NULL	Property name in the CTM_Config_Table.
ObjectType	NUMBER(5)	NOT NULL	Type of object.

1.1.236 Service_Availability_Table

The Service_Availability_Table contains information about the service availability.

Primary key: SERVICE_AVAILABILITY_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-226: Service_Availability_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SERVICE_INSTANCE_NAME	VARCHAR2(256)	NOT NULL	The service instance name.
STATUS NUMBER	NUMBER(1,0)	NOT NULL	The service status.
FIRST_ACTIVATED	DATE	NOT NULL	The date when the service was first activated.
LAST_CRASH	DATE	NOT NULL	The date when the service last crashed.
RUNNING_SINCE	DATE	NOT NULL	The service begin date.
PAST_UPTIME	NUMBER(22,0)	NOT NULL	The total time the service has been up.
DEACTIVATED	DATE	NOT NULL	The date when the service was deactivated.
PERC_UPTIME	NUMBER(3,0)	NOT NULL	The percentage of time the service has been up.

1.1.237 Subnetwork_Table

The Subnetwork_Table contains information about the subnetworks created by Prime Optical users.

Primary key: SUBNETWORK_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-227: Subnetwork_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SubnetID	NUMBER(8)	—	Autogenerated key for the table.
SubnetName	VARCHAR2(64)	—	Name of the subnetwork.
SubnetSystemTitle	VARCHAR2(64)	—	Another name of the subnetwork.
SubnetUserLabel	VARCHAR2(64)	—	Another name of the subnetwork.

Column Name	Oracle Data Type	Integrity Constraints	Description
SubnetType	NUMBER(1)	—	Type of subnetwork. Values are: 1: SONET 2: SDH 3: Other 4: Unknown
SubnetTopology	NUMBER(2)	—	Type of subnetwork topology. Values are: 1: Other 2: Point-to-point 3: Linear 4: UPSR 5: SNC 6: BLSR 7: MSSP 8: MESH 9: Open UPSR 10: Open SNC 11: Hybrid 12: Singleton 13: Unknown
SubnetGneID	NUMBER(4)	NOT NULL	Primary key of the GNE table.
NPId	NUMBER(10)	—	ID of the network partition associated with the subnetwork.

1.1.238 SVC_Config_Table

Primary key: SVC_Config_Table_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-228: SVC_Config_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SID	NUMBER(10)	NOT NULL	Service ID.
SVCName	VARCHAR2(64)	NOT NULL	Service name.
NEModelType	NUMBER(4)	NOT NULL	Model type of the NE.
SVCKind	NUMBER(10)	NOT NULL	Service type. Values are: 0: NE 1: PM 2: ELE 3: NET
NPID	NUMBER(10)	NOT NULL	Network partition ID.

Column Name	Oracle Data Type	Integrity Constraints	Description
NECount	NUMBER(10)	NOT NULL	Number of NEs managed by the service.
State	NUMBER(5)	NOT NULL	State of the service. Values are: 0: Stopped 1: Running 2: Failed
DiscoveryMode	NUMBER(2)	DEFAULT 1	Discovery mode of the service. Valid values are: 0: Manual 1: Automatic

1.1.239 SVC_Info_Table

The SVC_Info_Table contains information about the service package.

Primary key: SVC_INFO_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-229: SVC_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SvcID	NUMBER(10)	NOT NULL	Service ID.
SvcName	VARCHAR(64)	NOT NULL	Service name.
SvcPackageID	NUMBER(10)	NOT NULL	Unique ID of the service package.
SvcKind	NUMBER(10)	NOT NULL	Service kind. Values are: 0: NE 1: PM
SvcConfigName	VARCHAR(64)	NOT NULL	Service configuration name.
IsNPDependent	NUMBER(1)	NOT NULL	Values are: 0: False 1: True
IOR	VARCHAR2(64)	—	IOR representation.

1.1.240 SVC_NEModel_Table

The SVC_NEModel_Table contains information about the service NEs.

Primary key: SVC_NEMODEL_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-230: SVC_NEModel_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SvcClassName	VARCHAR(64)	NOT NULL	Service name.
SvcNeModelType	NUMBER(4)	NOT NULL	Associated model type.
SvcNeType	NUMBER(4)	NOT NULL	Service NE type. Values are: 0: Cerent NE 1: Non Cerent NE
SvcKind	NUMBER(10)	NOT NULL	Service type. Values are: 0: NE 1: PM

1.1.241 SVC_Package_Table

The SVC_Package_Table contains information about the service package.

Primary key: SVC_PACKAGE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-231 : SVC_Package_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SvcPackageId	NUMBER(10)	NOT NULL	Unique sequence number.
SvcPackageName	VARCHAR2(64)	NOT NULL	Unique name for the package.

1.1.242 SVC_Property_Table

The SVC_Property_Table contains information about the service property.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-232 : SVC_Property_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
SectionName	VARCHAR2(40)	—	Name of the section.
PropertyName	VARCHAR2(40)	—	Name of the parameter.
ActiveValue	VARCHAR2(256)	—	Active value of the parameter.

Column Name	Oracle Data Type	Integrity Constraints	Description
SID	NUMBER(10)	—	Service ID. Refer to SVC_Config_Table.

1.1.243 Sway_User_Port_Table

The Sway_User_Port_Table is obsolete.

1.1.244 Table_Info

The Table_Info table contains information about the sortable columns in the Prime Optical client GUI.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-233 : Table_Info

Column Name	Oracle Data Type	Integrity Constraints	Description
TABLENAME	VARCHAR2(30)	NOT NULL	Name of the table.
NAME	VARCHAR2(30)	NOT NULL	Name of the column.
VALUE	VARCHAR2(256)	NOT NULL	Order of the sortable columns.

1.1.245 TCA_Event_Table

The TCA_Event_Table contains TCA information about TCAs received from NEs.

Primary key: none

Tablespace: EVENT_DATA_TABLESPACE

Table 1-234: TCA_Event_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	ID to access this table for an NE. Same as NEDBAccessID in NE_Info_Table.
ModuleOrIfIndex	NUMBER(20)	NOT NULL	Unique integer that identifies an interface in the NE. Same as ifStaticIndex of Interface_Table.
AlarmEventType	NUMBER(5)	NOT NULL	Type of alarm/event table. List of valid types ranges from 1 to n and can be found in the Alarm_Event_Type_ENUM_

Column Name	Oracle Data Type	Integrity Constraints	Description
Table.			
ModuleType	NUMBER(8)	DEFAULT 0	Module type.
PhysicalLoc	NUMBER(20)	DEFAULT 0	Physical location.
TcaEventType	NUMBER(5)	NOT NULL	Type of TCA. Valid types can be found in the TCA_Event_Type_
Table.			
TcaLocation	VARCHAR2(20)	DEFAULT 'Not Applicable'	Displays near or far end TCA location.
TcaPeriod	VARCHAR2(20)	DEFAULT 'Not Applicable'	Displays TCA period: 15 minutes or 24 hours.
TcaValue	VARCHAR2(20)	—	Displays TCA value, which crossed threshold.
Direction	VARCHAR2(10)	DEFAULT 'N/A'	Displays direction of the line, Rx or Tx.
TimeStamp	DATE	NOT NULL	Prime Optical time stamp.
NEEventTimeStamp	DATE	NOT NULL	NE time stamp.
ModelType	NUMBER(10)	DEFAULT 0	Model type.
ObjectType	NUMBER(10)	DEFAULT 0	Object type.
StrObjInstance	VARCHAR2(256)	—	StrObjInstance name.

1.1.246 TCA_Event_Type_Table

The TCA_Event_Type_Table contains static information about TCA events.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-235: TCA_Event_Type_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NE_Model_Index	NUMBER(10)	NOT NULL	Index used in the TCA_Event_Type_Table.
TCA_Number	NUMBER(10)	NOT NULL	TCA number.
TCA_Type	VARCHAR2(64)	—	TCA type.
TCA_CTM_Type	VARCHAR2(64)	—	TCA Prime Optical type.
TCA_TMF_Type	VARCHAR2(64)	—	TCA TMF type.

Column Name	Oracle Data Type	Integrity Constraints	Description
TCA_GWC_Type	VARCHAR2(64)	—	TCA GateWay/CORBA type.

1.1.247 Threshold_Table

The Threshold_Table contains the 15-minute and 1-minute threshold values for various entities in an NE.

Primary key: THRESHOLD_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-236: Threshold_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ThresholdType	NUMBER(2)	NOT NULL	Type of threshold value that is stored in the row. Values are: 1: OC-3 2: STS 3: VT 4: DS-1 5: Ethernet
ThresholdSubType	NUMBER(4)	NOT NULL	Subtype.
ThresholdString	VARCHAR2(128)	—	Display string for threshold type and subtype.
FifteenMinValue	NUMBER(10)	NOT NULL	15-minute threshold value.
OneDayValue	NUMBER(10)	NOT NULL	1-day threshold value.

1.1.248 TL1_Encoding_Table

Primary key: PK_TL1_ENCODING_TABLE

Tablespace: BASE_DATA_TABLESPACE

Table 1-237: TL1_Encoding_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
ENUM	NUMBER(4)	—	ENUM for each encoding type.
ENCODE_TYPE	VARCHAR(32)	NOT NULL	Name of the encoding type.

Column Name	Oracle Data Type	Integrity Constraints	Description
DEFAULT_PORT	NUMBER(5)	—	Port number for the TL1 tunnel.

1.1.249 TL1_Tunnel_Gateway_Table

Primary key: TL1_TUNNEL_GATEWAY_TABLE

Tablespace: BASE_DATA_TABLESPACE

Table 1-238: TL1_Tunnel_Gateway_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NGNE_PKID	NUMBER(5)	—	Primary key for the non-Cisco GNE (NGNE).
NGNE_TID	VARCHAR(128)	—	Tunnel agent ID for the NGNE.
NGNE_IPADDRESS	NUMBER(10)	NOT NULL	IP address of the NGNE.
NGNE_USERNAME	VARCHAR2(64)	—	Username to log into the NGNE.
NGNE_USER_PASSWORD	VARCHAR2(64)	—	Password to log into the NGNE.
NGNE_TL1PORT	NUMBER(5)	—	NGNE TL1 port number.

1.1.250 TL1_Tunnel_Info_Table

The TL1_Tunnel_Info_Table stores information about the TL1 tunnels.

Primary key: PK_TL1_TUNNEL_INFO_TABLE

Tablespace: BASE_DATA_TABLESPACE

Table 1-239: TL1_Tunnel_Info_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
Tunnel_PKID	NUMBER(5)	—	Primary key for the tunnel.
TNE_NE_DBAccesss_ID	NUMBER(10)	UNIQUE NOT NULL	Foreign key from the NE_INFO_Table.
FK_NGNE_PKID	NUMBER(5)	—	Foreign key from the TL1_Tunnel_Gateway_Table.
TL1_Encoding	NUMBER(4)	—	ENUM for the encoding type.

Column Name	Oracle Data Type	Integrity Constraints	Description
TL1_Port_Number	NUMBER(5)	—	Port number for the TL1 tunnel. Default value is 3081.
Tunnel_Status	NUMBER(1)	—	Status of tunnel. Values are: 0: Initial 1: Open 2: Close
Tunnel_Description	VARCHAR(256)	—	Description for the tunnel.

1.1.251 Transaction_Log_Table

The Transaction_Log_Table stores information about Prime Optical transactions.

Primary key: TRANSACTION_LOG_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-240: Transaction_Log_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
TransId	NUMBER(20)	NOT NULL	Transaction ID generated by the database sequence.
NEDBAccessID	NUMBER(10)	NOT NULL	Source node for circuit and network provisioning; same as NEDBAccessID in the NE_Info_Table.
TimeStamp	DATE	—	When the transaction occurred.
UserName	VARCHAR2(64)	—	Name of the Prime Optical user.
ActionName	VARCHAR2(64)	—	Name of the transaction.
Result	NUMBER(1)	—	Whether the transaction succeeded or failed. Values are: 0: Failure 1: Success
Parms	VARCHAR2(2048)	—	Name/value pairs: name1:values1, and so on.
ClassName	VARCHAR2(64)	—	Name of the class from which the transaction was generated.
Module	NUMBER(2)	—	ID of the predefined modules.
LogMessage	VARCHAR2(1024)	—	Transaction log message.
Service	VARCHAR2(64)	—	Name of the service that logged this entry (for example, ONS 15454 NE Service).

1.1.252 TP_SRV

The TP_SRV describes the TP service as an end-to-end circuit.

Primary key: TP_SRV_PK

Tablespace: NGXP_DATA_TABLESPACE

Table 1-241 : TP_SRV

Column Name	Oracle Data Type	Integrity Constraints	Description
SERVICEID	NUMBER(10)	NOT NULL	Service ID. This is an internal field.
CTMUNIQID	NUMBER(10)	NOT NULL	Unique Prime Optical ID used to identify the contributor in the service. This is an internal field.
SERVICENAME	VARCHAR2(256)	—	Service name.
DESCRIPTION	VARCHAR2(256)	—	Description of the service has been provisioned.
CTMWORKINGLSPUNIQID	NUMBER(10)	NOT NULL	Unique Prime Optical ID used to reference the LSP contributors for the working path. This is an internal field.
CTMPROTECTLSPUNIQID	NUMBER(10)	NOT NULL	Unique Prime Optical ID used to reference the LSP contributors for the protected path.
DISCOVERYSTATE	NUMBER(1)	NOT NULL	Discovery status of the service. Values are: 0: Partial 1: Discovered
OPERSTATE	NUMBER(1)	NOT NULL	Summary of the operational state of all the contributors.
ISPROTECTED	NUMBER(1)	NOT NULL	Whether the service is protected or unprotected. Values are: 0: Unprotected 1: Protected
WLSPNUM	NUMBER(10)	NOT NULL	LSP number used for the working path.
PLSPNUM	NUMBER(10)	NOT NULL	LSP number used for the protected path.
TXBANDWIDTH	NUMBER(20)	NOT NULL	Transmission bandwidth value has been provisioned.
RXBANDWIDTH	NUMBER(20)	NOT NULL	Receiver bandwidth value has been provisioned.
TUNKEY	VARCHAR2(256)	—	Tunnel key identifying the service.
EPCOUNT	NUMBER(4)	NOT NULL	Number of the endpoint contributors. This is an internal field.

1.1.253 Unmanaged_NE_Inventory_Table

The Unmanaged_NE_Inventory_Table contains information about the unmanaged NE inventory table.

Primary key: UNMANAGED_NE_INVENTORY_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-242: Unmanaged_NE_Inventory_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEDBAccessID	NUMBER(10)	NOT NULL	NE database access ID.
SERIALNUMBER	VARCHAR2(32)	—	—

1.1.254 User_Map_Table

The User_Map_Table contains information about groups and NEs in the Domain Explorer tree that are assigned to a Prime Optical user.

Primary key: USER_MAP_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-243: User_Map_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserID	NUMBER(4)	NOT NULL	User ID. Link to the User_Table.
TreeNodeType	NUMBER(1)	NOT NULL	Type of tree node. Values are: 1: Root node 2: Group node 3: NE node
GroupOrNEID	NUMBER(8)	NOT NULL	If TreeNodeType = 3, this column is the ID to access other NE tables; same as NEDBAccessID in the NE_Info_Table. If TreeNodeType = 2, this column is the group ID; same as GroupID in the Group_Info_Table. If TreeNodeType = 1, value is 0.

1.1.255 User_NE_Table

The User_NE_Table stores information about the NEs each user can view.

Primary key: None

Tablespace: BASE_DATA_TABLESPACE

Table 1-244: User_NE_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserID	NUMBER(4)	—	ID of the user.
Grouporneid	NUMBER(10)	—	Group ID or NEDBaccessid.

1.1.256 User_Preference_Table

The User_Preference_Table stores information about user preferences.

Primary key: USER_PREFERENCE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-245: User_Preference_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserName	VARCHAR2(64)	NOT NULL	Prime Optical username.
ClientModule	NUMBER(4)	—	Client module ID. Values are: 0: Network Map 1: Control Panel
AutoRefresh	NUMBER(1)	—	Whether autorefresh is on or off. Values are: 1: On 0: Off

1.1.257 User_Table

The User_Table contains information about all Prime Optical users.

Primary key: USER_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-246: User_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
UserID	NUMBER(4)	NOT NULL	User ID. Link to the User_Map_Table. Each user has a unique ID.
UserName	VARCHAR2(64)	UNIQUE,	Unique name of the user.

Column Name	Oracle Data Type	Integrity Constraints	Description
		NOT NULL	
UserPassword	RAW(256)	—	User password stored in Message Digest 5 (MD5) encrypted form.
TypeOfUser	NUMBER(1)	—	Type of user. Values are: 1: Visionway user 2: Serviceway user
SubTypeOfUser	NUMBER(3)	—	Subtype of user.
AreaName	VARCHAR2(64)	—	Name of the network area associated with the user.
LastLoginTime	DATE	—	Time when the user last logged in.
UserLoginDisabled	NUMBER(1)		Value is 1 if the user login has been disabled.
UserDescription	VARCHAR2(256)	—	Description of the user.
CmsUser	VARCHAR2(64)	—	CTC username.
CmsPassword	VARCHAR2(64)	—	CTC password.
FailedAttempts	NUMBER(1)	DEFAULT 0	Number of failed user attempts.
StatusOfUser	NUMBER(1)	DEFAULT 0	User login status. Values are: 0: User is logged in 1: User is not logged in
PasswordSetTime	DATE	—	Time stamp when the password was set.
LastLoginFailtime	DATE	—	Time stamp when the last login failed.
lockedstate	NUMBER(1)	DEFAULT 0	Lockout state. Values are: 0: Not locked 1: Locked
PropagateToNE	NUMBER(1)	DEFAULT 0	Whether to propagate the user to the NE. Values are: 0: Do not propagate user to NE 1: Propagate user to NE
EnablePswdChange	NUMBER(1)	DEFAULT 1	Whether the password change option is enabled or disabled. Values are: 0: Disabled 1: Enabled
FirstUse	NUMBER(1)	DEFAULT 1	Values are: 0: False 1: True

Column Name	Oracle Data Type	Integrity Constraints	Description
UseGlobalLockout	NUMBER(1)	DEFAULT 1	Whether to enable global lockout. Values are: 0: False 1: True
LockoutEnabled	NUMBER(1)	DEFAULT 1	Whether lockout is enabled. Values are: 0: False 1: True
LockoutPeriod	VARCHAR(3)	DEFAULT 30	Period after which the client is locked out.
UseGlobalLogout	NUMBER(1)	DEFAULT 1	Whether to enable global logout. Values are: 0: False 1: True
LogoutEnabled	NUMBER(1)	DEFAULT 1	Whether logout is enabled. Values are: 0: False 1: True
LogoutPeriod	VARCHAR(4)	DEFAULT 60	Period after which the client is logged out.
EnableMultipleLogin	NUMBER(1)	DEFAULT 1	Whether to allow multiple user logins. Values are: 0: Disabled 1: Enabled
AutoDisableInterval	NUMBER(3)	DEFAULT 0	Active login interval. The range is from 0 to 360 days.

1.1.258 User_Type_Table

The User_Type_Table contains information about the different Prime Optical user types.

Primary key: USER_TYPE_TABLE_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-247: User_Type_Table

Column Name	Oracle Data Type	Integrity Constraints	Description

Column Name	Oracle Data Type	Integrity Constraints	Description
UserTypeID	NUMBER(4)	NOT NULL	User type ID that is unique for each user. Values are: 1: SuperUser 2: SysAdmin 3: NetworkAdmin 4: Operator 5: Provisioner
UserTypeName	VARCHAR2(64)	NOT NULL	Name of the user type.
Properties	NUMBER(1)	—	Defines whether the user has NEs assigned to him/her or not. Values are: 0: Assign no NEs 1: Assign all NEs 2: Assign Nes
Description	VARCHAR2(256)	—	Description of the user type.
isDefault	NUMBER(1)	DEFAULT 1	Defines if a user is a default user. Values are: 0: Custom user profile 1: Default user profile
AllowedSession	NUMBER(2)	DEFAULT 1	Defines if a user is allowed single or multiple user logins. Values are: 0: Allow multiuser login 1: Allow 1 user login 2: Allow 2 user login 3: Allow 3 user login 4: Allow 4 user login 5: Allow 5 user login 6: Allow 6 user login 7: Allow 7 user login 8: Allow 8 user login 9: Allow 9 user login 10: Allow 10 user login

1.1.259 Version_Table

The Version_Table contains version information about the NEs supported by Prime Optical.

Primary key: VERSION_PK

Tablespace: BASE_DATA_TABLESPACE

Table 1-248: Version_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
NEType	NUMBER(4)	NOT NULL	Type of NE.
SoftwareVersion	VARCHAR2(128)	NOT NULL	Major software version supplied by the NE.
NEModelIndex	NUMBER(4)	NOT NULL	The Version_Table is created when the product is installed. This number is used in the Model_Type_Table and in the NE_Info_Table. Values are: 0: Unknown 1: Cisco ONS 15303 2: Cisco ONS 15304 3: Cisco ONS 15454 5: Cisco ONS 15327 1_02 7: Cisco ONS 15200 9: Cisco ONS 15454 3_03 12: Cisco ONS 15540 14: Cisco ONS 15454 SDH 15: Cisco ONS 15540 ESP 0_5 16: Cisco ONS 15540 XPS 17: Cisco ONS 15530 18: Cisco ONS 15501 19: Cisco ONS 15454 3_10 20: Cisco ONS 15454 3_20 21: Cisco ONS 15454 3_30 22: Cisco ONS 15327 3_30 23: Cisco ONS 15200 1_1 24: Cisco ONS 15216 EDFA2 25: Cisco ONS 15216 EDFA1 26: Cisco ONS 15216 OADM2 27: Cisco ONS 15216 OADM1 28: Cisco ONS 15216 DWDM 30: Cisco ONS 15600 32: Cisco ONS 15454 3_04 33: Cisco ONS 15454 3_40 34: Cisco ONS 15327 3_40 35: Cisco ONS 15454SDH 3_40 36: Cisco ONS 15216 OSC 37: Cisco ONS 15216 DCU 41: Cisco ONS 15216 OADM1 2_1 42: Cisco ONS 15454 4_00 43: Cisco ONS 15327 4_00 44: Cisco ONS 15454SDH 4_00 46: Cisco ONS 15305 1_0 47: Cisco ONS 15454 4_01 48: Cisco ONS 15327 4_01 49: Cisco ONS 15454 SDH 4_01

Column Name	Oracle Data Type	Integrity Constraints	Description
			50: Cisco ONS 15600 1_10 51: Unmanaged NE 53: Cisco ONS 15216 unidirectional filter 54: Cisco ONS 15501 AC 55: Cisco ONS 15454 4_60 56: Cisco ONS 15327 4_60 57: Cisco ONS 15454SDH 4_60 58: Cisco ONS 15216 EDFA3 60: Cisco ONS 15216 EDFA2 2_3 61: Cisco ONS 15454 5_00 62: Cisco ONS 15454 SDH 5_00 63: Cisco ONS 15327 5_00 64: Cisco ONS 15600 5_00 65: Cisco ONS 15310CL 5_00 66: Cisco ONS 15216 EDFA2 2_4 69: Cisco ONS 15305 1_1 70: Cisco ONS 15305 2_0 71: HFR 72: GSR IOX 73: CATALYST 6509 74: ONS15600SDH 1_40 75: ONS15454 4_70 76: ONS15454SDH 4_70 77: ONS15454 6_00 78: ONS15454SDH 6_00 79: ONS15327 6_00 80: ONS15600 6_00 81: ONS15310CL 6_00 82: ONS15305CTC 3_00 83: ONS15216 OADM1 2_3 84: ONS15530 DPV_I 85: ONS15540 ESP DPV_I 86: ONS15540 XPS DPV_I 87: ONS15454 5_02 88: ONS15454SDH 5_02 89: ONS15327 5_02 90: ONS15600 5_02 91: ONS15310CL 5_02 101: ONS15305 2_01 102: ONS15530 BD_I 105: ONS15454 7_00 106: ONS15454SDH 7_00 107: ONS15327 7_00 108: ONS15600 7_00

Column Name	Oracle Data Type	Integrity Constraints	Description
			109: ONS15310CL 7_00 110: ONS15365 1_00 112: Cisco 7600 113: Cisco 7600 114: ONS15310MA 7_00 115: ONS15454 6_20 116: ONS15454SDH 6_20 117: ONS15310CL 6_20 118: ONS15327 6_20 119: ONS15600 6_20 120: ONS15454 7_20 121: ONS15454SDH 7_20 122: ONS15327 7_20 123: ONS15600 7_20 124: ONS15310CL 7_20 125: ONS15310MA 7_20 126: ONS15454 8_00 127: ONS15454SDH 8_00 128: ONS15600 8_00 129: ONS15600SDH 8_00 130: ONS15310CL 8_00 131: ONS15310MA 8_00 132: ONS15454 8_50 133: ONS15454SDH 8_50 134: ONS15600 8_50 135: ONS15600SDH 8_50 136: ONS15310CL 8_50 137: ONS15310MA 8_50 138: ONS15454 9_00 139: ONS15454SDH 9_00 140: ONS15600 9_00 141: ONS15600SDH 9_00 142: ONS15310CL 9_00 143: ONS15310MA 9_00 144: ONS15310MASDH 9_00 145: ONS15454 9_10 146: ONS15454SDH 9_10 147: ONS15600 9_10 148: ONS15600SDH 9_10 149: ONS15310CL 9_10 150: ONS15310MA 9_10 151: ONS15310MASDH 9_10 152: ONS15454 9_20 153: ONS15454SDH 9_20 157: ONS15310MA 9_20

Column Name	Oracle Data Type	Integrity Constraints	Description
			158: ONS15310MASDH 9_20 159: CRS 3_9
NEModelFeatures	VARCHAR2(2)	DEFAULT 'NN'	Used to specify the features of the model.
ModelName	VARCHAR2(50)	NOT NULL	Name of the model.
FactoryDefault	NUMBER(1)	DEFAULT 0	Default supported NE versions. When a new database is installed, this table is prepopulated with the supported NE versions. When a user adds a new NE version, the FactoryDefault column is set to zero, indicating a nonfactory-supplied version.

1.1.260 Wavelength_Table

The Wavelength_Table contains information about wavelength values and channel numbers.

Primary key: PK_WAVELENGTH_TABLE

Tablespace: BASE_DATA_TABLESPACE

Table 1-249: Wavelength_Table

Column Name	Oracle Data Type	Integrity Constraints	Description
Alias	VARCHAR2(4000)	NOT NULL	The string name, which is used to display the wavelength value on the Prime Optical GUI.
CTCString	VARCHAR2(4000)	NOT NULL	The CTC code used to identify a specific wavelength in string format.
CTCValue	NUMBER(8)	NOT NULL	The CTC code used to identify a specific wavelength in number format. The fields CTCString and CTCValue are used to exchange information between Prime Optical and CTC layers.
OddEven	NUMBER(8)	NOT NULL	A numeric field used to specify whether a wavelength is odd or even. A value of 1 indicates an odd-numbered wavelength; a value of 2 indicates an even-numbered wavelength.

Column Name	Oracle Data Type	Integrity Constraints	Description
LCBand	NUMBER(8)	NOT NULL	A numeric field used to specify whether a wavelength belongs to the C or L band. A value of 1 indicates a C-band wavelength; a value of 2 indicates an L-band wavelength.
Channel32	NUMBER(8)	NOT NULL	A numeric field used for channel numbers in the 32 group. A value of 0 indicates that the wavelength does not belong to the 32 group.
Channel40	NUMBER(8)	NOT NULL	A numeric field used for channel numbers in the 40 group. A value of 0 indicates that the wavelength does not belong to the 40 group.

1.1.261 Wavemux_NE

The Wavemux_NE table is obsolete.

1.1.262 1.1.264 XML_Parser_Static_Tree_Table

The XML_Parser_Static_Tree_Table stores extensible markup language (XML) parsing information used by the Prime Optical server to parse the XML node passed by the CTC model layer.

Primary key: XML_PARSER_STATIC_TREE_PK

Tablespace: BASE_DATA_TABLESPACE

2 Database Schema Views

This chapter describes the database views that Prime Optical creates and uses. The following views are listed in alphabetical order by category:

- Common PM Views
- Prime Optical Base Views
- Cisco ONS 152xx Views
- Cisco ONS 15216 Views
- Cisco ONS 153xx Views
- Cisco ONS 15454 Views
- Cisco ONS 155xx Views
- Cisco ONS 156xx Views

2.1 Common PM Views

2.1.1 NE_Module_Level_Debug_View

```
create or replace view NE_MODULE_LEVEL_DEBUG_VIEW AS
SELECT
    nit.NESysID AS NESYSID,
    nmld.NEDBACCESSID AS NEDBACCESSID,
    nmld.SERVICEINSTID AS SERVICEINSTID,
    nmld.SERVICEID AS SERVICEID,
    DECODE(LENGTH(RTRIM(LTRIM(nmld.MODULENAME))),0,'false',null,'false','true') AS
ENABLED,
    nmld.MODULENAME AS MODULENAME
FROM
    NE_MODULE_LEVEL_DEBUG nmld,
    NE_INFO_TABLE nit
WHERE
    nit.NEDBAccessID = nmld.NEDBACCESSID(+)
union
SELECT
    'SIID' AS NESYSID,
    -100 AS NEDBACCESSID,
    SERVICEINSTID,
    SERVICEID,
    DECODE(LENGTH(RTRIM(LTRIM(MODULENAME))),0,'false',null,'false','true') AS ENABLED,
    MODULENAME
FROM
    NE_MODULE_LEVEL_DEBUG
WHERE
    NEDBACCESSID = -100 AND
    SERVICEINSTID > -100
union
SELECT
    'SIID' AS NESYSID,
    -100 AS NEDBACCESSID,
    -100 AS SERVICEINSTID,
    SERVICEID,
    DECODE(LENGTH(RTRIM(LTRIM(MODULENAME))),0,'false',null,'false','true') AS ENABLED,
    MODULENAME
FROM
    NE_MODULE_LEVEL_DEBUG
WHERE
    NEDBACCESSID = -100 AND
    SERVICEINSTID = -100;
```

2.1.2 SDH_HO_FarEnd_PM_View

```
CREATE OR REPLACE VIEW Sdh_Ho_FarEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       ho454sdh.ModuleType,
       ho454sdh.Physicalloc,
       ho454sdh.NEIfIndex,
       ho454sdh.TimeStamp,
       ho454sdh.NEIfPMStatus,
       ho454sdh.inMaintenance,
       ho454sdh.fehpeb as FEHPEB,
       ho454sdh.fehpes as FEHPES,
       ho454sdh.fehpses as FEHPSES,
       ho454sdh.fehpuas as FEHPUAS,
    ho454sdh.fehpfc as FEHPFC,
       ho454sdh.fehpbbe as FEHPBBE,
       ho454sdh.fehpesr as FEHPESR,
       ho454sdh.fehpsesr as FEHPSESR,
       ho454sdh.fehpbber as FEHPBBER,
       ho454sdh.thresholdList as ThresholdList,
       ho454sdh.IS24H as IS24H,
     ho454sdh.ModelType,
       ho454sdh.ObjectType,
       ho454sdh.StrObjInstance,
       ho454sdh.INVALIDITYLIST,
       ho454sdh.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    ho454sdh.StrCorbaTidAid
FROM   ONS15454SDH_HO_PM_TABLE ho454sdh,
       ne_info_table ne,
       module_type_table en
WHERE
       ho454sdh.Nedbaccessid = ne.nedbaccessid AND
       ho454sdh.ModuleType   = en.ModuleType   AND
    ne.nemodeltype      = en.modeltype;
```

2.1.3 SDH_HO_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sdh_Ho_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       getModuleName_pm(ne.nenodeid, ho454sdh.ModuleType) as ModuleName,
       decode(ho454sdh.ModuleType,'',ms454sdh.ModuleType,ho454sdh.ModuleType) as
ModuleType,
       ho454sdh.Physicalloc,
       ho454sdh.NEIfIndex,
       decode(ho454sdh.TimeStamp,'',ms454sdh.timestamp,ho454sdh.Timestamp) as Timestamp,
       ho454sdh.NEIfPMStatus,
       ho454sdh.inMaintenance,
       ho454sdh.hpeb as HPEB,
       ho454sdh.hpes as HPES,
       ho454sdh.hpses as HPSES,
       ho454sdh.hpuas as HPUAS,
    ho454sdh.hpfc as HPFC,
       ho454sdh.hpbbe as HPBBE,
       GETPMPARAMETERVALUE(ho454sdh.hpppjcpdet,ms454sdh.msppjcpdet) as PPJC_PDET,
       GETPMPARAMETERVALUE(ho454sdh.hpnpjcpdet,ms454sdh.msnpjcpdet) as NPJC_PDET,
       GETPMPARAMETERVALUE(ho454sdh.hpppjcpgen,ms454sdh.msppjcpgen) as PPJC_PGEN,
       GETPMPARAMETERVALUE(ho454sdh.hpnpjcpgen,ms454sdh.msnpjcpgen) as NPJC_PGEN,
       ho454sdh.hppjdiff as PJ_DIFF,
       ho454sdh.hppjpsec as PJP_SEC,
       ho454sdh.hppjnsec as PJN_SEC,
       ho454sdh.hpesr as HPESR,
       ho454sdh.hpsesr as HPSESR,
       ho454sdh.hpbber as HPBBER,
       ho454sdh.thresholdList as ThresholdList,
       ho454sdh.IS24H as IS24H,
       ho454sdh.ModelType,
       ho454sdh.ObjectType,
       ho454sdh.StrObjInstance,
       ho454sdh.INVALIDITYLIST,
```

```
       ho454sdh.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    ho454sdh.StrCorbaTidAid
FROM   ONS15454SDH_HO_PM_TABLE ho454sdh,
       ONS15454SDH_MS_PM_TABLE ms454sdh,
       ne_info_table ne
WHERE
        ms454sdh.Nedbaccessid(+) = ho454sdh.nedbaccessid and
        ms454sdh.timestamp(+) = ho454sdh.Timestamp and
        ms454sdh.moduletype(+)   =ho454sdh.ModuleType and
        ms454sdh.physicalloc(+) = ho454sdh.physicalloc and
        ms454sdh.IS24H(+)      = ho454sdh.IS24H  and
```

2.1.4 SDH_MS_FarEnd_PM_View

```
CREATE OR REPLACE VIEW Sdh_Ms_FarEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       ms454sdh.ModuleType,
       ms454sdh.Physicalloc,
       ms454sdh.NEIfIndex,
       ms454sdh.TimeStamp,
       ms454sdh.NEIfPMStatus,
       ms454sdh.inMaintenance,
       ms454sdh.femseb as FEMSEB,
       ms454sdh.femses as FEMSES,
       ms454sdh.femsses as FEMSSES,
       ms454sdh.femsuas as FEMSUAS,
    ms454sdh.femsfc as FEMSFC,
       ms454sdh.femsbbe as FEMSBBE,
       ms454sdh.femsesr as FEMSESR,
       ms454sdh.femssesr as FEMSSESR,
       ms454sdh.femsbber as FEMSBBER,
       ms454sdh.thresholdList as ThresholdList,
       ms454sdh.IS24H as IS24H,
     ms454sdh.ModelType,
       ms454sdh.ObjectType,
       ms454sdh.StrObjInstance,
       ms454sdh.INVALIDITYLIST,
       ms454sdh.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    ms454sdh.StrCorbaTidAid
FROM   ONS15454SDH_MS_PM_TABLE ms454sdh,
       ne_info_table ne,
       module_type_table en
WHERE
       ms454sdh.Nedbaccessid = ne.nedbaccessid AND
       ms454sdh.ModuleType    = en.ModuleType   AND
    ne.nemodeltype      = en.modeltype;
```

2.1.5 SDH_MS_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sdh_Ms_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       ms454sdh.ModuleType,
       ms454sdh.Physicalloc,
       ms454sdh.NEIfIndex,
       ms454sdh.TimeStamp,
       ms454sdh.NEIfPMStatus,
       ms454sdh.inMaintenance,
       ms454sdh.mseb as MSEB,
       ms454sdh.mses as MSES,
       ms454sdh.msses as MSSES,
       ms454sdh.msuas as MSUAS,
       ms454sdh.msfc as MSFC,
       ms454sdh.msbbe as MSBBE,
```

```
       ms454sdh.mspsc as MSPSC,
       ms454sdh.mspsd as MSPSD,
       ms454sdh.mspscw as MSPSCW,
       ms454sdh.mspsdw as MSPSDW,
       ms454sdh.mspscs as MSPSCS,
       ms454sdh.mspsds as MSPSDS,
       ms454sdh.mspscr as MSPSCR,
       ms454sdh.mspsdr as MSPSDR,
       ms454sdh.msesr as MSESR,
       ms454sdh.mssesr as MSSESR,
       ms454sdh.msbber as MSBBER,
       ms454sdh.thresholdList as ThresholdList,
       ms454sdh.IS24H as IS24H,
       ms454sdh.ModelType,
       ms454sdh.ObjectType,
       ms454sdh.StrObjInstance,
       ms454sdh.INVALIDITYLIST,
       ms454sdh.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    ms454sdh.StrCorbaTidAid
FROM   ONS15454SDH_MS_PM_TABLE ms454sdh,
       ne_info_table ne,
       module_type_table en
WHERE
       ms454sdh.Nedbaccessid = ne.nedbaccessid AND
       ms454sdh.ModuleType    = en.ModuleType   AND
    ne.nemodeltype      = en.modeltype;
```

2.1.6 SDH_RS_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sdh_Rs_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       rs454sdh.ModuleType,
       rs454sdh.Physicalloc,
       rs454sdh.NEIfIndex,
       rs454sdh.TimeStamp,
       rs454sdh.NEIfPMStatus,
       rs454sdh.inMaintenance,
       rs454sdh.rseb as RSEB,
       rs454sdh.rses as RSES,
       rs454sdh.rsses as RSSES,
       rs454sdh.rssefs as RSSEFS,
       rs454sdh.rsbbe as RSBBE,
       rs454sdh.rsesr as RSESR,
       rs454sdh.rssesr as RSSESR,
       rs454sdh.rsbber as RSBBER,
       rs454sdh.thresholdList as ThresholdList,
       rs454sdh.IS24H as IS24H,
       rs454sdh.ModelType,
       rs454sdh.ObjectType,
       rs454sdh.StrObjInstance,
       rs454sdh.INVALIDITYLIST,
       rs454sdh.rowid as Myrow,
    ne.NEAliasID as NEAliasID
    rs454sdh.StrCorbaTidAid
FROM   ONS15454SDH_RS_PM_TABLE rs454sdh,
       ne_info_table ne,
       module_type_table en
WHERE
       rs454sdh.Nedbaccessid = ne.nedbaccessid AND
       rs454sdh.ModuleType    = en.ModuleType   AND
    ne.nemodeltype      = en.modeltype;
```

2.1.7 SONET_ENET_PM_View

```
CREATE OR REPLACE VIEW SONET_ENET_PM_VIEW  AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
```

```
       en.ModuleType,
       enet.Physicalloc,
       enet.NEIfIndex,
       enet.TimeStamp,
       enet.NEIfPmStatus,
       enet.InMaintenance,
       enet.ThresholdList,
       enet.IfInOctets,
       enet.RxPackets,
       enet.IfInUcastPkts,
       enet.IfInMulticastPkts,
       enet.IfInBroadcastPkts,
       enet.IfInErrors,
       enet.IfOutOctets,
       decode(en.Moduletype,102,to_char(enet.TxPackets),
                            103,to_char(enet.TxPackets),
                            118,to_char(enet.TxPackets),
                            119,to_char(enet.TxPackets),
                            181,to_char(enet.TxPackets),
                            182,to_char(enet.TxPackets),
                            183,to_char(enet.TxPackets),
                            184,to_char(enet.TxPackets),
                            271,to_char(enet.TxPackets),
                            272,to_char(enet.TxPackets),
                            273,to_char(enet.TxPackets),
                            300,to_char(enet.TxPackets),
                            301,to_char(enet.TxPackets),
                            330,to_char(enet.TxPackets),
                            355,to_char(enet.TxPackets),

get_enet_value2(enet.IfOutUcastPkts,enet.IfOutMulticastPkts,enet.IfOutBroadcastPkts)
             ) as TxPackets,
       enet.IfOutUcastPkts,
       enet.IfOutMulticastPkts,
       enet.IfOutBroadcastPkts,
       enet.Dot3StatsAlignmentErrors,
       enet.Dot3StatsFCSErrors,
       enet.Dot3StatsSingleCollisionFrames,
       enet.Dot3StatsMultiCollisionFrames,
       enet.Dot3StatsDeferredTransmissions,
       enet.Dot3StatsLateCollisions,
       enet.Dot3StatsExcessiveCollisions,
       enet.EtherStatsUndersizePkts,
       enet.EtherStatsFragments,
       enet.EtherStatsOversizePkts,
       enet.EtherStatsJabbers,
       decode(en.Moduletype,102, to_char(enet.RxTotalErrors),
                            103, to_char(enet.RxTotalErrors),
                            118, to_char(enet.RxTotalErrors),
                            119, to_char(enet.RxTotalErrors),
                             62, to_char(enet.RxTotalErrors),

get_enet_value2(enet.Dot3StatsFCSErrors,enet.Dot3StatsAlignmentErrors,enet.EtherStatsUnder
SizePKTS, enet.ETHERSTATSOVERSIZEPKTS,enet.ETHERSTATSFRAGMENTS,enet.ETHERSTATSJABBERS)
             )  as Rx_TOTAL_ERRORS,
       decode(en.Moduletype, 102, enet.RxRunts,
                             103, enet.RxRunts,
                             118, enet.RxRunts,
                             119, enet.RxRunts,
                              62, enet.RxRunts,
                         187, enet.RxRunts,
              enet.ETHERSTATSFRAGMENTS
             ) as Rx_RUNTS,
       enet.RxGIANTS as rx_giants,
       decode(en.Moduletype, 102, enet.TxCollisions,
                             103, enet.TxCollisions,
                118, enet.TxCollisions,
                             119, enet.TxCollisions,

get_enet_value(enet.Dot3StatsSingleCollisionFrames,enet.Dot3StatsMultiCollisionFrames)
             ) as Tx_Collisions,
       enet.RxPauseFrames,
       enet.TxPauseFrames,
       enet.RxPktsDropInternalCongestion,
       enet.TxPktsDropInternalCongestion,
       enet.IS24H,
```

```
       enet.EtherUtilizationStats,
       enet.RxGMACDropCounts,
       enet.RxThresholdOversizes,
       enet.PortDropCounts,
       enet.TxGiants,
       enet.HDLCPktDrops,
       enet.RxControlFrames,
       enet.RXUnknownOpCodeFrames,
       enet.IfInErrorBytePkts,
       enet.IfInFramingErrorPkts,
       enet.IfInJunkInterPkts,
       enet.IfInDiscards,
       enet.IfOutDiscards,
       enet.Dot3StatsFrameTooLong,
       enet.Dot3StatsCarrierSenseErrors,
       enet.Dot3StatsSQETestErrors,
       enet.EtherStatsPkts64Octets,
       enet.EtherStatsPkts65to127Octets,
       enet.EtherStatsPkts128to255Octets,
       enet.EtherStatsPkts256to511Octets,
       enet.EtherStatsPkts512to1023Octets,
       enet.EtherStatsPkts1024to1518Octets,
       enet.EtherStatsBroadcastPkts,
       enet.EtherStatsMulticastPkts,
       enet.EtherStatsOctets,
       enet.EtherStatsCollisions,
       enet.EtherStatsCollisionFrames,
       enet.EtherStatsCRCAlignErrors,
       enet.MediaIndStatsRxFramesTruncated,
       enet.MediaIndStatsRxFramesTooLong,
       enet.MediaIndStatsRxFramesBadCRC,
       enet.MediaIndStatsTxFramesBadCRC,
       enet.MediaIndStatsRxShortPkts,
       enet.MediaIndStatsOversizeDropped,
       enet.IfOutErrors,
       enet.EtherStatsPkts,
       enet.Dot3StatsInternalMacTxErrors,
       enet.Dot3StatsInternalMacRxErrors,
       enet.Dot3StatsSymbolErrors,
       enet.RxEtherUtilizationStats,
       enet.TxEtherUtilizationStats,
       enet.StatsLineLastClearedTime,
       enet.MEDIAINDSTATSTXFRAMESTOOLONG,
    decode(lag.LACPDUsRx,NULL, -1,lag.LACPDUsRx) as LACPDUSRX,
       decode(lag.MarkerPDUsRx,NULL, -1,lag.MarkerPDUsRx) as MARKERPDUS,
       decode(lag.MarkerResponsePDUsRx,NULL, -1,lag.MarkerResponsePDUsRx) as
MARKERRESPONSEPDUSRX,
       decode(lag.UnknownRx,NULL, -1,lag.UnknownRx) as UNKNOWNRX,
       decode(lag.IllegalRx,NULL, -1,lag.IllegalRx) as ILLEGALRX,
       decode(lag.LACPDUsTx,NULL, -1,lag.LACPDUsTx) as LACPDUSTX,
       decode(lag.MarkerPDUsTx,NULL, -1,lag.MarkerPDUsTx) as MARKERPDUSTX,
       decode(lag.MarkerResponsePDUsTx,NULL, -1,lag.MarkerResponsePDUsTx) as
MARKERRESPONSEPDUSTX,
       enet.ETHERSTATSTXFIFOOVERFLOWEVENTS,
       enet.ModelType,
       enet.ObjectType,
       enet.StrObjInstance,
       enet.InvalidityList,
       enet.rowid as myrow,
    ni.NEAliasID as NEAliasID,
    enet.StrCorbaTidAid

FROM
       ons15454_enet_pm_table enet, left outer join ons15454_LAG_pm_table lag
       on (enet.NEDBAccessID = lag.NEDBAccessID AND
           enet.TIMESTAMP = lag.TIMESTAMP AND
           enet.MODULETYPE = lag.MODULETYPE AND
           enet.PHYSICALLOC = lag.PHYSICALLOC AND
           enet.NEIFINDEX = lag.NEIFINDEX AND
           enet.IS24H = lag.IS24H),
        ne_info_table ni,
       module_type_table en
WHERE
       enet.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       enet.NEDBAccessID  = ni.NEDBAccessID
```

```
UNION ALL
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       enet.Physicalloc,
       enet.NEIfIndex,
       enet.TimeStamp,
       enet.NEIfPmStatus,
       enet.InMaintenance,
       enet.thresholdList as ThresholdList,
       enet.IfInOctets,
       to_number('-1') as RXPACKETS,
       enet.IfInUcastPkts,
       enet.IfInMulticastPkts,
       enet.IfInBroadcastPkts,
       enet.IfInErrors,
       enet.IfOutOctets,
       to_char(-1) as TXPACKETS,
       enet.IfOutUcastPkts,
       enet.IfOutMulticastPkts,
       enet.IfOutBroadcastPkts,
       enet.Dot3StatsAlignmentErrors,
       enet.Dot3StatsFCSErrors,
    to_number('-1') as DOT3STATSSINGLECOLLISIONFRAMES,
    to_number('-1') as DOT3STATSMULTICOLLISIONFRAMES,
    to_number('-1') as DOT3STATSDEFERREDTRANSMISSIONS,
    to_number('-1') as DOT3STATSLATECOLLISIONS,
    to_number('-1') as DOT3STATSEXCESSIVECOLLISIONS,
       enet.EtherStatsUndersizePkts,
       enet.EtherStatsFragments,
    enet.EtherStatsOversizePkts,
    enet.EtherStatsJabbers,
    to_char(-1) as RX_TOTAL_ERRORS,
    to_number('-1') as RX_RUNTS,
    to_number('-1') as RX_GIANTS,
    to_number('-1') as TX_COLLISIONS,
    enet.RxPauseFrames,
       enet.TxPauseFrames,
    to_number('-1') as RXPKTSDROPINTERNALCONGESTION,
    to_number('-1') as TXPKTSDROPINTERNALCONGESTION,
    enet.IS24H,
    to_number('-1') as ETHERUTILIZATIONSTATS,
    to_number('-1') as RXGMACDROPCOUNTS,
    to_number('-1') as RXTHRESHOLDOVERSIZES,
    enet.EtherStatsDropEvents,
    to_number('-1') as TXGIANTS,
    to_number('-1') as HDLCPKTDROPS,
    to_number('-1') as RXCONTROLFRAMES,
    to_number('-1') as RXUNKNOWNOPCODEFRAMES,
    to_number('-1') as IFINERRORBYTEPKTS,
    to_number('-1') as IFINFRAMINGERRORPKTS,
    to_number('-1') as IFINJUNKINTERPKTS,
    enet.IfInDiscards,
        enet.IfOutDiscards,
    enet.Dot3StatsFrameTooLong,
    to_number('-1') as DOT3STATSCARRIERSENSEERRORS,
    to_number('-1') as DOT3STATSSQETESTERRORS,
       enet.EtherStatsPkts64Octets,
       enet.EtherStatsPkts65to127Octets,
       enet.EtherStatsPkts128to255Octets,
       enet.EtherStatsPkts256to511Octets,
       enet.EtherStatsPkts512to1023Octets,
       enet.EtherStatsPkts1024to1518Octets,
       enet.EtherStatsBroadcastPkts,
       enet.EtherStatsMulticastPkts,
       enet.EtherStatsOctets,
    to_number('-1') as ETHERSTATSCOLLISIONS,
    to_number('-1') as ETHERSTATSCOLLISIONFRAMES,
    to_number('-1') as ETHERSTATSCRCALIGNERRORS,
    to_number('-1') as MEDIAINDSTATSRXFRAMESTRUNCATED,
    to_number('-1') as MEDIAINDSTATSRXFRAMESTOOLONG,
    to_number('-1') as MEDIAINDSTATSRXFRAMESBADCRC,
    to_number('-1') as MEDIAINDSTATSTXFRAMESBADCRC,
    to_number('-1') as MEDIAINDSTATSRXSHORTPKTS,
```

```
    to_number('-1') as MEDIAINDSTATSOVERSIZEDROPPED,
       enet.IfOutErrors,
       enet.EtherStatsPkts,
       enet.Dot3StatsInternalMacTxErrors,
       enet.Dot3StatsInternalMacRxErrors,
        enet.Dot3StatsSymbolErrors,
       enet.RxEtherUtilizationStats,
       enet.TxEtherUtilizationStats,
       to_Date('') as STATSLINELASTCLEAREDTIME,
       to_number('-1') as MEDIAINDSTATSTXFRAMESTOOLONG,
    to_number('-1') as LACPDUSRX,
       to_number('-1') as MARKERPDUS,
       to_number('-1') as MARKERRESPONSEPDUSRX,
       to_number('-1') as UNKNOWNRX,
       to_number('-1') as ILLEGALRX,
       to_number('-1') as LACPDUSTX,
       to_number('-1') as MARKERPDUSTX,
       to_number('-1') as MARKERRESPONSEPDUSTX,
       to_number('-1') as ETHERSTATSTXFIFOOVERFLOWEVENTS,

       enet.ModelType,
       enet.ObjectType,
       enet.StrObjInstance,
       enet.InvalidityList,
       enet.rowid as myrow,
    ni.NEAliasID as NEAliasID,
    pos.StrCorbaTidAid
FROM
       ons15600_enet_pm_table enet,
       ne_info_table ni,
       module_type_table en
WHERE
       enet.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       enet.NEDBAccessID  = ni.NEDBAccessID;
```

2.1.8 SONET_Line_FarEnd_PM_View

```
CREATE OR REPLACE VIEW Sonet_Line_FarEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       line454.ModuleType,
       line454.Physicalloc,
       line454.NEIfIndex,
       line454.TimeStamp,
       line454.NEIfPMStatus,
       line454.inMaintenance,
       line454.fecodeviolations as CV_LFE,
       line454.feerrsecs as ES_LFE,
       line454.feseverrsecs as SES_LFE,
       line454.feunavailablesecs as UAS_LFE,
       line454.fefcl as FC_LFE,
       line454.thresholdList as ThresholdList,
       line454.IS24H as IS24H,
       line454.ModelType,
       line454.ObjectType,
       line454.StrObjInstance,
       line454.InvalidityList as InvalidityList,
       line454.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    line454.StrCorbaTidAid
FROM   ONS15454_SONET_LINE_PM_TABLE line454,
       ne_info_table ne,
       module_type_table en
WHERE
       line454.Nedbaccessid = ne.nedbaccessid AND
       line454.ModuleType    = en.ModuleType   AND
ne.nemodeltype      = en.modeltype;
```

2.1.9 SONET_Line_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sonet_Line_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       line454.ModuleType,
       line454.Physicalloc,
       line454.NEIfIndex,
       line454.TimeStamp,
       line454.NEIfPMStatus,
       line454.inMaintenance,
       line454.codeviolations as CV_L,
       line454.errsecs as ES_L,
       line454.severrsecs as SES_L,
       line454.unavailablesecs as UAS_L,
       line454.PpjcPdet as PpjcPdet,
       line454.NpjcPdet as NpjcPdet,
       line454.PpjcPgen as PpjcPgen,
       line454.NpjcPgen as NpjcPgen,
       line454.FCL as FC_L,
       line454.PSC as PSC,
       line454.PSD as PSD,
       line454.PjDiff as PjDiff,
       line454.PjpSec as PjpSec,
       line454.PjnSec as PjnSec,
       line454.PSCW as PSC_W,
       line454.PSDW as PSD_W,
       line454.PSCR as PSC_R,
       line454.PSDR as PSD_R,
       line454.PSCS as PSC_S,
       line454.PSDS as PSD_S,
       line454.thresholdList as ThresholdList,
       line454.IS24H as IS24H,
       line454.ModelType,
       line454.ObjectType,
       line454.StrObjInstance,
       line454.InvalidityList as InvalidityList,
       line454.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
        line454.StrCorbaTidAid
FROM   ONS15454_SONET_LINE_PM_TABLE line454,
       ne_info_table ne,
       module_type_table en
WHERE
       line454.Nedbaccessid = ne.nedbaccessid AND
       line454.ModuleType    = en.ModuleType   AND
       ne.nemodeltype      = en.modeltype;
```

2.1.10 SONET_Physical_PM_View

```
CREATE OR REPLACE VIEW SONET_PHYSICAL_PM_VIEW AS
SELECT
        ni.NESysID as NESYSID,
        en.ModuleName as MODULENAME,
        phy.PhysicalLoc as PHYSICALLOC,
        phy.NEIfIndex as NEIFINDEX,
        phy.TimeStamp as TIMESTAMP,
        phy.NEIfPMStatus as NEIFPMSTATUS,
        phy.inMaintenance as INMAINTENANCE,
        phy.LaserBiasCurrent,
        phy.OpticalPowerTransmitted,
        phy.OpticalPowerReceived,
        phy.MinLaserBias as MINLASERBIAS,
        phy.AvgLaserBias as AVGLASERBIAS,
        phy.MaxLaserBias as MAXLASERBIAS,
        phy.MinLaserTemp as MINLASERTEMP,
        phy.AvgLaserTemp as AVGLASERTEMP,
        phy.MaxLaserTemp as MAXLASERTEMP,
```

```
        phy.MinReceivedLaserTemp as MINRECEIVEDLASERTEMP,
        phy.AvgReceivedLaserTemp as AVGRECEIVEDLASERTEMP,
        phy.MaxReceivedLaserTemp as MAXRECEIVEDLASERTEMP,
    phy.MinTransReceiverVoltage as MINTRANSRECEIVERVOLTAGE,
        phy.AvgTransReceiverVoltage as AVGTRANSRECEIVERVOLTAGE,
        phy.MaxTransReceiverVoltage as MAXTRANSRECEIVERVOLTAGE,
        phy.MinTransmittedPower as MINTRANSMITTEDPOWER,
        phy.AvgTransmittedPower as AVGTRANSMITTEDPOWER,
        phy.MaxTransmittedPower as MAXTRANSMITTEDPOWER,
        phy.MinReceivedPower as MINRECEIVEDPOWER,
        phy.AvgReceivedPower as AVGRECEIVEDPOWER,
        phy.MaxReceivedPower as MAXRECEIVEDPOWER,
    phy.MinPassThroughPower as MINPASSTHROUGHPOWER,
    phy.AvgPassThroughPower as AVGPASSTHROUGHPOWER,
    phy.MaxPassThroughPower as MAXPASSTHROUGHPOWER,
    phy.MinTxOSCPower as MINTXOSCPOWER,
    phy.AvgTxOSCPower as AVGTXOSCPOWER,
    phy.MaxTxOSCPower as MAXTXOSCPOWER,
    phy.MinRxOSCPower as MINRXOSCPOWER,
    phy.AvgRxOSCPower as AVGRXOSCPOWER,
    phy.MaxRxOSCPower as MAXRXOSCPOWER,
        ni.NEDbaccessid as NEDBACCESSID,
        phy.thresholdList as ThresholdList,
        en.ModuleType as MODULETYPE,
        phy.IS24H  as IS24H,
        phy.ModelType,
        phy.ObjectType,
        phy.StrObjInstance,
        phy.InvalidityList as InvalidityList,
        phy.rowid as myrow,
    ne.NEAliasID as NEAliasID,
       phy.StrCorbaTidAid
FROM
        ONS15454_PHY_LAYER_PM_Table phy,
        ne_info_table ni,
        module_type_table en
WHERE
        phy.ModuleType   = en.ModuleType   AND
        ni.NEModelType   = en.ModelType    AND
        phy.NEDBAccessID = ni.NEDBAccessID;
```

2.1.11 SONET_POS_PM_View

```
CREATE OR REPLACE VIEW SONET_POS_PM_VIEW  AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       pos.Physicalloc,
       pos.NEIfIndex,
       pos.TimeStamp,
       pos.NEIfPmStatus,
       pos.InMaintenance,
       pos.PosStatsRxPreHdlcBytes,
       pos.PosStatsRxPostHdlcBytes,
       pos.PosStatsRxPackets,
       pos.PosStatsRxNormalPackets,
       pos.PosStatsRxShorts,
       pos.PosStatsRxRunts,
       pos.PosStatsRxLongs,
       pos.PosStatsRxTotalErrors,
       pos.PosStatsRxCrcErrors,
       pos.PosStatsRxInputDropPackets,
       pos.PosStatsRxInputAbortPackets,
       pos.PosStatsTxPreHdlcBytes,
       pos.PosStatsTxPostHdlcBytes,
       pos.PosStatsTxPackets,
       pos.PosStatsDropCounts,
       pos.EtherStatsDropEvents,
       pos.GfpStatsRxSBitErrors,
       pos.GfpStatsRxMBitErrors,
       pos.GfpStatsRxTypeInvalid,
```

```
       pos.GfpStatsRxCRCErrors,
       pos.GfpStatsRxCIDInvalid,
       pos.GfpStatsCSFRaised,
       pos.GfpStatsRxFrame,
       pos.GfpStatsTxFrame,
       pos.GfpStatsRxOctets,
       pos.GfpStatsTxOctets,
       pos.GfpStatsLFDRaised,
       pos.MediaIndStatsRxFramesTruncated,
       pos.MediaIndStatsRxFramesTooLong,
       pos.MediaIndStatsRxFramesBadCRC,
       pos.MediaIndStatsRxShortPkts,
       pos.hdlcInOctets,
       pos.hdlcRxAborts,
       pos.hdlcOutOctets,
       pos.ifInOctets,
       pos.ifOutOctets,
       pos.rxPktsDropInternalCongestion,
       pos.ifInDiscards,
       pos.RxEtherUtilizationStats as Rx_Utilization_Stats,
       pos.TxEtherUtilizationStats as Tx_Utilization_Stats,
       pos.StatsLineLastClearedTime,
       pos.ifInPayloadCrcErrors,
       pos.ifOutPayloadCrcErrors,
       pos.ifOutOversizePackets,
       pos.HDLCPKTDROPS,
   pos.ifOutDiscards,
       pos.ThresholdList,
       pos.IS24H,
       pos.ModelType,
       pos.ObjectType,
       pos.StrObjInstance,
       pos.InvalidityList,
       pos.rowid as myrow,
   ni.NEAliasID as NEAliasID
FROM
       ons15454_pos_pm_table pos,
       ne_info_table ni,
       module_type_table en
WHERE
       pos.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       pos.NEDBAccessID  = ni.NEDBAccessID
UNION ALL
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       pos.Physicalloc,
       pos.NEIfIndex,
       pos.TimeStamp,
       pos.NEIfPmStatus,
       pos.InMaintenance,
       to_number('-1') as PosStatsRxPreHdlcBytes,
       to_number('-1') as PosStatsRxPostHdlcBytes,
       pos.RXTOTALPACKETS,
       to_number('-1') as PosStatsRxNormalPackets,
       to_number('-1') as PosStatsRxShorts,
       to_number('-1') as PosStatsRxRunts,
       to_number('-1') as PosStatsRxLongs,
       to_number('-1') as PosStatsRxTotalErrors,
       to_number('-1') as PosStatsRxCrcErrors,
       to_number('-1') as PosStatsRxInputDropPackets,
       to_number('-1') as PosStatsRxInputAbortPackets,
       to_number('-1') as PosStatsTxPreHdlcBytes,
       to_number('-1') as PosStatsTxPostHdlcBytes,
       pos.TXTOTALPACKETS,
       to_number('-1') as PosStatsDropCounts,
       to_number('-1') as EtherStatsDropEvents,
       pos.GfpStatsRxSBitErrors,
       pos.GfpStatsRxMBitErrors,
       pos.GfpStatsRxTypeInvalid,
       pos.GfpStatsRxCRCErrors,
       to_number('-1') as GfpStatsRxCIDInvalid,
       to_number('-1') as GfpStatsCSFRaised,
```

```
       pos.GfpStatsRxFrame,
       pos.GfpStatsTxFrame,
       pos.GfpStatsRxOctets,
       pos.GfpStatsTxOctets,
       to_number('-1') as GfpStatsLFDRaised,
       pos.MediaIndStatsRxFramesTruncated,
       pos.MediaIndStatsRxFramesTooLong,
       pos.MediaIndStatsRxFramesBadCRC,
       pos.MediaIndStatsRxShortPkts,
       pos.hdlcInOctets,
       pos.hdlcRxAborts,
       pos.hdlcOutOctets,
       to_number('-1') as ifInOctets,
       to_number('-1') as ifOutOctets,
       pos.RXPKTSDROPPEDINERNALCONGESTION,
       to_number('-1') as ifInDiscards,
       pos.RxEtherUtilizationStats as Rx_Utilization_Stats,
       pos.TxEtherUtilizationStats as Tx_Utilization_Stats,
       to_Date('') as StatsLineLastClearedTime,
       to_number('-1') as ifInPayloadCrcErrors,
       to_number('-1') as ifOutPayloadCrcErrors,
       to_number('-1') as ifOutOversizePackets,
       to_number('-1') as HDLCPKTDROPS,
   to_number('-1') as ifOutDiscards
       pos.thresholdList as ThresholdList,
       pos.IS24H,
       pos.ModelType,
       pos.ObjectType,
       pos.StrObjInstance,
       pos.InvalidityList,
       pos.rowid as myrow,
   ni.NEAliasID as NEAliasID
FROM
       ons15600_pos_pm_table pos,
       ne_info_table ni,
       module_type_table en
WHERE
       pos.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       pos.NEDBAccessID  = ni.NEDBAccessID;
```

2.1.12 SONET_Section_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sonet_Section_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       sec454.ModuleType,
       sec454.Physicalloc,
       sec454.NEIfIndex,
       sec454.TimeStamp,
       sec454.NEIfPMStatus,
       sec454.inMaintenance,
       sec454.severrframesecs as SEFS_S,
       sec454.codeviolations as CV_S,
       sec454.errsecs as ES_S,
       sec454.severrsecs as SES_S,
       sec454.thresholdList as ThresholdList,
       sec454.IS24H as IS24H,
       sec454.ModelType,
       sec454.ObjectType,
       sec454.StrObjInstance,
       sec454.InvalidityList as InvalidityList,
       sec454.rowid as Myrow,
   ne.NEAliasID as NEAliasID,
   sec454.StrCorbaTidAid
FROM   ONS15454_SONET_SEC_PM_TABLE sec454,
       ne_info_table ne,
       module_type_table en
WHERE
       sec454.Nedbaccessid = ne.nedbaccessid AND
       sec454.ModuleType    = en.ModuleType   AND
```

```
    ne.nemodeltype      = en.modeltype;
```

2.1.13 SONET_STSPath_FarEnd_PM_View

```
CREATE OR REPLACE VIEW Sonet_STSPATH_FarEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       path454.ModuleType,
       path454.Physicalloc,
       path454.NEIfIndex,
       path454.TimeStamp,
       path454.NEIfPMStatus,
       path454.inMaintenance,
       path454.fesonetpathcodeviolations as CV_PFE,
       path454.fesonetpatherrsecs as ES_PFE,
       path454.fesonetpathseverrsecs as SES_PFE,
       path454.fesonetpathunavailablesecs as UAS_PFE,
       path454.fesonetpathfc as FC_PFE,
       path454.thresholdList as ThresholdList,
       path454.IS24H as IS24H,
       path454.ModelType,
       path454.ObjectType,
       path454.StrObjInstance,
       path454.InvalidityList as InvalidityList,
       path454.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    path454.StrCorbaTidAid
FROM   ONS15454_SONET_PATH_PM_TABLE path454,
       ne_info_table ne,
       module_type_table en
WHERE
       path454.Nedbaccessid = ne.nedbaccessid AND
       path454.ModuleType   = en.ModuleType   AND
    ne.nemodeltype      = en.modeltype;
```

2.1.14 SONET_STSPath_NearEnd_PM_View

```
CREATE OR REPLACE VIEW Sonet_STSPATH_NearEnd_pm_view AS
SELECT
       ne.NESysID,
       ne.NEDbaccessid,
       en.ModuleName,
       decode(path454.ModuleType,'',line454.ModuleType,
                  path454.ModuleType) as ModuleType,
       path454.Physicalloc,
       path454.NEIfIndex,
       decode(path454.TimeStamp,'',line454.timestamp,
                     path454.Timestamp) as Timestamp,
       path454.NEIfPMStatus,
       path454.inMaintenance,
       path454.sonetpathcodeviolations as CV_P,
       path454.sonetpatherrsecs as ES_P,
       path454.sonetpathseverrsecs as SES_P,
       path454.sonetpathunavailablesecs as UAS_P,
       path454.sonetpathfc as FC_P,
       GETPMPARAMETERVALUE(path454.ppjcpdet,line454.ppjcpdet) as PPJC_PDET,
       GETPMPARAMETERVALUE(path454.npjcpdet,line454.npjcpdet) as NPJC_PDET,
       GETPMPARAMETERVALUE(path454.ppjcpgen,line454.ppjcpgen) as PPJC_PGEN,
       GETPMPARAMETERVALUE(path454.npjcpgen,line454.npjcpgen) as NPJC_PGEN,
       GETPMPARAMETERVALUE(path454.pjdiff,line454.pjdiff) as PJ_DIFF,
       GETPMPARAMETERVALUE(path454.pjpsec,line454.pjpsec) as PJP_SEC,
       GETPMPARAMETERVALUE(path454.pjnsec,line454.pjnsec) as PJN_SEC,
       path454.StsPathWidth,
       mergedinvaliditylist(line454.pjnsec,line454.ThresholdList,path454.ThresholdList) as
ThresholdList,
       path454.IS24H as IS24H,
       path454.ModelType,
       path454.ObjectType,
       path454.StrObjInstance,
       mergedinvaliditylist(line454.pjnsec,line454.invaliditylist,path454.invaliditylist)
as InvalidityList,
```

```
       path454.rowid as Myrow,
    ne.NEAliasID as NEAliasID,
    path454.StrCorbaTidAid
FROM   ONS15454_SONET_PATH_PM_TABLE path454,
       ONS15454_SONET_LINE_PM_TABLE line454,
       ne_info_table ne,
       module_type_table en
WHERE
       (path454.Nedbaccessid = ne.nedbaccessid and
        path454.ModuleType   = en.ModuleType ) and
       (line454.Nedbaccessid(+) = path454.nedbaccessid and
        line454.moduletype(+)   =path454.ModuleType   and
        line454.timestamp(+) = path454.Timestamp and
        line454.physicalloc(+) = path454.physicalloc and
        line454.IS24H(+)      = path454.IS24H ) and
    ne.nemodeltype      = en.modeltype;
```

2.2 Prime Optical Base Views

2.2.1 Active_Alarm_View_With_Ack

```
CREATE OR REPLACE VIEW active_alarm_view_with_ack AS
SELECT
       aa.ActiveAlarmSeverity AS severityIcon,
       ack.AckIndex AS AckIcon,
       aa.ActiveAlarmComment AS noteIcon,
       aa.ActiveAlarmServEff,
       aa.ActiveAlarmIndex AS ID,
       ni.NESysID AS NetworkElement,
       aa.PhysicalLoc,
       getModuleName(aa.NEDBAccessID, aa.ModuleType, en.EventType) AS ModuleType,
       getInterfaceId(aa.nedbaccessid,aa.ModuleOrIfIndex,aa.ObjectType,en.EnumString) AS
InterfaceID,
       en.EnumString AS Type,
       aa.ActiveAlarmSeverity AS Severity,
       aa.ActiveAlarmTimeStamp AS RaisedTimeStamp,
       aa.ClearAlarmTimeStamp AS ClearTimeStamp,
       aa.ActiveAlarmAdditionalInfo AS Description,
       ack.AckUserName AS AckUser,
       decode(to_char(ack.AckTimeStamp,'DD-MON-YYYY'),'01-JAN-
0001',aa.ClearAlarmTimeStamp,
                                                        '','',
                                                           ack.acktimestamp) AS
AckTimeStamp,
       aa.ActiveAlarmComment AS NoteDescription,
       aa.NEDBAccessID,
       aa.nealarmtimestamp,
       ack.IsAutoAck,
       aa.AlarmStatus,
       aa.NeAlarmClearTimestamp,
       aa.AlarmSeqNum,
       GetPortName(aa.nedbaccessid, aa.physicalloc, aa.ModuleorIfindex, aa.moduletype) as
PortName,
       aa.ModuleOrIfIndex as IntegerInterfaceid,
       aa.ExternalCondition,
       aa.ActiveAlarmType,
       en.eventtype,
       aa.rowid AS myrow,
       aa.ModelType,
       aa.ObjectType,
       aa.StrObjInstance,
       ni.NEAliasID as NEAliasID
    decode(to_char(nvl(aa.Direction,'')),'','N/A',aa.Direction) AS Direction,
       decode(to_char(nvl(aa.Location,'')),'','N/A',aa.Location) AS Location

FROM
       ne_info_table ni,
       active_alarm_table aa,
       alarm_event_type_enum_table en,
       ack_table ack
WHERE
       aa.NEDBAccessID    = ni.NEDBAccessID AND
```

```
       aa.ActiveAlarmType = en.EnumIndex    AND
       aa.AckIndex        = ack.AckIndex    AND
       ((aa.alarmstatus=1 and aa.ackindex=0) or
       aa.alarmstatus=0 )
union all
SELECT
       aa.ActiveAlarmSeverity AS severityIcon,
       ack.AckIndex AS AckIcon,
       aa.ActiveAlarmComment AS noteIcon,
       aa.ActiveAlarmServEff,
       aa.ActiveAlarmIndex AS ID,
       (select activevalue from ctm_config_table where sectionname='transportManager' and
propertyname='ems-sys-id') as  NetworkElement,
       aa.PhysicalLoc,
       getModuleName(aa.NEDBAccessID, aa.ModuleType, en.EventType) AS ModuleType,
       getInterfaceId(aa.nedbaccessid,aa.ModuleOrIfIndex,aa.ObjectType,en.EnumString) AS
InterfaceID,
       en.EnumString AS Type,
       aa.ActiveAlarmSeverity AS Severity,
       aa.ActiveAlarmTimeStamp AS RaisedTimeStamp,
       aa.ClearAlarmTimeStamp AS ClearTimeStamp,
       aa.ActiveAlarmAdditionalInfo AS Description,
       ack.AckUserName AS AckUser,
       decode(to_char(ack.AckTimeStamp,'DD-MON-YYYY'),'01-JAN-
0001',aa.ClearAlarmTimeStamp,
                                                        '','',
                                                           ack.acktimestamp) AS
AckTimeStamp,
       aa.ActiveAlarmComment AS NoteDescription,
       aa.NEDBAccessID,
       to_date('') AS NEAlarmTimestamp,
       ack.IsAutoAck,
       aa.AlarmStatus,
       aa.NEAlarmClearTimestamp,
       aa.AlarmSeqNum,
       '' as PortName,
       aa.ModuleOrIfIndex   as IntegerInterfaceId,
       aa.ExternalCondition,
       aa.ActiveAlarmType,
       en.eventtype,
       aa.rowid AS myrow,
       aa.ModelType,
       aa.ObjectType,
       aa.StrObjInstance,
(select activevalue from ctm_config_table where sectionname='transportManager' and
propertyname='ems-sys-id') as NEAliasID
       decode(to_char(nvl(aa.Direction,'')),'','N/A',aa.Direction) AS Direction,
       decode(to_char(nvl(aa.Location,'')),'','N/A',aa.Location) AS Location
FROM
       active_alarm_table aa,
       alarm_event_type_enum_table en,
       ack_table ack
Where aa.ActiveAlarmType = en.EnumIndex   AND
      aa.AckIndex        = ack.AckIndex   AND
      ((aa.alarmstatus=1 and aa.ackindex=0) or
       aa.alarmstatus=0 )                and
      aa.nedbaccessid=0;
```

2.2.2 Admin_Job_View

```
create or replace view Admin_Job_View as
select
       ajt.jobid,
       ajt.taskid,
       ajt.jobtype,
       ajt.jobowner,
       ni.nesysid,
       ajt.jobstatus,
       ajt.creationtime,
       ajt.scheduledtime,
       ajt.starttime,
       ajt.endtime,
       ajt.usercomments,
       ajt.additionalinfo,
```

```
        ajt.nedbaccessid,
        ni.nemodeltype,
        ajt.rowid as myrow,
        ajt.operation_info,
        ajt.display_info,
        ni.NEAliasID
from
        Admin_Job_Table ajt,
        ne_info_Table ni
where
        ajt.nedbaccessid = ni.nedbaccessid;
```

2.2.3 Alarm_Count_View (Deprecated)

```
create or replace view alarm_count_view as
select
       mn.nedbaccessid,
       mn.minoralarmcount,
       mj.majoralarmcount,
       cr.criticalalarmcount,
       wr.warningalarmcount,
       cl.clearedalarmcount,
       ack.ackalarmcount
  from (select nedbaccessid, totalalarmcount as majoralarmcount
          from alarm_count_table
         where alarmcounttype=6) mj,
       (select nedbaccessid, totalalarmcount as minoralarmcount
          from alarm_count_table
         where alarmcounttype=5) mn,
       (select nedbaccessid, totalalarmcount as criticalalarmcount
          from alarm_count_table
         where alarmcounttype=7) cr,
       (select nedbaccessid, totalalarmcount as warningalarmcount
          from alarm_count_table
         where alarmcounttype=4) wr,
       (select nedbaccessid, totalalarmcount as clearedalarmcount
          from alarm_count_table
         where alarmcounttype=-1) cl,
       (select nedbaccessid, totalalarmcount as ackalarmcount
          from alarm_count_table
         where alarmcounttype=-2) ack
  where mj.nedbaccessid=mn.nedbaccessid
    and mj.nedbaccessid=cr.nedbaccessid
    and mj.nedbaccessid=wr.nedbaccessid
    and mj.nedbaccessid=cl.nedbaccessid
    and mj.nedbaccessid=ack.nedbaccessid;
```

2.2.4 Alarm_Log_View

```
create or replace view alarm_log_view AS
SELECT
       ae.AlarmEventSeverity AS severityIcon,
       0 as AckIcon,
       ''                as NoteIcon,
       ae.ServiceEffecting,
       ae.EventIndex as ID,
       nvl(ni.NESysID,'CTM')  AS NetworkElement,
       ae.PhysicalLoc as Physicalloc,
       GetModuleName(ae.nedbaccessid, ae.moduletype, en.EventType) AS ModuleType,
       GetInterfaceId(ae.nedbaccessid, ae.ModuleOrIfIndex, ae.ObjectType, en.EnumString)
AS InterfaceID,
       en.EnumString AS Type,
       ec.EnumString AS Cause,
       ae.AlarmEventSeverity AS severity,
       ae.AlarmEventTimeStamp AS RaisedTimeStamp,
       to_date('')  as ClearTimestamp,
       ae.OtherInfo AS Description,
       '' as Ackuser,
       to_date('')  as AckTimestamp,
       ''           as NoteDescription,
       ae.Nedbaccessid as Nedbaccessid,
       decode(ae.nedbaccessid,0,to_date(''),
       ae.NEEventtimestamp) as NeAlarmTimestamp,
```

```
       -1 as Alarmstatus,
       2 as IsAutoAck,
       to_char('Event')      as EventOrAlarm,
       to_date('') as NEAlarmClearTimestamp,
       ''          as CTMDuration,
       GetPortName(ae.nedbaccessid, ae.physicalloc, ae.ModuleOrIfindex, ae.Moduletype) as
PortName,
       ae.ModuleOrIfIndex  as IntegerInterfaceId,
       en.eventtype,
       nvl(ni.NEAliasID,'CTM')  as NEAliasID,
       ae.rowid AS myrow,
       ae.ModelType,
       ae.ObjectType,
       ae.StrObjInstance,
       decode(to_char(nvl(ae.Direction,'')),'','N/A',ae.Direction) AS
Direction,              decode(to_char(nvl(ae.Location,'')),'','N/A',ae.Location) AS
Location,
    ''      as ExternalCondition
FROM
       ne_info_table ni,
       alarm_event_table ae,
       alarm_event_type_enum_table en,
       alarm_event_cause_table ec
WHERE
       ae.NEDBAccessID    = ni.NEDBAccessID(+) AND
       ae.AlarmEventType  = en.EnumIndex    AND
       ae.AlarmEventCause = ec.EnumIndex
UNION
SELECT
      aa.ActiveAlarmSeverity   as SeverityIcon,
      aa.AckIndex        as AckIcon,
      aa.ActiveAlarmComment       as NoteIcon,
      aa.ActiveAlarmServeff  as ServiceEffecting,
      aa.ActiveAlarmIndex    as ID,
      nvl(ni.NESysID,(select activevalue from ctm_config_table where
sectionname='transportManager' and propertyname='ems-sys-id'))  as NetworkElement,
      aa.Physicalloc    as Physicalloc,
      GetModulename(aa.nedbaccessid, aa.moduletype, en.EventType)  as ModuleType,
      GetInterfaceId(aa.nedbaccessid, aa.ModuleOrIfIndex, aa.ObjectType, en.EnumString)
as InterfaceId,
      en.EnumString           as Type,
      ''              as Cause,
      aa.ActiveAlarmSeverity       as Severity,
      aa.ActiveAlarmTimeStamp  as RaisedTimestamp,
      aa.ClearAlarmTimeStamp   as ClearTimestamp,
      aa.ActiveAlarmAdditionalInfo      as Description,
      ack.AckUserName          as AckUser,
      decode(to_char(ack.AckTimeStamp,'DD-MON-YYYY'),'01-JAN-0001',aa.ClearAlarmTimeStamp,
                                                        '','',
                                                          ack.acktimestamp) AS AckTimeStamp,
      aa.ActiveAlarmComment  as NoteDescription,
      aa.NEDBAccessID     as Nedbaccessid,
      decode(aa.nedbaccessid,0,to_date(''),
                               aa.NeAlarmTimestamp) as NeAlarmtimestamp,
      aa.AlarmStatus      as AlarmStatus,
      ack.IsAutoAck        as IsAutoAck,
      to_char('Alarm')          as EventOrAlarm,
      aa.NEAlarmClearTimestamp  as NEAlarmClearTimestamp,
      ctmduration(aa.ClearAlarmTimeStamp-aa.ActiveAlarmTimeStamp) as CTMDuration,
      GetPortName(aa.nedbaccessid, aa.physicalloc, aa.ModuleOrIfindex, aa.moduletype) as
PortName,
      aa.ModuleOrIfIndex   as IntegerInterfaceId,
      en.eventtype,
nvl(ni.NEAliasID,(select activevalue from ctm_config_table where
sectionname='transportManager' and propertyname='ems-sys-id'))  as NEAliasID,
      aa.rowid            as Myrow,
      aa.ModelType,
      aa.ObjectType,
      aa.StrObjInstance
    decode (to_char(nvl(aa.Direction,'')),'','N/A',aa.Direction) AS Direction,
      decode (to_char(nvl(aa.Location,'')),'','N/A',aa.Location) AS Location,
    aa.ExternalCondition
From
       ne_info_table ni,
       active_alarm_table aa,
       alarm_event_type_enum_table en,
```

```
       ack_table ack
Where
       aa.NEDBAccessID    = ni.NEDBAccessID(+) AND
       aa.ActiveAlarmType = en.EnumIndex    AND
       aa.AckIndex        = ack.AckIndex    AND
       aa.alarmstatus     = 1               AND
       aa.ackindex        <> 0;
```

2.2.5 Audit_Log_View

```
create or replace view audit_log_view AS
SELECT nvl(n.nesysid,'CTM') as NESYSID,
       nvl(t.username, 'Internal') as username,
       t.timestamp,
       t.module,
       t.classname,
       0 as linenumber,
       (t.actionname || t.logmessage) as LogMessage,
       nvl(t.nedbaccessid,0)  as nedbaccessid,
       t.service,
       nvl(n.NEAliasID,'CTM') as NEAliasID,
       t.rowid as Myrow
FROM   transaction_log_table t,
       ne_info_table         n
WHERE  t.nedbaccessid=n.nedbaccessid(+);
```

2.2.6 BLSR_Table_View

```
create or replace view BLSR_TABLE_VIEW as
SELECT
        blsr.BLSRID as BLSRID,
       blsr.RINGID as RINGID,
        blsr.RINGTYPE as RINGTYPE,
        blsr.LINERATE as LINERATE,
        blsr.STATUS as STATUS,
        blsr.RINGREVERSION as RINGREVERSION,
        blsr.SPANREVERSION as SPANREVERSION,
        blsr.NODES as NODES,
        node.NODEID as NODEID,
       blsr.rowid as myrow
FROM    BLSR_TABLE blsr,
        BLSR_NODE node
WHERE
        blsr.BLSRID = node.BLSRID;
```

2.2.7 Children_View

```
CREATE OR REPLACE VIEW children_view AS
SELECT
        eit.NEDBAccessId,
        eit.PhysicalLoc,
        eit.IfIndex,
        eit.ModuleType,
        mm.ParentIndex,
        eit.rowid as myrow,
        eit.ModelType,
        eit.ObjectIndex,
        eit.StrObjInstance
FROM
        eqpt_info_table eit,
        objecttypetable mm
WHERE
        eit.ObjectIndex = mm.ObjectIndex AND eit.ModelType = mm.ModelType;
```

2.2.8 Circuit_Ctpspan_View

```
create or replace view circuit_ctpspan_view as
select
```

```
       span.cktnodeid,
       span.cktuniqueid,
       ctpsrc.cktctpnodeid as spansrcnodeid,
       ctpdest.cktctpnodeid as spandestnodeid,
       span.cktlinkid as cktlinkid
  from circuit_ctp_tbl ctpsrc,
       circuit_ctp_tbl ctpdest,
       circuit_span_bs_tbl   span
 where ctpsrc.ctpaccessid=span.cktspansrcctpaccessid and
       ctpdest.ctpaccessid=span.cktspandstctpaccessid;
```

2.2.9 Circuit_Ctp_Drop_View

```
create or replace view circuit_ctp_drop_view as
select ne.nesysid,
       ctp.cktnodeid,
       ctp.cktuniqueid,
       ctp.cktctpnodeid,
       ctp.cktctpPhysicalLoc,
       ctp.CktctpIfIndex,
       ctp.cktctpModuleType,
       ctp.cktctpDropProt,
       ctp.CktctpPathProt,
       ctp.cktctpnedbaccessid,
       ctp.CktctpModelType,
       ctp.CktctpObjectType
from ne_info_table ne,
     circuit_ctp_tbl ctp
where  ne.nedbaccessid=ctp.CKTCTPNeDBAccessId
  and  ctp.ctp_type=2;
```

2.2.10 Circuit_Ctp_Sec_View

```
create or replace view circuit_ctp_sec_view as
select ne.nesysid,
       ctp.cktnodeid,
       ctp.cktuniqueid,
       ctp.cktctpnodeid,
       ctp.cktctpPhysicalLoc,
       ctp.CktctpIfIndex,
       ctp.cktctpModuleType,
       ctp.cktctpDropProt,
       ctp.cktctpnedbaccessid,
       ctp.CktctpModelType,
       ctp.CktctpObjectType
from ne_info_table ne,
     circuit_ctp_tbl ctp
where  ne.nedbaccessid=ctp.CKTCTPNeDBAccessId
  and  ctp.ctp_type=1;
```

2.2.11 Circuit_Ctp_Src_View

```
create or replace view circuit_ctp_src_view as
select ne.nesysid,
       ctp.cktnodeid,
       ctp.cktuniqueid,
       ctp.cktctpnodeid,
       ctp.cktctpPhysicalLoc,
       ctp.CktctpIfIndex,
       ctp.cktctpModuleType,
       ctp.cktctpDropProt,
       ctp.cktctpnedbaccessid,
       ctp.CktctpModelType,
       ctp.CktctpObjectType
from ne_info_table ne,
     circuit_ctp_tbl ctp
where  ne.nedbaccessid=ctp.CKTCTPNeDBAccessId
  and  ctp.ctp_type=0;
```

2.2.12 Circuit_Ctp_View

```
CREATE OR REPLACE VIEW CIRCUIT_CTP_VIEW AS
SELECT
        c.CTPACCESSID as CTPACCESSID,
        c.CTP_TYPE as CTP_TYPE,
        c.CKTNODEID as CKTNODEID,
        c.CKTUNIQUEID as CKTUNIQUEID,
        c.CKTCTPNODEID as CKTCTPNODEID,
        c.CKTCTPNEDBACCESSID as CKTCTPNEDBACCESSID,
        c.CKTCTPMODULETYPE as CKTCTPMODULETYPE,
        c.CKTCTPIFINDEX as CKTCTPIFINDEX,
        c.CKTCTPOBJECTTYPE as CKTCTPOBJECTTYPE,
        c.CKTCTPPHYSICALLOC as CKTCTPPHYSICALLOC,
        c.CKTCTPMODELTYPE as CKTCTPMODELTYPE,
        c.CKTCTPDROPPROT as CKTCTPDROPPROT,
        c.CKTCTPPATHPROT as CKTCTPPATHPROT,
        c.CKTCTPDROPPROTTYPE as CKTCTPDROPPROTTYPE,
        c.CKTCTPPORTNAME as CKTCTPPORTNAME,
        c.CKTCTPADDITIONALINFO as CKTCTPADDITIONALINFO,
        c.NPID as NPID,
        ckt.ROWID as MYROW
from    circuit_ctp_tbl c,
        circuit_bs_tbl ckt
where
        c.CktNodeId = ckt.CktNodeId(+) and
        c.CktUniqueId = ckt.CktUniqueId(+);
```

2.2.13 Circuit_Ctp_XC_View

```
CREATE OR REPLACE VIEW CIRCUIT_CTP_XC_VIEW AS
SELECT  DISTINCT
        b.CKTNAME as CktName,
        a.CKTNODEID as NodeId,
         a.CKTUNIQUEID as UniqueId,
        a.CKTCTPNODEID as NENodeId,
        a.CKTCTPNEDBACCESSID as NEDBAccessId,
        b.CKTDIRECTION as CktDirection,
        b.CKTTYPE as CktType,
        b.CKTSIZE as CktSize

from    circuit_ctp_tbl a,
        circuit_bs_tbl b,
        circuit_ctp_tbl z
where
        a.CTPACCESSID  != z.CTPACCESSID and
         a.CKTCTPNODEID =  z.CKTCTPNODEID and
        a.CKTNODEID    =  z.CKTNODEID and
        a.CKTUNIQUEID  =  z.CKTUNIQUEID and
        a.CKTNODEID    =  b.CKTNODEID and
        a.CKTUNIQUEID  =  b.CKTUNIQUEID and
        b.CKTSTATE     != 12 and
        b.CKTTYPE not in (5, 6, 21, 22);
```

2.2.14 Circuit_Node_GUI_View

```
CREATE OR REPLACE VIEW circuit_node_gui_view AS
WITH
   CTP AS ( select min(ctp_tb.CTPACCESSID)  from circuit_ctp_tbl ctp_tb, circuit_bs_tbl
circ where ctp_tb.CktNodeId = circ.CktNodeId and ctp_tb.CktUniqueId = circ.CktUniqueId and
ctp_tb.ctp_type = 2 group by  circ.cktname, circ.CktUniqueID
          )
SELECT
        c.CktName as Name,
        c.CktType as Type,
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
```

```
        (select nesysid from ne_info_table where nedbaccessid=src.CKTCTPNeDBAccessId)  as
SrcNodeid,
        src.cktctpnodeid as cktsrcnodeid,
        nvl(getModulename1(src.cktctpnodeid, src.cktctpModuleType),'SYSTEM')  as
SrcModuleType,
        src.cktctpPhysicalLoc as SrcPhysicalLoc,
        src.CktctpIfIndex as SrcInterfaceId,
        nvl(getModulename1(sec.cktctpnodeid, sec.cktctpModuleType),'SYSTEM') as
SecSrcModuleType,
        sec.CktctpPhysicalLoc as SecSrcPhysicalLoc,
        sec.CktctpIfIndex as SecSrcInterfaceId,
        collapseddestnodename(c.CKTNODEID,c.CKTUNIQUEID) as destnodename,
        collapsedmodulename(c.CKTNODEID,c.CKTUNIQUEID)as cktdestModuleName,
        collapsedphy(c.CKTNODEID,c.CKTUNIQUEID) as DestPhysicalLoc,
        collapsedifindex(c.CKTNODEID,c.CKTUNIQUEID) as DestInterfaceId,
        c.CktIsMonitor as Ismonitor,
        src.CktctpDropProt as SrcDropProt,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        c.CustomerId as CustomerId,
        c.ServiceId as ServiceId,
        dest.cktctpnodeid as DestNodeId,
        c.CktEnhancedState,
        c.CktProtectionType,
        dest.CktctpPathProt  as DestPathProtection,
        c.cktComment as CircuitNote,
        get_nesysid(sec.CKTCTPNeDBAccessId) as SecSrcNodeid,
        sec.cktctpnodeid as cktsecsrcnodeid,

eqptinfoport(src.cktctpnedbaccessid,src.cktctpphysicalloc,0,src.cktctpmoduletype)
as CktSrcPortName,

eqptinfoport(sec.cktctpnedbaccessid,sec.cktctpphysicalloc,0,sec.cktctpmoduletype)
as CktSecSrcPortName,

collapsedeqptportname(c.CKTNODEID,c.CKTUNIQUEID) as CktDestPortName,
         ( select count(*)  from circuit_vlan_tbl vlan where vlan.cktnodeid=c.cktnodeid
and vlan.cktuniqueid=c.cktuniqueid ) as NoOfVlans,  /* *@INDEX A 5 */
        nvl(vcg.VCGDirection,-1) as VCGDirection,
        nvl(vcg.ParentCktNodeId,-1) as ParentCktNodeId,
        nvl(vcg.ParentCktUniqueId,-1) as ParentCktUniqueId,
        c.CktOchncChannel,
        c.CktOchncDir,
        src.CktctpModelType as SrcModelType,
        src.CktctpObjectType as SrcObjectType,
        sec.CktctpModelType as CktSecSrcModelType,
        sec.CktctpObjectType as SecSrcObjectType,
        collapsedmodeltype(c.CKTNODEID,c.CKTUNIQUEID) as DestModelType,
        collapsedobjectindex(c.CKTNODEID,c.CKTUNIQUEID) as DestObjectType,
        c.isDuplicateName,
        nvl(getPortModuleName(src.cktctpnodeid, src.cktctpModuleType),'SYSTEM')  as
SrcPortModuleType,
        nvl(getPortModuleName(sec.cktctpnodeid, sec.cktctpModuleType),'SYSTEM') as
SecSrcPortModuleType,
        collapsedportmodulename(c.CKTNODEID,c.CKTUNIQUEID)as cktdestPortModuleName,
        c.rowid as myrow,
        dest.cktctpPhysicalLoc as CktDestPhysicalLoc,
        nvl(getPortModuleName(dest.cktctpnodeid, dest.cktctpModuleType),'SYSTEM') as
DestPortModuleType,
        nvl(vcg.VCATSTATE,-1) as VcatState,
        src.cktctpModuleType as CktSrcModuleType,
        (select count(*) from circuit_span_bs_tbl where cktnodeid=c.cktnodeid and
cktuniqueid=c.cktuniqueid) as NoOfSpans,
       c.CktAliasName as alias_name,
        c.CktAdditionalInfo,
    c.isOpenVcat,
        c.isOverlay,
       get_nesysid(dest.CKTCTPNeDBAccessId) as DestSysId,
       ( select count(DISTINCT NPID) from circuit_node_bs_tbl where cktnodeid=c.cktnodeid
and cktuniqueid=c.cktuniqueid ) as npnum,
        ( select count(*) from circuit_vcg_tbl where parentcktnodeid=c.cktnodeid and
parentcktuniqueid=c.cktuniqueid ) as vcatmemnum
gmpls.actp_threshold as ActpValue,
gmpls.opt_value      as OpticalValue
from    circuit_bs_tbl c,
```

```
        circuit_ctp_tbl src,
        circuit_ctp_tbl sec,
        circuit_ctp_tbl dest,
        circuit_vcg_tbl vcg,
       gmpls_value_table gmpls
where
        c.CktNodeId             = src.CktNodeId(+)  and
        c.CktUniqueId           = src.CktUniqueId(+) and
       gmpls.cktnodeid           = src.CktNodeId and
       gmpls.cktuniqueid         = src.CktUniqueId and
    0 = src.ctp_type(+) and
        c.CktNodeId             = sec.CktNodeId(+) and
        c.CktUniqueId           = sec.CktUniqueId(+) and
    1 = sec.ctp_type(+) and
        c.CktNodeId             = dest.CktNodeId(+) and
        c.CktUniqueId           = dest.CktUniqueId(+) and
    2 = dest.ctp_type(+) and
    ((dest.ctpaccessid IS NULL) OR (dest.cktctpadditionalinfo = 'FIRSTDROP'))
        and
        c.CktNodeId             = vcg.CktNodeId(+) and
        c.CktUniqueId           = vcg.CktUniqueId(+) and
        c.cktstate             !=12;
```

2.2.15 Circuit_Node_GUI_View_CTM46

```
CREATE OR REPLACE VIEW circuit_node_gui_view_ctm46 AS
SELECT
        c.CktName as Name,
        c.CktType as Type,
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
        (select nesysid from ne_info_table where nenodeid=c.cktsrcnodeid and
c.cktsrcnodeid !=0 )  as SrcNodeid,
        c.cktsrcnodeid,
        nvl(getSlotModuleName(c.cktsrcnodeid, c.CktSrcModuleType),'SYSTEM')  as
SrcModuleType,
        c.CktSrcPhysicalLoc as SrcPhysicalLoc,
        c.CktSrcIfIndex as SrcInterfaceId,
        nvl(getSlotModuleName(c.cktsecsrcnodeid, c.CktSecSrcModuleType),'SYSTEM') as
SecSrcModuleType,
        c.CktSecSrcPhysicalLoc as SecSrcPhysicalLoc,
        c.CktSecSrcIfIndex as SecSrcInterfaceId,
        collapseddestnodename(c.CKTNODEID,c.CKTUNIQUEID) as destnodename,
        collapsedmodulename(c.CKTNODEID,c.CKTUNIQUEID)as cktdestModuleName,
        collapsedphy(c.CKTNODEID,c.CKTUNIQUEID) as DestPhysicalLoc,
        collapsedifindex(c.CKTNODEID,c.CKTUNIQUEID) as DestInterfaceId,
        c.CktIsMonitor as Ismonitor,
        c.CktSrcDropProt as SrcDropProt,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        c.CustomerId as CustomerId,
        c.ServiceId as ServiceId,
        dest.cktdestnodeid as DestNodeId,
        spantbl.CktSpanSrcNodeId,
        spantbl.CktSpanDestNodeId,
        spantbl.cktlinkid,
        c.CktEnhancedState,
        c.CktProtectionType,
        dest.CktDestPathProt  as DestPathProtection,
        c.cktComment as CircuitNote,
        (select nesysid from ne_info_table where nenodeid=c.cktsecsrcnodeid and
c.cktsecsrcnodeid !=0 )  as SecSrcNodeid,
        c.cktsecsrcnodeid,

eqptinfoport(getNEDBACCESSID(c.CKTSRCNODEID),c.CKTSRCPHYSICALLOC,0,c.CKTSRCMODULETYPE ) as
CktSrcPortName,

eqptinfoport(getNEDBACCESSID(c.CKTSECSRCNODEID),c.CKTSECSRCPHYSICALLOC,0,c.CKTSECSRCMODULE
TYPE ) as CktSecSrcPortName,
        collapsedeqptportname(c.CKTNODEID,c.CKTUNIQUEID) as CktDestPortName,
```

```
       (select count(*) from circuit_vlan_tbl where cktnodeid=c.cktnodeid and
cktuniqueid=c.cktuniqueid) as NoOfVlans,
        nvl(vcg.VCGDirection,-1) as VCGDirection,
        nvl(vcg.ParentCktNodeId,-1) as ParentCktNodeId,
        nvl(vcg.ParentCktUniqueId,-1) as ParentCktUniqueId,
        c.CktOchncChannel,
        c.CktOchncDir,
        c.CktSrcModelType as SrcModelType,
        c.CktSrcObjectType as SrcObjectType,
        c.CktSecSrcModelType as CktSecSrcModelType,
        c.CktSecSrcObjectType as SecSrcObjectType,
        collapsedmodeltype(c.CKTNODEID,c.CKTUNIQUEID) as DestModelType,
        collapsedobjectindex(c.CKTNODEID,c.CKTUNIQUEID) as DestObjectType,
        c.isDuplicateName,
        nvl(getPortModuleName(c.cktsrcnodeid, c.CktSrcModuleType),'SYSTEM')  as
SrcPortModuleType,
        nvl(getPortModuleName(c.cktsecsrcnodeid, c.CktSecSrcModuleType),'SYSTEM') as
SecSrcPortModuleType,
        collapsedportmodulename(c.CKTNODEID,c.CKTUNIQUEID)as cktdestPortModuleName,
        c.rowid as myrow,
        dest.cktdestphysicalloc as CktDestPhysicalLoc,
        nvl(getPortModuleName(dest.cktdestnodeid, dest.CktDestModuleType),'SYSTEM') as
DestPortModuleType,
        spantbl.CktSpanSrcPhysicalLoc,
        spantbl.CktSpanDestPhysicalLoc
From
        circuit_tbl c,
        circuit_vcg_tbl vcg,
        circuit_dest_tbl dest,
        circuit_span_tbl spantbl
where
        c.CktNodeId             = dest.CktNodeId(+)  and
        c.CktUniqueId           = dest.CktUniqueId(+) and
        c.CktNodeId             = spantbl.CktNodeId(+) and
        c.CktUniqueId           = spantbl.CktUniqueId(+) and
        c.CktNodeId             = vcg.CktNodeId(+) and
        c.CktUniqueId           = vcg.CktUniqueId(+) and
        c.cktState              != 12 ;
```

2.2.16 Circuit_Node_View

```
CREATE OR REPLACE VIEW circuit_node_view AS
SELECT
        c.CktName as Name,
        c.CktType as Type,
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
        (select nesysid from ne_info_table where nenodeid=c.cktsrcnodeid and
c.cktsrcnodeid !=0)  as SrcNodeid,
        nvl(getModulename1(c.cktsrcnodeid, c.CktSrcModuleType),'SYSTEM')  as
SrcModuleType,
        c.CktSrcPhysicalLoc as SrcPhysicalLoc,
        c.CktSrcModelType as SrcModelType,
        c.CktSrcIfIndex as SrcInterfaceId,
        c.CktSrcObjectType as SrcObjectType,
        nvl(getModulename1(c.cktsecsrcnodeid, c.CktSecSrcModuleType),'SYSTEM') as
SecSrcModuleType,
        c.CktSecSrcPhysicalLoc as SecSrcPhysicalLoc,
        c.CktSecSrcModelType as SecSrcModelType,
        c.CktSecSrcIfIndex as SecSrcInterfaceId,
        c.CktSecSrcObjectType as SecSrcObjectType,
        (select nesysid from ne_info_table where nenodeid=dest.cktdestnodeid and
dest.cktdestnodeid != 0) as destnodeid,
        nvl(getModulename1(dest.cktdestnodeid, dest.CktDestModuleType),'SYSTEM') as
DestModuleType,
        dest.CktDestPhysicalLoc as DestPhysicalLoc,
        dest.CktDestIfIndex as DestInterfaceId,
        dest.CktDestObjectType as DestObjectType,
        c.CktIsMonitor as Ismonitor,
        c.CktSrcDropProt as SrcDropProt,
        dest.CktDestDropProt as DestDropProt,
        dest.CktDestPathProt as DestPathProt,
```

```
        0  as Vlanid,
        c.CktName as vlanname,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        c.CustomerId as CustomerId,
        c.ServiceId as ServiceId,
        spantbl.CktSpanSrcNodeId,
        spantbl.CktSpanDestNodeId,
        spantbl.cktlinkid,
        c.CktEnhancedState,
        c.CktProtectionType,
        c.rowid as myrow
from
        circuit_tbl c,
        circuit_dest_tbl dest,
        circuit_span_tbl spantbl
where
        c.CktNodeId             = dest.CktNodeId(+)  and
        c.CktUniqueId           = dest.CktUniqueId(+) and
        c.CktNodeId             = spantbl.CktNodeId(+) and
        c.CktUniqueId           = spantbl.CktUniqueId(+) and
        c.cktState              != 12 ;
```

2.2.17 Circuit_Span_View

```
create or replace view Circuit_Span_View as
select
        circuit.CktName as Name,
        circuit.CktType as Type,
        circuit.CktSize as CktSize,
        ne_src.NESYSID as SrcNodeId,
        nvl(getModuleName2( ctp_src.CKTCTPMODULETYPE, ctp_src.CKTCTPMODELTYPE ) ,'SYSTEM'
) as SrcModuleType,
        ctp_src.CKTCTPPHYSICALLOC as SrcPhysicalLoc,
        ctp_src.CKTCTPIFINDEX as SrcInterfaceId,
        ne_dst.NESYSID as DestNodeId,
        nvl(getModuleName2( ctp_dst.CKTCTPMODULETYPE, ctp_dst.CKTCTPMODELTYPE ) ,'SYSTEM'
) as DestModuleType,
        ctp_dst.CKTCTPPHYSICALLOC as DestPhysicalLoc,
        ctp_dst.CKTCTPIFINDEX as InterfaceId,
        spancircuit.CKTSPANSRCSTATE,
        spancircuit.CKTSPANDESTSTATE,
        spancircuit.IsCktForwarding as Forwarding,
        spancircuit.IsCktSpanInUPSR as InUpsr,
        spancircuit.IsCktSpanActive as active,
        spancircuit.CktSpanProtOp as ProtOp,
        spancircuit.CktNodeId as NodeId,
        spancircuit.CktUniqueId as UniqueId,
        ne_src.NEDBACCESSID as NEDBAccessid,
        ne_dst.NEDBACCESSID as DestNEDBAccessid,
        spancircuit.CktLinkId as linkid,
        ctp_src.CKTCTPNODEID as SrcNENodeId,
        ctp_dst.CKTCTPNODEID as DestNENodeID,
        ctp_src.CKTCTPMODULETYPE as SrcNEModuleType,
        ctp_dst.CKTCTPMODULETYPE as DestNEModuleType,
        ctp_src.CKTCTPPORTNAME as SrcPortName,
        ctp_dst.CKTCTPPORTNAME as DestPortName,
        ctp_src.CKTCTPMODELTYPE as SrcModelType,
        ctp_src.CKTCTPOBJECTTYPE as SrcObjectType,
        ctp_dst.CKTCTPMODELTYPE as DestModelType,
        ctp_dst.CKTCTPOBJECTTYPE as DestObjectType,
        nvl(getPortModuleName2( ctp_src.CKTCTPMODULETYPE, ctp_src.CKTCTPMODELTYPE )
,'SYSTEM' ) as SrcPortModuleType,
        nvl(getPortModuleName2( ctp_dst.CKTCTPMODULETYPE, ctp_dst.CKTCTPMODELTYPE )
,'SYSTEM' ) as DestPortModuleType,
        spancircuit.rowid as myrow,
        circuit.CKTOCHNCCHANNEL as Channel
from
    CIRCUIT_BS_Tbl circuit,
    circuit_ctp_tbl ctp_src,
    circuit_ctp_tbl ctp_dst,
    circuit_span_BS_tbl spancircuit,
    ne_info_table ne_src,
```

```
    ne_info_table ne_dst
where
        ctp_src.CTPACCESSID = spancircuit.CKTSPANSRCCTPACCESSID
  and   ctp_dst.CTPACCESSID = spancircuit.CKTSPANDSTCTPACCESSID
  and   spancircuit.CktNodeId = circuit.CktNodeId
  and   spancircuit.CktUniqueId = circuit.CktUniqueId
  and   ne_src.nenodeid = ctp_src.CKTCTPNODEID
  and   ne_dst.nenodeid = ctp_dst.CKTCTPNODEID
  and   ((ctp_src.ctp_type = 3 and ctp_dst.ctp_type = 4);
```

2.2.18 Circuit_Span_View_CTM46

```
create or replace view Circuit_Span_View_ctm46 as
select
        circuit.CktName as Name,
        circuit.CktType as Type,
        circuit.CktSize as CktSize,
        neInfo1.NESysID as SrcNodeId,
        nvl(getSlotModuleName(spancircuit.cktspansrcnodeid,
spancircuit.CktspanSrcModuleType),'SYSTEM') as SrcModuleType,
        spancircuit.CktSpanSrcPhysicalLoc as SrcPhysicalLoc,
        spancircuit.CktSpanSrcIfIndex as SrcInterfaceId,
        neInfo2.NESysID as DestNodeId,
        nvl(getSlotModuleName(spancircuit.cktspandestnodeid,
spancircuit.CktspandestModuleType),'SYSTEM') as DestModuleType,
        spancircuit.CktSpanDestPhysicalLoc as DestPhysicalLoc,
        spancircuit.CktSpanDestIfIndex as InterfaceId,
        spancircuit.CktSpanSrcState as SrcState,
        spancircuit.CktSpanDestState as DestState,
        spancircuit.IsCktForwarding as Forwarding,
        spancircuit.IsCktSpanInUPSR as InUpsr,
        spancircuit.IsCktSpanActive as active,
        spancircuit.CktSpanProtOp as ProtOp,
        spancircuit.CktNodeId as NodeId,
        spancircuit.CktUniqueId as UniqueId,
        neinfo1.NEDBAccessid,
        neinfo2.NEDBAccessid as DestNEDBAccessid,
        spancircuit.CktLinkId as linkid,
        spancircuit.CktSpanSrcNodeId as SrcNENodeId,
        spancircuit.CktSpanDestNodeId as DestNENodeID,
        spancircuit.CktspanSrcModuleType as SrcNEModuleType,
        spancircuit.CktspandestModuleType as DestNEModuleType,

eqptinfoport(getNEDBACCESSID(spancircuit.CKTSPANSRCNODEID),spancircuit.CKTSPANSRCPHYSICALL
OC,0,spancircuit.CKTSPANSRCMODULETYPE) as SrcPortName,

eqptinfoport(getNEDBACCESSID(spancircuit.CKTSPANDESTNODEID),spancircuit.CKTSPANDESTPHYSICA
LLOC,0,spancircuit.CKTSPANDESTMODULETYPE) as DestPortName,
        spancircuit.CktSpanSrcModelType as SrcModelType,
        spancircuit.CktSpanSrcObjectType as SrcObjectType,
        spancircuit.CktSpanDestModelType as DestModelType,
        spancircuit.CktSpanDestObjectType as DestObjectType,
        nvl(getPortModuleName(spancircuit.cktspansrcnodeid,
spancircuit.CktspanSrcModuleType),'SYSTEM') as SrcPortModuleType,
        nvl(getPortModuleName(spancircuit.cktspandestnodeid,
spancircuit.CktspandestModuleType),'SYSTEM') as DestPortModuleType,
        spancircuit.rowid as myrow
from
        CIRCUIT_TBL circuit,
        NE_INFO_TABLE neInfo1,
        NE_INFO_TABLE neInfo2,
        circuit_span_tbl spancircuit
where
        spancircuit.CktNodeId = circuit.CktNodeId and
        spancircuit.CktUniqueId = circuit.CktUniqueId and
        spancircuit.CktSpanSrcNodeID = neInfo1.NENodeID and
        spancircuit.CktSpanDestNodeId = neInfo2.NENodeID;
```

2.2.19 Circuit_UPSR_View

```
CREATE OR REPLACE FORCE VIEW CTMANAGER.CIRCUIT_UPSR_VIEW
(
   CKTNAME,
```

```
   CKTNODEID,
   CKTUNIQUEID,
   UPSRNODEID,
   WCTPPHYLOC,
   WCTPIFINDEX,
   PCTPPHYLOC,
   PCTPIFINDEX,
   ACTIVEISWORKING,
   SWITCHSTATE
)
AS
   SELECT   ckt.CKTNAME AS CKTNAME,
            upsr.CKTNODEID AS CKTNODEID,
            upsr.CKTUNIQUEID AS CKTUNIQUEID,
            upsr.NODEID AS UPSRNODEID,
            upsr.WORKINGCTPPHYSICALLOC AS WCTPPHYLOC,
            upsr.WORKINGCTPIFINDEX AS WCTPIFINDEX,
            upsr.PROTECTCTPPHYSICALLOC AS PCTPPHYLOC,
            upsr.PROTECTCTPIFINDEX AS PCTPIFINDEX,
            upsr.CURRENTISWORKING AS ACTIVEISWORKING,
            upsr.SWITCHSTATE AS SWITCHSTATE

     FROM   circuit_upsr_bs_tbl upsr, circuit_bs_tbl ckt
    WHERE
       upsr.CktNodeId = ckt.CktNodeId AND
       upsr.CktUniqueId = ckt.CktUniqueId;
```

2.2.20 Circuit_View

```
create or replace view Circuit_View AS
select
        c.CktName as Name,
        c.CktType as Type,
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
        (select nesysid from ne_info_table where nenodeid=c.cktsrcnodeid and
c.cktsrcnodeid !=0)  as SrcNodeid,
        nvl(getModulename1(c.cktsrcnodeid, c.CktSrcModuleType),'SYSTEM')  as
SrcModuleType,
        c.CktSrcPhysicalLoc as SrcPhysicalLoc,
        c.CktSrcIfIndex as SrcInterfaceId,
        nvl(getModulename1(c.cktsecsrcnodeid, c.CktSecSrcModuleType),'SYSTEM') as
SecSrcModuleType,
        c.CktSecSrcPhysicalLoc as SecSrcPhysicalLoc,
        c.CktSecSrcIfIndex as SecSrcInterfaceId,
        (select nesysid from ne_info_table where nenodeid=dest.cktdestnodeid and
dest.cktdestnodeid !=0 ) as destnodeid,
        nvl(getModulename1(dest.cktdestnodeid, dest.CktDestModuleType),'SYSTEM') as
DestModuleType,
        dest.CktDestPhysicalLoc as DestPhysicalLoc,
        dest.CktDestIfIndex as DestInterfaceId,
        c.CktIsMonitor as Ismonitor,
        c.CktSrcDropProt as SrcDropProt,
        dest.CktDestDropProt as DestDropProt,
        dest.CktDestPathProt as DestPathProt,
        vlan.Cktvlanid  as Vlanid,
        vlan.cktvlanname  as Vlanname,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        c.CustomerId as CustomerId,
        c.ServiceId as ServiceId,
        dest.CKTDESTNODEID as DESTNENODEID,
        c.CKTSRCNODEID as SRCNENODEID,
        c.CKTSECSRCNODEID as SECSRCNENODEID,
        c.CktEnhancedState,
        c.CktProtectionType,
        c.CktOchncChannel,
        c.CktOchncDir,
        c.CktSrcModelType as SrcModelType,
        c.CktSrcObjectType as SrcObjectType,
        c.CktSecSrcModelType as CktSecSrcModelType,
        c.CktSecSrcObjectType as SecSrcObjectType,
```

```
        dest.CktDestModelType as CktDestModelType,
        dest.CktDestObjectType as CktDestObjectType,
        c.rowid as myrow
from
        circuit_tbl c,
        circuit_dest_tbl dest,
        circuit_vlan_view vlan
where
        c.CktNodeId             = dest.CktNodeId(+)  and
        c.CktUniqueId           = dest.CktUniqueId(+) and
        c.CktNodeId             = vlan.CktNodeid   and
        c.CktUniqueid            = vlan.CktUniqueid;
```

2.2.21 Circuit_VLAN_View

```
CREATE OR REPLACE VIEW circuit_vlan_view AS
SELECT
        b.CktVlanId,
        b.CktVlanName,
        b.CktNodeId,
        b.CktUniqueId,
        a.rowid as myrow
FROM
        circuit_vlan_tbl b ,
        circuit_bs_tbl a
WHERE
        a.CktNodeId   = b.CktNodeId   AND
        a.CktUniqueId = b.CktUniqueId
UNION
SELECT
        0,
        '',
        a.CktNodeId,
        a.CktUniqueId,
        a.rowid as myrow
FROM
        circuit_bs_tbl a
WHERE   a.rowid NOT IN
                        (  SELECT a.rowid
                           FROM   circuit_bs_tbl a,
                                  circuit_vlan_tbl b
                           WHERE  a.CktNodeId   = b.CktNodeId   AND
                                  a.CktUniqueId = b.CktUniqueId
                        ) ;
```

2.2.22 Ckt_Span_Dest_Mt_View

```
CREATE OR REPLACE VIEW ckt_span_dest_mt_view AS
SELECT DISTINCT
       cst.cktspandestnodeid as cktspandestnodeid,
       nvl(spandest.modulename,'System') as Modulename,
       cst.cktspandestphysicalloc,
       spandest.moduletype as ModuleType
FROM
      (select distinct
              mtt.modulename   as modulename,
              mtt.moduletype   as moduletype,
              cstbl.cktspandestphysicalloc as cktspandestphysicalloc,
              cstbl.cktspandestnodeid as cktspandestnodeid
         from module_type_table mtt,
              ons15454_ne_inventory_table cni,
              circuit_span_tbl cstbl,
              ne_info_table   ne
        where mtt.moduletype= cni.equipmenttype
          and
cni.physicalloc=round(cstbl.cktspandestphysicalloc/power(2,16))*(power(2,16))
          and ne.nenodeid = cstbl.cktspandestnodeid
          and cni.nedbaccessid=ne.nedbaccessid
          and ne.nemodeltype=mtt.modeltype
          and (mtt.modeltype=3 or mtt.modeltype=5 or mtt.modeltype=14 or mtt.modeltype=18)
      ) spandest,
```

```
      circuit_span_tbl cst
  WHERE cst.cktspandestnodeid = spandest.cktspandestnodeid(+)
    AND cst.cktspandestphysicalloc = spandest.cktspandestphysicalloc(+);
```

2.2.23 Ckt_Span_Src_Mt_View

```
CREATE OR REPLACE VIEW ckt_span_src_mt_view AS
SELECT DISTINCT
       cst.cktspansrcnodeid as cktspansrcnodeid,
       nvl(spansrc.modulename,'System') as Modulename,
       cst.cktspansrcphysicalloc,
       spansrc.moduletype as ModuleType
FROM
      (select distinct
              mtt.modulename   as modulename,
              mtt.moduletype   as moduletype,
              cstbl.cktspansrcnodeid as cktspansrcnodeid,
              cstbl.cktspansrcphysicalloc as cktspansrcphysicalloc
         from module_type_table mtt,
              ons15454_ne_inventory_table cni,
              circuit_span_tbl cstbl,
              ne_info_table ne
        where mtt.moduletype= cni.equipmenttype
          and cni.physicalloc=round(cstbl.cktspansrcphysicalloc/power(2,16))*(power(2,16))
          and ne.nenodeid=cstbl.cktspansrcnodeid
          and cni.nedbaccessid=ne.nedbaccessid
          and ne.nemodeltype=mtt.modeltype
          and (mtt.modeltype=3 or mtt.modeltype=5 or mtt.modeltype=14 or mtt.modeltype=18)
      ) spansrc,
      circuit_span_tbl cst
  WHERE cst.cktspansrcnodeid=spansrc.cktspansrcnodeid(+)
    AND cst.cktspansrcphysicalloc=spansrc.cktspansrcphysicalloc(+) ;
```

2.2.24 Client_User_View

```
create or replace view Client_User_VIEW As
select
       username  as Username
from user_table
union
select distinct
      username  as Username
from  deleted_user_table;
```

2.2.25 CTM_Unknown_User_View

```
create or replace view CTM_Unknown_User_View as
select
    CTMUnknownUserTable.IPAddress,
    CTMUnknownUserTable.FailedAttempts,
   CTMUnknownUserTable.LastLoginFailTime,
    decode(CTMUnknownUserTable.LockedState, 0, 'Unlocked',
                                            1, 'Locked') AS LockedState,
    CTMUnknownUserTable.ROWID AS MYROW
FROM
    CTM_Unknown_User_Table CTMUnknownUserTable;
```

2.2.26 Data_Card_View

```
CREATE OR REPLACE VIEW DATA_CARD_VIEW AS
SELECT
    B.NEID,
    B.SLOT,
    B.PORTNUM,
    B.SUBIF,
    B.BG,
    A.CIR,
    A.CIR_BURST,
```

```
    A.PIR,
    A.PIR_BURST,
    A.CCOSTRANSMIT
FROM
    INTF_SUB_INTF_VIEW B,
    QOS_VIEW A
WHERE
    A.NEID(+) = B.NEID and A.SLOT(+) = B.SLOT and A.PORTNUM(+) = B.PORTNUM and A.BG(+) =
B.BG;
```

2.2.27 Domain_NE_Inventory_View

```
create or replace view domain_ne_inventory_view as
select
       distinct
    ne.NESysID,
       ne.NEModelType,
    getNEIPAddr(ne.NEDBAccessID) NEIPAddr,
    getNEIPAddr (gne.ne.NEDBAccessID) gneipaddr,
       ne.isNEConnected,
       ne.NEState,
       ne.NEPMEnabled,
       ne.NERobustPmEnabled,
       ne.NEVersion,
       ne.GNEID,
       ne.NESubNetworkID,
       np.NPId,
       ne.NESnmpCommString,
       ne.NEConfigMode,
       ne.NECLLICode,
       ne.NEDescription,
       ne.NEDbaccessid,
       ne.NEDiscoveryState,
       mod.MODELTYPENAME,
       sub.SUBNETNAME,
       np.NPNAME,
       gne.GNESYSID,
       ne.NEAliasID,
       ne.rowid as myrow
    ne.ISTUNNELREQUIRED,
       isIPV6(ne.NEDBAccessID) isipv6_ne,
       isIPV6(gne.NEDBAccessID) isipv6_gne
        ne.ISDSS
 from  ne_info_table ne,
       network_partition_table np,
       subnetwork_table sub,
       model_type_table mod,
       gne_table gne
where ne.NeSubNetworkid=sub.subnetid
  and sub.npid= np.npid
  and ne.NEMODELTYPE=mod.MODELTYPE
  and ne.GNEID=gne.GNEID(+);
```

2.2.28 DSP_Circuit_View

```
create or replace view DSP_CIRCUIT_VIEW AS
select
        src.CKTCTPNODEID as SRCNENODEID,
        src.CktCTPPhysicalLoc as SrcPhysicalLoc,
        src.CKTCTPMODULETYPE as SRCMODULETYPE,
        dest.CKTCTPNODEID as DESTNENODEID,
        dest.CktCTPPhysicalLoc as DestPhysicalLoc,
        dest.CKTCTPMODULETYPE as DESTMODULETYPE
from
      circuit_bs_tbl c,
        circuit_ctp_src_view src,
        circuit_ctp_drop_view dest
where
        c.CKTNODEID = src.CKTNODEID(+) and
      c.CKTUNIQUEID = src.CKTUNIQUEID(+) and
        c.CKTNODEID = dest.CKTNODEID(+) and
       c.CKTUNIQUEID = dest.CKTUNIQUEID(+) and
```

```
        ( src.CKTCTPMODULETYPE = 6684672 or src.CKTCTPMODULETYPE = 6750208 or
src.CKTCTPMODULETYPE = 10747904 or src.CKTCTPMODULETYPE = 14483456 or
dest.CKTCTPMODULETYPE = 6684672 or dest.CKTCTPMODULETYPE = 6750208 or
dest.CKTCTPMODULETYPE = 10747904 or dest.CKTCTPMODULETYPE = 14483456) and
        c.cktState              != 12;
```

2.2.29 DSP_Eqpt_View

```
create or replace view DSP_EQPT_VIEW AS
select
        e.NEDBACCESSID as NEDBACCESSID,
        n.NENODEID as NENODEID,
        e.PHYSICALLOC as PHYSICALLOC,
        e.EQUIPMENTTYPE as MODULETYPE
from
        ONS15454_NE_INVENTORY_TABLE e,
        NE_INFO_TABLE n
where
        e.NEDBACCESSID = n.NEDBACCESSID(+) and
        ( e.EQUIPMENTTYPE = 102 or e.EQUIPMENTTYPE = 103 or e.EQUIPMENTTYPE = 164 or
e.EQUIPMENTTYPE = 221) and
        ( n.NEMODELINDEX = 47 or n.NEMODELINDEX = 49 or n.NEMODELINDEX = 55 or
n.NEMODELINDEX = 57 or n.NEMODELINDEX = 61 or n.NEMODELINDEX = 62 or n.NEMODELINDEX = 65
or n.NEMODELINDEX = 77 or  n.NEMODELINDEX = 78 or  n.NEMODELINDEX = 75 or  n.NEMODELINDEX
= 76 or n.NEMODELINDEX = 87 or n.NEMODELINDEX = 88 or n.NEMODELINDEX = 91 or
n.NEMODELINDEX = 81);
```

2.2.30 GWCORBA_PM_Mapping_View

```
create or replace view GWCORBA_PM_MAPPING_VIEW AS
SELECT DISTINCT
    gwpm.MODELTYPE AS MODELTYPE,
    gwpm.LAYERRATE AS LAYERRATE,
    gwpm.OBJECTINDEX AS OBJECTINDEX,
    gwpm.LOCATION AS LOCATION,
    gwpm.DIRECTION AS DIRECTION,
    gwpm.TERMINATIONPOINT AS TERMINATIONPOINT,
    pam.IS24H AS IS24H,
    gwpm.PMVIEWNAME AS PMVIEWNAME,
    ot.METADATAINDEX AS METADATAINDEX,
    pam.PMPARAMID AS PMPARAMID,
    pam.PMNAME AS PMNAME,
    pci.TMFNAME AS TMFNAME,
    pam.ATTRINDEX AS ATTRINDEX,
    pam.VALIDITY_ATTRINDEX AS VALIDITY_ATTRINDEX,
    pci.PMTABLECOLNO AS PMTABLECOLNO
from
        GWCORBA_PM_MAPPING_TABLE gwpm,
    OBJECTTYPETABLE ot,
    PM_ATTRID_MAPPING_TABLE pam,
    PM_CERENT_INFO_TABLE pci
where
    gwpm.OBJECTINDEX = ot.OBJECTINDEX AND
    gwpm.MODELTYPE = ot.MODELTYPE AND
    gwpm.PMVIEWNAME = pam.PMVIEWNAME AND
    ot.METADATAINDEX = pam.METADATAINDEX AND
    pam.PMPARAMID = pci.PMPARAMID AND
    gwpm.LOCATION = pci.LOCATION AND
    gwpm.DIRECTION = pci.DIRECTION AND
    TMFNAME NOT LIKE ('N/A');
```

2.2.31 INTF_Sub_Intf_View

```
CREATE OR REPLACE VIEW INTF_SUB_INTF_VIEW AS
SELECT DISTINCT
    ert.NEDBACCESSID as NEID,
    getSlot(ert.RELATEDPHYSICALLOC) as SLOT,
    getPort(ert.RELATEDPHYSICALLOC) as PORTNUM,
    getBGFromBGObj(ert.SELFIFINDEX) as BG,
    getSubIf(ert.RELATEDIFINDEX) as SUBIF
FROM
```

```
    Eqpt_Info_Table eqpt,
    Eqpt_RelationShip_Table ert
WHERE
    eqpt.NEDBACCESSID = ert.NEDBACCESSID AND eqpt.PHYSICALLOC = ert.RELATEDPHYSICALLOC
    AND eqpt.IFINDEX = ert.RELATEDIFINDEX AND ert.SELFOBJECTINDEX  = 165
    AND (eqpt.OBJECTINDEX = 172 OR eqpt.OBJECTINDEX = 173  OR eqpt.OBJECTINDEX = 175 OR
eqpt.OBJECTINDEX = 176);
```

2.2.32 Inventory_Table_View

```
create or replace view Inventory_Table_View as
select
    ni.NESysID,
    inv.ModuleName,
    inv.ModulePartNum,
    inv.ModuleCLEICode,
    inv.ModuleHWVersion,
    inv.ModuleSWVersion,
    inv.ModuleSerialNumber,
    inv.ModuleDateOfManuf,
    inv.ModuleDateOfEntry,
    inv.ModuleHoursInUse,
    ni.NEDBAccessId,
    inv.rowid as myrow
from
    NE_INFO_TABLE ni,
    NE_INVENTORY_TABLE inv
where
    ni.NEDBAccessID = inv.NEDBAccessID ;
```

2.2.33 IOS_Active_Cards

```
create or replace view IOS_ACTIVE_CARDS as
select distinct
       a.NEDBACCESSID as NEDBACCESSID,
       a.PHYSICALLOC as PHYSICALLOC,
       a.EQUIPMENTTYPE as MODULETYPE,
       a.NAME as MODULENAME,
       b.NESYSID as NEID,
       b.NEMODELTYPE as NEMODELTYPE,
       a.rowid as MYROW
from ONS15454_NE_INVENTORY_TABLE a, NE_INFO_TABLE  b
where EQUIPMENTTYPE in ( 102,103,164,221 ) and
      EQUIPMENTSTATE = 5 and
      a.NEDBACCESSID = b.NEDBACCESSID;
```

2.2.34 IOS_Users_View

```
CREATE OR REPLACE VIEW IOS_USERS_VIEW AS
select
       c.NEDBACCESSID as NEDBACCESSID,
       c.MODULETYPE as MODULETYPE,
       c.PHYSICALLOC as PHYSICALLOC,
       c.IFINDEX as IFINDEX,
       c.COL2 as USERNAME,
       c.COL5 as PASSWORD,
       c.COL3 as PRIVILEGE,
       a.MODULENAME as MODULENAME,
       b.NESYSID as NEID,
       c.MODELTYPE as NEMODELTYPE,
       c.rowid as MYROW,
       b.NEAliasID as NEAliasID
from EQPT_INFO_TABLE c,NE_INFO_TABLE b,MODULE_TYPE_TABLE a
where c.OBJECTINDEX = 221 and
c.NEDBACCESSID = b.NEDBACCESSID and
c.MODELTYPE = a.MODELTYPE and
c.MODULETYPE = a.MODULETYPE;
```

2.2.35 Link_Utilization_View

```
CREATE OR REPLACE VIEW LINK_UTILIZATION_VIEW
AS SELECT     a.CKTCTPMODULETYPE as CKTMODULETYPE,
           a.CKTCTPPHYSICALLOC as CKTPHYSICALLOC,
           a.CKTCTPIFINDEX as CKTIFINDEX,
           b.CKTNAME as CKTNAME,
           b.CKTPROTECTIONTYPE as CKTPROTECTIONTYPE,
           b.CUSTOMERID as CKTCUSTOMERID,
           b.SERVICEID as CKTSERVICEID,
           b.CKTSIZE as CKTSIZE,
           a.CKTCTPIFINDEX as STRIPPED_IFINDEX,
           a.CKTCTPNEDBACCESSID as NEDBACCESSID,
           b.CKTTYPE as CKTTYPE,
           b.CKTUSELAP as CKTUSELAP,
           a.CKTCTPIFINDEX as STRIPPED_CKTSRC_IFINDEX,
           collapsedifindex(b.CKTNODEID,b.CKTUNIQUEID)  as CKTDEST_IFINDEX,
           b.CKTUNIQUEID as CKTUNIQUEID,
           b.CKTNODEID as CKTNODEID,
           a.CKTCTPOBJECTTYPE as CKTOBJECTTYPE,
           collapsedobjectindex(b.CKTNODEID,b.CKTUNIQUEID) as cktDestObjectType,
           0 as modelType,
           nvl(d.CKTSPANPROTOP, -1) as SWITCHSTATE,
           b.CKTALIASNAME as CKTALIASNAME,
           nvl(d.CKTLINKID,-1) as LINKID,
           b.cktochncchannel as ochncchannel,
           b.Rowid as myrowFROM   CIRCUIT_CTP_TBL a, CIRCUIT_BS_TBL b, CIRCUIT_SPAN_BS_TBL
d
WHERE
            a.CKTUNIQUEID = b.CKTUNIQUEID AND
            a.CKTNODEID = b.CKTNODEID AND
b.CKTTYPE != 5 AND b.CKTTYPE != 6 AND b.CKTTYPE != 21 AND b.CKTTYPE != 22 AND
            (
              ( a.CTP_TYPE = 3  AND a.CTPACCESSID = d.CKTSPANSRCCTPACCESSID ) OR
              ( a.CTP_TYPE = 4  AND a.CTPACCESSID = d.CKTSPANDSTCTPACCESSID )
            )
UNION ALL
SELECT   a.CKTCTPMODULETYPE as CKTMODULETYPE,
           a.CKTCTPPHYSICALLOC as CKTPHYSICALLOC,
           a.CKTCTPIFINDEX as CKTIFINDEX,
           b.CKTNAME as CKTNAME,
           b.CKTPROTECTIONTYPE as CKTPROTECTIONTYPE,
           b.CUSTOMERID as CKTCUSTOMERID,
           b.SERVICEID as CKTSERVICEID,
           b.CKTSIZE as CKTSIZE,
           a.CKTCTPIFINDEX as STRIPPED_IFINDEX,
           a.CKTCTPNEDBACCESSID as NEDBACCESSID,
           b.CKTTYPE as CKTTYPE,
           b.CKTUSELAP as CKTUSELAP,
           a.CKTCTPIFINDEX as STRIPPED_CKTSRC_IFINDEX,
           collapsedifindex(b.CKTNODEID,b.CKTUNIQUEID)  as CKTDEST_IFINDEX,
           c.CKTUNIQUEID = b.CKTUNIQUEID AND
            c.CKTNODEID = b.CKTNODEID AND
a.CKTCTPOBJECTTYPE as CKTOBJECTTYPE,
           collapsedobjectindex(b.CKTNODEID,b.CKTUNIQUEID) as cktDestObjectType,
           0 as modelType,
           -1 as SWITCHSTATE,
           b.CKTALIASNAME as CKTALIASNAME,
          -1  as LINKID,
           b.cktochncchannel as ochncchannel,
           b.Rowid as myrow
FROM   CIRCUIT_CTP_TBL a, CIRCUIT_BS_TBL b
WHERE
            a.CKTUNIQUEID = b.CKTUNIQUEID AND
            a.CKTNODEID = b.CKTNODEID AND
            b.CKTTYPE != 5 AND b.CKTTYPE != 6 AND b.CKTTYPE != 21 AND b.CKTTYPE != 22 AND
            ((a.CTP_TYPE = 0 OR a.CTP_TYPE = 1 OR a.CTP_TYPE = 2));
```

2.2.36 Mcard_L2topology_Map_View

```
CREATE OR REPLACE VIEW mcard_l2topology_map_view AS
```

```
SELECT
      l2tct.Toponodeid as Toponodeid,
      l2tct.Topouniqueid as Topouniqueid,
      nit.NeDbAccessId as NeDbId,
      cst.CktCTPPhysicalLoc as PhysicalLoc,
      cst.CktCTPModuleType as ModuleType
FROM  circuit_ctp_tbl cst,
      L2topology_Circuits_Table l2tct,
      Ne_Info_Table nit
WHERE nit.nedbaccessid=cdt.CKTCTPNeDBAccessId and
      cdt.ctp_type=2 AND
    cst.CktNodeId    = l2tct.NodeId AND
      cst.CktUniqueId  = l2tct.UniqueId AND
      cst.CktCTPNodeId = nit.NeNodeId AND
      cst.CktCTPModuleType IN (6684672,6750208,10747904,14483456)
UNION ALL
SELECT
      l2tct.Toponodeid as Toponodeid,
      l2tct.Topouniqueid as Topouniqueid,
      nit.NeDbAccessId as NeDbId,
      cdt.cktCTPPhysicalLoc as PhysicalLoc,
      cdt.cktCTPModuleType as ModuleType
FROM  circuit_ctp_tbl cdt,
      L2topology_Circuits_Table l2tct,
      Ne_Info_Table nit
WHERE nit.nedbaccessid=cdt.CKTCTPNeDBAccessId and
      cdt.ctp_type=2 AND
    cdt.CktNodeId     = l2tct.NodeId AND
      cdt.CktUniqueId   = l2tct.UniqueId AND
      cdt.CktCTPNodeId = nit.NeNodeId AND
      cdt.cktCTPModuleType IN (6684672,6750208,10747904,14483456);
```

2.2.37 MLBCC_View

```
create or replace force view MLBCC_VIEW AS
SELECT
    l2topocardview.TopoNodeId AS TopoNodeId,
    l2topocardview.TopoUniqueId AS TopoUniqueId,
        neInfo.NESysID AS NESysID,
        l2topocardview.NeDbID AS NEID,
        l2topocardview.SlotNumber AS SlotNumber,
        MODULE.ModuleName AS EquipmentType,
        l2topocardview.configState AS configState,
        l2topocardview.POS0_Status AS POS0_Status,
        l2topocardview.POS1_Status AS POS1_Status,
        l2topocardview.RPR0 AS RPR0,
        getMLcardAddnlInfo( l2topocardview.AdditionalInfo, inv.EquipmentState) AS
AdditionalInfo,
        getProtectMLcardInfo( l2topocardview.ProtectCardNeDbId,
l2topocardview.ProtectCardSlotNumber ) AS ProtectCard,
        getBccCosInfo(l2topocardview.NeDbID, inv.PhysicalLoc, 181, 'AVVID_VOICE_VIDEO') AS
COS_AVVID_VOICEVIDEO,
        getBccCosInfo(l2topocardview.NeDbID, inv.PhysicalLoc, 181, 'SP_MANAGEMENT') AS
COS_SP_MANAGEMENT,
        getBccCosInfo(l2topocardview.NeDbID, inv.PhysicalLoc, 181, 'CIR') AS
COS_COMMITTED,
        getBccCosInfo(l2topocardview.NeDbID, inv.PhysicalLoc, 181, 'AVVID_CONTROL') AS
COS_AVVID_CONTROL,
        getBccBwInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'SP_MANAGEMENT') AS
BW_SP_MANAGEMENT,
        getBccBwInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'CIR') AS BW_COMMITTED,
        getBccBwInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'AVVID_CONTROL') AS
BW_AVVID_CONTROL,
        getBccBwInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'BEST_EFFORT') AS
BW_BEST_EFFORT,
        getBccCosCommitGrp(l2topocardview.NeDbID, inv.PhysicalLoc, 0, 189) AS COS_COMMIT,
        getBccCosCommitGrp(l2topocardview.NeDbID, inv.PhysicalLoc, 11, 189) AS BW_GROUP1,
        getBccCosCommitGrp(l2topocardview.NeDbID, inv.PhysicalLoc, 12, 189) AS COS_GROUP1,
        getBccCosCommitGrp(l2topocardview.NeDbID, inv.PhysicalLoc, 21, 189) AS BW_GROUP2,
        getBccCosCommitGrp(l2topocardview.NeDbID, inv.PhysicalLoc, 22, 189) AS COS_GROUP2,
```

```
        getBccClassInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'SP_MANAGEMENT') AS
CLASS_SP_MANAGEMENT,
        getBccClassInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'CIR') AS CLASS_COMMITTED,
        getBccClassInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'AVVID_CONTROL') AS
CLASS_AVVID_CONTROL,
        getBccClassInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'BEST_EFFORT') AS
CLASS_BEST_EFFORT,
        getBccClassInfo(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 180, 'AVVID_VOICE_VIDEO') AS
CLASS_AVVID_VOICE_VIDEO,
        getBccClassRPR80217(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 284, 3) AS BW_A0,
        getBccClassRPR80217(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 284, 1) AS BW_A1,
        getBccClassRPR80217(l2topocardview.NeDbID,
getQOSPhLoc(l2topocardview.SlotNumber,inv.PhysicalLoc), 284, 2) AS BW_B_CIR
FROM
        NE_INFO_TABLE neInfo,
        ONS15454_NE_INVENTORY_TABLE inv,
        MODULE_TYPE_TABLE MODULE,
        L2_TOPOLOGY_CARDS_VIEW l2topocardview
WHERE
    l2topocardview.NESYSID = neInfo.NESysID AND
    inv.EquipmentType = MODULE.ModuleType AND
        neInfo.NEModelType = MODULE.ModelType; AND
    l2topocardview. NeDbID = inv.NeDbID AND
        l2topocardview.SlotNumber = getSlot(inv.PhysicalLoc);
```

2.2.38 MLL_Inv_View

```
create or replace view MLL_INV_VIEW AS
select distinct
       NEDBACCESSID as NEDBACCESSID,
       LAYERRATE as LAYERRATE
from ( select
        info.NEDBACCESSID as NEDBACCESSID,
        m.LAYERRATE as LAYERRATE
from
        MODULE_TYPE_MLL_INFO m,
        ne_info_table info,
        ONS15454_ne_inventory_table inv
where
        info.NEDBAccessID = inv.NEDBAccessID AND
        info.NEModelType  = m.ModelType AND
        inv.EquipmentType = m.ModuleType
UNION ALL
select
        ne.NEDBACCESSID as NEDBACCESSID,
        m.LAYERRATE as LAYERRATE
from
        MODULE_TYPE_MLL_INFO m,
        ne_info_table ne,
        eqpt_info_table eit
where
        ne.NEDBACCESSID = eit.NEDBACCESSID AND
        ne.NEModelType  = m.ModelType    AND
        eit.ModuleType  = m.ModuleType   AND
        ne.NEModelType in (4,6,10)       AND
        m.ModuleType  >  2003
UNION ALL
select
        t1.NEDBACCESSID as NEDBACCESSID,
        m.LAYERRATE as LAYERRATE
from
        MODULE_TYPE_MLL_INFO m,
        NE_INFO_TABLE t1,
        ONS15216_NE_INVENTORY_TABLE t2
where
        t1.NEDBACCESSID = t2.NEDBACCESSID AND
    t1.NEModelType = m.MODELTYPE
```

```
UNION ALL
select
        n.NEDBACCESSID as NEDBACCESSID,
        0 as LAYERRATE
from
        NE_INFO_TABLE n
where
        n.NEMODELTYPE in (24,33)
UNION ALL
select
        n.NEDBACCESSID as NEDBACCESSID,
        m.LAYERRATE as LAYERRATE
from
        MODULE_TYPE_MLL_INFO m,
        NE_INFO_TABLE n,
        EQPT_INFO_TABLE eqpt
where
        n.NEDBACCESSID = eqpt.NEDBACCESSID and
        m.MODELTYPE = n.NEMODELTYPE and
        m.MODULETYPE = eqpt.MODULETYPE and
        ((n.NEMODELTYPE = 9) or (n.NEMODELTYPE<14 and n.NEMODELTYPE>10))
UNION ALL
select
        neInfo.NEDBACCESSID as NEDBACCESSID,
        mll.LAYERRATE as LAYERRATE
from
        MODULE_TYPE_MLL_INFO mll,
        NE_INFO_TABLE neInfo,
        ONS1530x_NE_INVENTORY_TABLE neInv
where
        neInfo.NEDBACCESSID = neInv.NEDBACCESSID and
        neInfo.NEModelType  = mll.ModelType and
        neInv.Type  = mll.ModuleType and
        neInfo.NEModelType in (21,22));
```

2.2.39 NE_Alarm_View

```
create or replace view NE_Alarm_View as
select
    aat.NEDBAccessID,
    aat.ActiveAlarmSeverity,
    aat.ModuleOrIfIndex,
    aat.AlarmSeqNum,
        aat.ModuleType,
        aat.PhysicalLoc,
        aat.alarmStatus,
    aat.rowid as myrow,
        aat.ModelType,
        aat.ObjectType,
        aat.StrObjInstance
from
    ACTIVE_ALARM_TABLE aat,
    NE_INFO_TABLE nit
where
    aat.NEDBAccessID = nit.NEDBAccessID ;
```

2.2.40 NE_Circuit_NodeID_View

```
create or replace view NE_Circuit_NodeID_View as
SELECT
       a.nedbaccessid, b.cktname
  FROM ne_info_table a, circuit_bs_tbl b
 WHERE a.NENodeID = b.CktNodeId and
       b.cktstate != 12;
```

2.2.41 NE_Circuit_View

```
CREATE OR REPLACE VIEW NE_CIRCUIT_VIEW AS
SELECT DISTINCT
        ckt.CKTNODEID as CKTNODEID,
```

```
        ckt.CKTUNIQUEID as CKTUNIQUEID,
        ckt.CKTNAME as CKTNAME,
        c.CKTCTPNODEID as CKTCTPNODEID
from    circuit_ctp_tbl c,
        circuit_bs_tbl ckt
where
        c.CktNodeId = ckt.CktNodeId(+) and
        ckt.cktstate != 12 and
        c.CktUniqueId = ckt.CktUniqueId(+);
```

2.2.42 NE_Circuit_Vlan_View

```
create or replace view NE_Circuit_Vlan_View as
SELECT
       a.NEDBACCESSID, c.CKTVLANID
  FROM NE_INFO_TABLE a, CIRCUIT_TBL b, CIRCUIT_VLAN_TBL c
 WHERE b.CKTUNIQUEID = c.CKTUNIQUEID
   AND b.CKTNODEID = c.CKTNODEID
   AND b.CKTNODEID = a.NENODEID;
```

2.2.43 NE_Info_NP_View

```
create or replace view NE_Info_NP_VIEW As
select ne.NEDBAccessID,
       ne.NESysID,
       getNEIPAddr(ne.NEDBAccessID) ne.NEIPAddr,
       ne.NENSAPAddr,
       ne.NEState,
       ne.isNEConnected,
       decode(ne.GNEID,-1,'Not Applicable',
                          gne.GNESysID ) as GneName,
       ne.NEDescription,
       ne.NEConfigMode,
       ne.NESnmpCommString,
       ne.NECLLICode,
       ne.NEStateChangeTime,
       ne.NEModelType,
       ne.NEModelIndex,
       ne.NEModelStatus,
       decode(ne.nesubnetworkID,-1,'Not Applicable',
                            sub.SubnetName) as SubnetworkName,
       ne.NETimeOffset,
       ne.NELEDStatus,
       ne.NESystemTitle,
       ne.NEUserLabel,
       ne.NEVendorName,
       ne.NEVersion,
       ne.NENumActiveCriticalErrors,
       ne.NENumActiveMajorErrors,
       ne.NENumActiveMinorErrors,
       ne.NEPmEnabled,
       ne.NENodeID,
       ne.NEInventoryLastChanged,
       ne.NESubnetMask,
       ne.NEDefaultGateway,
       ne.NENumUnActiveCriticalErrors,
       ne.NENumUnActiveMajorErrors,
       ne.NENumUnActiveMinorErrors,
       ne.NENumCleared,
       ne.DisplayModelName,
       ne.GWTL1USERNAME,
       ne.GWTL1PASSWD,
       ne.ServiceID,
       ne.DeleteState,
       np.NPId,
       np.NPName
    isIPV6(ne.NEDBAccessID) isipv6
from   ne_info_table ne,
       subnetwork_table sub,
       gne_table        gne,
       network_partition_table np
```

```
Where  ne.GNEID=gne.gneid(+) AND
       ne.NESubNetworkID=sub.subnetID AND
       sub.NPId=np.NPId;
```

2.2.44 NE_Inventory_View

```
create or replace view NE_Inventory_View as
select
    nit.NESysID,
    ninvt.ModuleIndex,
    ninvt.ModuleName,
    ninvt.ModulePartNum,
    ninvt.ModuleCLEICode,
    ninvt.ModuleHWVersion,
    ninvt.ModuleSWVersion,
    ninvt.ModuleSerialNumber,
    ninvt.ModuleDateOfManuf,
    ninvt.ModuleDateOfEntry,
    ninvt.ModuleHoursInUse,
    nit.NEDBAccessId,
    ninvt.rowid as myrow
from
    NE_INVENTORY_TABLE ninvt,
    NE_INFO_TABLE nit
where
    ninvt.NEDBAccessID = nit.NEDBAccessID ;
```

2.2.45 NE_SNMPV3_FILTERS_VIEW

NE_SNMPV3_FILTERS_VIEW aggregates rows from NE_SNMPV3_FILTERS_TABLE and NE_INFO_TABLE to show the NE SYSID. Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_FILTERS_VIEW AS
select
    NE_INFO_TABLE.NEDBACCESSID,
    NE_INFO_TABLE.NESYSID,
    NE_SNMPV3_NFILTER_TABLE.FILTER_PROFILE_NAME,
    NE_SNMPV3_NFILTER_TABLE.SUBTREEOID,
    NE_SNMPV3_NFILTER_TABLE.BIT_MASK,
    NE_SNMPV3_NFILTER_TABLE.FILTER_TYPE
from
    NE_SNMPV3_NFILTER_TABLE join NE_INFO_TABLE
    ON NE_SNMPV3_NFILTER_TABLE.NEDBACCESSID = NE_INFO_TABLE.NEDBACCESSID
WHERE
    NE_INFO_TABLE.NESTATE = 1
AND
    NE_INFO_TABLE.NEDISCOVERYSTATE = 4;
```

2.2.46 NE_SNMPV3_GROUPS_VIEW

```
CREATE OR REPLACE VIEW NE_SNMPV3_GROUPS_VIEW AS
select
    NE_INFO_TABLE.NEDBACCESSID,
    NE_INFO_TABLE.NESYSID,
    NE_SNMPV3_GROUPS_TABLE.GROUP_NAME,
    NE_SNMPV3_GROUPS_TABLE.SECURITY_LEVEL,
    NE_SNMPV3_GROUPS_TABLE.READ_VIEW_ACCESS,
    NE_SNMPV3_GROUPS_TABLE.WRITE_VIEW_ACCESS,
    NE_SNMPV3_GROUPS_TABLE.NOTIFY_VIEW_ACCESS
from
    NE_SNMPV3_GROUPS_TABLE join NE_INFO_TABLE
    ON NE_SNMPV3_GROUPS_TABLE.NEDBACCESSID = NE_INFO_TABLE.NEDBACCESSID
WHERE
    NE_INFO_TABLE.NESTATE = 1
AND
    NE_INFO_TABLE.NEDISCOVERYSTATE = 4;
```

2.2.47 NE_SNMPV3_LOCAL_USER_VIEW

VIEW NE_SNMPV3_LOCAL_USER_VIEW aggregates rows from NE_SNMPV3_USERS_TABLE and NE_INFO_TABLE to show the NE SYSID and select Local Users only. A join on Engine ID is used to identify sysid of remote NE. Then a selection is done comparing NEDBACCESSID. Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_LOCAL_USER_VIEW AS
select NE_INFO_TABLE.NEDBACCESSID,
NE_INFO_TABLE.NESYSID,
NE_SNMPV3_USERS_TABLE.USER_NAME,
NE_SNMPV3_USERS_TABLE.GROUP_NAME,
NE_SNMPV3_USERS_TABLE.AUTHENTICATION_PROTOCOL,
NE_SNMPV3_USERS_TABLE.PRIVACY_PROTOCOL,
NE_SNMPV3_USERS_TABLE.Engine_Id_Len,
NE_SNMPV3_USERS_TABLE.Boots
from NE_SNMPV3_USERS_TABLE, NE_INFO_TABLE
where NE_SNMPV3_USERS_TABLE.ENGINE_ID = NE_INFO_TABLE.NE_SNMP_EID
and NE_SNMPV3_USERS_TABLE.NEDBACCESSID = NE_INFO_TABLE.NEDBACCESSID
AND NE_INFO_TABLE.NESTATE = 1
AND NE_INFO_TABLE.NEDISCOVERYSTATE = 4;
```

2.2.48 NE_SNMPV3_PROXY_FWD_VIEW

VIEW NE_SNMPV3_PROXY_FWD_VIEW aggregates rows from NE_SNMPV3_PROXY_FWD_TABLE and NE_INFO_TABLE to show the NE SYSIDs for remote NEs. The left outer join is used for the same reason described above for the NE_SNMPV3_REMOTE_USER_VIEW. Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_PROXY_FWD_VIEW AS
select NE2.NEDBACCESSID,
NE2.NESYSID,
NE_SNMPV3_PROXY_FWD_TABLE.CONTEXT_ENGINE_ID,
NE_SNMPV3_PROXY_FWD_TABLE.TARGET_IP,
NE_SNMPV3_PROXY_FWD_TABLE.LOCAL_USER,
NE_SNMPV3_PROXY_FWD_TABLE.REMOTE_USER,
NE_SNMPV3_PROXY_FWD_TABLE.PROXY_TYPE,
NE_SNMPV3_PROXY_FWD_TABLE.SECURITY_LEVEL,
NE1.NESYSID AS REMOTE_NE,
NE_SNMPV3_PROXY_FWD_TABLE.ROW_ID
from (NE_SNMPV3_PROXY_FWD_TABLE left outer join NE_INFO_TABLE NE1
on NE_SNMPV3_PROXY_FWD_TABLE.CONTEXT_ENGINE_ID = NE1.NE_SNMP_EID)
join NE_INFO_TABLE NE2 on NE2.NEDBACCESSID = NE_SNMPV3_PROXY_FWD_TABLE.NEDBACCESSID
WHERE NE2.NESTATE = 1
AND NE2.NEDISCOVERYSTATE = 4;
```

2.2.49 NE_SNMPV3_PROXY_TRAP_FWD_VIEW

NE_SNMPV3_PROXY_TRAP_FWD_VIEW aggregates rows from NE_SNMPV3_PROXY_TRAP_FWD_TABLE and NE_INFO_TABLE to show the NE SYSID, and NESYSID for remote NE. Left outer join is used for the same reason described above (see comment for NE_SNMPV3_REMOTE_USER_VIEW). Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_PROXY_TRAP_FWD_VIEW AS
select NE2.NEDBACCESSID,
NE2.NESYSID,
NE_SNMPV3_PROXY_TRAP_FWD_TABLE.CONTEXT_ENGINE_ID,
NE_SNMPV3_PROXY_TRAP_FWD_TABLE.TAG,
NE_SNMPV3_PROXY_TRAP_FWD_TABLE.INCOMING_USER,
NE_SNMPV3_PROXY_TRAP_FWD_TABLE.SECURITY_LEVEL,
NE1.NESYSID AS REMOTE_NE,
NE_SNMPV3_PROXY_TRAP_FWD_TABLE.ROW_ID
from (NE_SNMPV3_PROXY_TRAP_FWD_TABLE left outer join NE_INFO_TABLE NE1
on NE_SNMPV3_PROXY_TRAP_FWD_TABLE.CONTEXT_ENGINE_ID = NE1.NE_SNMP_EID)
join NE_INFO_TABLE NE2 on NE2.NEDBACCESSID = NE_SNMPV3_PROXY_TRAP_FWD_TABLE.NEDBACCESSID
WHERE NE2.NESTATE = 1
AND NE2.NEDISCOVERYSTATE = 4;
```

2.2.50 NE_SNMPV3_REMOTE_USER_VIEW

View NE_SNMPV3_REMOTE_USER_VIEW aggregates rows from NE_SNMPV3_USERS_TABLE and NE_INFO_TABLE to show the NE SYSID and select Remote Users only. Select NEs that are in the In Service state and DISCOVERED only.

The query can be summarized as follows: NE_INFO_TABLE [join on DBACCESSID] NE_SNMPV3_USERS_TABLE [left outer join on ENGINEID] NE_INFO_TABLE. The first join produces the NESYSID of the NE described by the NE_SNMPV3_USERS_TABLE row. The second join produces the NESYSID of the REMOTE NE. A left outer join is required because the ENGINE ID cannot be found in the NE_INFO_TABLE (see example below). In this case, rows cannot be discarded. The where clause is used to select remote users only.

Example: Engine ID cannot match the one in the NE_INFO_TABLE.

Engine ID in NE_INFO_TABLE is populated during synchronization. Suppose you have:
GNE --- ENE-2
|
|
ENE-1

If you then preprovision all NEs (GNE, ENE-1, ENE-2), but you put only the GNE in the In Service state, the SnmpV3UserTable will be populated for the GNE. The query will not match the Engine ID for remote users.

```
CREATE OR REPLACE VIEW NE_SNMPV3_REMOTE_USER_VIEW AS
select
    NE1.NEDBACCESSID,
    NE1.NESYSID,
    NE_SNMPV3_USERS_TABLE.ENGINE_ID,
    NE_SNMPV3_USERS_TABLE.USER_NAME,
    NE_SNMPV3_USERS_TABLE.AUTHENTICATION_PROTOCOL,
    NE_SNMPV3_USERS_TABLE.PRIVACY_PROTOCOL,
    NE_INFO_TABLE.NESYSID AS REMOTE_NE,
    NE_SNMPV3_USERS_TABLE.ENGINE_ID_LEN,
    NE_SNMPV3_USERS_TABLE.BOOTS
from
    (NE_SNMPV3_USERS_TABLE left outer join NE_INFO_TABLE
```

```
    on NE_SNMPV3_USERS_TABLE.ENGINE_ID = NE_INFO_TABLE.NE_SNMP_EID)
    join NE_INFO_TABLE NE1 on NE1.NEDBACCESSID = NE_SNMPV3_USERS_TABLE.NEDBACCESSID
where
    (NE_SNMPV3_USERS_TABLE.NEDBACCESSID != NE_INFO_TABLE.NEDBACCESSID
    or NE_INFO_TABLE.NEDBACCESSID is null)
AND
    NE1.NESTATE = 1
    NE1.NEDISCOVERYSTATE = 4;
```

2.2.51 NE_SNMPV3_TRAP_DEST_VIEW

NE_SNMPV3_TRAP_DEST_VIEW aggregates rows from NE_SNMPV3_TRAP_DEST_TABLE and NE_INFO_TABLE to show the NE SYSID. Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_TRAP_DEST_VIEW AS
select
    NE_INFO_TABLE.NEDBACCESSID,
    NE_INFO_TABLE.NESYSID,
    NE_SNMPV3_TRAP_DEST_TABLE.TARGET_ADDRESS,
    NE_SNMPV3_TRAP_DEST_TABLE.UDP_PORT,
    NE_SNMPV3_TRAP_DEST_TABLE.USER_NAME,
    NE_SNMPV3_TRAP_DEST_TABLE.SECURITY_LEVEL,
    NE_SNMPV3_TRAP_DEST_TABLE.FILTER_PROFILE_NAME,
    NE_SNMPV3_TRAP_DEST_TABLE.PROXY_TRAPS_ONLY,
    NE_SNMPV3_TRAP_DEST_TABLE.PROXY_TAG
from
    NE_SNMPV3_TRAP_DEST_TABLE join NE_INFO_TABLE
    ON NE_SNMPV3_TRAP_DEST_TABLE.NEDBACCESSID = NE_INFO_TABLE.NEDBACCESSID
WHERE
    NE_INFO_TABLE.NESTATE = 1
AND
    NE_INFO_TABLE.NEDISCOVERYSTATE = 4;
```

2.2.52 NE_SNMPV3_VIEWS_VIEW

NE_SNMPV3_VIEWS_VIEW aggregates rows from NE_SNMPV3_VIEWS_TABLE and NE_INFO_TABLE to show the NE SYSID. Select NEs that are In Service state and DISCOVERED only.

```
CREATE OR REPLACE VIEW NE_SNMPV3_VIEWS_VIEW AS
select
    NE_INFO_TABLE.NEDBACCESSID,
    NE_INFO_TABLE.NESYSID,
    NE_SNMPV3_VIEWS_TABLE.VIEW_NAME,
    NE_SNMPV3_VIEWS_TABLE.SUBTREEOID,
    NE_SNMPV3_VIEWS_TABLE.MASK,
    NE_SNMPV3_VIEWS_TABLE.TYPE
from
    NE_SNMPV3_VIEWS_TABLE join NE_INFO_TABLE
    ON NE_SNMPV3_VIEWS_TABLE.NEDBACCESSID = NE_INFO_TABLE.NEDBACCESSID
WHERE
    NE_INFO_TABLE.NESTATE = 1
AND
    NE_INFO_TABLE.NEDISCOVERYSTATE = 4;
```

2.2.53 NE_Swimage_View

```
create or replace view NE_SWIMAGE_VIEW as
SELECT
        ne.NESysId,
        mtt.modelTypeName AS ModelName,
        ne.NEState,
        sw.TYPE,
        sw.activeimagename,
```

```
        sw.standbyimagename,
        sw.defaultimagename,
        ne.nedbaccessid,
        sw.inprogress,
        sw.physicalloc,
        mtt.modeltype,
        ne.NEAliasID,
        sw.partialupgrade
        sw.rowid as myrow
from
        NE_SWIMAGE_TABLE sw,
        NE_INFO_TABLE ne,
        Model_Type_Table mtt
where
        sw.nedbaccessid = ne.nedbaccessid AND
        mtt.ModelType = ne.NEModelType;
```

2.2.54 NGNE_Info_View

```
CREATE OR REPLACE VIEW NGNE_INFO_VIEW(TNE_NE_DBACCESS_ID, TL1_PORT, NGNE_PKID, NGNE_TID,
NGNE_IPADDRESS, NGNE_USERNAME, NGNE_USER_PASSWORD) AS
SELECT DISTINCT
        ti.TNE_NE_DBACCESS_ID,
        tg.NGNE_TL1PORT,
        tg.NGNE_PKID,
        tg.NGNE_TID,
        tg.NGNE_IPADDRESS,
        tg.NGNE_USERNAME,
        tg.NGNE_USER_PASSWORD
from
        TL1_TUNNEL_GATEWAY_TABLE   tg,
        TL1_TUNNEL_INFO_TABLE      ti
where
        tg.NGNE_PKID = ti.FK_NGNE_PKID ;
rem
rem NE_SNMPV3_GROUPS_VIEW aggregates rows from NE_SNMPV3_GROUPS_TABLE and NE_INFO_TABLE to
show the NE SYSID
rem Select NEs that are In Service state and DISCOVERED only.
rem
```

2.2.55 OSS_Alarm_Fltr_AidDeny_View

```
create or replace view OSS_Alarm_Fltr_AidDeny_VIEW as
select
    alarmfltrtbl.AlarmFilterId,
    alarmenum.AIDString,
    alarmfltrtbl.RowId AS myrow
FROM
    OSS_Alarm_Fltr_AID_Deny_Tbl alarmfltrtbl,
    OSS_AID_ENUM_TABLE alarmenum
WHERE
    alarmfltrtbl.AidId = alarmenum.AIDId;
```

2.2.56 OSS_Alarm_Fltr_TidDeny_VIEW

```
create or replace view OSS_Alarm_Fltr_TidDeny_VIEW as
select
    neInfo.NEDBAccessId,
    alarmfltrtbl.AlarmFilterId,
    neInfo.NESysId,
    alarmfltrtbl.ROWID AS MYROW
FROM
    OSS_Alarm_Fltr_TID_Deny_Tbl alarmfltrtbl,
    NE_INFO_TABLE neInfo
WHERE
    alarmfltrtbl.NEDBAccessID = neInfo.NEDBAccessID;
```

2.2.57 OSS_Corba_User_View

```
create or replace view OSS_Corba_User_View as
select
    corbaUserTable.OSSName,
    corbaUserTable.IPAddress,
    corbaUserTable.OSSPasswd,
    corbaUserTable.ROWID AS MYROW
FROM
    OSS_CORBA_USER_TABLE corbaUserTable;
```

2.2.58 OSS_User_View

```
create or replace view OSS_User_View as
select
    userTable.OSSName,
    getOUTIPAddr(userTable.OSSName) IPAddress,
    userTable.ListenerModeFlag,
    userTable.ActUserDenyFlag,
    userTable.DomainMgrFlag,
    userTable.AlarmFilterFlag,
    filterTable.FilterName,
    userTable.OSSPASSWD,
    userTable.ALARMFILTERID,
    userTable.ROWID AS MYROW
FROM
    OSS_ALARM_FILTER_TABLE filterTable,
    OSS_USER_TABLE userTable
WHERE
       filterTable.AlarmFilterId = userTable.AlarmFilterId;
```

2.2.59 POS_Ports_View

```
CREATE OR REPLACE VIEW POS_PORTS_VIEW AS
select
       c.NEDBACCESSID as NEDBACCESSID,
       c.MODULETYPE as MODULETYPE,
       c.MODELTYPE as NEMODELTYPE,
       a.MODULENAME as MODULENAME,
       c.PHYSICALLOC as PHYSICALLOC,
       c.IFINDEX as IFINDEX,
       c.COL5 as POSSTATE,
       c.COL15 as MTUSIZE
from EQPT_INFO_TABLE c,NE_INFO_TABLE b,MODULE_TYPE_TABLE a
where (c.OBJECTINDEX = 166 OR c.OBJECTINDEX = 284) and
c.NEDBACCESSID = b.NEDBACCESSID and
c.MODELTYPE = a.MODELTYPE and
c.MODULETYPE = a.MODULETYPE;
```

2.2.60 QoSTemplate_General_View

```
CREATE OR REPLACE VIEW QOSTEMPLATE_GENERAL_VIEW AS
SELECT
  QOSTEMPLATEID as TID,
  QOSTEMPLATENAME as TNAME,
  QOSTEMPLATETYPE as TTYPE,
  QOSTEMPLATEDESCRIPTION as TDESC
FROM L2QOSTEMPLATE_TABLE;
```

2.2.61 QoS_View

```
CREATE OR REPLACE VIEW QOS_VIEW AS
SELECT DISTINCT
    eqpt.NEDBACCESSID as NEID,
    getSlot(eqpt.PHYSICALLOC) as SLOT,
    getPort(eqpt.PHYSICALLOC) as PORTNUM,
    getBGFromQoSObj(eqpt.IFINDEX) as BG,
```

```
    eqpt.COL4 as CIR,
    eqpt.COL5 as CIR_BURST,
    eqpt.COL17 as PIR,
    eqpt.COL18 as PIR_BURST,
    eqpt.COL11 as CCOSTRANSMIT
FROM
    Eqpt_Info_Table eqpt
WHERE
    eqpt.OBJECTINDEX = 180 AND (getPolicyMapType(eqpt.IFINDEX) = 3 OR
getPolicyMapType(eqpt.IFINDEX) = 5);
```

2.2.62 Roll_View

```
CREATE OR REPLACE VIEW roll_view AS
SELECT
        (select CktName from circuit_bs_tbl where cktnodeid = r.rollfromcktnodeid and
cktuniqueid=r.rollfromcktuniqueid) as rollFromCkt,
        (select CktName from circuit_bs_tbl where cktnodeid = r.rolltocktnodeid and
cktuniqueid=r.rolltocktuniqueid) as rollToCkt,
        r.rollstate as rollState,
        r.rollvalidsignal as rollValidSignal,
        r.rollmode as rollMode,
        (select nesysid from ne_info_table where nenodeid=r.rollpathnodeid and
r.rollpathnodeid !=0 )  as PathNodeId,
        r.rollpathnodeid,
        nvl(getSlotModuleName(r.rollpathnodeid, r.rollpathmoduletype),'SYSTEM')  as
rollPathModuleType,
        r.rollpathphysicalloc as rollPathPhySicalLoc,
        r.rollpathifindex as rollPathIfIndex,
        (select nesysid from ne_info_table where nenodeid=r.rollfromnodeid and
r.rollfromnodeid !=0 )  as FromNodeId,
        r.rollfromnodeid,
        nvl(getSlotModuleName(r.rollfromnodeid, r.rollfrommoduletype),'SYSTEM')  as
rollFromModuleType,
        r.rollfromphysicalloc as rollFromPhySicalLoc,
        r.rollfromifindex as rollFromIfIndex,
        (select nesysid from ne_info_table where nenodeid=r.rolltonodeid and
r.rolltonodeid !=0 )  as ToNodeId,
        r.rolltonodeid,
        nvl(getSlotModuleName(r.rolltonodeid, r.rolltomoduletype),'SYSTEM')  as
rollToModuleType,
        r.rollTophysicalloc as rollToPhySicalLoc,
        r.rollToifindex as rollToIfIndex,
        r.rollNodeid as nodeId,
        r.Connectionindex as connectionIndex,
        r.canBeCompleted as canBeCompleted,
        r.canBeFinished as canBeFinished,
        r.canBeCancelled as canBeCancelled,
        r.rollPathModelType as rollPathModelType,
        r.rollpathobjectindex as rollPathObjectIndex,
        r.rollFromModelType as rollFromModelType,
        r.rollfromobjectindex as rollfromobjectindex,
        r.rollToModelType as rollToModelType,
        r.rolltoobjectindex as rollToObjectindex,
        nvl(getPortModuleName(r.rollpathnodeid, r.rollpathmoduletype),'SYSTEM')  as
rollPathPortModuleType,
        nvl(getPortModuleName(r.rollfromnodeid, r.rollfrommoduletype),'SYSTEM')  as
rollFromPortModuleType,
        nvl(getPortModuleName(r.rolltonodeid, r.rolltomoduletype),'SYSTEM')  as
rollToPortModuleType,
        r.rollname as rollName,
        r.canBeForced as canBeForced,
        r.rowid as myrow
From
        roll_table r;
```

2.2.63 Self_Monitoring_View

```
create or replace view Self_Monitoring_view AS
SELECT sp.DisplayName,
       sp.ModelType,
       sm.CollectionTime,
       sm.Value,
```

```
       sm.nedbaccessid,
       ne.Nesysid as Nesysid,
       ne.NEAliasID as NEAliasID,
       sm.rowid as MyRow
FROM   SERVER_MONITOR_TABLE sm,
       SERVER_PARAMETER_TABLE sp,
       ne_info_table ne
WHERE  sp.ParameterIndex=sm.ParameterIndex
  and  sm.nedbaccessid=ne.nedbaccessid
union
SELECT sp.DisplayName,
       sp.ModelType,
       sm.CollectionTime,
       sm.Value,
       sm.nedbaccessid,
       pn.Nesysid as Nesysid,
       'CTM' as NEAliasID,
       sm.rowid as MyRow
FROM   SERVER_MONITOR_TABLE sm,
       SERVER_PARAMETER_TABLE sp,
       purged_ne_table pn
WHERE  sp.ParameterIndex=sm.ParameterIndex
  and  sm.nedbaccessid=pn.nedbaccessid
union
SELECT sp.DisplayName,
       sp.ModelType,
       sm.CollectionTime,
       sm.Value,
       sm.nedbaccessid,
       'CTM' as Nesysid,
       'CTM' as NEAliasID,
       sm.rowid as MyRow
FROM   SERVER_MONITOR_TABLE sm,
       SERVER_PARAMETER_TABLE sp
WHERE  sp.ParameterIndex=sm.ParameterIndex
  AND  sm.nedbaccessid=0;
```

2.2.64 Service_Availability_Log_View

```
CREATE OR REPLACE VIEW SERVICE_AVAILABILITY_LOG_VIEW AS
select
    SERVICE_INSTANCE_NAME AS SERVICE_INSTANCE_NAME,
        FIRST_ACTIVATED AS FIRST_ACTIVATED,
        DEACTIVATED AS DEACTIVATED,
    STATUS AS STATUS,
    LAST_CRASH AS LAST_CRASH,
    PAST_UPTIME AS TOTAL_UPTIME,
    PERC_UPTIME AS PERC_UPTIME
  FROM SERVICE_AVAILABILITY_TABLE WHERE STATUS = 0;
```

2.2.65 SPAN_LINK_View

```
CREATE OR REPLACE VIEW SPAN_LINK_VIEW
AS
SELECT
    spn.CKTSPANBSID as SPANID,
    lnk.LINKID as LINKID,
    spn.CKTLINKID as ORIGINAL_SPAN_LINKID,
    lnk.LINKPROTTYPE as LINKPROT,
    lnk.LINKTYPE as LINKTYPE,
    circuit.CKTPROTECTIONTYPE as CIRCUITPROT,
    circuit.CKTTYPE as CIRCUITTYPE,
    circuit.CKTNAME as CIRCUITNAME,
    spn.rowid as SPANROWID
FROM
  CIRCUIT_CTP_TBL ctp1, CIRCUIT_CTP_TBL ctp2, CIRCUIT_SPAN_BS_TBL spn, LINK_TABLE lnk,
CIRCUIT_BS_TBL circuit
WHERE
         spn.CKTSPANSRCCTPACCESSID = ctp1.CTPACCESSID
         AND
         spn.CKTSPANDSTCTPACCESSID = ctp2.CTPACCESSID
         AND
         circuit.CKTNODEID = spn.CKTNODEID
```

```
         AND
         circuit.CKTUNIQUEID = spn.CKTUNIQUEID
         AND
         (( ctp1.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND ctp1.CKTCTPNEDBACCESSID =
lnk.LINKSRCNODE AND
            ctp2.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND ctp2.CKTCTPNEDBACCESSID =
lnk.LINKDSTNODE )
         OR
         ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND ctp1.CKTCTPNEDBACCESSID =
lnk.LINKDSTNODE AND
           ctp2.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND ctp2.CKTCTPNEDBACCESSID =
lnk.LINKSRCNODE ))
         AND lnk.LINKTYPE = getlinktype( circuit.CKTPROTECTIONTYPE, circuit.CKTTYPE,
lnk.LINKTYPE, spn.CKTSPANBSID)
         AND lnk.LINKTYPE <> 1;
```

2.2.66 SPR_View

```
CREATE OR REPLACE VIEW SPR_VIEW AS
select
       c.NEDBACCESSID as NEDBACCESSID,
       c.MODULETYPE as MODULETYPE,
       c.MODELTYPE as NEMODELTYPE,
       a.MODULENAME as MODULENAME,
       c.PHYSICALLOC as PHYSICALLOC,
       c.IFINDEX as IFINDEX,
       c.COL15 as MTUSIZE
from EQPT_INFO_TABLE c,NE_INFO_TABLE b,MODULE_TYPE_TABLE a
where c.OBJECTINDEX = 186 and
c.NEDBACCESSID = b.NEDBACCESSID and
c.MODELTYPE = a.MODELTYPE and
c.MODULETYPE = a.MODULETYPE;
```

2.2.67 SW_Active_Alarm_View

```
create or replace view SW_Active_Alarm_View as
select distinct
    aa.ActiveAlarmSeverity,
    ni.NESysID,
    if.ifName,
    aa.ActiveAlarmTimeStamp,
    en.EnumString,
    aa.ActiveAlarmServEff,
    aa.ActiveAlarmAdditionalInfo,
    aa.NEDBAccessID,
    aa.ModuleOrIfIndex,
    aa.rowid as myrow
from
    NE_INFO_TABLE ni,
    INTERFACE_TABLE if,
    ACTIVE_ALARM_TABLE aa,
    ALARM_EVENT_TYPE_ENUM_TABLE en
where
    aa.NEDBAccessID = ni.NEDBAccessID AND
    aa.ModuleOrIfIndex= if.ifStaticIndex AND
    aa.ActiveAlarmType = en.EnumIndex ;
```

2.2.68 SW_User_Port_View

```
create or replace view SW_User_Port_View as
select
    p.userid,
    p.nedbaccessid,
    p.neportifindex,
    i.ifName,
    p.rowid as myrow
from
    sway_user_port_table p,
    interface_table i
```

```
where
    p.NEPortIfIndex = i.ifstaticindex;
```

2.2.69 TL1_Active_Alarm_View

```
CREATE OR REPLACE VIEW tl1_active_alarm_view AS
SELECT
       ni.NESysID,
       aa.ModuleOrIfIndex,
       aa.AlarmStatus,
       aa.ActiveAlarmSeverity,
       en.TL1String,
       aa.ActiveAlarmServEff,
       TO_CHAR(aa.ActiveAlarmTimeStamp, 'MM/DD/YYYY HH24:MI:SS') AS ActiveAlarmTimeStamp,
       en.EnumString,
       eq.ModuleName,
       aa.NEDBAccessID,
       ni.NEModelType,
       TO_CHAR(aa.PhysicalLoc) AS PhysicalLoc,
       ni.DeleteState,
       aa.ObjectType,
      aa.STROBJINSTANCE
FROM
       ne_info_table ni,
       active_alarm_table aa,
       alarm_event_type_enum_table en,
       module_type_table eq
WHERE
       aa.ActiveAlarmSeverity > 4               AND
       aa.ModuleType          = eq.ModuleType   AND
       ni.NEModelType         = eq.ModelType    AND
       aa.NEDBAccessID        = ni.NEDBAccessID AND
       aa.ActiveAlarmType     = en.EnumIndex    AND
       aa.AlarmStatus         = 0;
```

2.2.70 TL1_NE_View

```
create or replace view TL1_NE_View as
select distinct
        si.SubnetName,
        ni.NEIPAddr,
    ni.NEIPV6Addr,
        ni.NESysID,
        ni.GNEID,
        ni.NEModelType,
        ni.NEVersion,
        (select GNESysID from GNE_Table where GNEID=ni.GNEID) as GNESysID,
        ni.NEState,
        ni.isNEConnected,
        ni.NESubNetworkID,
        ni.NEDBAccessID,
        ni.NEModelIndex,
        ni.DeleteState
           ni.IsTunnelRequired,
       isIPV6(ni.NEDBAccessID) isipv6
from
        NE_INFO_TABLE      ni,
        Subnetwork_Table   si,
        GNE_Table          gi
where
        ni.NESubnetworkID = si.SubnetID AND
        ni.GNEID          = gi.GNEID(+)
UNION
select distinct
        si.SubnetName,
        ni.NEIPAddr,
        ni.NESysID,
        ni.GNEID,
        ni.NEModelType,
        ni.NEVersion,
        si.SubnetName,
```

```
        ni.NEState,
        ni.isNEConnected,
        ni.NESubNetworkID,
        ni.NEDBAccessID,
        ni.NEModelIndex,
        ni.DeleteState
from
        NE_INFO_TABLE      ni,
        Subnetwork_Table   si
where
        ni.NESubnetworkID = si.SubnetID AND
        ni.NEModelType    = 34 ;
```

2.2.71 Topology_Circuit_View

```
create or replace view TOPOLOGY_CIRCUIT_VIEW as
WITH
   SRC  AS (select cktctpnedbaccessid,
                   CktUniqueId,
                   cktctpnodeid,
                   cktnodeid,
                   CktCtpModuleType,
                   cktctpPhysicalLoc,
                   CktctpModelType,
                   CktctpIfIndex,
                   cktctpDropProt from circuit_ctp_tbl where ctp_type = 0 ),
   SEC  AS (select cktctpnedbaccessid,
                   CktUniqueId,
                   cktctpnodeid,
                   cktnodeid,
                   CktCtpModuleType,
                   cktctpPhysicalLoc,
                   CktctpModelType,
                   CktctpIfIndex,
                   cktctpDropProt from circuit_ctp_tbl where ctp_type = 1 ),
   DEST AS ( select cktctpnedbaccessid,
                   CktUniqueId,
                   cktctpnodeid,
                   cktnodeid,
                   CktCtpModuleType,
                   cktctpPhysicalLoc,
                   CktctpModelType,
                   CktctpIfIndex,
                   CktCtpPathProt,
                   cktctpDropProt from circuit_ctp_tbl where ctp_type = 2 ),
   spantbl AS ( select spancircuit.CktNodeId as NodeId,
                       spancircuit.CktUniqueId as UniqueId
                from CIRCUIT_BS_Tbl circuit,
                     circuit_ctp_tbl ctp_src,
                     circuit_ctp_tbl ctp_dst,
                     circuit_span_BS_tbl spancircuit,
                     ne_info_table ne_src,
                     ne_info_table ne_dst
                where ctp_src.CTPACCESSID = spancircuit.CKTSPANSRCCTPACCESSID
                      and   ctp_dst.CTPACCESSID = spancircuit.CKTSPANDSTCTPACCESSID
                      and   spancircuit.CktNodeId = circuit.CktNodeId
                      and   spancircuit.CktUniqueId = circuit.CktUniqueId
                      and   ne_src.nenodeid = ctp_src.CKTCTPNODEID
                      and   ne_dst.nenodeid = ctp_dst.CKTCTPNODEID
                      and   (ctp_src.ctp_type = 3 and ctp_dst.ctp_type = 4)  ),
   span AS ( select span.cktnodeid, span.cktuniqueid, ctpsrc.cktctpnodeid asspansrcnodeid,
                    ctpdest.cktctpnodeid as spandestnodeid,
                    span.cktlinkid as cktlinkid
             from circuit_ctp_tbl ctpsrc,
                  circuit_ctp_tbl ctpdest,
                  circuit_span_bs_tbl   span
             where ctpsrc.ctpaccessid=span.cktspansrcctpaccessid and
                   ctpdest.ctpaccessid=span.cktspandstctpaccessid
           )
SELECT
       l2topo.Topology_Name,
        c.CktName as Name,
        c.CktAliasName as Alias,
        c.CktType as Type,
```

```
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
        get_nesysid(src.cktctpnedbaccessid) as SrcNodeid,      /* *@INDEX A 4 */
        src.cktctpnodeid as cktsrcnodeid,
       nvl(getModulename1(src.cktnodeid, src.CktCtpModuleType),'SYSTEM')  as
SrcModuleType,
        src.CktCtpPhysicalLoc as SrcPhysicalLoc,
        src.CktCtpModelType as SrcModelType,
        src.CktCtpIfIndex as SrcInterfaceId,
        nvl(getModulename1(sec.cktnodeid, sec.CktCtpModuleType),'SYSTEM') as
SecSrcModuleType,
        sec.CktCtpPhysicalLoc as SecSrcPhysicalLoc,
        sec.CktCtpModelType as CktSecSrcModelType,
        sec.CktCtpIfIndex as SecSrcInterfaceId,
        collapseddestnodename(c.CKTNODEID,c.CKTUNIQUEID) as destnodename,
        collapsedmodulename(c.CKTNODEID,c.CKTUNIQUEID) as cktdestModuleName,
       dest.CktCtpPhysicalLoc as DestPhysicalLoc,
        dest.CktCtpModelType as CktDestModelType,
        collapsedifindex(c.CKTNODEID,c.CKTUNIQUEID) as DestInterfaceId,
        c.CktIsMonitor as Ismonitor,
        src.CktCtpDropProt as SrcDropProt,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        dest.cktctpnodeid as DestNodeId,
        span.cktlinkid,
        span.SpanSrcNodeId as CktSpanSrcNodeId,
        span.SpanDestNodeId as CktSpanDestNodeId,
        c.CktEnhancedState,
        c.CktProtectionType,
        dest.CktCtpPathProt  as DestPathProtection,
        l2topo.rowid as myrow,
    l2topo.TOPONODEID,
    l2topo.TOPOUNIQUEID,
    eqptinfoport(src.cktctpnedbaccessid,src.cktctpphysicalloc,0,src.cktctpmoduletype) as
CktSrcPortName,
    eqptinfoport(sec.cktctpnedbaccessid,sec.cktctpphysicalloc,0,sec.cktctpmoduletype) as
CktSecSrcPortName,
    collapsedeqptportname(c.CKTNODEID,c.CKTUNIQUEID) as CktDestPortName,
    sec.CktNodeId as CktSecSrcNodeId,
    c.customerId as CktCusiomerId,
    c.serviceId as CktServiceId,
    c.cktComment as CktNote,
    l2topo.TopoType,
    l2topo.TopoSize,
        l2topo.TopoAliasName as TopoAliasName,
    l2topo.TopoDescription as TopoDescription,
        l2topo.AlarmSeverity as TopoAlarmSeverity,
        l2topo.TopoStatus as TopoStatus,
        l2topo.TopoProtection as TopoProtection,
        l2topo.TopoOperStatus as TopoOprStatus,
        l2topo.TOPORESYNCSTATUS as TopoResyncStatus,
    l2topo.BW_SP_Management,
    l2topo.BW_Committed,
    l2topo.BW_Best_Effort,
    l2topo.COS_COMMIT,
        l2topo.BW_Utilized,
        l2topo.BW_Available,
        l2topo.BW_AVVID_CONTROL,
        l2topo.COS_SP_MANAGEMENT,
        l2topo.COS_COMMITTED,
        l2topo.COS_AVVID_VOICEVIDEO,
        l2topo.COS_AVVID_CONTROL,
        l2topo.BW_Group1,
       l2topo.BW_Group2,
        l2topo.COS_GROUP1,
        l2topo.COS_GROUP2,
       l2topo.BW_A0,
        l2topo.BW_A1,
        l2topo.BW_B_CIR,
        l2topo.CLASS_SP_MGMT,
        l2topo.CLASS_BEST_EFFORT,
        l2topo.CLASS_CIR,
        l2topo.CLASS_AVVID_CTRL,
        l2topo.CLASS_AVVID_VOICEVIDEO,
```

```
        nvl(vcg.VCGDirection,-1) as VCGDirection,
        nvl(vcg.ParentCktNodeId,-1) as ParentCktNodeId,
        nvl(vcg.ParentCktUniqueId,-1) as ParentCktUniqueId
FROM
    L2Topology_Table l2topo,
        L2Topology_Circuits_Table l2circuits,
        circuit_bs_tbl c,
        src,
        sec,
        dest,
        span,
        spantbl,
        circuit_vcg_tbl vcg
where
    l2topo.TopoNodeID = l2circuits.TopoNodeID(+) AND
    l2topo.TopoUniqueId = l2circuits.TopoUniqueId(+) AND
    l2circuits.NodeId = c.CktNodeId AND
    l2circuits.UniqueId = c.CktUniqueId and
    c.CktNodeId             = src.CktNodeId(+)  and
    c.CktUniqueId           = src.CktUniqueId(+) and
    c.CktNodeId             = sec.CktNodeId(+) and
    c.CktUniqueId           = sec.CktUniqueId(+) and
    c.CktNodeId             = dest.CktNodeId(+) and
    c.CktUniqueId           = dest.CktUniqueId(+) and
    c.CktNodeId             = span.CktNodeId(+) and
    c.CktUniqueId           = span.CktUniqueId(+) and
    c.CktNodeId             = vcg.CktNodeId(+) and
    c.CktUniqueId           = vcg.CktUniqueId(+) and
    c.CktNodeId             = spantbl.NodeId(+) and
    c.CktUniqueId           = spantbl.UniqueId(+) and
    c.cktState != 12 AND
    (l2topo.TopoStatus > 2 OR l2topo.TopoStatus < 2);
```

2.2.72 Topology_Circuit_View_CTM5_0

```
create or replace view TOPOLOGY_CIRCUIT_VIEW_CTM5_0 as
SELECT
        l2topo.Topology_Name,
        c.CktName as Name,
        c.CktType as Type,
        c.CktSize as CktSize,
        c.CktDirection as Direction,
        c.CktState as State,
        (select nesysid from ne_info_table where nenodeid=c.cktsrcnodeid and
c.cktsrcnodeid !=0 )  as SrcNodeid,
        c.cktsrcnodeid,
        nvl(getModulename1(c.cktsrcnodeid, c.CktSrcModuleType),'SYSTEM')  as
SrcModuleType,
        c.CktSrcPhysicalLoc as SrcPhysicalLoc,
        c.CktSrcModelType as SrcModelType,
        c.CktSrcIfIndex as SrcInterfaceId,
        nvl(getModulename1(c.cktsecsrcnodeid, c.CktSecSrcModuleType),'SYSTEM') as
SecSrcModuleType,
        c.CktSecSrcPhysicalLoc as SecSrcPhysicalLoc,
        c.CktSecSrcModelType as CktSecSrcModelType,
        c.CktSecSrcIfIndex as SecSrcInterfaceId,
        c.cktDestNodename as destnodename,
        c.cktdestModuleName,
        c.CktDestPhysicalLoc as DestPhysicalLoc,
        c.CktDestModelType as CktDestModelType,
        c.CktDestIfIndex  as DestInterfaceId,
        c.CktIsMonitor as Ismonitor,
        c.CktSrcDropProt as SrcDropProt,
        c.CktNodeId as NodeId,
        c.CktUniqueId as UniqueId,
        c.CktDescription as Description,
        dest.cktdestnodeid as DestNodeId,
        spantbl.cktlinkid,
        spantbl.CktSpanSrcNodeId,
        spantbl.CktSpanDestNodeId,
        c.CktEnhancedState,
        c.CktProtectionType,
        dest.CktDestPathProt  as DestPathProtection,
        l2topo.rowid as myrow,
```

```
    l2topo.TOPONODEID,
    l2topo.TOPOUNIQUEID,
    c.cktSrcPortName as CktSrcPortName,
    c.cktSecSrcPortName as CktSecSrcPortName,
    c.cktDestPortName as CktDestPortName,
    c.cktSecSrcNodeId as CktSecSrcNodeId,
    c.customerId as CktCusiomerId,
    c.serviceId as CktServiceId,
    c.cktComment as CktNote,
    l2topo.TopoType,
    l2topo.TopoSize,
    l2topo.TopoDescription as TopoDescription,
    l2topo.AlarmSeverity as TopoAlarmSeverity,
    l2topo.TopoStatus as TopoStatus,
    l2topo.TopoProtection as TopoProtection,
    l2topo.TopoOperStatus as TopoOprStatus,
    l2topo.TOPORESYNCSTATUS as TopoResyncStatus,
    l2topo.BW_SP_Management,
    l2topo.BW_Committed,
    l2topo.BW_Best_Effort,
    l2topo.COS_COMMIT,
        l2topo.BW_Utilized,
        l2topo.BW_Available,
        l2topo.BW_AVVID_CONTROL,
        l2topo.COS_SP_MANAGEMENT,
        l2topo.COS_COMMITTED,
        l2topo.COS_AVVID_VOICEVIDEO,
        l2topo.COS_AVVID_CONTROL,
        l2topo.BW_Group1,
    l2topo.BW_Group2,
        l2topo.COS_GROUP1,
        l2topo.COS_GROUP2
FROM
    L2Topology_Table l2topo,
        L2Topology_Circuits_Table l2circuits,
        circuit_tbl c,
        circuit_dest_tbl dest,
        circuit_span_tbl spantbl
where
    l2topo.TopoNodeID = l2circuits.TopoNodeID(+) AND
    l2topo.TopoUniqueId = l2circuits.TopoUniqueId(+) AND
        l2circuits.NodeId = c.CktNodeId AND
        l2circuits.UniqueId = c.CktUniqueId and
        c.CktNodeId = dest.CktNodeId(+)  and
        c.CktUniqueId = dest.CktUniqueId(+) and
        c.CktNodeId = spantbl.CktNodeId(+) and
        c.CktUniqueId = spantbl.CktUniqueId(+) and
        c.cktState != 12 AND
        l2topo.TopoStatus != 2;
```

2.2.73 Topology_GUI_View

```
CREATE OR REPLACE VIEW topology_gui_view AS
SELECT DISTINCT
        l2Topology.Topology_Name as Name,
        l2Topology.TopoAliasName as Alias,
    l2Topology.TopoNodeID as TopoNodeID,
    l2Topology.TopoUniqueId as TopoUniqueID,
        l2Topology.TopoType as TopoType,
        l2Topology.TopoSize as TopoSize,
        l2Topology.TopoDescription as Description,
        l2Topology.AlarmSeverity as Severity,
        l2Topology.TopoStatus as TopoStatus,
        l2Topology.TopoProtection as Protection,
        l2Topology.TopoOperStatus as OprStatus,
        l2Topology.TOPORESYNCSTATUS as ResyncStatus,
        l2Topology.rowId as myrow,
    c.CktType as CktType,
    l2Topology.BW_SP_Management,
    l2Topology.BW_Committed,
    l2Topology.BW_Best_Effort,
        l2Topology.BW_AVVID_CONTROL,
        l2Topology.COS_SP_MANAGEMENT,
```

```
        l2Topology.COS_COMMITTED,
        l2Topology.COS_AVVID_VOICEVIDEO,
        l2Topology.COS_AVVID_CONTROL,
        l2Topology.BW_Group1,
        l2Topology.BW_Group2,
        l2Topology.COS_GROUP1,
        l2Topology.COS_GROUP2
FROM
    L2Topology_Table l2Topology,
        L2Topology_Circuits_Table l2circuits,
        circuit_tbl c
WHERE
        l2Topology.TopoNodeID = l2circuits.TopoNodeID AND
        l2Topology.TopoUniqueId = l2circuits.TopoUniqueId AND
        l2circuits.NodeId = c.CktNodeId AND
        l2circuits.UniqueId = c.CktUniqueId AND
    l2Topology.TopoStatus != 2;
```

2.2.74 Trace_Vlan_View

```
CREATE OR REPLACE VIEW trace_vlan_view AS
SELECT DISTINCT
        eit1.COL250 as VLANID,
        eit2.COL10 as BRIDGEID,
        eit2.COL13 as DESIGNATEDROOT,
        eit1.NEDBACCESSID,
        ni.NESYSID,
        eit1.physicalloc,
        ni.CONFIGSYNCSTATUS as NODEMANAGERSTATE,
        eit1.COL13 as UNTAGGED,
        eit2.COL7 as BRIDGEMAXAGE,
        eit2.COL8 as BRIDGEHELLOTIME,
        eit2.COL15 as ROOTPORT,
        eit2.COL14 as ROOTCOST,
        eit1.COL5 as STPSTATE,
        eit1.COL11 as STPENABLED,
        st.SUBNETID,
        ni.rowid as myrow
FROM
        eqpt_info_table eit1,
        eqpt_info_table eit2,
        eqpt_info_table eit3,
        ne_info_table ni,
        subnetwork_table st
WHERE
        ni.NEDBACCESSID   = eit1.NEDBACCESSID AND
        ni.NEDBACCESSID   = eit2.NEDBACCESSID(+) AND
        ni.NEDBACCESSID   = eit3.NEDBACCESSID(+) AND
        (eit1.OBJECTINDEX=105 OR eit1.OBJECTINDEX=205) AND
        11=eit2.OBJECTINDEX(+) AND
        104=eit3.OBJECTINDEX(+) AND eit3.COL12!='LINEAR_NO_SWITCH' AND
        ni.CONFIGSYNCSTATUS=3 AND
        (ni.NESTATE=1 OR ni.NESTATE=4) AND
        st.SUBNETID = ni.NESUBNETWORKID;
```

2.2.75 Unmanaged_NE_Inventory_View

```
create or replace view UNMANAGED_NE_INVENTORY_VIEW as
SELECT DISTINCT
    t1.NEDBACCESSID,
    t1.NEMODELINDEX,
    t1.NESYSID,
    t1.NEDESCRIPTION,
    t3.MODELNAME,
    t1.NECONFIGMODE,
    t1.NEVERSION,
    t2.SERIALNUMBER,
    t1.NECLLICODE,
    t1.GNEID,
    t1.NESUBNETWORKID,
    t2.rowid as NYROW
FROM
    NE_INFO_TABLE t1,
```

```
    UNMANAGED_NE_INVENTORY_TABLE t2,
    VERSION_TABLE t3
WHERE
    t1.NEDBACCESSID = t2.NEDBACCESSID
    AND
    t1.NEMODELINDEX = t3.NEMODELINDEX;
```

2.2.76 User_NE_View

```
create or replace view User_NE_View as
select
        distinct
        grouporneid  as GROUPORNEID,
        userid       as USERID,
        'AA' as MYROW
from   User_Ne_table;
```

2.2.77 User_View

```
create or replace view User_View as
select
    umt.UserID,
    umt.TreeNodeType,
    umt.GroupOrNEID,
    ut.UserName,
    ut.rowid as myrow
from
    USER_MAP_TABLE umt,
    USER_TABLE ut
where
    umt.UserID = ut.UserID ;
```

2.2.78 Vlan_Drop_Ports_QoS_View

```
create or replace view VLAN_DROP_PORTS_QOS_VIEW as
SELECT DISTINCT
  l2if_qos.MATCH_ANY,
  l2if_qos.MATCH_DSCP,
  l2if_qos.MATCH_COS,
  l2if_qos.MATCH_IP_PRECEDENCE,
  l2if_qos.MATCH_DSCP_VALUE,
  l2if_qos.MATCH_COS_VALUE,
  l2if_qos.MATCH_IP_PRECEDENCE_VALUE,
  l2if_qos.ANDOR,
  l2if_qos.CLASS_NAME,
  l2if_qos.CIR_TYPE,
  l2if_qos.MAX_RATE_KBPS,
  l2if_qos.BURST_SIZE_BPS,
  l2if_qos.CIR_COS_TYPE,
  l2if_qos.CIR_COS_VALUE,
  l2if_qos.EXCESS_ACTION,
  l2if_qos.PIR_BURST_BPS,
  l2if_qos.PIR_KBPS,
  l2if_qos.PIR_COS_TYPE,
  l2if_qos.PIR_COS_VALUE,
  l2if_qos.VIOLATE_ACTION,
  l2if_qos.VIOLATE_COS_VALUE,
  l2if_qos.BEFF_TYPE,
  l2if_qos.BEFF_RATE_KBPS,
  l2if_qos.BEFF_BURST_SIZE_BPS,
  l2if_qos.NEID,
  l2if_qos.PORT_NUMBER,
  l2if_qos.SLOT_NUMBER,
  l2if_qos.BRIDGE_GROUP_NUMBER,
  l2if_qos.rowId as myrow,
  l2if_qos.PORT_SUB_INTERFACE
FROM
    L2_Interface_Table l2if,
    L2_Interface_QOSClass_Table l2if_qos
WHERE
  l2if.NEID =  l2if_qos.NEID and
```

```
  l2if.SLOT_NUMBER =   l2if_qos.SLOT_NUMBER and
  l2if.BRIDGE_GROUP_NUMBER = l2if_qos.BRIDGE_GROUP_NUMBER and
  l2if.PORT_NUMBER = l2if_qos.PORT_NUMBER and
  l2if.PORT_SUB_INTERFACE = l2if_qos.PORT_SUB_INTERFACE;
  order by CLASS_NAME;
```

2.2.79 Vlan_Drop_Ports_View

```
create or replace view VLAN_DROP_PORTS_VIEW as
SELECT DISTINCT
    l2topo.Topology_Name,
    l2vlan.VLAN_Number,
    l2vlan.Customer_Id,
    l2vlan.Service_Id,
    l2if.NEID,
    l2if.PORT_NUMBER,
    l2if.SLOT_NUMBER,
    l2if.BRIDGE_GROUP_NUMBER,
    l2if.PORT_SUB_INTERFACE,
    l2if.INTERFACE_NAME,
    l2if.INTERFACE_TYPE,
    l2if.PORT_TYPE,
    l2if.CONNECTION_TYPE,
    l2if.PORT_STATE,
    l2if.QOSTEMPLATE_NAME,
    l2topo.TOPONODEID,
    l2topo.TOPOUNIQUEID,
    l2if.rowId as myrow,
    l2fp.PM_STATUS,
    neIf.NESYSID,
    l2if.RSTP_STATUS,
    bg.BRIDGE_PROTOCOL
    GETIPSLA_IPADDR(l2if.NEID,l2if.SLOT_NUMBER,l2if.BRIDGE_GROUP_NUMBER) as IP_ADDRESS,
    GETIPSLA_SUBNETMASK(l2if.NEID,l2if.SLOT_NUMBER,l2if.BRIDGE_GROUP_NUMBER) as
SUBNET_MASK
FROM
    L2Topology_Table l2topo,
    L2_Vlan_Table l2vlan,
    L2_Frontport_Table l2fp,
    Bridge_Group_Table bg,
    NE_INFO_TABLE neIf,
    /* OUTER JOIN ADDED TO FIX THE PROBLEM ON THE EMPTY PORT CHANNEL TABLE */
    (L2_Interface_Table l2if LEFT OUTER JOIN Port_Channel_Table l2pc ON (l2pc.NE_NODE_ID =
           l2if.NEID AND l2pc.SLOT_NUMBER = l2if.SLOT_NUMBER AND l2pc.PORT_CHANNEL_ID =
           l2if.PORT_NUMBER AND l2if.INTERFACE_TYPE = 552))
WHERE
    l2vlan.TopoNodeID = l2topo.TopoNodeID AND
    l2vlan.TopoUniqueId = l2topo.TopoUniqueId AND
    l2if.INTERFACE_TYPE != 166 AND
    l2if.INTERFACE_TYPE != 174 AND
    l2if.INTERFACE_TYPE != 186 AND
    l2if.INTERFACE_TYPE != 190 AND
    l2if.INTERFACE_TYPE != 284 AND
    l2if.INTERFACE_TYPE != 285 AND
    bg.toponodeid = l2vlan.topoNodeid AND
    bg.topouniqueid = l2vlan.topouniqueid AND
    bg.vlan_number = l2vlan.vlan_number AND
    l2if.NEID = bg.NEID AND
    l2if.Slot_Number = bg.slot_number AND
    ((l2fp.NEID = l2if.NEID AND
    l2fp.SLOT_NUMBER = l2if.SLOT_NUMBER AND
    l2fp.PORT_NUMBER = l2if.PORT_NUMBER AND
    l2if.INTERFACE_TYPE != 552) OR
    (l2pc.NE_NODE_ID = l2if.NEID AND
    l2pc.SLOT_NUMBER = l2if.SLOT_NUMBER AND
    l2pc.PORT_CHANNEL_ID = l2if.PORT_NUMBER AND
    l2fp.PM_STATUS = 0 AND
    l2if.INTERFACE_TYPE = 552)) AND
    l2if.BRIDGE_GROUP_NUMBER = bg.Bridge_group_number AND
    l2if.NEID = neIf.NEDBACCESSID;
```

2.2.80 Vlan_Service_View

```
create or replace view VLAN_SERVICE_VIEW as
SELECT DISTINCT
    l2topo.Topology_Name,
    l2topo.TopoNodeID,
    l2topo.TopoUniqueId,
    l2vlan.VLAN_Number,
    l2vlan.Customer_Id,
    l2vlan.Service_Id,
    l2vlan.VlanStatus,
    l2topo.TopoResyncStatus,
    l2topo.TopoType,
    l2topo.TopoStatus,
    l2topo.TopoSize,
    l2vlan.ManagedVlan,
    l2vlan.rowId as myrow
FROM
    L2Topology_Table l2topo,
    L2_Vlan_Table l2vlan
WHERE
    l2vlan.TopoNodeID = l2topo.TopoNodeID AND
    l2vlan.TopoUniqueId = l2topo.TopoUniqueId AND
    l2vlan.VlanStatus != 2;
```

2.3 Cisco ONS 152xx Views

```
CREATE OR REPLACE VIEW ONS15200_inventory_view AS
SELECT
       ne.NESysId,
       eit.PhysicalLoc as PhysicalLoc,
       mtt.ModuleName as ModuleName,
       to_number(eit.Col1) as SERVICESTATE,
       nvl(eit.Col8, 'N/A') as ITUCHNO,
       to_number(eit.Col9) as DATARATE,
       to_number(eit.Col11) as APPLICATIONCODE,
       eit.Col12 as SERIALNO,
       eit.Col13 as NEID,
       eit.Col14 as PARTNO,
       nvl(eit.Col15, 'N/A') as NCBSWVERSION,
       nvl(eit.Col16, 'N/A') as NCBBOARDREVNO,
       eit.Col17 as CLEICODE,
       nvl(eit.Col26, 'N/A') as CONTAINERNAME,
       to_number(eit.Col44) as POWERSPLIT,
       nvl(eit.Col91, 'N/A') as NAME,
       nvl(eit.Col92, 'N/A') as PRIMARYNCBID,
       nvl(eit.Col93, 'N/A') as SECONDARYNCBID,
       ne.NEDBAccessId,
       eit.IFINDEX,
       mtt.ModuleType,
       to_number(eit.Col250) as CONFLICTINGCLIP,
       eit.rowid as myrow
FROM
       ne_info_table ne,
       eqpt_info_table eit,
       module_type_table mtt
WHERE
       ne.NEDBAccessId = eit.NEDBAccessId AND
       ne.NEModelType  = mtt.ModelType    AND
       eit.ModuleType  = mtt.ModuleType   AND
       mtt.ModuleType  > 7002             AND
       mtt.ModuleType  < 7300;
```

2.4 Cisco ONS 15216 Views

2.4.1 ONS15216_Active_User_View

```
create or replace view ONS15216_Active_User_view as
select
        a.nesysid as NEId,
        b.userid as UserId,
        b.nedbaccessid,
        b.rowid as myrow,
       a.nealiasid as ALIASID
from
        ons15216_active_user_table  b,
        ne_info_table a
where
        b.nedbaccessid = a.nedbaccessid;
```

2.4.2 ONS15216_Inventory_View

```
create or replace view ONS15216_INVENTORY_VIEW as
SELECT DISTINCT
    t1.NEDBACCESSID,
    t1.NEMODELINDEX,
    t1.NESYSID,
    t1.NEDESCRIPTION,
    t1.NECONFIGMODE,
    t1.NEVERSION,
    t2.WAVELENGTH,
    t2.PARTNUMBER,
    t2.CLEICODE,
    t2.SERIALNUMBER,
    t1.NECLLICODE,
    t1.GNEID,
    t1.NESUBNETWORKID,
    t2.MODULENUMBER,
    t2.rowid as MYROW,
    t1.NEALIASID as ALIASID
FROM
    NE_INFO_TABLE t1,
    ONS15216_NE_INVENTORY_TABLE t2
WHERE
    t1.NEDBACCESSID = t2.NEDBACCESSID;
```

2.4.3 ONS15216_User_View

```
create or replace view ONS15216_User_view as
select
        a.nesysid as NEId,
        b.userid as UserId,
        b.nedbaccessid,
        c.privilege as Privilege,
        b.lastlogintime as LastLoginTime,
    b.Timeout as Timeout,
        b.rowid as myrow,
        a.nealiasid as ALIASID
from
        ons15216_user_table  b,
        ne_info_table a,
        ons15216_privilege_table c
where
        b.nedbaccessid = a.nedbaccessid AND
        b.privilegeid = c.privilegeid;
```

2.5 Cisco ONS 153xx Views

2.5.1 ONS1530x_Dcc_PM_View

```
CREATE OR REPLACE VIEW ons1530x_dcc_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.IFINUCASTPKTS,
       pi.IFINDISCARDS,
       pi.IFINERRORS,
       pi.IFINOCTETS,
       pi.IFOUTOCTETS,
       pi.IFINNUCASTPKTS,
       pi.IFINUNKNOWNPROTOS,
       pi.IFOUTUCASTPKTS,
       pi.IFOUTNUCASTPKTS,
       pi.IFOUTDISCARDS,
       pi.IFOUTERRORS,
       pi.IFINMULTICASTPKTS,
       pi.IFINBROADCASTPKTS,
       pi.IFOUTMULTICASTPKTS,
       pi.IFOUTBROADCASTPKTS,
       pl.NEDBAccessId,
       mt.ModuleType,
       pi.is24H,
       pl.NEModelType,
       pi.OBJECTTYPE,
       pi.STROBJINSTANCE,
       pi.rowid as myrow,
       pl.NEAliasID
FROM
       ne_info_table pl,
       module_type_table mt,
       ONS1530x_IP_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
       mt.ModuleType   = pi.ModuleType AND
       pi.iftype = 2;
```

2.5.2 ONS1530x_HO_FE_PM_View

```
CREATE OR REPLACE VIEW ons1530x_ho_fe_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.FEBBE,
       pi.FEES,
       pi.FESES,
       pi.FEUAS,
       pl.NEDBAccessId,
       mt.ModuleType,
       pi.is24H,
       pl.NEModelType,
       pi.OBJECTTYPE,
       pi.STROBJINSTANCE,
       pi.rowid as myrow,
       pl.NEAliasID
FROM
       ne_info_table pl,
       module_type_table mt,
       ONS1530x_HO_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
```

```
       mt.ModuleType   = pi.ModuleType ;
```

2.5.3 ONS1530x_HO_PM_View

```
CREATE OR REPLACE VIEW ons1530x_ho_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.NEBBE,
       pi.NEES,
       pi.NESES,
       pi.NEUAS,
       pl.NEDBAccessId,
       mt.ModuleType,
       pi.is24H,
       pl.NEModelType,
       pi.OBJECTTYPE,
       pi.STROBJINSTANCE,
       pi.rowid as myrow,
       pl.NEAliasID
FROM
       ne_info_table pl,
       module_type_table mt,
       ONS1530x_HO_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
       mt.ModuleType   = pi.ModuleType ;
```

2.5.4 ONS1530x_Inventory_View

```
CREATE OR REPLACE VIEW ONS1530x_inventory_view AS
SELECT
       info.NESysId,
       nvl(mtt2.ModuleName,'EMPTY SLOT') as ModuleName,
       inv.PhysicalLoc as PhysicalLoc,
       inv.InstallState as InstallState,
       mtt1.ModuleName as ExpectedModule,
       nvl(inv.ServiceState,'N/A') as ServiceState,
       nvl(inv.SoftwareVersion,'N/A') as SoftwareVersion,
       nvl(inv.SerialNumber,'N/A') as SerialNumber,
       nvl(inv.HardwareVersion,'N/A') as HardwareVersion,
       info.NEDBAccessId,
       info.rowid as myrow,
       inv.Type as ModuleType,
       info.NEAliasID
FROM
       ne_info_table info,
       ONS1530x_ne_inventory_table inv,
       module_type_table mtt1,
       module_type_table mtt2
WHERE
       info.NEDBAccessID = inv.NEDBAccessID AND
       info.NEModelType  = mtt1.ModelType AND
       info.NEModelType  = mtt2.ModelType AND
       mtt1.moduleType   = inv.ExpectedModule AND
       mtt2.moduleType   = inv.ModuleName;
```

2.5.5 ONS1530x_Lan_PM_View

```
CREATE OR REPLACE VIEW ons1530x_lan_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.IFINUCASTPKTS,
```

```
        pi.IFINDISCARDS,
        pi.IFINERRORS,
        pi.IFINOCTETS,
        pi.IFOUTOCTETS,
        pi.IFINNUCASTPKTS,
        pi.IFINUNKNOWNPROTOS,
        pi.IFOUTUCASTPKTS,
        pi.IFOUTNUCASTPKTS,
        pi.IFOUTDISCARDS,
        pi.IFOUTERRORS,
        pi.IFINMULTICASTPKTS,
        pi.IFINBROADCASTPKTS,
        pi.IFOUTMULTICASTPKTS,
        pi.IFOUTBROADCASTPKTS,
        pl.NEDBAccessId,
        mt.ModuleType,
        pi.is24H,
        pl.NEModelType,
        pi.OBJECTTYPE,
        pi.STROBJINSTANCE,
        pi.rowid as myrow,
        pl.NEAliasID
FROM
        ne_info_table pl,
        module_type_table mt,
        ONS1530x_IP_PM_TABLE pi
WHERE
        pl.NEDbAccessID = pi.NEDbAccessId AND
        pl.NEModelType  = mt.ModelType    AND
        mt.ModuleType   = pi.ModuleType AND
        pi.iftype = 0;
```

2.5.6 ONS1530x_LO_FE_PM_View

```
CREATE OR REPLACE VIEW ons1530x_lo_fe_pm_view AS
SELECT
        pl.NESysId,
        mt.ModuleName,
        pi.PhysicalLoc,
        pi.NeIfIndex,
        pi.TimeStamp,
        pi.FEBBE,
        pi.FEES,
        pi.FESES,
        pi.FEUAS,
        pl.NEDBAccessId,
        mt.ModuleType,
        pi.is24H,
        pl.NEModelType,
        pi.OBJECTTYPE,
        pi.STROBJINSTANCE,
        pi.rowid as myrow,
        pl.NEAliasID
FROM
        ne_info_table pl,
        module_type_table mt,
        ONS1530x_LO_PM_TABLE pi
WHERE
        pl.NEDbAccessID = pi.NEDbAccessId AND
        pl.NEModelType  = mt.ModelType    AND
        mt.ModuleType   = pi.ModuleType ;
```

2.5.7 ONS1530x_LO_PM_View

```
CREATE OR REPLACE VIEW ons1530x_lo_pm_view AS
SELECT
        pl.NESysId,
        mt.ModuleName,
        pi.PhysicalLoc,
        pi.NeIfIndex,
        pi.TimeStamp,
        pi.NEBBE,
        pi.NEES,
```

```
        pi.NESES,
        pi.NEUAS,
        pl.NEDBAccessId,
        mt.ModuleType,
        pi.is24H,
        pl.NEModelType,
        pi.OBJECTTYPE,
        pi.STROBJINSTANCE,
        pi.rowid as myrow,
        pl.NEAliasID
FROM
        ne_info_table pl,
        module_type_table mt,
        ONS1530x_LO_PM_TABLE pi
WHERE
        pl.NEDbAccessID = pi.NEDbAccessId AND
        pl.NEModelType  = mt.ModelType    AND
        mt.ModuleType   = pi.ModuleType ;
```

2.5.8 ONS1530x_MS_FE_PM_View

```
CREATE OR REPLACE VIEW ons1530x_ms_fe_pm_view AS
SELECT
        pl.NESysId,
        mt.ModuleName,
        pi.PhysicalLoc,
        pi.NeIfIndex,
        pi.TimeStamp,
        pi.FEBBE,
        pi.FEES,
        pi.FESES,
        pi.FEUAS,
        pl.NEDBAccessId,
        mt.ModuleType,
        pi.is24H,
        pl.NEModelType,
        pi.OBJECTTYPE,
        pi.STROBJINSTANCE,
        pi.rowid as myrow,
        pl.NEAliasID
FROM
        ne_info_table pl,
        module_type_table mt,
        ONS1530x_MS_PM_TABLE pi
WHERE
        pl.NEDbAccessID = pi.NEDbAccessId AND
        pl.NEModelType  = mt.ModelType    AND
        mt.ModuleType   = pi.ModuleType ;
```

2.5.9 ONS1530x_MS_PM_View

```
CREATE OR REPLACE VIEW ons1530x_ms_pm_view AS
SELECT
        pl.NESysId,
        mt.ModuleName,
        pi.PhysicalLoc,
        pi.NeIfIndex,
        pi.TimeStamp,
        pi.NEBBE,
        pi.NEES,
        pi.NESES,
        pi.NEUAS,
        pl.NEDBAccessId,
        mt.ModuleType,
        pi.is24H,
        pl.NEModelType,
        pi.OBJECTTYPE,
        pi.STROBJINSTANCE,
        pi.rowid as myrow,
        pl.NEAliasID
FROM
```

```
       ne_info_table pl,
       module_type_table mt,
       ONS1530x_MS_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
       mt.ModuleType   = pi.ModuleType ;
```

2.5.10 ONS1530x_RS_PM_View

```
CREATE OR REPLACE VIEW ons1530x_rs_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.NEBBE,
       pi.NEES,
       pi.NESES,
       pi.NEUAS,
       pl.NEDBAccessId,
       mt.ModuleType,
       pi.is24H,
       pl.NEModelType,
       pi.OBJECTTYPE,
       pi.STROBJINSTANCE,
       pi.rowid as myrow,
       pl.NEAliasID
FROM
       ne_info_table pl,
       module_type_table mt,
       ONS1530x_RS_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
       mt.ModuleType   = pi.ModuleType ;
```

2.5.11 ONS1530x_Wan_PM_View

```
CREATE OR REPLACE VIEW ons1530x_wan_pm_view AS
SELECT
       pl.NESysId,
       mt.ModuleName,
       pi.PhysicalLoc,
       pi.NeIfIndex,
       pi.TimeStamp,
       pi.IFINUCASTPKTS,
       pi.IFINDISCARDS,
       pi.IFINERRORS,
       pi.IFINOCTETS,
       pi.IFOUTOCTETS,
       pi.IFINNUCASTPKTS,
       pi.IFINUNKNOWNPROTOS,
       pi.IFOUTUCASTPKTS,
       pi.IFOUTNUCASTPKTS,
       pi.IFOUTDISCARDS,
       pi.IFOUTERRORS,
       pi.IFINMULTICASTPKTS,
       pi.IFINBROADCASTPKTS,
       pi.IFOUTMULTICASTPKTS,
       pi.IFOUTBROADCASTPKTS,
       pl.NEDBAccessId,
       mt.ModuleType,
       pi.is24H,
       pl.NEModelType,
       pi.OBJECTTYPE,
       pi.STROBJINSTANCE,
       pi.rowid as myrow,
       pl.NEAliasID
FROM
       ne_info_table pl,
       module_type_table mt,
```

```
       ONS1530x_IP_PM_TABLE pi
WHERE
       pl.NEDbAccessID = pi.NEDbAccessId AND
       pl.NEModelType  = mt.ModelType    AND
       mt.ModuleType   = pi.ModuleType AND
       pi.iftype = 1;

commit;
```

2.6 Cisco ONS 15454 Views

2.6.1 APC_DOMAIN_NE_VIEW

```
SELECT DISTINCT
      ne.NEDbaccessid as NEDBAccessID,                    /* *@INDEX A 0
*/
      adt.NEDbaccessid as NEAPCDBAccessID
FROM  APC_Domain_Table adt,
      ne_info_table ne
WHERE
      ne.NEDbaccessid  = ne_belongs_to_APC_domain(adt.Sides,ne.NEDbaccessid);
```

2.6.2 Ctc_User_Profiles_View

```
create or replace view Ctc_User_Profiles_view as
select
       a.userid as UserId,
       b.privilege as Privilege,
       a.userdesc as UserDescription,
       a.userpassword as Password
from
       ctc_userprofile_table  a,
       ons15454_privilege_table b
where
       a.privilegeid = b.privilegeid;
```

2.6.3 L2_IPSLA_View

```
CREATE OR REPLACE VIEW L2_IPSLA_VIEW AS
SELECT DISTINCT
       ipslaTable.Ipsla_Number as Ipsla_Number,
       ipslaTable.TopoNodeId as TopoNodeId,
       ipslaTable.TopoUniqueId as TopoUniqueId,
       ipslaTable.Vlan_Number as Vlan_Number,
       neInfo.NESysID as NESysID,
       getSlot(ipslaTable.source_phyloc) as SlotNumber,
       module.ModuleName as EquipmentType,
       ipslaTable.Source_Addr as Source_Addr,
       ipslaTable.Destination_Addr as Destination_Addr,
       ipslaTable.Operation_type as Operation_type,
       ipslaTable.Operation_status as Operation_status
FROM
       L2_IPSLA_TABLE ipslaTable,
       Ne_Info_table neInfo,
       ONS15454_ne_inventory_table inv,
       module_type_table module
WHERE
       ipslaTable.source_neid = neInfo.NeDbaccessID and
       ipslaTable.source_neid = inv.NeDbaccessID and
       getSlot(ipslaTable.source_phyloc) = getSlot(inv.PhysicalLoc) and
       inv.EquipmentType = module.ModuleType and
       neInfo.NEModelType = module.ModelType;
```

2.6.4 L2_Topology_Cards_View

```
create or replace view L2_TOPOLOGY_CARDS_VIEW as
SELECT DISTINCT
```

```
        cardView.TopoNodeId as TopoNodeId,
        cardView.TopoUniqueId as TopoUniqueId,
        neInfo.NESysID as NESysID,
        mcard.SlotNumber as SlotNumber,
        module.ModuleName as EquipmentType,
        mcard.configState as configState,
        getPosPortState( l2topo.TopoType, mcard.NeDbaccessID, mcard.SlotNumber, 0) as
POS0_Status,
        getPosPortState( l2topo.TopoType, mcard.NeDbaccessID, mcard.SlotNumber, 1) as
POS1_Status,
        getPosPortStateForRPR802( l2topo.TopoType, mcard.NeDbaccessID, mcard.SlotNumber,
0) as RPR0,
        getMLcardAddnlInfo( mcard.AdditionalInfo, inv.EquipmentState) as AdditionalInfo,
        getProtectMLcardInfo( mcard.ProtectCardNeDbId, mcard.ProtectCardSlotNumber ) as
ProtectCard,
        inv.EquipmentState as EquipmentState,
    neInfo.NEDBAccessId as NEDBId,
        mcard.ProtectCardNeDbId ProtNeDBId,
        mcard.ProtectCardSlotNumber ProtSlotNum,
        getProtectMLcardConfigState( mcard.ProtectCardNeDbId, mcard.ProtectCardSlotNumber
as ProtConfigState
FROM
        MCard_L2Topology_map_view cardView,
        Ne_Info_table neInfo,
        ONS15454_ne_inventory_table inv,
        module_type_table module,
        MCard_Table mcard,
        L2Topology_Table l2topo
WHERE
        mcard.NeDbaccessID = neInfo.NeDbaccessID and
        mcard.SlotNumber = getSlot(cardView.PhysicalLoc) and
        mcard.NeDbaccessID = cardView.NeDbId and
        mcard.NeDbaccessID = inv.NeDbaccessID and
        mcard.SlotNumber = getSlot(inv.PhysicalLoc) and
        inv.EquipmentType = module.ModuleType and
        neInfo.NEModelType = module.ModelType and
        l2topo.TopoNodeId = cardView.TopoNodeId and
        l2topo.TopoUniqueId = cardView.TopoUniqueId;
```

2.6.5 NE_Audit_Trail_View

```
create or replace view NE_Audit_Trail_view as
select
        a.nesysid as NEId,
        c.neuserid as UserId,
        c.nedbaccessid,
        c.sequencenumber as SequenceNumber,
        c.timestamp as TimeStamp,
        c.operationdescription as OperationDescription,
        c.operationstatus as OperationStatus,
        c.rowid as myrow,
        a.NEALIASID as NEALIASID
from
        ne_info_table a,
        ne_audit_trail_table c
where
        a.nedbaccessid = c.nedbaccessid;
```

2.6.6 ONS15454SDH_E1_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454sdh_E1_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.FeRxPEb,
       csdh.FeRxPEs,
       csdh.FeRxPSes,
       csdh.FeRxPBbe,
```

```
        csdh.FeRxPUas,
        csdh.FeRxPEsr,
        csdh.FeRxPSesr,
        csdh.FeRxPBber,
        csdh.FeTxPEb,
        csdh.FeTxPEs,
        csdh.FeTxPSes,
        csdh.FeTxPBbe,
        csdh.FeTxPUas,
        csdh.FeTxPEsr,
        csdh.FeTxPSesr,
        csdh.FeTxPBber,
        csdh.NEDBAccessID as NEDBAccessID,
        csdh.thresholdList as ThresholdList,
        en.ModuleType,
        csdh.IS24H,
        csdh.ModelType,
        csdh.ObjectType,
        csdh.StrObjInstance,
        csdh.InvalidityList,
        csdh.rowid as myrow,
        ni.NEAliasID as NEAliasID,
         csdh.StrCorbaTidAid
FROM
        ons15454sdh_e1_pm_table csdh,
        ne_info_table ni,
        module_type_table en
WHERE
        csdh.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.7 ONS15454SDH_E1_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_E1_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csdh.PhysicalLoc,
        csdh.NEIfIndex as NEIfIndex,
        csdh.TimeStamp as TimeStamp,
        csdh.NEIfPmStatus as NEIfPmStatus,
        csdh.InMaintenance as InMaintenance,
        csdh.cvl as cvl,
        csdh.esl as esl,
        csdh.sesl as sesl,
        csdh.lossl as lossl,
        csdh.RxPEb as RxPEb,
        csdh.RxPBbe as RxPBbe,
        csdh.RxPes as RxPes,
        csdh.RxPses as RxPses,
        csdh.RxPUas as RxPUas,
        csdh.RxPEsr as RxPEsr,
        csdh.RxPSesr as RxPSesr,
        csdh.RxPBber as RxPBber,
        csdh.TxAiss as TxAiss,
        csdh.TxPEb as TxPEb,
        csdh.TxPBbe as TxPBbe,
        csdh.TxPes as TxPes,
        csdh.TxPses as TxPses,
        csdh.TxPUas as TxPUas,
        csdh.TxPEsr as TxPEsr,
        csdh.TxPSesr as TxPSesr,
        csdh.TxPBber as TxPBber,
        csdh.RxAiss as RxAiss,
        csdh.NEDBAccessID as NEDBAccessID,
        csdh.thresholdList as ThresholdList,
        en.ModuleType,
        csdh.IS24H,
        csdh.ModelType,
        csdh.ObjectType,
        csdh.StrObjInstance,
        csdh.InvalidityList,
```

```
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         csdh.StrCorbaTidAid
FROM
       ons15454sdh_e1_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.8 ONS15454SDH_E3_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_E3_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.cvl as cvl,
       csdh.esl as esl,
       csdh.sesl as sesl,
       csdh.lossl as lossl,
       csdh.RxPes as RxPes,
       csdh.RxPses as RxPses,
       csdh.RxPUas as RxPUas,
       csdh.RxPEsr as RxPEsr,
       csdh.RxPSesr as RxPSesr,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
    csdh.StrCorbaTidAid
FROM
       ons15454sdh_e3_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.9 ONS15454SDH_E4_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_E4_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance InMaintenance,
       csdh.CodeViolations as CV,
       csdh.ErrSecs as ES,
       csdh.SevErrSecs as SES,
       csdh.UnavailableSecs as UAS,
       csdh.BackGroundBlockErrors as BBE,
       csdh.ESR as Esr,
   csdh.SESR as SESR,
   csdh.BBER as BBER,
       csdh.NEDBAccessID as NEDBAccessId,
       csdh.thresholdList as ThresholdList,
```

```
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        csdh.StrCorbaTidAid
FROM
       ONS15454sdh_E4_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.10 ONS15454SDH_HO_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454sdh_ho_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.FeHpEb as FeHpEb,
       csdh.FeHpEs as FeHpEs,
       csdh.FeHpSes as    FeHpSes,
       csdh.FeHpUas as FeHpUas,
       csdh.FeHpBbe as FeHpBbe,
       csdh.FeHpEsr as FeHpEsr,
       csdh.FeHpSesr as FeHpSesr,
       csdh.FeHpBber as  FeHpBberas,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
   csdh.StrCorbaTidAid
FROM
       ONS15454sdh_ho_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.11 ONS15454SDH_HO_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_ho_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.HpEb as HpEb,
       csdh.HpEs as HpEs,
       csdh.HpSes as HpSes,
       csdh.HpUas as HpUas,
       csdh.HpBbe as HpBbe,
```

```
       csdh.HpEsr as HpEsr,
       csdh.HpSesr as HpSesr,
       csdh.HpBber as HpBber,
       csdh.HpPpjcPdet as HpPpjcPdet,
       csdh.HpNpjcPdet as HpNpjcPdet,
       csdh.HpPpjcPGen as HpPpjcPGen,
       csdh.HpNpjcPGen as HpNpjcPGen,
       csdh.HpPjDiff as HpPjDiff,
       csdh.HpPjpSec as HpPjpSec,
       csdh.HpPjnSec as HpPjnSec,
       csdh.StsPathWidth as StsPathWidth,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        csdh.StrCorbaTidAid
FROM
       ONS15454sdh_ho_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.12 ONS15454SDH_LO_VC11_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454sdh_lo_vc11_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
    csdh.FeLoEb as vc11FeLoEb,
    csdh.FeLoBbe as vc11FeLoBbe,
    csdh.FeLoEs as vc11FeLoEs,
    csdh.FeLoSes as vc11FeLoSes,
    csdh.FeLoUas as vc11FeLoUas,
    csdh.FeLoEsr as vc11FeLoEsr,
    csdh.FeLoSesr as vc11FeLoSesr,
    csdh.FeLoBber as vc11FeLoBber,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         csdh.StrCorbaTidAid
FROM
       ONS15454sdh_lo_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID AND
       csdh.LoType = 2;
```

2.6.13 ONS15454SDH_LO_VC11_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_lo_vc11_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.LoEb as vc11LoEb,
       csdh.LoBbe as vc11LoBbe,
       csdh.LoEs as vc11LoEs,
       csdh.LoSes as vc11LoSes,
       csdh.LoUas as vc11LoUas,
       csdh.LoEsr as vc11LoEsr,
       csdh.LoSesr as vc11LoSesr,
       csdh.LoBber as vc11LoBber,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        csdh.StrCorbaTidAid
FROM
       ONS15454sdh_lo_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID AND
       csdh.LoType = 2;
```

2.6.14 ONS15454SDH_LO_VC12_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454sdh_lo_vc12_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
    csdh.FeLoEb as vc12FeLoEb,
    csdh.FeLoBbe as vc12FeLoBbe,
    csdh.FeLoEs as vc12FeLoEs,
    csdh.FeLoSes as vc12FeLoSes,
    csdh.FeLoUas as vc12FeLoUas,
    csdh.FeLoEsr as vc12FeLoEsr,
    csdh.FeLoSesr as vc12FeLoSesr,
    csdh.FeLoBber as vc12FeLoBber,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        csdh.StrCorbaTidAid
FROM
       ONS15454sdh_lo_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
```

```
        csdh.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csdh.NEDBAccessID  = ni.NEDBAccessID AND
        csdh.LoType = 1;
```

2.6.15 ONS15454SDH_LO_VC12_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_lo_vc12_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csdh.PhysicalLoc,
        csdh.NEIfIndex as NEIfIndex,
        csdh.TimeStamp as TimeStamp,
        csdh.NEIfPmStatus as NEIfPmStatus,
        csdh.InMaintenance as InMaintenance,
        csdh.LoEb as vc12LoEb,
        csdh.LoBbe as vc12LoBbe,
        csdh.LoEs as vc12LoEs,
        csdh.LoSes as vc12LoSes,
        csdh.LoUas as vc12LoUas,
        csdh.LoEsr as vc12LoEsr,
        csdh.LoSesr as vc12LoSesr,
        csdh.LoBber as vc12LoBber,
        csdh.NEDBAccessID as NEDBAccessID,
        csdh.thresholdList as ThresholdList,
        en.ModuleType,
        csdh.IS24H,
        csdh.ModelType,
        csdh.ObjectType,
        csdh.StrObjInstance,
        csdh.InvalidityList,
        csdh.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csdh.StrCorbaTidAid
FROM
        ONS15454sdh_lo_pm_table csdh,
        ne_info_table ni,
        module_type_table en
WHERE
        csdh.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csdh.NEDBAccessID  = ni.NEDBAccessID AND
        csdh.LoType = 1;
```

2.6.16 ONS15454SDH_LO_VC3_PM_FE_View

```
CREATE  OR REPLACE VIEW ONS15454sdh_lo_vc3_pm_fe_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csdh.PhysicalLoc,
        csdh.NEIfIndex as NEIfIndex,
        csdh.TimeStamp as TimeStamp,
        csdh.NEIfPmStatus as NEIfPmStatus,
        csdh.InMaintenance as InMaintenance,
    csdh.FeLoEb as vc3FeLoEb,
    csdh.FeLoBbe as vc3FeLoBbe,
    csdh.FeLoEs as vc3FeLoEs,
    csdh.FeLoSes as vc3FeLoSes,
    csdh.FeLoUas as vc3FeLoUas,
    csdh.FeLoEsr as vc3FeLoEsr,
    csdh.FeLoSesr as vc3FeLoSesr,
    csdh.FeLoBber as vc3FeLoBber,
        csdh.NEDBAccessID as NEDBAccessID,
        csdh.thresholdList as ThresholdList,
        en.ModuleType,
        csdh.IS24H,
        csdh.ModelType,
        csdh.ObjectType,
        csdh.StrObjInstance,
        csdh.InvalidityList,
        csdh.rowid as myrow,
```

```
        ni.NEAliasID as NEAliasID,
          csdh.StrCorbaTidAid
FROM
        ONS15454sdh_lo_pm_table csdh,
        ne_info_table ni,
        module_type_table en
WHERE
        csdh.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csdh.NEDBAccessID  = ni.NEDBAccessID AND
        csdh.LoType = 0;
```

2.6.17 ONS15454SDH_LO_VC3_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_lo_vc3_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csdh.PhysicalLoc,
        csdh.NEIfIndex as NEIfIndex,
        csdh.TimeStamp as TimeStamp,
        csdh.NEIfPmStatus as NEIfPmStatus,
        csdh.InMaintenance as InMaintenance,
        csdh.LoEb as vc3LoEb,
        csdh.LoBbe as vc3LoBbe,
        csdh.LoEs as vc3LoEs,
        csdh.LoSes as vc3LoSes,
        csdh.LoUas as vc3LoUas,
        csdh.LoEsr as vc3LoEsr,
        csdh.LoSesr as vc3LoSesr,
        csdh.LoBber as vc3LoBber,
        csdh.NEDBAccessID as NEDBAccessID,
        csdh.thresholdList as ThresholdList,
        en.ModuleType,
        csdh.IS24H,
        csdh.ModelType,
        csdh.ObjectType,
        csdh.StrObjInstance,
        csdh.InvalidityList,
        csdh.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csdh.StrCorbaTidAid
FROM
        ONS15454sdh_lo_pm_table csdh,
        ne_info_table ni,
        module_type_table en
WHERE
        csdh.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csdh.NEDBAccessID  = ni.NEDBAccessID AND
        csdh.LoType = 0;
```

2.6.18 ONS15454SDH_MS_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454sdh_ms_pm_fe_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csdh.PhysicalLoc,
        csdh.NEIfIndex as NEIfIndex,
        csdh.TimeStamp as TimeStamp,
        csdh.NEIfPmStatus as NEIfPmStatus,
        csdh.InMaintenance as InMaintenance,
        csdh.FeMsEb as FeMsEb,
        csdh.FeMsEs as FeMsEs,
        csdh.FeMsSes as FeMsSes,
        csdh.FeMsUas as FemsUas,
        csdh.FeMsBbe as FeMsBbe,
        csdh.FeMsEsr as FeMsEsr,
        csdh.FeMsSesr as FeMsSesr,
        csdh.FeMsBber as FeMsBber,
```

```
       csdh.NEDBAccessID as NEDBAccessId,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         csdh.StrCorbaTidAid
FROM
       ONS15454sdh_ms_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.19 ONS15454SDH_MS_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_ms_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance InMaintenance,
       csdh.MsEb as MsEb,
       csdh.MsEs as MsEs,
       csdh.MsSes as MsSes,
       csdh.MsUas as MsUas,
       csdh.MsPpjcPdet as MsPpjcPdet,
       csdh.MsNpjcPdet as MsNpjcPdet,
       csdh.MsPpjcPGen as MsPpjcPGen,
       csdh.MsNpjcPGen as MsNpjcPGen,
       csdh.MsPsc as MsPsc,
       csdh.MsPsd as MsPsd,
       csdh.MSPscW as MSPscW,
       csdh.MSPsdW as MSPsdW,
       csdh.MSPscS as MSPscS,
       csdh.MSPsdS   as MSPsdS,
       csdh.MSPscR   as MSPscR,
       csdh.MSPsdR   as MSPsdR,
       csdh.MsBbe as MsBbe,
   csdh.MSPscmsPsd as MSPscmsPsd,
   csdh.MsFc as MsFc,
   csdh.MsEsr,
   csdh.MsSesr,
   csdh.MsBber,
       csdh.NEDBAccessID as NEDBAccessId,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID
         csdh.StrCorbaTidAid
FROM
       ONS15454sdh_ms_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.20 ONS15454SDH_RS_PM_View

```
CREATE OR REPLACE VIEW ONS15454sdh_rs_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.RsEb   as RsEb,
       csdh.RsEs   as RsEs,
       csdh.RsSes  as RsSes,
       csdh.RsBbe      as RsBbe,
       csdh.RsSefs as RsSefs,
       csdh.RsEsr as RsEsr,
       csdh.RsSesr as RsSesr,
       csdh.RsBber as RsBber,
       csdh.RSOFS as RsOfs,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        csdh.StrCorbaTidAid

FROM
       ONS15454sdh_rs_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.21 APC_DOMAIN_VIEW

```
CREATE OR REPLACE VIEW APC_Domain_VIEW as
SELECT
      ne.NESysid,
      adt.SideOut,
      adt.ApcState,
      adt.ApcAdminState,
      adt.NEDbaccessid,
      adt.Sides
FROM
       APC_Domain_Table adt,
      ne_info_table ne
WHERE
      ne.NEDbaccessid  = adt.NEDbaccessid;
```

2.6.22 ONS15454_8B10B_PM_View

```
CREATE OR REPLACE VIEW ONS15454_8B10B_PM_VIEW AS
SELECT
    ni.NESysID,
    ni.NEDbaccessid,
    en.ModuleName,
    en.ModuleType,
    fc.Physicalloc,
    fc.NEIfIndex,
    fc.TimeStamp,
    fc.NEIfPmStatus,
    fc.ThresholdList,
    fc.InMaintenance,
```

```
    fc.validPackets,
    fc.invalidPackets,
    fc.codeGroupViolations,
    fc.idleOrderedSets,
    fc.nonIdleOrderedSets,
    fc.dataCodeGroups,
    fc.rxTotalPackets,
    fc.ifInErrors,
    fc.statsEncodingDispErrors,
    fc.dataOrderedSets,
    fc.INVALIDORDEREDSETS,
    fc.dataPayLoad,
    fc.InvldOrdrdSetDispErrorSum,
    fc.IS24H,
    fc.ModelType,
    fc.ObjectType,
    fc.StrObjInstance,
    fc.InvalidityList,
    fc.rowid as myrow,
    ni.NEAliasID as NEAliasID,
     fc.StrCorbaTidAid
FROM
    ons15454_8B10B_pm_table fc,
    ne_info_table ni,
    module_type_table en
WHERE
    fc.ModuleType    = en.ModuleType   AND
    ni.NEModelType     = en.ModelType    AND
    fc.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.23 ONS15454_Active_User_View

```
create or replace view ONS15454_Active_User_view as
select
        a.nesysid as NEId,
        b.userid as UserId,
        b.nedbaccessid,
        b.clientIp as ClientIp,
        b.SessionType,
        b.LastLoginTime,
        b.LastActivityTime,
        b.SessionTicket,
        b.rowid as myrow,
       a.NEAliasID as NEAliasID
from
        ons15454_active_user_table  b,
        ne_info_table a
where
        b.nedbaccessid = a.nedbaccessid;
```

2.6.24 ONS15454_APC_Result_View

```
CREATE OR REPLACE VIEW ONS15454_APC_RESULT_VIEW as
select
    ne.NESysid,
    eit.col11       as Side,
    eit.col4        as Port,
    eit.col10       as Param,
    eit.col5        as LastModification,
    eit.col6        as LastCheck,
    ne.NEDbaccessid
From
    ne_info_table     ne,
    eqpt_info_table   eit
Where
   ne.NEDbaccessid  = eit.NEDbaccessid AND
   eit.ObjectIndex  = 280
ORDER BY 7,2;
```

2.6.25 ONS15454_BLSR_Config_View

```
CREATE or REPLACE view ONS15454_BLSR_CONFIG_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as Type,
      eit.col5         as Rate,
      eit.col6         as RingID,
      eit.col7         as NodeID,
      eit.col8         as RingReversion,
      eit.col9         as SpanReversion,
      eit.col10        as WestLine,
      eit.col11        as EastLine,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4007;
```

2.6.26 ONS15454_BLSR_Squelch_View

```
CREATE or REPLACE view ONS15454_BLSR_SQUELCH_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as STSNumber,
      eit.col5         as WestSource,
      eit.col6         as WestDest,
      eit.col7         as EastSource,
      eit.col8         as EastDest,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4009;
```

2.6.27 ONS15454_CMS_Storage_View

```
create or replace view ONS15454_CMS_Storage_View as
select
        storage.JarID,
        storage.FileName as Name,
        storage.JarTime as TimeStamp,
        storage.UserName as UserName,
```

```
        storage.rowid as myrow
from
        ONS15454_CMS_Storage_Table storage;
```

2.6.28 ONS15454_COS_PM_View

```
CREATE OR REPLACE VIEW ONS15454_COS_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       portCOS.Physicalloc,
       portCOS.NEIfIndex,
       portCOS.TimeStamp,
       portCOS.NEIfPmStatus,
       portCOS.InMaintenance,
       portCOS.InterfaceName,
       portCOS.InterfaceDirection,
       portCOS.ClassOfServiceLevel,
       portCOS.PostPolicyPackets,
       portCOS.PostPolicyBytes,
       portCOS.DropPackets,
       portCOS.DropBytes,
       portCOS.IS24H,
       portCOS.ModelType,
       portCOS.ObjectType,
       portCOS.StrObjInstance,
       portCOS.InvalidityList,
       portCOS.rowid as myrow,
      ni.NEAliasID
FROM
       ONS15454_COS_pm_table portCOS,
       ne_info_table ni,
       module_type_table en
WHERE
       portCOS.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       portCOS.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.29 ONS15454_DS1_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_DS1_PM_FE_VIEW  AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.FeDS1FDLPathCV as FDLCodeViolations,
       csonet.FeDS1FDLPathES as FDLErrSecs,
       csonet.FeDS1FDLPathESA as FDLErrSecsTypeA,
       csonet.FeDS1FDLPathESB as FDLErrSecsTypeB,
       csonet.FeDS1FDLPathSES as FDLSevErrSecs,
       csonet.FeDS1FDLPathSEFS as FDLSevErrFrameSecs,
       csonet.FeDS1FDLPathCSS as FDLControlledSlipSecs,
       csonet.FeDS1FDLPathUAS as FDLUnavailableSecs,
       csonet.FeDS1FDLPathFC as FDLFailureCount,
       csonet.FeDS1FDLLineES as FDLLineErrSecs,
       csonet.FEDS1FDLESNP as FDLNPErrSecs,
       csonet.FEDS1FDLSESNP as FDLNPSevErrSecs,
       csonet.FEDS1FDLUASNP as FDLNPUnavailableSecs,
       csonet.FEDS1FDLPATHESR as FDLPEsr,
       csonet.FEDS1FDLPATHSESR as FDLPSesr,
       csonet.DS1ESNP as NPErrSecs,
       csonet.DS1SESNP as NPSevErrSecs,
       csonet.DS1UASNP as NPUnavailableSecs,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       en.ModuleType,
       csonet.IS24H,
```

```
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
        csonet.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csonet.StrCorbaTidAid
FROM
        ONS15454_ds1_pm_table csonet,
        ne_info_table ni,
        module_type_table en
WHERE
        csonet.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.30 ONS15454_DS1_PM_View

```
CREATE OR REPLACE VIEW ONS15454_ds1_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csonet.PhysicalLoc,
        csonet.NEIfIndex as NEIfIndex,
        csonet.TimeStamp as TimeStamp,
        csonet.NEIfPmStatus as NEIfPmStatus,
        csonet.InMaintenance as InMaintenance,
        csonet.DS1LineCodeViolations as LineCodeViolations,
        csonet.DS1LineErrSecs as LineErrSecs,
        csonet.DS1LineSevErrSecs as LineSevErrSecs,
        csonet.DS1RxPathAIS as RxAis,
        csonet.DS1RxPathErrSecs as RxErrSecs,
        csonet.DS1RxPathSAS as RxSas,
        csonet.DS1RxPAthSevErrSecs as RxSevErrSecs,
        csonet.DS1RxPathUnavailableSecs as RxUnavailSecs,
        csonet.DS1RxpathCodeViolations as RxCodeViolations,
        csonet.DS1TxPathAis as TxAis,
        csonet.DS1TxPathErrSecs as TxErrSecs,
        csonet.DS1TxPathSAS as TxSas,
        csonet.DS1TxPathSevErrSecs as TxSevErrSecs,
        csonet.DS1TxPathUnavailableSecs as TxUnavailSecs,
        csonet.DS1TxPathCodeViolations as TxCodeViolations,
        csonet.DS1LineLOSS,
        csonet.DS1ESNP as NPErrSecs,
        csonet.DS1SESNP as NPSevErrSecs,
        csonet.DS1UASNP as NPUnavailableSecs,
        csonet.DS1RXPATHBBE as RxPBbe,
        csonet.DS1RXPATHESR as RxPEsr,
        csonet.DS1RXPATHSESR as RxPSesr,
        csonet.DS1RXPATHBBER as RxPBber,
        csonet.DS1TXPATHBBE as TxPBbe,
        csonet.DS1TXPATHESR as TxPEsr,
        csonet.DS1TXPATHSESR as TxPSesr,
        csonet.DS1TXPATHBBER as TxPBber,
        csonet.DS1RXPATHFC as RxPFc,
        csonet.NEDBAccessID as NEDBAccessID,
        csonet.thresholdList as ThresholdList,
        en.ModuleType,
        csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
        csonet.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csonet.StrCorbaTidAid

FROM
        ONS15454_ds1_pm_table csonet,
        ne_info_table ni,
        module_type_table en
WHERE
        csonet.ModuleType    = en.ModuleType   AND
```

```
        ni.NEModelType       = en.ModelType    AND
        csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.31 ONS15454_DS3_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_ds3_pm_fe_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csonet.PhysicalLoc,
        csonet.NEIfIndex as NEIfIndex,
        csonet.TimeStamp as TimeStamp,
        csonet.NEIfPmStatus as NEIfPmStatus,
        csonet.InMaintenance as InMaintenance,
        csonet.FeDS3CpBitCodeViolations as CPBitCodeViolations,
        csonet.FeDS3CpBitErrSecs as CPBitErrSecs,
        csonet.FeDS3CpBitSAS as CPBitSAS,
        csonet.FeDS3CpBitSevErrSecs as CPBitSevErrSecs,
        csonet.FeDS3CpBitUnavailableSecs as CPBitUnavailableSecs,
        csonet.NEDBAccessID as NEDBAccessID,
        csonet.thresholdList as ThresholdList,
        en.ModuleType,
        csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
        csonet.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csonet.StrCorbaTidAid
FROM
        ONS15454_ds3_pm_table csonet,
        ne_info_table ni,
        module_type_table en
WHERE
        csonet.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.32 ONS15454_DS3_PM_View

```
CREATE OR REPLACE VIEW ONS15454_ds3_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csonet.PhysicalLoc,
        csonet.NEIfIndex as NEIfIndex,
        csonet.TimeStamp as TimeStamp,
        csonet.NEIfPmStatus as NEIfPmStatus,
        csonet.InMaintenance as InMaintenance,
        csonet.DS3LineCodeViolations as LineCodeViolations,
        csonet.DS3LineErrSecs as LineErrSecs,
        csonet.DS3LineSevErrSecs as LineSevErrSecs,
        csonet.DS3LineLoss as LineLoss,
        csonet.DS3PBitAis as PbitAis,
        csonet.DS3PBitCodeViolations as PbitCodeViolations,
        csonet.DS3PBitErrSecs as PbitErrSecs,
        csonet.DS3PbitSas as PbitSas,
        csonet.DS3PbitSevErrSecs as PbitSevErrSecs,
        csonet.DS3PBitUnavailableSecs as PbitUnavailableSecs,
        csonet.DS3cpBitCodeViolations as CpbitCodeViolations,
        csonet.DS3cpBitErrSecs as CpbitErrSecs,
        csonet.DS3cpbitSas as CpbitSas,
        csonet.DS3cpBitSevErrSecs as CpbitSevErrSecs,
        csonet.DS3CpBitUnavailableSecs as CpbitUnavailableSecs,
        csonet.NEDBAccessID as NEDBAccessID,
        csonet.thresholdList as ThresholdList,
        en.ModuleType,
        csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
```

```
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         csonet.StrCorbaTidAid
FROM
       ONS15454_ds3_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.33 ONS15454_DWDM_Ring_View

```
CREATE OR REPLACE VIEW ONS15454_DWDM_RING_VIEW as
select
      ne.NESysid,
      eit.col6        as RingName,
      eit.col3        as WestDirection,
      eit.col5        as EastDirection,
       ne.NEDbaccessid,
      ne.rowid        as myrow
From  ne_info_table     ne,
      eqpt_info_table   eit
Where
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      eit.ObjectIndex  = 235;
```

2.6.34 ONS15454_Enet_PM_View

```
CREATE OR REPLACE VIEW ONS15454_enet_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       enet.Physicalloc,
       enet.NEIfIndex,
       enet.TimeStamp,
       enet.NEIfPmStatus,
       enet.InMaintenance,
       enet.ThresholdList,
       enet.IfInOctets,
       enet.RxPackets,
       enet.IfInUcastPkts,
       enet.IfInMulticastPkts,
       enet.IfInBroadcastPkts,
       enet.IfInErrors,
       enet.IfOutOctets,
       decode(en.Moduletype,102,to_char(enet.TxPackets),
                            103,to_char(enet.TxPackets),
                    118,to_char(enet.TxPackets),
                            119,to_char(enet.TxPackets),
                181,to_char(enet.TxPackets),
                182,to_char(enet.TxPackets),
                183,to_char(enet.TxPackets),
                184,to_char(enet.TxPackets),
                14,to_char(enet.TxPackets),
                15,to_char(enet.TxPackets),
                62,to_char(enet.TxPackets),
                187,to_char(enet.TxPackets),
                120,to_char(enet.TxPackets),
                121,to_char(enet.TxPackets),
                163,to_char(enet.TxPackets),
                164,to_char(enet.TxPackets),
                221,to_char(enet.TxPackets),
                163,to_char(enet.TxPackets),
                164,to_char(enet.TxPackets),
                221,to_char(enet.TxPackets),
                243,to_char(enet.TxPackets),
                229,to_char(enet.TxPackets),
```

```
               271,to_char(enet.TxPackets),
               272,to_char(enet.TxPackets),
               273,to_char(enet.TxPackets),
               283,to_char(enet.TxPackets),
               284,to_char(enet.TxPackets),
               285,to_char(enet.TxPackets),
               293,to_char(enet.TxPackets),
               300,to_char(enet.TxPackets),
               301,to_char(enet.TxPackets),
               330,to_char(enet.TxPackets),
               355,to_char(enet.TxPackets),
           ) as TxPackets,
       enet.IfOutUcastPkts,
       enet.IfOutMulticastPkts,
       enet.IfOutBroadcastPkts,
       enet.Dot3StatsAlignmentErrors,
       enet.Dot3StatsFCSErrors,
       enet.Dot3StatsSingleCollisionFrames,
       enet.Dot3StatsMultiCollisionFrames,
       enet.Dot3StatsDeferredTransmissions,
       enet.Dot3StatsLateCollisions,
       enet.Dot3StatsExcessiveCollisions,
       enet.EtherStatsUndersizePkts,
       enet.EtherStatsFragments,
       enet.EtherStatsOversizePkts,
       enet.EtherStatsJabbers,
       decode(en.Moduletype, 102, to_char(enet.RxTotalErrors),
                          103, to_char(enet.RxTotalErrors),
                          118, to_char(enet.RxTotalErrors),
                          119, to_char(enet.RxTotalErrors),
                             62, to_char(enet.RxTotalErrors),
                          187, to_char(enet.RxTotalErrors),

get_enet_value2(enet.Dot3StatsFCSErrors,enet.Dot3StatsAlignmentErrors,enet.EtherStatsUnder
SizePKTS, enet.ETHERSTATSOVERSIZEPKTS,enet.ETHERSTATSFRAGMENTS,enet.ETHERSTATSJABBERS)
              )  as Rx_TOTAL_ERRORS,
       decode(en.Moduletype, 102, enet.RxRunts,
                             103, enet.RxRunts,
                          118, enet.RxRunts,
                             119, enet.RxRunts,
                          62, enet.RxRunts,
                          187, enet.RxRunts,
             enet.ETHERSTATSFRAGMENTS
           ) as Rx_RUNTS,
       enet.RxGIANTS as rx_giants,
       enet.TxCollisions as Tx_Collisions,
       enet.RxPauseFrames,
       enet.TxPauseFrames,
       enet.RxPktsDropInternalCongestion,
       enet.TxPktsDropInternalCongestion,
       enet.IS24H,
       enet.EtherUtilizationStats,
       enet.RxGMACDropCounts,
       enet.RxThresholdOversizes,
       enet.PortDropCounts,
       enet.TxGiants,
       enet.HDLCPktDrops,
       enet.RxControlFrames,
       enet.RXUnknownOpCodeFrames,
       enet.IfInErrorBytePkts,
       enet.IfInFramingErrorPkts,
       enet.IfInJunkInterPkts,
       enet.IfInDiscards,
       enet.IfOutDiscards,
       enet.Dot3StatsFrameTooLong,
       enet.Dot3StatsCarrierSenseErrors,
       enet.Dot3StatsSQETestErrors,
       enet.EtherStatsPkts64Octets,
       enet.EtherStatsPkts65to127Octets,
       enet.EtherStatsPkts128to255Octets,
       enet.EtherStatsPkts256to511Octets,
       enet.EtherStatsPkts512to1023Octets,
       enet.EtherStatsPkts1024to1518Octets,
       enet.EtherStatsBroadcastPkts,
       enet.EtherStatsMulticastPkts,
       enet.EtherStatsOctets,
```

```
       enet.EtherStatsCollisions,
       enet.EtherStatsCollisionFrames,
       enet.EtherStatsCRCAlignErrors,
       enet.MediaIndStatsRxFramesTruncated,
       enet.MediaIndStatsRxFramesTooLong,
       enet.MediaIndStatsRxFramesBadCRC,
       enet.MediaIndStatsTxFramesBadCRC,
       enet.MediaIndStatsRxShortPkts,
       enet.MediaIndStatsOversizeDropped,
       enet.IfOutErrors,
       enet.EtherStatsPkts,
       enet.Dot3StatsInternalMacTxErrors,
       enet.Dot3StatsInternalMacRxErrors,
       enet.Dot3StatsSymbolErrors,
       enet.RxEtherUtilizationStats,
       enet.TxEtherUtilizationStats,
       enet.StatsLineLastClearedTime,
       enet.MEDIAINDSTATSTXFRAMESTOOLONG,
     enet.ModelType,
       enet.ObjectType,
       enet.StrObjInstance,
       enet.InvalidityList,
       enet.rowid as myrow,
       ni.NEAliasID as NEAliasID,
       enet.Dot3StatsCtrInUnknownOpCodes,
       enet.Dot3StatsInPauseFrames,
       enet.Dot3StatsOutPauseFrames,
       enet.EtherStatsPkts1519to1522Octets
decode(lag.LACPDUsRx,NULL, -1,lag.LACPDUsRx) as LACPDUsRx,
       decode(lag.MarkerPDUsRx,NULL, -1,lag.MarkerPDUsRx) as MarkerPDUsRx,
       decode(lag.MarkerResponsePDUsRx,NULL, -1,lag.MarkerResponsePDUsRx) as
MarkerResponsePDUsRx,
       decode(lag.UnknownRx,NULL, -1,lag.UnknownRx) as UnknownRx,
       decode(lag.IllegalRx,NULL, -1,lag.IllegalRx) as IllegalRx,
       decode(lag.LACPDUsTx,NULL, -1,lag.LACPDUsTx) as LACPDUsTx,
       decode(lag.MarkerPDUsTx,NULL, -1,lag.MarkerPDUsTx) as MarkerPDUsTx,
       decode(lag.MarkerResponsePDUsTx,NULL, -1,lag.MarkerResponsePDUsTx) as
MarkerResponsePDUsTx,
       enet.ETHERSTATSTXFIFOOVERFLOWEVENTS,
       enet.IFHCINOCTETS,
       enet.IFHCINUCASTPKTS,
       enet.IFHCINMULTICASTPKTS,
       enet.IFHCINBROADCASTPKTS,
       enet.IFHCOUTOCTETS,
       enet.IFHCOUTMULTICASTPKTS,
       enet.IFHCOUTBROADCASTPKTS,
       enet.ETHERSTATSHIGHCAPACITYPKTS,
       enet.ETHERSTATSHIGHCAPACITYOCTETS,
       enet.ETHERSTATSHC64OCTETS,
       enet.ETHERSTATSHC65TO127OCTETS,
       enet.ETHERSTATSHC128TO255OCTETS,
       enet.ETHERSTATSHC256TO511OCTETS,
       enet.ETHERSTATSHC512TO1023OCTETS,
       enet.ETHERSTATSHC1024TO1518OCTETS,
       enet.CISRXREPORTS,
       enet.CISRXLEAVES,
       enet.CISTXREPORTS,
       enet.CISTXLEAVES,
       enet.CISTXGENERALQUERIES,
       enet.CISTXGROUPSPECIFICQUERIES,
       enet.CISRXGENERALQUERIES,
       enet.CISRXGROUPSPECIFICQUERIES,
       enet.CISRXVALIDPACKETS,
       enet.CISRXINVALIDPACKETS,
       enet.StrCorbaTidAid

FROM
       ons15454_enet_pm_table enet, left outer join ons15454_LAG_pm_table lag
       on (enet.NEDBAccessID = lag.NEDBAccessID AND
           enet.TIMESTAMP = lag.TIMESTAMP AND
           enet.MODULETYPE = lag.MODULETYPE AND
           enet.PHYSICALLOC = lag.PHYSICALLOC AND
           enet.NEIFINDEX = lag.NEIFINDEX AND
           enet.IS24H = lag.IS24H),
       ne_info_table ni,
       module_type_table en
```

```
WHERE
       enet.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       enet.NEDBAccessID  = ni.NEDBAccessID AND
       enet.OBJECTTYPE   != 5138;
```

2.6.35 ONS15454_ESCON_PM_View

```
CREATE OR REPLACE VIEW ONS15454_ESCON_PM_VIEW AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        enet.Physicalloc,
        enet.NEIfIndex,
        enet.TimeStamp,
        enet.NEIfPmStatus,
        enet.ThresholdList,
        enet.InMaintenance,
        enet.ifInOctets,
        enet.ifOutOctets,
    enet.ifInErrors,
        enet.TxPackets,
        enet.RxPackets,
        enet.ifInDiscards,
        enet.ifOutDiscards,
    enet.mediaIndStatsRxFramesBadCrc,
        enet.StatsLineLastClearedTime,
    enet.RxEtherUtilizationStats,
    enet.TxEtherUtilizationStats,
        enet.IS24H,
        enet.ModelType,
        decode(enet.ObjectType,5138,5131) as ObjectType,
        enet.StrObjInstance,
        enet.InvalidityList,
        enet.rowid as myrow,
    ni.NEALIASID as NEALIASID,
     enet.StrCorbaTidAid
FROM
        ons15454_enet_pm_table enet,
        ne_info_table ni,
        module_type_table en
WHERE
        enet.ModuleType    = en.ModuleType   AND
        ni.NEModelType     = en.ModelType    AND
        enet.NEDBAccessID  = ni.NEDBAccessID AND
        enet.OBJECTTYPE = 5138;
```

2.6.36 ONS15454_FC_PM_View

```
CREATE OR REPLACE VIEW ONS15454_FC_PM_VIEW AS
SELECT
    ni.NESysID,
    ni.NEDbaccessid,
    en.ModuleName,
    en.ModuleType,
    fc.Physicalloc,
    fc.NEIfIndex,
    fc.TimeStamp,
    fc.NEIfPmStatus,
    fc.ThresholdList,
    fc.InMaintenance,
    fc.fibreStatsInvalidOrderedSets,
    fc.fibreStatsEncodingDispErrors,
    fc.fibreStatsRxTotalErrors,
    fc.fibreStatsRxFramesTruncated,
    fc.fibreStatsRxFramesTooLong,
    fc.fibreStatsRxFramesBadCRC,
    fc.fibreStatsRxFrames,
    fc.fibreStatsRxOctets,
```

```
    fc.fibreStatsRxDiscards,
    fc.fibreStatsTxFramesBadCRC,
    fc.fibreStatsTxFrames,
    fc.fibreStatsTxOctets,
    fc.fibreStatsTxDiscards,
    fc.fibreStatsLinkResets,
    fc.gfpStatsRxSBitErrors,
    fc.gfpStatsRxMBitErrors,
    fc.gfpStatsRxTypeInvalid,
    fc.gfpStatsRxSblkCRCErrors,
    fc.RxFCUtilizationStats,
    fc.TxFCUtilizationStats,
    fc.StatsLineLastClearedTime,
    fc.gfpStatsCSFRaised,
    fc.gfpStatsRoundTripLatency,
    fc.IfInErrors,
    fc.FibreStatsLinkRecoveries,
    fc.FCIngressRxDistanceExtBuffers,
    fc.FCEgressTxDistanceExtBuffers,
    fc.FibreStatsRxCredits,
    fc.FibreStatsTxCredits,
    fc.FibreStatsZeroTxCredits,
    fc.TxPackets,
    fc.RxPackets,
    fc.FibreStatsRxRecvrReady,
    fc.FibreStatsTxRecvrReady,
    fc.InvldOrdrdSetDispErrorSum,
    fc.IfOutOversizePkts,
    fc.IS24H,
    fc.ModelType,
    fc.ObjectType,
    fc.StrObjInstance,
    fc.InvalidityList,
    fc.rowid as myrow,
    ni.NEAliasID as NEAliasID,
    fc.StrCorbaTidAid
    fc.fibreStatsTxFramesTruncated,
    fc.fibreStatsTxFramesTooLong,
    fc.IfOutErrors
FROM
    ons15454_fc_pm_table fc,
    ne_info_table ni,
    module_type_table en
WHERE
    fc.ModuleType    = en.ModuleType   AND
    ni.NEModelType     = en.ModelType    AND
    fc.NEDBAccessID  = ni.NEDBAccessID AND
    fc.OBJECTTYPE <> 5137;
```

2.6.37 ONS15454_GFP_PM_View

```
CREATE OR REPLACE VIEW ONS15454_gfp_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       gfp.Physicalloc,
       gfp.NEIfIndex,
       gfp.TimeStamp,
       gfp.NEIfPmStatus,
       gfp.InMaintenance,
       gfp.GfpStatsRxSBitErrors,
       gfp.GfpStatsRxMBitErrors,
       gfp.GfpStatsRxTypeInvalid,
       gfp.GfpStatsRxCRCErrors,
       gfp.GfpStatsRxCIDInvalid,
       gfp.GfpStatsCSFRaised,
       gfp.GfpStatsRxFrame,
       gfp.GfpStatsTxFrame,
       gfp.GfpStatsRxOctets,
       gfp.GfpStatsTxOctets,
       gfp.GfpStatsLFDRaised,
```

```
        gfp.GfpStatsRoundTripLatency,
        gfp.GfpStatsRxSBlkCRCErrors,
        gfp.GfpStatsRxDistanceExtBuffers,
        gfp.GfpStatsTxDistanceExtBuffers,
        gfp.gfpRxCmfFrame,
        gfp.gfpTxCmfFrame,
        gfp.ifInPayloadCrcErrors,
        gfp.thresholdList as ThresholdList,
        gfp.IS24H,
        gfp.ModelType,
        gfp.ObjectType,
        gfp.StrObjInstance,
        gfp.InvalidityList,
        gfp.rowid as myrow,
        ni.NEAliasID as NEAliasID,
         gfp.StrCorbaTidAid
FROM
        ons15454_gfp_pm_table gfp,
        ne_info_table ni,
        module_type_table en
WHERE
        gfp.ModuleType    = en.ModuleType   AND
        ni.NEModelType     = en.ModelType    AND
        gfp.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.38 ONS15454_Inventory_View

```
CREATE OR REPLACE VIEW ONS15454_inventory_view AS
SELECT
        info.NESysID as NESysId,
        inv.Name as Name,
        inv.HWPartNumber as HdrPartNum,
        inv.CleiCode as CleiCode,
        inv.HardwareRevision as HardRevision,
        inv.SerialNumber as SerialNum,
        inv.PhysicalLoc as PhysicalLoc,
        module.ModuleName as EquipType,
        inv.AdminState as AdminState,
        inv.ServiceState as ServiceState,
        inv.Name as ActualEquipType,
        inv.EquipmentState as EquipState,
        inv.FirmwareVersion as FirmwareVersion,
        inv.ApplicationFileName as ApplicationFileName,
        inv.EntityID as EntityId,
        inv.ExtraInfo as ExtraInfo,
        info.NEDBAccessId,
        module.ModuleType,
        inv.InventoryCode,
        inv.ProductId,
        inv.VersionId,
        inv.rowid as myrow,
        info.NEAliasID as NEAliasID
          get_inventory_note(inv.SerialNumber) as Note
FROM
        ne_info_table info,
        ONS15454_ne_inventory_table inv,
        module_type_table module
WHERE
        info.NEDBAccessID = inv.NEDBAccessID AND
        info.NEModelType  = module.ModelType AND
        inv.EquipmentType = module.ModuleType;
```

2.6.39 ONS15454_IPSLA_PM_View

```
CREATE OR REPLACE VIEW ONS15454_IPSLA_PM_VIEW AS
SELECT
         ni.NESysID,
         ni.NEDbaccessid,
         en.ModuleName,
         en.ModuleType,
         ipsla.Physicalloc,
         ipsla.NEIfIndex,
         ipsla.TimeStamp,
```

```
        ipsla.NEIfPmStatus,
        ipsla.ThresholdList,
        ipsla.InMaintenance,
        ipsla.rttMonStatsStartTimeIndex,
        ipsla.rttMonStatsCompletions,
        ipsla.rttMonStatsOverThresholds,
        ipsla.rttMonStatsNumOfRTT,
        ipsla.rttMonStatsRTTSum,
        ipsla.rttMonStatsRTTSum2Low,
        ipsla.rttMonStatsRTTSum2High,
        ipsla.rttMonStatsRTTMin,
        ipsla.rttMonStatsRTTMax,
        ipsla.rttMonStatsMinOfPositivesSD,
        ipsla.rttMonStatsMaxOfPositivesSD,
        ipsla.rttMonStatsNumOfPositivesSD,
        ipsla.rttMonStatsSumOfPositivesSD,
        ipsla.rttMonStatsSum2PositivesSDLow,
        ipsla.rttMonStatsSum2PositivesSDHigh,
        ipsla.rttMonStatsMinOfNegativesSD,
        ipsla.rttMonStatsMaxOfNegativesSD,
        ipsla.rttMonStatsNumOfNegativesSD,
        ipsla.rttMonStatsSumOfNegativesSD,
        ipsla.rttMonStatsSum2NegativesSDLow,
        ipsla.rttMonStatsSum2NegativesSDHigh,
        ipsla.rttMonStatsMinOfPositivesDS,
        ipsla.rttMonStatsMaxOfPositivesDS,
        ipsla.rttMonStatsNumOfPositivesDS,
        ipsla.rttMonStatsSumOfPositivesDS,
        ipsla.rttMonStatsSum2PositivesDSLow,
        ipsla.rttMonStatsSum2PositivesDSHigh,
        ipsla.rttMonStatsMinOfNegativesDS,
        ipsla.rttMonStatsMaxOfNegativesDS,
        ipsla.rttMonStatsNumOfNegativesDS,
        ipsla.rttMonStatsSumOfNegativesDS,
        ipsla.rttMonStatsSum2NegativesDSLow,
        ipsla.rttMonStatsSum2NegativesDSHigh,
        ipsla.rttMonStatsPacketLossSD,
        ipsla.rttMonStatsPacketLossDS,
        ipsla.rttMonStatsPacketOutOfSequence,
        ipsla.rttMonStatsPacketMIA,
        ipsla.rttMonStatsPacketLateArrival,
        ipsla.rttMonStatsError,
        ipsla.rttMonStatsBusies,
        ipsla.rttMonStatsOWSumSD,
        ipsla.rttMonStatsOWSum2SDLow,
        ipsla.rttMonStatsOWSum2SDHigh,
        ipsla.rttMonStatsOWMinSD,
        ipsla.rttMonStatsOWMaxSD,
        ipsla.rttMonStatsOWSumDS,
        ipsla.rttMonStatsOWSum2DSLow,
        ipsla.rttMonStatsOWSum2DSHigh,
        ipsla.rttMonStatsOWMinDS,
        ipsla.rttMonStatsOWMaxDS,
        ipsla.rttMonStatsNumOfOW,
        ipsla.rttMonStatsOWMinSDNew,
        ipsla.StatsLineLastClearedTime,
        ipsla.IS24H,
        ipsla.ModelType,
        ipsla.ObjectType,
        ipsla.StrObjInstance,
        ipsla.InvalidityList,
        ipsla.rowid as myrow,
        ni.NEAliasID as NEAliasID,
         ipsla.StrCorbaTidAid
FROM
        ons15454_ipsla_pm_table ipsla,
        ne_info_table ni,
        module_type_table en
WHERE
        ipsla.ModuleType    = en.ModuleType   AND
        ni.NEModelType     = en.ModelType    AND
        ipsla.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.40 ONS15454_LEX_PM_View

```
CREATE OR REPLACE VIEW ONS15454_lex_pm_view AS
SELECT
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       lex.Physicalloc,
       lex.NEIfIndex,
       lex.TimeStamp,
       lex.NEIfPmStatus,
       lex.InMaintenance,
       lex.MediaIndStatsRxFramesBadCRC,
       lex.HdlcInOctets,
       lex.HdlcRxAborts,
       lex.HdlcOutOctets,
       lex.IfInOctets,
       lex.IfOutOctets,
       lex.PosStatsRxPackets,
       lex.PosStatsTxPackets,
       lex.ifInPayloadCrcErrors,
       lex.PosStatsRxTotalErrors,
       lex.IS24H,
       lex.ModelType,
       lex.ObjectType,
       lex.StrObjInstance,
       lex.InvalidityList,
       lex.rowid as myrow,
    ni.NEAliasID as NEAliasID,
       lex.StrCorbaTidAid
FROM
       ons15454_pos_pm_table lex,
       ne_info_table ni,
       module_type_table en
WHERE
       lex.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       lex.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.41 ONS15454_ISC_PM_View

```
CREATE OR REPLACE VIEW ONS15454_ISC_PM_VIEW AS
SELECT
    ni.NESysID,
    ni.NEDbaccessid,
    en.ModuleName,
    en.ModuleType,
    fc.Physicalloc,
    fc.NEIfIndex,
    fc.TimeStamp,
    fc.NEIfPmStatus,
    fc.ThresholdList,
    fc.InMaintenance,
    fc.fibreStatsRxOctets,
    fc.fibreStatsTxOctets,
    fc.TxPackets,
    fc.RxPackets,
    fc.InvldOrdrdSetDispErrorSum,
        fc.RxFCUtilizationStats,
        fc.TxFCUtilizationStats,
        fc.StatsLineLastClearedTime,
    fc.IS24H,
    fc.ModelType,
    decode(fc.ObjectType,5137,5131) as ObjectType,
        fc.StrObjInstance,
    fc.InvalidityList,
    fc.rowid as myrow,
    ni.NEAliasID as NEAliasID,
     fc.StrCorbaTidAid
FROM
    ons15454_fc_pm_table fc,
    ne_info_table ni,
```

```
    module_type_table en
WHERE
    fc.ModuleType    = en.ModuleType   AND
    ni.NEModelType     = en.ModelType    AND
    fc.NEDBAccessID  = ni.NEDBAccessID AND
    fc.OBJECTTYPE = 5137;
```

2.6.42 ONS15454_OPTICS_PM_VIEW

```
CREATE OR REPLACE VIEW ONS15454_OPTICS_PM_VIEW AS
SELECT
        ni.NESysID as NESYSID,
        en.ModuleName as MODULENAME,
        phy.PhysicalLoc as PHYSICALLOC,
        phy.NEIfIndex as NEIFINDEX,
        phy.TimeStamp as TIMESTAMP,
        phy.NEIfPMStatus as NEIFPMSTATUS,
        phy.inMaintenance as INMAINTENANCE,
        phy.MinLaserBias as MINLASERBIAS,
        phy.AvgLaserBias as AVGLASERBIAS,
        phy.MaxLaserBias as MAXLASERBIAS,
        phy.MinLaserTemp as MINLASERTEMP,
        phy.AvgLaserTemp as AVGLASERTEMP,
        phy.MaxLaserTemp as MAXLASERTEMP,
        phy.MinReceivedLaserTemp as MINRECEIVEDLASERTEMP,
        phy.AvgReceivedLaserTemp as AVGRECEIVEDLASERTEMP,
        phy.MaxReceivedLaserTemp as MAXRECEIVEDLASERTEMP,
        phy.MinTransReceiverVoltage as MINTRANSRECEIVERVOLTAGE,
        phy.AvgTransReceiverVoltage as AVGTRANSRECEIVERVOLTAGE,
        phy.MaxTransReceiverVoltage as MAXTRANSRECEIVERVOLTAGE,
        phy.MinTransmittedPower as MINTRANSMITTEDPOWER,
        phy.AvgTransmittedPower as AVGTRANSMITTEDPOWER,
        phy.MaxTransmittedPower as MAXTRANSMITTEDPOWER,
        phy.MinReceivedPower as MINRECEIVEDPOWER,
        phy.AvgReceivedPower as AVGRECEIVEDPOWER,
        phy.MaxReceivedPower as MAXRECEIVEDPOWER,
        phy.MinPassThroughPower as MINPASSTHROUGHPOWER,
       phy.AvgPassThroughPower as AVGPASSTHROUGHPOWER,
       phy.MaxPassThroughPower as MAXPASSTHROUGHPOWER,
        ni.NEDbaccessid as NEDBACCESSID,
        phy.thresholdList as ThresholdList,
        en.ModuleType as MODULETYPE,
        phy.IS24H  as IS24H,
        phy.ModelType,
        phy.ObjectType,
        phy.StrObjInstance,
        phy.InvalidityList,
        phy.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        phy.StrCorbaTidAid
FROM
        ONS15454_PHY_LAYER_PM_Table phy,
        ne_info_table ni,
        module_type_table en
WHERE
        phy.ModuleType   = en.ModuleType   AND
        ni.NEModelType   = en.ModelType    AND
        phy.NEDBAccessID = ni.NEDBAccessID;
```

2.6.43 ONS15454_OSPF_Area_Range_View

```
create or replace view ONS15454_OSPF_AREA_RANGE_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as RangeAddress,
      eit.col5         as MaskLength,
      eit.col6         as Mask,
      eit.col7         as Advertise,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType  as ModuleType,
```

```
      ne.rowid        as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4003;
```

2.6.44 ONS15454_OTN_Fec_PM_View

```
CREATE OR REPLACE VIEW ONS15454_OTN_fec_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       cotn.PhysicalLoc,
       cotn.NEIfIndex as NEIfIndex,
       cotn.TimeStamp as TimeStamp,
       cotn.NEIfPmStatus as NEIfPmStatus,
       cotn.InMaintenance as InMaintenance,
       cotn.BitErrsCorrected as BitErrsCorrected,
       cotn.ByteErrsCorrected as ByteErrsCorrected,
       cotn.ZeroBitErrsDetected as ZeroBitErrsDetected,
       cotn.OneBitErrsDetected as OneBitErrsDetected,
       cotn.UnCorrectableWord as UnCorrectableWord,
       cotn.NEDBAccessID as NEDBAccessID,
       cotn.thresholdList as ThresholdList,
    en.ModuleType,
       cotn.IS24H,
       cotn.ModelType,
       cotn.ObjectType,
       cotn.StrObjInstance,
       cotn.InvalidityList,
       cotn.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         cotn.StrCorbaTidAid
FROM
       ONS15454_OTN_FEC_PM_Table cotn,
       ne_info_table ni,
       module_type_table en
WHERE
       cotn.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       cotn.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.45 ONS15454_OTN_Path_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_otn_path_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       cotn.PhysicalLoc,
       cotn.NEIfIndex as NEIfIndex,
       cotn.TimeStamp as TimeStamp,
       cotn.NEIfPmStatus as NEIfPmStatus,
       cotn.InMaintenance InMaintenance,
       cotn.NEDBAccessID as NEDBAccessId,
       en.ModuleType,
       cotn.IS24H,
       cotn.thresholdList as ThresholdList,
       cotn.FeErrSecs as FeErrSecs,
       cotn.FeSevErrSecs as FeSevErrSecs,
       cotn.FeUnavailableSecs as FeUnavailableSecs,
       cotn.FeBackGroundBlockErrs as FeBackGroundBlockErrs,
       cotn.FeFailureCount as FeFailureCount,
       cotn.FeErrSecsRatio as FeErrSecsRatio,
       cotn.FeSevErrSecsRatio as FeSevErrSecsRatio,
       cotn.FeBackGroundBlockErrRatio as FeBackGroundBlockErrRatio,
       cotn.ModelType,
       cotn.ObjectType,
```

```
        cotn.StrObjInstance,
        cotn.InvalidityList,
        cotn.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          cotn.StrCorbaTidAid
FROM
        ONS15454_OTN_PATH_PM_Table cotn,
        ne_info_table ni,
        module_type_table en
WHERE
        cotn.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        cotn.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.46 ONS15454_OTN_Path_PM_View

```
CREATE OR REPLACE VIEW ONS15454_OTN_path_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        cotn.PhysicalLoc,
        cotn.NEIfIndex as NEIfIndex,
        cotn.TimeStamp as TimeStamp,
        cotn.NEIfPmStatus as NEIfPmStatus,
        cotn.InMaintenance as InMaintenance,
        cotn.ErrSecs         as ErrSecs,
        cotn.SevErrSecs      as SevErrSecs,
        cotn.UnavailableSecs as UnAvailableSecs,
        cotn.BackGroundBlockErr as BackGroundBlockErr,
        cotn.FailureCount as FailureCount,
        cotn.ErrSecsRatio as ErrSecsRatio,
        cotn.SevErrSecsRatio as SevErrSecsRatio,
        cotn.BackGroundBlockErrRatio as BackGroundBlockErrRatio,
        cotn.NEDBAccessID as NEDBAccessID,
        cotn.thresholdList as ThresholdList,
        en.ModuleType,
        cotn.IS24H,
        cotn.ModelType,
        cotn.ObjectType,
        cotn.StrObjInstance,
        cotn.InvalidityList,
        cotn.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          cotn.StrCorbaTidAid
FROM
        ONS15454_OTN_PATH_PM_Table cotn,
        ne_info_table ni,
        module_type_table en
WHERE
        cotn.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        cotn.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.47 ONS15454_OTN_Sec_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_otn_sec_pm_fe_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        cotn.PhysicalLoc,
        cotn.NEIfIndex as NEIfIndex,
        cotn.TimeStamp as TimeStamp,
        cotn.NEIfPmStatus as NEIfPmStatus,
        cotn.InMaintenance InMaintenance,
        cotn.NEDBAccessID as NEDBAccessId,
        en.ModuleType,
        cotn.IS24H,
        cotn.thresholdList as ThresholdList,
        cotn.FeErrSecs as FeErrSecs,
        cotn.FeSevErrSecs as FeSevErrSecs,
        cotn.FeUnavailableSecs as FeUnavailableSecs,
```

```
       cotn.FeBackGroundBlockErrs as FeBackGroundBlockErrs,
       cotn.FeFailureCount as FeFailureCount,
       cotn.FeErrSecsRatio as FeErrSecsRatio,
       cotn.FeSevErrSecsRatio as FeSevErrSecsRatio,
       cotn.FeBackGroundBlockErrRatio as FeBackGroundBlockErrRatio,
       cotn.ModelType,
       cotn.ObjectType,
       cotn.StrObjInstance,
       cotn.InvalidityList,
       cotn.rowid as myrow,
       ni.NEAliasID as NEAliasID,
        cotn.StrCorbaTidAid

FROM
       ONS15454_OTN_SEC_PM_Table cotn,
       ne_info_table ni,
       module_type_table en
WHERE
       cotn.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       cotn.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.48 ONS15454_OTN_Sec_PM_View

```
CREATE OR REPLACE VIEW ONS15454_OTN_sec_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       cotn.PhysicalLoc,
       cotn.NEIfIndex as NEIfIndex,
       cotn.TimeStamp as TimeStamp,
       cotn.NEIfPmStatus as NEIfPmStatus,
       cotn.InMaintenance as InMaintenance,
       cotn.ErrSecs         as ErrSecs,
       cotn.SevErrSecs      as SevErrSecs,
       cotn.UnavailableSecs as UnAvailableSecs,
       cotn.BackGroundBlockErr as BackGroundBlockErr,
       cotn.FailureCount as FailureCount,
       cotn.ErrSecsRatio as ErrSecsRatio,
       cotn.SevErrSecsRatio as SevErrSecsRatio,
       cotn.BackGroundBlockErrRatio as BackGroundBlockErrRatio,
       cotn.NEDBAccessID as NEDBAccessID,
       cotn.thresholdList as ThresholdList,
       en.ModuleType,
       cotn.IS24H,
       cotn.ModelType,
       cotn.ObjectType,
       cotn.StrObjInstance,
       cotn.InvalidityList,
       cotn.rowid as myrow,
       ni.NEAliasID as NEAliasID,
     cotn.StrCorbaTidAid
FROM
       ONS15454_OTN_SEC_PM_Table cotn,
       ne_info_table ni,
       module_type_table en
WHERE
       cotn.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       cotn.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.49 ONS15454_Phy_Layer_PM_View

```
CREATE OR REPLACE VIEW ONS15454_PHY_LAYER_PM_VIEW AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        phy.PhysicalLoc,
        phy.NEIfIndex,
```

```
        phy.TimeStamp,
        phy.NEIfPMStatus,
        phy.inMaintenance,
        phy.LaserBiasCurrent,
        phy.OpticalPowerTransmitted,
        phy.OpticalPowerReceived,
        phy.thresholdList as ThresholdList,
        phy.IS24H,
        phy.ModelType,
        phy.ObjectType,
        phy.StrObjInstance,
        phy.InvalidityList,
        phy.rowid as myrow,
    ni.NEAliasID as NEAliasID,
        phy.StrCorbaTidAid

FROM
        ONS15454_PHY_LAYER_PM_Table phy,
        ne_info_table ni,
        module_type_table en
WHERE
        phy.ModuleType   = en.ModuleType   AND
        ni.NEModelType   = en.ModelType    AND
        phy.NEDBAccessID = ni.NEDBAccessID;
```

2.6.50 ONS15454_POS_PM_View

```
CREATE OR REPLACE VIEW ONS15454_pos_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       pos.Physicalloc,
       pos.NEIfIndex,
       pos.TimeStamp,
       pos.NEIfPmStatus,
       pos.InMaintenance,
       pos.PosStatsRxPreHdlcBytes,
       pos.PosStatsRxPostHdlcBytes,
       pos.PosStatsRxPackets,
       pos.PosStatsRxNormalPackets,
       pos.PosStatsRxShorts,
       pos.PosStatsRxRunts,
       pos.PosStatsRxLongs,
       pos.PosStatsRxTotalErrors,
       pos.PosStatsRxCrcErrors,
       pos.PosStatsRxInputDropPackets,
       pos.PosStatsRxInputAbortPackets,
       pos.PosStatsTxPreHdlcBytes,
       pos.PosStatsTxPostHdlcBytes,
       pos.PosStatsTxPackets,
       pos.PosStatsDropCounts,
       pos.EtherStatsDropEvents,
       pos.GfpStatsRxSBitErrors,
       pos.GfpStatsRxMBitErrors,
       pos.GfpStatsRxTypeInvalid,
       pos.GfpStatsRxCRCErrors,
       pos.GfpStatsRxCIDInvalid,
       pos.GfpStatsCSFRaised,
       pos.GfpStatsRxFrame,
       pos.GfpStatsTxFrame,
       pos.GfpStatsRxOctets,
       pos.GfpStatsTxOctets,
       pos.GfpStatsLFDRaised,
       pos.MediaIndStatsRxFramesTruncated,
       pos.MediaIndStatsRxFramesTooLong,
       pos.MediaIndStatsRxFramesBadCRC,
       pos.MediaIndStatsRxShortPkts,
       pos.hdlcInOctets,
       pos.hdlcRxAborts,
       pos.hdlcOutOctets,
       pos.ifInOctets,
       pos.ifOutOctets,
```

```
       pos.rxPktsDropInternalCongestion,
       pos.ifInDiscards,
       pos.RxEtherUtilizationStats as Rx_Utilization_Stats,
       pos.TxEtherUtilizationStats as Tx_Utilization_Stats,
       pos.StatsLineLastClearedTime,
       pos.ifInPayloadCrcErrors,
       pos.ifOutPayloadCrcErrors,
       pos.ifOutOversizePackets,
       pos.HDLCPKTDROPS,
       pos.ifOutDiscards,
       pos.thresholdList as ThresholdList,
       pos.IS24H,
       pos.ModelType,
       pos.ObjectType,
       pos.StrObjInstance,
       pos.InvalidityList,
       pos.rowid as myrow,
       ni.NEAliasID as NEAliasID,
     pos.StrCorbaTidAid

FROM
       ons15454_pos_pm_table pos,
       ne_info_table ni,
       module_type_table en
WHERE
       pos.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       pos.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.51 ONS15454_Ring_Map_View

```
CREATE or REPLACE view ONS15454_RING_MAP_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as NodeId,
      eit.col5         as IPAddress,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4008;
```

2.6.52 ONS15454_RPR_Client_In_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_CLIENT_IN_PM_VIEW AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        rpr.Physicalloc,
        rpr.NEIfIndex,
        rpr.TimeStamp,
        rpr.NEIfPmStatus,
        rpr.ThresholdList,
        rpr.InMaintenance,
        rpr.ClientInUcastClassCFrames,
        rpr.ClientInUcastClassCOctets,
        rpr.ClientInMcastClassCFrames,
        rpr.ClientInMcastClassCOctets,
```

```
        rpr.ClientInUcastClassBEirFrames,
        rpr.ClientInUcastClassBEirOctets,
        rpr.ClientInMcastClassBEirFrames,
        rpr.ClientInMcastClassBEirOctets,
        rpr.ClientInUcastClassBCirFrames,
        rpr.ClientInUcastClassBCirOctets,
        rpr.ClientInMcastClassBCirFrames,
        rpr.ClientInMcastClassBCirOctets,
        rpr.ClientInUcastClassAFrames,
        rpr.ClientInUcastClassAOctets,
        rpr.ClientInMcastClassAFrames,
        rpr.ClientInMcastClassAOctets,
        rpr.ClientInBcastFrames,
        rpr.IS24H,
        rpr.ModelType,
        rpr.ObjectType,
        rpr.StrObjInstance,
        rpr.InvalidityList,
        rpr.rowid as myrow,
        ni.NEAliasID as NEAliasID,
    rpr.TimeLastCleared,
     rpr.StrCorbaTidAid
FROM
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.53 ONS15454_RPR_ClientOut_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_CLIENTOUT_PM_VIEW  AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        rpr.Physicalloc,
        rpr.NEIfIndex,
        rpr.TimeStamp,
        rpr.NEIfPmStatus,
        rpr.ThresholdList,
        rpr.InMaintenance,
        rpr.ClientOutUcastClassCFrames,
        rpr.ClientOutUcastClassCOctets,
        rpr.ClientOutMcastClassCFrames,
        rpr.ClientOutMcastClassCOctets,
        rpr.ClientOutUcastClassBEirFrames,
        rpr.ClientOutUcastClassBEirOctets,
        rpr.ClientOutMcastClassBEirFrames,
        rpr.ClientOutMcastClassBEirOctets,
        rpr.ClientOutUcastClassBCirFrames,
        rpr.ClientOutUcastClassBCirOctets,
        rpr.ClientOutMcastClassBCirFrames,
        rpr.ClientOutMcastClassBCirOctets,
        rpr.ClientOutUcastClassAFrames,
        rpr.ClientOutUcastClassAOctets,
        rpr.ClientOutMcastClassAFrames,
        rpr.ClientOutMcastClassAOctets,
        rpr.ClientOutBcastFrames,
        rpr.IS24H,
        rpr.ModelType,
        rpr.ObjectType,
        rpr.StrObjInstance,
        rpr.InvalidityList,
        rpr.rowid as myrow,
        ni.NEAliasID as NEAliasID,
    RPR_CLIENTOUT_PM_VIEW,
     rpr.StrCorbaTidAid
FROM
```

```
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.54 ONS15454_RPR_Error_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_ERROR_PM_VIEW AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       rpr.Physicalloc,
       rpr.NEIfIndex,
       rpr.TimeStamp,
       rpr.NEIfPmStatus,
       rpr.ThresholdList,
       rpr.InMaintenance,
       rpr.ErrorBadParityFrames,
       rpr.ErrorBadHecFrames,
       rpr.ErrorTtlExpFrames,
       rpr.ErrorTooLongFrames,
       rpr.ErrorTooShortFrames,
       rpr.ErrorBadFcsFrames,
       rpr.ErrorSelfSrcUcastFrames,
       rpr.ErrorPmdAbortFrames,
       rpr.ErrorBadAddrFrames,
       rpr.ErrorContainedFrames,
       rpr.ErrorScffErrors,
       rpr.PortCounterError,
       rpr.ErrorOverSizeFrames,
       rpr.IS24H,
       rpr.ModelType,
       rpr.ObjectType,
       rpr.StrObjInstance,
       rpr.InvalidityList,
       rpr.rowid as myrow,
       ni.NEAliasID as NEAliasID,
    RPR_ERROR_PM_VIEW,
     rpr.StrCorbaTidAid
FROM
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.55 ONS15454_RPR_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_PM_VIEW AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       rpr.Physicalloc,
       rpr.NEIfIndex,
       rpr.TimeStamp,
       rpr.NEIfPmStatus,
       rpr.ThresholdList,
           rpr.InMaintenance,
       rpr.SpanInUcastClassCFrames,
       rpr.SpanInUcastClassCOctets,
       rpr.SpanInMcastClassCFrames,
       rpr.SpanInMcastClassCOctets,
       rpr.SpanInUcastClassBEirFrames,
```

```
rpr.SpanInUcastClassBEirOctets,
rpr.SpanInMcastClassBEirFrames,
rpr.SpanInMcastClassBEirOctets,
rpr.SpanInUcastClassBCirFrames,
rpr.SpanInUcastClassBCirOctets,
rpr.SpanInMcastClassBCirFrames,
rpr.SpanInMcastClassBCirOctets,
rpr.SpanInUcastClassAFrames,
rpr.SpanInUcastClassAOctets,
rpr.SpanInMcastClassAFrames,
rpr.SpanInMcastClassAOctets,
rpr.SpanInCtrlFrames,
rpr.SpanInOamEchoFrames,
rpr.SpanInOamFlushFrames,
rpr.SpanInOamOrgFrames,
rpr.SpanInTopoAtdFrames,
rpr.SpanInTopoChkSumFrames,
rpr.SpanInTopoTpFrames,
rpr.SpanOutUcastClassCFrames,
rpr.SpanOutUcastClassCOctets,
rpr.SpanOutMcastClassCFrames,
rpr.SpanOutMcastClassCOctets,
rpr.SpanOutUcastClassBEirFrames,
rpr.SpanOutUcastClassBEirOctets,
rpr.SpanOutMcastClassBEirFrames,
rpr.SpanOutMcastClassBEirOctets,
rpr.SpanOutUcastClassBCirFrames,
rpr.SpanOutUcastClassBCirOctets,
rpr.SpanOutMcastClassBCirFrames,
rpr.SpanOutMcastClassBCirOctets,
rpr.SpanOutUcastClassAFrames,
rpr.SpanOutUcastClassAOctets,
rpr.SpanOutMcastClassAFrames,
rpr.SpanOutMcastClassAOctets,
rpr.SpanOutCtrlFrames,
rpr.SpanOutOamEchoFrames,
rpr.SpanOutOamFlushFrames,
rpr.SpanOutOamOrgFrames,
rpr.SpanOutTopoAtdFrames,
rpr.SpanOutTopoChkSumFrames,
rpr.SpanOutTopoTpFrames,
rpr.ClientInUcastClassCFrames,
rpr.ClientInUcastClassCOctets,
rpr.ClientInMcastClassCFrames,
rpr.ClientInMcastClassCOctets,
rpr.ClientInUcastClassBEirFrames,
rpr.ClientInUcastClassBEirOctets,
rpr.ClientInMcastClassBEirFrames,
rpr.ClientInMcastClassBEirOctets,
rpr.ClientInUcastClassBCirFrames,
rpr.ClientInUcastClassBCirOctets,
rpr.ClientInMcastClassBCirFrames,
rpr.ClientInMcastClassBCirOctets,
rpr.ClientInUcastClassAFrames,
rpr.ClientInUcastClassAOctets,
rpr.ClientInMcastClassAFrames,
rpr.ClientInMcastClassAOctets,
rpr.ClientInBcastFrames,
rpr.ClientOutUcastClassCFrames,
rpr.ClientOutUcastClassCOctets,
rpr.ClientOutMcastClassCFrames,
rpr.ClientOutMcastClassCOctets,
rpr.ClientOutUcastClassBEirFrames,
rpr.ClientOutUcastClassBEirOctets,
rpr.ClientOutMcastClassBEirFrames,
rpr.ClientOutMcastClassBEirOctets,
rpr.ClientOutUcastClassBCirFrames,
rpr.ClientOutUcastClassBCirOctets,
rpr.ClientOutMcastClassBCirFrames,
rpr.ClientOutMcastClassBCirOctets,
rpr.ClientOutUcastClassAFrames,
rpr.ClientOutUcastClassAOctets,
rpr.ClientOutMcastClassAFrames,
rpr.ClientOutMcastClassAOctets,
rpr.ClientOutBcastFrames,
rpr.ErrorBadParityFrames,
```

```
        rpr.ErrorBadHecFrames,
        rpr.ErrorTtlExpFrames,
        rpr.ErrorTooLongFrames,
        rpr.ErrorTooShortFrames,
        rpr.ErrorBadFcsFrames,
        rpr.ErrorSelfSrcUcastFrames,
        rpr.ErrorPmdAbortFrames,
        rpr.ErrorBadAddrFrames,
        rpr.ErrorContainedFrames,
        rpr.ErrorScffErrors,
        rpr.PortCounterError,
        rpr.ErrorOverSizeFrames,
        rpr.IS24H,
        rpr.ModelType,
        rpr.ObjectType,
        rpr.StrObjInstance,
        rpr.InvalidityList,
        rpr.rowid as myrow,
        ni.NEAliasID as NEAliasID,
        rpr.TimeLastCleared,
         rpr.StrCorbaTidAid
FROM
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.56 ONS15454_RPR_Span_In_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_SPAN_IN_PM_VIEW AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        rpr.Physicalloc,
        rpr.NEIfIndex,
        rpr.TimeStamp,
        rpr.NEIfPmStatus,
        rpr.ThresholdList,
        rpr.InMaintenance,
        rpr.SpanInUcastClassCFrames,
        rpr.SpanInUcastClassCOctets,
        rpr.SpanInMcastClassCFrames,
        rpr.SpanInMcastClassCOctets,
        rpr.SpanInUcastClassBEirFrames,
        rpr.SpanInUcastClassBEirOctets,
        rpr.SpanInMcastClassBEirFrames,
        rpr.SpanInMcastClassBEirOctets,
        rpr.SpanInUcastClassBCirFrames,
        rpr.SpanInUcastClassBCirOctets,
        rpr.SpanInMcastClassBCirFrames,
        rpr.SpanInMcastClassBCirOctets,
        rpr.SpanInUcastClassAFrames,
        rpr.SpanInUcastClassAOctets,
        rpr.SpanInMcastClassAFrames,
        rpr.SpanInMcastClassAOctets,
        rpr.SpanInCtrlFrames,
        rpr.SpanInOamEchoFrames,
        rpr.SpanInOamFlushFrames,
        rpr.SpanInOamOrgFrames,
        rpr.SpanInTopoAtdFrames,
        rpr.SpanInTopoChkSumFrames,
        rpr.SpanInTopoTpFrames,
        rpr.IS24H,
        rpr.ModelType,
        rpr.ObjectType,
        rpr.StrObjInstance,
        rpr.InvalidityList,
        rpr.rowid as myrow,
```

```
        ni.NEAliasID as NEAliasID,
    rpr.TimeLastCleared,
     rpr.StrCorbaTidAid
FROM
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.57 ONS15454_RPR_Span_Out_PM_View

```
CREATE OR REPLACE VIEW ONS15454_RPR_SPAN_OUT_PM_VIEW AS
SELECT
        ni.NESysID,
        ni.NEDbaccessid,
        en.ModuleName,
        en.ModuleType,
        rpr.Physicalloc,
        rpr.NEIfIndex,
        rpr.TimeStamp,
        rpr.NEIfPmStatus,
        rpr.ThresholdList,
        rpr.InMaintenance,
        rpr.SpanOutUcastClassCFrames,
        rpr.SpanOutUcastClassCOctets,
        rpr.SpanOutMcastClassCFrames,
        rpr.SpanOutMcastClassCOctets,
        rpr.SpanOutUcastClassBEirFrames,
        rpr.SpanOutUcastClassBEirOctets,
        rpr.SpanOutMcastClassBEirFrames,
        rpr.SpanOutMcastClassBEirOctets,
        rpr.SpanOutUcastClassBCirFrames,
        rpr.SpanOutUcastClassBCirOctets,
        rpr.SpanOutMcastClassBCirFrames,
        rpr.SpanOutMcastClassBCirOctets,
        rpr.SpanOutUcastClassAFrames,
        rpr.SpanOutUcastClassAOctets,
        rpr.SpanOutMcastClassAFrames,
        rpr.SpanOutMcastClassAOctets,
        rpr.SpanOutCtrlFrames,
        rpr.SpanOutOamEchoFrames,
        rpr.SpanOutOamFlushFrames,
        rpr.SpanOutOamOrgFrames,
        rpr.SpanOutTopoAtdFrames,
        rpr.SpanOutTopoChkSumFrames,
        rpr.SpanOutTopoTpFrames,
        rpr.IS24H,
        rpr.ModelType,
        rpr.ObjectType,
        rpr.StrObjInstance,
        rpr.InvalidityList,
        rpr.rowid as myrow,
        ni.NEAliasID as NEAliasID,
    rpr.TimeLastCleared,
     rpr.StrCorbaTidAid
FROM
       ons15454_rpr_pm_table rpr,
       ne_info_table ni,
       module_type_table en
WHERE
       rpr.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       rpr.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.58 ONS15454_Security_View

```
CREATE or REPLACE view ONS15454_SECURITY_VIEW as
SELECT
```

```
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col5         as Name,
      eit.col6         as Password,
      eit.col7         as SecurityLevel,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4051;
```

2.6.59 ONS15454_SNMP_Trap_Dest_View

```
CREATE or REPLACE view ONS15454_SNMP_TRAP_DEST_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as IPAddress,
      eit.col5         as CommunityName,
      eit.col6         as UPDPort,
      eit.col7         as TrapVersion,
      eit.col8         as MaxTrapsPerSecond,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid        as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4005;
```

2.6.60 ONS15454_SONET_Line_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_line_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance InMaintenance,
       csonet.NEDBAccessID as NEDBAccessId,
       en.ModuleType,
       csonet.IS24H,
       csonet.thresholdList as ThresholdList,
       csonet.FeCodeViolations as FeCodeViolations,
       csonet.FeErrSecs as FeErrSecs,
       csonet.FeSevErrSecs as FeSevErrSecs,
       csonet.FeUnavailableSecs as FeUnavailableSecs,
       csonet.FeFcL as FeFcL,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
```

```
       ni.NEAliasID as NEAliasID,
    csonet.StrCorbaTidAid
FROM
       ONS15454_sonet_line_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.61 ONS15454_SONET_Line_PM_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_line_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance InMaintenance,
       csonet.CodeViolations as CodeViolations,
       csonet.ErrSecs as ErrSecs,
       csonet.SevErrSecs as SevErrSecs,
       csonet.UnavailableSecs as UnavailSecs,
       csonet.PpjcPdet as PpjcPdet,
       csonet.NpjcPdet as NpjcPdet,
       csonet.PpjcPgen as PpjcPgen,
       csonet.NpjcPgen as NpjcPgen,
       csonet.FcL as Fcl,
       csonet.Psc as Psc,
       csonet.Psd as Psd,
       csonet.PjDiff as PjDiff,
       csonet.PjpSec as PjpSec,
       csonet.PjnSec as PjnSec,
       csonet.Pscw   as Pscw,
       csonet.Psdw   as Psdw,
       csonet.Pscr   as Pscr,
       csonet.Psdr   as Psdr,
       csonet.Pscs   as Pscs,
       csonet.Psds   as Psds,
       csonet.NEDBAccessID as NEDBAccessId,
       csonet.thresholdList as ThresholdList,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID
    csonet.StrCorbaTidAid

FROM
       ONS15454_sonet_line_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.62 ONS15454_SONET_Path_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_path_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
```

```
        csonet.TimeStamp as TimeStamp,
        csonet.NEIfPmStatus as NEIfPmStatus,
        csonet.InMaintenance as InMaintenance,
        csonet.NEDBAccessID as NEDBAccessID,
        csonet.thresholdList as ThresholdList,
        csonet.FeSonetPathCodeViolations as FePathCodeViolations,
        csonet.FeSonetPathErrSecs as FePathErrSecs,
        csonet.FeSonetPathSevErrSecs as FePathSevErrSecs,
        csonet.FeSonetPathUnavailableSecs as FePathUnavailableSecs,
        csonet.FeSonetPathFC as FePathFC,
        en.ModuleType,
        csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
        csonet.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csonet.StrCorbaTidAid
FROM
        ONS15454_sonet_path_pm_table csonet,
        ne_info_table ni,
        module_type_table en
WHERE
        csonet.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.63 ONS15454_SONET_Path_PM_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_path_pm_view AS
SELECT
        ni.NESysID as NESysId,
        en.ModuleName as ModuleName,
        csonet.PhysicalLoc,
        csonet.NEIfIndex as NEIfIndex,
        csonet.TimeStamp as TimeStamp,
        csonet.NEIfPmStatus as NEIfPmStatus,
        csonet.InMaintenance as InMaintenance,
        csonet.SonetPathCodeViolations as CodeViolations,
        csonet.SonetPathErrSecs as ErrSecs,
        csonet.SonetPathSevErrSecs SevErrSecs,
        csonet.SonetPathUnavailableSecs as LineUnavailSecs,
        csonet.SonetPathFC  as SonetPathFC,
        csonet.PpjcPdet as PpjcPdet,
        csonet.NpjcPdet as NpjcPdet,
        csonet.PpjcPgen as PpjcPgen,
        csonet.NpjcPgen as NpjcPgen,
        csonet.PjDiff as PjDiff,
        csonet.PjpSec as PjpSec,
        csonet.PjnSec as PjnSec,
        csonet.StsPathWidth as StsPathWidth,
        csonet.NEDBAccessID as NEDBAccessID,
        csonet.thresholdList as ThresholdList,
        en.ModuleType,
        csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
        csonet.rowid as myrow,
        ni.NEAliasID as NEAliasID,
          csonet.StrCorbaTidAid
FROM
        ONS15454_sonet_path_pm_table csonet,
        ne_info_table ni,
        module_type_table en
WHERE
        csonet.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.64 ONS15454_SONET_Section_PM_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_section_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.SevErrFrameSecs as ErrFrameSecs,
       csonet.CodeViolations  as CodeViolations,
       csonet.ErrSecs         as ErrSecs,
       csonet.SevErrSecs      as SevErrSecs,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID
        csonet.StrCorbaTidAid

FROM
       ONS15454_sonet_sec_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.6.65 ONS15454_SONET_VT2_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_vt2_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       csonet.FeSonetVTCodeViolations as FeVT2CodeViolations,
       csonet.FeSonetVTErrSecs as FeVT2ErrSecs,
       csonet.FeSonetVTSevErrSecs as FeVT2SevErrSecs,
       csonet.FeSonetLineUnavailableSecs as FeVt2LineUnavailableSecs,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID
         csonet.StrCorbaTidAid
FROM
       ONS15454_sonet_vt_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID AND
       csonet.VTType = 1;
```

2.6.66 ONS15454_SONET_VT2_PM_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_vt2_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.SonetVTCodeViolations as Vt2CodeViolations,
       csonet.SonetVTErrSecs as Vt2ErrSecs,
       csonet.SonetVTSevErrSecs as Vt2LineSevErrSecs,
       csonet.SonetLineUnavailableSecs as Vt2LineUnavailSecs,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID
    csonet.StrCorbaTidAid
FROM
       ONS15454_sonet_vt_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID AND
       csonet.VTType = 1 ;
```

2.6.67 ONS15454_SONET_VT_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_vt_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       csonet.FeSonetVTCodeViolations as FeVT15CodeViolations,
       csonet.FeSonetVTErrSecs as FeVT15ErrSecs,
       csonet.FeSonetVTSevErrSecs as FeVT15SevErrSecs,
       csonet.FeSonetLineUnavailableSecs as FeVt15LineUnavailableSecs,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID
    csonet.StrCorbaTidAid
FROM
       ONS15454_sonet_vt_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID AND
       csonet.VTType = 0;
```

2.6.68 ONS15454_SONET_VT_PM_View

```
CREATE OR REPLACE VIEW ONS15454_sonet_vt_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csonet.PhysicalLoc,
       csonet.NEIfIndex as NEIfIndex,
       csonet.TimeStamp as TimeStamp,
       csonet.NEIfPmStatus as NEIfPmStatus,
       csonet.InMaintenance as InMaintenance,
       csonet.SonetVTCodeViolations as Vt15CodeViolations,
       csonet.SonetVTErrSecs as Vt15ErrSecs,
       csonet.SonetVTSevErrSecs as Vt15LineSevErrSecs,
       csonet.SonetLineUnavailableSecs as Vt15LineUnavailSecs,
       csonet.NEDBAccessID as NEDBAccessID,
       csonet.thresholdList as ThresholdList,
       en.ModuleType,
       csonet.IS24H,
       csonet.ModelType,
       csonet.ObjectType,
       csonet.StrObjInstance,
       csonet.InvalidityList,
       csonet.rowid as myrow,
       ni.NEAliasID as NEAliasID,
         csonet.StrCorbaTidAid
FROM
       ONS15454_sonet_vt_pm_table csonet,
       ne_info_table ni,
       module_type_table en
WHERE
       csonet.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csonet.NEDBAccessID  = ni.NEDBAccessID AND
       csonet.VTType = 0 ;
```

2.6.69 ONS15454_Spanning_Trees_View

```
CREATE or REPLACE view ONS15454_SPANING_TREES_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col5         as RowNo,
      eit.col6         as Priority,
      eit.col7         as BridgeMaxAge,
      eit.col8         as BridgeHelloTime,
      eit.col8         as BridgeForwardDelay,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4010;
```

2.6.70 ONS15454_Static_Route_View

```
create or replace view ons15454_static_route_view as
select
      ne.NESysid,
      mtt.modulename  as ModuleName,
      eit.Physicalloc as Physicalloc,
```

```
      eit.IfIndex     as IfIndex,
      eit.col2        as Destination,
      eit.col3        as Mask,
      eit.col4        as NextHop,
      eit.col5        as Cost,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType  as ModuleType,
      ne.rowid        as myrow
From  ne_info_table     ne,
      eqpt_info_table   eit,
      module_type_table mtt
Where
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4001;
```

2.6.71 ONS15454_Threshold_View

```
CREATE or REPLACE view ONS15454_THRESHOLD_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col5         as SlotPort,
      eit.col6         as Variable,
      eit.col7         as AlarmType,
      eit.col8         as SampleType,
      eit.col9         as SamplePeriod,
      eit.col10        as RisingThreshold,
      eit.col11        as FallingThreshold,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid         as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4050;
```

2.6.72 ONS15454_User_View

```
create or replace view ONS15454_User_view as
select
        a.nesysid as NEId,
        b.userid as UserId,
        b.nedbaccessid,
        c.privilege as Privilege,
        b.lockout as LockOut,
        b.lastlogintime as LastLoginTime,
        b.failedlogincount as FailedLoginCount,
    b.disabled as Disabled,
    b.ChangePwdNxtLogin as ChangePasswordOnNextLogin,
        b.rowid as myrow
from
        ons15454_user_table  b,
        ne_info_table a,
        ons15454_privilege_table c
where
        b.nedbaccessid = a.nedbaccessid AND
        b.privilegeid = c.privilegeid;
```

2.6.73 ONS15454_Virtual_Link_View

```
CREATE or REPLACE view ONS15454_VIRTUAL_LINK_VIEW as
SELECT
      ne.NESysid,
      mtt.modulename   as ModuleName,
      eit.Physicalloc  as Physicalloc,
      eit.IfIndex      as IfIndex,
      eit.col4         as Neighbor,
      eit.col5         as TransitDelay,
      eit.col6         as RetransmitInt,
      eit.col7         as HelloInt,
      eit.col8         as DeadInt,
      eit.col9         as AuthType,
      eit.col10        as AuthKey,
      eit.NEDbaccessid as NEDbaccessid,
      eit.ModuleType   as ModuleType,
      ne.rowid        as myrow
 FROM ne_info_table      ne,
      eqpt_info_table    eit,
      module_type_table  mtt
WHERE
      ne.NEDbaccessid  = eit.NEDbaccessid  AND
      ne.NEModelType   = mtt.ModelType     AND
      eit.ModuleType   = mtt.ModuleType    AND
      mtt.Modeltype    = 3                 AND
      mtt.ModuleType   = 0                 AND
      eit.ObjectIndex  = 4004;
```

2.6.74 PM_Param_ColNo_View

```
CREATE OR REPLACE VIEW PM_PARAM_COLNO_VIEW AS
SELECT  DISTINCT
        attr.PMVIEWNAME,
        attr.PMNAME,
        attr.PMPARAMID,
        cer.PMTABLECOLNO
FROM
        PM_ATTRID_MAPPING_TABLE attr,
    PM_CERENT_INFO_TABLE cer
WHERE
        attr.PMPARAMID=cer.PMPARAMID;
```

2.6.75 PM_Metadata_Attrid_View

```
CREATE OR REPLACE VIEW PM_METADATA_ATTRID_VIEW AS
SELECT  DISTINCT
        attr.PMVIEWNAME,
        attr.METADATAINDEX,
        attr.ATTRINDEX,
        met.NEMODELINDEX,
        met.MODULETYPE,
        met.PMPARAMID
FROM
        PM_ATTRID_MAPPING_TABLE attr,
        PM_METADATA_INFO_VIEW  met
WHERE
        attr.PMPARAMID=met.PMPARAMID;
```

2.6.76 PM_Metadata_Iinfo_View

```
CREATE OR REPLACE VIEW PM_METADATA_INFO_VIEW AS
SELECT
        pmdata.NEMODEL,
        pmdata.NEMODELINDEX,
        pmdata.MODULETYPE,
        pmdata.INTERFACETYPE,
        pmdata.PMCATEGORY,
```

```
        pmdata.OBJECTTYPEID,
        pmdata.PMTABLECOLNO,
        pmdata.PMPARAMID,
        pmdata.PARAMPOSITION,
        pmdata.VALIDITYPOSITION,
        pmdata.TCANUMBER,
        pmdata.TCALOCATION,
        pmdata.DIRECTION,
        pmdata.rowid as MYROW
        FROM
        PM_METADATA_TABLE pmdata
UNION ALL
SELECT
        pminfo.NEMODEL,
        pminfo.NEMODELINDEX,
        pminfo.MODULETYPE,
        pminfo.INTERFACETYPE,
        pmdata.PMCATEGORY,
        pmdata.OBJECTTYPEID,
        pmdata.PMTABLECOLNO,
        pmdata.PMPARAMID,
        pmdata.PARAMPOSITION,
        pmdata.VALIDITYPOSITION,
        pmdata.TCANUMBER,
        pmdata.TCALOCATION,
        pmdata.DIRECTION,
        pmdata.rowid as MYROW
        FROM
        PM_METADATA_INFO_TABLE pminfo,
        PM_METADATA_TABLE pmdata
        WHERE
        pminfo.NEMODEL = pmdata.NEMODEL and
        pminfo.BASENEMODELINDEX = pmdata.NEMODELINDEX and
        pminfo.BASEMODULETYPE = pmdata.MODULETYPE and
        pminfo.INTERFACETYPE = pmdata.INTERFACETYPE
        ORDER BY 1,2,3,4,9;
```

2.6.77 PM_Metadata_View

```
CREATE OR REPLACE VIEW PM_METADATA_VIEW AS
SELECT
    pmm.NEMODEL,
    pmm.NEMODELINDEX,
    pmm.MODULETYPE,
    pmm.PMCATEGORY,
    pmm.OBJECTTYPEID,
    pmm.PMTABLECOLNO,
    pmm.PMPARAMID,
    pmm.INTERFACETYPE,
    pmm.PARAMPOSITION,
    pmm.VALIDITYPOSITION,
    pmm.TCANUMBER,
    pmm.TCALOCATION,
    pmm.DIRECTION,
    pmm.rowid as myrow
FROM
        PM_METADATA_TABLE pmm;
```

2.7 Cisco ONS 155xx Views

2.7.1 ONS155xx_CDL_Flow_Config_View

```
create or replace view ONS155xx_CDL_FLOW_CONFIG_VIEW AS
select
    cdl.NEDBACCESSID,
    neInfo.NESysID as NESYSID,
    ifTable.PHYSICALLOC,
    ifTable.MODULETYPE,
    ifTable.NEIFINDEX as IFINDEX,
```

```
    cdl.CDLXMITFLOWID,
    cdl.CDLRCVFLOWID,
    ifTable.MODELTYPE,
    ifTable.OBJECTTYPE,
    neInfo1.NEAliasID
from
    ONS155xx_CDL_FLOW_CONFIG cdl,
    NE_INFO_TABLE neInfo,
    MIB2_INTERFACE_TABLE ifTable
where
    cdl.ifdbaccessid = ifTable.ifdbaccessid and
    cdl.NEDBACCESSID = neInfo.NEDBACCESSID;
```

2.7.2 ONS155xx_CDL_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_CDL_pm_view AS
SELECT
       ni.NESysID,
       en.ModuleName,
       pm.PhysicalLoc,
       pm.NEIfIndex,
       pm.Interval,
       pm.TimeStamp,
       pm.NEIfPmStatus,
       pm.inMaintenance,
       pm.NEDBAccessID,
       pm.ThresholdMask,
       pm.RxHEC,
       pm.RxNonCDL,
       pm.RxInvalidFlowID,
       pm.NetEthernetCRC,
       en.ModuleType,
       pm.rowid as myrow,
       pm.ModelType,
       pm.ObjectType,
       ni.NEAliasID
FROM
       ONS155xx_cdl_pm_table pm,
       ne_info_table ni,
       module_type_table en
WHERE
       pm.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       pm.NEDBAccessID  = ni.NEDBAccessID AND
       pm.NEIfPMStatus != 0;
```

2.7.3 ONS155xx_Circuit_CC_View

```
create or replace view ONS155xx_Circuit_CC_View as
select
        circuit.CktName as Name,
        circuit.CktType as Type,
        circuit.CktSize as CktSize,
        neInfo1.NESysID as SrcNodeId,
        ccSrcDestModuleType.srcmoduleName as SrcModuleType,
        cccircuit.CktCCSrcPhysicalLoc as SrcPhysicalLoc,
        cccircuit.CktCCSrcIfIndex as SrcInterfaceId,
        neInfo1.NESysID as DestNodeId,
        ccSrcDestModuleType.destmoduleName DestModuleType,
        cccircuit.CktCCDestPhysicalLoc as DestPhysicalLoc,
        cccircuit.CktCCDestIfIndex as DestInterfaceId,
       cccircuit.CktCCSrcWavelength as SrcWavelength,
       cccircuit.CktCCDestWavelength as DestWavelength,
        cccircuit.CktCCSrcState as SrcState,
        cccircuit.CktCCDestState as DestState,
       cccircuit.IsCktCCWorking as working,
        cccircuit.IsCktCCActive as active,
        cccircuit.CktNodeId as NodeId,
        cccircuit.CktUniqueId as UniqueId,
       ccLink.CktCCLinkId as LinkId,
        cccircuit.rowid as myrow,
```

```
        cccircuit.CktCCSrcModelType as SrcModelType,
        cccircuit.CktCCSrcObjectType as SrcObjectType,
        cccircuit.CktCCDestModelType as DestModelType,
        cccircuit.CktCCDestObjectType as DestObjectType,
        circuit.CktAliasName
from
        CIRCUIT_TBL circuit,
        NE_INFO_TABLE neInfo1,
        ONS155xx_circuit_cc_tbl cccircuit,
       ONS155xx_CKT_CC_LINK_View ccLink,
        ONS155xx_CKT_CC_SRC_DEST_VIEW ccSrcDestModuleType
where
        cccircuit.CktNodeId = circuit.CktNodeId and
        cccircuit.CktUniqueId = circuit.CktUniqueId and
        cccircuit.CktCCNodeID = neInfo1.NEDBAccessID and
        cccircuit.CktCCDBID = ccSrcDestModuleType.cktccdbid and
       cccircuit.CKTCCDBID = ccLink.CktCCDBId(+) ;
```

2.7.4 ONS155xx_Circuit_Dest_View

```
create or replace view ONS155xx_Circuit_Dest_View AS
select
        circuit.CktName as Name,
        circuit.CktDescription as Description,
        circuit.CktType as Type,
        circuit.CktSize as CktSize,
        circuit.CktDirection as Direction,
        circuit.CktState as State,
        neinfo1.NESYSID as SrcNodeId,
        circuit.CktSrcModuleType as SrcModuleType,
        circuit.CktSrcPhysicalLoc as SrcPhysicalLoc,
        circuit.CktSrcIfIndex as SrcInterfaceId,
        circuit.CktSecSrcModuleType as SecSrcModuleType,
        circuit.CktSecSrcPhysicalLoc as SecSrcPhysicalLoc,
        circuit.CktSecSrcIfIndex as SecSrcInterfaceId,
        neinfo2.NESYSID as DestNodeId,
        circuit.CktDestModuleType as DestModuleType,
        circuit.CktDestPhysicalLoc as DestPhysicalLoc,
        circuit.CktDestIfIndex as DestInterfaceId,
        circuit.CktSrcDropProtType as SrcDropProtType,
        circuit.CktDestDropProtType as DestDropProtType,
        circuit.wavelength as AdditionalInfo,
        circuit.CktNodeId as NodeId,
        circuit.CktUniqueId as UniqueId,
        circuit.CircuitNote as CircuitNote,
       circuit.myrow as myrow,
        circuit.CktSrcModelType as SrcModelType,
        circuit.CktSrcObjectType as SrcObjectType,
        circuit.CktSecSrcModelType as SecSrcModelType,
        circuit.CktSecSrcObjectType as SecSrcObjectType,
        circuit.CktDestModelType as DestModelType,
        circuit.CktDestObjectType as DestObjectType,
        neinfo1.NEAliasID,
        circuit.cktaliasname as CktAliasName
from
        ONS155xx_CIRCUIT_SRCDEST_VIEW circuit,
       ne_info_table neinfo1,
       ne_info_table neinfo2
where
       circuit.CktSrcNodeID       = neinfo1.NEDBACCESSID   and
       circuit.CktDestNodeID      = neinfo2.NEDBACCESSID;
```

2.7.5 ONS155xx_Circuit_Final_View

```
create or replace view ONS155xx_Circuit_Final_View AS
select  DISTINCT
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
```

```
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
        circuit.NodeId,
        circuit.UniqueId,
       circuit.SpanSrcNodeId,
       circuit.SpanDestNodeId,
       cktLink.LinkId  as SpanLinkId,
       cdl.CDLXMITFLOWID as CdlTxFlowId,
       cdl.CDLRCVFLOWID as CdlRxFlowId,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.SecSrcModelType,
        circuit.SecSrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType,
       circuit.CktAliasName
from
        ONS155xx_CIRCUIT_SPAN_VIEW circuit,
       ONS155xx_CKT_LINK_View cktLink,
       ONS155xx_CDL_FLOW_CONFIG_VIEW cdl
where
       circuit.NodeId            = cktLink.CktNodeId(+) and
       circuit.UniqueId          = cktLink.CktUniqueId(+) and
       circuit.SrcNodeId         = cdl.NESYSID (+) and
       circuit.SrcModuleType     = cdl.MODULETYPE (+) and
       circuit.SrcPhysicalLoc    = cdl.PHYSICALLOC (+) and
       circuit.SrcModelType      = cdl.MODELTYPE (+) and
       circuit.SrcObjectType     = cdl.OBJECTTYPE (+) and
       circuit.SrcInterfaceId    = cdl.IFINDEX (+) ;
```

2.7.6 ONS155xx_Circuit_Path_View

```
create or replace view ONS155xx_CIRCUIT_PATH_VIEW AS
select DISTINCT
        circuitpath.Name,
        circuitpath.Type,
        circuitpath.CktSize,
        circuitpath.PathNumber,
        circuitpath.SrcNodeId,
        circuitpath.SrcModuleType,
        circuitpath.SrcPhysicalLoc,
        circuitpath.SrcInterfaceId,
        circuitpath.DestNodeId,
        circuitpath.DestModuleType,
        circuitpath.DestPhysicalLoc,
        circuitpath.DestInterfaceId,
        circuitpath.PathStatus,
        circuitpath.IsWorking,
        circuitpath.IsActive,
        circuitpath.DownInterfaces,
        circuitpath.CktNodeId,
        circuitpath.CktUniqueId,
        circuitpath.PathDBID,
       circuitpath.DestOpticalPower,
       circuitpath.LinkId,
       circuitpath.PathDescr,
       circuitpath.PathDestPwrMonNodeID,
        circuitpath.PathDestPwrMonPhysicalLoc,
       circuitpath.PathDestPwrMonIfIndex,
       circuitpath.PathDestPwrMonModuleType,
```

```
        cktstack.LLCktName,
        cktstack.LLCktType,
        cktstack.LLCktSize,
       cktstack.LLCktPathDescr,
        cktstack.LLCKTNODEID,
        cktstack.LLCKTUNIQUEID,
       cktstack.LLCktPathDBID as LLPathDBID,
       circuitpath.PathDBID as myrow,
        circuitpath.SrcModelType,
        circuitpath.SrcObjectType,
        circuitpath.DestModelType,
        circuitpath.DestObjectType,
        circuitpath.PathDestPwrMonModelType,
        circuitpath.PathDestPwrMonObjectType
from
       ONS155xx_CKT_PATH_STACK_VIEW cktstack,
       ONS155xx_CKT_PATH_FINAL_VIEW circuitpath
where
       circuitpath.CktNodeId     =   cktstack.HLCktNodeId(+) and
       circuitpath.CktUniqueId       =   cktstack.HLCktUniqueId(+) and
       circuitpath.PathDBID      =   cktstack.HLCktPathDBID(+) ;
```

2.7.7 ONS155xx_Circuit_Segment_View

```
create or replace view ONS155xx_Circuit_Segment_View as
select
        spancircuit.CktName as Name,
        spancircuit.CktType as Type,
        spancircuit.CktSize as CktSize,
        spancircuit.CktSpanSrcNodeId as SrcNodeId,
        spanModuleType.srcmoduleName as SrcModuleType,
        spancircuit.CktSpanSrcPhysicalLoc as SrcPhysicalLoc,
        spancircuit.CktSpanSrcIfIndex as SrcInterfaceId,
        spancircuit.CktSpanDestNodeId as DestNodeId,
        spanModuleType.destmoduleName DestModuleType,
        spancircuit.CktSpanDestPhysicalLoc as DestPhysicalLoc,
        spancircuit.CktSpanDestIfIndex as DestInterfaceId,
        spancircuit.CktSpanAdditionalInfo as SrcWavelength,
        spancircuit.CktSpanAdditionalInfo as DestWavelength,
        spancircuit.CktSpanSrcState as SrcState,
        spancircuit.CktSpanDestState as DestState,
        spancircuit.IsCktSpanWorking as working,
        spancircuit.IsCktSpanActive as active,
        spancircuit.CktNodeId as NodeId,
        spancircuit.CktUniqueId as UniqueId,
        spancircuit.rowid as myrow,
        spancircuit.CktSpanSrcModelType as SrcModelType,
        spancircuit.CktSpanSrcObjectType as SrcObjectType,
        spancircuit.CktSpanDestModelType as DestModelType,
        spancircuit.CktSpanDestObjectType as DestObjectType
from
        ONS155xx_CKT_SPAN_JOIN_View spancircuit,
        ONS155xx_CKT_SPAN_MT_VIEW spanModuleType
where
        spancircuit.cktspanid = spanModuleType.cktspanid ;
```

2.7.8 ONS155xx_Circuit_Span_View

```
create or replace view ONS155xx_Circuit_Span_View AS
select
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
```

```
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
        circuit.NodeId,
        circuit.UniqueId,
       span.CktSpanSrcNodeId as SpanSrcNodeId,
       span.CktSpanDestNodeId as SpanDestNodeId,
       span.CktLinkId  as SpanLinkId,
        circuit.CircuitNote as CircuitNote,
       circuit.myrow as myrow,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.SecSrcModelType,
        circuit.SecSrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType,
       circuit.CktAliasName
from
        ONS155xx_CIRCUIT_DEST_VIEW circuit,
       CIRCUIT_SPAN_TBL span
where
       circuit.NodeId            = span.CKTNODEID(+) and
       circuit.UniqueId          = span.CKTUNIQUEID(+) ;
```

2.7.9 ONS155xx_Circuit_SrcDest_View

```
create or replace view ONS155xx_Circuit_SRCDEST_VIEW AS
select
        circuit.CktNodeId,
        circuit.CktUniqueId,
        circuit.CktName,
        circuit.CktDescription,
        circuit.CktType,
        circuit.CktSize,
        circuit.CktDirection,
        circuit.CktState,
        circuit.CktSrcNodeId,
        circuit.CktSrcModuleType,
        circuit.CktSrcPhysicalLoc,
        circuit.CktSrcIfIndex,
        circuit.CktSecSrcModuleType,
        circuit.CktSecSrcPhysicalLoc,
        circuit.CktSecSrcIfIndex,
        destcircuit.CktDestNodeId,
        destcircuit.CktDestModuleType,
        destcircuit.CktDestPhysicalLoc,
        destcircuit.CktDestIfIndex,
        circuit.CktSrcDropProt,
        circuit.CktSrcDropProtType,
        destcircuit.CktDestDropProt,
        destcircuit.CktDestDropProtType,
        destcircuit.CktDestPathProt,
        circuit.CktAdditionalInfo as Wavelength,
        circuit.cktComment as CircuitNote,
    circuit.rowid as myrow,
        circuit.CktSrcModelType,
        circuit.CktSrcObjectType,
        circuit.CktSecSrcModelType,
        circuit.CktSecSrcObjectType,
        destcircuit.CktDestModelType,
        destcircuit.CktDestObjectType,
       circuit.CktAliasName
from
        circuit_tbl circuit,
        circuit_dest_tbl destcircuit
where
        circuit.CktNodeId             = destcircuit.CktNodeId           and
```

```
        circuit.CktUniqueId           = destcircuit.CktUniqueId  ;
```

2.7.10 ONS155xx_Circuit_Stack_View

```
create or replace view ONS155xx_Circuit_Stack_View AS
select DISTINCT
        hlcircuit.CktName as HLCktName,
        hlcircuit.CktDescription as HLCktDescription,
        hlcircuit.CktType as HLCktType,
        hlcircuit.CktSize as HLCktSize,
        llcircuit.CktName as LLCktName,
        llcircuit.CktDescription as LLCktDescription,
        llcircuit.CktType as LLCktType,
        llcircuit.CktSize as LLCktSize,
        cktstack.LLToHLCktAssocType as AssocType,
        cktstack.HLCKTNODEID,
        cktstack.HLCKTUNIQUEID,
        cktstack.LLCKTNODEID,
        cktstack.LLCKTUNIQUEID
from
        CIRCUIT_TBl llcircuit,
        CIRCUIT_TBL hlcircuit,
        ONS155xx_CIRCUIT_STACK_TBL cktstack
where
        cktstack.LLCKTNODEID        = llcircuit.CktNodeId and
        cktstack.LLCKTUNIQUEID      = llcircuit.CktUniqueId and
        cktstack.HLCKTNODEID        = hlcircuit.CktNodeId and
        cktstack.HLCKTUNIQUEID      = hlcircuit.CktUniqueId ;
```

2.7.11 ONS155xx_Circuit_View

```
create or replace view ONS155xx_Circuit_View AS
select DISTINCT
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
        circuit.NodeId,
        circuit.UniqueId,
       circuit.SpanSrcNodeId,
       circuit.SpanDestNodeId,
       circuit.SpanLinkId,
       cktstack.LLCktName,
        cktstack.LLCktDescription,
        cktstack.LLCktType,
        cktstack.LLCktSize,
       cktstack.AssocType,
       cktstack.LLCktNodeId as LLNodeId,
       cktstack.LLCktUniqueId as LLUniqueId,
       circuit.CdlTxFlowId,
       circuit.CdlRxFlowId,
       circuit.UniqueId as myrow,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.SecSrcModelType,
```

```
        circuit.SecSrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType,
        circuit.CktAliasName
from
        ONS155xx_CIRCUIT_FINAL_VIEW circuit,
       ONS155xx_CIRCUIT_STACK_VIEW cktstack
where
        circuit.NodeId               = cktstack.HLCKTNODEID(+) and
        circuit.UniqueId          = cktstack.HLCKTUNIQUEID(+) ;
```

2.7.12 ONS155xx_CKT_CC_Dest_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_ckt_cc_dest_mt_view AS
SELECT
       cct.cktccdbid as cktccdbid,
       nvl(ccsrc.modulename,'System') as Modulename
FROM
      (select
              mtt.modulename   as modulename,
              cctbl.cktccdbid as cktccdbid
         from module_type_table mtt,
              ONS155xx_circuit_cc_tbl cctbl,
              ne_info_table ne
        where mtt.moduletype=cctbl.cktccdestmoduletype
          and ne.nemodeltype = mtt.modeltype
          and ne.nedbaccessid = cctbl.CKTCCNODEID
      ) ccsrc,
      ONS155xx_circuit_cc_tbl cct
  WHERE cct.cktccdbid=ccsrc.cktccdbid(+) ;
```

2.7.13 ONS155xx_CKT_CC_Detailed_View

```
create or replace view ONS155xx_CKT_CC_DETAILED_View as
select
        ccview.Name,
        ccview.Type,
        ccview.CktSize,
        ccview.PathNumber,
        ccview.PathHopNumber,
        gccToCktCC.GranularCCSeqNum as SeqNumber,
        gcc.GccDescr as SeqDescr,
       ccview.SrcNodeId,
        ccview.SrcModuleType,
        ccview.SrcPhysicalLoc,
        ccview.SrcInterfaceId,
        ccview.DestNodeId,
        ccview.DestModuleType,
        ccview.DestPhysicalLoc,
        ccview.DestInterfaceId,
        ccview.NodeId,
        ccview.UniqueId,
        ccview.PathDBID,
        ccview.LinkId,
        gcc.rowid as myrow,
        ccview.SrcModelType,
        ccview.SrcObjectType,
        ccview.DestModelType,
        ccview.DestObjectType
from
        ONS155xx_CKT_CC_PATH_View ccview,
        ONS155xx_GRA_CC_TO_CKT_CC_TBL gccToCktcc,
       ONS155xx_GRANULAR_CC_TBL gcc
where
        ccview.cktccdbid = gccToCktCC.CKTCROSSCONNECTDBID and
       gccToCktCC.GRANULARCCDBID = gcc.GCCDBID;
```

2.7.14 ONS155xx_CKT_CC_Link_View

```
create or replace view ONS155xx_CKT_CC_LINK_View as
```

```
select
        cccircuit.CktCCDBId,
        cccircuit.CktNodeId,
        cccircuit.CktUniqueId,
        gcc.GCCAssocLinkDBID as CktCCLinkId
from
        ONS155xx_GRA_CC_TO_CKT_CC_TBL gccToCktCC,
        ONS155xx_circuit_cc_tbl cccircuit,
        ONS155xx_GRANULAR_CC_TBL gcc
where
        cccircuit.CktCCDBId     =   gccToCktCC.CktCrossConnectDBId and
        gcc.GCCDBId             =   gccToCktCC.GranularCCDBId and
        gcc.GCCAssocLinkType    =   3;
```

2.7.15 ONS155xx_CKT_CC_Path_View

```
create or replace view ONS155xx_CKT_CC_Path_View as
select
        circuit.Name as Name,
        circuit.Type as Type,
        circuit.CktSize as CktSize,
        circuit.PATHNUMBER as PathNumber,
       pathhop.PathHopID as PathHopNumber,
       neInfo1.NESysID as SrcNodeId,
        cccircuit.CktCCSrcModuleType as SrcModuleType,
        cccircuit.CktCCSrcPhysicalLoc as SrcPhysicalLoc,
        cccircuit.CktCCSrcIfIndex as SrcInterfaceId,
        neInfo1.NESysID as DestNodeId,
        cccircuit.CktCCDestModuleType as DestModuleType,
        cccircuit.CktCCDestPhysicalLoc as DestPhysicalLoc,
        cccircuit.CktCCDestIfIndex as DestInterfaceId,
        cccircuit.CktCCSrcWavelength as SrcWavelength,
        cccircuit.CktCCDestWavelength as DestWavelength,
        cccircuit.CktCCSrcState as SrcState,
        cccircuit.CktCCDestState as DestState,
        cccircuit.IsCktCCWorking as working,
        cccircuit.IsCktCCActive as active,
        cccircuit.CktNodeId as NodeId,
        cccircuit.CktUniqueId as UniqueId,
       circuit.PathDBID as PathDBID,
       ccLink.CktCCLinkID as LinkID,
       cccircuit.CktCCDBID as CktCCDBID,
        cccircuit.rowid as myrow,
        cccircuit.CktCCSrcModelType as SrcModelType,
        cccircuit.CktCCSrcObjectType as SrcObjectType,
        cccircuit.CktCCDestModelType as DestModelType,
        cccircuit.CktCCDestObjectType as DestObjectType,
       circuit.CktAliasName
from
        ONS155xx_CKT_PATH_JOIN_VIEW circuit,
       ONS155xx_PATH_HOP_TBL pathhop,
        NE_INFO_TABLE neInfo1,
        ONS155xx_circuit_cc_tbl cccircuit,
       ONS155xx_CKT_CC_LINK_View ccLink
where
        cccircuit.CktNodeId = circuit.CktNodeId and
        cccircuit.CktUniqueId = circuit.CktUniqueId and
        cccircuit.CktCCNodeID = neInfo1.NEDBAccessID and
       circuit.PATHDBID = pathhop.PATHDBID and
       pathhop.PathHopType = 1 and
       pathhop.PathHopDBID = cccircuit.CKTCCDBID and
       cccircuit.CktCCDBID = ccLink.CktCCDBID(+) ;
```

2.7.16 ONS155XX_CKT_CC_Src_Dest_View

```
CREATE OR REPLACE VIEW ONS155XX_CKT_CC_SRC_DEST_VIEW AS
SELECT
       srcview.cktccdbid as cktccdbid,
       nvl(srcview.modulename,'System') as SrcModulename,
       nvl(destview.modulename, 'System') as DestModulename
FROM
```

```
      ONS155XX_CKT_CC_SRC_MT_VIEW srcview,
      ONS155XX_CKT_CC_DEST_MT_VIEW destview
WHERE srcview.cktccdbid=destview.cktccdbid ;
```

2.7.17 ONS155xx_CKT_CC_Src_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_ckt_cc_src_mt_view AS
SELECT
       cct.cktccdbid as cktccdbid,
       nvl(ccsrc.modulename,'System') as Modulename
FROM
       (select
             mtt.modulename   as modulename,
             cctbl.cktccdbid as cktccdbid
         from module_type_table mtt,
             ONS155xx_circuit_cc_tbl cctbl,
             ne_info_table ne
        where mtt.moduletype=cctbl.cktccsrcmoduletype
          and ne.nemodeltype = mtt.modeltype
          and ne.nedbaccessid = cctbl.CKTCCNODEID
      ) ccsrc,
      ONS155xx_circuit_cc_tbl cct
  WHERE cct.cktccdbid=ccsrc.cktccdbid(+) ;
```

2.7.18 ONS155xx_CKT_Endpoint_View

```
create or replace view ONS155xx_CKT_ENDPOINT_VIEW AS
select DISTINCT
    dest.CKTNODEID,
    dest.CKTUNIQUEID,
    dest.CKTDESTNODEID as NODEID,
    dest.CKTDESTPHYSICALLOC as PHYSICALLOC,
    dest.CKTDESTIFINDEX as IFINDEX,
    dest.CKTDESTMODULETYPE as MODULETYPE,
    dest.CktDESTDROPPROTTYPE as PROTTYPE,
    dest.CKTDESTMODELTYPE as MODELTYPE,
    dest.CKTDESTOBJECTTYPE as OBJECTTYPE
from
    CIRCUIT_DEST_TBL dest
union all
select DISTINCT
    ckt.CKTNODEID,
    ckt.CKTUNIQUEID,
    ckt.CKTSRCNODEID as NODEID,
        ckt.CKTSRCPHYSICALLOC as PHYSICALLOC,
    ckt.CKTSRCIFINDEX as IFINDEX,
    ckt.CKTSRCMODULETYPE as MODULETYPE,
    ckt.CktSRCDROPPROTTYPE as PROTTYPE,
        ckt.CKTSRCMODELTYPE as MODELTYPE,
        ckt.CKTSRCOBJECTTYPE as OBJECTTYPE
from
    CIRCUIT_TBL ckt
union all
select  DISTINCT
    ckt.CKTNODEID,
    ckt.CKTUNIQUEID,
    ckt.CKTSRCNODEID as NODEID,
        ckt.CKTSECSRCPHYSICALLOC as PHYSICALLOC,
    ckt.CKTSECSRCIFINDEX as IFINDEX,
    ckt.CKTSECSRCMODULETYPE as MODULETYPE,
    ckt.CktSRCDROPPROTTYPE as PROTTYPE,
        ckt.CKTSECSRCMODELTYPE as MODELTYPE,
        ckt.CKTSECSRCOBJECTTYPE as OBJECTTYPE
from
    CIRCUIT_TBL ckt
where
    ckt.CktSecSrcPhysicalLoc > 0;
```

2.7.19 ONS155XX_CKT_Link_Final_View

```
create or replace view ONS155XX_CKT_LINK_FINAL_VIEW AS
select  DISTINCT
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
       cdl.CDLXMITFLOWID as CdlTxFlowId,
       cdl.CDLRCVFLOWID as CdlRxFlowId,
        circuit.NodeId,
        circuit.UniqueId,
       cktLink.LinkId  as SpanLinkId,
           circuit.CircuitNote as CircuitNote,
       circuit.MyRow as myrow,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.SecSrcModelType,
        circuit.SecSrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType
from
        ONS155xx_CIRCUIT_DEST_VIEW circuit,
       ONS155xx_CDL_FLOW_CONFIG_VIEW cdl,
       ONS155xx_CKT_LINK_View cktLink
where
       circuit.NodeId            = cktLink.CktNodeId and
       circuit.UniqueId          = cktLink.CktUniqueId and
       circuit.SrcNodeId         = cdl.NESYSID (+) and
       circuit.SrcModuleType     = cdl.MODULETYPE (+) and
       circuit.SrcPhysicalLoc    = cdl.PHYSICALLOC (+) and
       circuit.SrcModelType      = cdl.MODELTYPE (+) and
       circuit.SrcObjectType     = cdl.OBJECTTYPE (+) and
       circuit.SrcInterfaceId    = cdl.IFINDEX (+) ;
```

2.7.20 ONS155xx_CKT_Link_View

```
create or replace view ONS155xx_CKT_LINK_View as
select DISTINCT
        ccLink.CktNodeId,
        ccLink.CktUniqueId,
        ccLink.CktCCLinkId as LinkId
from
        ONS155xx_CKT_CC_LINK_View ccLink
union all
select DISTINCT
        span.CktNodeId,
        span.CktUniqueId,
        span.CktLinkId as LinkId
from
        CIRCUIT_SPAN_TBL span   ;
```

2.7.21 ONS155XX_CKT_NE_Final_View

```
create or replace view ONS155XX_CKT_NE_FINAL_VIEW AS
select  DISTINCT
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
       cdl.CDLXMITFLOWID as CdlTxFlowId,
       cdl.CDLRCVFLOWID as CdlRxFlowId,
        circuit.NodeId,
        circuit.UniqueId,
       circuit.SpanSrcNodeId,
       circuit.SpanDestNodeId,
        circuit.CircuitNote as CircuitNote,
       circuit.myrow as myrow,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.SecSrcModelType,
        circuit.SecSrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType,
       circuit.CktAliasName
from
        ONS155xx_CIRCUIT_SPAN_VIEW circuit,
       ONS155xx_CDL_FLOW_CONFIG_VIEW cdl
where
       circuit.SrcNodeId          = cdl.NESYSID (+) and
       circuit.SrcModuleType      = cdl.MODULETYPE (+) and
       circuit.SrcPhysicalLoc     = cdl.PHYSICALLOC (+) and
       circuit.SrcModelType       = cdl.MODELTYPE (+) and
       circuit.SrcObjectType      = cdl.OBJECTTYPE (+) and
       circuit.SrcInterfaceId     = cdl.IFINDEX (+) ;
```

2.7.22 ONS155xx_CKT_OPT_Pwr_View

```
create or replace view ONS155xx_CKT_OPT_PWR_VIEW AS
select
       pwr.NEDBACCESSID,
       pwr.PwrSrcPhysicalLoc,
       pwr.PwrSrcIfIndex,
       pwr.PwrSrcModuleType,
       pwr.OpticalPower,
       pwr.PwrSrcModelType,
       pwr.PwrSrcObjectType
from
       ONS155XX_IF_OPTICAL_POWER pwr
where
       pwr.PwrMonDirection = 1 and
       pwr.PwrMonLocation = 3 ;
```

2.7.23 ONS155xx_CKT_Path_Final_View

```
create or replace view ONS155xx_CKT_PATH_FINAL_VIEW AS
```

```
select DISTINCT
        circuit.Name,
        circuit.Type,
        circuit.CktSize,
        circuit.PathNumber,
        neinfo1.NESYSID as SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        neinfo2.NESYSID as DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.PathStatus,
        circuit.IsWorking,
        circuit.IsActive,
        circuit.DownInterfaces,
        circuit.CktNodeId,
        circuit.CktUniqueId,
        circuit.PathDBID,
       pathDestPwr.PathDestOpticalPower as DestOpticalPower,
       pathLink.LinkId    as LinkId,
       circuit.PathDescr,
       pathDestPwr.PathDestPwrMonNodeID,
        pathDestPwr.PathDestPwrMonPhysicalLoc,
       pathDestPwr.PathDestPwrMonIfIndex,
       pathDestPwr.PathDestPwrMonModuleType,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.DestModelType,
        circuit.DestObjectType,
        pathDestPwr.PathDestPwrMonModelType,
        pathDestPwr.PathDestPwrMonObjectType,
        neInfo1.NEAliasID
from
       ONS155xx_CKT_PATH_JOIN_VIEW circuit,
       ONS155xx_PATH_DEST_PWR_VIEW pathDestPwr,
       ONS155xx_Path_Link_View pathLink,
        NE_INFO_TABLE neinfo1,
        NE_INFO_TABLE neinfo2
where
        circuit.SrcNodeID             = neinfo1.NEDBACCESSID   and
        circuit.DestNodeID            = neinfo2.NEDBACCESSID and
        circuit.PATHDBID           = pathDestPwr.PATHDBID   and
       circuit.PATHDBID            = pathLink.PATHDBID (+) ;
```

2.7.24 ONS155xx_CKT_Path_Join_View

```
create or replace view ONS155xx_CKT_PATH_JOIN_VIEW AS
select
        circuit.CktName as Name,
        circuit.CktType as Type,
        circuit.CktSize as CktSize,
        cktpath.PathSeqNum as PathNumber,
        path.PathSrcNodeID as SrcNodeId,
        path.PathSrcModuleType as SrcModuleType,
        path.PathSrcPhysicalLoc as SrcPhysicalLoc,
        path.PathSrcIfIndex as SrcInterfaceId,
        path.PathDestNodeID as DestNodeId,
        path.PathDestModuleType as DestModuleType,
        path.PathDestPhysicalLoc as DestPhysicalLoc,
        path.PathDestIfIndex as DestInterfaceId,
        path.PathStatus as PathStatus,
        path.IsWorking as IsWorking,
        path.IsActive as IsActive,
        path.DownInterfaces as DownInterfaces,
        circuit.CktNodeId as CktNodeId,
        circuit.CktUniqueId as CktUniqueId,
        path.PATHDBID as PathDBID,
       path.PathDescr,
        path.PathSrcModelType as SrcModelType,
        path.PathSrcObjectType as SrcObjectType,
        path.PathDestModelType as DestModelType,
```

```
        path.PathDestObjectType as DestObjectType
from
        CIRCUIT_TBL circuit,
        ONS155xx_CIRCUIT_PATH_TBL cktpath,
        ONS155xx_PATH_TBL path
where
        circuit.CktNodeId             = cktpath.CktNodeId           and
        circuit.CktUniqueId           = cktpath.CktUniqueId         and
        cktpath.PathDBID              = path.PATHDBID ;
```

2.7.25 ONS155xx_CKT_Path_Stack_View

```
create or replace view ONS155xx_Ckt_Path_Stack_View AS
select DISTINCT
        hlcircuit.Name as HLCktName,
        hlcircuit.Type as HLCktType,
        hlcircuit.CktSize as HLCktSize,
       hlcircuit.PathDescr as HLCktPathDescr,
        llcircuit.Name as LLCktName,
        llcircuit.Type as LLCktType,
        llcircuit.CktSize as LLCktSize,
       llcircuit.PathDescr as LLCktPathDescr,
        cktstack.LLToHLCktAssocType as AssocType,
        cktstack.HLCKTNODEID,
        cktstack.HLCKTUNIQUEID,
       cktstack.HLCktPathDBID,
        cktstack.LLCKTNODEID,
        cktstack.LLCKTUNIQUEID,
       cktstack.LLCktPathDBID
from
       ONS155xx_CKT_PATH_JOIN_VIEW llcircuit,
       ONS155xx_CKT_PATH_JOIN_VIEW hlcircuit,
        ONS155xx_CIRCUIT_STACK_TBL cktstack
where
        cktstack.LLCKTNODEID        = llcircuit.CktNodeId and
        cktstack.LLCKTUNIQUEID      = llcircuit.CktUniqueId and
       cktstack.LLCKTPATHDBID    = llcircuit.PATHDBID and
        cktstack.HLCKTNODEID        = hlcircuit.CktNodeId and
        cktstack.HLCKTUNIQUEID      = hlcircuit.CktUniqueId and
       cktstack.HLCKTPATHDBID    = hlcircuit.PATHDBID ;
```

2.7.26 ONS155xx_CKT_Segment_Path_View

```
create or replace view ONS155xx_CKT_Segment_Path_View as
select
        circuit.Name as Name,
        circuit.Type as Type,
        circuit.CktSize as CktSize,
        circuit.PathNumber as PathNumber,
       pathhop.PathHopId as PathHopNumber,
       neinfo1.NESysID as SrcNodeId,
        spancircuit.CktSpanSrcModuleType as SrcModuleType,
        spancircuit.CktSpanSrcPhysicalLoc as SrcPhysicalLoc,
        spancircuit.CktSpanSrcIfIndex as SrcInterfaceId,
        neinfo2.NESysID as DestNodeId,
        spancircuit.CktSpanDestModuleType as DestModuleType,
        spancircuit.CktSpanDestPhysicalLoc as DestPhysicalLoc,
        spancircuit.CktSpanDestIfIndex as DestInterfaceId,
        spancircuit.CktSpanAdditionalInfo as SrcWavelength,
        spancircuit.CktSpanAdditionalInfo as DestWavelength,
        spancircuit.CktSpanSrcState as SrcState,
        spancircuit.CktSpanDestState as DestState,
        spancircuit.IsCktSpanWorking as working,
        spancircuit.IsCktSpanActive as active,
        spancircuit.CktNodeId as NodeId,
        spancircuit.CktUniqueId as UniqueId,
       circuit.PathDBID as PathDBID,
       spancircuit.CktLinkId as LinkId,
       spancircuit.CktSpanID as CktSpanDBID,
        spancircuit.rowid as myrow,
        spancircuit.CktSpanSrcModelType as SrcModelType,
```

```
        spancircuit.CktSpanSrcObjectType as SrcObjectType,
        spancircuit.CktSpanDestModelType as DestModelType,
        spancircuit.CktSpanDestObjectType as DestObjectType,
       neInfo1.NEAliasID
from
        ONS155xx_CKT_PATH_JOIN_VIEW circuit,
        NE_INFO_TABLE neInfo1,
        NE_INFO_TABLE neInfo2,
        circuit_span_tbl spancircuit,
       ONS155xx_PATH_HOP_TBL pathhop
where
        spancircuit.CktNodeId = circuit.CktNodeId and
        spancircuit.CktUniqueId = circuit.CktUniqueId and
        spancircuit.CktSpanSrcNodeID = neInfo1.NEDBAccessID and
        spancircuit.CktSpanDestNodeId = neInfo2.NEDBAccessID and
       circuit.PathDBID = pathhop.PathDBID and
       pathhop.PathHopType = 2 and
       pathhop.PathHopDBID = spancircuit.CktSpanID ;
```

2.7.27 ONS155xx_CKT_SEG_Detailed_View

```
create or replace view ONS155xx_CKT_SEG_DETAILED_View as
select
        segmentview.Name,
        segmentview.Type,
        segmentview.CktSize,
       segmentview.PathNumber,
       segmentview.PathHopNumber,
       1 as SeqNumber,
       link.LinkName as SeqDescr,
        segmentview.SrcNodeId,
        segmentview.SrcModuleType,
        segmentview.SrcPhysicalLoc,
        segmentview.SrcInterfaceId,
        segmentview.DestNodeId,
        segmentview.DestModuleType,
        segmentview.DestPhysicalLoc,
        segmentview.DestInterfaceId,
        segmentview.NodeId,
        segmentview.UniqueId,
       segmentview.PathDBID,
       segmentview.LinkId,
        segmentview.myrow,
        segmentview.SrcModelType,
        segmentview.SrcObjectType,
        segmentview.DestModelType,
        segmentview.DestObjectType
from
        ONS155xx_CKT_Segment_PATH_View segmentview,
       LINK_TABLE link
where
        segmentview.LinkId = link.LinkId ;
```

2.7.28 ONS155xx_CKT_Span_Dest_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_ckt_span_dest_mt_view AS
SELECT DISTINCT
       cst.cktspandestnodeid as cktspandestnodeid,
       nvl(spandest.modulename,'System') as Modulename,
       cst.cktspandestphysicalloc,
       spandest.moduletype as ModuleType
FROM
      (select distinct
              mtt.modulename   as modulename,
              mtt.moduletype   as moduletype,
              cstbl.cktspandestphysicalloc as cktspandestphysicalloc,
              cstbl.cktspandestnodeid as cktspandestnodeid
         from module_type_table mtt,
              ons15454_ne_inventory_table cni,
              circuit_span_tbl cstbl,
              ne_info_table   ne
        where mtt.moduletype= cni.equipmenttype
```

```
          and
cni.physicalloc=round(cstbl.cktspandestphysicalloc/power(2,16))*(power(2,16))
          and ne.nenodeid = cstbl.cktspandestnodeid
          and cni.nedbaccessid=ne.nedbaccessid
          and ne.nemodeltype=mtt.modeltype
          and (mtt.modeltype=3 or mtt.modeltype=5 or mtt.modeltype=14 or mtt.modeltype=18)
) spandest,
      circuit_span_tbl cst
  WHERE cst.cktspandestnodeid = spandest.cktspandestnodeid(+)
    AND cst.cktspandestphysicalloc = spandest.cktspandestphysicalloc(+);
```

2.7.29 ONS155xx_CKT_Span_Det_View

```
create or replace view ONS155xx_CKT_SPAN_DET_VIEW as
select
        ccview.Name,
        ccview.Type,
        ccview.CktSize,
        ccview.PathNumber,
       ccview.PathHopNumber,
       ccview.SeqNumber,
       ccview.SeqDescr,
       ccview.SrcNodeId,
        ccview.SrcModuleType,
        ccview.SrcPhysicalLoc,
        ccview.SrcInterfaceId,
        ccview.DestNodeId,
        ccview.DestModuleType,
        ccview.DestPhysicalLoc,
        ccview.DestInterfaceId,
        ccview.NodeId,
        ccview.UniqueId,
       ccview.PathDBID,
       ccview.LinkId,
        ccview.myrow,
        ccview.SrcModelType,
        ccview.SrcObjectType,
        ccview.DestModelType,
        ccview.DestObjectType
from
        ONS155xx_CKT_CC_DETAILED_View ccview
union all
select
        segmentview.Name,
        segmentview.Type,
        segmentview.CktSize,
       segmentview.PathNumber,
       segmentview.PathHopNumber,
       segmentview.SeqNumber,
       segmentview.SeqDescr,
        segmentview.SrcNodeId,
        segmentview.SrcModuleType,
        segmentview.SrcPhysicalLoc,
        segmentview.SrcInterfaceId,
        segmentview.DestNodeId,
        segmentview.DestModuleType,
        segmentview.DestPhysicalLoc,
        segmentview.DestInterfaceId,
        segmentview.NodeId,
        segmentview.UniqueId,
       segmentview.PathDBID,
       segmentview.LinkId,
        segmentview.myrow,
        segmentview.SrcModelType,
        segmentview.SrcObjectType,
        segmentview.DestModelType,
        segmentview.DestObjectType
from
        ONS155xx_CKT_SEG_DETAILED_View segmentview ;
```

2.7.30 ONS155xx_CKT_Span_Join_View

```
create or replace view ONS155xx_CKT_SPAN_JOIN_View as
```

```
select
        circuit.CktName,
        circuit.CktType,
        circuit.CktSize,
        neInfo1.NESysID as CktSpanSrcNodeId,
        spancircuit.CktSpanSrcModuleType,
        spancircuit.CktSpanSrcPhysicalLoc,
        spancircuit.CktSpanSrcIfIndex,
        neInfo2.NESysID as CktSpanDestNodeId,
        spancircuit.CktSpanDestModuleType,
        spancircuit.CktSpanDestPhysicalLoc,
        spancircuit.CktSpanDestIfIndex,
        spancircuit.CktSpanAdditionalInfo,
        spancircuit.CktSpanSrcState,
        spancircuit.CktSpanDestState,
        spancircuit.IsCktSpanWorking,
        spancircuit.IsCktSpanActive,
        spancircuit.CktNodeId,
        spancircuit.CktUniqueId,
        spancircuit.CktSpanID,
        spancircuit.CktLinkId,
        spancircuit.rowid as myrow,
        spancircuit.CktSpanSrcModelType,
        spancircuit.CktSpanSrcObjectType,
        spancircuit.CktSpanDestModelType,
        spancircuit.CktSpanDestObjectType,
        neInfo1.NEAliasID
from
        CIRCUIT_TBL circuit,
        NE_INFO_TABLE neInfo1,
        NE_INFO_TABLE neInfo2,
        circuit_span_tbl spancircuit
where
        spancircuit.CktNodeId = circuit.CktNodeId and
        spancircuit.CktUniqueId = circuit.CktUniqueId and
        spancircuit.CktSpanSrcNodeID = neInfo1.NEDBAccessID and
        spancircuit.CktSpanDestNodeId = neInfo2.NEDBAccessID ;
```

2.7.31 ONS155xx_CKT_Span_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_CKT_SPAN_MT_VIEW AS
SELECT DISTINCT
      cktspan.cktnodeid    as cktnodeid,
      cktspan.cktuniqueid  as cktuniqueid,
      cktspan.cktspanid as cktspanid,
      cktspan.cktspansrcphysicalloc as srcPhysicalLoc,
      nvl(srcmt.modulename,'System') as SrcModulename,
      cktspan.cktspandestphysicalloc as DestPhysicalLoc,
      nvl(destmt.modulename, 'System') as DestModuleName,
      cktspan.cktspansrcmodeltype as srcModelType,
      cktspan.cktspansrcobjecttype as srcObjectType,
      cktspan.cktspandestmodeltype as DestModelType,
      cktspan.cktspandestobjecttype as DestObjectType
FROM
      (select
              ckttbl.cktnodeid as cktnodeid,
              ckttbl.cktuniqueid as cktuniqueid,
             ckttbl.cktspansrcobjecttype as objecttype,
              mtt.modulename   as modulename,
             ckttbl.cktspansrcphysicalloc as physicalloc,
             ckttbl.cktspansrcmodeltype as modeltype
         from ONS155xx_module_type_view mtt,
              circuit_span_tbl ckttbl
        where mtt.moduletype=ckttbl.cktspansrcmoduletype
      ) srcmt,
      (select
              ckttbl.cktnodeid as cktnodeid,
              ckttbl.cktuniqueid as cktuniqueid,
             ckttbl.cktspandestphysicalloc as physicalloc,
              mtt.modulename   as modulename,
             ckttbl.cktspandestmodeltype as modeltype,
             ckttbl.cktspandestobjecttype as objecttype
         from ONS155xx_module_type_view mtt,
              circuit_span_tbl ckttbl
```

```
        where mtt.moduletype=ckttbl.cktspandestmoduletype
      ) destmt,
      Circuit_Span_Tbl cktspan
WHERE cktspan.cktnodeid = srcmt.cktnodeid(+)
  AND cktspan.cktuniqueid = srcmt.cktuniqueid(+)
  AND cktspan.cktspansrcphysicalloc = srcmt.physicalloc(+)
  AND cktspan.cktnodeid = destmt.cktnodeid(+)
  AND cktspan.cktuniqueid = destmt.cktuniqueid (+)
  AND cktspan.cktspandestphysicalloc = destmt.physicalloc(+)
  AND cktspan.cktspandestmodeltype = destmt.modeltype(+)
  AND cktspan.cktspandestobjecttype = destmt.objecttype(+);
```

2.7.32 ONS155xx_CKT_Span_Src_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_ckt_span_src_mt_view AS
SELECT DISTINCT
       cst.cktspansrcnodeid as cktspansrcnodeid,
       nvl(spansrc.modulename,'System') as Modulename,
       cst.cktspansrcphysicalloc,
       cst.cktspansrcmodeltype,
       cst.cktspansrcobjecttype
FROM
      (select distinct
              mtt.modulename   as modulename,
              cstbl.cktspansrcnodeid as cktspansrcnodeid,
              cstbl.cktspansrcphysicalloc as cktspansrcphysicalloc,
              cstbl.cktspansrcmodeltype as cktspansrcmodeltype,
              cstbl.cktspansrcobjecttype as cktspansrcobjecttype
         from module_type_table mtt,
              circuit_span_tbl cstbl,
              ne_info_table ne
        where mtt.moduletype=cstbl.cktspansrcmoduletype
          and ne.nemodeltype = mtt.modeltype
           and ne.nedbaccessid=cstbl.cktspansrcnodeid
      ) spansrc,
      circuit_span_tbl cst
  WHERE cst.cktspansrcnodeid=spansrc.cktspansrcnodeid(+)
    AND cst.cktspansrcphysicalloc=spansrc.cktspansrcphysicalloc(+)
    AND cst.cktspansrcmodeltype=spansrc.cktspansrcmodeltype(+)
    AND cst.cktspansrcobjecttype=spansrc.cktspansrcobjecttype(+) ;
```

2.7.33 ONS155xx_CKT_Span_View

```
create or replace view ONS155xx_CKT_SPAN_VIEW as
select
        ccview.Name,
        ccview.Type,
        ccview.CktSize,
        ccview.PathNumber,
       ccview.PathHopNumber,
       ccview.SrcNodeId,
        ccview.SrcModuleType,
        ccview.SrcPhysicalLoc,
        ccview.SrcInterfaceId,
        ccview.DestNodeId,
        ccview.DestModuleType,
        ccview.DestPhysicalLoc,
        ccview.DestInterfaceId,
        ccview.NodeId,
        ccview.UniqueId,
       ccview.PathDBID,
       ccview.LinkId,
        ccview.myrow,
        ccview.SrcModelType,
        ccview.SrcObjectType,
        ccview.DestModelType,
        ccview.DestObjectType
from
        ONS155xx_CKT_CC_PATH_View ccview
union all
select
```

```
        segmentview.Name,
        segmentview.Type,
        segmentview.CktSize,
       segmentview.PathNumber,
       segmentview.PathHopNumber,
        segmentview.SrcNodeId,
        segmentview.SrcModuleType,
        segmentview.SrcPhysicalLoc,
        segmentview.SrcInterfaceId,
        segmentview.DestNodeId,
        segmentview.DestModuleType,
        segmentview.DestPhysicalLoc,
        segmentview.DestInterfaceId,
        segmentview.NodeId,
        segmentview.UniqueId,
       segmentview.PathDBID,
       segmentview.LinkId,
        segmentview.myrow,
        segmentview.SrcModelType,
        segmentview.SrcObjectType,
        segmentview.DestModelType,
        segmentview.DestObjectType
from
        ONS155xx_CKT_Segment_PATH_View segmentview ;
```

2.7.34 ONS155xx_CKT_Srcdest_MT_View

```
CREATE OR REPLACE VIEW ONS155xx_CKT_SRCDEST_MT_VIEW AS
SELECT
      cktsrcdest.cktnodeid    as cktnodeid,
      cktsrcdest.cktuniqueid  as cktuniqueid,
      cktsrcdest.cktsrcphysicalloc as srcPhysicalLoc,
      nvl(srcmt.modulename,'System') as SrcModulename,
      cktsrcdest.cktsecsrcphysicalloc as secsrcphysicalloc,
      nvl(secsrcmt.modulename, 'System') as SecSrcModuleName,
      cktsrcdest.cktdestphysicalloc as DestPhysicalLoc,
      nvl(destmt.modulename, 'System') as DestModuleName,
      cktsrcdest.cktsrcmodeltype as srcModelType,
      cktsrcdest.cktsrcobjecttype as srcObjectType,
      cktsrcdest.cktsecsrcmodeltype as secsrcmodeltype,
      cktsrcdest.cktsecsrcobjecttype as secsrcobjecttype,
      cktsrcdest.cktdestmodeltype as DestModelType,
      cktsrcdest.cktdestobjecttype as DestObjectType
FROM
      (select
              ckttbl.cktnodeid as cktnodeid,
              ckttbl.cktuniqueid as cktuniqueid,
             ckttbl.cktsrcphysicalloc as physicalloc,
              mtt.modulename   as modulename,
             ckttbl.cktsrcmodeltype as modeltype,
             ckttbl.cktsrcobjecttype as objecttype
         from ONS155xx_module_type_view mtt,
              circuit_tbl ckttbl
        where mtt.moduletype=ckttbl.cktsrcmoduletype
      ) srcmt,
      (select
              ckttbl.cktnodeid as cktnodeid,
              ckttbl.cktuniqueid as cktuniqueid,
             ckttbl.cktsecsrcphysicalloc as physicalloc,
              mtt.modulename   as modulename,
             ckttbl.cktsecsrcmodeltype as modeltype,
             ckttbl.cktsecsrcobjecttype as objecttype
         from ONS155xx_module_type_view mtt,
              circuit_tbl ckttbl
        where mtt.moduletype=ckttbl.cktsecsrcmoduletype
      ) secsrcmt,
      (select
              ckttbl.cktnodeid as cktnodeid,
              ckttbl.cktuniqueid as cktuniqueid,
             ckttbl.cktdestphysicalloc as physicalloc,
              mtt.modulename   as modulename,
             ckttbl.cktdestmodeltype as modeltype,
             ckttbl.cktdestobjecttype as objecttype
```

```
         from ONS155xx_module_type_view mtt,
              circuit_dest_tbl ckttbl
        where mtt.moduletype=ckttbl.cktdestmoduletype
      ) destmt,
      ONS155xx_Circuit_SRCDEST_VIEW cktsrcdest
WHERE cktsrcdest.cktnodeid = srcmt.cktnodeid(+)
  AND cktsrcdest.cktuniqueid = srcmt.cktuniqueid(+)
  AND cktsrcdest.cktsrcphysicalloc = srcmt.physicalloc(+)
  AND cktsrcdest.cktsrcmodeltype = srcmt.modeltype(+)
  AND cktsrcdest.cktsrcmodeltype = srcmt.objecttype(+)
  AND cktsrcdest.cktnodeid = secsrcmt.cktnodeid(+)
  AND cktsrcdest.cktuniqueid = secsrcmt.cktuniqueid(+)
  AND cktsrcdest.cktsecsrcphysicalloc = secsrcmt.physicalloc(+)
  AND cktsrcdest.cktsecsrcmodeltype = secsrcmt.modeltype(+)
  AND cktsrcdest.cktsecsrcobjecttype = secsrcmt.objecttype(+)
  AND cktsrcdest.cktnodeid = destmt.cktnodeid(+)
  AND cktsrcdest.cktuniqueid = destmt.cktuniqueid (+)
  AND cktsrcdest.cktdestphysicalloc = destmt.physicalloc(+)
  AND cktsrcdest.cktdestmodeltype = destmt.modeltype(+)
  AND cktsrcdest.cktdestobjecttype = destmt.objecttype(+);
```

2.7.35 ONS155xx_CKT_SRC_CDL_View

```
create or replace view ONS155xx_CKT_SRC_CDL_VIEW AS
select
    circuit.CKTNODEID,
    circuit.CKTUNIQUEID,
    circuit.CKTNAME,
    cdl.CDLXMITFLOWID,
    cdl.CDLRCVFLOWID
from
    CIRCUIT_TBL circuit,
    ONS155xx_CDL_FLOW_CONFIG_VIEW cdl
where
    circuit.CKTSRCNODEID      = cdl.NEDBACCESSID and
    circuit.CKTSRCMODULETYPE  = cdl.MODULETYPE and
    circuit.CKTSRCPHYSICALLOC = cdl.PHYSICALLOC and
    circuit.CKTSRCMODELTYPE           = cdl.MODELTYPE and
    circuit.CKTSRCOBJECTTYPE  = cdl.OBJECTTYPE and
    circuit.CKTSRCIFINDEX     = cdl.IFINDEX ;
```

2.7.36 ONS155XX_CKT_Stack_Final_View

```
create or replace view ONS155XX_CKT_STACK_FINAL_VIEW AS
select DISTINCT
        circuit.Name,
        circuit.Description,
        circuit.Type,
        circuit.CktSize,
        circuit.Direction,
        circuit.State,
        circuit.SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        circuit.SecSrcModuleType,
        circuit.SecSrcPhysicalLoc,
        circuit.SecSrcInterfaceId,
        circuit.DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.SrcDropProtType,
        circuit.DestDropProtType,
        circuit.AdditionalInfo,
       cdl.CDLXMITFLOWID as CdlTxFlowId,
       cdl.CDLRCVFLOWID as CdlRxFlowId,
       cktstack.LLCktName,
        cktstack.LLCktDescription,
        cktstack.LLCktType,
        cktstack.LLCktSize,
```

```
        cktstack.AssocType,
        cktstack.LLCktNodeId as LLNodeId,
        cktstack.LLCktUniqueId as LLUniqueId,
         circuit.NodeId,
         circuit.UniqueId,
            circuit.CircuitNote as CircuitNote,
        circuit.myrow as myrow,
         circuit.SrcModelType,
         circuit.SrcObjectType,
         circuit.SecSrcModelType,
         circuit.SecSrcObjectType,
         circuit.DestModelType,
         circuit.DestObjectType
from
         ONS155xx_CIRCUIT_DEST_VIEW circuit,
        ONS155xx_CDL_FLOW_CONFIG_VIEW cdl,
        ONS155xx_CIRCUIT_STACK_VIEW cktstack
where
         circuit.NodeId                = cktstack.HLCKTNODEID and
         circuit.UniqueId          = cktstack.HLCKTUNIQUEID and
        circuit.SrcNodeId          = cdl.NESYSID (+) and
        circuit.SrcModuleType      = cdl.MODULETYPE (+) and
        circuit.SrcPhysicalLoc     = cdl.PHYSICALLOC (+) and
        circuit.SrcModelType       = cdl.MODELTYPE (+) and
        circuit.SrcObjectType      = cdl.OBJECTTYPE (+) and
        circuit.SrcInterfaceId     = cdl.IFINDEX (+) ;
```

2.7.37 ONS155xx_Discovery_Info_View

```
CREATE OR REPLACE VIEW ONS155xx_Discovery_Info_View AS
SELECT
       'Main' as NESYSID,
       disc.startTime,
       disc.endTime,
       disc.status,
       disc.startMode,
       disc.descr,
       disc.rowid as myrow,
       ni.NEAliasID
FROM
       ONS155xx_Discovery_Info_Table disc,
       ne_info_table ni
WHERE
       disc.NEDBAccessID  = 0
union
SELECT
       NESYSID,
       disc.startTime,
       disc.endTime,
       disc.status,
       disc.startMode,
       disc.descr,
       disc.rowid as myrow,
       ni.NEAliasID
FROM
       ONS155xx_Discovery_Info_Table disc,
       ne_info_table ni
WHERE
          disc.NEDBAccessID  = ni.NEDBAccessID ;
```

2.7.38 ONS155xx_Dcrp_View

```
create or replace view Ons155xx_Dcrp_View as
select
        ddt.dcrpname,
        ddt.dcrpseverity,
        dt.dcrpdescr,
        dt.dcrptimestamp,
       neInfo1.NESysId as NodeId1,
       neInfo2.NESysId as NodeId2,
```

```
        ddt.dcrpprobcause,
        ddt.dcrprecaction,
        dt.dcrpdetails,
       dt.rowid as myrow,
      neInfo1.NEAliasID
from
        Ons155xx_Dcrp_Table dt,
        Ons155xx_Dcrp_Definition_Table ddt,
        ne_info_Table neInfo1,
       ne_info_Table neInfo2
where
        dt.dcrptypedbid= ddt.dcrptypedbid AND
    dt.nedbaccessid1 = neInfo1.nedbaccessid (+) AND
    dt.nedbaccessid2 = neInfo2.nedbaccessid (+);
```

2.7.39 ONS155xx_Ether_Hist_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_Ether_Hist_pm_view AS
SELECT
        ni.NESysID,
        en.ModuleName,
        pm.PhysicalLoc,
        pm.NEIfIndex,
        pm.Interval,
        pm.TimeStamp,
        pm.NEIfPmStatus,
        pm.inMaintenance,
        pm.NEDBAccessID,
        pm.ThresholdMask,
        pm.Octets,
        pm.Pkts,
        pm.CRCAlignErrors,
        pm.UndersizePkts,
        pm.OversizePkts,
        en.ModuleType,
        pm.rowid as myrow,
    pm.ModelType,
    pm.ObjectType,
        ni.NEAliasID
FROM
        ONS155xx_ETHER_HIST_pm_table pm,
        ne_info_table ni,
        module_type_table en
WHERE
        pm.ModuleType    = en.ModuleType   AND
        ni.NEModelType = en.ModelType AND
        pm.NEDBAccessID = ni.NEDBAccessID AND
        pm.NEIfPMStatus != 0;
```

2.7.40 ONS155xx_FCM_PE_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_FCM_PE_pm_view AS
SELECT
        ni.NESysID,
        en.ModuleName,
        pm.PhysicalLoc,
        pm.NEIfIndex,
        pm.Interval,
        pm.TimeStamp,
        pm.NEIfPmStatus,
        pm.inMaintenance,
        pm.NEDBAccessID,
        pm.ThresholdMask,
        pm.RxLinkResets,
        pm.TxLinkResets,
        pm.LinkResets,
        pm.RxOfflineSequences,
        pm.TxOfflineSequences,
        pm.LinkFailures,
        pm.LossofSynchs,
        pm.LossofSignals,
        pm.PrimSeqProtocolErrors,
        pm.InvalidTxWords,
```

```
        pm.InvalidCRCs,
        pm.InvalidOrderedSets,
        pm.FrameTooLongs,
        pm.TruncatedFrames,
        pm.AddressErrors,
        pm.DelimiterErrors,
        pm.EncodingDisparityErrors,
        pm.OtherErrors,
        en.ModuleType,
        pm.rowid as myrow,
        pm.ModelType,
        pm.ObjectType,
        pm.IfInOctets,
        pm.IfInUcastPkts,
        pm.IfOutOctets,
        pm.IfOutUcastPkts,
        ni.NEAliasID
FROM
        ONS155xx_FCM_PE_pm_table pm,
        ne_info_table ni,
        module_type_table en
WHERE
        pm.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        pm.NEDBAccessID  = ni.NEDBAccessID AND
        pm.NEIfPMStatus != 0;
```

2.7.41 Ons155xx_Flash_Part_View

```
create or replace view Ons155xx_Flash_Part_View as
select
    ni.NESysId,
         fpt.flashpartname,
         fdt.flashdevname,
        fdt.CPUSlotNum,
        fdt.FlashDevIndex,
         fpt.flashpartsize,
         fpt.flashpartfreesize,
         fpt.flashpartfilenamelen,
         fpt.flashPartIndex,
        fdt.status,
        fpt.rowid as myrow,
       ni.NEAliasID
from
         Ons155xx_Flash_Partition_Table fpt,
         Ons155xx_Flash_Device_Table fdt,
         ne_info_Table ni
where
         fdt.nedbaccessid = ni.nedbaccessid AND
    fpt.flashdevDbid = fdt.flashdevdbid;
```

2.7.42 ONS155xx_Inventory_View

```
CREATE OR REPLACE VIEW ONS155xx_INVENTORY_VIEW AS
SELECT
         info.NESysID as NESysID,
         eqpt.COL7 as Name,
        eqpt.COL2 as Descr,
         mtbl.ModuleName as ModuleName,
         eqpt.COL5 as Type,
         eqpt.COL8 as HardRev,
         eqpt.COL9 as FirmRev,
         eqpt.COL10 as SoftRev,
         eqpt.COL11 as SerialNo,
         eqpt.COL12 as MfgName,
         eqpt.COL13 as ModelName,
         eqpt.COL14 as Alias,
         eqpt.COL15 as AssetID,
        eqpt.physicalloc,
        info.NEDBAccessId,
         eqpt.COL18 as CLEICode,
    eqpt.ModuleType as ModuleType,
        eqpt.rowid as myrow,
```

```
       info.NEAliasID
FROM
        ne_info_table info,
        EQPT_INFO_TABLE eqpt,
        MODULE_TYPE_TABLE mtbl
WHERE
        info.NEDBAccessID = eqpt.NEDBACCESSID and
        info.NEModelType = mtbl.modeltype and
        eqpt.moduletype = mtbl.ModuleType AND
            (eqpt.COL16 = 'true(1)' OR mtbl.ModuleType IN (102, 2020, 1325, 1326, 1337,
0));
```

2.7.43 ONS155XX_Link_Correlator

```
CREATE OR REPLACE FORCE VIEW ONS155XX_LINK_CORRELATOR AS
select
        link1.LINKSRCNODE as NEDBID,
        link1.LINKID as FILTERLINKID,
        link1.LINKSRCPHYSICALLOC as FILTERLINKSRCPHYLOC,
        link1.LINKDSTPHYSICALLOC as FILTERLINKDSTPHYLOC,
        link3.LINKID as WDMLINKID
from
      LINK_TABLE link1,
      LINK_TABLE link2,
      LINK_TABLE link3
where
    (
      (link1.LINKDSTMODULETYPE = 73 or link1.LINKSRCMODULETYPE = 73)
       AND
       (
         (
           (link2.LINKDSTMODULETYPE = 71 AND link2.LINKSRCMODULETYPE = 71)
           AND
            (
              (
                (link1.LINKDSTNODE = link2.LINKDSTNODE)
                AND
              (
                (TRUNC(link1.LINKSRCPHYSICALLOC / 65536) = TRUNC(link2.LINKDSTPHYSICALLOC
/ 65536))
                OR
                (TRUNC(link1.LINKDSTPHYSICALLOC / 65536) = TRUNC(link2.LINKDSTPHYSICALLOC
/ 65536))
              )
              )
              OR
              (
                (link1.LINKDSTNODE = link2.LINKSRCNODE)
                AND
              (
                (TRUNC(link1.LINKSRCPHYSICALLOC / 65536) = TRUNC(link2.LINKSRCPHYSICALLOC
/ 65536))
                OR
                (TRUNC(link1.LINKDSTPHYSICALLOC / 65536) = TRUNC(link2.LINKSRCPHYSICALLOC
/ 65536))
              )
              )
            )
          AND
            link2.LINKID = link3.LINKID
         )
       )
    )
    OR
      (link1.LINKDSTMODULETYPE = 73 OR link1.LINKSRCMODULETYPE = 73)
       AND (link2.LINKDSTMODULETYPE = 72
           AND link1.LINKDSTNODE = link2.LINKSRCNODE
           AND ( (TRUNC(link1.LINKSRCPHYSICALLOC / 65536) = TRUNC(link2.LINKSRCPHYSICALLOC
/ 65536))
                 OR
                 (TRUNC(link1.LINKDSTPHYSICALLOC / 65536) = TRUNC(link2.LINKSRCPHYSICALLOC
/ 65536))
               )
```

```
             AND (link3.LINKDSTMODULETYPE = 71
                  AND (link2.LINKDSTNODE = link3.LINKDSTNODE  OR link2.LINKDSTNODE =
link3.LINKSRCNODE)
                   AND
                 (
                  (TRUNC(link2.LINKDSTPHYSICALLOC / 65536) =
TRUNC(link3.LINKSRCPHYSICALLOC / 65536))
                  OR
                  (TRUNC(link2.LINKDSTPHYSICALLOC / 65536) =
TRUNC(link3.LINKDSTPHYSICALLOC / 65536))
                 )
                  )
         )
         OR
         (link2.LINKSRCMODULETYPE = 72
           AND link1.LINKDSTNODE = link2.LINKDSTNODE
           AND ( (TRUNC(link1.LINKSRCPHYSICALLOC / 65536) = TRUNC(link2.LINKDSTPHYSICALLOC
/ 65536))
                 OR
                 (TRUNC(link1.LINKDSTPHYSICALLOC / 65536) = TRUNC(link2.LINKDSTPHYSICALLOC
/ 65536))
               )
             AND (link3.LINKDSTMODULETYPE = 71
                  AND (link2.LINKSRCNODE = link3.LINKDSTNODE  OR link2.LINKSRCNODE =
link3.LINKSRCNODE)
                   AND
                 (
                  (TRUNC(link2.LINKSRCPHYSICALLOC / 65536) =
TRUNC(link3.LINKSRCPHYSICALLOC / 65536))
                  OR
                  (TRUNC(link2.LINKSRCPHYSICALLOC / 65536) =
TRUNC(link3.LINKDSTPHYSICALLOC / 65536))
                 )
                  )
         )
;
```

2.7.44 ONS155xx_Module_Type_View

```
CREATE OR REPLACE VIEW ONS155xx_Module_Type_view AS
SELECT DISTINCT
      mtt.moduleType    as ModuleType,
      mtt.moduleName   as ModuleName
FROM
      NE_INFO_TABLE neInfo,
      MODULE_TYPE_TABLE mtt
WHERE neinfo.nemodeltype = mtt.modeltype and
      neinfo.nemodeltype in (9, 11, 12, 13);
```

2.7.45 ONS155xx_OPT_Power_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_opt_power_pm_view AS
SELECT
       ni.NESysID,
       en.ModuleName,
       pm.PhysicalLoc,
       pm.NEIfIndex,
       pm.Interval,
       pm.TimeStamp,
       pm.NEIfPmStatus,
       pm.inMaintenance,
       pm.NEDBAccessID,
       pm.ThresholdMask,
       pm.MeanPower,
       pm.MaxPower,
       pm.MinPower,
       pm.MeanAmbTemp,
       pm.MaxAmbTemp,
       pm.MinAmbTemp,
       pm.MeanLaserTemp,
       pm.MaxLaserTemp,
```

```
        pm.MinLaserTemp,
        pm.MeanBiasCurrent,
        pm.MaxBiasCurrent,
        pm.MinBiasCurrent,
        pm.AvailableSecs,
        pm.Direction,
        en.ModuleType,
        pm.rowid as myrow,
        pm.ModelType,
        pm.ObjectType,
        ni.NEAliasID
FROM
        ONS155xx_opt_power_pm_table pm,
        ne_info_table ni,
        module_type_table en
WHERE
        pm.ModuleType    = en.ModuleType   AND
        ni.NEModelType       = en.ModelType    AND
        pm.NEDBAccessID  = ni.NEDBAccessID AND
        pm.NEIfPMStatus != 0;
```

2.7.46 ONS155XX_Path_CKT_Final_View

```
create or replace view ONS155XX_PATH_CKT_FINAL_VIEW AS
select DISTINCT
         circuit.Name,
         circuit.Type,
         circuit.CktSize,
         circuit.PathNumber,
         neinfo1.NESYSID as SrcNodeId,
         circuit.SrcModuleType,
         circuit.SrcPhysicalLoc,
         circuit.SrcInterfaceId,
         neinfo2.NESYSID as DestNodeId,
         circuit.DestModuleType,
         circuit.DestPhysicalLoc,
         circuit.DestInterfaceId,
         circuit.PathStatus,
         circuit.IsWorking,
         circuit.IsActive,
         circuit.DownInterfaces,
         circuit.CktNodeId,
         circuit.CktUniqueId,
         circuit.PathDBID,
        pathDestPwr.PathDestOpticalPower as DestOpticalPower,
        circuit.PathDescr,
        pathDestPwr.PathDestPwrMonNodeID,
         pathDestPwr.PathDestPwrMonPhysicalLoc,
        pathDestPwr.PathDestPwrMonIfIndex,
        pathDestPwr.PathDestPwrMonModuleType,
         circuit.SrcModelType,
         circuit.SrcObjectType,
         circuit.DestModelType,
         circuit.DestObjectType,
         pathDestPwr.PathDestPwrMonModelType,
         pathDestPwr.PathDestPwrMonObjectType
from
        ONS155xx_CKT_PATH_JOIN_VIEW circuit,
        ONS155xx_PATH_DEST_PWR_VIEW pathDestPwr,
         NE_INFO_TABLE neinfo1,
         NE_INFO_TABLE neinfo2
where
         circuit.SrcNodeID             = neinfo1.NEDBACCESSID and
         circuit.DestNodeID            = neinfo2.NEDBACCESSID and
         circuit.PATHDBID           = pathDestPwr.PATHDBID ;
```

2.7.47 ONS155xx_Path_Dest_Pwr_View

```
create or replace view ONS155xx_PATH_DEST_PWR_VIEW AS
select
```

```
        path1.PATHDBID,
        path1.PATHDESTNODEID,
        path1.PathDescr,
        path1.PathDestOpticalPower,
        path1.PathDestPwrMonNodeID,
                path1.PathDestPwrMonPhysicalLoc,
        path1.PathDestPwrMonIfIndex,
        path1.PathDestPwrMonModuleType,
                path1.PathDestPwrMonModelType,
                path1.PathDestPwrMonObjectType
from
        ONS155xx_PATH_PWR_NO_MON_VIEW path1
union all
select
        path2.PATHDBID,
        path2.PATHDESTNODEID,
        path2.PathDescr,
        path2.PathDestOpticalPower,
        path2.PathDestPwrMonNodeID,
                path2.PathDestPwrMonPhysicalLoc,
        path2.PathDestPwrMonIfIndex,
        path2.PathDestPwrMonModuleType,
                path2.PathDestPwrMonModelType,
                path2.PathDestPwrMonObjectType
from
        ONS155xx_PATH_PWR_NO_AVA_VIEW path2
union all
select
        path3.PATHDBID,
        path3.PATHDESTNODEID,
        path3.PathDescr,
        path3.PathDestOpticalPower,
        path3.PathDestPwrMonNodeID,
                path3.PathDestPwrMonPhysicalLoc,
        path3.PathDestPwrMonIfIndex,
        path3.PathDestPwrMonModuleType,
                path3.PathDestPwrMonModelType,
                path3.PathDestPwrMonObjectType
from
        ONS155xx_PATH_PWR_MONITOR_VIEW path3 ;
```

2.7.48 ONS155XX_Path_Link_Final_View

```
create or replace view ONS155XX_PATH_LINK_FINAL_VIEW AS
select DISTINCT
        circuit.Name,
        circuit.Type,
        circuit.CktSize,
        circuit.PathNumber,
        neinfo1.NESYSID as SrcNodeId,
        circuit.SrcModuleType,
        circuit.SrcPhysicalLoc,
        circuit.SrcInterfaceId,
        neinfo2.NESYSID as DestNodeId,
        circuit.DestModuleType,
        circuit.DestPhysicalLoc,
        circuit.DestInterfaceId,
        circuit.PathStatus,
        circuit.IsWorking,
        circuit.IsActive,
        circuit.DownInterfaces,
        circuit.CktNodeId,
        circuit.CktUniqueId,
        circuit.PathDBID,
       pathDestPwr.PathDestOpticalPower as DestOpticalPower,
       pathLink.LinkId    as LinkId,
       circuit.PathDescr,
       pathDestPwr.PathDestPwrMonNodeID,
        pathDestPwr.PathDestPwrMonPhysicalLoc,
       pathDestPwr.PathDestPwrMonIfIndex,
       pathDestPwr.PathDestPwrMonModuleType,
        circuit.SrcModelType,
        circuit.SrcObjectType,
        circuit.DestModelType,
```

```
        circuit.DestObjectType,
        pathDestPwr.PathDestPwrMonModelType,
        pathDestPwr.PathDestPwrMonObjectType
from
       ONS155xx_CKT_PATH_JOIN_VIEW circuit,
       ONS155xx_PATH_DEST_PWR_VIEW pathDestPwr,
       ONS155xx_Path_Link_View pathLink,
        NE_INFO_TABLE neinfo1,
        NE_INFO_TABLE neinfo2
where
        circuit.SrcNodeID             = neinfo1.NEDBACCESSID   and
        circuit.DestNodeID            = neinfo2.NEDBACCESSID and
        circuit.PATHDBID           = pathDestPwr.PATHDBID   and
       circuit.PATHDBID            = pathLink.PATHDBID ;
```

2.7.49 ONS155xx_Path_Link_View

```
create or replace view ONS155xx_Path_Link_View as
select
        pathhop.PATHDBID,
       ccLink.CktCCLinkId as LinkId
from
       ONS155xx_PATH_HOP_TBL pathhop,
       ONS155xx_CKT_CC_LINK_View ccLink
where
       pathhop.PathHopType = 1 and
       pathhop.PathHopDBID = ccLink.CKTCCDBID
union all
select
        pathhop.PATHDBID,
       span.CktLinkId as LinkId
from
        CIRCUIT_SPAN_TBL span,
       ONS155xx_PATH_HOP_TBL pathhop
where
       pathhop.PathHopType = 2 and
       pathhop.PathHopDBID = span.CktSpanID ;
```

2.7.50 ONS155xx_Path_Pwr_Monitor_View

```
create or replace view ONS155xx_PATH_PWR_MONITOR_VIEW AS
select DISTINCT
        path.PATHDBID,
        path.PathDestNodeID,
        path.PathDestPhysicalLoc,
        path.PathDestIfIndex,
        path.PathDestModuleType,
        path.pathDescr,
        nvl(pwr.OpticalPower, 'NotAvailable') as PathDestOpticalPower,
        path.PathDestPwrMonNodeID,
        path.PathDestPwrMonPhysicalLoc,
        path.PathDestPwrMonIfIndex,
        path.PathDestPwrMonModuleType,
        path.PathDestModelType,
        path.PathDestObjectType,
        path.PathDestPwrMonModelType,
        path.PathDestPwrMonObjectType
from
        ONS155xx_PATH_TBL path,
        ONS155xx_CKT_OPT_PWR_VIEW pwr
where
        path.PathDestPwrMonNodeID != -1 and
        path.PathDestPwrMonNodeID = pwr.NEDBACCESSID(+) and
        path.PathDestPwrMonPhysicalLoc = pwr.PwrSrcPhysicalLoc(+) and
        path.PathDestPwrMonModelType = pwr.PwrSrcModelType(+) and
        path.PathDestPwrMonObjectType = pwr.PwrSrcObjectType(+) and
        path.PathDestPwrMonIfIndex = pwr.PwrSrcIfIndex(+) and
        path.PathDestPwrMonModuleType = pwr.PwrSrcModuleType(+) ;
```

2.7.51 ONS155xx_Path_Pwr_No_Ava_View

```
create or replace view ONS155xx_PATH_PWR_NO_AVA_VIEW AS
select
        path.PATHDBID,
        path.PathDestNodeId,
        path.PathDescr,
        'NotAvailable' as PathDestOpticalPower,
        path.PathDestPwrMonNodeID,
        path.PathDestPwrMonPhysicalLoc,
        path.PathDestPwrMonIfIndex,
        path.PathDestPwrMonModuleType,
        path.PathDestPwrMonModelType,
        path.PathDestPwrMonObjectType
from
        ONS155xx_PATH_TBL path
where
       path.PathStatus = 3 and
        path.PathDestPwrMonNodeID = -1;
```

2.7.52 ONS155xx_Path_Pwr_No_Mon_View

```
create or replace view ONS155xx_PATH_PWR_NO_MON_VIEW AS
select
        path.PATHDBID,
        path.PathDestNodeId,
        path.PathDescr,
        'NotMonitored' as PathDestOpticalPower,
        path.PathDestPwrMonNodeID,
        path.PathDestPwrMonPhysicalLoc,
        path.PathDestPwrMonIfIndex,
        path.PathDestPwrMonModuleType,
        path.PathDestPwrMonModelType,
        path.PathDestPwrMonObjectType
from
        ONS155xx_PATH_TBL path
where
       path.PathStatus in (1,2) and
        path.PathDestPwrMonNodeID = -1;
```

2.7.53 ONS155xx_Path_Stack_Final_View

```
create or replace view ONS155xx_PATH_STACK_FINAL_VIEW AS
select DISTINCT
        circuitpath.Name,
        circuitpath.Type,
        circuitpath.CktSize,
        circuitpath.PathNumber,
        circuitpath.SrcNodeId,
        circuitpath.SrcModuleType,
        circuitpath.SrcPhysicalLoc,
        circuitpath.SrcInterfaceId,
        circuitpath.DestNodeId,
        circuitpath.DestModuleType,
        circuitpath.DestPhysicalLoc,
        circuitpath.DestInterfaceId,
        circuitpath.PathStatus,
        circuitpath.IsWorking,
        circuitpath.IsActive,
        circuitpath.DownInterfaces,
        circuitpath.CktNodeId,
        circuitpath.CktUniqueId,
        circuitpath.PathDBID,
       circuitpath.DestOpticalPower,
       circuitpath.PathDescr,
       circuitpath.PathDestPwrMonNodeID,
        circuitpath.PathDestPwrMonPhysicalLoc,
       circuitpath.PathDestPwrMonIfIndex,
       circuitpath.PathDestPwrMonModuleType,
        cktstack.LLCktName,
        cktstack.LLCktType,
        cktstack.LLCktSize,
       cktstack.LLCktPathDescr,
        cktstack.LLCKTNODEID,
        cktstack.LLCKTUNIQUEID,
```

```
        cktstack.LLCktPathDBID as LLPathDBID,
        circuitpath.PathDBID as myrow,
         circuitpath.SrcModelType,
         circuitpath.SrcObjectType,
         circuitpath.DestModelType,
         circuitpath.DestObjectType,
         circuitpath.PathDestPwrMonModelType,
         circuitpath.PathDestPwrMonObjectType
from
        ONS155xx_CKT_PATH_STACK_VIEW cktstack,
        ONS155XX_PATH_CKT_FINAL_VIEW circuitpath
where
        circuitpath.CktNodeId     =   cktstack.HLCktNodeId and
        circuitpath.CktUniqueId       =   cktstack.HLCktUniqueId and
        circuitpath.PathDBID      =   cktstack.HLCktPathDBID ;
```

2.7.54 ONS155xx_Phy_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_phy_pm_view AS
SELECT
       ni.NESysID,
       en.ModuleName,
       pm.PhysicalLoc,
       pm.NEIfIndex,
       pm.Interval,
       pm.TimeStamp,
       pm.NEIfPmStatus,
       pm.inMaintenance,
       pm.NEDBAccessID,
       pm.ThresholdMask,
       pm.RxCVRD,
       pm.RxCRC,
       pm.TxEncapFarEndPktErrors,
       en.ModuleType,
       pm.rowid as myrow,
       pm.ModelType,
       pm.ObjectType,
       ni.NEAliasID
FROM
       ONS155xx_phy_pm_table pm,
       ne_info_table ni,
       module_type_table en
WHERE
       pm.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       pm.NEDBAccessID  = ni.NEDBAccessID AND
       pm.NEIfPMStatus != 0;
```

2.7.55 ONS155xx_SONET_Section_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_sonet_section_pm_view AS
SELECT
       ni.NESysID,
       en.ModuleName,
       pm.PhysicalLoc,
       pm.NEIfIndex,
       pm.Interval,
       pm.TimeStamp,
       pm.NEIfPmStatus,
       pm.InMaintenance,
       pm.NEDBAccessID,
       pm.ThresholdMask,
       pm.ErrSecs,
       pm.SevErrSecs,
       pm.SevErrFramingSecs,
       pm.CodeViolations,
       en.ModuleType,
       pm.rowid as myrow,
       pm.ModelType,
       pm.ObjectType,
       ni.NEAliasID
FROM
       ONS155xx_sonet_sec_pm_table pm,
```

```
       ne_info_table ni,
       module_type_table en
WHERE
       pm.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       pm.NEDBAccessID  = ni.NEDBAccessID AND
       pm.NEIfPMStatus != 0;
```

2.7.56 ONS155xx_Tca_PM_View

```
CREATE OR REPLACE VIEW ONS155xx_tca_pm_view AS
SELECT
       aa.NEDBAccessID,
       aa.ModuleType,
       aa.PhysicalLoc,
       aa.ModuleOrIfIndex,
       aa.ActiveAlarmTimestamp,
       aa.ClearAlarmTimestamp,
       aa.ActiveAlarmType,
       aa.ActiveAlarmSeverity,
       aa.ActiveAlarmIndex,
       aa.rowid AS myrow,
       aa.ModelType,
       aa.ObjectType,
       aa.StrObjInstance
FROM
       ne_info_table ni,
       active_alarm_table aa
WHERE
       aa.NEDBAccessID  = ni.NEDBAccessID AND
       ni.NEPMENABLED IN (1,2) AND
       ( aa.ActiveAlarmType BETWEEN 9975 AND 9999 OR
         aa.ActiveAlarmType BETWEEN 14500 AND 14610 OR
         aa.ActiveAlarmType BETWEEN 14701 AND 15000) ;
```

2.8 Cisco ONS 156xx Views

2.8.1 ONS15600sdh_HO_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15600sdh_ho_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.FeHpEb as FeHpEb,
       csdh.FeHpEs as FeHpEs,
       csdh.FeHpSes as  FeHpSes,
       csdh.FeHpUas as FeHpUas,
       csdh.FeHpFc as FeHpFc,
       csdh.FeHpBbe as FeHpBbe,
       csdh.FeHpEsr as FeHpEsr,
       csdh.FeHpSesr as FeHpSesr,
       csdh.FeHpBber as  FeHpBber,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ONS15454sdh_ho_pm_table csdh,
       ne_info_table ni,
```

```
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.2 ONS15600sdh_HO_PM_View

```
CREATE OR REPLACE VIEW ONS15600sdh_ho_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.HpEb as HpEb,
       csdh.HpEs as HpEs,
       csdh.HpSes as HpSes,
       csdh.HpUas as HpUas,
       csdh.HpFc as HpFc,
       csdh.HpBbe as HpBbe,
       csdh.HpEsr as HpEsr,
       csdh.HpSesr as HpSesr,
       csdh.HpBber as HpBber,
    csdh.HpPpjcpdet as HpPjpIn,
       csdh.Hpnpjcpdet as HpPjpOut,
       csdh.HpPpjcpgen as HpPjnIn,
       csdh.HpNpjcpgen as HpPjnOut, csdh.StsPathWidth as StsPathWidth,
       csdh.NEDBAccessID as NEDBAccessID,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ONS15454sdh_ho_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.3 ONS15600sdh_MS_PM_FE_View

```
CREATE OR REPLACE VIEW ONS15600sdh_ms_pm_fe_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.FeMsEb as FeMsEb,
       csdh.FeMsEs as FeMsEs,
       csdh.FeMsSes as FeMsSes,
       csdh.FeMsUas as FeMsUas,
       csdh.FeMsFc as FeMsFc,
       csdh.FeMsBbe as FeMsBbe,
       csdh.NEDBAccessID as NEDBAccessId,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
```

```
       csdh.InvalidityList,
       csdh.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ONS15454sdh_ms_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.4 ONS15600sdh_MS_PM_View

```
CREATE OR REPLACE VIEW ONS15600sdh_ms_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance InMaintenance,
       csdh.MsEb as MsEb,
       csdh.MsEs as MsEs,
       csdh.MsSes as MsSes,
       csdh.MsUas as MsUas,
       csdh.MsFc as MsFc,
       csdh.MsBbe as MsBbe,
       csdh.MsPsc as MsPsc,
       csdh.MsPsd as MsPsd,
       csdh.MSPscS as MSPscS,
       csdh.MSPsdS as MSPsdS,
       csdh.MSPscW as MSPscW,
       csdh.MSPsdW as MSPsdW,
       csdh.MSPscR as MSPscR,
       csdh.MSPsdR as MSPsdR,
       csdh.NEDBAccessID as NEDBAccessId,
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ONS15454sdh_ms_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.5 ONS15600sdh_RS_PM_View

```
CREATE OR REPLACE VIEW ONS15600sdh_rs_pm_view AS
SELECT
       ni.NESysID as NESysId,
       en.ModuleName as ModuleName,
       csdh.PhysicalLoc,
       csdh.NEIfIndex as NEIfIndex,
       csdh.TimeStamp as TimeStamp,
       csdh.NEIfPmStatus as NEIfPmStatus,
       csdh.InMaintenance as InMaintenance,
       csdh.RsEb as RsEb,
       csdh.RsEs as RsEs,
       csdh.RsSes as RsSes,
       csdh.RsOfs as RsSefs,
       csdh.RsBbe as RsBbe,
       csdh.NEDBAccessID as NEDBAccessID,
```

```
       csdh.thresholdList as ThresholdList,
       en.ModuleType,
       csdh.IS24H,
       csdh.ModelType,
       csdh.ObjectType,
       csdh.StrObjInstance,
       csdh.InvalidityList,
       csdh.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ONS15454sdh_rs_pm_table csdh,
       ne_info_table ni,
       module_type_table en
WHERE
       csdh.ModuleType    = en.ModuleType   AND
       ni.NEModelType       = en.ModelType    AND
       csdh.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.6 ONS15600_Enet_PM_View

```
CREATE OR REPLACE VIEW ONS15600_enet_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       enet.Physicalloc,
       enet.NEIfIndex,
       enet.TimeStamp,
       enet.NEIfPmStatus,
       enet.InMaintenance,
       enet.IfInOctets,
       enet.IfInUcastPkts,
       enet.IfInMulticastPkts,
       enet.IfInBroadcastPkts,
       enet.IfInDiscards,
       enet.IfInErrors,
       enet.IfOutOctets,
       enet.IfOutUcastPkts,
       enet.IfOutMulticastPkts,
       enet.IfOutBroadcastPkts,
       enet.IfOutDiscards,
       enet.Dot3StatsAlignmentErrors,
       enet.Dot3StatsFCSErrors,
       enet.Dot3StatsFrameTooLong,
       enet.EtherStatsUndersizePkts,
       enet.EtherStatsFragments,
       enet.EtherStatsPkts64Octets,
       enet.EtherStatsPkts65to127Octets,
       enet.EtherStatsPkts128to255Octets,
       enet.EtherStatsPkts256to511Octets,
       enet.EtherStatsPkts512to1023Octets,
       enet.EtherStatsPkts1024to1518Octets,
       enet.EtherStatsBroadcastPkts,
       enet.EtherStatsMulticastPkts,
       enet.EtherStatsOversizePkts,
       enet.EtherStatsJabbers,
       enet.EtherStatsOctets,
       enet.EtherStatsDropEvents,
       enet.RxPauseFrames,
       enet.TxPauseFrames,
       enet.EtherStatsPkts,
       enet.IfOutErrors,
       enet.Dot3StatsInternalMacRxErrors,
       enet.Dot3StatsInternalMacTxErrors,
       enet.Dot3StatsSymbolErrors,
       enet.RxEtherUtilizationStats,
       enet.TxEtherUtilizationStats,
       enet.thresholdList as ThresholdList,
       enet.IS24H,
       enet.ModelType,
       enet.ObjectType,
       enet.StrObjInstance,
       enet.InvalidityList,
```

```
       enet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ons15600_enet_pm_table enet,
       ne_info_table ni,
       module_type_table en
WHERE
       enet.ModuleType    = en.ModuleType   AND
       ni.NEModelType     = en.ModelType    AND
       enet.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.7 ONS15600_Pos_PM_View

```
CREATE OR REPLACE VIEW ONS15600_pos_pm_view AS
SELECT
       ni.NESysID,
       ni.NEDbaccessid,
       en.ModuleName,
       en.ModuleType,
       pos.Physicalloc,
       pos.NEIfIndex,
       pos.TimeStamp,
       pos.NEIfPmStatus,
       pos.InMaintenance,
       pos.hdlcInOctets,
       pos.hdlcOutOctets,
       pos.rxTotalPackets,
       pos.txTotalPackets,
       pos.hdlcRxAborts,
       pos.MediaIndStatsRxFramesBadCRC,
       pos.GfpStatsRxFrame,
       pos.GfpStatsTxFrame,
       pos.GfpStatsRxOctets,
       pos.GfpStatsTxOctets,
       pos.GfpStatsRxCRCErrors,
       pos.GfpStatsRxMBitErrors,
       pos.GfpStatsRxSBitErrors,
       pos.GfpStatsRxTypeInvalid,
       pos.RxPktsDroppedInernalCongestion,
       pos.MediaIndStatsRxShortPkts,
       pos.MediaIndStatsRxFramesTruncated,
       pos.MediaIndStatsRxFramesTooLong,
       pos.RxEtherUtilizationStats,
       pos.TxEtherUtilizationStats,
       pos.thresholdList as ThresholdList,
       pos.IS24H,
       pos.ModelType,
       pos.ObjectType,
       pos.StrObjInstance,
       pos.InvalidityList,
       pos.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
       ons15600_pos_pm_table pos,
       ne_info_table ni,
       module_type_table en
WHERE
       pos.ModuleType    = en.ModuleType   AND
       ni.NEModelType    = en.ModelType    AND
       pos.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.8 ONS15600_SONET_Line_FE_PM_View

```
CREATE OR REPLACE VIEW ONS15600_SONET_LINE_FE_PM_VIEW AS
SELECT
    ni.NESysID as NESysId,
    en.ModuleName as ModuleName,
    csonet.PhysicalLoc,
    csonet.NEIfIndex as NEIfIndex,
    csonet.TimeStamp as TimeStamp,
    csonet.NEIfPMStatus as NEIfPMStatus,
    csonet.inMaintenance as InMaintenance,
    csonet.FECodeViolations as FECodeViolations,
```

```
    csonet.FEErrSecs as FEErrSecs,
    csonet.FESevErrSecs as FESevErrSecs,
    csonet.FEUnavailableSecs as FEUnavailableSecs,
    csonet.FEFcL as FEFcL, csonet.NEDBAccessID as NEDBAccessId,
    csonet.thresholdList as ThresholdList,
    en.ModuleType,
    csonet.IS24H,
     csonet.ModelType,
     csonet.ObjectType,
     csonet.StrObjInstance,
     csonet.InvalidityList,
    csonet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
    ONS15454_Sonet_Line_PM_Table csonet,
    ne_info_table ni,
    module_type_table en
WHERE
    csonet.ModuleType    = en.ModuleType   AND
    ni.NEModelType       = en.ModelType    AND
    csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.9 ONS15600_SONET_Line_PM_View

```
CREATE OR REPLACE VIEW ONS15600_SONET_LINE_PM_VIEW AS
SELECT
    ni.NESysID as NESysId,
    en.ModuleName as ModuleName,
    csonet.PhysicalLoc,
    csonet.NEIfIndex as NEIfIndex,
    csonet.TimeStamp as TimeStamp,
    csonet.NEIfPMStatus as NEIfPMStatus,
    csonet.inMaintenance as InMaintenance,
    csonet.CodeViolations as CodeViolations,
    csonet.ErrSecs as ErrSecs,
    csonet.SevErrSecs as SevErrSecs,
    csonet.UnavailableSecs as UnavailSecs,
    csonet.PpjcPdet as PpjcPdet,
        csonet.NpjcPdet as NpjcPdet,
        csonet.PpjcPgen as PpjcPgen,
        csonet.NpjcPgen as NpjcPgen,
    csonet.FcL as Fcl,
    csonet.Psc as Psc,
    csonet.Psd as Psd,
    csonet.PjDiff as PjDiff,
        csonet.PjpSec as PjpSec,
        csonet.PjnSec as PjnSec,
    csonet.Pscw as Pscw,
    csonet.Psdw as Psdw,
    csonet.Pscr as Pscr,
    csonet.Psdr as Psdr,
    csonet.Pscs as Pscs,
    csonet.Psds as Psds,
    csonet.NEDBAccessID as NEDBAccessId,
    csonet.thresholdList as ThresholdList,
    en.ModuleType,
    csonet.IS24H,
        csonet.ModelType,
        csonet.ObjectType,
        csonet.StrObjInstance,
        csonet.InvalidityList,
    csonet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
    ONS15454_Sonet_Line_PM_Table csonet,
    ne_info_table ni,
    module_type_table en
WHERE
    csonet.ModuleType  = en.ModuleType   AND
    ni.NEModelType     = en.ModelType    AND
    csonet.NEDBAccessID    = ni.NEDBAccessID;
```

2.8.10 ONS15600_SONET_Path_FE_PM_View

```
CREATE OR REPLACE VIEW ONS15600_SONET_PATH_FE_PM_VIEW AS
SELECT
    ni.NESysID as NESysId,
    en.ModuleName as ModuleName,
    csonet.PhysicalLoc,
    csonet.NEIfIndex as NEIfIndex,
    csonet.TimeStamp as TimeStamp,
    csonet.NEIfPmStatus as NEIfPmStatus,
    csonet.inMaintenance as InMaintenance,
    csonet.FESonetPathCodeViolations as FESonetPathCodeViolations,
    csonet.FESonetPathErrSecs as FESonetPathErrSecs,
    csonet.FESonetPathSevErrSecs as FESonetPathSevErrSecs,
    csonet.FESonetPathUnavailableSecs as FESonetPathUnavailableSecs,
    csonet.FESonetPathFC as FESonetPathFC,   csonet.NEDBAccessID as NEDBAccessID,
    csonet.thresholdList as ThresholdList,
    en.ModuleType,
    csonet.IS24H,
     csonet.ModelType,
     csonet.ObjectType,
     csonet.StrObjInstance,
     csonet.InvalidityList,
    csonet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
    ONS15454_Sonet_Path_PM_Table csonet,
    ne_info_table ni,
    module_type_table en
WHERE
    csonet.ModuleType    = en.ModuleType   AND
    ni.NEModelType       = en.ModelType    AND
    csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.11 ONS15600_SONET_Path_PM_View

```
CREATE OR REPLACE VIEW ONS15600_SONET_PATH_PM_VIEW AS
SELECT
    ni.NESysID as NESysId,
    en.ModuleName as ModuleName,
    csonet.PhysicalLoc,
    csonet.NEIfIndex as NEIfIndex,
    csonet.TimeStamp as TimeStamp,
    csonet.NEIfPMStatus as NEIfPmStatus,
    csonet.inMaintenance as InMaintenance,
    csonet.SonetPathCodeViolations as CodeViolations,
    csonet.SonetPathErrSecs as ErrSecs,
    csonet.SonetPathSevErrSecs SevErrSecs,
    csonet.SonetPathUnavailableSecs as LineUnavailSecs,
    csonet.SonetPathFC  as SonetPathFC,
    csonet.PpjcPdet as PpjcPdet,
    csonet.NpjcPdet as NpjcPdet,
    csonet.PpjcPgen as PpjcPgen,
    csonet.NpjcPgen as NpjcPgen,
    csonet.PjDiff as PjDiff,
    csonet.PjpSec as PjpSec,
    csonet.PjnSec as PjnSec,
    csonet.StsPathWidth as StsPathWidth,
    csonet.NEDBAccessID as NEDBAccessID,
    csonet.thresholdList as ThresholdList,
    en.ModuleType,
    csonet.IS24H,
     csonet.ModelType,
     csonet.ObjectType,
     csonet.StrObjInstance,
     csonet.InvalidityList,
    csonet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
    ONS15454_Sonet_Path_PM_Table csonet,
    ne_info_table ni,
    module_type_table en
WHERE
```

```
    csonet.ModuleType    = en.ModuleType   AND
    ni.NEModelType       = en.ModelType    AND
    csonet.NEDBAccessID  = ni.NEDBAccessID;
```

2.8.12 ONS15600_SONET_Sec_PM_View

```
CREATE OR REPLACE VIEW ONS15600_Sonet_Sec_PM_View AS
SELECT
    ni.NESysID as NESysId,
    en.ModuleName as ModuleName,
    csonet.PhysicalLoc as PhysicalLoc,
    csonet.NEIfIndex as NEIfIndex,
    csonet.TimeStamp as TimeStamp,
    csonet.NEIfPMStatus as NEIfPMStatus,
    csonet.inMaintenance as InMaintenance,
    csonet.SevErrFrameSecs as ErrFrameSecs,
    csonet.CodeViolations as CodeViolations,
    csonet.ErrSecs as ErrSecs,
    csonet.SevErrSecs as SevErrSecs,
    csonet.NEDBAccessID as NEDBAccessID,
    csonet.thresholdList as ThresholdList,
    en.ModuleType,
    csonet.IS24H,
     csonet.ModelType,
     csonet.ObjectType,
     csonet.StrObjInstance,
     csonet.InvalidityList,
    csonet.rowid as myrow,
    ni.NEAliasID as NEAliasID
FROM
    ONS15454_Sonet_Sec_PM_Table csonet,
    ne_info_table ni,
    module_type_table en
WHERE
    csonet.ModuleType    = en.ModuleType   AND
    ni.NEModelType       = en.ModelType    AND
    csonet.NEDBAccessID  = ni.NEDBAccessID;
```

3 Database Schema Functions

This chapter describes the functions defined in Prime Optical tables. It contains the following section:

- Prime Optical Functions

3.1 Prime Optical Functions

This section describes the functions that Prime Optical uses. Functions read parameters and return a value. The functions are listed in alphabetical order.

3.1.1 CollapsedDestNodeName

```
CREATE or replace function collapseddestnodename(CKTNODEIDIN varchar,CKTUNIQUEIDIN
varchar) return varchar2
is
cursor cktdestnodename is
select CktDestNodeId
from circuit_dest_tbl
where CKTNODEID=CKTNODEIDIN
and CKTUNIQUEID =CKTUNIQUEIDIN
and CktDestNodeId is not null;
finaldestNode varchar2(10000);
DestNedbaccessid varchar2(1000);
BEGIN

for rec in cktdestnodename loop
    SELECT nedbaccessid INTO DestNedbaccessid
    FROM ne_info_table
    WHERE nenodeid= rec.CktDestNodeId and rec.CktDestNodeId != 0 ;

   if cktdestnodename%ROWCOUNT = 1 then
    finaldestNode:= DestNedbaccessid;
  else
    finaldestNode := finaldestNode||','||DestNedbaccessid;
  end if;
end loop;
return finaldestNode;
END;
```

3.1.2 CollapsedDestNodeName

```
CREATE or replace function collapseddestnodename (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktdestnodename is
    select CktCtpNodeId
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type = 2
    and CktCtpNodeId is not null;
    finaldestNode varchar2(10000);
    DestNedbaccessid varchar2(1000);
BEGIN

    for rec in cktdestnodename loop
        SELECT nedbaccessid INTO DestNedbaccessid
        FROM ne_info_table
        WHERE nenodeid= rec.CktCtpNodeId and rec.CktCtpNodeId != 0 ;
```

```
       if cktdestnodename%ROWCOUNT = 1 then
           finaldestNode:= DestNedbaccessid;
       else
           finaldestNode := finaldestNode||','||DestNedbaccessid;
       end if;
    end loop;

    return finaldestNode;

END collapseddestnodename;
```

3.1.3 CollapsedEqptPortName

```
CREATE or replace function collapsedeqptportname(CKTNODEIDIN varchar,CKTUNIQUEIDIN
varchar)
return varchar2
IS
   cursor eqptcollpasedport is
   select  CktCtpNodeId,
   cktCtpPhysicalloc,
   cktCtpIfindex,
   cktCtpModuleType
   from circuit_ctp_tbl
   where CKTNODEID=CKTNODEIDIN
   and CKTUNIQUEID =CKTUNIQUEIDIN
   and ctp_type = 2
   and CktCtpNodeId is not null
   and cktCtpPhysicalloc is not null
   and cktCtpIfindex is not null
   and cktCtpModuleType is not null;
   finaleqptinfoport varchar2(10000);
   DestNedbaccessid varchar2(1000);
BEGIN

   for rec in eqptcollpasedport loop
       SELECT nedbaccessid INTO DestNedbaccessid
       FROM ne_info_table
       WHERE nenodeid= rec.CktCtpNodeId and rec.CktCtpNodeId != 0;

       if eqptcollpasedport%ROWCOUNT = 1 then
           finaleqptinfoport:= '@'||eqptinfoport(DestNedbaccessid,rec.cktCtpPhysicalloc,0,
rec.cktCtpModuleType);
       else
           finaleqptinfoport :=
finaleqptinfoport||','||'@'||eqptinfoport(DestNedbaccessid,rec.cktCtpPhysicalloc,0,
rec.cktCtpModuleType);
       end if;

   end loop;

   return finaleqptinfoport;

END collapsedeqptportname;
```

3.1.4 CollapsedIfIndex

```
CREATE or replace function collapsedifindex (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktobjectindexcursor is
    select CktCtpIfIndex
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type = 2
    and CktCtpIfIndex is not null;
    finalifindex varchar2(10000);
BEGIN

    for rec in cktobjectindexcursor loop
        if cktobjectindexcursor%ROWCOUNT = 1 then
```

```
            finalifindex:= rec.CktCtpIfIndex;
        else
            finalifindex := finalifindex||','||rec.CktCtpIfIndex;
        end if;
    end loop;

    return finalifindex;

END collapsedifindex;
```

3.1.5 CollapsedIfIndex

```
CREATE or replace function collapsedifindex (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktobjectindexcursor is
    select CktCtpIfIndex
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type = 2
    and CktCtpIfIndex is not null;
    finalifindex varchar2(10000);
BEGIN

    for rec in cktobjectindexcursor loop
        if cktobjectindexcursor%ROWCOUNT = 1 then
            finalifindex:= rec.CktCtpIfIndex;
        else
            finalifindex := finalifindex||','||rec.CktCtpIfIndex;
        end if;
    end loop;

    return finalifindex;

END collapsedifindex;
```

3.1.6 CollapsedModelType

```
CREATE or replace function collapsedmodeltype (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktmodeltypecursor is
    select CKTCTPMODELTYPE
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and CTP_TYPE = 2
    and CKTCTPMODELTYPE is not null;

    finalmoduletype varchar2(10000);
BEGIN

    for rec in cktmodeltypecursor loop
        if cktmodeltypecursor%ROWCOUNT = 1 then
            finalmoduletype:= rec.CKTCTPMODELTYPE;
        else
            finalmoduletype := finalmoduletype||','||rec.CKTCTPMODELTYPE;
        end if;
    end loop;

    return finalmoduletype;

END collapsedmodeltype;
```

3.1.7 CollapsedModuleName

```
CREATE or replace function collapsedmodulename (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktmodulenamecursor is
```

```
    select CKTCTPMODULETYPE
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type = 2
    and CKTCTPMODULETYPE is not null;
    finalcktdestModuleType varchar2(10000);
    slotmoduletype number;

BEGIN

    for rec in cktmodulenamecursor loop
    SELECT trunc(rec.CKTCTPMODULETYPE/power(2,16)) INTO slotmoduletype FROM dual;
        if cktmodulenamecursor%ROWCOUNT = 1 then
            finalcktdestModuleType:= slotmoduletype;
        else
            finalcktdestModuleType := finalcktdestModuleType||','||slotmoduletype;
        end if;
    end loop;

    return finalcktdestModuleType;

END collapsedmodulename;
```

3.1.8 CollapsedObjectIndex

```
CREATE or replace function collapsedobjectindex(CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktifindexcursor is
    select CktCtpObjectType
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and CTP_TYPE = 2;
    finalifindex varchar2(10000);
BEGIN

    for rec in cktifindexcursor loop
        if cktifindexcursor%ROWCOUNT = 1 then
            finalifindex:= rec.CktCtpObjectType;
        else
            finalifindex := finalifindex||','||rec.CktCtpObjectType;
        end if;
    end loop;

    return finalifindex;

END;
```

3.1.9 CollapsedPhy

```
CREATE or replace function collapsedphy (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    cursor cktphysicalloc_cursor is
    select CKTCTPPHYSICALLOC
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type = 2
    and CKTCTPPHYSICALLOC is not null;
    finalphy varchar2(10000);
BEGIN

    for rec in cktphysicalloc_cursor loop
        if cktphysicalloc_cursor%ROWCOUNT = 1 then
            finalphy:= rec.CKTCTPPHYSICALLOC;
        else
            finalphy := finalphy||','||rec.CKTCTPPHYSICALLOC;
        end if;
```

```
    end loop;

    return finalphy;

END collapsedphy;
```

3.1.10 CollapsedPort

```
CREATE or replace function collapsedport(CKTNODEIDIN varchar,CKTUNIQUEIDIN varchar) return
varchar2
is
cursor cktdestcursor is
select cktdestportname
from circuit_dest_tbl
where CKTNODEID=CKTNODEIDIN
and CKTUNIQUEID =CKTUNIQUEIDIN
and cktdestportname is not null;
finaldest varchar2(10000);
BEGIN

for rec in cktdestcursor loop
   if cktdestcursor%ROWCOUNT = 1 then
    finaldest:= rec.cktdestportname;
  else
    finaldest := finaldest||','||rec.cktdestportname;
  end if;
end loop;
return finaldest;
END;
```

3.1.11 CollapsedPortModuleName

```
CREATE or replace function collapsedportmodulename(CKTNODEIDIN varchar,CKTUNIQUEIDIN
varchar) return varchar2
is
cursor cktmodulenamecursor is
select cktctptModuleType
from circuit_ctp_tbl
where CKTNODEID=CKTNODEIDIN
and CKTUNIQUEID =CKTUNIQUEIDIN
and ctp_type = 2
and cktctptModuleType is not null;
finalcktdestModuleType varchar2(10000);
portmoduletype number;
BEGIN

for rec in cktmodulenamecursor loop
   SELECT trunc(mod(rec.cktctptModuleType,power(2,16))) INTO portmoduletype FROM dual;
   if cktmodulenamecursor%ROWCOUNT = 1 then
    finalcktdestModuleType:= portmoduletype;
  else
    finalcktdestModuleType := finalcktdestModuleType||','||portmoduletype;
  end if;
end loop;
return finalcktdestModuleType;
END;
```

3.1.12 ConvertAxxIfIndex

```
CREATE OR REPLACE function convertaxxifindex(ifindex in number)
RETURN number
IS
newifindex NUMBER ;
firstByte INTEGER ;
secondByte INTEGER ;
thirdByte INTEGER ;
forthByte INTEGER ;
fifthByte INTEGER ;
result number(20);
BEGIN
```

```
if ifindex = 0 then
    return 0;
else
    newifindex := ifindex - 1;
end if;

fifthByte := mod(newifindex, 4);
newifindex := floor(newifindex / 4);
forthByte := mod(newifindex, 8);
newifindex := floor(newifindex / 8);
thirdByte := mod(newifindex, 4);
newifindex := floor(newifindex / 4);
--secondByte := mod(newifindex, 4) + 1;
secondByte := mod(newifindex, 3) + 1;
newifindex := floor(newifindex / 3);
--firstByte := mod(newifindex, 4) + 1;
firstByte := mod(newifindex, 3) + 1;

result := 0;
result := result + (firstByte * POWER(2,56));
result := result + (secondByte * POWER(2,48));
result := result + (thirdByte * POWER(2,40));
result := result + (forthByte * POWER(2,32));
result := result + (fifthByte * POWER(2,24));

RETURN result;
END convertaxxifindex;
```

3.1.13 ConvertLinkModuleType

```
CREATE OR REPLACE function convertLinkModuleType (aModuleType number, aModelType number)
RETURN number
AS
    finalModuleType number(8);
BEGIN
    if (aModelType in (3,14,5,24,18,30)) then
        if(aModuleType < 0 or aModuleType > 2000) then
            finalModuleType := aModuleType;
        else
            finalModuleType := aModuleType * power(2, 16);
        end if;
    else
        finalModuleType := aModuleType;
    end if;

    return finalModuleType;
END;
```

3.1.14 CountCoSPh

```
CREATE OR REPLACE FUNCTION countCosPh(cosPhloc IN NUMBER)
RETURN NUMBER
IS CURSOR cntPh
IS SELECT UNIQUEID FROM TOPOLOGY_CIRCUIT_VIEW
WHERE SRCPHYSICALLOC=cosPhloc OR DESTPHYSICALLOC=cosPhloc;
BEGIN
FOR rec IN cntPh LOOP
IF cntPh%ROWCOUNT = 1 THEN       return 1;
ELSE       return 0;
END IF;   END LOOP;   return 0;
END countCosPh; / show errors;
```

3.1.15 CTMDuration

```
create or replace function ctmduration(p_ctmduration in number)
return varchar2
IS
       CTMDuration_time varchar2(50);
```

```
        dddd  varchar2(4);
        hh    number:=0;
        mm    number:=0;
        ss    number:=0;
        hh_c    varchar2(4);
        mm_c    varchar2(4);
        ss_c    varchar2(4);

BEGIN
        IF (p_ctmduration = 0 ) THEN
           CTMDuration_time:='';
        ELSE
           SELECT trunc(p_ctmduration) INTO dddd FROM dual;
           SELECT trunc((p_ctmduration-dddd)*24) INTO hh FROM dual;
           IF ( hh <10 and hh>0 ) THEN
              hh_c:='0'||hh||'h:';
           ELSIF (hh > 10) THEN
              hh_c:=hh||'h:';
           END IF;
           SELECT trunc((p_ctmduration-dddd)*24*60)-hh*60 INTO mm FROM dual;
           IF (mm <10 and mm>0) THEN
              mm_c:='0'||mm||'m:';
           ELSIF (mm>10) THEN
              mm_c:=mm||'m:';
           END IF;
           SELECT trunc((p_ctmduration-dddd)*24*60*60)-hh*60*60-mm*60 INTO ss from dual;
           IF (ss < 10 and ss>0 ) THEN
               ss_c:='0'||ss||'s';
           ELSIF (ss>10) THEN
               ss_c:=ss||'s';
           END IF;

           IF (dddd=0 or dddd is null) THEN
               CTMDuration_time:=rtrim(hh_c||mm_c||ss_c,':');
           ELSE
               CTMDuration_time:=rtrim(dddd||'d:'||hh_c||mm_c||ss_c,':');
           END IF;
        END IF;
return CTMDuration_time;
END ctmduration;
```

3.1.16 DisplayIf

The displayif function is used to translate neifindex.

```
CREATE or REPLACE function displayif (neifindex in number)
RETURN varchar2
IS

   iftype       number;
   if1          number;
   if2          number;
   o_if2        number;
   if3          number;
   o_if3        number;
   ifstring     varchar2(10);
   o_interface    varchar2(20);
   i_neifindex  number;

BEGIN

   IF neifindex < 0 THEN
      i_neifindex := neifindex+power(2,32);
   ELSE
      i_neifindex:=neifindex;
   END IF;

   SELECT trunc(i_neifindex/power(2,24)) INTO iftype FROM dual;

   SELECT decode(iftype,1,'System',
                        2,'Equipment',
                        3,'BITS',
                        4,'Sync-NE',
```

```
                    5,'Env Alarm',
                    6,'SDCC',
                    7,'Env CTL',
                    8,'Chassis',
                    9,'Ext-Sync',
                    10,'AIP',
                    11,'FAN',
                    12,'Port',
                    13,'CRS',
                    14,'RSMS',
                    15,'STM1E',
                    20,'EC-N',
                    21,'OC-N',
                    30,'DS-0',
                    31,'DS-1',
                    32,'DS-2',
                    33,'DS-3',
                    35,'STSRNG',
                    36,'AICI_AEP',
                    37,'AICI_AIE',
                    38,'UDC_A',
                    39,'UDC_B',
                    40,'VT-1',
                    45,'STS',
                    47,'UCP_CKT',
                    48,'UCP_IPCC',
                    49,'UCP_NBR',
                    55,'Ethernet',
                    56,'Ethernet-G',
                    57,'Ethernet-Brg',
                    58,'Ethernet-gige',
                    60,'PPP',
                    63,'E4',
                    65,'ATM',
                    66,'AAL5',
                    70,'VIDEO',
                    75,'XTC',
                    76,'CAP',
                    77,'CHANNEL',
                    80,'STM',
                    81,'E1',
                    82,'E3',
                    83,'DS3I',
                    90,'GENERIC',
                    100,'PIRELLI',
                    101,'STS',
                    102,'STS1',
                    103,'STS3',
                    104,'STS6',
                    105,'STS9',
                    106,'STS12',
                    107,'STS24',
                    108,'STS48',
                    109,'STS192',
                    110,'STS254',
                    111,'VC4-2C',
                    112,'VC4-3C',
                    113,'VC4-4C',
                    114,'VC4-8C',
                    115,'VC4-16C',
                    116,'VC4-64C',
                    117,'VC11',
                    118,'VC4',
                    119,'VC3',
                    120,'VC12',
                    201,'SER-DS1',
                    202,'E1-Port',
                    203,'SER-E1',
                    204,'STM1',
                    205,'VC4',
                    206,'VC3',
                    207,'VC2',
                    208,'VC12'
                    )
INTO ifstring FROM dual;
```

```
     IF (ifstring='Equipment') THEN
         o_interface:= 'Equipment';
     ELSE
         SELECT trunc(mod(i_neifindex,power(2,8))) INTO if3 FROM dual;
         SELECT decode(if3,0,'',if3) INTO o_if3 FROM dual;
         SELECT mod(trunc(i_neifindex/power(2,8)),power(2,8))  INTO if2 FROM dual;
         SELECT decode(if2,0,'',if2) INTO o_if2 FROM dual;
         SELECT mod(trunc(i_neifindex/power(2,16)),power(2,8)) INTO if1 FROM dual;

         IF ( if1 != 0 ) THEN
             o_interface := ifstring||' #'||if1||' '||o_if2||' '||o_if3;
         ELSE
             o_interface := ifstring||' '||o_if2||' '||o_if3;
         END IF;
     END IF;

RETURN o_interface;

END displayif;
```

3.1.17 DisplayIPAll

```
CREATE OR REPLACE
function  displayIpAll (NEIpaddr6 in raw, NEIpaddr in number, ipv6 in number)
return varchar2
IS

       first_sec    number;
       second_sec   number;
       third_sec    number;
       fourth_sec   number;
       ipaddr       varchar2(100);
       neipaddr_o   number;
       isipv6       number;
BEGIN
--dbms_output.put_line(neipaddr);

if ipv6 = 1 then
   select rawtohex(NEIpaddr6) into ipaddr from dual;
else
  if neipaddr < 0 then
     neipaddr_o:=NEIpaddr+power(2,32);
  else
     neipaddr_o:=NEIpaddr;
  end if;

  SELECT trunc(mod(NEIPaddr_o/power(2,24), power(2,8))) into first_sec from dual;
  SELECT trunc(mod(NEIPaddr_o/power(2,16), power(2,8))) INTO  second_sec from dual;
  SELECT trunc(mod(NEIPaddr_o/power(2,8),power(2,8))) INTO third_sec from dual;
  SELECT trunc(mod(NEIPaddr_o,power(2,8))) INTO fourth_sec from dual;

  ipaddr:=first_sec||'.'||second_sec||'.'||third_sec||'.'||fourth_sec;
end if;
RETURN ipaddr;
END displayIpAll;
```

3.1.18 DisplayIP

```
Create or replace function  displayIp (NEIpaddr in number)
return varchar2
IS

       first_sec    number;
       second_sec   number;
       third_sec    number;
       fourth_sec   number;
       ipaddr       varchar2(100);
       neipaddr_o   number;
    isipv6  number;
BEGIN
```

```
--dbms_output.put_line(neipaddr);

if neipaddr < 0 then
   neipaddr_o:=neipaddr+power(2,32);
else
   neipaddr_o:=neipaddr;
end if;

SELECT trunc(mod(NEIPaddr_o/power(2,24), power(2,8))) into first_sec from dual;
SELECT trunc(mod(NEIPaddr_o/power(2,16), power(2,8))) INTO  second_sec from dual;
SELECT trunc(mod(NEIPaddr_o/power(2,8),power(2,8))) INTO third_sec from dual;
SELECT trunc(mod(NEIPaddr_o,power(2,8))) INTO fourth_sec from dual;

ipaddr:=first_sec||'.'||second_sec||'.'||third_sec||'.'||fourth_sec;

RETURN ipaddr;
END displayip;

/
rem
rem return the string of the IP address in the two format ipv4 or ipv6
rem
```

3.1.19 DisplayPhyLoc

The displayphloc function is used to determine the physical location.

```
Create or replace function  displayphyloc (physicalloc in number)
return varchar2
IS
    o_physicalloc     varchar2(40);
    nemodelindex      number;
    port              number;
    slot              number;
    pim               number;
    ppm               number;
    rack              number;
    shelf             number;
    nemodel           varchar2(30);
BEGIN

SELECT trunc(mod(physicalloc,power(2,10))) INTO  port from dual;
SELECT mod(trunc(physicalloc/power(2,10)),power(2,6)) INTO ppm from dual;
SELECT mod(trunc(physicalloc/power(2,16)),power(2,8)) INTO pim from dual;
SELECT mod(trunc(physicalloc/power(2,24)),power(2,8)) INTO slot from dual;
SELECT mod(trunc(physicalloc/power(2,32)),power(2,8)) INTO shelf from dual;
SELECT mod(trunc(physicalloc/power(2,40)),power(2,8)) INTO rack from dual;

o_physicalloc :='Rack:'||rack||',Shelf:'||shelf||',Slot:'||slot||',Port:'||port;

RETURN o_physicalloc;
END displayphyloc;
```

3.1.20 EqptInfoPort

```
CREATE or replace function eqptinfoport(NEDBACCESSID_param number,
 PHYSICALLOC_param number,
 IFINDEX_param     number,
 MODULEINDEX_param  number
) return varchar2
is
 portnameselectedcol200 varchar2(4000);
 portnameselectedcol3 varchar2(4000);
 calc_OBJECTINDEX number;
 PHYSICALLOC_temp number;
 NEMODELTYPE_param number;
 NEMODELINDEX_param number;
 calc_Ifindex number(20);
 SlotModuleType number;
 PortModuleType number;
 PpmModuleType number;
```

```
BEGIN
  begin
  select
        NEMODELTYPE,
        NEMODELINDEX
  into
        NEMODELTYPE_param,
        NEMODELINDEX_param
   from ne_info_table
  where nedbaccessid = NEDBACCESSID_param;
  exception
    when no_data_found then
     null;
 end;
 SlotModuleType:= getSlotModuleType(MODULEINDEX_param);
 SELECT trunc(mod(MODULEINDEX_param,power(2,16))) INTO PortModuleType FROM dual;
 PpmModuleType:= getPpm(PHYSICALLOC_param);

 --create_file('/oraclesw9i','testf.log', 'NEMODELTYPE_param
:'||NEMODELTYPE_param||'NEMODELINDEX_param :'||NEMODELINDEX_param);

 if (SlotModuleType = 157 or SlotModuleType = 146 or SlotModuleType = 232 or
SlotModuleType = 233 or SlotModuleType = 234) then
     calc_OBJECTINDEX:= mapobjectIndex(neModelType_param , nemodelIndex_param ,
getport(PHYSICALLOC_param) ,0, PortModuleType);
 else
     if (PpmModuleType = 63) then
     -- ppm not used;
         calc_OBJECTINDEX:= mapobjectIndex(neModelType_param , nemodelIndex_param ,
getport(PHYSICALLOC_param) ,0, SlotModuleType);
     else
         calc_OBJECTINDEX:= mapobjectIndex(neModelType_param , nemodelIndex_param ,
PpmModuleType ,0, SlotModuleType);
     end if;

 -- append_file('/oraclesw9i', 'testf.log', 'calc_OBJECTINDEX: '||calc_OBJECTINDEX);
 --  append_file('/oraclesw9i','testf.log', 'PHYSICALLOC_param:'||PHYSICALLOC_param);
  -- append_file('/oraclesw9i','testf.log','calc_Ifindex: '||calc_Ifindex);

 PHYSICALLOC_temp :=  PHYSICALLOC_param;
 if  ( SlotModuleType = 47 ) then
     PHYSICALLOC_temp := getPort_xtc(PHYSICALLOC_param, SlotModuleType);
end if;

  if (NEMODELTYPE_param = 38) then
   if     (SlotModuleType = 233) then
      if ( calc_OBJECTINDEX = 341 or calc_OBJECTINDEX = 342) then
        PHYSICALLOC_temp := PHYSICALLOC_temp - 28;
      end if;
   elsif  (SlotModuleType = 234) then
      if ( calc_OBJECTINDEX = 341 or calc_OBJECTINDEX = 342) then
        PHYSICALLOC_temp := PHYSICALLOC_temp - 84;
      end if;
   end if;
 end if;

        begin
                select col200,col3 into portnameselectedcol200,portnameselectedcol3
                from eqpt_info_table
                where NEDBACCESSID = NEDBACCESSID_param
                and   PHYSICALLOC      = PHYSICALLOC_temp
                and   IFINDEX          = 0
                and   MODULETYPE       = SlotModuleType
                and   OBJECTINDEX      = calc_OBJECTINDEX;
        exception
                when no_data_found then
                   null;
       end;

       if (calc_OBJECTINDEX = 193 or calc_OBJECTINDEX = 194) then
          return portnameselectedcol3;
       end if;

       return portnameselectedcol200;
END;
/
```

3.1.21 Get_Enet_Value

```
CREATE OR REPLACE function get_enet_value(
   param1  IN  number,
   param2  IN  number,
   param3  IN  number default -4,
   param4  IN  number default -4,
   param5  IN  number default -4,
   param6  IN  number default -4)
return number
IS
       r_value number  ;
       val1 number  ;
       val2 number  ;
       val3 number  ;
       val4 number  ;
       val5 number  ;
       val6 number  ;
BEGIN

     IF(param3 =-4 and param4 =-4 and param5 =-4 and param6 =-4) THEN
        IF(param1 < 0 and param2 < 0) THEN
            IF ( (param1=param2)) THEN
                    r_value:=param1;
                ELSIF (param1 = -3 or param2 = -3)THEN
            r_value:= -3;
        ELSIF ( param1 = -2 or param2 = -2 ) THEN
                    r_value:= -2;
            ELSIF ( param1 = -1 or param2 = -1 ) THEN
                    r_value:= -1;
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 =-4) and (param5 =-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0) THEN
            IF ( (param1=param2) and (param2=param3) ) THEN
                    r_value:=param1;
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3)THEN
            r_value:= -3;
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 ) THEN
                    r_value:= -2;
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 ) THEN
                    r_value:= -1;
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 =-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0) THEN
            IF ( ((param1=param2) and (param2=param3) and (param3=param4))) THEN
                    r_value:=param1;
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3)THEN
            r_value:= -3;
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 ) THEN
                    r_value:=-2;
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 ) THEN
                    r_value:=-1;
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 >-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0 and param5 < 0) THEN
            IF ( ((param1=param2) and (param2=param3) and (param3=param4) and
(param4=param5))) THEN
                    r_value:=param1;
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3 or param5
=-3)THEN
            r_value:= -3;
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 or param5 = -2)
THEN
                    r_value:=-2;
```

```
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 or param5 = -
1) THEN
                    r_value:=-1;
                END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 >-4) and (param6 >-4)) THEN
    IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0 and param5 < 0 and param6 <
0) THEN
        IF ( ((param1=param2) and (param2=param3) and (param3=param4) and (param4=param5)
and (param5=param6))) THEN
                    r_value:=param1;
                ELSIF(param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3 or param5 =-
3 or param6=-3) THEN
            r_value:= -3;
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 or param5 = -2 or
param6 = -2) THEN
                    r_value:=-2;
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 or param5 = -
1 or param6 = -1) THEN
                    r_value:=-1;
            END IF;
            return r_value;
    END IF;

     END IF;



     IF ( param1 <0 ) THEN
     val1:= 0;
     ELSE
     val1:= param1;
     END IF;
     IF ( param2 <0 ) THEN
     val2:= 0;
     ELSE
     val2:= param2;
     END IF;
     IF ( param3 <0 ) THEN
     val3:= 0;
     ELSE
     val3:= param3;
     END IF;
     IF ( param4 <0 ) THEN
     val4:= 0;
     ELSE
     val4:= param4;
     END IF;
     IF ( param5 <0 ) THEN
     val5:= 0;
     ELSE
     val5:= param5;
     END IF;
     IF ( param6 <0 ) THEN
     val6:= 0;
     ELSE
     val6:= param6;
     END IF;

     r_value:=val1+val2+val3+val4+val5+val6;


return r_value;
END get_enet_value;
```

3.1.22 Get_Enet_Value2

```
CREATE OR REPLACE function get_enet_value2(
   param1  IN  number,
   param2  IN  number,
   param3  IN  number default -4,
```

```
   param4  IN  number default -4,
   param5  IN  number default -4,
   param6  IN  number default -4)
return varchar
IS
       r_value varchar(25)  ;
       val1 number  ;
       val2 number  ;
       val3 number  ;
       val4 number  ;
       val5 number  ;
       val6 number  ;
BEGIN

     IF(param3 =-4 and param4 =-4 and param5 =-4 and param6 =-4) THEN
        IF(param1 < 0 and param2 < 0) THEN
            IF ( (param1=param2)) THEN
                r_value:=to_char(param1);
                ELSIF (param1 = -3 or param2 = -3)THEN
            r_value:=to_char(-3);
        ELSIF ( param1 = -2 or param2 = -2 ) THEN
                    r_value:=to_char(-2);
            ELSIF ( param1 = -1 or param2 = -1 ) THEN
                    r_value:=to_char(-1);
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 =-4) and (param5 =-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0) THEN
            IF ( (param1=param2) and (param2=param3) ) THEN
                    r_value:=to_char(param1);
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3)THEN
            r_value:=to_char(-3);
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 ) THEN
                    r_value:=to_char(-2);
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 ) THEN
                    r_value:=to_char(-1);
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 =-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0) THEN
            IF ( ((param1=param2) and (param2=param3) and (param3=param4))) THEN
                    r_value:=to_char(param1);
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3)THEN
            r_value:=to_char(-3);
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 ) THEN
                    r_value:=to_char(-2);
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 ) THEN
                    r_value:=to_char(-1);
            END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 >-4) and (param6 =-4)) THEN
        IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0 and param5 < 0) THEN
            IF ( ((param1=param2) and (param2=param3) and (param3=param4) and
(param4=param5))) THEN
                    r_value:=to_char(param1);
                ELSIF (param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3 or param5
=-3)THEN
            r_value:=to_char(-3);
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 or param5 = -2)
THEN
                    r_value:=to_char(-2);
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 or param5 = -
1) THEN
                    r_value:=to_char(-1);
                END IF;
            return r_value;
        END IF;

     ELSIF((param3 > -4) and (param4 > -4) and (param5 >-4) and (param6 >-4)) THEN
```

```
    IF(param1 < 0 and param2 < 0 and param3 < 0 and param4 < 0 and param5 < 0 and param6 <
0) THEN
        IF ( ((param1=param2) and (param2=param3) and (param3=param4) and (param4=param5)
and (param5=param6))) THEN
                    r_value:=to_char(param1);
                ELSIF(param1 = -3 or param2 = -3 or param3 = -3 or param4 = -3 or param5 =-
3 or param6=-3) THEN
            r_value:=to_char(-3);
        ELSIF ( param1 = -2 or param2 = -2 or param3 = -2 or param4 = -2 or param5 = -2 or
param6 = -2) THEN
                    r_value:=to_char(-2);
            ELSIF ( param1 = -1 or param2 = -1 or param3 = -1 or param4 = -1 or param5 = -
1 or param6 = -1) THEN
                    r_value:=to_char(-1);
            END IF;
            return r_value;
    END IF;

     END IF;



     IF ( param1 <0 ) THEN
     val1:= 0;
     ELSE
     val1:= param1;
     END IF;
     IF ( param2 <0 ) THEN
     val2:= 0;
     ELSE
     val2:= param2;
     END IF;
     IF ( param3 <0 ) THEN
     val3:= 0;
     ELSE
     val3:= param3;
     END IF;
     IF ( param4 <0 ) THEN
     val4:= 0;
     ELSE
     val4:= param4;
     END IF;
     IF ( param5 <0 ) THEN
     val5:= 0;
     ELSE
     val5:= param5;
     END IF;
     IF ( param6 <0 ) THEN
     val6:= 0;
     ELSE
     val6:= param6;
     END IF;

     r_value:=to_char(val1+val2+val3+val4+val5+val6);


return r_value;
END get_enet_value2;
```

3.1.23 GetBccBwInfo

```
CREATE OR REPLACE FUNCTION getBccBwInfo(neId IN NUMBER, phLoc IN NUMBER, objIdx IN NUMBER,
COLUMN IN VARCHAR2)
RETURN VARCHAR2
IS
       retVal  VARCHAR2(256);
BEGIN
       SELECT COL16 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND COL2 = COLUMN;
       RETURN retVal;
END getBccBwInfo;
```

3.1.24 GetBccClassInfo

```
CREATE OR REPLACE FUNCTION getBccClassInfo(neId IN NUMBER, phLoc IN NUMBER, objIdx IN
NUMBER, COLUMN IN VARCHAR2)
RETURN VARCHAR2
IS
       retVal  VARCHAR2(256);
BEGIN
       SELECT COL19 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND COL2 = COLUMN;
       RETURN retVal;
END getBccClassInfo;
```

3.1.25 GetBccClassRPR80217

```
CREATE OR REPLACE FUNCTION getBccClassRPR80217(neId IN NUMBER, phLoc IN NUMBER, objIdx IN
NUMBER, COLUMN IN NUMBER)
RETURN NUMBER
IS
       retVal  NUMBER(3);
BEGIN
    IF (COLUMN = 1) THEN -- Select column called COL1
      SELECT COL1 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND IFINDEX = 0;
    END IF;
    IF (COLUMN = 2) THEN -- Select column called COL2
      SELECT COL2 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND IFINDEX = 0;
    END IF;
    IF (COLUMN = 3) THEN -- Select column called COL3
      SELECT COL3 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND IFINDEX = 0;
    END IF;
    RETURN retVal;

END getBccClassRPR80217;
```

3.1.26 GetBccCosCommitGrp

```
CREATE OR REPLACE FUNCTION getBccCosCommitGrp(neId IN NUMBER, phLoc IN NUMBER, infoNum IN
NUMBER, objIdx IN NUMBER)
RETURN VARCHAR2
IS
       retVal  VARCHAR2(256);
       ifIdx   NUMBER(20);
BEGIN
     IF infoNum = 0 THEN -- cos
       SELECT COL2 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId=neId AND
ObjectIndex=objIdx AND
PHYSICALLOC=phLoC AND COL2 IS NOT NULL;
     END IF;
     IF infoNum = 11 THEN -- Multicast Group 1
       ifIdx := getIfCosGrp(neId, phLoc, 1);
       SELECT COL4 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId=neId AND
ObjectIndex=objIdx AND
PHYSICALLOC=phLoC AND IFINDEX=ifIdx AND COL4 IS NOT NULL;
     END IF;
     IF infoNum = 12 THEN -- Class Off service Group 1
       ifIdx := getIfCosGrp(neId, phLoc, 1);
       SELECT COL3 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId=neId AND
ObjectIndex=objIdx AND
PHYSICALLOC=phLoC AND IFINDEX=ifIdx AND COL3 IS NOT NULL;
     END IF;
     IF infoNum = 21 THEN -- Multicast Group 2
       ifIdx := getIfCosGrp(neId, phLoc, 2);
```

```
       SELECT COL4 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId=neId AND
ObjectIndex=objIdx AND
PHYSICALLOC=phLoC AND IFINDEX=ifIdx AND COL4 IS NOT NULL;
     END IF;
     IF infoNum = 22 THEN -- Class Off service Group 2
       ifIdx := getIfCosGrp(neId, phLoc, 2);
       SELECT COL3 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId=neId AND
ObjectIndex=objIdx AND
PHYSICALLOC=phLoC AND IFINDEX=ifIdx AND COL3 IS NOT NULL;
     END IF;
     return retVal;

END getBccCosCommitGrp;
```

3.1.27 GetBccCosInfo

```
CREATE OR REPLACE FUNCTION getBccCosInfo(neId IN NUMBER, phLoc IN NUMBER, objIdx IN
NUMBER, COLUMN IN VARCHAR2)
RETURN VARCHAR2
IS
       retVal  VARCHAR2(256);
BEGIN
       SELECT COL26 INTO retVal FROM EQPT_INFO_TABLE WHERE NeDbAccessId = neId AND
ObjectIndex = objIdx AND
PHYSICALLOC = phLoc AND COL2 = COLUMN;
       RETURN retVal;
END getBccCosInfo;
```

3.1.28 GetBGFromBGObj

```
Create or replace function  getBGFromBGObj(neifindex in number)
return number
IS
if1 number;
i_neifindex number;

BEGIN

IF neifindex < 0 THEN
i_neifindex := neifindex+power(2,32);
ELSE
i_neifindex:=neifindex;
END IF;

SELECT mod(trunc(i_neifindex/power(2,16)),power(2,8)) INTO if1 FROM dual;

RETURN if1;
END getBGFromBGObj;
```

3.1.29 GetBGFromQoSObj

```
Create or replace function  getBGFromQoSObj (neifindex in number)
return number
IS
if2 number;
o_if2 number;
i_neifindex number;
BEGIN

IF neifindex < 0 THEN
i_neifindex := neifindex+power(2,32);
ELSE
i_neifindex:=neifindex;
END IF;

SELECT mod(trunc(i_neifindex/power(2,8)),power(2,8)) INTO if2 FROM dual;
SELECT decode(if2,0,'',if2) INTO o_if2 FROM dual;

RETURN o_if2;
```

```
END getBGFromQoSObj;
```

3.1.30 GetCktDestNodeId

```
CREATE or replace function getCktDestNodeId (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return number
IS
    cursor cktnodeidcursor is
    select CktCtpNodeId
    from circuit_ctp_tbl
    where CktNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 2;

    finalinodeids varchar2(10000);
BEGIN

    for rec in cktnodeidcursor loop
        if cktnodeidcursor%ROWCOUNT = 1 then
            finalinodeids:= rec.CktCtpNodeId;
        else
            finalinodeids := finalinodeids||','||rec.CktCtpNodeId;
        end if;
    end loop;

    return finalinodeids;

END getCktDestNodeId;
```

3.1.31 GetCktSecSrcIfIndex

```
CREATE or replace function getCktSecSrcIfIndex (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return number
IS
    o_CktSecSrcIfIndex number;
BEGIN
    select CktCtpIfIndex into o_CktSecSrcIfIndex
    from circuit_ctp_tbl
    where CktCtpNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 1;

    return o_CktSecSrcIfIndex;

END getCktSecSrcIfIndex;
```

3.1.32 GetCktSecSrcPhysicalLoc

```
CREATE or replace function getCktSecSrcPhysicalLoc (CKTNODEIDIN number,CKTUNIQUEIDIN
number)return number
IS
    o_CktSecSrcPhysicalLoc number;
BEGIN
    select CktCtpPhysicalLoc into o_CktSecSrcPhysicalLoc
    from circuit_ctp_tbl
    where CktCtpNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 1;

    return o_CktSecSrcPhysicalLoc;

END getCktSecSrcPhysicalLoc;
```

3.1.33 GetCktSecSrcPortName

```
CREATE or replace function getCktSecSrcPortName (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
```

```
    o_CktSecSrcPortName varchar2(64);
BEGIN
    select CKTCTPPORTNAME into o_CktSecSrcPortName
    from circuit_ctp_tbl
    where CktNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 1;

    return o_CktSecSrcPortName;

END getCktSecSrcPortName;
/
```

3.1.34 GetCktSrcIfIndex

```
CREATE or replace function getCktSrcIfIndex (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return number
IS
    o_CktSrcIfIndex number;
BEGIN
    select CktCtpIfIndex into o_CktSrcIfIndex
    from circuit_ctp_tbl
    where CktCtpNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 0;

    return o_CktSrcIfIndex;

END getCktSrcIfIndex;
```

3.1.35 GetCktSrcPhysicalLoc

```
CREATE or replace function getCktSrcPhysicalLoc (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return number
IS
    o_CktSrcPhysicalLoc number;
BEGIN
    select CktCtpPhysicalLoc into o_CktSrcPhysicalLoc
    from circuit_ctp_tbl
    where CktCtpNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 0;

    return o_CktSrcPhysicalLoc;

END getCktSrcPhysicalLoc;
```

3.1.36 GetCktSrcPortName

```
CREATE or replace function getCktSrcPortName (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
IS
    o_CktSrcPortName varchar2(64);
BEGIN
    select CKTCTPPORTNAME into o_CktSrcPortName
    from circuit_ctp_tbl
    where CktNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 0;

    return o_CktSrcPortName;

END getCktSrcPortName;
```

3.1.37 GetCollapsedPortname

```
CREATE or replace function getcollapsedPortname(CKTNODEIDIN number,CKTUNIQUEIDIN number)
return varchar2
```

```
IS
    cursor eqptcollpasedport is
    select  CktNodeId,
            CktCtpPhysicalLoc,
            CktCtpIfIndex,
            CktCtpModuleType
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and CktNodeId is not null
    and CktCtpPhysicalLoc is not null
    and CktCtpIfIndex is not null
    and CktCtpModuleType is not null;

    finaleqptinfoport varchar2(10000);
    DestNedbaccessid varchar2(1000);
BEGIN

    for rec in eqptcollpasedport loop
        SELECT nedbaccessid INTO DestNedbaccessid
        FROM ne_info_table
        WHERE nenodeid= rec.CktNodeId and rec.CktNodeId != 0;

        if eqptcollpasedport%ROWCOUNT = 1 then
            finaleqptinfoport:=
'@'||eqptinfoport(DestNedbaccessid,rec.CktCtpPhysicalLoc,rec.CktCtpIfIndex,rec.CktCtpModul
eType);
        else
            finaleqptinfoport :=
finaleqptinfoport||','||'@'||eqptinfoport(DestNedbaccessid,rec.CktCtpPhysicalLoc,rec.CktCt
pIfIndex,rec.CktCtpModuleType);
        end if;
    end loop;

    return finaleqptinfoport;

END;
```

3.1.38 GetEqptInfoPhysicalLoc

```
Create or replace function getEqptInfoPhysicalLoc(physicalLoc in number,moduleIndex in
number)
return number
IS
    eqptphysicalLoc   number;
    eqptport          number;
    slot              number;
    port              number;
    pim               number;
    ppm               number;
    slotmoduletype    number;
    portmoduletype    number;
BEGIN
    slot:= getSlot(physicalLoc);
    port:= getPortWithPimPpm(physicalLoc);
    pim:= getPIM(physicalLoc);
    ppm:= getPPM(physicalLoc);
    slotmoduletype:=getSlotModuleType(moduleIndex);
    SELECT trunc(mod(moduleIndex,power(2,16))) INTO portmoduletype FROM dual;
    IF (pim = 0 AND slotmoduletype = 157 AND portmoduletype != 34 AND ppm = 0) THEN
        port:= port + 21;
        eqptphysicalLoc:= getPhysicalLoc(slot,port);
    ELSE
        eqptphysicalLoc:= physicalLoc;
    END IF;
RETURN eqptphysicalLoc;
END getEqptInfoPhysicalLoc;
```

3.1.39 GetIfCosGrp

```
CREATE OR REPLACE FUNCTION getIfCosGrp(neId IN NUMBER, phLoc IN NUMBER, first_last IN
NUMBER )
  RETURN NUMBER
```

```
IS
    CURSOR MCursor IS
    SELECT IFINDEX
    FROM EQPT_INFO_TABLE
    WHERE NEDBACCESSID=neId AND OBJECTINDEX=189 AND PHYSICALLOC=phLoc AND COL3 IS NOT
NULL;
    IfIndex NUMBER(20);
    find NUMBER(20);
BEGIN
    IF (first_last = 1) THEN -- find first
      find := 1;
    END IF;
    IF (first_last = 2) THEN -- find last
      find := 0;
    END IF;

    FOR rec IN MCursor LOOP
        IF MCursor%ROWCOUNT = first_last THEN
           IfIndex := rec.IFINDEX;
           return IfIndex;
        END IF;
        IfIndex := rec.IFINDEX;
    END LOOP;

    RETURN IfIndex;

END;
```

3.1.40 GetInterfaceId

```
create or replace function GetInterfaceId
       (p_nedbaccessid    IN number,
        p_moduleorIfIndex IN number,
        p_objectType     IN number,
        p_type             IN varchar2)
return varchar2
IS
       o_interfaceId  varchar2(64);
       o_modeltype    number:=0;
BEGIN
       IF (p_moduleOrIfindex = 0 ) THEN
        IF (p_objectType = 0 ) THEN
              select activevalue into o_interfaceId from ctm_config_table
             where sectionname='transportManager' and propertyname='ems-sys-id';
        ELSE
              SELECT nemodeltype INTO o_modeltype
              FROM ne_info_table
              WHERE nedbaccessid=p_nedbaccessid;
              IF (o_modeltype = 34) THEN
                 o_interfaceId:='N/A';
              ELSE
                 o_interfaceId:='0';
              END IF;
        END IF;
       END IF;

       IF (p_moduleOrIfindex = -1) THEN
          o_interfaceId:='N/A';
       END IF;

       IF ((p_moduleOrIfindex != 0 ) and (p_moduleOrIfindex != -1)) THEN
           IF (p_nedbaccessid =0 ) THEN
              IF (p_type like 'Maximum login attempts exceeded%') THEN
                BEGIN
                 SELECT username INTO o_interfaceId
                   FROM user_table
                  WHERE userid=p_moduleorIfIndex;
                 exception
                  when no_data_found then
                  o_interfaceId:= displayIp(p_moduleOrIfindex);
                END;
              ELSIF (p_type like 'Server Monitor Threshold Crossed%') THEN
                    select displayname into o_interfaceId
```

```
                    FROM SERVER_PARAMETER_TABLE
                   WHERE ParameterIndex=p_moduleorIfIndex;
              ELSE
                 SELECT nesysid INTO o_interfaceId
                   FROM ne_info_table
                  WHERE nedbaccessid=p_moduleorIfIndex AND nemodeltype!=34;
              END IF;
           ELSE
               SELECT nemodeltype INTO o_modeltype
               FROM ne_info_table
               WHERE nedbaccessid=p_nedbaccessid;

               IF (o_modeltype = 34) THEN
                 o_interfaceId:='N/A';
               ELSE
                  o_interfaceId:=p_moduleorIfIndex;
               END IF;
           END IF;
       END IF;
RETURN o_interfaceId;
END GetInterfaceId;
```

3.1.41 Get_Inventory_Note

```
CREATE OR REPLACE
FUNCTION "GET_INVENTORY_NOTE" (sn IN VARCHAR2) RETURN VARCHAR2
AS
    retVal  VARCHAR2(4000);
    cnt NUMBER;
BEGIN
  select count(*) into cnt from ONS15454_INVENTORY_NOTES where SERIALNUMBER = sn;
  if (cnt=0) THEN
    retVal:='';
  ELSE
    select NOTE into retVal from ONS15454_INVENTORY_NOTES where SERIALNUMBER = sn;
  END IF;
  RETURN retVal;
END GET_INVENTORY_NOTE;
```

3.1.42 GetLinkType

```
CREATE OR REPLACE function GetLinkType
       (p_circuitprot   IN number,
        p_circuittype IN number,
        p_linktype IN number,
        p_spanid IN number
       )
return number
IS
       o_linkcount     number:=0;
       o_linktype_18   number:=18;
       o_linktype_13   number:=13;
       o_linktype_4    number:=4;
       o_linktype_0    number:=0;
       o_linktype      number:=p_linktype;
BEGIN
IF ((p_circuitprot = 5) OR (p_circuitprot = 6)) THEN
              select count(*) into o_linkcount from
               CIRCUIT_CTP_TBL ctp1, CIRCUIT_CTP_TBL ctp2, CIRCUIT_SPAN_BS_TBL spn,
LINK_TABLE lnk, CIRCUIT_BS_TBL circuit
              where
               spn.CKTSPANSRCCTPACCESSID = ctp1.CTPACCESSID
               AND
               spn.CKTSPANDSTCTPACCESSID = ctp2.CTPACCESSID
               AND
               circuit.CKTNODEID = spn.CKTNODEID
               AND
               circuit.CKTUNIQUEID = spn.CKTUNIQUEID
               AND
               (
                ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE AND
```

```
                ctp2.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE )
                OR
               ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE AND
                ctp2.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE )
               )
               AND lnk.linktype in (0, 4)
               AND spn.CKTSPANBSID = p_spanid;
                   if ( o_linkcount > 1 ) then
                   o_linktype := o_linktype_4;
               else
                   o_linktype := o_linktype_0;
               end if;
       else
               if (p_linktype = 4 ) THEN
                   o_linktype := o_linktype_0;
               end if;
       END IF;
if ( p_circuittype = 38 ) THEN
              select count(*) into o_linkcount from
               CIRCUIT_CTP_TBL ctp1, CIRCUIT_CTP_TBL ctp2, CIRCUIT_SPAN_BS_TBL spn,
LINK_TABLE lnk, CIRCUIT_BS_TBL circuit
             where
              spn.CKTSPANSRCCTPACCESSID = ctp1.CTPACCESSID
              AND
              spn.CKTSPANDSTCTPACCESSID = ctp2.CTPACCESSID
              AND
              circuit.CKTNODEID = spn.CKTNODEID
              AND
              circuit.CKTUNIQUEID = spn.CKTUNIQUEID
              AND
              (
               ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE AND
                ctp2.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE )
                OR
               ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE AND
                ctp2.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE )
               )
               AND lnk.linktype = o_linktype_18
               AND spn.CKTSPANBSID = p_spanid;
                   if ( o_linkcount > 0 ) then
                   o_linktype := o_linktype_18;
               else
                   select count(*) into o_linkcount from
                      CIRCUIT_CTP_TBL ctp1, CIRCUIT_CTP_TBL ctp2, CIRCUIT_SPAN_BS_TBL spn,
LINK_TABLE lnk, CIRCUIT_BS_TBL circuit
                   where
                      spn.CKTSPANSRCCTPACCESSID = ctp1.CTPACCESSID
                   AND
                      spn.CKTSPANDSTCTPACCESSID = ctp2.CTPACCESSID
                   AND
                      circuit.CKTNODEID = spn.CKTNODEID
                   AND
                      circuit.CKTUNIQUEID = spn.CKTUNIQUEID
                   AND
                      (( ctp1.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE AND
                         ctp2.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE )
                        OR
                        ( ctp1.CKTCTPPHYSICALLOC = lnk.LINKDSTPHYSICALLOC AND
ctp1.CKTCTPNEDBACCESSID = lnk.LINKDSTNODE AND
                          ctp2.CKTCTPPHYSICALLOC = lnk.LINKSRCPHYSICALLOC AND
ctp2.CKTCTPNEDBACCESSID = lnk.LINKSRCNODE ))
                   AND lnk.linktype = o_linktype_13
                   AND spn.CKTSPANBSID = p_spanid;
                   if ( o_linkcount > 0 ) THEN
                      o_linktype := o_linktype_13;
                   end if;
               end if;
```

```
       END IF;
    return   o_linktype;
END GetLinkType;
```

3.1.43 GetLowBits

```
CREATE OR REPLACE function getLowBits (aNum number, numOfBits integer)
RETURN integer
AS
    result number;
    modVal number;
BEGIN
        modVal := power (2, numOfBits);
        result := mod (aNum, modVal);
        return result;
END;
```

3.1.44 GetMgx88xxMgmtState

```
create or replace function getMgx88xxMgmtState(p_isConnected in number)
return number
IS
      i_mstate      number;
BEGIN
      IF ( p_isConnected = 1 ) THEN
           i_mstate := 1;
      ELSE
           i_mstate := 2;
      END IF;

RETURN i_mstate;

END getMgx88xxMgmtState;
```

3.1.45 GetMgx88xxSyncMode

```
create or replace function getMgx88xxSyncMode(p_discoveryState in number)
return number
IS
      i_mode      number;
BEGIN
      IF ( p_discoveryState = 1 ) THEN
           i_mode := 5;
      ELSE IF ( p_discoveryState = 2 ) THEN
           i_mode := 1;
      ELSE IF ( p_discoveryState = 3 ) THEN
           i_mode := 2;
      ELSE IF ( p_discoveryState = 4 ) THEN
           i_mode := 3;
      ELSE
           i_mode := 1;
      END IF;
      END IF;
      END IF;
      END IF;

RETURN i_mode;

END getMgx88xxSyncMode;
```

3.1.46 GetMLcardAddnlInfo

This function indicates whether the ML card has been preprovisioned.

```
CREATE OR REPLACE function getMLcardAddnlInfo(AddnlInfo in varchar2, state in number)
```

```
RETURN varchar2
IS
BEGIN
       IF (state = 0) THEN
              RETURN 'EID-7077: ML Card was Pre-provisioned';
       END IF;
       RETURN AddnlInfo;
END getMLcardAddnlInfo;
```

3.1.47 GetModelTypeFromPhysicalLoc

```
CREATE OR REPLACE function getModelTypeFromPhysicalloc (aPHYSICALLOC number)
RETURN number
AS
    aModelType number(20);
BEGIN
    aModelType := removeLowBits(aPHYSICALLOC, 56);
    return aModelType;
END;
```

3.1.48 GetModuleName

```
create or replace function getModuleName(p_nedbaccessid in number, p_moduletype in number,
p_eventType in number)
return varchar2
IS
      o_modulename   varchar2(64);
      i_modeltype      number;
BEGIN
      IF ( p_nedbaccessid = 0 or ( p_eventType >= 4 AND  p_moduletype = 9999) ) THEN
           o_modulename := 'N/A';
      ELSE
           SELECT nemodeltype INTO i_modeltype
             FROM ne_info_table
            WHERE nedbaccessid=p_nedbaccessid;

           SELECT ModuleName INTO o_modulename
             FROM module_type_table
            WHERE moduletype=p_moduletype
              AND modeltype=i_modeltype;
      END IF;

RETURN o_modulename;

Exception
      WHEN NO_DATA_FOUND
      THEN
      RETURN 'Unsupported';
END getModuleName;
```

3.1.49 GetModuleName_PM

```
create or replace function getModuleName_pm(p_nenodeid in number, p_moduletype in number)
return varchar2
IS
      o_modulename   varchar2(64);
      i_modeltype      number;

BEGIN
        IF (p_nenodeid=0 OR p_moduletype=0) THEN
           o_modulename:='SYSTEM';
        ELSE

           SELECT nemodeltype INTO i_modeltype
             FROM ne_info_table
            WHERE nenodeid=p_nenodeid;

           SELECT ModuleName INTO o_modulename
             FROM module_type_table
            WHERE moduletype=p_moduletype
              AND modeltype=i_modeltype;
```

```
       END IF;

RETURN o_modulename;
Exception
      WHEN NO_DATA_FOUND
      THEN
      RETURN 'Unsupported';
END getModuleName_pm;
```

3.1.50 GetModuleName1

```
create or replace function getModuleName1(p_nenodeid in number, p_moduletype in number)
return varchar2
IS
      o_modulename   varchar2(64);
      i_modeltype      number;
      p_slotmoduletype number;

BEGIN
        SELECT trunc(p_moduletype/power(2,16)) INTO p_slotmoduletype FROM dual;
        IF (p_nenodeid=0 OR p_slotmoduletype=0) THEN
           o_modulename:='SYSTEM';
        ELSE

           SELECT nemodeltype INTO i_modeltype
             FROM ne_info_table
            WHERE nenodeid=p_nenodeid;

           SELECT ModuleName INTO o_modulename
             FROM module_type_table
            WHERE moduletype=p_slotmoduletype
              AND modeltype=i_modeltype;
       END IF;

RETURN o_modulename;
Exception
      WHEN NO_DATA_FOUND
      THEN
      RETURN 'Unsupported';
END getModuleName1;
```

3.1.51 GetModuleName2

```
create or replace function getModuleName2 (p_moduletype number, p_modeltype number)
return varchar2
IS
    o_modulename   varchar2(64);
    p_slotmoduletype number;
BEGIN
    SELECT trunc(p_moduletype/power(2,16)) INTO p_slotmoduletype FROM dual;
    IF (p_slotmoduletype=0) THEN
        o_modulename:='SYSTEM';
    ELSE
        SELECT ModuleName INTO o_modulename
        FROM module_type_table
        WHERE moduletype=p_slotmoduletype
          AND modeltype=p_modeltype;
    END IF;

RETURN o_modulename;
Exception
    WHEN NO_DATA_FOUND
    THEN
    RETURN 'Unsupported';

END getModuleName2;
```

3.1.52 GetgetProtectMLcardConfigState

--This function returns the config status of a protected card, return -1 if not found.

```
CREATE OR REPLACE function getProtectMLcardConfigState(NEDbId in number, Slot in number)
RETURN number
IS
   confStValue number;
BEGIN
       IF (NEDbId = 0) THEN
              RETURN -1;
       END IF;
       Select configState into confStValue from MCard_Table where NeDbaccessID = NEDbId
and SlotNumber = Slot;
       RETURN confStValue;
END getProtectMLcardConfigState;
```

3.1.53 GetNEDBAccessID

```
create or replace function getNEDBACCESSID(NodeId_param number)
return number
is
nedbaccessout number;
begin
  begin
    select
        nedbaccessid
            into nedbaccessout
        from
            ne_info_table
        where
            NENODEID = NodeId_param
            and NENODEID !=0;
  exception
        when no_data_found then
        null;
  end;
return nedbaccessout;

end;
/

show error;

rem
rem Function that return the IP address
rem
```

3.1.54 GetNEIPAddr

```
CREATE OR REPLACE function getNEIPAddr (p_NEDBAccessid number)
return raw
is
nedbIPaddr raw(16);
nedbIPaddrV4 raw(4);
c_nedbIPaddr number;
isipv6 number;
begin
  select is_ipv6 into isipv6 from ne_info_table where nedbaccessid = p_NEDBAccessid;
  if isipv6 = 1 then
     select neipv6addr into nedbIPaddr from ne_info_table where nedbaccessid =
p_NEDBAccessid;
     return nedbIPaddr;
  else
     select neipaddr into c_nedbIPaddr from ne_info_table where  nedbaccessid =
p_NEDBAccessid;
     -- dbms_output.put_line ('IPv4 =' || c_nedbIPaddr || 'IPv4 BIN =' ||
binary_utility.to_hex(c_nedbIPaddr));
     nedbIpaddrV4 := utl_raw.CAST_FROM_BINARY_INTEGER(c_nedbIPaddr);
     return nedbIPaddrV4;
```

```
  end if;

exception
      when no_data_found then
          return null;
end getNEIPAddr;
/
show error;

rem
rem Function that check if the IP is a IPv4 (return 0) or a IPv6 (return 1)
rem
```

3.1.55 GetObjectTypeFromIfIndex

```
CREATE OR REPLACE function getObjectTypeFromIfIndex (aIfindex number)
RETURN number
AS
    aObjectType number(10);
BEGIN
    if(sign(aIfindex) = -1) then
        aObjectType := removeLowBits(aIfindex+power(2,32), 24);
    else
        aObjectType := removeLowBits(aIfindex, 24);
    end if;

    return aObjectType;
END;
```

3.1.56 GetObjectTypeFromIfIndex2

```
CREATE OR REPLACE function getObjectTypeFromIfIndex2 ( amodelType number,aIfindex number)
RETURN number
AS
    aObjectType number(10);
BEGIN
    if(sign(aIfindex) = -1) then
        aObjectType := removeLowBits(aIfindex+power(2,32), 24);
    else
        aObjectType := removeLowBits(aIfindex, 24);
    end if;

    if (amodelType in (3, 14, 5, 29, 18, 30)) THEN
        select decode(aobjectType, 19, 5130,
                                   83, 5073,
                                       aobjectType+5000) into aobjectType from dual;
    end if;

    return aObjectType;
END;
```

3.1.57 GetOutIPAddr

```
CREATE OR REPLACE
function getOUTIPAddr (p_ossname varchar)
return raw
is
nedbIPaddr raw(16);
nedbIPaddrV4 raw(4);
c_nedbIPaddr number;
isipv6 number;
begin
  select is_ipv6 into isipv6 from oss_user_table where ossname = p_ossname;
  if isipv6 = 1 then
     select IPV6Addr into nedbIPaddr from oss_user_table where  ossname = p_ossname;
     return nedbIPaddr;
  else
```

```
     select IPAddress into c_nedbIPaddr from oss_user_table where   ossname = p_ossname;
     -- dbms_output.put_line ('IPv4 =' || c_nedbIPaddr || 'IPv4 BIN =' ||
binary_utility.to_hex(c_nedbIPaddr));
     nedbIpaddrV4 := utl_raw.CAST_FROM_BINARY_INTEGER(c_nedbIPaddr);
     return nedbIPaddrV4;
  end if;

exception
      when no_data_found then
          return null;
end getOUTIPAddr;
```

3.1.58 GetPhysicalLoc

```
Create or replace function getPhysicalLoc(slot in number,port in number)
return number
IS
    physicalLoc   number;
    physlot       number;
BEGIN

    physlot:= slot * power(2,16);
    physicalLoc:= physlot + port;

RETURN physicalLoc;
END getPhysicalLoc;
```

3.1.59 GetPhysicalLocWithPIMPPM

```
Create or replace function getPhysicalLocWithPimPpm(shelf in number, slot in number, pim
in number, ppm in number, port in number)
  return number
IS
     physicalLoc   number;
     phyrack       number;
     physhelf      number;
     physlot       number;
     phypim        number;
     phyppm        number;
BEGIN
      phyrack:= 1 * power(2,40);
      physhelf:= shelf * power(2,32);
      physlot:= slot * power(2,24);
      phypim:= pim * power(2,16);
      phyppm:= ppm * power(2,10);
      physicalLoc:= phyrack + physhelf + physlot + phypim + phyppm + port;

RETURN physicalLoc;
END getPhysicalLocWithPimPpm;
```

3.1.60 GetPIM

```
Create or replace function  getPIM (physicalloc in number)
return number
IS
    pim                       number;
BEGIN

SELECT trunc(mod(trunc(physicalloc/power(2,16)),power(2,8))) INTO  pim from dual;

RETURN pim;
END getPIM;
```

3.1.61 GetPMParameterValue

```
CREATE OR REPLACE function GetPMParameterValue
       (p_paramvalue    IN number,
```

```
        l_paramvalue    IN number
        )
return number
IS
       o_paramvalue  number;
BEGIN
       IF (p_paramvalue = -1 AND l_paramvalue <> -1) THEN
          o_paramvalue:= l_paramvalue;
       ELSIF (l_paramvalue = -1 AND p_paramvalue <> -1) THEN
          o_paramvalue:= p_paramvalue;
       ELSE
         o_paramvalue:= p_paramvalue;
       END IF;

RETURN o_paramvalue;
END GetPMParameterValue;
```

3.1.62 GetPolicyMapType

```
Create or replace function  getPolicyMapType(neifindex in number)
return number
IS
if1 number;
i_neifindex number;

BEGIN

IF neifindex < 0 THEN
i_neifindex := neifindex+power(2,32);
ELSE
i_neifindex:=neifindex;
END IF;

SELECT mod(trunc(i_neifindex/power(2,16)),power(2,8)) INTO if1 FROM dual;

RETURN if1;
END getPolicyMapType;
```

3.1.63 GetPort

```
Create or replace function  getPort (physicalloc in number)
return number
IS
    port                        number;
    ppm                         number;
BEGIN
    SELECT trunc(mod(physicalloc,power(2,10))) INTO  port from dual;

RETURN port;
END getPort;
```

3.1.64 GetPort_XTC

```
Create or replace function  getPort_xtc (physicalloc in number, moduletype_param  in
number )
return number
IS
    ppm                         number;
    port                        number;
    final_physicalloc           number;
    two_bytes                   number := power(2,16);
BEGIN

ppm := getPPM(physicalloc);
port := getPort(physicalloc);

if ( moduletype_param = 47 ) then
 -- append_file('/oraclesw9i', 'testfjune9.log', 'Port No. Passed is  : '|| port);
 if ( port <= 28 ) then
   port := 1;
 else
```

```
    port := mod (port, 28);
    if (port = 0) then
        port := 28;
    end if;
  end if;

 -- append_file('/oraclesw9i', 'testfjune9.log', 'Returning port no. : '|| port);
end if;

final_physicalloc := trunc(physicalloc/two_bytes) * two_bytes;
final_physicalloc := final_physicalloc + port;

RETURN final_physicalloc;
END getPort_xtc;
```

3.1.65 GetPortModuleName

```
create or replace function getPortModuleName(p_nenodeid in number, p_moduletype in number)
return varchar2
IS
      o_modulename   varchar2(64);
      i_modeltype      number;
      p_portmoduletype number;
BEGIN
        SELECT trunc(mod(p_moduletype,power(2,16))) INTO p_portmoduletype FROM dual;
        IF (p_nenodeid=0 OR p_portmoduletype=0) THEN
           o_modulename:='SYSTEM';
        ELSE
          SELECT nemodeltype INTO i_modeltype
             FROM ne_info_table
            WHERE nenodeid=p_nenodeid;

           SELECT ModuleName INTO o_modulename
             FROM module_type_table
            WHERE moduletype=p_portmoduletype
              AND modeltype=i_modeltype;
       END IF;
RETURN o_modulename;
Exception
    WHEN NO_DATA_FOUND
    THEN
    RETURN 'Unsupported';
END getPortModuleName;
```

3.1.66 GetPortModuleName2

```
create or replace function getPortModuleName2(p_moduleindex number, p_modeltype number)
return varchar2
IS
    o_modulename   varchar2(64);
    p_portmoduletype number;
BEGIN
    SELECT mod(p_moduleindex, power(2,16))  INTO p_portmoduletype FROM dual;
    IF (p_portmoduletype=0 ) THEN
        o_modulename:= getModuleName2(p_moduleindex, p_modeltype);
    ELSE
        SELECT ModuleName INTO o_modulename
        FROM module_type_table
        WHERE moduletype=p_portmoduletype
              AND modeltype=p_modeltype;
    END IF;

RETURN o_modulename;
Exception
    WHEN NO_DATA_FOUND
    THEN
    RETURN 'Unsupported';
END getPortModuleName2;
```

3.1.67 GetPortName

```
CREATE or REPLACE function GetPortName
          (p_nedbaccessid IN number,
           p_physicalloc IN number,
           p_ifindex IN number,
           p_moduletype IN number)
RETURN varchar2
IS
   iftype       number;
   i_ifindex    number;
   o_portname   varchar2(256);
   model_type  number;
   module_type  number;
   physical_loc    number;
   ppm     number;
   slot    number;
   port    number;
BEGIN

   IF p_ifindex < 0 THEN
      i_ifindex := p_ifindex+power(2,32);
   ELSE
      i_ifindex:=p_ifindex;
   END IF;
   module_type := p_moduletype;
   ppm := GetPpm(p_physicalloc);

   SELECT round(i_ifindex/power(2,24)) INTO iftype FROM dual;
   SELECT nemodeltype INTO model_type FROM ne_info_table WHERE nedbaccessid =
p_nedbaccessid;
   BEGIN

   IF ( iftype = 31 and p_moduletype=47 ) THEN    /* 31=ds1 47=XTC */
        SELECT col200 INTO o_portname
          FROM eqpt_info_table
         WHERE nedbaccessid=p_nedbaccessid
           AND physicalloc=p_physicalloc
           AND objectindex=204
           AND moduletype=p_moduletype;
   ELSIF (iftype = 33 and p_moduletype = 47 ) THEN /* 33=ds3 47=XTC */
        SELECT col200 INTO o_portname
          FROM eqpt_info_table
         WHERE nedbaccessid=p_nedbaccessid
           AND physicalloc=p_physicalloc
           AND objectindex=208
           AND moduletype=p_moduletype;
   ELSIF (p_moduletype = 34 ) THEN /* 34 = ds1 port on XTC or 15310*/
        SELECT col200 INTO o_portname
          FROM eqpt_info_table
         WHERE nedbaccessid=p_nedbaccessid
           AND physicalloc=p_physicalloc
           AND objectindex in (204,340) /* 204=Ds1Line 340=Ds1LineModel */
           AND moduletype in (47,233,234); /* 47=XTC 233=DS1_28_DS3_EC1_3
234=DS1_84_DS3_EC1_3 */
   ELSIF (p_moduletype = 32 ) THEN /* 32 = ds3 port on XTC or 15310*/
        SELECT col200 INTO o_portname
          FROM eqpt_info_table
         WHERE nedbaccessid=p_nedbaccessid
           AND physicalloc=p_physicalloc
           AND objectindex in (208,341) /* 208=Ds3Line 341=Ds3LineModel */
           AND moduletype in (47,233,234); /* 47=XTC 233=DS1_28_DS3_EC1_3
234=DS1_84_DS3_EC1_3 */
   ELSIF (p_moduletype = 102 or p_moduletype = 103 or p_moduletype = 163 or p_moduletype =
164 or p_moduletype = 187 ) THEN /* For ML series cards*/
        IF (p_ifindex = 1) THEN
             SELECT col200 INTO o_portname
               FROM eqpt_info_table
              WHERE nedbaccessid=p_nedbaccessid
                AND physicalloc=p_physicalloc
                AND objectindex in (136,257);
    ELSE
            SELECT col200 INTO o_portname
              FROM eqpt_info_table
```

```
              WHERE nedbaccessid=p_nedbaccessid
                AND physicalloc=p_physicalloc
                AND objectindex in (134,255);
    END IF;
   ELSE
        SELECT col200 INTO o_portname
          FROM eqpt_info_table
         WHERE nedbaccessid=p_nedbaccessid
           AND physicalloc=p_physicalloc
           AND moduleType=p_moduletype
           AND objectindex in
(106,109,114,115,121,122,123,125,134,136,141,142,143,147,169,184,185,193,196,198,204,206,2
08,209,216,252,254,255,257,267,271,281);
   END IF;
  EXCEPTION
       WHEN TOO_MANY_ROWS
       THEN null;
  END;

RETURN o_portname;

END GetPortName;
```

3.1.68 GetPortWithPIMPPM

```
Create or replace function  getPortWithPimPpm (physicalloc in number)
return number
IS
    port                        number;
BEGIN

SELECT trunc(mod(physicalloc,power(2,13))) INTO  port from dual;

RETURN port;
END getPortWithPimPpm;
```

3.1.69 GetPosPortState

This function takes the topology type, NE ID, slot, and port numbers and returns the POS port state.

```
CREATE OR REPLACE function getPosPortState(topoType in number, neId in number, slotNo in
number, portNo in number)
RETURN varchar2
IS
       state  varchar2(256);
BEGIN
      if (topoType !=3) then
       select COL5 into state from EQPT_INFO_TABLE where NeDbAccessId = neId and
ObjectIndex = 166 and getSlot(physicalLoc) = slotNo and        getPort(physicalLoc) =
portNo;
       return state;
     else
     state := 0;
     return state;

     end if;
END getPosPortState;
```

3.1.70 GetPosPortStateForRPR802

This function takes the topology type, NE ID, slot, and port numbers and returns the POS port state for 802.17 RPR topologies.

```
CREATE OR REPLACE function getPosPortStateForRPR802(topoType in number, neId in number,
slotNo in number, portNo in number)
```

```
RETURN varchar2
IS
       state  varchar2(256);
BEGIN
      if (topoType =3) then
       select COL5 into state from EQPT_INFO_TABLE where NeDbAccessId = neId and
ObjectIndex = 284 and getSlot(physicalLoc) = slotNo and
 IfIndex = portNo;
       return state;
     else
     state := 0;
     return state;

     end if;
END getPosPortStateForRPR802;
```

3.1.71 GetPPM

```
Create or replace function  getPPM (physicalloc in number)
return number
IS
    ppm                        number;
BEGIN

SELECT trunc(mod(trunc(physicalloc/power(2,10)),power(2,6))) INTO  ppm from dual;

RETURN ppm;
END getPPM;
```

3.1.72 GetProtectMLCardInfo

This function is used of the ML-series card is preprovisioned.

```
CREATE OR REPLACE function getProtectMLcardInfo(NEDbId in number, Slot in number)
RETURN varchar2
IS
   SysId        varchar2(256);
   CardModule   varchar2(256);
   ProtectCard  varchar2(256);
   equipType    number;
   model        number;
BEGIN
       IF (NEDbId = 0) THEN
              RETURN 'none';
       END IF;
       Select NESysID into SysId from Ne_Info_table where NeDbaccessID = NEDbId;
       Select NEModelType into model from Ne_Info_table where NeDbaccessID = NEDbId;
       Select EquipmentType into equipType from ONS15454_ne_inventory_table where
NeDbaccessID = NEDbId and getSlot(PhysicalLoc) = Slot;
       Select ModuleName into CardModule from module_type_table where moduleType =
equipType and modelType = model;
       ProtectCard := SysId || ':Slot ' || Slot || ' (' || CardModule || ')';
       RETURN ProtectCard;
END getProtectMLcardInfo;
```

3.1.73 GetQoSPhLoc

```
CREATE OR REPLACE FUNCTION Getqosphloc(slot IN NUMBER, cardPhloc IN NUMBER)
RETURN NUMBER
IS        PhLoc  NUMBER(20);        counter NUMBER;
BEGIN
PhLoc:=Getphysicallocwithpimppm(Getshelf(cardPhloc), slot, 255, 63, 0);
counter := countCosPh(PhLoc);
IF (counter = 0) THEN
PhLoc:=Getphysicallocwithpimppm(Getshelf(cardPhloc), slot, 255, 63, 1);
END IF;
```

```
RETURN PhLoc;
END Getqosphloc; /show errors;
```

3.1.74 GetShelf

```
Create or replace function  getShelf (physicalloc in number)
return number
IS
    shelf                       number;
BEGIN

SELECT trunc(mod(trunc(physicalloc/power(2,32)),power(2,8))) INTO  shelf from dual;

RETURN shelf;
END getShelf;
     END getShelf;
```

3.1.75 GetSlot

```
Create or replace function  getSlot (physicalloc in number)
return number
IS
    slot                        number;
BEGIN

SELECT mod(trunc(physicalloc/power(2,24)),power(2,8)) INTO slot from dual;

RETURN slot;
END getSlot;
```

3.1.76 GetSlotModuleName

```
create or replace function getSlotModuleName(p_nenodeid in number, p_moduletype in number)
return varchar2
IS
      o_modulename   varchar2(64);
      i_modeltype      number;
      p_slotmoduletype number;
BEGIN
        SELECT trunc(p_moduletype/power(2,16)) INTO p_slotmoduletype FROM dual;
        IF (p_nenodeid=0 OR p_slotmoduletype=0) THEN
           o_modulename:='SYSTEM';
        ELSE
          SELECT nemodeltype INTO i_modeltype
             FROM ne_info_table
            WHERE nenodeid=p_nenodeid;

           SELECT ModuleName INTO o_modulename
             FROM module_type_table
            WHERE moduletype=p_slotmoduletype
              AND modeltype=i_modeltype;
       END IF;
RETURN o_modulename;
Exception
    WHEN NO_DATA_FOUND
    THEN
    RETURN 'Unsupported';
END getSlotModuleName;
```

3.1.77 GetSlotModuleType

```
create or replace function getSlotModuleType(p_moduletype in number)
return varchar2
IS
      p_slotmoduletype number;
BEGIN
        SELECT trunc(p_moduletype/power(2,16)) INTO p_slotmoduletype FROM dual;
```

```
RETURN p_slotmoduletype;
END getSlotModuleType;
```

3.1.78 GetSpanTblCktSpanDestNodeId

```
CREATE or replace function getspantblCktSpanDestNodeId (CKTNODEIDIN number,CKTUNIQUEIDIN
number)return varchar2
IS
    cursor ctpspandestnodeidcursor is
    select CKTCTPNODEID
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID =CKTUNIQUEIDIN
    and ctp_type=4;

    ctpspandestnodeid_list varchar2(10000);
BEGIN

    for rec in ctpspandestnodeidcursor loop
        if ctpspandestnodeidcursor%ROWCOUNT = 1 then
            ctpspandestnodeid_list:= rec.CKTCTPNODEID;
        else
            ctpspandestnodeid_list := ctpspandestnodeid_list||','||rec.CKTCTPNODEID;
        end if;
    end loop;

    return ctpspandestnodeid_list;

END getspantblCktSpanDestNodeId;
```

3.1.79 GetSpanTblCktSpanSrcNodeId

```
CREATE or replace function getspantblCktSpanSrcNodeId (CKTNODEIDIN number,CKTUNIQUEIDIN
number)return varchar2
IS
    cursor ctpspansrcnodeidcursor is
    select CKTCTPNODEID
    from circuit_ctp_tbl
    where CKTNODEID=CKTNODEIDIN
    and CKTUNIQUEID=CKTUNIQUEIDIN
    and ctp_type=3;

    finalspansrcnodeid_list varchar2(10000);
BEGIN

    for rec in ctpspansrcnodeidcursor loop
        if ctpspansrcnodeidcursor%ROWCOUNT = 1 then
            finalspansrcnodeid_list:= rec.CKTCTPNODEID;
        else
            finalspansrcnodeid_list := finalspansrcnodeid_list||','||rec.CKTCTPNODEID;
        end if;
    end loop;

    return finalspansrcnodeid_list;

END getspantblCktSpanSrcNodeId;
```

3.1.80 GetSubIf

```
Create or replace function  getSubIf(neifindex in number)
return number
IS
i_neifindex number;
subifindex number;
BEGIN

IF neifindex < 0 THEN
i_neifindex := neifindex+power(2,32);
ELSE
i_neifindex:=neifindex;
END IF;
```

```
SELECT mod(trunc(i_neifindex/power(2,16)),power(2,16)) INTO subifindex FROM dual;

RETURN subifindex;
END getSubIf;
```

3.1.81 GetValidCktSecSrcNodeId

```
CREATE or replace function getValidCktSecSrcNodeId (CKTNODEIDIN number,CKTUNIQUEIDIN
number)return number
IS
    o_CktSedSrcNodeId number;
BEGIN
    select CktCtpNodeId into o_CktSedSrcNodeId
    from circuit_ctp_tbl
    where CktNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 1;

    return o_CktSedSrcNodeId;

END getValidCktSecSrcNodeId;
```

3.1.82 GetValidCktSrcNodeId

```
CREATE or replace function getValidCktSrcnodeid (CKTNODEIDIN number,CKTUNIQUEIDIN number)
return number
IS
    o_ValidCktSrcnodeid number;
BEGIN
    select CktCtpNodeId into o_ValidCktSrcnodeid
    from circuit_ctp_tbl
    where CktNodeId = CKTNODEIDIN
      and CktUniqueId = CKTUNIQUEIDIN
      and ctp_type = 0;

    return o_ValidCktSrcnodeid;

END getValidCktSrcnodeid;
```

3.1.83 GetValidModelType

```
create or replace function getValidModelType (CktNodeIdIN number, CktUniqueIdIN number,
ctp_typeIN number)
return number
IS
    o_ValidModelType number;
BEGIN
    select CKTCTPMODELTYPE into o_ValidModelType
    from circuit_ctp_tbl
    where CktNodeId = CktNodeIdIN
      and CktUniqueId = CktUniqueIdIN
      and ctp_type = ctp_typeIN;

    return o_ValidModelType;

END getValidModelType;
```

3.1.84 GetValidModuleType

```
create or replace function getValidModuleType (CktNodeIdIN number, CktUniqueIdIN number,
ctp_typeIN number)
return number
IS
    o_ValidModuleType number;
BEGIN
    select CKTCTPMODULETYPE into o_ValidModuleType
    from circuit_ctp_tbl
    where CktNodeId = CktNodeIdIN
```

```
      and CktUniqueId = CktUniqueIdIN
      and ctp_type = ctp_typeIN;

    return o_ValidModuleType;

END getValidModuleType;
```

3.1.85 GetValidObjectType

```
create or replace function getValidObjectType (CktNodeIdIN number, CktUniqueIdIN number,
ctp_typeIN number)return number
IS
    o_ValidObjectType number;
BEGIN
    select CKTCTPOBJECTTYPE into o_ValidObjectType
    from circuit_ctp_tbl
    where CktNodeId = CktNodeIdIN
      and CktUniqueId = CktUniqueIdIN
      and ctp_type = ctp_typeIN;

    return o_ValidObjectType;

END getValidObjectType;
```

3.1.86 Has_Next

- used by function ne_belongs_to_APC_domain to determine if a node belongs to an APC domain

```
CREATE OR REPLACE FUNCTION has_next
    (base_string       IN VARCHAR2,
     looking_for       IN VARCHAR2,
     start_looking_at  IN INTEGER,
     found_at_position OUT INTEGER,
     token OUT VARCHAR2) RETURN BOOLEAN
IS
target VARCHAR2(1000);
BEGIN
    -- add delimiters at the end of the base string
    target := concat(base_string,looking_for);
    -- find position of next delimiter
    found_at_position := INSTR (target,looking_for,start_looking_at,1);
    -- read the token
    if found_at_position > start_looking_at then
       token := SUBSTR(base_string,start_looking_at,found_at_position-start_looking_at);
    else
       token := '-1';
    end if;
    RETURN (found_at_position <> 0);
END has_next;
```

3.1.87 IS_NUMBER

```
CREATE OR REPLACE function IS_NUMBER(str in varchar2, rv in number:=-1) return number IS
dummy number;
begin
dummy := to_number(str);
return dummy;
Exception WHEN OTHERS then
return rv;
end;
```

3.1.88 IsIPv6

```
CREATE OR REPLACE
function isIPV6 (NEDBid number)
return number
```

```
is
  isipv6 number;

begin
  select IS_IPV6 into isipv6 from ne_info_table where nedbaccessid = NEDBid;


return isipv6;
 Exception
    when no_data_found then
        return 0;
end isIPV6;
```

3.1.89 MapObjectIndex

```
CREATE or replace function mapobjectIndex(neModelType number,
nemodelIndex number,
portnumber number,
iftype number,
moduletype number) return number
is
ObjIndex number;
BEGIN
  if neModelType = 38 then
      if moduletype = 34 then
         return 340;
      elsif moduletype = 32 then
         return 341;
      elsif moduletype = 31 then
         return 342;
      end if;
  end if;
    if neModelType = 39 then
      if moduletype = 68 then
         return 340;
      elsif moduletype = 69 then
         return 340;
      elsif moduletype = 32 then
         return 341;
      elsif moduletype = 33 then
         return 341;
      end if;
  end if;
  if (neModelType = 38 or neModelType = 39) and
     (moduleType = 28 or moduleType = 29 or moduleType = 30) then
    return 334;
  end if;
  if (neModelType = 18 or neModelType = 30) and
     (moduleType = 28 or moduleType = 29 or moduleType = 30) then
    return 147;
  end if;
  if    moduleType = 9 or moduleType = 106 or moduleType = 10  or
        moduleType = 11 or moduleType = 12 or moduleType = 13 or
        moduleType = 28 or moduleType = 29 or moduleType = 30 or
       moduleType =60 or moduleType = 85 or moduleType = 86 or moduleType = 87 or
moduleType = 113 then
                return 109;
  elsif neModelType = 3 and moduleType = 4  then
      return 114;
  elsif neModelType = 3 and moduleType = 5 then
      return 114;
  elsif neModelType = 5 and moduleType = 4  then
      return 204;
  elsif neModelType = 14 and moduleType = 4  then
      return 114;
  elsif neModelType = 14 and moduleType = 5 then
      return 114;
  elsif moduleType = 6 or moduleType = 7 or moduleType = 58  then
      return 121;
  elsif moduleType = 57 then
      return 122;
  elsif moduleType = 102 or moduleType = 103 then
      return 136;
  elsif moduleType = 8 then
```

```
      if nemodelIndex >= 61 then
         return 267;
      else
         return 123;
      end if;
  elsif moduletype = 47 and portnumber  <= 28 then
       return 204;
  elsif moduletype = 47 and portnumber >28  then
       return 208;
  elsif moduletype = 7 or moduletype = 8 then
     return 114;
  elsif moduletype = 14 then
      return 106;
  elsif moduletype = 15 then
     return 206;
  elsif moduletype = 31 then
     return 109;
  elsif moduletype = 32 then
     return 208;
  elsif moduletype = 34 then
     return 204;
  elsif moduletype = 62 then
    return 209;
  elsif moduletype = 118 and portnumber = 1 then
   return 141;
  elsif moduletype = 118 and portnumber = 2 then
    return 142;
  elsif moduletype = 119 and portnumber >= 1 and portnumber <= 4 then
    return 141;
  elsif moduletype = 119 and  portnumber =5 then
     return 142;
  elsif moduletype = 120 and portnumber = 1 then
    return  141;
  elsif moduletype = 120 and portnumber = 2 then
    return 142;
  elsif moduletype = 121 and portnumber = 1 then
    return 141;
  elsif (moduletype = 121 and (portnumber = 2 or portnumber = 3)) then
   return 142;
  elsif moduletype = 127 and portnumber = 1 then
        return 198;
  elsif (moduletype = 127  and (portnumber =  2 or portnumber = 3)) then
   return 193;
  elsif moduletype = 128 and portnumber = 1 then
    return 198;
  elsif moduletype = 128 and portnumber >= 2 and portnumber <= 7 then
   return 193;
  elsif moduletype = 129 and portnumber = 2 then
   return 196;
  elsif (moduletype = 129 and (portnumber = 1 or portnumber = 3 or portnumber = 4)) then
      return 193;
  elsif moduletype = 130 and portnumber = 6 then
    return 196;
  elsif moduletype = 130 and portnumber >= 1 and portnumber <= 5 then
    return 193;
  elsif moduletype = 131 and portnumber >= 1 and portnumber <= 32 then
    return 194;
  elsif moduletype = 131 and portnumber = 33 then
   return 193;
  elsif moduletype = 132 and portnumber >= 1 and portnumber <= 32 then
     return 194;
  elsif moduletype = 132 and portnumber = 33 then
    return 193;
  elsif moduletype = 133 and portnumber >= 1 and portnumber <= 8 then
     return 194;
  elsif moduletype = 133 and (portnumber = 9 or portnumber = 10) then
    return 195;
  elsif (moduletype = 134 and (portnumber = 1 or portnumber = 2)) then
    return 194;
  elsif moduletype = 134 and portnumber >= 3 and portnumber <= 6 then
    return 193;
 elsif moduletype = 135 and portnumber >= 1 and portnumber <= 4 then
   return 194;
 elsif moduletype = 135 and portnumber >= 5 and portnumber <= 8 then
   return 193;
 elsif moduletype = 136 and portnumber >= 1 and portnumber <= 8 then
```

```
  return 194;
elsif moduletype = 136 and portnumber >= 9 and portnumber <= 12 then
   return 193;
elsif moduletype = 137 and portnumber >= 1 and portnumber <= 2 then
  return 195;
elsif moduletype = 137 and portnumber >= 3 and portnumber <= 6 then
  return 193;
elsif moduletype = 138 and portnumber >= 1 and portnumber <= 8 then
   return 195;
elsif moduletype = 138 and portnumber >= 9 and portnumber <= 12 then
   return 193;
elsif moduletype = 143 then
      return 216;
elsif moduletype = 145 then
      return 267;
elsif moduletype = 163 or moduletype = 164  or moduletype = 187 or moduletype = 285 or
moduletype = 283 or moduletype = 284 then
      return 257;
elsif moduletype = 108 or moduletype = 83 or moduletype = 64 or moduletype = 65 or
moduletype =66 or
      moduletype = 67 then
    if nemodelindex = 33 then
      return 121;
    else
     return 114;
    end if;
elsif neModelType = 3 and moduleType = 218 then
      return 114;
elsif moduletype = 219 or moduletype = 253 or moduletype = 254 or moduletype = 220 then
      return 301;
elsif moduletype = 201 or moduletype = 202 or moduletype = 203 then
      return 109;
elsif moduletype = 204 then
      return 206;
elsif moduletype = 205 then
      return 106;
elsif moduletype = 206 or moduletype =207 or moduletype =208 then
      return 435;
elsif moduletype = 209 and portnumber<3 then
      return 436;
elsif moduletype = 209 and portnumber>=3 then
      return 434;
elsif moduletype = 210 or moduletype =211 then
      return 431;
elsif moduletype = 215 then
      return 437;
elsif moduletype = 183 and portnumber<9 then
      return 141;
elsif moduletype = 183 and portnumber>=9 then
      return 142;
elsif moduletype = 184 and portnumber<9 then
      return 141;
elsif moduletype = 184 and portnumber>=9 then
      return 184;
elsif moduletype = 181 then
      return 184;
elsif moduletype = 264 then
      return 194;
elsif moduletype = 265 then
      return 194;
elsif moduletype = 294 then
      return 193;
elsif moduletype = 295 then
      return 193;
elsif moduletype = 297 then
      return 193;
elsif moduletype =182 and portnumber<5 then
      return 184;
elsif moduletype =182 and portnumber=5 then
      return 142;
elsif moduletype = 271 and portnumber<3 then
      return 141;
elsif moduletype = 271 and portnumber>=3 then
      return 142;
elsif moduletype = 272 and portnumber<21 then
      return 141;
```

```
elsif moduletype = 272 and portnumber>=21 then
      return 142;
elsif moduletype = 273 and portnumber<17 then
      return 141;
elsif moduletype = 273 and portnumber>=17 then
      return 142;
elsif moduletype = 358 and portnumber<=1 then
      return 141;
elsif moduletype = 358 and portnumber>=2 then
      return 142;
elsif moduletype = 300 and portnumber<=1 then
      return 141;
elsif moduletype = 300 and portnumber>=2 then
      return 142;
elsif moduletype = 301 and portnumber<=4 then
      return 141;
elsif moduletype = 301 and portnumber>=5 then
      return 142;
elsif moduletype = 356 and portnumber<=4 then
      return 141;
elsif moduletype = 356 and portnumber>=5 then
      return 142;
elsif moduletype = 293 then
      return 142;
elsif moduletype = 248 and portnumber<9 then
      return 109;
elsif moduletype = 248 and portnumber>=9 then
    if neModelType = 3 then
      return 143;
    elsif neModelType = 14 then
      return 114;
    end if;
else
    return 0;
end if;
END;
```

3.1.90 MergedInvalidityList

```
CREATE OR REPLACE function mergedInvalidityList(lineTableValue in integer,
lineInvalidityList in varchar2, pathInvalidityList in varchar2) return varchar2 as

  TYPE column_name IS RECORD (
    table_name      VARCHAR2(50),
    view_name      VARCHAR2(50)
  );
TYPE column_names  IS TABLE OF column_name index by binary_integer;
column_list column_names;

tblName USER_TAB_COLUMNS.TABLE_NAME%TYPE;
dbColumnNo USER_TAB_COLUMNS.COLUMN_ID%TYPE;
occurence integer;
value USER_TAB_COLUMNS.COLUMN_ID%TYPE;
startPos integer;
offSet integer;
mergedInvalidityList varchar2(100);


begin
if(lineInvalidityList is null ) then
    return pathInvalidityList;
end if;

if(length(trim(lineInvalidityList)) =0 ) then
    return pathInvalidityList;
end if;

if(lineTableValue = -1) then
    return pathInvalidityList;
end if;
    tblName := 'ONS15454_SONET_LINE_PM_TABLE';
    mergedInvalidityList := pathInvalidityList;

    column_list(1).table_name := 'PPJCPDET';
```

```
    column_list(1).view_name := 'PPJC_PDET';
    column_list(2).table_name := 'NPJCPDET';
    column_list(2).view_name := 'NPJC_PDET';
    column_list(3).table_name := 'PPJCPGEN';
    column_list(3).view_name := 'PPJC_PGEN';
    column_list(4).table_name := 'NPJCPGEN';
    column_list(4).view_name := 'NPJC_PGEN';
    column_list(5).table_name := 'PJDIFF';
    column_list(5).view_name := 'PJ_DIFF';
    column_list(6).table_name := 'PJPSEC';
    column_list(6).view_name := 'PJP_SEC';
    column_list(7).table_name := 'PJNSEC';
    column_list(7).view_name :=  'PJN_SEC';

for recordNumber in 1..column_list.count loop

    select column_id into dbColumnNo from USER_TAB_COLUMNS where table_name = tblName and
column_name = column_list(recordNumber).table_name;


    occurence := 1;
        startPos := 1;

        loop
        offSet := instr(lineInvalidityList,':',1,occurence);
        occurence := occurence + 1;

        exit when offSet = 0;

        if(startPos = 1) then
        value := to_number(substr(lineInvalidityList,startPos,offSet-startPos),9999);
        else
        value := to_number(substr(lineInvalidityList,startPos +1,offSet-startPos -
1),9999);
        end if;

        if((dbColumnNo -1) = value) then
    select column_id into dbColumnNo from USER_TAB_COLUMNS where
table_name='ONS15454_SONET_PATH_PM_TABLE' and
column_name=column_list(recordNumber).table_name;
    mergedInvalidityList := mergedInvalidityList || (dbColumnNo -1)  || ':';

    exit;
    end if;
       startPos := offSet;
    end loop;


end loop;
    if(mergedInvalidityList is null) then
       return '';
    end if;
 return mergedInvalidityList;
end;
```

3.1.91 NE_Belongs_to_APC_Domain

-- to determine if a node belongs to an APC domain

```
CREATE OR REPLACE FUNCTION ne_belongs_to_APC_domain
    (sides IN VARCHAR2, neid in NUMBER) RETURN number
IS
token varchar2(15):='-1';
neid_in_APC_domain varchar2(15):='-1';
found_at integer;
found_in_token_at integer;
start_point integer := 1;
-- return value -1 menas that the NEDBID is not part of the domain
retval NUMBER:= -1;
BEGIN
    WHILE
      -- find if there is a next elemnt and assign the elemnt to token
      (has_next(sides,',',start_point,found_at,token))
```

```
    LOOP
       -- take from the elemnt the first part that is the NEDBID
       if has_next(token,' ',1,found_in_token_at,neid_in_APC_domain) then
          -- is the NEDBID we are looking fore
          if to_number(neid_in_APC_domain) = neid then
              -- if yes return it by the function...to be used in the where cluse of the
view
                  retval:= neid;
              exit;
                   end if;
       end if;
       --go to next elemnt
       start_point := found_at + 1;
    END LOOP;
    return retval;
END ne_belongs_to_APC_domain;
```

3.1.92 RemoveFirstByteIf

```
CREATE or REPLACE function removefirstbyteif (neifindex in number)
RETURN number
IS

   iftype       number;
   o_neifindex  number;
   i_neifindex  number;

BEGIN

   IF neifindex < 0 THEN
      i_neifindex := neifindex+power(2,32);
   ELSE
      i_neifindex:=neifindex;
   END IF;

     select mod(i_neifindex,power(2,24)) INTO  o_neifindex from dual;
return o_neifindex;
END removefirstbyteif;
```

3.1.93 RemoveLowBits

```
CREATE OR REPLACE function removeLowBits (aNum number, numOfBits integer)
RETURN integer
AS
    result number;
    modVal number;
BEGIN
        modVal := power (2, numOfBits);
        result := trunc (aNum / modVal);
        --result := result * modVal;
        return result;
END;
```

3.1.94 Time_Diff

```
CREATE or REPLACE FUNCTION time_diff (
DATE_1 IN DATE, DATE_2 IN DATE) RETURN NUMBER IS

NDATE_1   NUMBER;
NDATE_2   NUMBER;
NSECOND_1 NUMBER(5,0);
NSECOND_2 NUMBER(5,0);

BEGIN
  -- Get Julian date number from first date (DATE_1)
  NDATE_1 := TO_NUMBER(TO_CHAR(DATE_1, 'J'));

  -- Get Julian date number from second date (DATE_2)
  NDATE_2 := TO_NUMBER(TO_CHAR(DATE_2, 'J'));
```

```
  -- Get seconds since midnight from first date (DATE_1)
  NSECOND_1 := TO_NUMBER(TO_CHAR(DATE_1, 'SSSSS'));

  -- Get seconds since midnight from second date (DATE_2)
  NSECOND_2 := TO_NUMBER(TO_CHAR(DATE_2, 'SSSSS'));

  RETURN (((NDATE_2 - NDATE_1) * 86400)+(NSECOND_2 - NSECOND_1));

END time_diff;
```

3.1.95 TYPE AlarmInfoListType

--create or replace TYPE AlarmInfoListType as table of AlarmInfoType;

```
create TYPE AlarmInfoListType as table of AlarmInfoType;
```

3.1.96 TYPE AlarmInfoType

--create or replace TYPE AlarmInfoType as OBJECT (

```
create TYPE AlarmInfoType as OBJECT (
        nedbaccessid                number,
        Alarmseqnum                 number,
        ModuleOrIfIndex             number,
        ActiveAlarmTimeStamp        date,
        ActiveAlarmType             number,
        ActiveAlarmSeverity         number,
        ActiveAlarmServEff          number,
        ActiveAlarmAdditionalInfo   varchar2(512),
        AckIndex                    number,
        ActiveAlarmFlag             number,
        ActiveAlarmComment          varchar2(2014),
        ClearAlarmTimeStamp         date,
        NEAlarmTimeStamp            date,
        NEAlarmClearTimeStamp       date,
        ModuleType                  number,
        Physicalloc                 number,
        alarmstatus                 number,
        linkid                      link_id_tabtype,
        actiontype                  number,
        externalcondition           varchar2(1024),
        ModelType                   number,
        ObjectType                  number,
        StrObjInstance              VARCHAR2(256),
        StrDirection                VARCHAR2(6),
        StrLocation                 VARCHAR2(4)
);
```

3.1.97 TYPE Link_Id_Tabtype

```
create or replace TYPE link_id_tabtype is Table of number;
/
```

--drop type AlarmInfoListType;
--drop type AlarmInfoType;

A. Oracle Built-In Data Types

This chapter describes Oracle built-in data types used in Prime Optical tables.

The information in the following table is provided as a reference only. For additional information about Oracle built-in data types, refer to Oracle documentation.

Table A-1: Oracle Built-in Data Types

Data Type	Description
BFILE	Binary data that is stored in an external file.
BLOB	Binary data.
CHAR(size)	Fixed-length character data of length-size bytes.
CLOB	Single-byte character data.
DATE	Fixed-length date and time data, ranging from January 1, 4712 BCE (B.C.) to December 31, 4712 CE (A.D.).
FLOAT(n)	Decimal data type.
LONG	Variable-length character data.
LONG RAW	Variable-length raw binary data.
MLSLABEL	Trusted Oracle data type.
NCHAR(size)	Fixed-length character data of length-size characters or bytes, depending on the national character set.
NCLOB	Single-byte or fixed-length multibyte national character set (NCHAR) data.
NUMBER(p, s)	Variable-length numeric data. Maximum precision p and/or scale s is 38.
NVARCHAR2(size)	Variable-length character data of length-size characters or bytes, depending on the national character set. You must specify a maximum size.
RAW(size)	Variable-length raw binary data. You must specify a maximum size.
ROWID	Binary data representing row addresses.
VARCHAR2(size)	Variable-length character data. You must specify a maximum size.

B. Config_Table Parameters

The Ctm_Config_Table stores Prime Optical configuration information. The following table defines the parameters in the Ctm_Config_Table and is arranged by section name value (as listed in the Ctm_Config_Table.sql file).

Table B-1: Ctm_Config_Table Parameters

Parameter Name	Definition
transportManager	
jmoco-port	Port where JMOCO events are sent.
main-error-level	Error levels supported in Prime Optical. Values are trace, debug, minor, major, critical.
min-alarm-update-interval	Minimum time, in seconds, when an active_alarm_update event is sent.
ems-sys-id	System ID that indicates who sent the event (for example, if the event is generated by the EMS, the system ID is PrimeOptical).
preferred-jar-path	Path to obtain Cisco Transport Controller (CTC) jar files.
preferred-jar-version	Version of the CTC jar that needs to be picked.
tl1-connection-timeout	Timeout period if a TL1 connection has not been used.
jmoco-trace-frequency	(Not used)
jmoco-max-user-sessions	Maximum number of sessions a jmoco engine can support.
oss-thread-count	Maximum number of threads that can be spawned in Prime Optical.
oss-work-queue-threshold	Maximum number of work objects that can be added into the work queue.
oss-max-threads	Maximum number of threads that can be read from the work queue.
oss-jmoco-threads	Number of threads used to read events from the work queue.
oss-jmoco-event-threads	Number of threads used to distribute events from the work queue to registered users.
oss-socket-send-size	Maximum size of the buffer that the socket can support to send.
oss-socket-recv-size	Maximum size of the buffer that the socket can support to receive.

Parameter Name	Definition
refresh-data-timeout	Specifies the Refresh Data Timer interval, in milliseconds. When a discovery change event is received, the Refresh Data Timer is started. The timer records the number of NEs in synch status. When the timer expires, it retrieves the number of NEs in synch and compares this value to the recorded one. If the values are equal, it forces a Refresh Data action.
gatewayTL1	
gwTL1ForwardingAgent	Whether Prime Optical GateWay/TL1 is installed, stopped, or started.
response-parameters	(Not used)
ne-name	
alarm-logic	
in-progress-frequency	
oss-mux	
oss-pt	
load-db-nes	
deny-act-user	
oss-port	Prime Optical GateWay/TL1 port.
15454-vasco-tl1-port	Vasco TL1 port for the ONS 15454.
15454-tl1-port	TL1 port.
15327-tl1-port	
15216-oadm2-tl1-port	
15216-edfa-tl1-port	
15310-tl1-port	
15454SDH-tl1-port	
15216-edfa3-tl1-port	
15600-tl1-port	
15600-tl1-telnet-port	(Not used)
15530-tl1-port	TL1 port.
15540-esp-tl1-port	
15540-espx-tl1-port	
poll-frequency	Time, in seconds, that Prime Optical GateWay/TL1 checks the status of the NE.
error-level	Debug level for the Prime Optical GateWay/TL1 service module.

Parameter Name	Definition
gwTL1Timeout	Time, in seconds, before Prime Optical GateWay/TL1 times out.
gwTL1LoginTimeout	Enables or disables the Prime Optical GateWay/TL1 timeout period if the ACT-USER command is not performed before the timeout period.
gwTL1CmdTimeout	Enables or disables the Prime Optical GateWay/TL1 timeout period if a valid TL1 command is not performed before the timeout period.
gwTL1Connections	Number of TL1 connections to Prime Optical GateWay/TL1.
enable-gwtl1-advisory	Enables or disables display of advisory messages.
gwtl1-advisory-message	Prime Optical GateWay/TL1 advisory message. The default message reads "NOTICE: This is a private computer system. Unauthorized access or use may lead to prosecution."
command-echo	Allows you to view the command line when you enter commands during a GateWay/TL1 session. This also allows you to log all messages received during a GateWay/TL1 session. The default is false, meaning that GateWay/TL1 session echoing is disabled.
gatewaySNMP	
gatewaySNMP	Whether Prime Optical GateWay/SNMP is installed, stopped, or started.
snmp-hosts	SNMP trap destination hosts.
gatewayCORBA	
gatewayCORBA	Whether Prime Optical GateWay/CORBA is installed, stopped, or started.
corba-request-timeout	CORBA timeout value.
thread-pool-size	Thread pool size used in the CORBA gateway.
iiop-listen-port	CORBA gateway listening port.
error-level	Debug level for Prime Optical GateWay/CORBA.
heartbeat-interval	Interval, in minutes, for the heartbeat event.
enable-gwcorba-encryption	Whether the CORBA gateway encryption feature is enabled or disabled.
max-number-active-sessions	Specifies the number of Prime Optical GateWay/CORBA sessions that can be active at the same time. The range is from 4 to 25; the default is 4.

Parameter Name	Definition
max-events_per_consumers	Sets the MaxEventsPerConsumer administrative quality of service (QoS) parameter on the notification channel. The notification server uses this property to limit the maximum number of events in a given channel allowed to queue at any one time. The default value is 0, meaning that the notification server does not limit the maximum number of events that can be queued. If no limits are imposed on the queue, the notification server might run out of memory, because the server must keep all events in memory until they are consumed by all registered consumers. Any change to this value should be done with extreme caution. If you set the value too low, the NMS cannot receive all notifications. If you set the value too high, the Prime Optical notification server runs out of memory. The current value can handle alarm bursts of 10,000 events per minute.
notification_service_name	Defines the service name used by the resolve_initial_reference function to get a reference to the notification service. The Prime Optical GateWay/CORBA installation installs the notification service. However, if you want to use your own notification service, you can modify this parameter.
notification_service_naming_context	Defines the naming context of the notification service. This property is used when the resolve_initial_reference function fails to resolve the notification service. Prime Optical GateWay/CORBA contacts the naming service to resolve the name context defined in this property. The value of this property must match the value published by your notification server.
notification_service_factory_ior_name	Sets the notification service factory Information Object Repository (IOR) filename located in the /opt/CiscoTransportManagerServer/openfusion/domains/OpenFusion/localhost/NotificationService/NotificationSingleton/NotificationService.ior directory. The FactoryIORFile property defines the path to a text file that contains the IOR of the notification service. This property is used only after the resolve_initial_reference function and the naming service both fail. Prime Optical GateWay/CORBA opens the file as defined by the URL format in this property and retrieves the IOR. This parameter allows you to run your notification service on a different host to improve performance.

Parameter Name	Definition
notification_service_port_number	Sets the port number that the notification service uses to listen for incoming requests. The port number is set in the IOR for the notification service. The use IOR and use IOR endpoint properties are set properly. If set to 0 (the default), the port number is allocated by the operating system.
session_port_number	Configures the IIOP listening port. The Prime Optical GateWay/CORBA service listens to CORBA requests on this port. If set to 0 (the default), the session port number is allocated by the operating system.
name_service_server_list	Defines where the name servers are running. Accepts a comma-separated list of hostnames.
name_service_root_ior	Defines the path to find the naming service's IOR on each host defined on the server list. The complete path is constructed as <http://<item>_of_ServerList><RootIORLoc>.
ne	
poll-frequency	Wait duration for the next health poll to start.
corba-request-timeout	CORBA timeout value.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
ne-thread-pool-size	Number of threads that are available for NE discovery.
ne-discov-thread-pool-size	Number of threads that are available for initial poll, alarm resync, and config resync.
snmp-trap-thread-pool-size	Number of threads that are available for the SNMP trap event converter.
alarm-thread-pool-size	Number of threads that are available for the alarm dispatcher.
ne-circuit-discovery-thread-pool-size	Number of threads that are available for circuit discovery.
pm	
number-of-collectors	Number of threads that are available for the collector thread pool. This thread pool is used to poll the NEs and obtain performance statistics.
number-of-writers	Number of threads that are available for the database writer thread pool. This thread pool is used to write performance statistics in the database.
number-of-nes	(Not used)

Parameter Name	Definition
store-non-zeros-only	Runs PM data collection in optimized mode. If the performance counts are zero, this parameter controls whether or not it should be written to the database.
pm-disk-limit	(Not used)
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
pm-15-min-retrieval-enabled	Controls PM alarm generation. When enabled and PM collection fails for a 15-minute interval, a PM alarm is emitted.
pm-15-min-retrieval-secs-between-retries	Number of seconds to wait before Prime Optical retries and polls the NE. Used when PM collection fails in normal mode for a 15-minute interval.
pm-15-min-retrieval-number-of-retries	Number of times Prime Optical should retry and poll the NEs. Used when PM collection fails in normal mode for a 15-minute interval.
pm-1-day-retrieval-enabled	Controls PM alarm generation. When enabled and PM collection fails for a one-day interval, a PM alarm is emitted.
pm-1-day-retrieval-secs-between-retries	Number of times Prime Optical should retry and poll the NEs. Used when PM collection fails in normal mode for a one-day interval.
pm-1-day-retrieval-number-of-retries	Number of times Prime Optical should retry and poll the NEs. Used when PM collection fails in normal mode for a one-day interval.
max-realtime-sessions	Maximum number of concurrent real-time PM sessions allowed per client.
max-realtime-objects	Maximum number of objects that can be polled in a single real-time PM session.
max-realtime-rows	Maximum number of rows that are held in the real-time in-memory transient buffer on a per-session basis. Once this limit is reached, the buffer rolls over, meaning that the chronologically oldest rows are removed from the buffer, making room for the new rows.
cm	
config-vlan-thread-count	Number of threads that are available for VLAN discovery.
config-vlan-autosync-delay	Delay after which time the topologies and VLANs are synchronized with the NE.

Parameter Name	Definition
bandwidth-util-dsp	In the Control Panel window > SONET/SDH NE Service pane, check the Enable Bandwidth DSP check box to enable the check for available bandwidth during VLAN creation. If you create a VLAN that could cause oversubscription of the available bandwidth for the L2 topology, and if the Enable Bandwidth DSP check box is checked, you receive an error message that oversubscription might occur. The check box is disabled by default.
bandwidth-util-dsp-sdh	In the Control Panel window > SONET/SDH NE Service pane, check the Enable Bandwidth DSP check box to enable the check for available bandwidth during VLAN creation. If you create a VLAN that could cause oversubscription of the available bandwidth for the L2 topology, and if the Enable Bandwidth DSP check box is checked, you receive an error message that oversubscription might occur. The check box is disabled by default.
config-dsp-max-idle-time	Set to check if sufficient time has passed from the last event received and all topology audits have been completed.
config-dsp-msync-wait-time	Time to wait until the set time for Node Manager Sync to be completed.
config-max-partial-topo-time	Time to schedule a task to scan and mark all partial topologies as incomplete.
config-sync-thread-run-delay	Frequency of tasks that check if the data service module is ready for provisioning (data provisioning ready state).
config-topo-incomplete-thread-run- delay	Time to schedule a task to declare the incomplete topologies.
dsp-debug-telnet-port	Telnet port to turn on the DataService.log for debugging topology discovery-related information.
config-resync-thread-pool-size	(Not used)
config-poll-frequency	Polling period to check if the config resync time has been reached.
config-resync-frequency	Periodic frequency at which the config resync is performed.
config-request-queue-size	(Not used)
iiop-listen-port	(Not used)
backup-frequency	Frequency at which the backup should be performed (for example, daily, weekly, or just once).
backup-copies-ons15454	Number of copies of backed-up configuration and provisioning information residing in the flash memory of the NEs.

Parameter Name	Definition
backup-copies-ons15327	
backup-copies-ons15540	
backup-copies-ons15454sdh	
backup-copies-ons15600	
backup-copies-ons15600sdh	
backup-copies-ons15530	
backup-copies-ons15540xps	
backup-copies-ons15216edfa	
backup-copies-ons15310cl	
backup-copies-ons15216edfa3	
backup-copies-ons15305ctc	(Not used)
backup-status-ons15454	Current status of the task: Queued, Running, Canceled, Failed, Succeeded, or Waiting.
backup-status-ons15327	
backup-status-ons15540	
backup-status-ons15540xps	
backup-status-ons15530	
backup-status-ons15454sdh	
backup-status-ons15600	
backup-status-ons15600sdh	
backup-status-ons15216edfa	

Parameter Name	Definition
backup-status-ons15310cl	
backup-status-ons15216edfa3	
backup-status-ons15305ctc	(Not used)
backup-time-ons15454	Time for the NE configuration backup.
backup-time-ons15327	
backup-time-ons15540	
backup-time-ons15454sdh	
backup-time-ons15600	
backup-time-ons15600sdh	
backup-time-ons15530	
backup-time-ons15540xps	
backup-time-ons15216edfa	
backup-time-ons15310cl	
backup-time-ons15216edfa3	
backup-time-ons15305ctc	(Not used)
manual_backup-copies-ons15454	Number of copies that can be backed up manually. Each NE backup is performed by selecting the Configuration > Memory Backup > User Specified radio button.
manual_backup-copies-ons15327	
manual_backup-copies-ons15540	
manual_backup-copies-ons15454sdh	

Parameter Name	Definition
manual_backup-copies-ons15600	
manual_backup-copies-ons15600sdh	
manual_backup-copies-ons15530	
manual_backup-copies-ons15540xps	
manual_backup-copies-ons15216edfa	
manual_backup-copies-ons15216edfa3	
manual_backup-copies	
manual_backup-copies-ons15310cl	
manual_backup-copies-ons15305ctc	(Not used)
backup-directory	Name of the link to the source directory on the server for backup file upload operations. This name appears in the /opt/CiscoTransportManagerServer/webServer/htdocs folder. For example, if the /htdocs folder contains the link “admin -> /opt/CiscoTransportManagerServer/admin,” admin is the value of the backup-directory property.
archive-dest-dir	Destination directory on the client for archiving upload operations. The client installation path is the default path. For example, if the complete path for the archiving directory is C:/TransportManagerClient/archive, the value of the archive-dest-dir property is archive. The default value for this property is log.
archive-source-link-name	Name of the link to the source directory on the server for archiving upload operations. This name appears in the /opt/CiscoTransportManagerServer/webServer/htdocs folder. For example, if the /htdocs folder contains the link “archive -> /opt/CiscoTransportManagerServer/log,” archive is the value of the archive-source-link-name property. When you change the server source directory, the name of the link does not change.

Parameter Name	Definition
log-source-link-name	Name of the link to the source directory on the server for log file upload operations. This name appears in the /opt/CiscoTransportManagerServer/webServer/htdocs folder. For example, if the /htdocs folder contains the link "log -> /opt/CiscoTransportManagerServer/log," log is the value of the log-source-link-name property. When you change the server source directory, the name of the link does not change.
alarm-data	Parameters used for CTC-based automatic backup.
pm-data	
auditlog-data	
manual-alarm-data	(Not used)
manual-pm-data	(Not used)
manual-auditlog-data	(Not used)
auto-job-thread-pool-size	Number of threads used to read the automatic jobs from the work queue. The default is 10.
manual-job-thread-pool-size	Number of threads used to read the manual jobs from the work queue. The default is 10.
snmp	
snmpTrapService	Enables or disables the SNMP trap service upon Prime Optical server startup. The default is enabled.
snmp-trap-queue-size	Maximum number of traps that can be queued in the SNMP trap service before being distributed to NE services. When the queue reaches its maximum, incoming traps are dropped. The default is 10000.
snmp-trap-drop-size	Threshold to drop incoming traps. The default is 100.
snmp-trap-forwarding-port	SNMP gateway trap port. The default is 8765.
snmp-trap-port	Incoming trap port. The default is 162.
snmp-mibs-dir	MIB file directory. The default is /mibs.
st-error-level	Debug log level. The default is minor.
serviceManager	
sm-jvm-heap-init	Initial heap size allocated to the SM service.
sm-jvm-heap-max	Maximum heap size allocated to the SM service.
sm-error-level	Error level for the process. Values are debug, trace, minor, major, or critical.
service-stop-timer	Time interval for which the SM service waits for the service to stop.
diskspace-poll-period	(Not used)

Parameter Name	Definition
hello-poll-interval	Poll interval at which the hello operation is performed on an NE.
hello-stop-timer	Time interval for which the SM service waits for the service to shut down gracefully.
hello-kill-timer	Time interval after which the SM service forcibly kills the service when it does not respond.
ons15200-ne	
ne-service	(Not used)
ne-threshold	(Not used)
ne-error-level	(Not used)
ne-resync-frequency	(Not used)
ne-thread-pool-size	(Not used)
ne-jvm-heap-init	(Not used)
ne-jvm-heap-max	(Not used)
ons15216-ne	
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-threshold	Maximum number of NEs that can be handled. The default is 1000.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-resync-frequency	Periodic frequency at which configuration resynchronization is performed. This value can be modified from the Prime Optical client through Administration > Control Panel > NE Service > NE_service_name > Resync Interval.
ne-thread-pool-size	Number of threads that are available for NE discovery.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
ons15327-ne	
fault-sync	Enables or disables historical alarm collection.
ons15600-ne	
fault-sync	Enables or disables historical alarm collection.

Parameter Name	Definition
ons15600sdh-ne	
fault-sync	Enables or disables historical alarm collection.
ons15454-ne	
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-low-threshold	Threshold value for the maximum number of NEs that can be discovered.
ne-threshold	Maximum number of NEs that can be handled. The default is 500.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-resync-frequency	Periodic frequency at which configuration resynchronization is performed. This value can be modified from the Prime Optical client through Administration > Control Panel > NE Service > NE_service_name > Resync Interval.
ne-thread-pool-size	Number of threads that are available for NE discovery.
ne-ios-cr-thread-pool-size	Number of threads in a pool where each thread is used to process Cisco IOS configuration resync updates. The default is 10.
ne-ios-updates-cr-frequency	Number of seconds when Cisco IOS configuration resync updates are processed. The default is 60.
audit-trail-collection-interval	Wait duration for the next audit trail collection to start.
audit-trail-thread-count	Number of threads that can be used to collect the audit trail.
fault-sync	Enables or disables historical alarm collection.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
blsr-interval	Difference in interval between east and west port job execution.

Parameter Name	Definition
client-list-thread-pool-size	Number of threads in a pool where each thread is used to poll the ENE client list from the gateway NE (that is, the GNE with proxy turned on). The default is 20.
ctm-event-disp-thread-count	Number of threads that are available for dispatching events.
ctm-invupd-thread-count	Number of threads that are available to handle inventory updates.
client-list-poll-frequency	Interval at which the list of ENEs for all the GNEs in the domain is collected. The Proxy_Server_Table is modified with the changes and the GNE, ENE, and LNE changes are reflected in the Domain Explorer.
ons15454sdh-ne	
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-low-threshold	Threshold value for the maximum number of NEs that can be discovered.
ne-threshold	Maximum number of NEs that can be handled. The default is 600.
ne-resync-frequency	Periodic frequency at which configuration resynchronization is performed. This value can be modified from the Prime Optical client through Administration > Control Panel > NE Service > NE_service_name > Resync Interval.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-thread-pool-size	Number of threads that are available for NE discovery.
ne-ios-cr-thread-pool-size	Number of threads in a pool where each thread used to process Cisco IOS configuration resync updates. The default is 10.
ne-ios-updates-cr-frequency	Number of seconds when Cisco IOS config resync updates are processed. The default is 60.
audit-trail-collection-interval	Wait duration for the next audit trail collection to start.

Parameter Name	Definition
audit-trail-thread-count	Number of threads that can be used to collect the audit trail.
fault-sync	Enables or disables historical alarm collection.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
mspring-interval	Difference in interval between east and west port job execution.
client-list-poll-frequency	Interval at which the list of ENEs for all the GNEs in the domain is collected. The Proxy_Server_Table is modified with the changes and the GNE, ENE, and LNE changes are reflected in the Domain Explorer.
ons155xx-ne	
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-threshold	Maximum number of NEs that can be handled. The default is 1000.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-resync-frequency	Periodic frequency at which the configuration resynchronization is performed. This value can be modified from the Prime Optical client through Administration > Control Panel > NE Service > NE_service_name > Resync Interval.
discovery-threads-pool-size	Number of discovery threads that are available for discovering ONS 155xx NEs.
discovery-threads-p1	Number of threads used by the ONS 155xx NE service for 155xx device discovery, including inventory and alarms.
discovery-threads-p2	Number of threads used by the ONS 155xx NE service for topology and circuit discovery.
discovery-wait-on-ne-create	Wait time, in seconds, after an ONS 155xx NE has been added to the network.
discovery-enable-cdp	Enables or disables discovery of configured CDP neighbors.
discovery-auto-neighbor	Enables or disables discovery of configured non-CDP neighbors.
discovery-auto-delete-stale-links	Enables or disables automatic deletion of stale links from the database.

Parameter Name	Definition
tftp-timeout	Timeout period for any TFTP operations to and from the ONS 155xx and the server.
backup-config-before-restore	Enables or disables backup of the NE configuration before restoring the new configuration.
snmp-pdu-maxvars	Size of the SNMP PDU varbinds used while creating a connection with the NE.
EntityAlarmsToIgnore	Indicates whether there are any ONS 155xx entity alarms to be ignored.
ne-thread-pool-size	Number of threads that are available for NE discovery.
discovery-fast-mode	Enables or disables discovery of ONS 155xx NEs in fast mode.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
ons15305-ne	
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-threshold	Maximum number of NEs that can be handled. The default is 1000.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-resync-frequency	Periodic frequency at which the configuration resynchronization is performed. This value can be modified from the Prime Optical client through Administration > Control Panel > NE Service > NE_service_name > Resync Interval.
ne-thread-pool-size	Number of threads that are available for NE discovery.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
ne-sntp-default-sync-time	Value used for the resync time. The default is 30.
ons15310-ne	
fault-sync	Enables or disables historical alarm collection.
unmanaged-ne	

Parameter Name	Definition
ne-service	The NE service process starts automatically whenever at least one NE is added to the Prime Optical domain. If no NEs are present, this service does not run. If it ran previously, this service terminates as soon as no NEs are in the domain.
ne-error-level	NE service error log level. By default, minor severity log messages are entered in the NE service logs. This value can be modified from the Prime Optical client through Administration > Control Panel > Logging Properties.
ne-threshold	Maximum number of NEs that can be handled. The default is 500.
ne-jvm-heap-init	Initial heap size for the NE service process.
ne-jvm-heap-max	Maximum heap size for the NE service process.
ons15454-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 700.
pm-error-level	PM service error log level.
pm-http-retries	(Not used)
pm-wait-period	
pm-interface-ds1	Enables or disables performance statistics collection.
pm-interface-ds3	
pm-interface-sonet	
pm-interface-sts	
pm-interface-vt	
pm-interface-enet	Enables or disables Ethernet performance statistics collection.
pm-interface-physical	Enables or disables the physical performance statistics collection for optical cards.
pm-interface-dwdm	Enables or disables OTN section, line, and path performance statistics collection for DWDM cards.
pm-interface-E1	Enables or disables performance statistics collection for E1 cards.
pm-snmp	Enables or disables SNMP performance monitoring for cards on the ONS 15454 SONET.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.

Parameter Name	Definition
ons15454sdh-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-http-retries	(Not used)
pm-wait-period	
pm-interface-E1	Enables or disables performance statistics collection.
pm-interface-E3	
pm-interface-E4	
pm-interface-ds1	
pm-interface-ds3I	
pm-interface-sdh	
pm-interface-HO	
pm-interface-LO	
pm-interface-enet	Enables or disables Ethernet performance statistics collection.
pm-interface-physical	Enables or disables physical performance statistics collection for optical cards.
pm-interface-dwdm-sdh	Enables or disables OTN section, line, and path performance statistics collection for DWDM cards.
pm-snmp	Enables or disables SNMP performance monitoring for cards on the ONS 15454 SDH.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
ons15600-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-http-retries	(Not used)

Parameter Name	Definition
pm-wait-period	(Not used)
pm-interface-sonet	Enables or disables SONET section and line performance statistics collection.
pm-interface-sts	Enables or disables SONET STS path performance statistics collection.
pm-interface-enet	Enables or disables Ethernet performance statistics collection.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
ons15600sdh-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-http-retries	(Not used)
pm-wait-period	(Not used)
pm-interface-sdh	Enables or disables SDH performance statistics collection.
pm-interface-HO	Enables or disables higher-order (HO) path performance statistics collection.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
ons155xx-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-poll-delay	Indicates the delay time, in seconds, for 15-minute and 24-hour PM collections to start. The default is 90 seconds.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.

Parameter Name	Definition
ons15216-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
ons1530x-pm	
pm-service	Possible values are active or not-active. The default value is not-active. If the value of the property is active, the corresponding PM service will be started when PM data collection is enabled on the NE(s). The value of this property should be set to active to collect PM data on NEs. This value can be modified from the Prime Optical client through Administration > Control Panel > PM Service > NE_type.
pm-threshold	Maximum number of NEs managed by the PM service process. The default is 1000.
pm-error-level	PM service error log level.
pm-15m-delay	Time, in seconds, to wait after the quarter is exceeded before collecting PM data.
pm-24h-delay	Time, in seconds, to wait after the 24 hours are exceeded before collecting PM data.
pm-jvm-heap-init	Initial heap size for the PM service process.
pm-jvm-heap-max	Maximum heap size for the PM service process.
security	
password-reuse-count	Number of previously used passwords to compare against the new password. The range is from 0 to 5 passwords; the default is 5 passwords. A value of 0 disables this feature.
password-change-interval	Number of days a user must wait between password changes. The range is from 0 to 99 days; the default is 20 days. A value of 0 disables this feature.
password-differ-limit	Number of characters by which the new password must differ from the previous one. The range is from 1 to 5 characters; the default is 3 characters.
password-age	Sets the password aging interval to any interval between 0 and 999 days. You are prompted to change the password after the specified number of days. The default is 30 days.
password-	Sets an early warning period for password expiration. Prime Optical

Parameter Name	Definition
notification-enabled	supports values of 0 to (password aging – 1), with a maximum of 30. For example, if the password aging is configured for 30 days, the maximum early notification value would be 29 days. A value of 0 disables this feature.
password-notification-interval	Sets the number of days before the warning for password expiration. The default is 0.
max-unsuccessful-attempts	Sets the maximum number of attempts a user is allowed to log in before being locked out for the length of time specified in the Lockout Time field. The allowable range is from 0 to 10 retries; the default is 5 retries.
min-password-length	Minimum password length. The range is from 2 to 10 characters; the default is 6 characters.
max-password-length	Maximum password length. The range is from 10 to 12 characters; the default is 12 characters.
num-alpha-chars	Minimum number of alphabetic characters that the password must include. The range is from 0 to 2; the default is 2 alphabetic characters.
num-locase-alpha-chars	Minimum number of lowercase alphabetic characters that the password must include. The range is from 0 to 2; the default is 1 lowercase character.
num-upcase-alpha-chars	Minimum number of uppercase alphabetic characters that the password must include. The range is from 0 to 2; the default is 1 uppercase character.
num-numeric-chars	Minimum number of numeric characters that the password must include. The range is from 0 to 2; the default is 1 numeric character.
num-special-chars	Minimum number of special characters that the password must include. The range is from 0 to 2; the default is 1 special character.
allow-numeric-firstlast	Specifies whether a numeric character is allowed as the first or last character in the password. The default is true; the first or last character in the password can be a number.
allow-special-firstlast	Specifies whether a special character is allowed as the first or last character in the password. The default is true; the first or last character in the password can be a special character.
allow-circular-shift	Specifies whether the user ID or a circular shift of the ID can be used in the password. The default is false; the user ID or a circular shift of the ID cannot be used in the password.
special-char-set	Special character set to use. The default is TL1+UNIX.
lockout-period	Sets the number of minutes a user's Prime Optical session is inactive before Prime Optical automatically locks the user out. The allowable range is from 0 to 120 minutes in 1-minute increments; the default is 30 minutes. A value of 0 disables this feature.
client-lockout	Enables or disables client lockout.
client-lockout-period	Sets the number of minutes a user's Prime Optical session is inactive before Prime Optical automatically logs the user out. The allowable range is from 0 to 1440 minutes in 1-minute increments. A value of 0 disables this feature.

Parameter Name	Definition
client-logout	Enables or disables client logout.
client-logout-period	Sets the number of minutes a user's Prime Optical session is inactive before Prime Optical automatically logs the user out. The allowable range is from 0 to 1440 minutes in 1-minute increments. A value of 0 disables this feature.
security-advisory-message	(This parameter is not applicable to ONS 155xx NEs.) If checked, the advisory message that appears on login is enabled.
ons1530xusername	Usernames and passwords used to connect to a specific NE.
ons1530xpassword	
ons1530xGWTL1username	
ons1530xGWTL1password	
ons15454username	
ons15454password	
ons15454GWTL1username	
ons15454GWTL1password	
ons15454MLSeriesusername	
ons15454MLSeriespassword	
ons15327username	
ons15327password	
ons15327GWTL1username	
ons15327GWTL1password	
ons15454SDHusername	
ons15454SDHpassword	
ons15454SDHGWTL1username	
ons15454SDHGWTL1password	
ons15454SDHMLSeriesusername	
ons15454SDHMLSeriespassword	
ons15216EDFA2username	Usernames and passwords used to connect to a specific NE.

Parameter Name	Definition
ons15216EDFA2password	
ons15216EDFA3username	
ons15216EDFA3password	
ons15216OADMUserName	
ons15216OADMUserPass	
ons15216OADMGWTL1UserName	
ons15216OADMGWTL1UserPass	
ons15216EDFAGWTL1UserName	
ons15216EDFAGWTL1UserPass	
ons15216EDFAUserName	Usernames and passwords used to connect to a specific NE.
ons15216EDFAUserPass	
ons15216EDFA3GWTL1UserName	
ons15216EDFA3GWTL1UserPass	
ons15216EDFA3FtpDir	
ons15216EDFA3FtpUserName	
ons15216EDFA3FtpUserPass	
ons15305ctcusername	(Not used)
ons15305ctcpassword	(Not used)
ons155xxusername	(Not used)
ons155xxpassword	(Not used)
ons155xxGWTL1username	Username and password used to connect to the ONS 155xx GateWay/TL1 NE.
ons155xxGWTL1password	
telnet-password	Telnet password.
ne-global-user-name	NE global username.

Parameter Name	Definition
ne-global-user-password	NE global password.
ons15600username	Usernames and passwords used to connect to a specific NE.
ons15600password	
ons15600GWTL1username	
ons15600GWTL1password	
ons15600SDHusername	
ons15600SDHpassword	
ons15600SDHGWTL1username	
ons15600SDHGWTL1password	
ons15310CLusername	
ons15310CLpassword	
ons15310CLGWTL1username	
ons15310CLGWTL1password	
ons15310CLMLSeriesusername	
ons15310CLMLSeriespassword	
ctm-auth	This flag allows local fallback, which means that the policy server is not reachable when users log into the Prime Optical database.
ext-auth	This flag enables the external authentication feature.
ip-addr-url	IP address of the policy server.
ext-auth-type	SiteMinder protocol version.
sys-admin-enable	Allows the SysAdmin user to always log into Prime Optical.
authorizationport	Policy server port used for authorization.
authenticationport	Policy server port used for authentication.
accountingport	Policy server port used for accounting.
agentname	Name that the policy server uses to identify the Prime Optical agent.
sharedsecret	Default shared secret parameter used to connect to the policy server.
pollingtime	Polling frequency for the policy server to update the parameters.
failover	Number of failover servers.
numserver	Number of servers used for security.
servertimeout	Maximum amount of time to wait for policy server response.

Parameter Name	Definition
minconnection	Minimum number of connections to the policy server.
maxconnection	Maximum number of connections to the policy server.
connectionstep	Step used to connect to the policy server.
resourceget	Resources used by the policy server to trust users or Prime Optical parameters.
rouleget	Action entrusted to the resource by the policy server.
hostconfigfile	Name of the 5.x SiteMinder protocol configuration file.
sm_lib_path	Absolute path to the library.
sm_lib_name	Name of SiteMinder library.
sm_sampleagentip	Range of available IP addresses.
ui	
overwrite-alarm-notes	Enables or disables the ability to overwrite alarm notes created by another user.
overwrite-circuit-notes	Enables or disables the ability to overwrite circuit notes created by another user.
overwrite-job-notes	Enables or disables the ability to overwrite job notes created by another user.
unack-alarm	Enables or disables the ability to unacknowledged alarms that have been acknowledged.
ne-name-truncation	Allows you to specify how node names longer than 25 characters are truncated in the Network Map. If a string is longer than 25 characters, the first 22 characters display followed by an ellipsis (...). You can choose to truncate the first or last characters of the node name.
ne-no-topo-hosts	Allows you to manage the number of CTC instances for optimal CTC performance. The Cisco default is 10 CTC-based NEs. Although Prime Optical can support up to 2,500 CTC-based NEs, CTC cannot effectively manage more than a small subset of these NEs. Consequently, the Prime Optical domain must be partitioned into smaller CTC domains with one instance of CTC launched per CTC domain. The specified number of CTC-based NEs determines the number of GNEs, or topology hosts, that constitute each CTC management domain. CTC uses the GNEs provided to discover and display other nodes that are physically connected to the GNEs. For example, if each CTC-based NE ring contains three NEs (including the GNE), CTC discovers and manages 3 x 10 = 30 nodes when the value is set to 10. In environments where most CTC-based NE rings or subnetworks contain 10 to 16 NEs, set this number to a smaller value, such as 3.
clear-unack-alarms	(Not used)
subnetwork-auto-grouping	Enables or disables the ability to automatically group NEs in subnetworks.
enable-port-state-coloring	When port state coloring is enabled, the NE Explorer shelf views and card-level views report the port and alarm status as a background color.
default-audit-trail-	Enables or disables the audit trail. NEs that are added or discovered will

Parameter Name	Definition
state	be assigned this setting.
ne	
debug-telnet-port	(Not used)
server-monitor	
monitoring-enabled	Enables or disables monitoring.
no-of-threads	Number of threads used for monitoring.
polling-period	Frequency at which monitoring is performed.
Disk_Usage_Enabled	Enables disk monitoring.
Disk_Usage_Critical	Critical threshold value for disk usage.
Disk_Usage_Major	Major threshold value for disk usage.
Disk_Usage_Minor	Minor threshold value for disk usage.
Disk_Usage_Units	Units in which disk usage is measured.
Memory_Usage_SWAP_Enabled	Enables monitoring for swap.
Memory_Usage_SWAP_Critical	Critical threshold value for swap usage.
Memory_Usage_SWAP_Major	Major threshold value for swap usage.
Memory_Usage_SWAP_Minor	Minor threshold value for swap usage.
Memory_Usage_SWAP_Units	Units in which swap usage is measured.
Memory_Usage_RAM_Enabled	Enables monitoring for RAM.
Memory_Usage_RAM_Critical	Critical threshold value for memory usage.
Memory_Usage_RAM_Major	Major threshold value for memory usage.
Memory_Usage_RAM_Minor	Minor threshold value for memory usage.
Memory_Usage_RAM_Units	Units in which memory usage is measured.
CPU_Usage_Enabled	Enables monitoring for CPU usage.
CPU_Usage_Critical	Critical threshold value for CPU usage.
CPU_Usage_Major	Major threshold value for CPU usage.
CPU_Usage_Minor	Minor threshold value for CPU usage.
CPU_Usage_Units	Units in which CPU usage is measured.

Parameter Name	Definition
Prune_Time_Audit _Log_Enabled	Enables monitoring for the prune time audit log.
Prune_Time_Audit _Log_Critical	Critical threshold value for the prune time audit log.
Prune_Time_Audit _Log_Major	Major threshold value for the prune time audit log.
Prune_Time_Audit _Log_Minor	Minor threshold value for the prune time audit log.
Prune_Time_Audit _Log_Units	Units in which the prune time audit log is measured.
Prune_Time_Error_ Log_Enabled	Enables monitoring for the prune time error log.
Prune_Time_Error_ Log_Critical	Critical threshold value for the prune time error log.
Prune_Time_Error_ Log_Major	Major threshold value for the prune time error log.
Prune_Time_Error_ Log_Minor	Minor threshold value for the prune time error log.
Prune_Time_Error_ Log_Units	Units in which the prune time error log is measured.
Prune_Time_Audit _Trail_Log_ Enabled	Enables monitoring for the prune time audit trail log.
Prune_Time_Audit _Trail_Log_ Critical	Critical threshold value for the prune time audit trail log.
Prune_Time_Audit _Trail_Log_Major	Major threshold value for the prune time audit trail log.
Prune_Time_Audit _Trail_Log_Minor	Minor threshold value for the prune time audit trail log.
Prune_Time_Audit _Trail_Log_Units	Units in which the prune time audit trail log is measured.
Prune_Time_Purge _NE_Enabled	Enables monitoring for the prune time purge NE.
Prune_Time_Purge _NE_Critical	Critical threshold value for the prune time purge NE.
Prune_Time_Purge _NE_Major	Major threshold value for the prune time purge NE.
Prune_Time_Purge _NE_Minor	Minor threshold value for the prune time purge NE.
Prune_Time_Purge _NE_Units	Units in which the prune time purge NE is measured.

Parameter Name	Definition
Prune_Time_15min_PM_Enabled	Enables monitoring for the prune time 15-minute PM.
Prune_Time_15min_PM_Critical	Critical threshold value for the prune time 15-minute PM.
Prune_Time_15min_PM_Major	Major threshold value for the prune time 15-minute PM.
Prune_Time_15min_PM_Minor	Minor threshold value for the prune time 15-minute PM.
Prune_Time_15min_PM_Units	Units in which the prune time 15-minute PM is measured.
Prune_Time_1day_PM_Enabled	Enables monitoring for the prune time 24-hour PM.
Prune_Time_1day_PM_Critical	Critical threshold value for the prune time 24-hour PM.
Prune_Time_1day_PM_Major	Major threshold value for the prune time 24-hour PM.
Prune_Time_1day_PM_Minor	Minor threshold value for the prune time 24-hour PM.
Prune_Time_1day_PM_Units	Units in which the prune time 24-hour PM is measured.
Prune_Time_FM_Enabled	Enables monitoring for the prune time FM.
Prune_Time_FM_Critical	Critical threshold value for the prune time FM.
Prune_Time_FM_Major	Major threshold value for the rune time FM.
Prune_Time_FM_Minor	Minor threshold value for the prune time FM.
Prune_Time_FM_Units	Units in which the prune time FM is measured.
Prune_Time_Server_Monitor_Enabled	Enables monitoring for the prune time server monitor.
Prune_Time_Server_Monitor_Critical	Critical threshold value for the prune time server monitor.
Prune_Time_Server_Monitor_Major	Major threshold value for the prune time server monitor.
Prune_Time_Server_Monitor_Minor	Minor threshold value for the prune time server monitor.
Prune_Time_Server_Monitor_Units	Units in which the prune time server monitor is measured.

Parameter Name	Definition
Prune_Time_Job_ Monitor_Enabled	Enables monitoring for the prune time job monitor.
Prune_Time_Job_ Monitor_Critical	Critical threshold value for the prune time job monitor.
Prune_Time_Job_ Monitor_Major	Major threshold value for the prune time job monitor.
Prune_Time_Job_ Monitor_Minor	Minor threshold value for the prune time job monitor.
Prune_Time_Job_ Monitor_Units	Units in which the prune time job monitor is measured.
PM_Collection_Tim e_Enabled	Enables monitoring for the PM collection time.
PM_Collection_Tim e_Critical	Critical threshold value for the PM collection time.
PM_Collection_Tim e_Major	Major threshold value for the PM collection time.
PM_Collection_Tim e_Minor	Minor threshold value for the PM collection time.
PM_Collection_Tim e_Units	Units in which the PM collection time is measured.
Base_Circuit_Creati on_Time_Enabled	Enables monitoring for the base circuit creation time.
Base_Circuit_Creati on_Time_Critical	Critical threshold value for the base circuit creation time.
Base_Circuit_Creati on_Time_Major	Major threshold value for the base circuit creation time.
Base_Circuit_Creati on_Time_Minor	Minor threshold value for the base circuit creation time.
Base_Circuit_Creati on_Time_Units	Units in which the base circuit creation time is measured.
Circuit_Creation_Ti me_Per_Hop_ Enabled	Enables monitoring for the circuit creation time.
Circuit_Creation_Ti me_Per_Hop_ Critical	Critical threshold value for the circuit creation time.
Circuit_Creation_Ti me_Per_Hop_ Major	Major threshold value for the circuit creation time.
Circuit_Creation_Ti me_Per_Hop_ Minor	Minor threshold value for the circuit creation time.

Parameter Name	Definition
Circuit_Creation_Time_Per_Hop_Units	Units in which the circuit creation time is measured.
NE_Synch_Time_Enabled	Enables monitoring for the NE synchronization time.
NE_Synch_Time_Critical	Critical threshold value for the NE synch time.
NE_Synch_Time_Major	Major threshold value for the NE synch time.
NE_Synch_Time_Minor	Minor threshold value for the NE synch time.
NE_Synch_Time_Units	Units in which the NE synch time is measured.
Config_Resynch_Time_Enabled	Enables monitoring for the config resynchronization time.
Config_Resynch_Time_Critical	Critical threshold for the config resynchronization time.
Config_Resynch_Time_Major	Major threshold for the config resynchronization time.
Config_Resynch_Time_Minor	Minor threshold for the config resynchronization time.
Config_Resynch_Time_Units	Units in which the config resynchronization time is measured.
critical-process	
SMService	Indicates whether this process is critical or noncritical.
OracleService	
OSAgent	
ons15454-ne	
ons15454-pm	
ons15200-ne	(Not used)
ons155xx-ne	Indicates whether this process is critical or noncritical.
ons155xx-pm	
ons15454sdh-ne	
ons15454sdh-pm	
ons15216-ne	
ons15600-pm	
ons15600sdh-pm	
ons15305-ne	
ons1530x-pm	

Parameter Name	Definition
snmp-trap-service	Indicates whether this process is critical or noncritical.
gw-corba-service	
gwtl1-service	
database	
db-prune-db05-enabled	Indicates whether or not pruning of the archivelog directory /db05 is enabled. The default is not enabled.
db-prune-db05-threshold	Specifies that pruning of the /db05 directory will occur if the used capacity of /db05 crosses the threshold value. The default value is 80%.
db-prune-db05-backupdirectory	Backup directory for the /db05 directory. The default is /ctm_backup.
db-prune-db05-polling-period	Interval of pruning the /db05 directory if the threshold value is exceeded. The default is 15 minutes.
db-server-configuration	Prime Optical database and server configuration type. Values are small, medium, large, or high-end.
db-archive-15min-pm-days	Number of days of 15-minute PM data to keep in the database. The default is 30 days.
db-archive-1day-pm-days	Number of days of 24-hour PM data to keep in the database. The default is 30 days.
db-archive-fm-days	Number of days of FM data to keep in the database. The default is 7 days.
db-archive-auditlog-days	Number of days of Audit Log data to keep in the database. The default is 7 days.
db-archive-errorlog-days	Number of days of Error Log data to keep in the database. The default is 7 days.
db-archive-audittrail-days	Number of days of audit trail data to keep in the database. The default is 7 days.
db-archive-server-monitor-days	Number of days of server monitor entries to keep in the database. The default is 7 days.
db-archive-server-jobmonitor-days	Number of days of job monitor entries to keep in the database. The default is 7 days.
db-prune-15min-pm-enabled	Flag to check whether data pruning is enabled.
db-prune-1day-pm-enabled	
db-prune-pm-enabled	

Parameter Name	Definition
db-prune-fm-enabled	
db-prune-auditlog-enabled	
db-prune-errorlog-enabled	
db-prune-audittrail-enabled	
db-prune-server-monitor-enabled	
db-prune-server-jobmonitor-enabled	
daily-prune-15min-pm-time	Time of day when data will be pruned.
daily-prune-1day-pm-time	
daily-prune-pm-time	
daily-prune-fm-time	
daily-prune-errorlog-time	
daily-prune-auditlog-time	
daily-prune-audittrail-time	
daily-prune-server-monitor-time	
daily-prune-server-jobmonitor-time	
db-connection-pool-size	Number of connections in the JDBC connection pool. The default is 10.
db-connection-pool-size-pmservice	Number of connections in the JDBC connection pool for the PM service. The default is 10.
dbname	Database name, which is based on the release (for example, CTM8_5 for CTM R8.5).
dbversion	Version indicating the release (for example, 8.5).
db-diskspace-critical-threshold	Critical alarm is raised when used disk space crosses the threshold. The default is 90%.

Parameter Name	Definition
db-diskspace-major-threshold	Major alarm is raised when used disk space crosses the threshold. The default is 80%.
db-diskspace-minor-threshold	Minor alarm is raised when used disk space crosses the threshold. The default is 70%.
db-archivelog-enabled	Whether or not the database archivelog mode option is enabled. The default is false (not enabled).
db-backup-state	State of the database hot backup. The default is 0.
db-export-state	State of the database export. The default is 0.
db-max-sessions	Maximum number of sessions that can be opened with the database. The default is 32 sessions.
db-no-pruning-threads	Number of threads that are configured to run the db prune script. The default is 1.
db-backup-failure-reason	Reason for the hot database backup failure.
db-export-failure-reason	Reason for database export failure.
ha	
max-retries	If the first backup Prime Optical server is not available, this field provides the number of times the client should retry logging in before moving to the next server in the list.
wait-time	If disruption to the primary Prime Optical server occurs, this field provides the number of seconds the client should wait before attempting to log into the backup server.
server-name-list	Lists the Prime Optical servers that Prime Optical clients will log into if access to the primary Prime Optical server is disrupted.
debug-telnet-port	Lists the Telnet port number to use for debugging. The default port number is 9410.