Oracle Financial Services Analytical Applications

Data Model Extension Guidelines Document

Release 8.1.0.0.0

Mar 2021

OFSAA Data Model Extension Guidelines Document

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information on third party licenses, click here.

Document Control

Version Number	Revision Date	Change Log
3.0	May 2020	Migrated the document to a new format.
4.0	August 2020	Added the following section: • Create Subtype and Supertype Relationship.
5.0	September 2020	Added OIDF related content.

Table of Contents

1 Preface

This section provides information about the Oracle Financial Services Data Foundation Data Model Extension Guidelines document.

Topics:

- Purpose of this Document
- Audience
- Access to the Oracle Support
- Related Documents
- Conventions
- Abbreviations

1.1 Purpose of this Document

The purpose of this document is to explain the extension process of the OFSAA data model. The objective is to ensure that the implementation teams extend the OFSAA application as per guidelines, conformance, and standards established for content and design.

1.2 Audience

This document is intended to be used by Oracle Financial Services Analytical Applications Data Foundation System Administrator.

1.3 Access to the Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For more information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1.4 Related Documents

Oracle strives to keep this and all other related documents updated regularly; visit the OFSDF OHC Documentation Library and OIDF OHC Documentation Library to download the latest version available there. The list of related documents is provided as follows:

- Oracle Financial Services Analytical Applications (OFSAA) Data Model Naming Standards Guide Release 8.1.x
- Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x
- OHC Documentation Library for the OFSDF Application Pack Release 8.1.x.x.x product documents:
    - Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.0.0.0

  - Oracle Financial Services Data Foundation Application Pack User Guide Release 8.1.0.0.0
- OHC Documentation Library for the OIDF Application Pack Release 8.1.x.x.x documents:
  - Oracle Insurance Data Foundation Application Pack Installation and Configuration Guide Release 8.1.0.0.0
  - Oracle Insurance Data Foundation Application Pack User Guide Release 8.1.0.0.0
- My Oracle Support for the OFSDF Application Pack Release 8.x technical documents:
  - *Data Model Difference Reports*
  - *Changelog*
  - *SCD Metadata*
  - *T2T Metadata*
  - *Run Chart*
- My Oracle Support for the OIDF Application Pack Release 8.x technical documents:
  - *Data Model Difference Reports*
  - *Changelog*
  - *SCD Metadata*
  - *T2T Metadata*
  - *Run Chart*

1.5 Conventions

The following text conventions are used in this document.

Table 1: Document Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, file names, text that appears on the screen, or text that you enter.
Hyperlink	Hyperlink type indicates the links to external websites, internal document links to sections.

1.6 Abbreviations

The following table lists the abbreviations used in this document.

Table 2: Abbreviations

Conventions	Description
MOS	My Oracle Support
OFSAA	Oracle Financial Services Analytical Applications
OFSDF	Oracle Financial Services Data Foundation
OIDF	Oracle Insurance Data Foundation
OHC	Oracle Help Center

2 Overview of the Data Model

The OFSAA data model is organized into distinct areas based on the usage and function performed by each area of the model.

Topic:

- Overview of OFSAA Data Model

2.1 Overview of OFSAA Data Model

The OFSAA data model is organized into distinct areas based on usage and function performed by each area of the model.

The data model conforms to the data architecture processes for each function and the modeling methodology employed in each of these functional areas representing the usage pattern.

3 Structure of OFSAA Data Model

This section provides information about the structure of the OFSAA Data Model.

Topics:

- Common Staging Area
- Application-Specific Processing Area
- Common Results Area

3.1 Common Staging Area

The common staging area is the single source of data for all of the OFSAA. The primary purpose of staging is data acquisition and integration. The common staging area is a consolidation of entities fact, dimensions and others, which is completely de-normalized in structure and bears no relationship to one another in the physical model. Logical relationships exist and are established and validated through the staging area processes such as the Value code Mapping, Data Quality and GL Reconciliation processes.

3.2 Application-Specific Processing Area

The application-specific processing area is a collection of application-specific entities used for incoming data, Application maintenance and Application metadata. The modelling methodology employed is based on the processing needs of each Application. The modelling methodology varies from Application to Application. Various degrees of normalization to de-normalization and dimensional modelling structures are employed. The modelling in the processing area is governed by the needs of specific processing engines. Processing engines have evolved at different points of time and have different data needs structurally.

3.3 Common Results Area

The resulting area is a rich combination of source data, value-added metrics from each of the analytical applications and supports the derivation of higher-order value-added metrics across conformed dimensions spanning multiple functional domains regulatory, economic and more.

The common results area is a conformed set of facts and dimensions that combine results from various applications and create an integrated result environment for analytics across applications. Integrated reporting at transaction, account, and customer granularities are managed through an integrated common result modelling approach.

4 Extension Process

The Data model life cycle process involves understanding of data requirements, data architecture, and data flow requirements of the applications that are being implemented.

The governance process is broadly described as follows:

- Data Requirements

  The identification of the additional data requirements for the implementation of the application is determined through customization or configuration documents. Use-cases have to be established for different areas in the OFSAA model.

- Data Flow

  Data flow needs to be established for these additional data requirements.

- Model Design

  Model extensions to be designed based on the guidelines provided for each area. Extensions should also be raised as Enhancement Requests to respective applications through the standard support process. Enhancement Requests will be analyzed by the Product Engineering team and design approaches will be provided where applicable. Review the data model customization review template attached in the Appendix.

- Model Build

  Model extensions to be built based on the guidelines provided for each area.

  The supported tools are:

  - Erwin
  - Oracle SQL Data Modeler

  For information about extension through Oracle SQL Data Modeler, see Oracle SQL Data Modeler.

5 Design Guidelines for Staging Area

The following are the design guidelines for the staging area:

- Subject Area: The staging area is designed to hold data related to multiple applications. Entities are typically shared across subject areas.
- Entities: Typically, the Staging area will have wide tables to accommodate data requirements related to multiple OFSA applications in a single row.
- Relationships: The staging area model should not contain any relationships amongst entities. Referential integrity checks are performed as part of Data Quality checks.
- No redundancy of information: Information required for multiple applications need to be captured only once, in a way suitable to all the applications.
- Multiple periods: Staging is designed to capture and store data for multiple periods. This will be useful to support operational requirements such-as re-run of processes for earlier periods.
- No update: Staging data, as received from source systems should not be modified. In case of exceptions that require data updates, the following are the options:
  - Data needs to be corrected in source systems and re-loaded into staging.
  - Data quality framework should provide the ability to identify data quality issues and correct the data with an audit trail.
- Alpha-numeric Codes:
  - Master, data should be designed to hold alpha-numeric codes.
  - Stage tables corresponding to the dimension column should also be explicitly designed as alpha-numeric.
- Static Information: Data created and maintained within the application such as setup, static master information, and so on should not be represented in staging.
- Impact Analysis: The staging area data model is shared across multiple OFSA applications. Hence, when a property of any entity or attribute is modified, impact analysis needs to be performed with application owners before performing the change. A metadata browser can be used for performing the impact analysis.
- Data Origin: Each staging entity needs to include a column for storing the source of data. This information is useful to understand the lineage of data.
- Download Specification: The staging area is designed to receive data from source systems. Download specification for each application can be generated from the staging area model. User-defined properties for table and column are referenced in the generation of download specs. Mention UDP values for any attributes added as part of the extension process.
- Naming Standards: Staging data model should follow naming standards as detailed in the Appendix.
- Customization related User Defined Properties: Column level UDP’s named ‘Custom’ (NO/YES), ‘Customization Reason’ (Pending ER, Specific to Customer, Modified data element) and ‘Type of Change’ (Length, Datatype, Logical Name, Description, Addition) have to be specified for attributes added/modified as part of the extension process.

6 Design Guidelines for Processing Area

The data model for the processing area is designed as per the requirements for the applications. The following are the guidelines for processing area:

- Design: The processing area for each application is designed for the optimal performance of the calculation engines respective to the applications.
- Performance: Data will be moved from staging into the processing area. Typically, processing involves multiple SELECT and UPDATE SQL statements and hence data model should be designed to enable high-performance SELECT and UPDATE operations on the table.
- Relationships: The processing area data model should enforce non-identifying and identifying relationships amongst entities as applicable.
- Naming Standards: Processing data model should follow naming standards as detailed in the Appendix.
- Diagram Representation: Processing area entities should be grouped logically and represented in the erwin data model. In erwin, the default display view should have primary key designators.
- Limited History: Processing is designed to support the storage of limited history. This historical data is useful for verification of process outputs for earlier dates if required.

7 Design Guidelines for Reporting Area

The purpose of the Reporting data model is to perform the following:

- Address application-specific reporting requirements
- Address integrated reporting across applications
- Drill through and drill-across in reports

The following are the guidelines for reporting area:

1. Application-specific Reporting: Application-specific reporting area can have specific fact tables to store measures related to that application. However, dimensions that are referred to in any of the application-specific reporting need to be conformed across all the applications.
2. Integration of Results: Integration of results is achieved through common structures for account and customer as described as follows:
   - Common Account Summary
     - Consolidates measures at the granularity of the account that is output from different applications.
     - Any account-level reporting should be driven out of an integrated Account Summary set of entities. Account attributes captured from the source system and those which do not vary between runs need to be part of the Common Account Summary entity. Any application-specific attributes and measures that are computed by the application need to be part of application-specific account summary entities. Such a data model design enables integrated reporting of account information.
   - Common Customer Summary
     - Consolidates measures at the granularity of the customer that are output from different applications.
     - Any customer level reporting should be driven out of an integrated Customer Summary set of entities. Customer attributes captured from the source system and those, which do not vary between runs, need to be part of the Common Customer Summary entity. Any application-specific attributes and measures that are computed by the application need to be part of application-specific customer summary entities. Such a data model design enables integrated reporting of customer information.
3. Data Modeling Technique: Reporting data model is structured as typical star schemas' corresponding to analytical reporting requirements.
4. Aggregate Entities: Depending on performance requirements for each application, information can be reported out of aggregate entities. However, a drill through to the base entity from the aggregate entity is mandatory.
5. Reporting and Local Currency: Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.
6. Dimensional Conformance: Same dimensional information should not be represented in different forms within the reporting area.

7. Slowly Changing Dimensions: Dimensions that need to track changes in attributes need to be handled through slowly changing dimensions (SCD) process. Such dimension entities need to have a standard set of attributes required for the SCD process.
8. Measures: Same fact information should not be represented in multiple fact tables.
9. Intermediate Results: Reporting data model should hold both final and intermediate computed outputs of each application.
10. Performance: Data is going to be loaded into the reporting area from processing. Since reporting will involve multiple SELECT queries, the data model should be designed to enable high-performance SELECT operations on the table. Also, since the reporting model provides integrated information across applications, entities will be wide compared to entities modelled in the processing area.
11. Relationships: The reporting area data model should enforce non-identifying and identifying relationships between fact and dimension entities as applicable.
12. Naming Standards Reporting data model should follow naming standards as detailed in the Appendix.
13. Subject Area: Subject areas should be created for a logical set of reports. Ideally, each subject area should have a single fact and multiple dimension entities.
14. Diagram Representation: Within the subject area, fact entities should be represented in the center surrounded by dimension entities. In erwin, the default display view should have primary key designators.
15. Full History: The reporting area model is designed to support the storage of full history.

8 Configuring Entity UDPs for Application Data Interface of Data Integration Hub

In OFSAA DIH Application, the uploaded data model is visible as Application Data Interface (ADI). These ADI names are given through erwin data model entity UDP for each table. For any new table addition, which is added as a part of customization must consist of a valid value in the following entity UDPs to get the proper ADI names in DIH.

- OFSAA Data Interface Name: This Entity UDP expects the logical name of the table or logical name for a group of tables, which will be displayed as an ADI in DIH. This UDP is mandatory for all Stage and Results tables to get a valid ADI in DIH, which corresponds to each table in the data model.
- OFSAA Data Interface Subtype: This Entity UDP expects the logical name of a table where the table is part of a group of tables where we have a group name in OFSAA Data Interface Name UDP. This UDP is optional and applicable only if you have a one OFSAA Data Interface Name against a group of tables.

For illustration, see the following samples for OFSDF:

- STG_CASA, STG_LOAN_CONTRACTS, STG_TD_CONTRACTS, and FCT_COMMON_ACCOUNT_SUMMARY are defined as a group of tables, and the UDPs for the tables are given in the following table:

Table 3: Sample OFSDF Entity UDP for a group of tables

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_CASA	Customer Account	CASA Contracts
STG_LOAN_CONTRACTS	Customer Account	Loan Contracts
STG_TD_CONTRACTS	Customer Account	Term Deposits
FCT_COMMON_ACCOUNT_SUMMARY	Customer Account	Common Account Summary

- If there are Stage and Fact tables for a similar purpose, you must define the same name for the OFSAA Data Interface Name and OFSAA Data Interface Subtype.

Table 4: Sample OFSDF Entity UDP for a similar purpose Stage and Fact tables

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_ACCT_FEATURE_MAP	Account Feature Map	Account Feature Map
FCT_ACCOUNT_FEATURE_MAP	Account Feature Map	Account Feature Map

- If you have independent tables such as only one table, which is either a stage or dim or fact, then the OFSAA Data Interface Subtype can be left blank.

Table 5: Sample OFSDF Entity UDP for an independent table

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_VEHICLE_MASTER	Vehicle Master	
FCT_TXN_CHANNEL	Transaction Channel	
DIM_ACCOUNT	Account Dimension	

For illustration, see the following samples for OIDF:

- STG_ANNUITY_CONTRACTS, STG_LOAN_CONTRACTS, STG_RETIREMENT_ACCOUNTS, and FCT_COMMON_POLICY_SUMMARY are defined as a group of tables, and the UDPs for the tables are given in the following table:

Table 6: Sample OIDF Entity UDP for a group of tables

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_ANNUITY_CONTRACTS	Customer Account	Annuity Contracts
STG_LOAN_CONTRACTS	Customer Account	Loan Contracts
STG_RETIREMENT_ACCOUNTS	Customer Account	Retirement Accounts
FCT_COMMON_POLICY_SUMMARY	Customer Account	Common Policy Summary

- If there are Stage and Fact tables for a similar purpose, you must define the same name for the OFSAA Data Interface Name and OFSAA Data Interface Subtype

Table 7: Sample OIDF Entity UDP for a similar purpose Stage and Fact tables

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_PARTY_ROLE_MAP	Party Role Map	Party Role Map
FCT_PARTY_ROLE_MAP	Party Role Map	Party Role Map

- If you have independent tables such as only one table, which is either a stage or dim or fact, then the OFSAA Data Interface Subtype can be left blank.

Table 8: Sample OIDF Entity UDP for an independent table

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_VEHICLE_MASTER	Vehicle Master	
FCT_CLAIM_TRANSACTION	Claim Transaction	
DIM_ACCOUNT	Account Dimension	

9 Create Subtype and Supertype Relationship

This section provides information to create the Subtype and Supertype relationship between two entities.

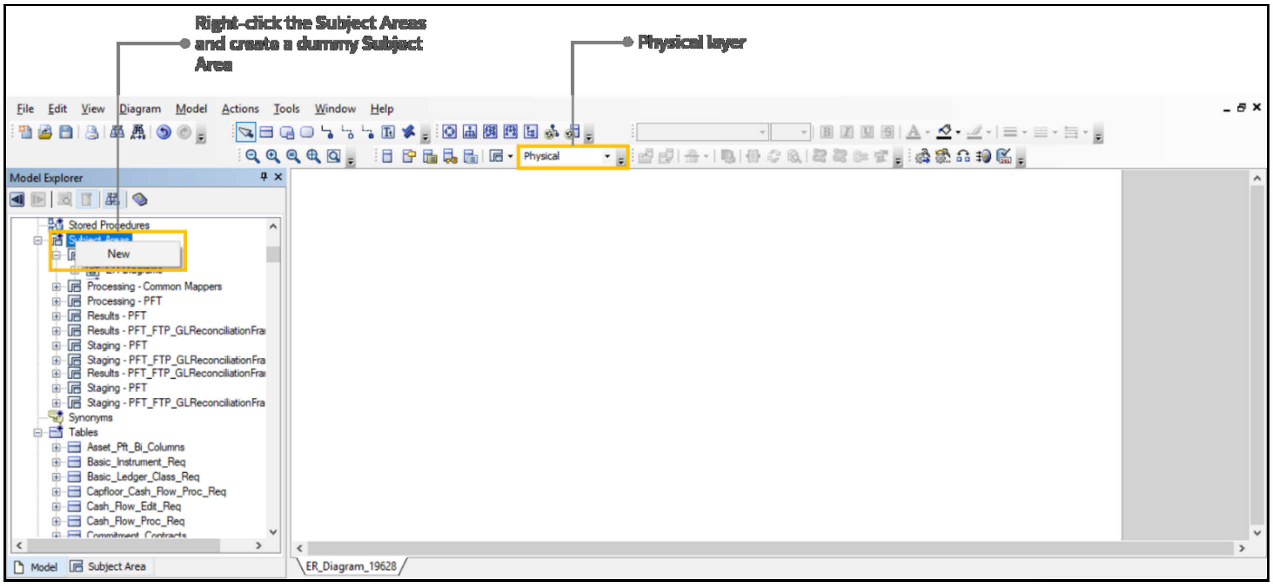
A scenario where the Subtype and Supertype relationship can be created is when an entity's (parent entity) columns must be inherited into another (child entity).

For information about selecting and using the Subtype and Supertype in the data model merge process, see the Data Model Merge Guidelines section.

To create a Subtype and Supertype relationship between two entities, follow these steps:

1. Open the **erwin Data Modeler** application. Select the **Physical** layer. In the **Model Explorer** section, right-click the **Subject Areas** and click **New** to create a dummy Subject Area.

   Figure 1: Open erwin Data Modeler and create a new Subject Area

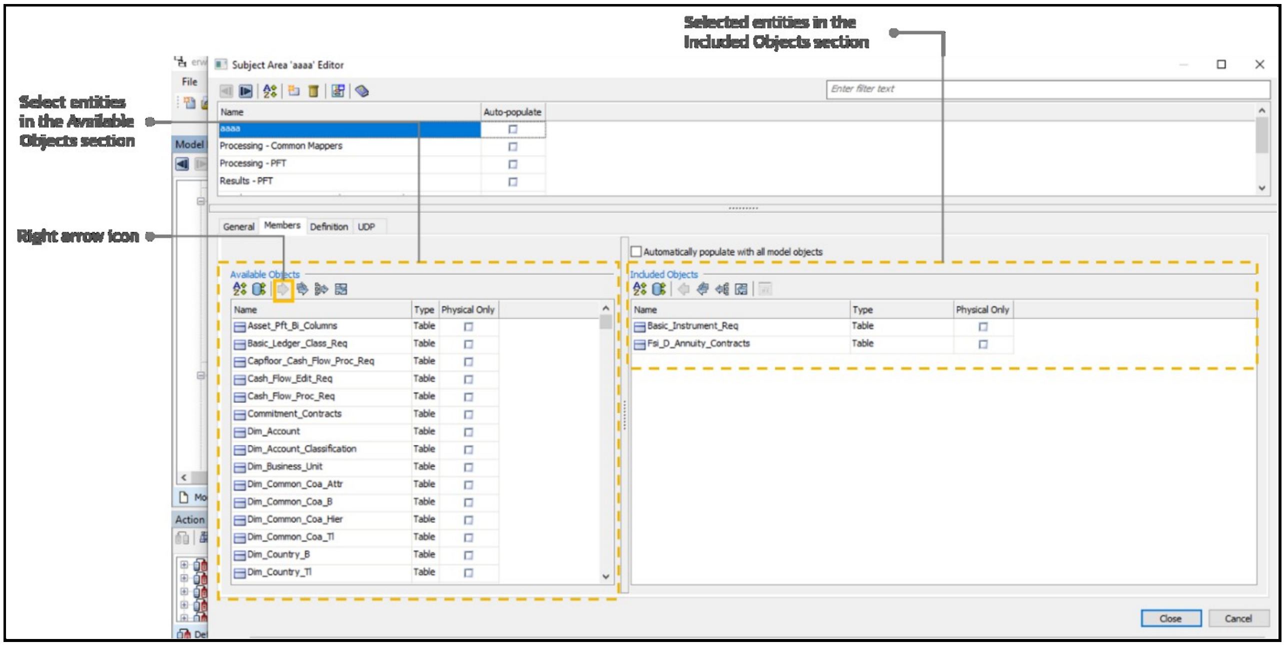


Right-click the Subject Areas and create a dummy Subject Area

Physical layer

2. The **Subject Area Editor** window opens. In the Available Objects section, select the entities that must be added into the newly created Subject Area, and click the right arrow icon to move the selected objects into the Included Objects section. Click **Close**.

Figure 2: Select entities for the new Subject Area


Selected entities in the Included Objects section
Select entities in the Available Objects section
Right arrow icon
Subject Area 'aaaa' Editor
Enter filter text
General
Members
Definition
UDP
Close
Cancel

Name	Auto-populate
aaaa	□
Processing - Common Mappers	□
Processing - PFT	□
Results - PFT	□

□ Automatically populate with all model objects

Available Objects

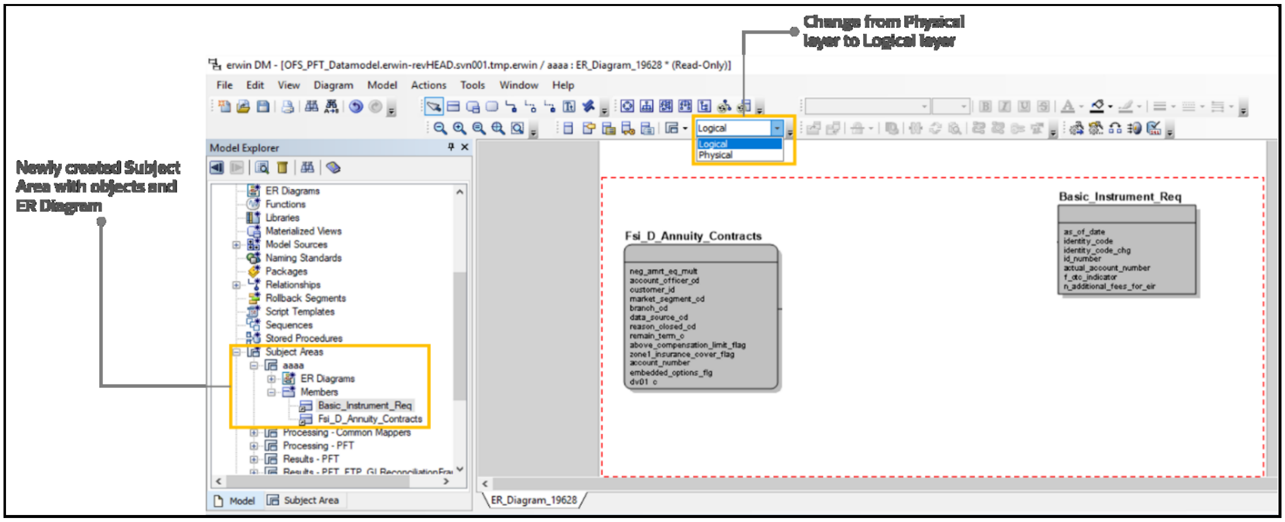
Name	Type	Physical Only
Asset_Pft_Bi_Columns	Table	□
Basic_Ledger_Class_Req	Table	□
Capfloor_Cash_Flow_Proc_Req	Table	□
Cash_Flow_Edit_Req	Table	□
Cash_Flow_Proc_Req	Table	□
Commitment_Contracts	Table	□
Dim_Account	Table	□
Dim_Account_Classification	Table	□
Dim_Business_Unit	Table	□
Dim_Common_Coa_Attr	Table	□
Dim_Common_Coa_B	Table	□
Dim_Common_Coa_Hier	Table	□
Dim_Common_Coa_Tl	Table	□
Dim_Country_B	Table	□
Dim_Country_Tl	Table	□

Included Objects

Name	Type	Physical Only
Basic_Instrument_Req	Table	□
Fsi_D_Annuity_Contracts	Table	□

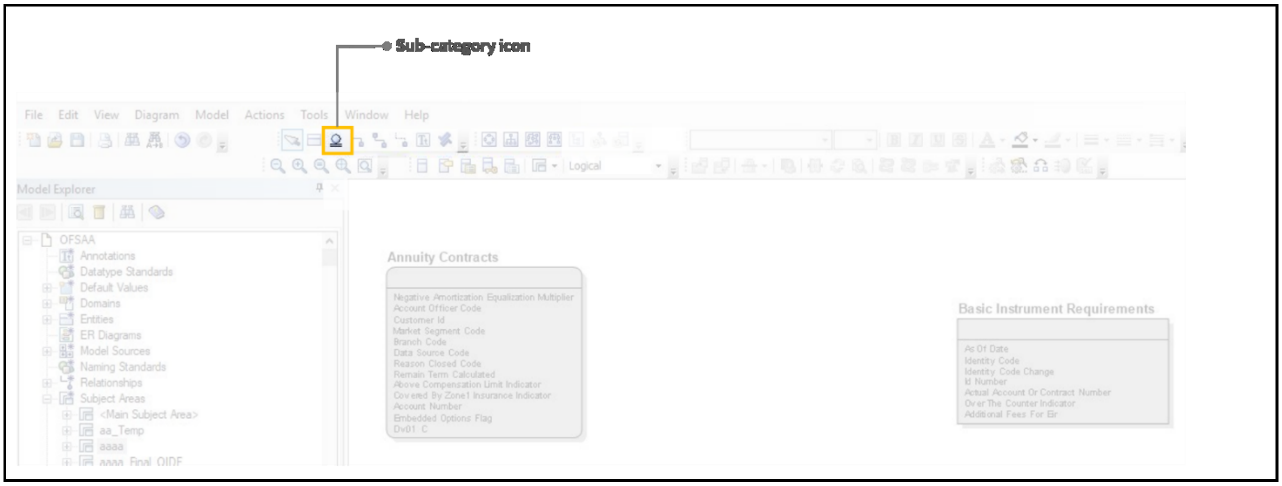
3. On the ER_Diagram canvas, the selected entities are displayed, and the newly created Subject Area with objects is listed in the Model Explorer section. Select the **Logical** layer.

Figure 3: Select the Logical layer



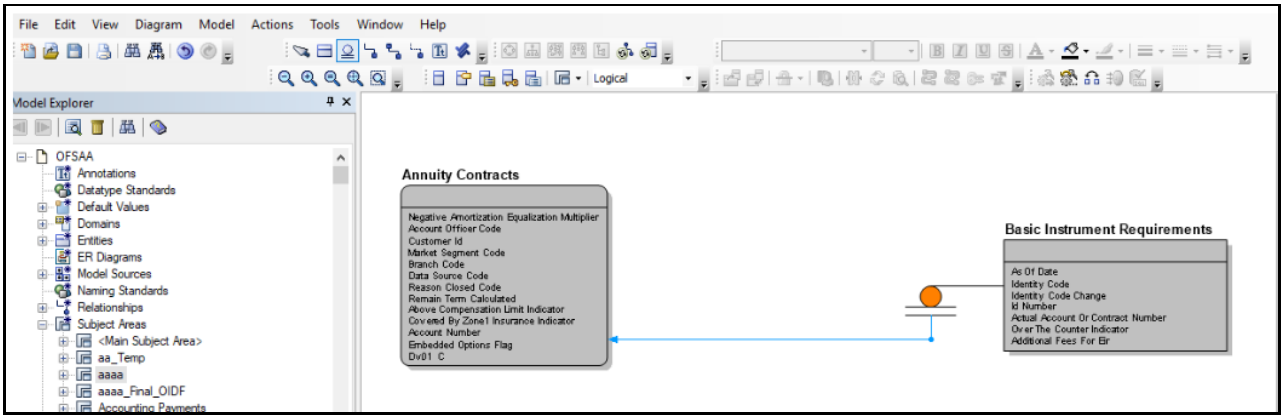
4. Select the parent entity on the ER_Diagram canvas. Click **Sub-category**.

Figure 4: Select the Sub-category icon



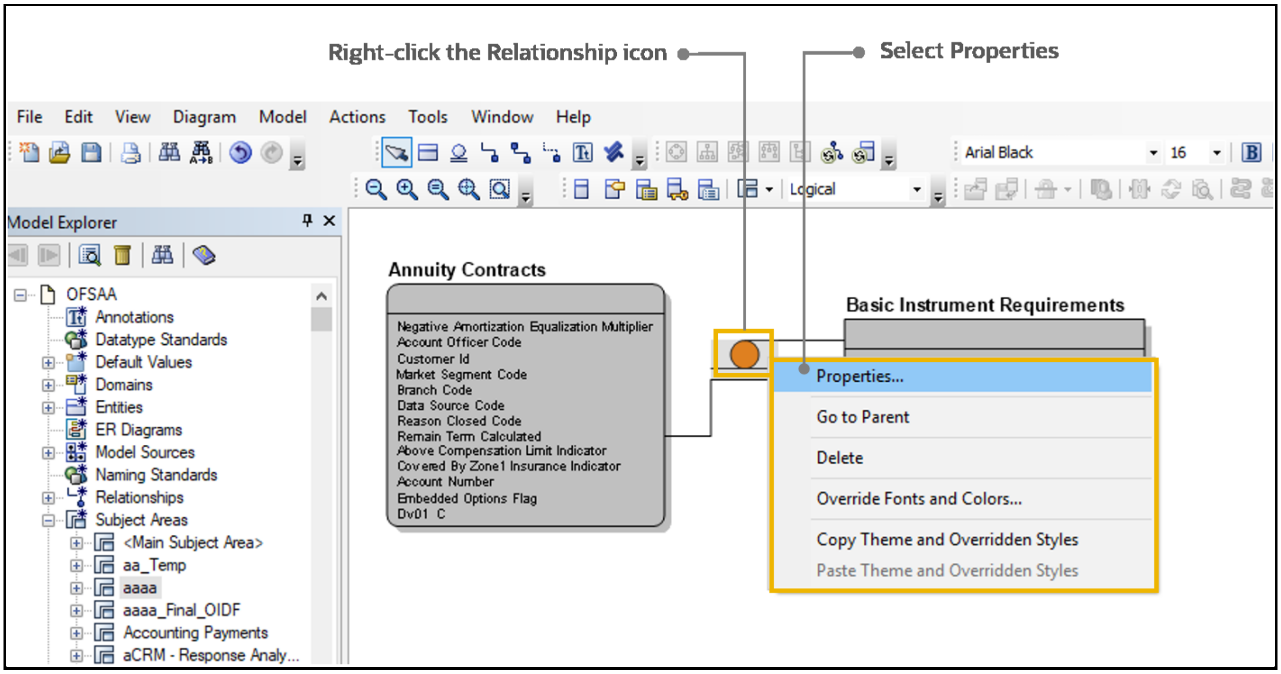
5. First, click the parent entity in the relationship. Then click the child entity. In the illustration, the parent entity Basic Instrument Requirements table is Supertype and the child entity Annuity Contracts table is Subtype.

Figure 5: Create the Supertype and Subtype relationship



6. To see the Supertype and Subtype relationship details of the parent-child entities, right-click **Relationship** connecting the two entities, and select **Properties**.

Figure 6: Select Properties

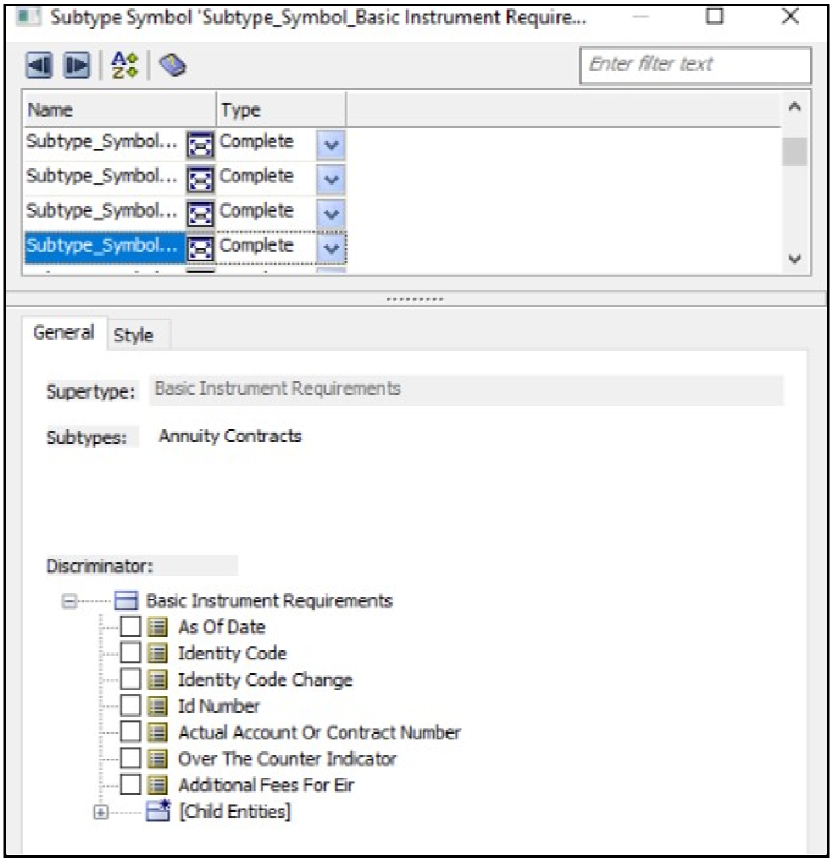


Right-click the Relationship icon

Select Properties

7. The **Properties** window opens. You can verify the Supertype and Subtype relationship details of the parent-child entities.

Figure 7: The Properties window



Name	Type
Subtype_Symbol...	Complete
Subtype_Symbol...	Complete
Subtype_Symbol...	Complete
Subtype_Symbol...	Complete

Supertype: Basic Instrument Requirements

Subtypes: Annuity Contracts

Discriminator:

- Basic Instrument Requirements
  - □ As Of Date
  - □ Identity Code
  - □ Identity Code Change
  - □ Id Number
  - □ Actual Account Or Contract Number
  - □ Over The Counter Indicator
  - □ Additional Fees For Eir
  - [Child Entities]

8. Save the settings. The Supertype and Subtype relationship is created between two entities.

10 Sample Extensions to Model

The following table summarizes some of the common types of extensions to model:

Table 9: Common types of extensions to the data model

Extension type	Staging	Processing or Reporting
Add new dimension	• Create an entity named STG_<DIMENSION NAME>_MASTER. • The primary key for the master entity should be the alphanumeric dimension key identifier and MIS Date. • Include attributes for the dimension as columns in the master table.	• Create a set of 4 tables if the dimension/hierarchy needs to be maintained within OFSAA – DIM_<DIMENSION NAME>_B – stores members; DIM_<DIMENSION NAME>_TL – stores translated members; DIM_<DIMENSION NAME>_ATTR • - stores values of attributes of dimension members; DIM_<DIMENSION NAME>_HIER • - stores hierarchy. • Create the dimension table if the dimension needs to be used in reporting. Design the structure of the dimension table depending on the requirement to have a level based or parent-child hierarchies. The dimension table should have support for slowly changing dimensions. • Configure data movement processes to load dimension tables in processing and reporting.
Add new Attributes to dimension	Enhance the staging master table to include this attribute.	• In case the dimension is maintained within OFSAA, configure this attribute for the dimension. The model need not be changed. • In case the dimension is used for reporting and attribute is required for reporting, then include this attribute in the dimension table as a column. • Modify data movement processes to load this additional attribute into dimension tables in processing and reporting.
Modify data type/data length for any attributes of the dimension	Check if this data type/data length change does not impact any applications.	• Modify the data type/data length in the corresponding dimension table in reporting. • Some of the dimensional attributes might be de-normalized in fact entities for performance reasons. Datatype/Data length needs to be • modified in all such cases.

Extension type	Staging	Processing or Reporting
Add new entities for receiving hierarchy data from any MDM system	Create a set of four entities named STG_<DIMENSION NAME>_B – holds the dimension members. STG_<DIMENSION NAME>_TL – holds the translated dimension members. STG_<DIMENSION NAME>_ATTR – holds the values of attributes of the dimension. STG_<DIMENSION NAME>_HIER - holds the hierarchy	• Create a set of 4 tables if the dimension/hierarchy needs to be maintained within ▪ OFSAA – DIM_<DIMENSION NAME>_B stores members ▪ DIM_<DIMENSION NAME>_TL ▪ stores translated members ▪ DIM_<DIMENSION NAME>_ATTR stores values of attributes of dimension members ▪ DIM_<DIMENSION NAME>_HIER stores hierarchy. • Create the dimension table if the dimension needs to be used in reporting. Design the structure of the dimension table depending on the requirement to have a level based or parent-child hierarchies. The dimension table should have support for slowly changing dimensions. • Configure data movement processes to load dimension tables in processing and reporting.
Add attributes to product processor	• Make sure that attributes that are being added do not exist in any form within the entity. • Extend product processor to hold these attributes.	• Extend processing fact tables to hold this information. Modify T2Ts to move this information from staging. • Extend a common account summary entity to hold this information if the attribute is required for reporting. • Modify T2Ts to move this information from staging to reporting.
Add attributes to any fact entity	• Make sure that attributes that are being added do not exist in any form within the entity. • Extend the entity with this additional attribute.	• Extend the processing/reporting entity to include this attribute. • Modify the data movement processes to move the information from staging.
Make changes to the primary key of the fact entity	• Make sure that the impact of changing the granularity is carried for all the OFSA applications that are being implemented. • Extend the fact entity in staging to include attributes as part of the PK. • Modify the data movement processes to move data as per the processing/reporting model.	• Make sure that the impact of changing the granularity is carried for all the OFSA applications that are being implemented. • Any primary key change in processing/reporting will impact OFSA applications. Hence, raise service requests in product support with the use-cases.

Extension type	Staging	Processing or Reporting
Add a new entity	• Make sure that the information to be captured in the new entity does not exist elsewhere in the staging model. • Modify the data movement processes to move data as per the processing/reporting model.	• In cases where data captured in the new entity in staging needs to be moved to an existing processing/reporting entity, modify the data movement processes. • In cases where there is a need to represent fact information in a separate entity, extend the processing/reporting model to create this entity. Build data movement processes to move data from staging.

11 Oracle SQL Data Modeler

Download the patch **29467329** from My Oracle Support (MOS) and follow the installation steps mentioned in the readme file attached with the patch. To configure and extend the model, follow the steps mentioned in the following sections. For more information, see OFSAA Data Model Extensions through SQL Data Modeler (Doc ID 2521335.1) on My Oracle Support.

> **NOTE** Before proceeding with the model customization procedure, take a backup of the `DB.xml` file, and config and atomic schemas.

Topics:

- Deploying the SQL Data Modeler Configuration Files
- Generating Customized Model

11.1 Deploying the SQL Data Modeler Configuration Files

Follow these steps to deploy SQL Data Modeler configuration files:

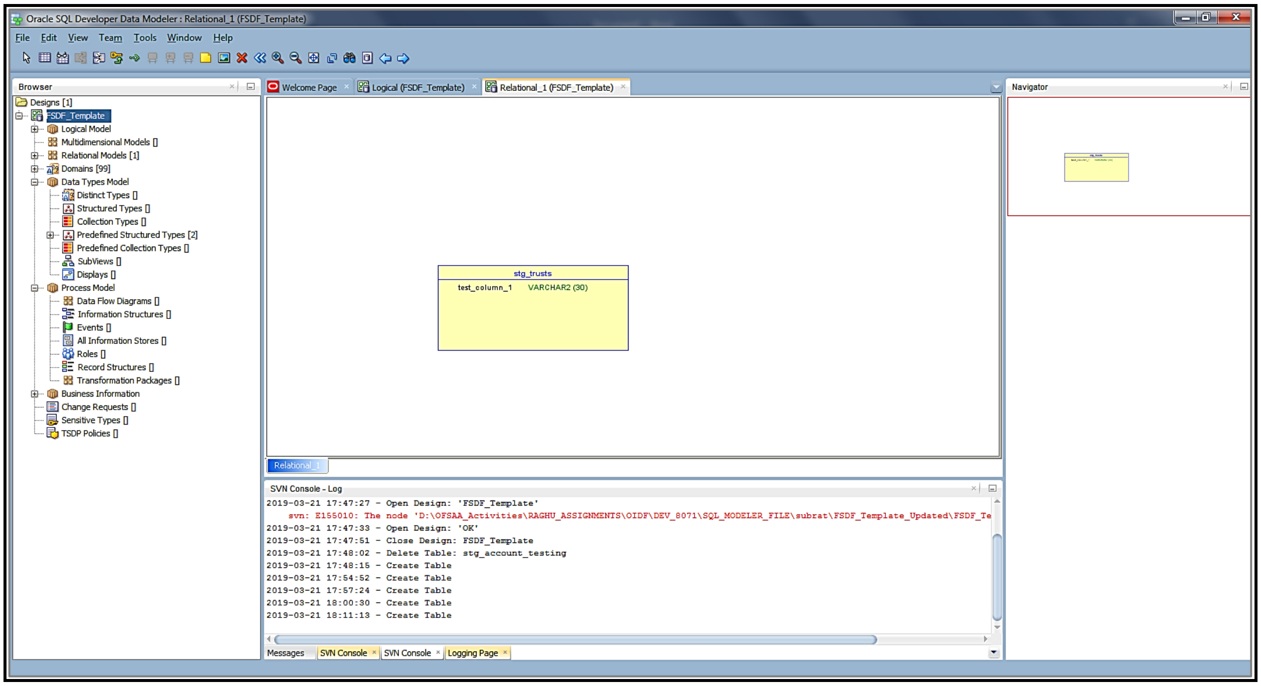
1. Log in to the SQL Developer Data Modeler application.
2. Copy the downloaded file `defaultdomains.xml` available at `<>/OFS_BFND/SQL_Modeler_Conf_Files/types` to `$fic_home /ficapp/common/FICServer/conf/SQLDM`.
3. Rename the `defaultdomains.xml` file (from the previous step) to `OFSAA_DOMAINS_MASTER.xml`.
4. Repeat the steps 1 and 2 each time, in case of changes, to the `defaultdomains.xml` file.
5. Perform the model changes as required.

11.2 Generating Customized Model

These are the generic tasks to generate a customized model:

- Add the Custom Column

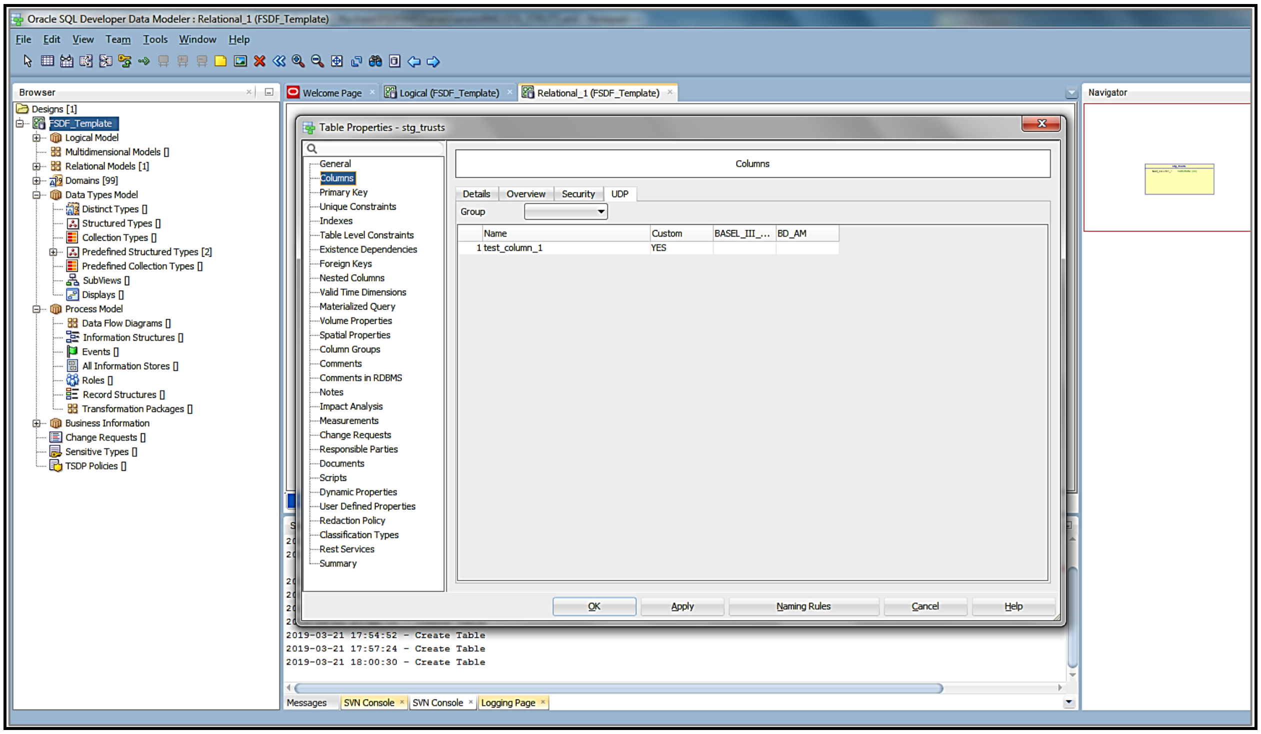
Figure 8: Add the custom column



```
2019-03-21 17:47:27 - Open Design: 'FSDF_Template'
   svn: E155010: The node 'D:\OFSAA_Activities\RAGHU_ASSIGNMENTS\OIDF\DEV_8071\SQL_MODELER_FILE\subrat\FSDF_Template_Updated\FSDF_Te
2019-03-21 17:47:33 - Open Design: 'OK'
2019-03-21 17:47:51 - Close Design: FSDF_Template
2019-03-21 17:48:02 - Delete Table: stg_account_testing
2019-03-21 17:48:15 - Create Table
2019-03-21 17:54:52 - Create Table
2019-03-21 17:57:24 - Create Table
2019-03-21 18:00:30 - Create Table
2019-03-21 18:11:13 - Create Table
```

- Set the User-defined property (UDP) for Column to YES.

Figure 9: Set the column UDP to YES

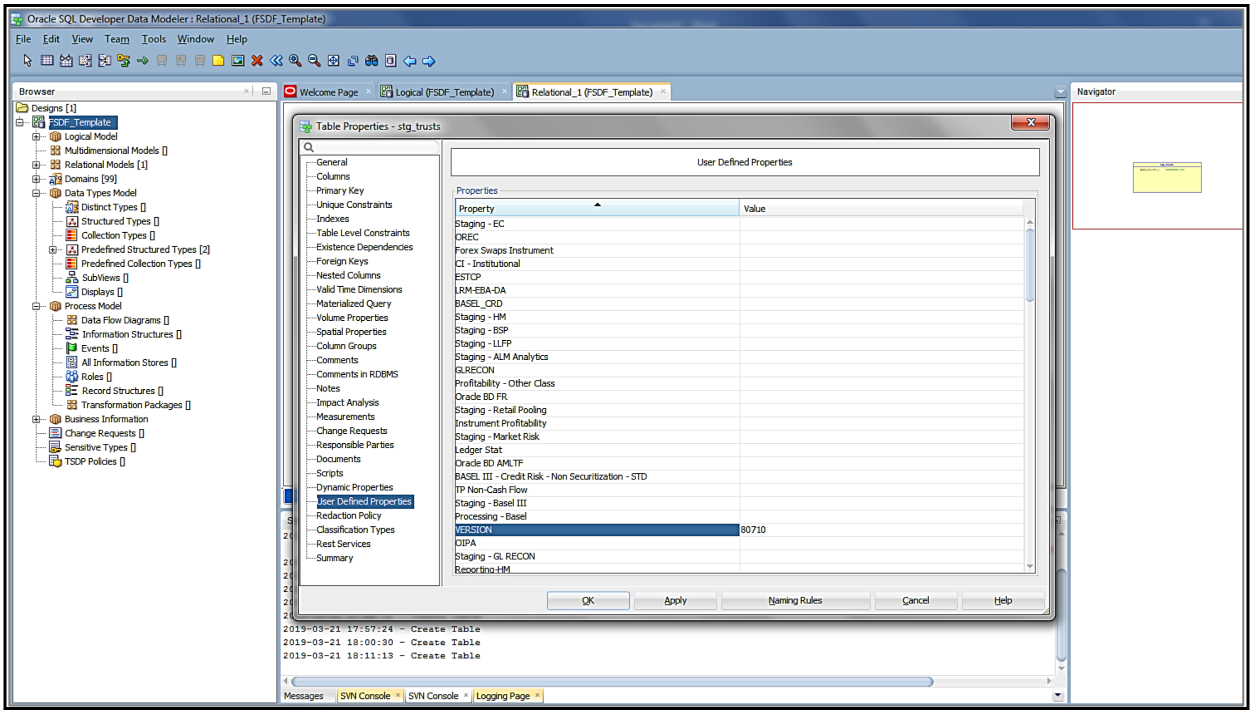


	Name	Custom	BASEL_III_...	BD_AM
1	test_column_1	YES		

```
2019-03-21 17:54:52 - Create Table
2019-03-21 17:57:24 - Create Table
2019-03-21 18:00:30 - Create Table
```

- Set Version UDP value to the current version.

Figure 10: Set Version UDP value to the current version



Property	Value
Staging - EC	
OREC	
Forex Swaps Instrument	
CI - Institutional	
ESTCP	
LRM-EBA-DA	
BASEL_CRD	
Staging - HM	
Staging - BSP	
Staging - LLFP	
Staging - ALM Analytics	
GLRECON	
Profitability - Other Class	
Oracle BD FR	
Staging - Retail Pooling	
Instrument Profitability	
Staging - Market Risk	
Ledger Stat	
Oracle BD AMLTF	
BASEL III - Credit Risk - Non Securitization - STD	
TP Non-Cash Flow	
Staging - Basel III	
Processing - Basel	
VERSION	80710
OIPA	
Staging - GL RECON	
Reporting-HM	

```
2019-03-21 17:57:24 - Create Table
2019-03-21 18:00:30 - Create Table
2019-03-21 18:11:13 - Create Table
```

12 Data Model Merge Guidelines

This section details the instructions to merge two data models. To merge the two data models:

1. Open the erwin model.
2. Navigate to the Actions menu and click the **Complete Compare** option.
3. The **Complete Compare** section opens, which consists of seven input forms.
4. To access any of the seven input forms, in the left panel of the Complete Compare section, click the required form title. The seven input forms are:
   - Overview
   - Left Model
   - Right Model
   - Left Object Selection
   - Right Object Selection
   - Type Selection
   - Advanced Options
5. **Overview:** This form provides a brief description of the Complete Compare function
6. **Left Model:** Using the **Left Model** input form, you can select a data model for the left side of the comparison in the Complete Compare section.

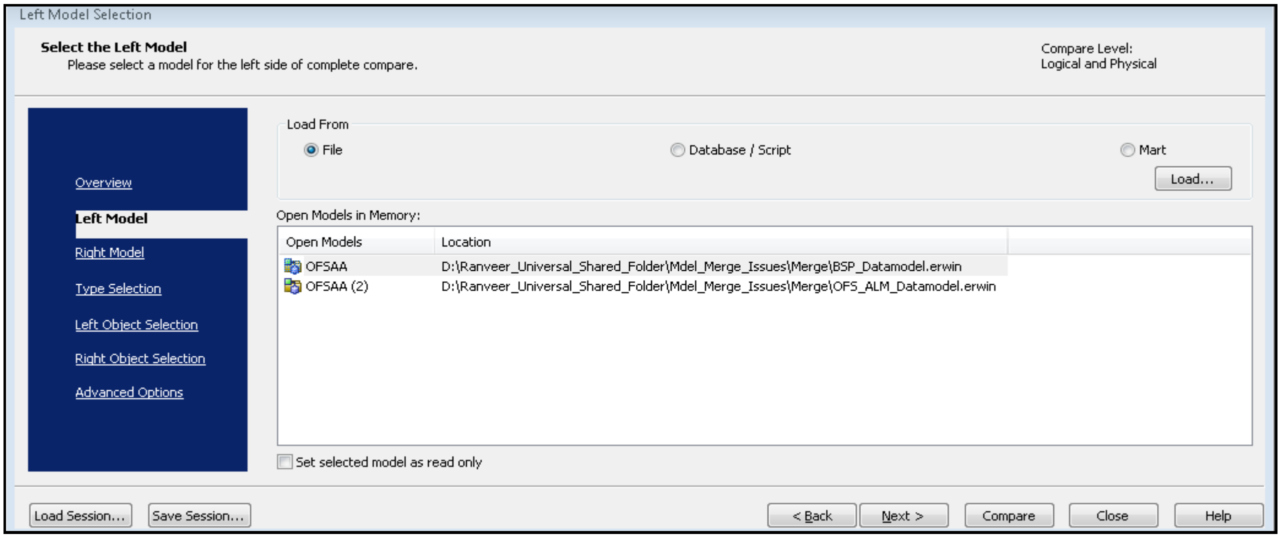
   To select the required model for the left side of the comparison, click **Load**. The **File Open** dialog appears. **Search** for the required data model file, select it and click **Open**. The selected data model loads into the Complete Compare section. You can see the selected data model in the Open Models in a **Memory** list box.

   Alternatively, you can locate the required data model file on your computer and double-click to open it. The selected data model opens by default in the erwin model. Alternatively, if available, you can select the required data model in the Open Models in a **Memory** list box.

   If the default selection of the data model for the left side of the comparison is not correct, follow step 4 to select the required data model. Else, click **Next** to go to the **Right Model** input form.

NOTE When you launch the Complete Compare function with one data model, that data model loads into the **Right Model** form of the Complete Compare section. When you open the second data model for comparison, the existing data model in the **Right Model** form moves to the **Left Model** form and the second data model loads into the **Right Model** form.

Figure 11: Left Model input form



Left Model Selection

Select the Left Model

Please select a model for the left side of complete compare.

Compare Level:
Logical and Physical

Load From

● File ○ Database / Script ○ Mart

Open Models in Memory:

Open Models	Location
OFSAA	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin
OFSAA (2)	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin

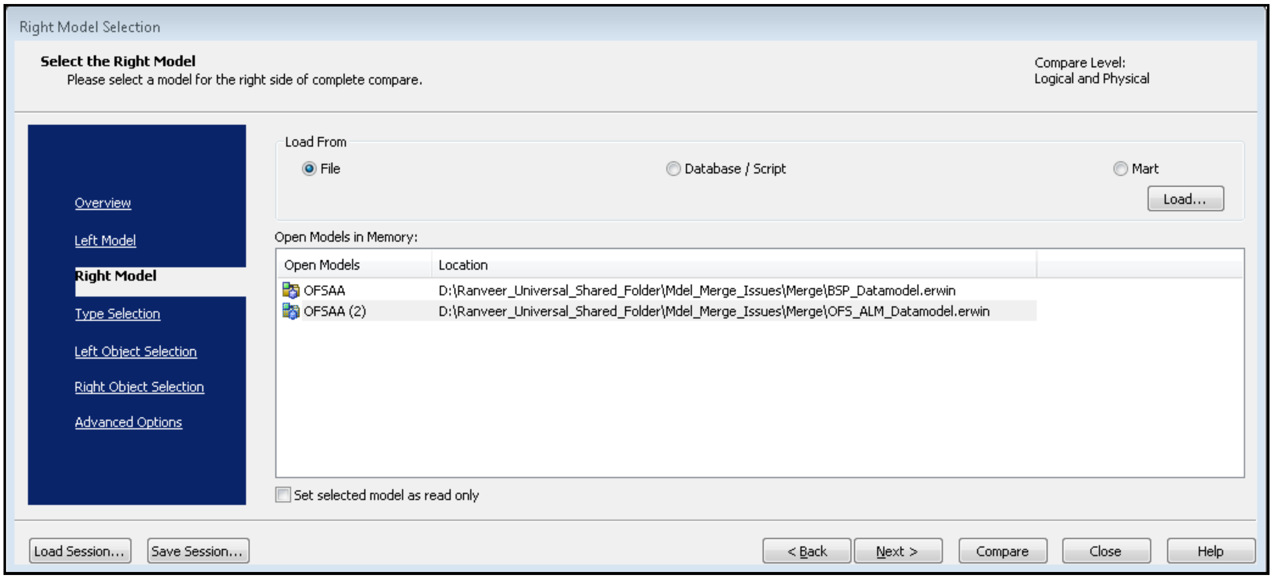
□ Set selected model as read only

7. **Right Model:** Using the **Right Model** input form, you can select the data model for the right side of the comparison in the Complete Compare dialog. The **Right Model** input form is identical to the **Left Model input form.**

   To select the required model for the right side of the comparison, click **Load**. The **File Open** dialog appears. Search for the required data model file, select it and click **Open**. The selected data model loads into the Complete Compare section. You can see the selected data model in the Open Models in a **Memory** list box. Click **Next** to go to the **Type Selection** input form.

NOTE When you launch the Complete Compare function with one data model, that data model loads into the **Right Model** form of the Complete Compare section. When you open the second data model for comparison, the existing data model in the **Right Model** form moves to the **Left Model** form and the second data model loads into the **Right Model** form.

Figure 12: Right Model input form



Select the Right Model
Please select a model for the right side of complete compare.

Compare Level:
Logical and Physical

Load From
● File ○ Database / Script ○ Mart

Open Models in Memory:

Open Models	Location
OFSAA	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin
OFSAA (2)	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin

□ Set selected model as read only

8. **Type Selection:** This form is critical in the entire Complete Compare process. Using this form, you can customize the layout for the merged output data model based on the existing layout of the input data models. This form restricts the comparison to specific object types and properties of the data models.

   a. In the **Type Selection** input form, expand **Model**. In the list box, select the required options using a checkbox.

NOTE

- If the **Type Resolution** dialog does not appear, go to the next form. Therefore, click **Next**.
- When you select the **Type Selection** form, it may prompt the display of the **Type Resolution** dialog to equalize the UDPs across the left and right models. This is a GDM-46 error.

   b. You can select the following type of multiple options in the **Type Selection** form:

      — Subject Areas

      — UDP

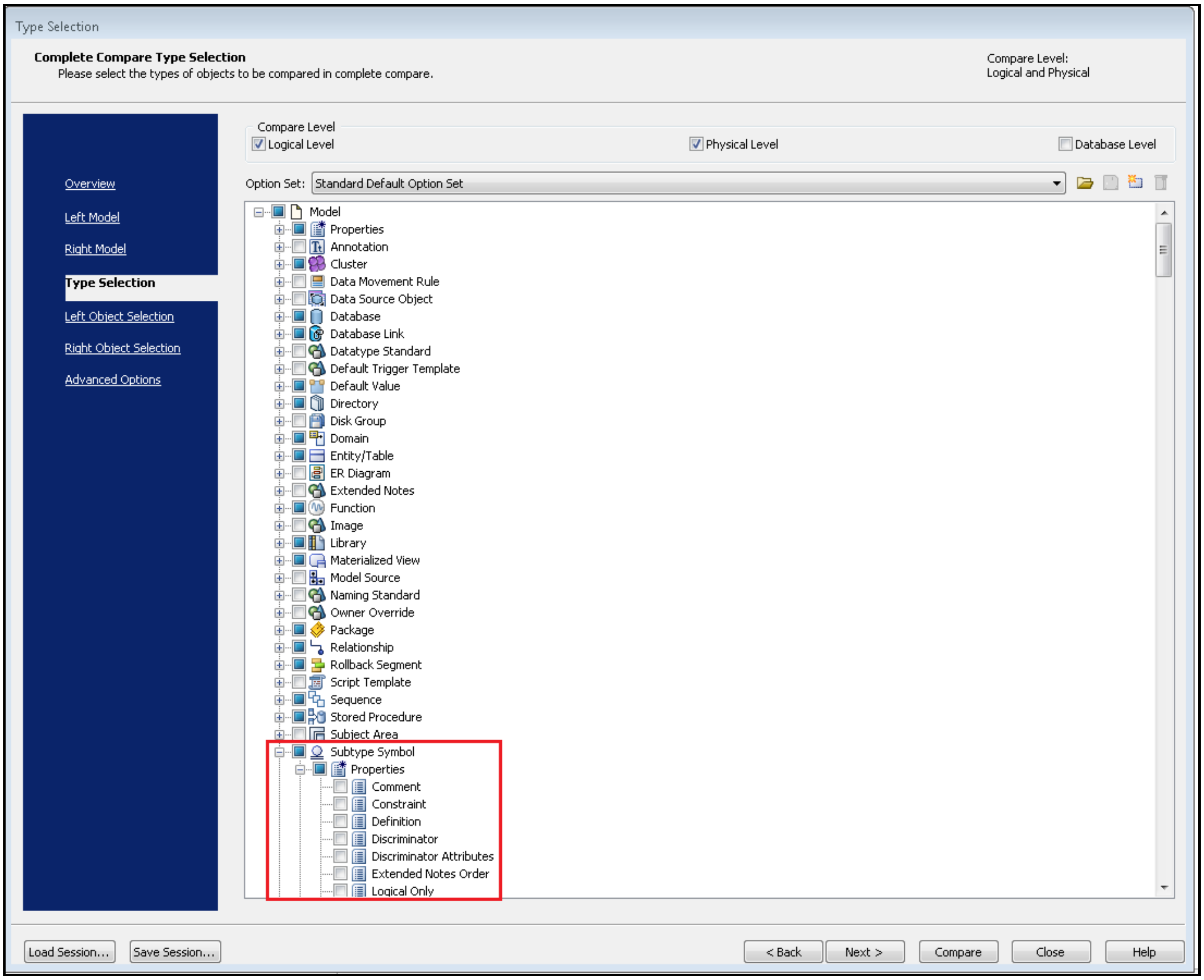
      — Physical Table and Physical Column

      — Logical Entity and Logical Attribute

      — Subtype and Supertype

   c. Below are the prompted values for Subtype (child table) set and Subtype items (child table elements) selection. To select all values from the Subtype Symbol, select the Subtype Symbol checkbox. Or to select individual Subtype items, expand Subtype Symbol, and select checkboxes against required Subtype items.

Figure 13: Type Selection form



Type Selection

Complete Compare Type Selection
Please select the types of objects to be compared in complete compare.

Compare Level:
Logical and Physical

Compare Level
☑ Logical Level ☑ Physical Level □ Database Level

Option Set: Standard Default Option Set

- ■ Model
  - ■ Properties
  - □ Annotation
  - ■ Cluster
  - □ Data Movement Rule
  - □ Data Source Object
  - ■ Database
  - ■ Database Link
  - □ Datatype Standard
  - □ Default Trigger Template
  - ■ Default Value
  - ■ Directory
  - □ Disk Group
  - ■ Domain
  - ■ Entity/Table
  - □ ER Diagram
  - □ Extended Notes
  - ■ Function
  - □ Image
  - ■ Library
  - ■ Materialized View
  - □ Model Source
  - □ Naming Standard
  - □ Owner Override
  - ■ Package
  - ■ Relationship
  - ■ Rollback Segment
  - □ Script Template
  - ■ Sequence
  - ■ Stored Procedure
  - □ Subject Area
  - ■ Subtype Symbol
    - ■ Properties
      - □ Comment
      - □ Constraint
      - □ Definition
      - □ Discriminator
      - □ Discriminator Attributes
      - □ Extended Notes Order
      - □ Logical Only

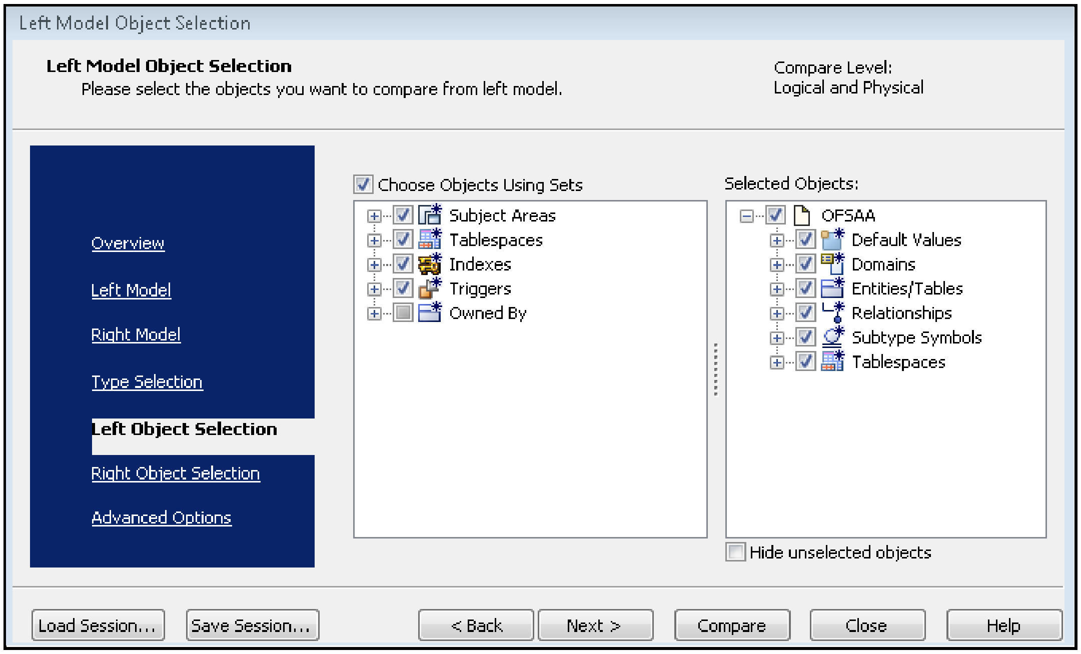
d. Then click **Next** to go to the **Left Object Selection** form.

9. **Left Object Selection:** Using the **Left Object Selection** input form, you can select which objects (table properties) can be included from the **Left Model** form in the comparison process.

   As shown in the image below, to choose the required **Object Sets** in the Choose Objects Using Sets list box, select the checkboxes against the Object Set names. The selected **Object Sets** appear in the Selected Objects list box on the right side.

NOTE You can select an individual item in each **Object** set. For example, expand **Subject Areas** object set, then select the individual checkboxes against required Subject Area item.

Figure 14: Left Object Selection input form



Left Model Object Selection

Left Model Object Selection
Please select the objects you want to compare from left model.

Compare Level:
Logical and Physical

☑ Choose Objects Using Sets

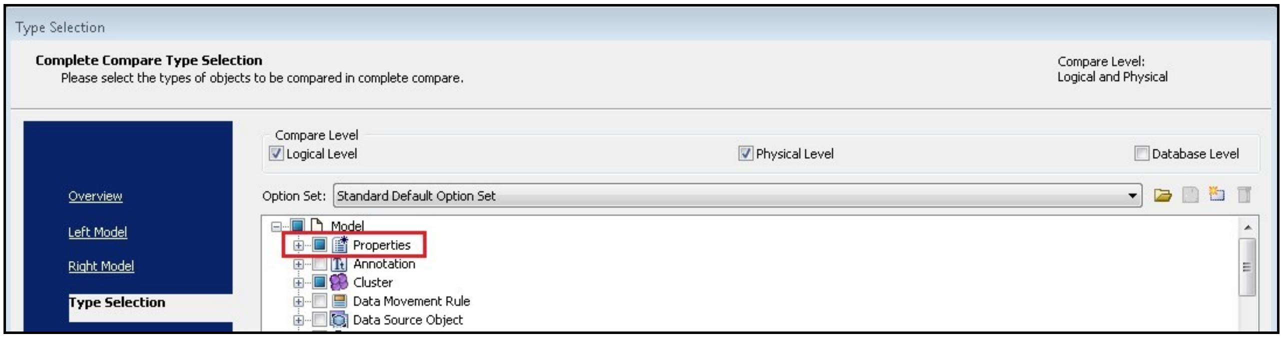
Selected Objects:

□ Hide unselected objects

Load Session... | Save Session... | < Back | Next > | Compare | Close | Help

To include tables in the comparison process, you must select table **Properties** in the **Type Selection** form. Then the tables will appear in the **Left Object Selection** input form.

Figure 15: Select table properties in the Type Selection form



Type Selection

Complete Compare Type Selection
Please select the types of objects to be compared in complete compare.

Compare Level:
Logical and Physical

Compare Level
☑ Logical Level ☑ Physical Level □ Database Level

Option Set: Standard Default Option Set

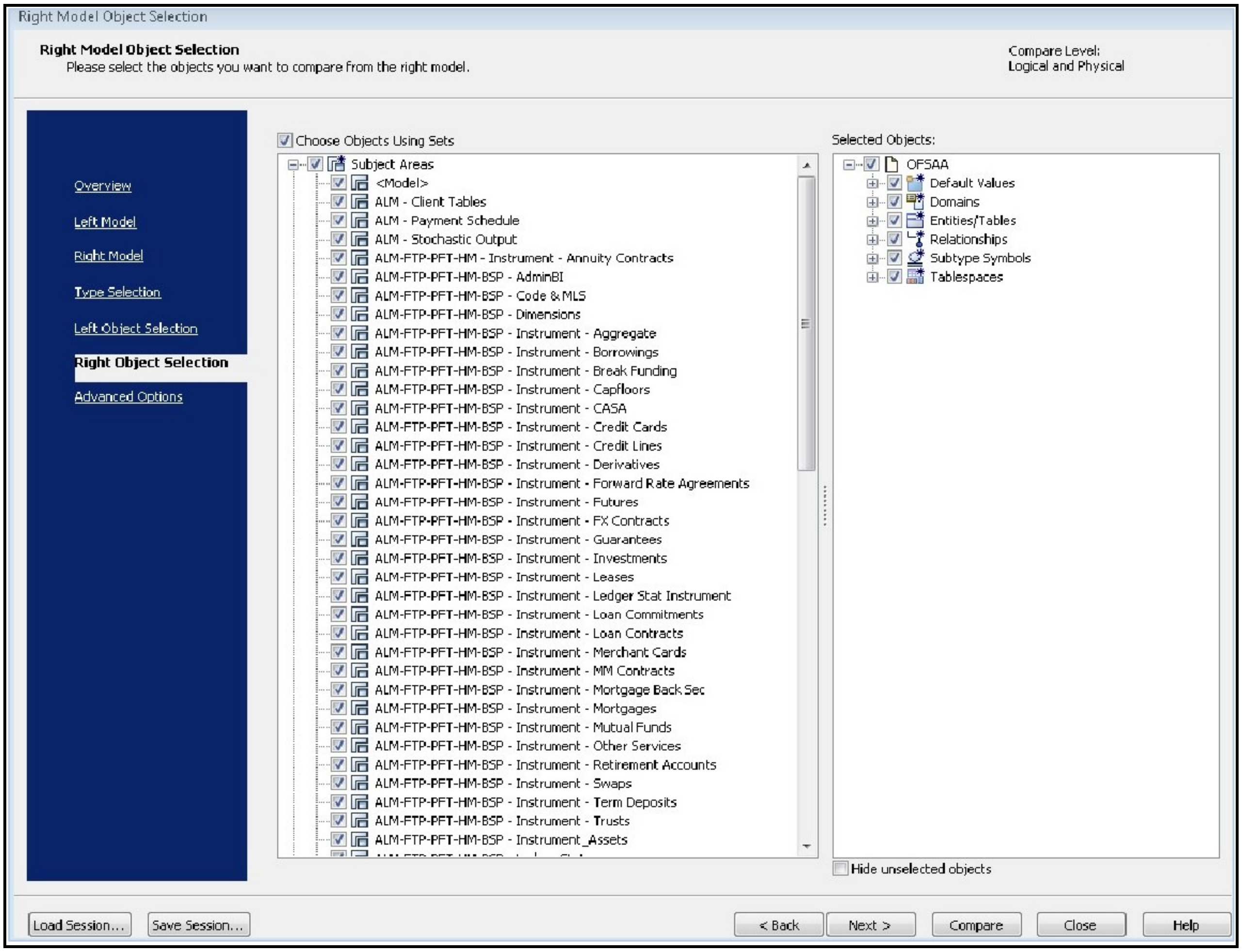
10. **Right Object Selection:** Using the **Right Object Selection** input form, you can select which objects (table properties) can be included from the **Right Model** form in the comparison process.

    To choose the required **Object Sets** from the Choose Objects Using Sets list box, select the checkboxes against the Object Set names (you can select Object Sets like Subject Areas, Tablespaces, Indexes, Triggers). The selected **Object Sets** appear in the Selected Objects list box on the right side. Click **Next** to go to the **Advanced Options** form.

NOTE

- You can select an individual item in each Object set. For example, if you are performing Subject Areas-wise Complete Compare, expand Subject Areas object set, then select the individual checkboxes against required Subject Area item. In the image below, based on apps requirement, we have selected all the Subject Areas.
- If there are very less number of tables to merge with a very huge model, then you can create a Dummy Subject Area in a small model and move the tables into Dummy Subject Area. To Complete Compare under the Left Object Selection form or the Right Object Selection form, select the Dummy Subject Area and perform Complete Compare.

Figure 16: Right Object Selection in the erwin Data Modeler for OFSDF



Right Model Object Selection
Please select the objects you want to compare from the right model.

Compare Level:
Logical and Physical

☑ Choose Objects Using Sets

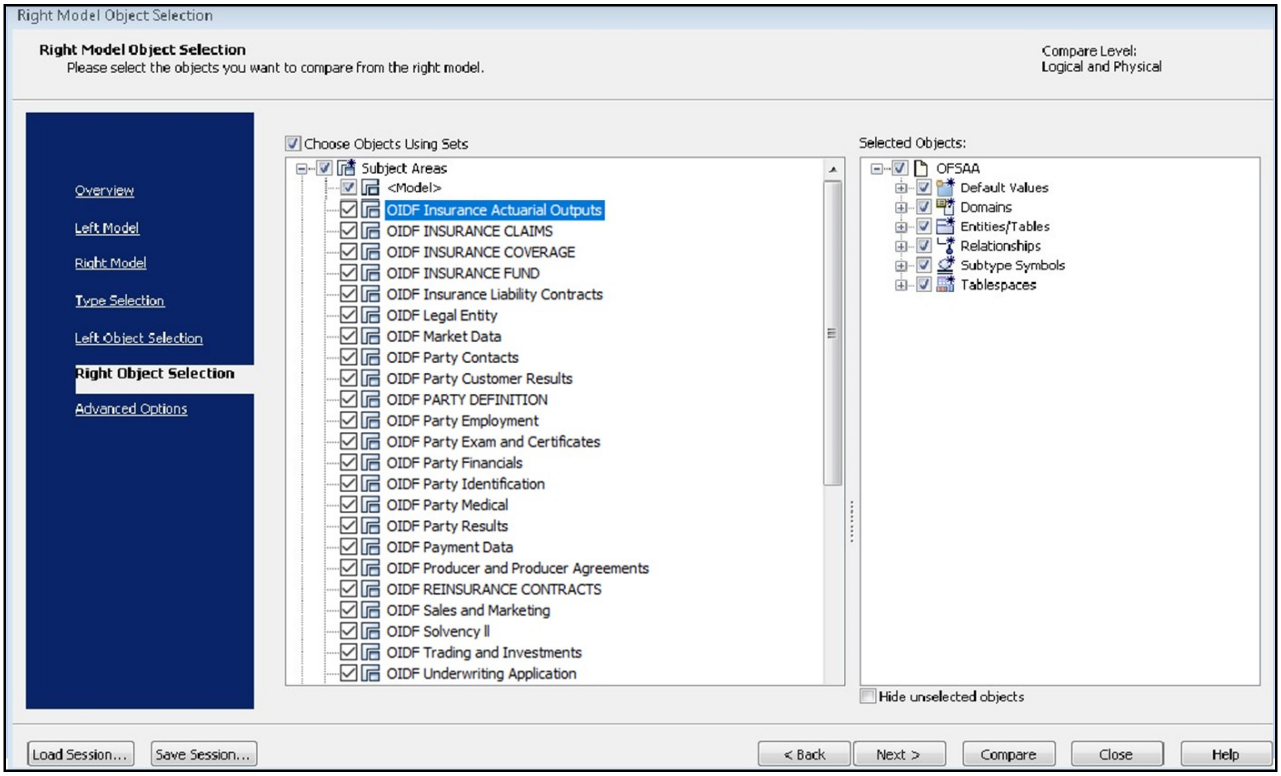
- ☑ Subject Areas
  - ☑ <Model>
  - ☑ ALM - Client Tables
  - ☑ ALM - Payment Schedule
  - ☑ ALM - Stochastic Output
  - ☑ ALM-FTP-PFT-HM - Instrument - Annuity Contracts
  - ☑ ALM-FTP-PFT-HM-BSP - AdminBI
  - ☑ ALM-FTP-PFT-HM-BSP - Code & MLS
  - ☑ ALM-FTP-PFT-HM-BSP - Dimensions
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Aggregate
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Borrowings
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Break Funding
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Capfloors
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - CASA
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Credit Cards
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Credit Lines
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Derivatives
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Forward Rate Agreements
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Futures
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - FX Contracts
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Guarantees
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Investments
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Leases
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Ledger Stat Instrument
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Loan Commitments
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Loan Contracts
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Merchant Cards
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - MM Contracts
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Mortgage Back Sec
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Mortgages
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Mutual Funds
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Other Services
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Retirement Accounts
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Swaps
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Term Deposits
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument - Trusts
  - ☑ ALM-FTP-PFT-HM-BSP - Instrument_Assets

Selected Objects:

- ☑ OFSAA
  - ☑ Default Values
  - ☑ Domains
  - ☑ Entities/Tables
  - ☑ Relationships
  - ☑ Subtype Symbols
  - ☑ Tablespaces

☐ Hide unselected objects

Figure 17: Right Object Selection in the erwin data modeler for OIDF



Right Model Object Selection

Right Model Object Selection
Please select the objects you want to compare from the right model.

Compare Level:
Logical and Physical

☑ Choose Objects Using Sets

- ☑ Subject Areas
  - ☑ <Model>
  - ☑ OIDF Insurance Actuarial Outputs
  - ☑ OIDF INSURANCE CLAIMS
  - ☑ OIDF INSURANCE COVERAGE
  - ☑ OIDF INSURANCE FUND
  - ☑ OIDF Insurance Liability Contracts
  - ☑ OIDF Legal Entity
  - ☑ OIDF Market Data
  - ☑ OIDF Party Contacts
  - ☑ OIDF Party Customer Results
  - ☑ OIDF PARTY DEFINITION
  - ☑ OIDF Party Employment
  - ☑ OIDF Party Exam and Certificates
  - ☑ OIDF Party Financials
  - ☑ OIDF Party Identification
  - ☑ OIDF Party Medical
  - ☑ OIDF Party Results
  - ☑ OIDF Payment Data
  - ☑ OIDF Producer and Producer Agreements
  - ☑ OIDF REINSURANCE CONTRACTS
  - ☑ OIDF Sales and Marketing
  - ☑ OIDF Solvency II
  - ☑ OIDF Trading and Investments
  - ☑ OIDF Underwriting Application

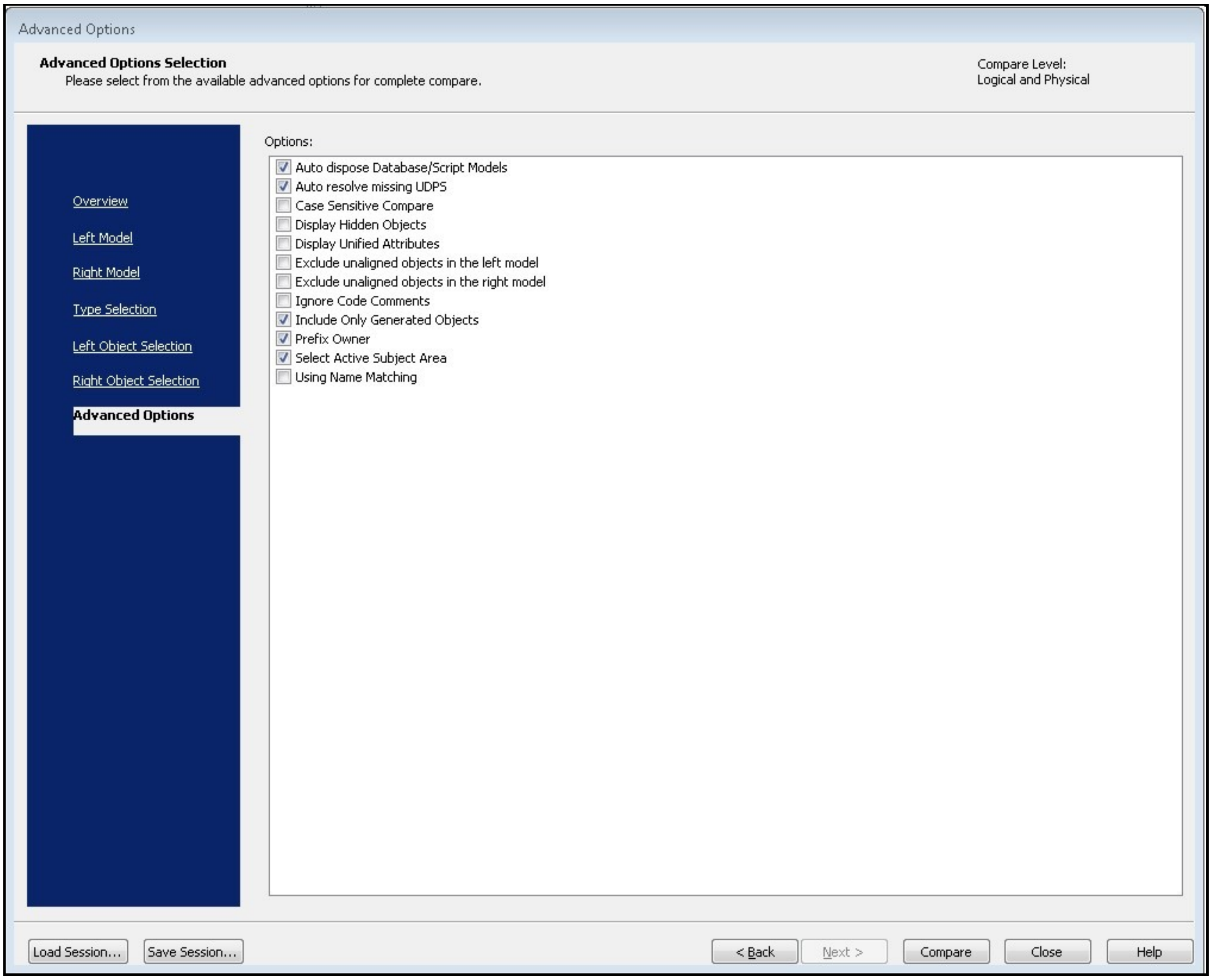
Selected Objects:

- ☑ OFSAA
  - ☑ Default Values
  - ☑ Domains
  - ☑ Entities/Tables
  - ☑ Relationships
  - ☑ Subtype Symbols
  - ☑ Tablespaces

□ Hide unselected objects

11. **Advanced Options:** This section consists of the functional options. In the image below, the selected options are mandatory. As per your requirement, select further options.

Figure 18: Advanced Options section



Advanced Options Selection
Please select from the available advanced options for complete compare.

Compare Level:
Logical and Physical

Options:

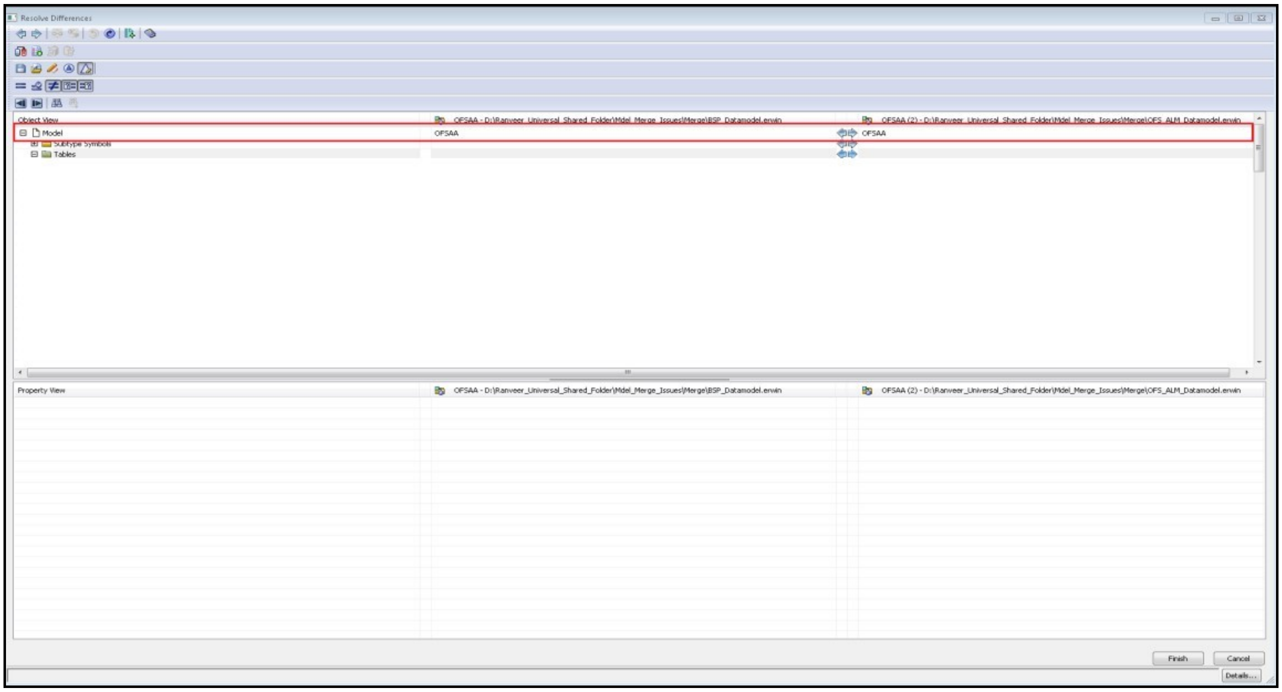
- ☑ Auto dispose Database/Script Models
- ☑ Auto resolve missing UDPS
- □ Case Sensitive Compare
- □ Display Hidden Objects
- □ Display Unified Attributes
- □ Exclude unaligned objects in the left model
- □ Exclude unaligned objects in the right model
- □ Ignore Code Comments
- ☑ Include Only Generated Objects
- ☑ Prefix Owner
- ☑ Select Active Subject Area
- □ Using Name Matching

12. To initiate the comparison process, click **Next** and then click **Compare**. The **Resolve Differences** window opens.

13. In the **Resolve Differences** window, there are three methods to merge data models:

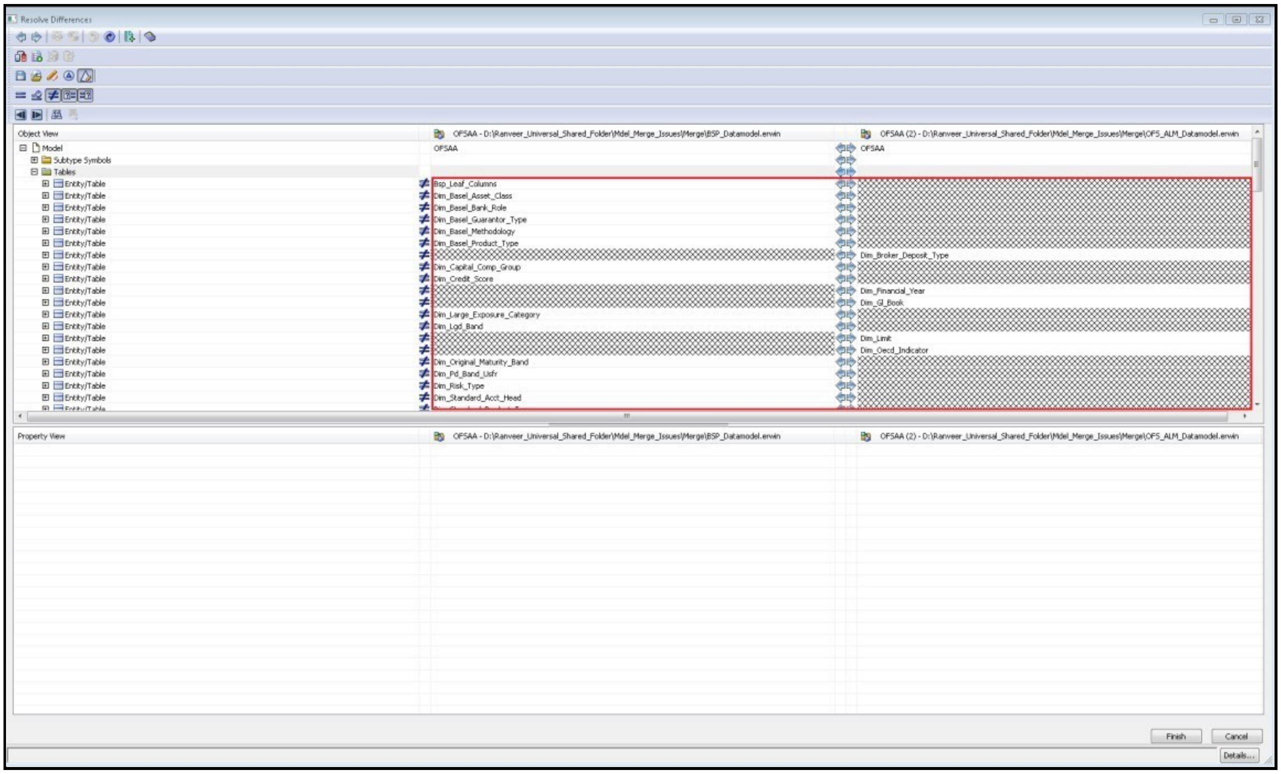
    - In the first method, you can merge two data models at the data model level. To merge a data model (Left Model) with another data model (Right Model), expand Model, then click the Right Arrow⇨. Alternatively, to merge the Right Model with Left Model, click the Left Arrow⇦.

Figure 19: Merge two data models at the data model level


Resolve Differences
Object View
OFSAA - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin
OFSAA (2) - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin
Model
OFSAA
OFSAA
Subtype Symbols
Tables
Property View
OFSAA - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin
OFSAA (2) - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin
Finish
Cancel
Details...

- In the second method, you can merge two data models at the table level. To merge two data models at the table level, expand Model, expand Tables, select Not Equal To ≠ from the menu bar. Then to merge the Right Model with the Left Model at table level, click the required table and click the Left Arrow ⇦, and repeat the same for each table. Or, to merge the Left Model with the Right Model at the table level, click the required table and click the Right Arrow ⇨, and repeat the same for each table.

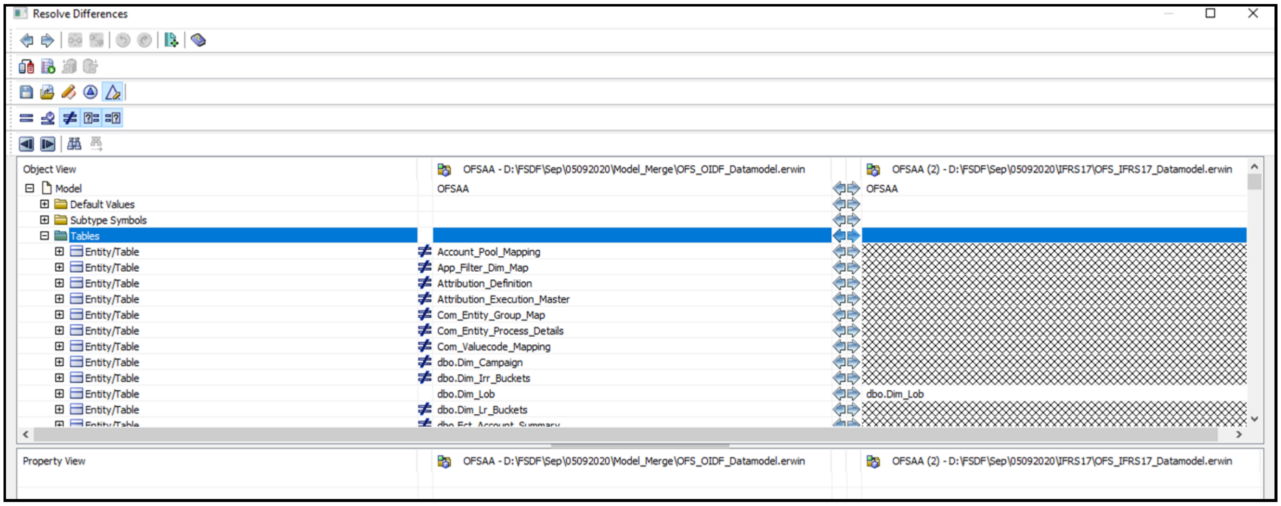
Figure 20: Merge the models at the table level in OFSDF



Object View	OFSAA - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin	OFSAA (2) - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin
Model	OFSAA	OFSAA
Subtype Symbols		
Tables		
Entity/Table	Bsp_Leaf_Columns	▨
Entity/Table	Dim_Basel_Asset_Class	▨
Entity/Table	Dim_Basel_Bank_Role	▨
Entity/Table	Dim_Basel_Guarantor_Type	▨
Entity/Table	Dim_Basel_Methodology	▨
Entity/Table	Dim_Basel_Product_Type	▨
Entity/Table	▨	Dim_Broker_Deposit_Type
Entity/Table	Dim_Capital_Comp_Group	▨
Entity/Table	Dim_Credit_Score	▨
Entity/Table	▨	Dim_Financial_Year
Entity/Table	▨	Dim_Gl_Book
Entity/Table	Dim_Large_Exposure_Category	▨
Entity/Table	Dim_Lgd_Band	▨
Entity/Table	▨	Dim_Limit
Entity/Table	▨	Dim_Oecd_Indicator
Entity/Table	Dim_Original_Maturity_Band	▨
Entity/Table	Dim_Pd_Band_Usfr	▨
Entity/Table	Dim_Risk_Type	▨
Entity/Table	Dim_Standard_Acct_Head	▨

Property View | OFSAA - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin | OFSAA (2) - D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin

Figure 21: Merge the models at the table level in OIDF



Object View	OFSAA - D:\FSDF\Sep\05092020\Model_Merge\OFS_OIDF_Datamodel.erwin	OFSAA (2) - D:\FSDF\Sep\05092020\IFRS17\OFS_IFRS17_Datamodel.erwin
Model	OFSAA	OFSAA
Default Values		
Subtype Symbols		
Tables		
Entity/Table	Account_Pool_Mapping	▨
Entity/Table	App_Filter_Dim_Map	▨
Entity/Table	Attribution_Definition	▨
Entity/Table	Attribution_Execution_Master	▨
Entity/Table	Com_Entity_Group_Map	▨
Entity/Table	Com_Entity_Process_Details	▨
Entity/Table	Com_Valuecode_Mapping	▨
Entity/Table	dbo.Dim_Campaign	▨
Entity/Table	dbo.Dim_Irr_Buckets	▨
Entity/Table	dbo.Dim_Lob	dbo.Dim_Lob
Entity/Table	dbo.Dim_Lr_Buckets	▨

Property View | OFSAA - D:\FSDF\Sep\05092020\Model_Merge\OFS_OIDF_Datamodel.erwin | OFSAA (2) - D:\FSDF\Sep\05092020\IFRS17\OFS_IFRS17_Datamodel.erwin

- In the third method, you can merge two data models at the Subtype and Supertype table level. To merge two data models at Subtype and Supertype level:

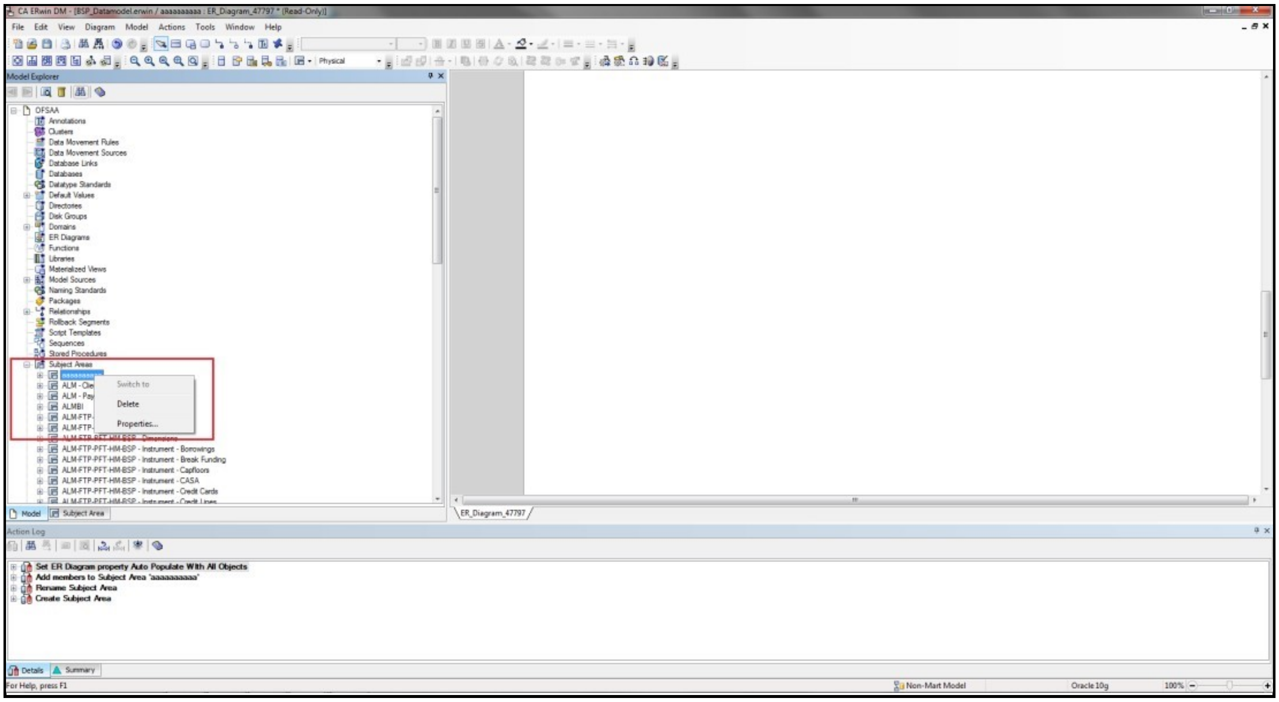
  During an ongoing Supertype Subtype model complete comparison process, you cannot move the Supertype table directly from the Left Model or the Right Model. You must find all the relationships between the Supertype table (parent table) and Subtype tables (child tables). Then you can move the Subtype table and after that, move the Supertype table.

  How to find the Supertype or Subtype table relationship?

i. Open the model where you are unable to move the table from left to right or from right to left. For illustration, BSP and ALM tables in OFSDF data model and Claims table in the OIDF data model along with Supertype and Subtype are considered.

ii. In **Subject Areas**, create a **Dummy Subject Area** with any name. Right-click **Dummy Subject Area** name, click **Properties**. The **Dummy Subject Area '[name]' Editor** window opens.

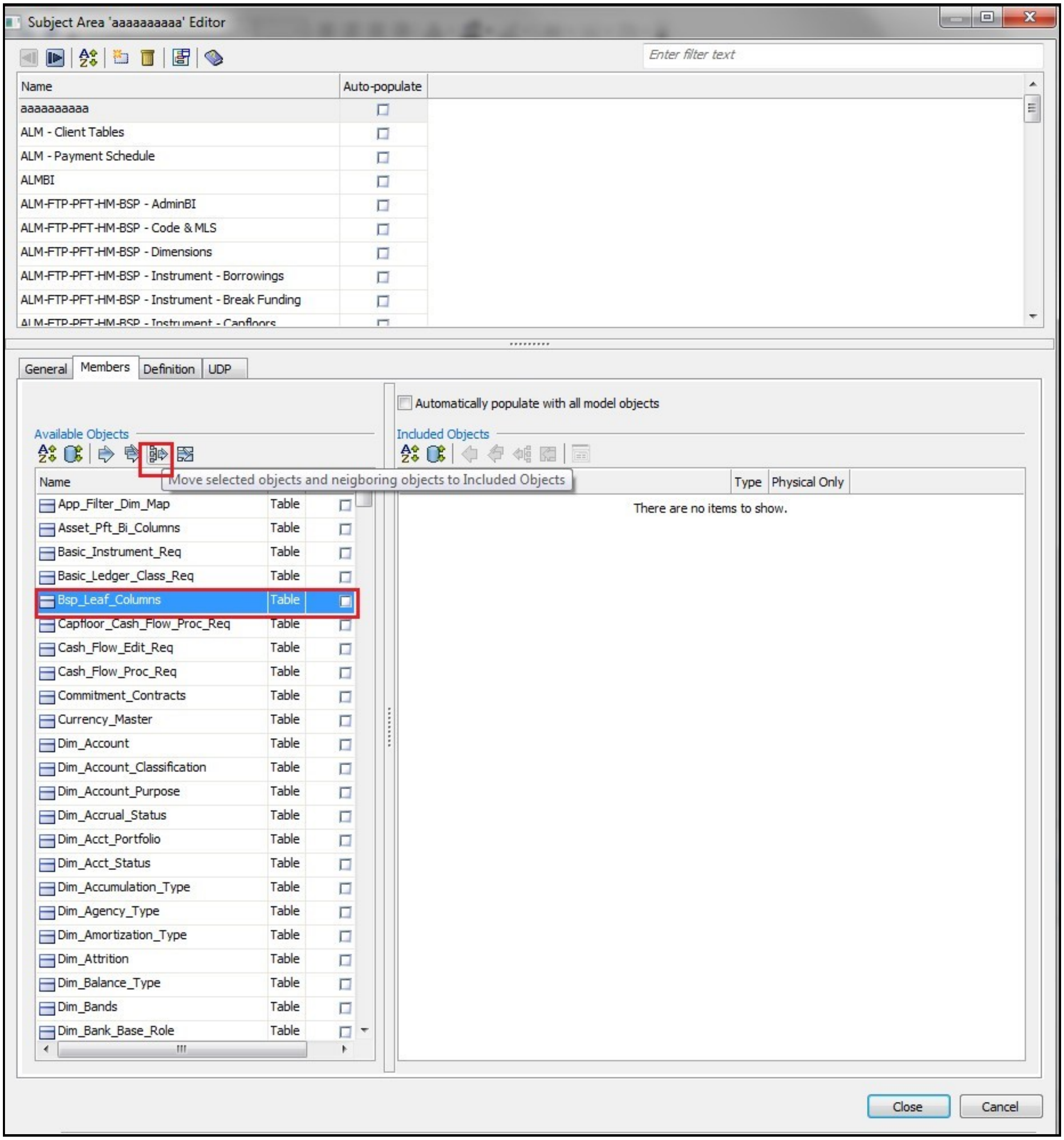
Figure 22: Create a Dummy Subject Area



Set ER Diagram property Auto Populate With All Objects
Add members to Subject Area 'aaaaaaaaaa'
Rename Subject Area
Create Subject Area

iii. In the **Members** tab, select the required Table Name and click Grouped Right Arrow . This action moves the selected object and the objects mapped to it from the **Available Objects** list box to the **Included Objects** list box. In this illustration, in the **Included Objects** list box, BSP_LEAF_COLUMNS is the Supertype (parent table) in OFSDF and DIM_CLAIM in OIDF and the remaining are Subtype (child tables).

Figure 23: Move the objects in OFSDF to merge the two data models at Subtype and Supertype level



Subject Area 'aaaaaaaaaa' Editor

Enter filter text

Name	Auto-populate
aaaaaaaaaa	□
ALM - Client Tables	□
ALM - Payment Schedule	□
ALMBI	□
ALM-FTP-PFT-HM-BSP - AdminBI	□
ALM-FTP-PFT-HM-BSP - Code & MLS	□
ALM-FTP-PFT-HM-BSP - Dimensions	□
ALM-FTP-PFT-HM-BSP - Instrument - Borrowings	□
ALM-FTP-PFT-HM-BSP - Instrument - Break Funding	□
ALM-FTP-PFT-HM-BSP - Instrument - Capfloors	□

□ Automatically populate with all model objects

Available Objects

Move selected objects and neigboring objects to Included Objects

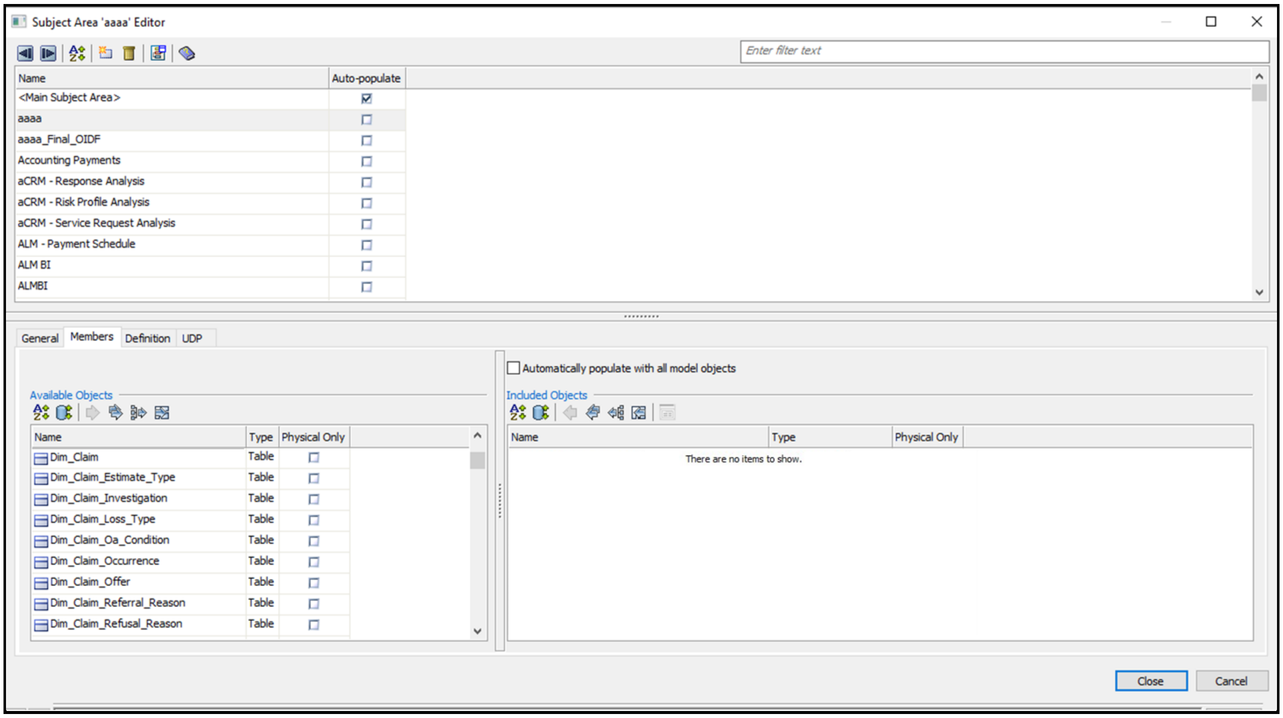
Name		
App_Filter_Dim_Map	Table	□
Asset_Pft_Bi_Columns	Table	□
Basic_Instrument_Req	Table	□
Basic_Ledger_Class_Req	Table	□
Bsp_Leaf_Columns	Table	□
Capfloor_Cash_Flow_Proc_Req	Table	□
Cash_Flow_Edit_Req	Table	□
Cash_Flow_Proc_Req	Table	□
Commitment_Contracts	Table	□
Currency_Master	Table	□
Dim_Account	Table	□
Dim_Account_Classification	Table	□
Dim_Account_Purpose	Table	□
Dim_Accrual_Status	Table	□
Dim_Acct_Portfolio	Table	□
Dim_Acct_Status	Table	□
Dim_Accumulation_Type	Table	□
Dim_Agency_Type	Table	□
Dim_Amortization_Type	Table	□
Dim_Attrition	Table	□
Dim_Balance_Type	Table	□
Dim_Bands	Table	□
Dim_Bank_Base_Role	Table	□

Included Objects

	Type	Physical Only

There are no items to show.

Figure 24: Move the objects in OIDF to merge the two data models at Subtype and Supertype level



Subject Area 'aaaa' Editor

Enter filter text

Name	Auto-populate
<Main Subject Area>	☑
aaaa	□
aaaa_Final_OIDF	□
Accounting Payments	□
aCRM - Response Analysis	□
aCRM - Risk Profile Analysis	□
aCRM - Service Request Analysis	□
ALM - Payment Schedule	□
ALM BI	□
ALMBI	□

General | Members | Definition | UDP

Available Objects

Name	Type	Physical Only
Dim_Claim	Table	□
Dim_Claim_Estimate_Type	Table	□
Dim_Claim_Investigation	Table	□
Dim_Claim_Loss_Type	Table	□
Dim_Claim_Oa_Condition	Table	□
Dim_Claim_Occurrence	Table	□
Dim_Claim_Offer	Table	□
Dim_Claim_Referral_Reason	Table	□
Dim_Claim_Refusal_Reason	Table	□

□ Automatically populate with all model objects

Included Objects

Name	Type	Physical Only

There are no items to show.

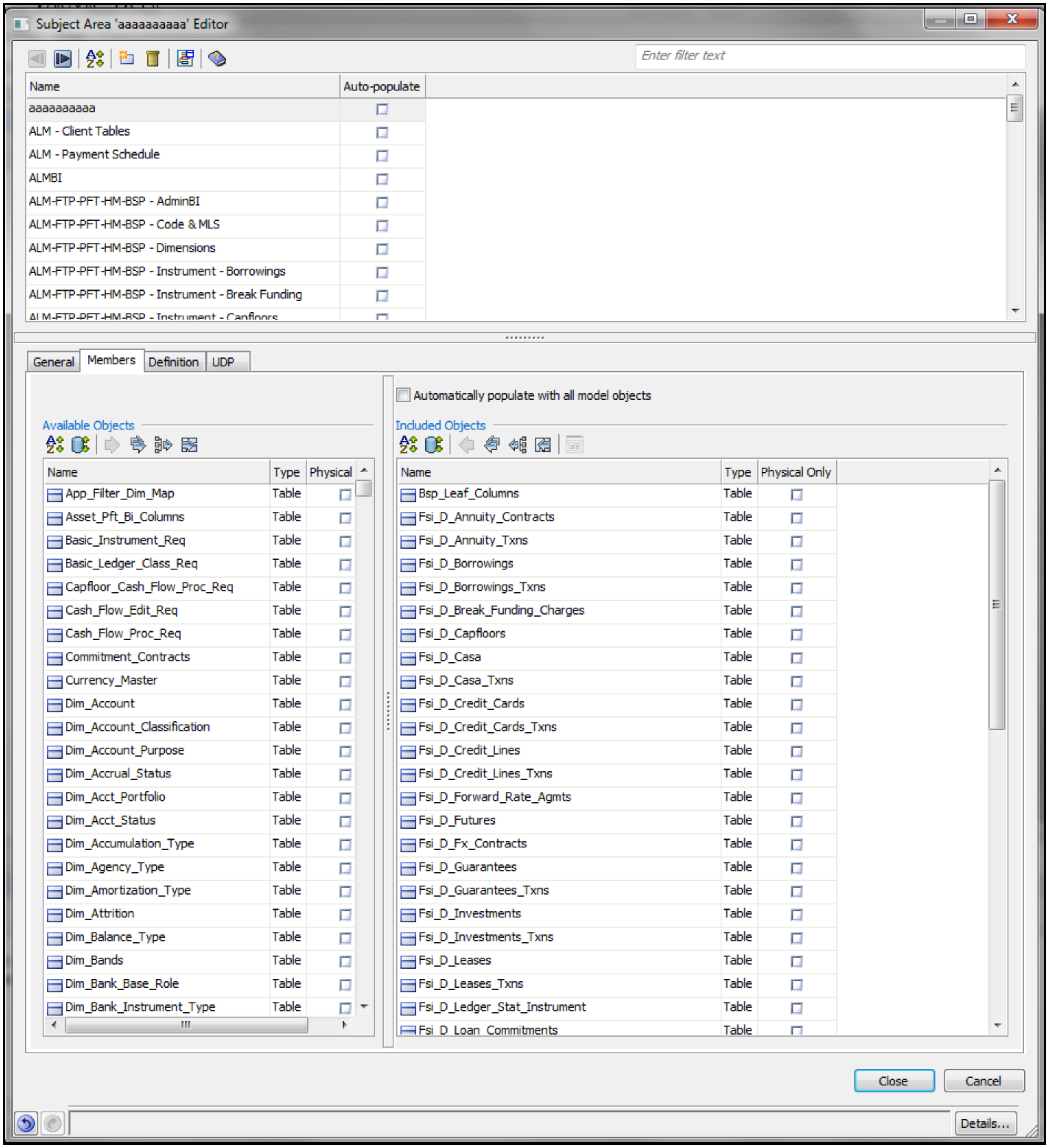
Close | Cancel

iv. The **Spanning Neighborhood** window opens. In the **Ancestors** and **Descendants** sections, in Level, select value 1 to search for a one-to-one parent-child table relationship. The list of the Subtype tables mapped to the Supertype table appears. In this example, the list of Subtype tables mapped to the Supertype tables BSP_LEAF_COLUMNS in OFSDF and DIM_CLAIM in OIDF appears.

v. To include the required Subtype tables (for this Dummy Subject Area) in the comparison process, select them and click **Add**. Close the **Spanning Neighborhood** window.

vi. In the **Subject Area '[name]' Editor** window, in the Included Objects list box, you can see the Supertype table (in this illustration, BSP_LEAF_COLUMNS in OFSDF and DIM_CLAIM in OIDF) and its Subtype tables.

Figure 25: Subject Area Editor window shows the Supertype table and its Subtype tables in OFSDF



Name	Auto-populate
aaaaaaaaaa	☐
ALM - Client Tables	☐
ALM - Payment Schedule	☐
ALMBI	☐
ALM-FTP-PFT-HM-BSP - AdminBI	☐
ALM-FTP-PFT-HM-BSP - Code & MLS	☐
ALM-FTP-PFT-HM-BSP - Dimensions	☐
ALM-FTP-PFT-HM-BSP - Instrument - Borrowings	☐
ALM-FTP-PFT-HM-BSP - Instrument - Break Funding	☐
ALM-FTP-PFT-HM-BSP - Instrument - Capfloors	☐

☐ Automatically populate with all model objects

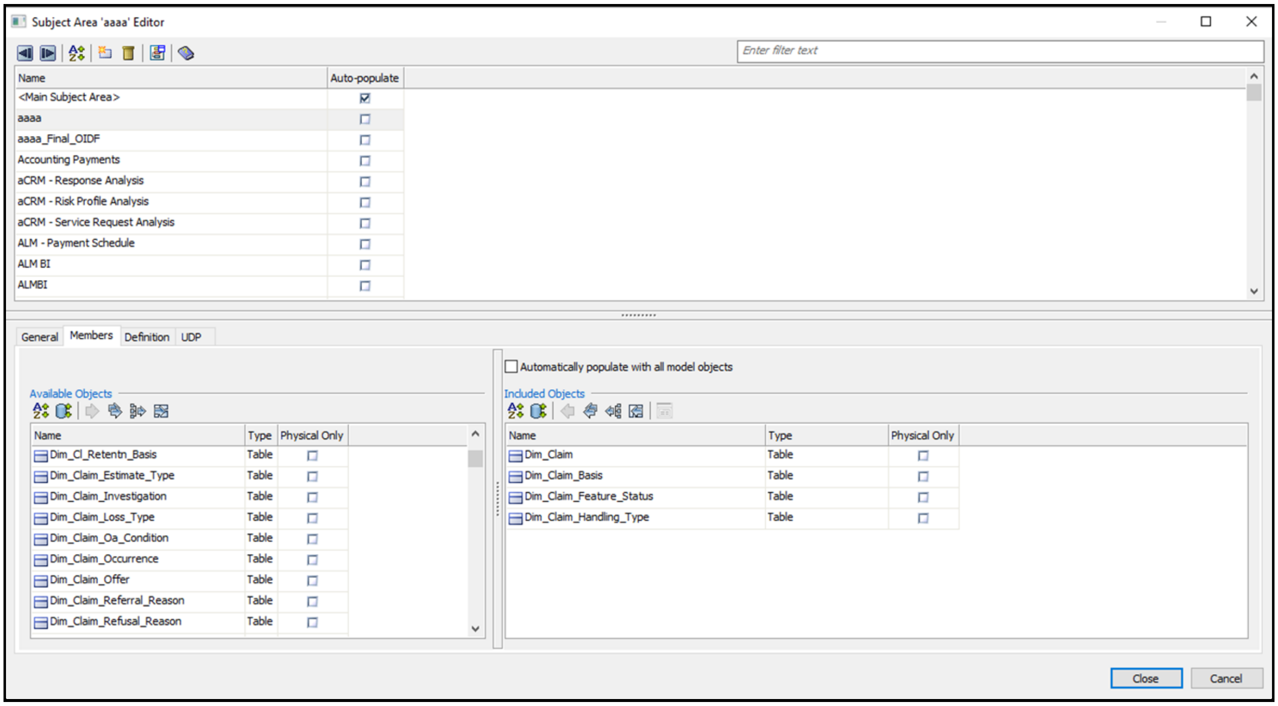
Available Objects

Name	Type	Physical
App_Filter_Dim_Map	Table	☐
Asset_Pft_Bi_Columns	Table	☐
Basic_Instrument_Req	Table	☐
Basic_Ledger_Class_Req	Table	☐
Capfloor_Cash_Flow_Proc_Req	Table	☐
Cash_Flow_Edit_Req	Table	☐
Cash_Flow_Proc_Req	Table	☐
Commitment_Contracts	Table	☐
Currency_Master	Table	☐
Dim_Account	Table	☐
Dim_Account_Classification	Table	☐
Dim_Account_Purpose	Table	☐
Dim_Accrual_Status	Table	☐
Dim_Acct_Portfolio	Table	☐
Dim_Acct_Status	Table	☐
Dim_Accumulation_Type	Table	☐
Dim_Agency_Type	Table	☐
Dim_Amortization_Type	Table	☐
Dim_Attrition	Table	☐
Dim_Balance_Type	Table	☐
Dim_Bands	Table	☐
Dim_Bank_Base_Role	Table	☐
Dim_Bank_Instrument_Type	Table	☐

Included Objects

Name	Type	Physical Only
Bsp_Leaf_Columns	Table	☐
Fsi_D_Annuity_Contracts	Table	☐
Fsi_D_Annuity_Txns	Table	☐
Fsi_D_Borrowings	Table	☐
Fsi_D_Borrowings_Txns	Table	☐
Fsi_D_Break_Funding_Charges	Table	☐
Fsi_D_Capfloors	Table	☐
Fsi_D_Casa	Table	☐
Fsi_D_Casa_Txns	Table	☐
Fsi_D_Credit_Cards	Table	☐
Fsi_D_Credit_Cards_Txns	Table	☐
Fsi_D_Credit_Lines	Table	☐
Fsi_D_Credit_Lines_Txns	Table	☐
Fsi_D_Forward_Rate_Agmts	Table	☐
Fsi_D_Futures	Table	☐
Fsi_D_Fx_Contracts	Table	☐
Fsi_D_Guarantees	Table	☐
Fsi_D_Guarantees_Txns	Table	☐
Fsi_D_Investments	Table	☐
Fsi_D_Investments_Txns	Table	☐
Fsi_D_Leases	Table	☐
Fsi_D_Leases_Txns	Table	☐
Fsi_D_Ledger_Stat_Instrument	Table	☐
Fsi_D_Loan_Commitments	Table	☐

Figure 26: Subject Area Editor window shows the Supertype table and its Subtype tables in OIDF



Subject Area 'aaaa' Editor

Enter filter text

Name	Auto-populate
<Main Subject Area>	☑
aaaa	☐
aaaa_Final_OIDF	☐
Accounting Payments	☐
aCRM - Response Analysis	☐
aCRM - Risk Profile Analysis	☐
aCRM - Service Request Analysis	☐
ALM - Payment Schedule	☐
ALM BI	☐
ALMBI	☐

☐ Automatically populate with all model objects

Available Objects

Name	Type	Physical Only
Dim_Cl_Retentn_Basis	Table	☐
Dim_Claim_Estimate_Type	Table	☐
Dim_Claim_Investigation	Table	☐
Dim_Claim_Loss_Type	Table	☐
Dim_Claim_Oa_Condition	Table	☐
Dim_Claim_Occurrence	Table	☐
Dim_Claim_Offer	Table	☐
Dim_Claim_Referral_Reason	Table	☐
Dim_Claim_Refusal_Reason	Table	☐

Included Objects

Name	Type	Physical Only
Dim_Claim	Table	☐
Dim_Claim_Basis	Table	☐
Dim_Claim_Feature_Status	Table	☐
Dim_Claim_Handling_Type	Table	☐

vii. Close the **Subject Area '[name]' Editor** window. In the data model window, click **Save** to save the changes made in the model.

viii. Repeat all the steps for the data model merge until the **Left Object Selection** form (that is, repeat steps from 1 to 8).

ix. In the **Left Objection Selection** form, in the **Choose Objects Using Sets** list box, select all the Subject Areas along with the newly created Dummy Subject Area. Into this Dummy Subject Area, you moved the Subtype and Supertype tables in previous steps. Click **Next**.

x. Repeat steps 10 and 11.

xi. To initiate the comparison process, click **Next** and then click **Compare**. The **Resolve Differences** window opens.

xii. In the **Resolve Differences** window, to merge two data models at table level, expand **Model**, expand **Tables**, select Not Equal To≠ from the menu bar. Select the Supertype table (BSP_LEAF_COLUMN for OFSDF and DIM_CLAIM for OIDF in this illustration), and move it to left (to merge Right Model with Left Model, click the Left Arrow⇦) or right (to merge Left Model with Right Model, click the Right Arrow⇨) side of the comparison as required.

In the considered illustration, BSP_LEAF_COLUMN for OFSDF and DIM_CLAIM for OIDF are the parent tables (Supertype) from the source table and its respective child tables (Subtype) are named as FSI_ACCID_HEALTH_INS_CO_CD, FSI_ACCID_HEALTH_INS_CO_MLS, DIM_ACCOUNT_CLASSIFICATION and so forth in OFSDF, and DIM_CLAIM_BASIS, DIM_CLAIM_FEATURE_STATUS and DIM_CLAIM_HANDLING_TYPE in OIDF. First, the child table names (Subtype table names) are moved from the source model to the target model, and then the Supertype table names BSP_LEAF_COLUMN in OFSDF and DIM_CLAIM in OIDF are moved to the target model.

14. Click **Finish** to complete the data model merging, and then click **Save** to save the output-merged model with merged data model as a part of the output model name.

13 Build Guidelines

Changes identified and approved needs to be included in the model per the naming standards.

- Table addition/modification
  - Data model standards to be followed for each change.
  - Make sure Entity name and description are provided.
  - Make sure values are set for relevant application usage UDP's like ALM, FTP, PFT, Basel II, CREC, and so on.
- Column addition/modification
  - Data model standards to be followed for each change.
  - Make sure the Attribute name, a description is provided.
  - Make sure values are set for relevant application usage UDP's like ALM, FTP, PFT, Basel II, CREC, etc.
  - Column level UDPs named 'Custom' (NO/YES), 'Customization Reason' (Pending ER, Specific to Customer, Modified data element) and 'Type of Change' (Length, Datatype, Logical Name, Description, Addition) have to be specified for attributes added/modified as part of the extension process.

> **NOTE** Make sure that the Model Properties, Model Level UDP – VERSION is not changed while extending the model as per the requirement.

14 Appendix

For more information related to the usage of data models, refer to the following documents:

- Oracle Financial Services Analytical Applications (OFSAA) Data Model Naming Standards Guide Release 8.1.x
- Data Model Customization Review Template

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised/recently released documents.

ORACLE