



## **erwin Data Modeler**

### **Feature Tour**

**Release 12.5**



## Legal Notices

This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the “Documentation”), is for your informational purposes only and is subject to change or withdrawal by Quest Software, Inc and/or its affiliates at any time. This Documentation is proprietary information of Quest Software, Inc and/or its affiliates and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Quest Software, Inc and/or its affiliates

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Quest Software, Inc and/or its affiliates copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Quest Software, Inc and/or its affiliates that all copies and partial copies of the Documentation have been returned to Quest Software, Inc and/or its affiliates or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, QUEST SOFTWARE, INC. PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL QUEST SOFTWARE, INC. BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF QUEST SOFTWARE, INC. IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Quest Software, Inc and/or its affiliates.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2025 Quest Software, Inc and/or its affiliates All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## Contact erwin

### Understanding your Support

Review [support maintenance programs and offerings](#).

### Registering for Support

Access the [erwin support](#) site and register for product support.

### Accessing Technical Support

For your convenience, erwin provides easy access to "One Stop" support for all editions of [erwin Data Modeler](#), and includes the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- erwin Support policies and guidelines
- Other helpful resources appropriate for your product

For information about other erwin products, visit <http://erwin.com/products>.

### Provide Feedback

If you have comments or questions, or feedback about erwin product documentation, you can send a message to [techpubs@erwin.com](mailto:techpubs@erwin.com).

### News and Events

Visit [News and Events](#) to get up-to-date news, announcements, and events. View video demos and read up on customer success stories and articles by industry experts.

# Contents

---

|  |    |
|--|----|
| Introduction .....   | 8  |
| MongoDB Support .....  | 9  |
| Teradata Support .....                                       | 10 |
| MongoDB: Schema Validation .....                             | 11 |
| AlloyDB Support Summary .....                                | 22 |
| Databricks Partner Connect .....                             | 23 |
| Database Support Updates .....                               | 23 |
| Git Support .....  | 25 |
| Bitbucket Support .....                                      | 25 |
| Denormalizing NoSQL Database Models .....                    | 26 |
| Project Explorer .....                                       | 33 |
| Project Explorer Pane .....                                  | 33 |
| Project Explorer Options .....                               | 35 |
| Managing Projects .....                                      | 36 |
| Using Project Explorer .....                                 | 37 |
| Use Case 1: Create Projects Using Existing Mart Models ..... | 38 |
| Creating Projects .....                                      | 39 |
| Creating Folders .....                                       | 40 |
| Importing Models from Mart .....                             | 41 |
| Deriving Target Models .....                                 | 42 |
| Deriving Physical Models .....                               | 43 |
| Deriving Logical Models .....                                | 44 |

---

|  |    |
|--|----|
| Reverse Engineering Models .....                             | 45 |
| Use Case 2: Create Projects Using New and Local Models ..... | 48 |
| Creating Projects .....                                      | 48 |
| Creating Folders .....                                       | 49 |
| Importing Models .....                                       | 51 |
| Creating Models .....  | 52 |
| Reverse Engineering Models .....                             | 53 |
| Creating Target Models .....                                 | 56 |
| Managing Projects .....                                      | 58 |
| Renaming Objects .....                                       | 58 |
| Deleting Objects .....                                       | 59 |
| Saving Project .....   | 59 |
| Forward Engineering Models .....                             | 60 |
| Generating Project Reports .....                             | 61 |
| Productivity and UI Enhancements .....                       | 64 |
| Copy Family .....  | 65 |
| Demand Loading .....   | 67 |
| erwin ER360 .....  | 69 |
| DM Connect for DI .....                                      | 70 |
| erwin Mart Portal Enhancements .....                         | 70 |
| Enterprise Modeling Compliance .....                         | 70 |
| Policies .....   | 71 |
| Rules .....  | 72 |

---

|   |    |
|---|----|
| Reports .....                                 | 73 |
| Using Enterprise Modeling Compliance .....    | 73 |
| Enterprise Glossary .....                     | 81 |
| Using Enterprise Glossary .....               | 82 |
| Setting up Unity Catalogs in Databricks ..... | 89 |

# Introduction

The Feature Tour guide walks Data Architects, Data Administrators, Application Administrators, Database Administrators, and Partners through the features introduced in erwin Data Modeler (DM) 12.5 release.

The features and enhancements introduced in this release are:

- [MongoDB Support](#)
- [Teradata Support](#)
- [MongoDB: Schema Validation](#)
- [AlloyDB Support](#)
- [Databricks Partner Connect](#)
- [Database Support Updates](#)
- [Git Support](#)
- [Bitbucket Support](#)
- [Denormalizing NoSQL Database Models](#)
- [Project Explorer](#)
- [Productivity and UI Enhancements](#)
- [erwin ER360](#)
- [DM Connect for DI](#)
- [Enterprise Modeling Compliance](#)
- [Enterprise Glossary](#)
- [erwin Mart Portal Enhancements](#)
- [Setting up Unity Catalogs in Databricks](#)

For additional information about a feature, in erwin Data Modeler, click **Help > Help Topics** or press **F1**.



## MongoDB Support

erwin Data Modeler (DM) now supports [MongoDB](#) 5.x and 6.x as a target database. This implementation supports the following objects:

- Database
- Collection
- Field
- Index
- Relationships
- Users
- View

Following are the supported data types:

| Numeric  | String Literals  | Date and Time  | Other   |
|--|--|--|---|
| <ul style="list-style-type: none"><li>• double</li><li>• binary</li><li>• int</li><li>• integer</li><li>• boolean</li><li>• minKey</li><li>• maxKey</li><li>• long</li><li>• decimal</li></ul> | <ul style="list-style-type: none"><li>• string</li></ul> | <ul style="list-style-type: none"><li>• date</li><li>• timestamp</li></ul> | <ul style="list-style-type: none"><li>• object</li><li>• array</li><li>• null</li><li>• objectId</li><li>• regex</li><li>• code</li></ul> |

# Teradata Support

erwin Data Modeler (DM) now supports [Teradata](#) 17.x as a target database. Apart from the existing objects, this implementation supports the following new objects:

- Maps
- Storage Format Schema
- Function Mapping
- Foreign Table
- Time Series Table
- Temporal Tables

Following are the newly supported data types:

- GRAPHIC
- VARCHAR
- VARBYTE
- VARGRAPHIC
- LONG VARCHAR
- LONG VARGRAPHIC
- BINARY LARGE OBJECT | BLOB
- CHARACTER LARGE OBJECT | CLOB
- [SYSUDTLIB.] XML
- [SYSUDTLIB.] JSON
- [SYSUDTLIB.] ST\_GEOMETRY
- [SYSUDTLIB.] MBR

## MongoDB: Schema Validation

erwin Data Modeler (DM) 12.5 now supports schema validation for MongoDB forward engineering scripts. The schema validator verifies fields in collections based on their data type. If enabled, this feature restricts forward engineering script generation in case the data in the fields is not in line with the assigned data type. Thus, enabling you to avoid any errors in the forward engineering script. This feature is enabled by default for fields in collections.

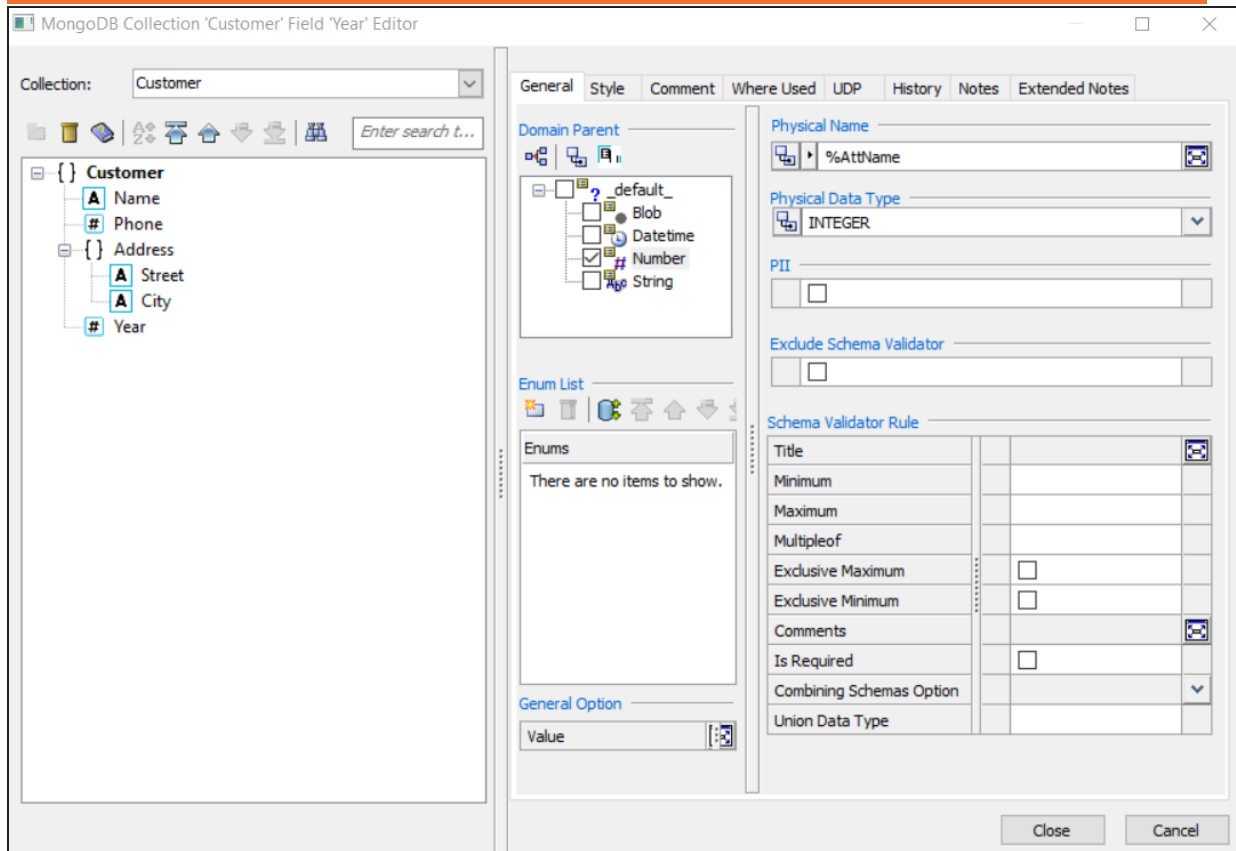
In case of any exceptions, you can select the Exclude Schema Validator check box to exclude validations for one or more fields. Doing so overrides the assigned data type of a field and generates the script successfully.

This topic walks you through the steps to generate schema validation for fields in a simple table (collection) and generate schema using an example. The following table lists the fields and assigned data types in a MongoDB collection, Customer.

| Fields   | Data Type |
|----------|-----------|
| Name     | STRING    |
| Year     | INTEGER   |
| Subjects | STRING    |
| Address  | OBJECT    |
| Street   | STRING    |
| City     | STRING    |
| Year     | INTEGER   |

The following image displays the fields in a collection and the assigned data types.

## Teradata Support



To generate schema validation for a MongoDB collection, follow these steps:

1. In erwin DM, create a collection with fields, and assign relevant datatypes.

Ensure that a database is assigned to the collection.


2. Right-click the collection and click **Field Properties**.  
The MongoDB Field Editor appears.
3. Select a field to set schema validation and other field properties.  
Use the following options under the **General** tab to configure schema validations for a field:

### **Exclude Schema Validator**

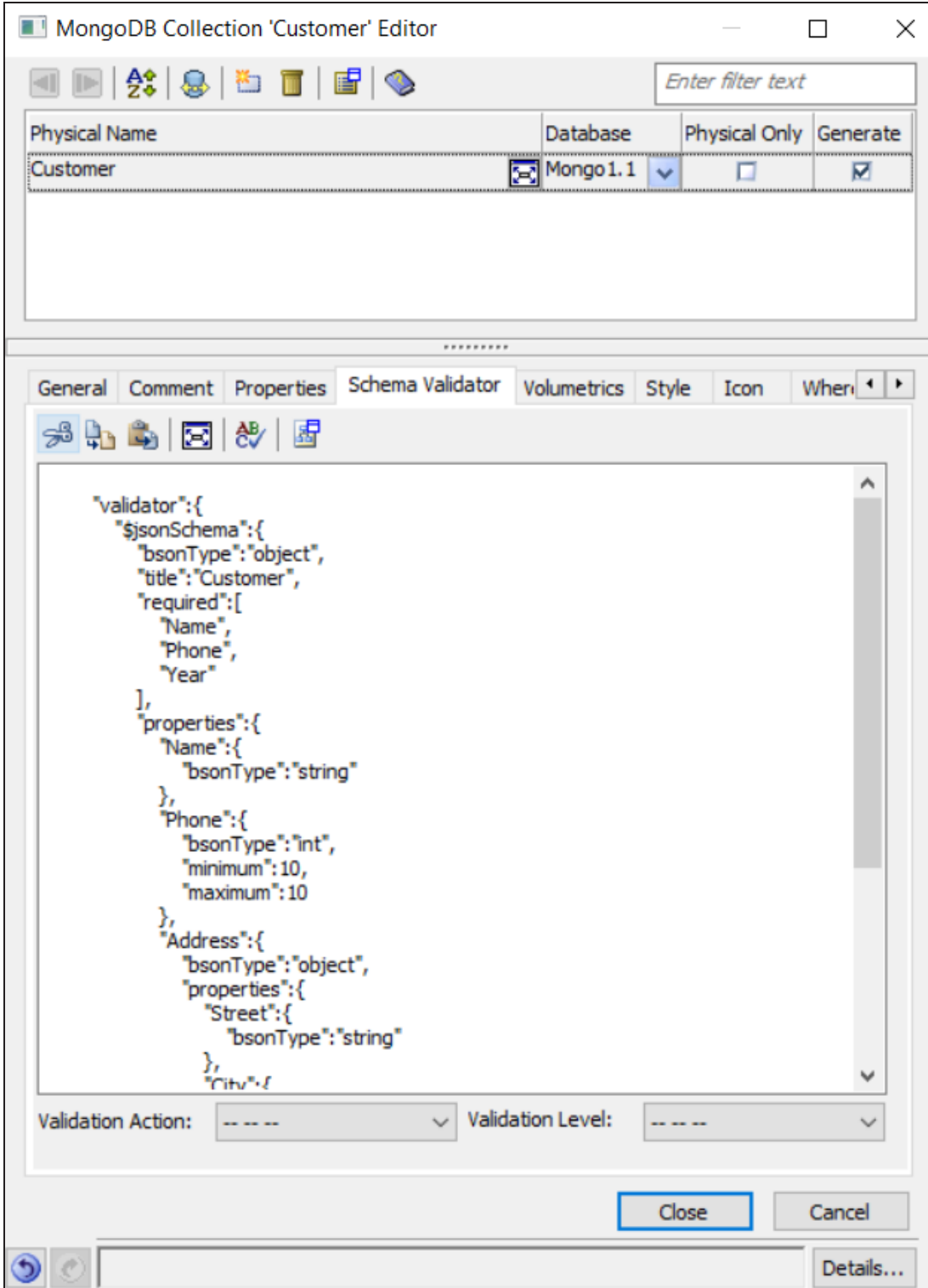
Use this option to exclude any validations set under Schema Validator Rule options. Selecting this option disables the Schema Validator Rule options.

### **Schema Validator Rule**


Use this option to set various validation rules for a field. These options available under this section differ based on field's data type. For more information on schema validation rule options, refer to [Defining MongoDB Fields](#) topic.

4. Under Schema Validator Rule section, set the required validation parameters and click **Close**.
5. Right-click the collection and click **Collection Properties**.  
The MongoDB Collection Editor appears
6. Go to the **Schema Validator** tab.
7. Click  to generate schema validation template.  
This displays the generated schema validation template based on the field properties

and assigned datatypes.



8. Select **Validation Action** and **Validation Level** options. For more information about these options, refer to [Defining MongoDB Collections](#) topic.

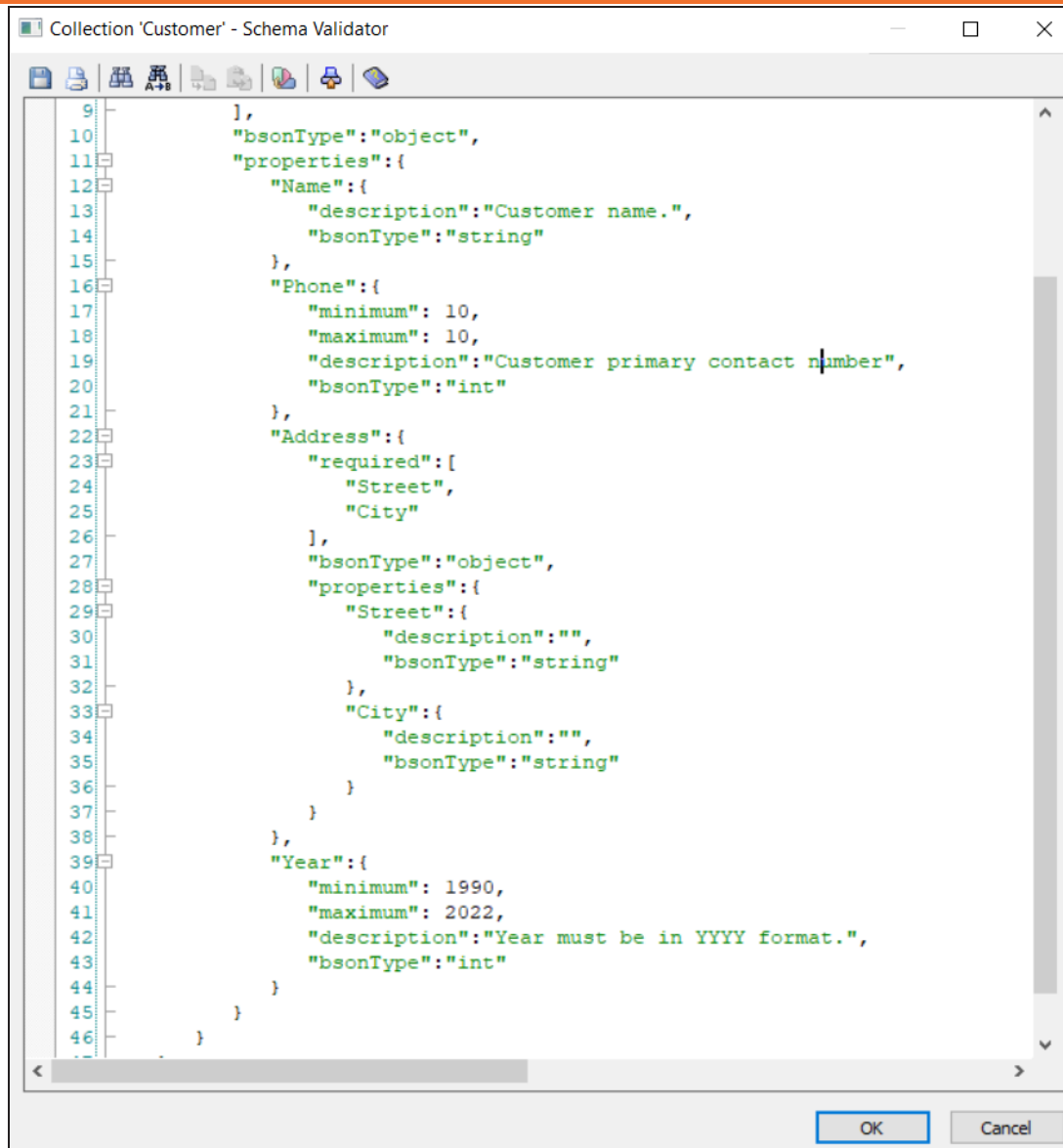
9. Click  to edit the schema.

The Collection Schema Validator editor appears. This displays the generated schema validation template with validations in edit mode.

10. Edit the schema template for validations and add a description for the fields based on your requirement.

In this example, the validation properties of the fields and descriptions are updated as shown in the following image.

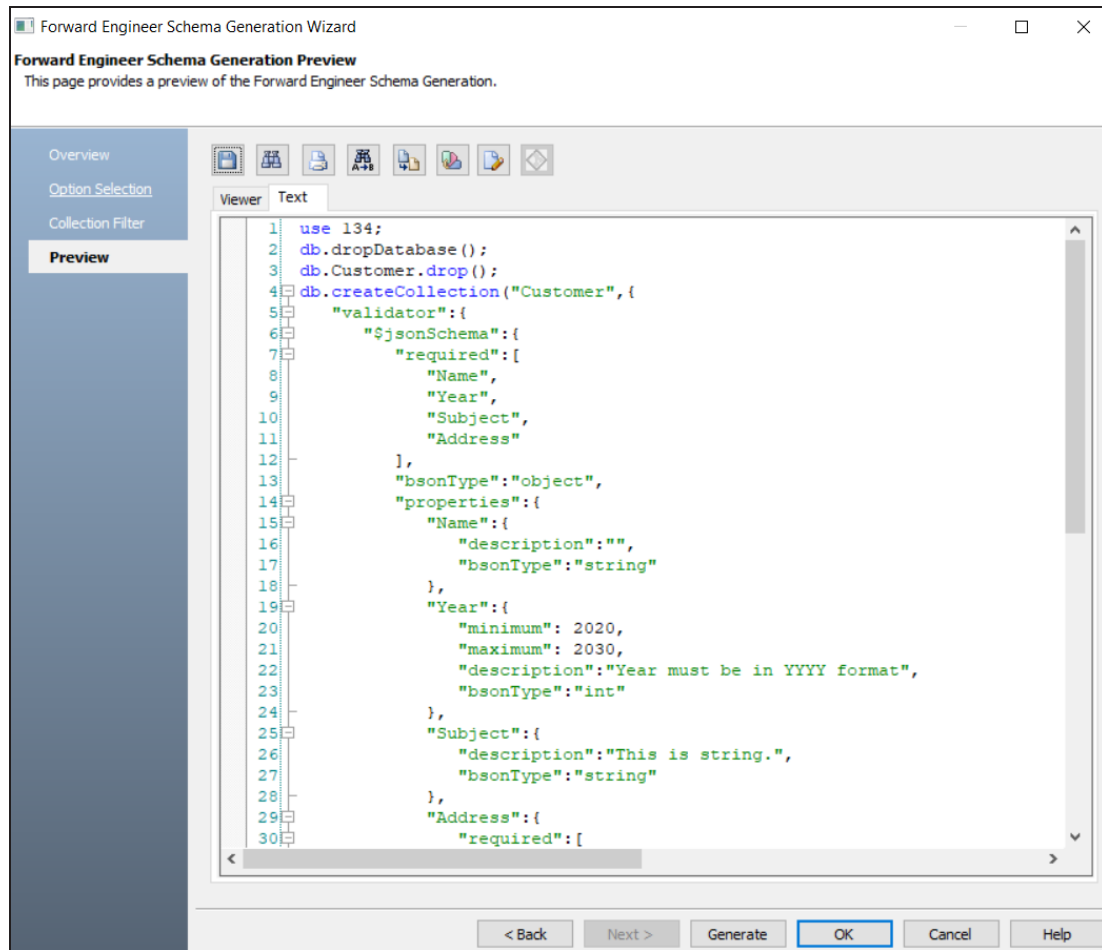




11. Once you have made changes to the schema template, click **OK**.  
The schema template validation changes are saved.
12. On the application ribbon, click **Action > Schema**.  
The Forward Engineer Schema Generation Wizard appears.

13. Go to **Preview** tab.

The updated schema with the validations appear.



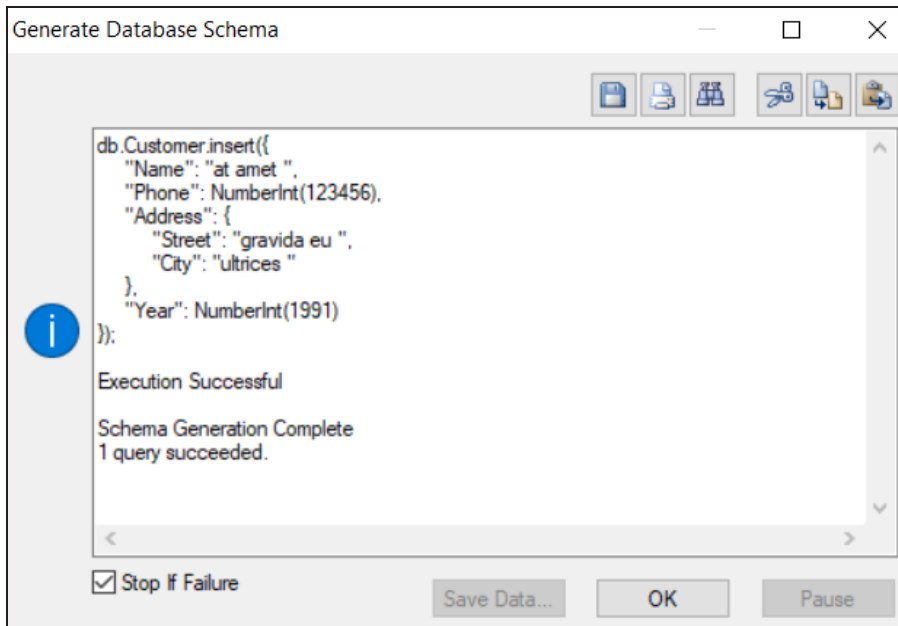
14. In the Preview section, edit the schema to correct the random integer values.

Generating schema validation template adds random integer values to the fields with INTEGER datatype. Ensure that you replace this with valid value to generate the schema successfully.

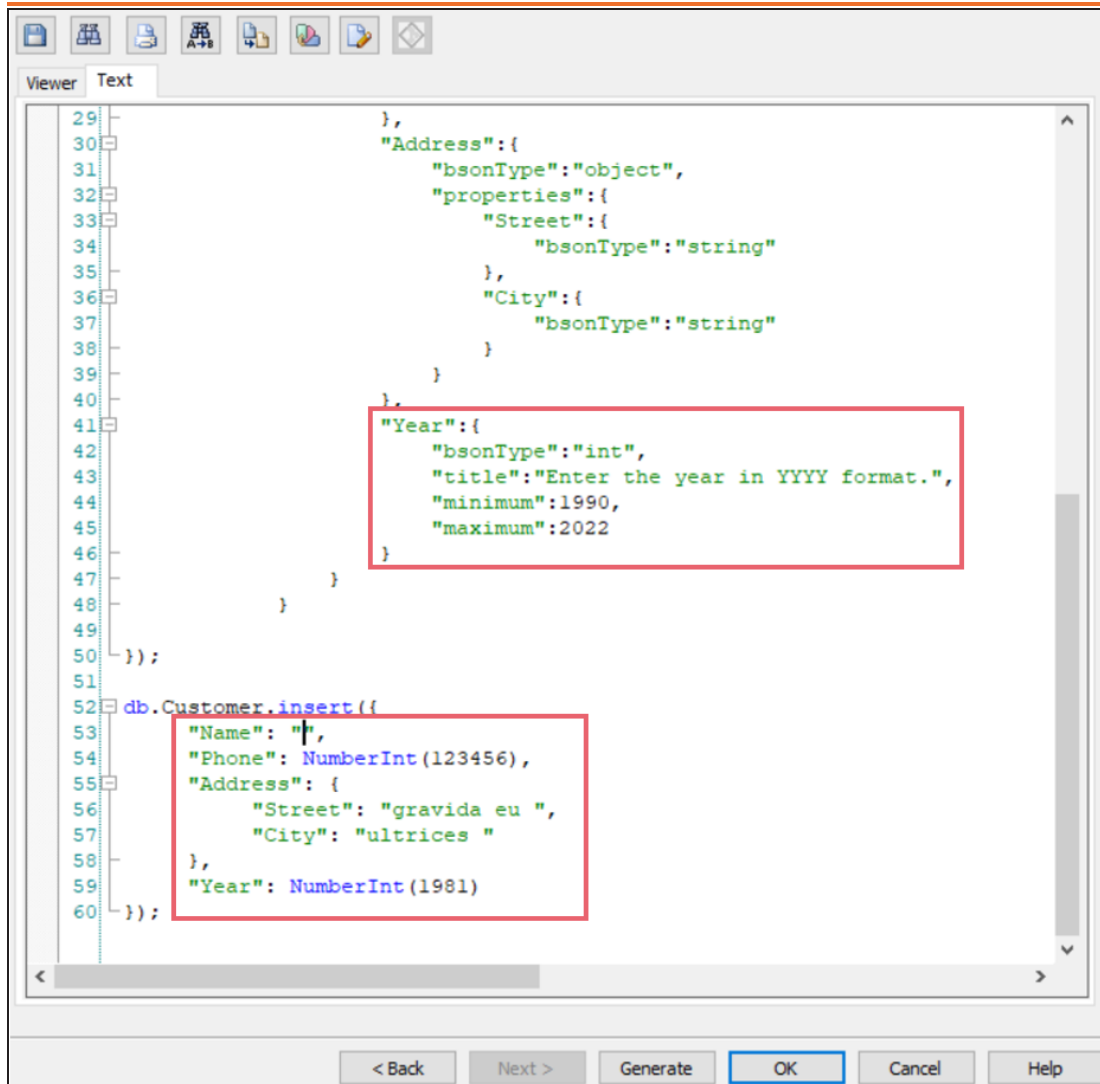
15. Click **Generate**.

The MongoDB Connection screen appears. Connect to the database. For more information, refer to the [Database Connection Parameters](#) topic.

The schema is generated successfully along with the validations.

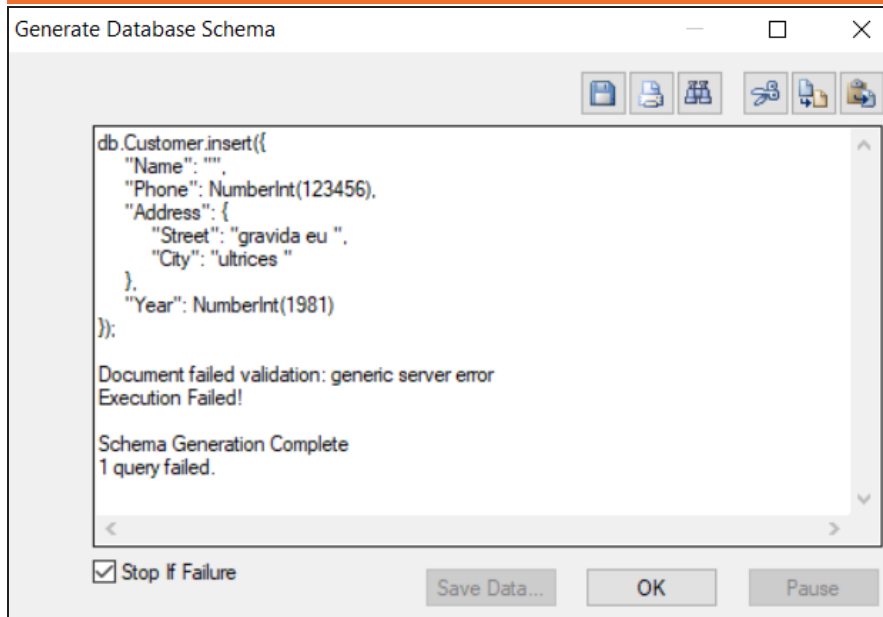


The schema generation wizard displays errors if the field-level validations are not met. For example, the schema in the following image accepts values between 1990 and 2022 in the Year field, and Name field is set as mandatory. The, Year field is updated with the value 1981, and Name field is blank.



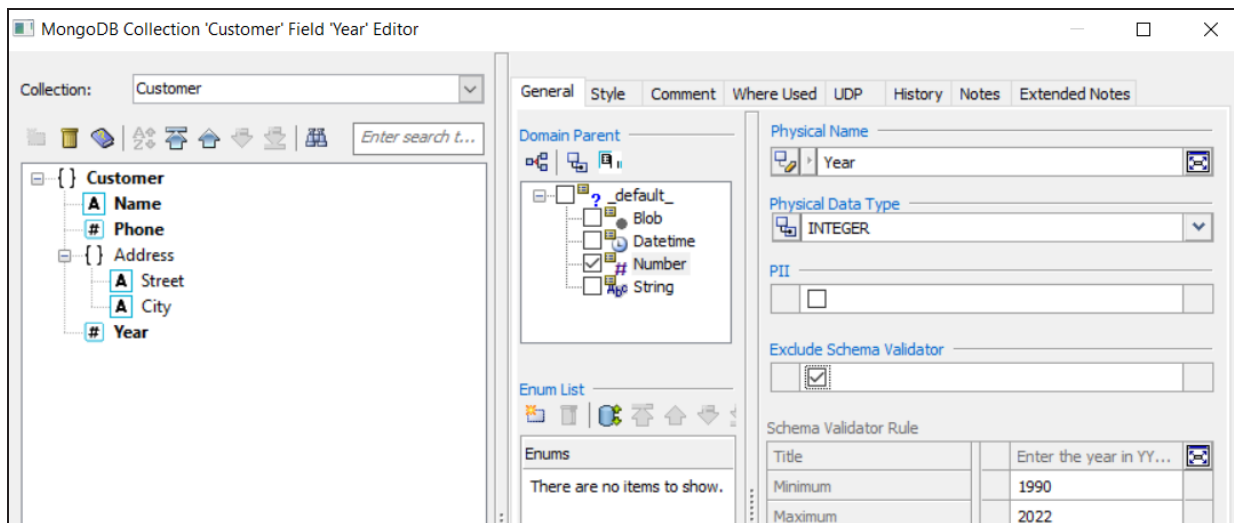
When you click **Generate**, the schema generation fails as the validation rules are not met and displays the following error.

## Teradata Support

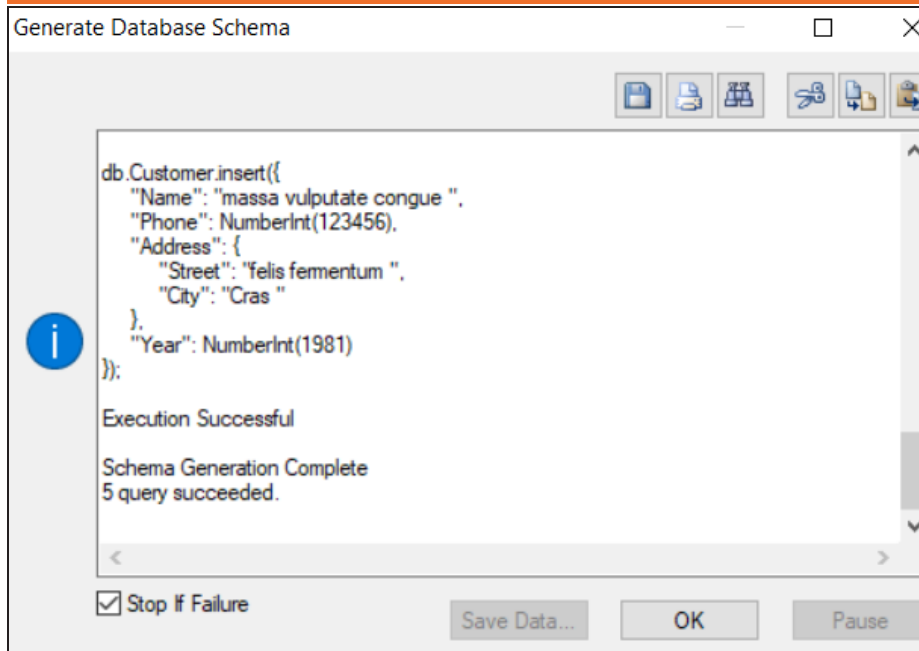


Alternatively, you can exclude validations for one or more fields to generate schema. This overrides the assigned data validation rule for a field and generates the script successfully.

To exclude schema validation for a field, open Collection Field Editor, select a field, and then, select the **Exclude Schema Validator** check box.



In the Forward Engineering Schema Wizard, when you click **Generate**, the schema is generated successfully.



## AlloyDB Support Summary

erwin Data Modeler (DM) now supports [AlloyDB](#) 1.1 a target database. property page editors. This implementation supports the following objects:

- Collations
- Databases
- Domains
- Functions
- Materialized Views
- Permissions
- Roles
- Schemas

## Teradata Support

---

- Sequences
- Tables
- Tablespaces
- Triggers
- User-Defined Types
- Views

For more information about specific AlloyDB property values, refer to the AlloyDB database documentation. The erwin® Data Modeler documentation for the physical property editors provides brief descriptions of the controls on each dialog and tab, which you can use as a point of reference while working with database design features.

As a best practice, use the AlloyDB ANSI ODBC driver for Reverse Engineering from Database (REDB) while using erwin® Data Modeler.

## Databricks Partner Connect

Databricks Partner Connect is now live and available for erwin DM. With this, you can connect to your Databricks database using Username-Password or Personal Access Token-Password authentication methods.

For more information on generating and using Personal Access Tokens, click [here](#).

Databricks Partner Connect is now live and available for erwin DM. Databricks as a target database also supports Databricks Unity Catalog.

## Database Support Updates

Support for the following databases has been updated:

| Database | Enhancements  |
|----------|---|
| ArangoDB | Views tab has been added to the ArangoDB Reverse Engineering Wizard.  |
| Avro     | Precision and Scale properties have been added to the Avro Field Editor. These properties are applicable only to the data type BYTES. |

## Teradata Support

| Database        | Enhancements  |
|-----------------|---|
| Google BigQuery | <ul style="list-style-type: none"><li>▪ Support for Table Search Index has been added. For more information, refer to <a href="#">Defining Google BigQuery Search Indexes</a> topic.</li><li>▪ Views and Materialized Views tabs have been added to the Google BigQuery Reverse Engineering Wizard.</li><li>▪ Google BigQuery ER diagrams have been updated to display the property mode of the column (NULL, NOT NULL, REPEATED/ARRAY), if the Display Column Null Option is selected in the ER Diagram Editor.</li><li>▪ You can now set physical table names of up to 1024 characters.</li><li>▪ You can now set physical column names of up to 300 characters.</li><li>▪ Default Collation property has been added to the Google BigQuery Dataset Editor.</li><li>▪ Default Collation property has been added to the Google BigQuery Table Editor.</li><li>▪ Collation and Default Expression properties have been added to the Google BigQuery Table Column Editor.</li><li>▪ Partition Options properties have been moved to the Partitions tab Google BigQuery Table Editor.</li><li>▪ Partition Expression section has been removed from the Partitions tab in the Google BigQuery Table Editor and Materialized View Editor. The partition expression is now displayed on the NoSQL tab in these Editors.</li><li>▪ Values in the Partition Expression Options now automatically resets when a column of a different Physical Data Type is selected on the Partition Columns section in the Google BigQuery Table Editor and Materialized View Editor.</li><li>▪ Option Selection tab has been updated, and Owner Override tab has been added to the Google BigQuery Forward Engineering Schema Generation Wizard.</li></ul> |
| Cassandra       | Materialized Views tab has been added to the Cassandra Reverse Engineering Wizard.  |
| Couchbase       | erwin Data Modeler (erwin DM) now supports using Couchbase with Python SDK instead of C SDK.  |



## Teradata Support

| Database | Enhancements  |
|----------|---|
| DynamoDB | <ul style="list-style-type: none"><li>You can now set physical table and index names of up to 255 characters.</li><li>Session Token authentication has been added for DynamoDB database connection.</li></ul>   |
| Redshift | Support for Redshift Encoding AZ64 has been added.  |
| MongoDB  | <ul style="list-style-type: none"><li>Schema validation options for forward engineering a model has been enhanced. For information on schema validation, refer <a href="#">MongoDB: Schema Validation</a>.</li><li>Display Field Null Option property has been replaced with Display Field is Required Option property on the MongoDB ER Diagram Editor.</li></ul>  |
| JSON     | <ul style="list-style-type: none"><li>Display Field Null Option property has been replaced with Display Field is Required Option property on the JSON ER Diagram Editor.</li><li>JSON Array Element option has been added to JSON Reverse Engineering Wizard to specify whether all or only first element of a JSON array is processed during reverse engineering.</li><li>Viewer tab in the JSON Forward Engineering Schema Generation Wizard has been updated to display the names of JSON objects and arrays and the number of child elements inside them.</li></ul> |

## Git Support

You can now view the DDL of a model version and compare it with another version of the model from the Catalogs pane.

## Bitbucket Support

You can now connect to a Bitbucket repository to push forward engineering scripts for a Mart Model to the Git repository. Bitbucket enables you to utilize features, such as version control and collaboration with team members. For information on connecting to Bitbucket, refer to the [Connecting to Git Repositories](#) topic.

# Denormalizing NoSQL Database Models

You can now perform advanced denormalization when you derive non-relational database models. This feature provides you with options to do a manual or automatic denormalization.

- Use the auto-denormalization option to merge tables with the target table automatically. This embeds the tables with one-one relationships as User Defined Type and one-to-many relationships as normal columns.
- Use the manual denormalization option to selectively merge columns from source tables to target tables. Further, manual denormalization provides you options to merge multiple columns into single combined column, decide the column embedding type and the embedding process, retain relationships and much more.

## Denormalizing Models

erwin Data Modeler 12.5 supports denormalization options for NoSQL databases in addition to Cassandra. This allows you to select the objects for the denormalization of a NoSQL database model before it is derived from a source model. It enables you merge the objects in the source model into the target NoSQL model based on the requirements.

The denormalization options for NoSQL databases appear only when the Advanced Denormalization option is selected while deriving a model. Once the model is derived, the Denormalization Wizard for the model appears.

The objects that you can denormalize differ based the type of a NoSQL database. Refer to the following table for the list of supported databases, objects, and its name mappings for denormalization.

| Database         | Object Names      |
|------------------|-------------------|
| Amazon Keyspaces | Tables, Columns   |
| Avro             | Documents, Fields |
| Cassandra        | Tables, Columns   |

## Denormalizing NoSQL Database Models

---

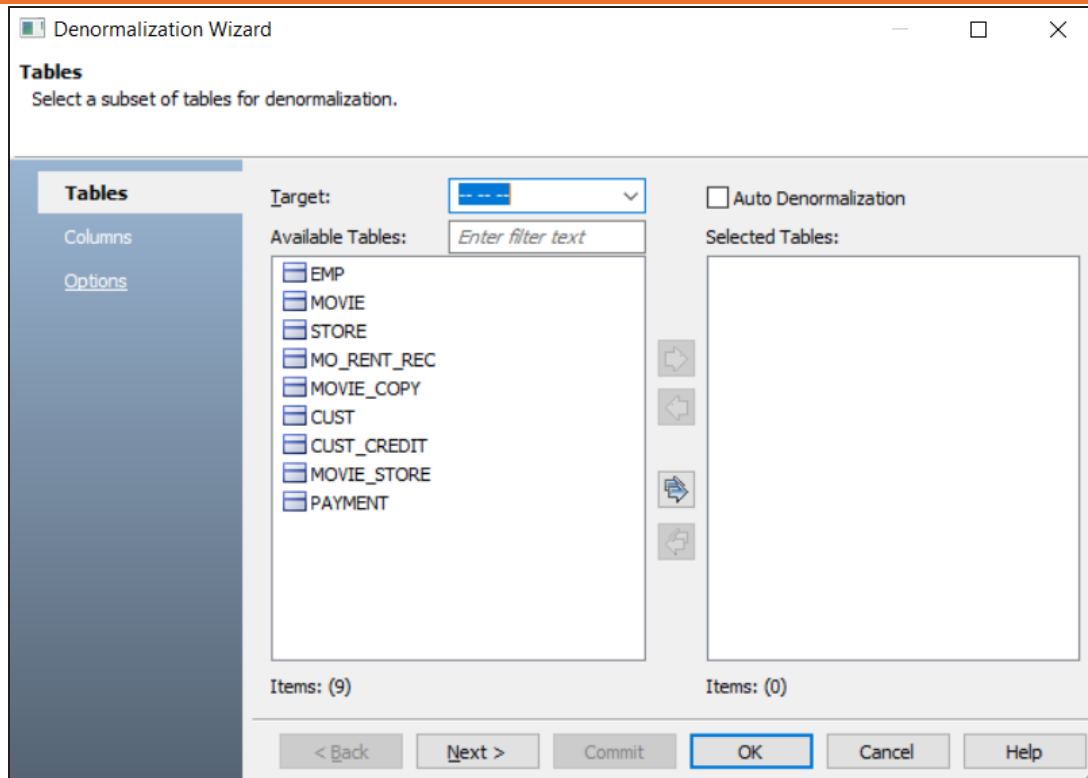
| Database        | Object Names         |
|-----------------|----------------------|
| Couchbase       | Tables, Items        |
| DynamoDB        | Tables, Items        |
| Google BigQuery | Tables, Columns      |
| JSON            | JSON Objects, Fields |
| MongoDB         | Collections, Fields  |
| Parquet         | Records, Fields      |

This topic walks you through the advanced denormalization options for Cassandra database. Similarly, you can denormalize other supported NoSQL databases using the advanced denormalization feature.

To denormalize the model further, follow these steps:


1. On the Tables section, click the **Target** drop-down to select a target table. All the tables will be merged into the selected target table.

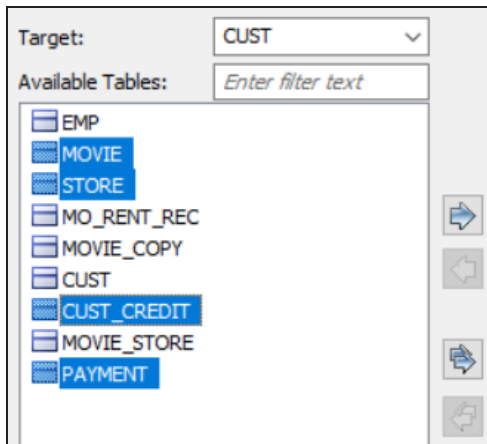
## Denormalizing NoSQL Database Models



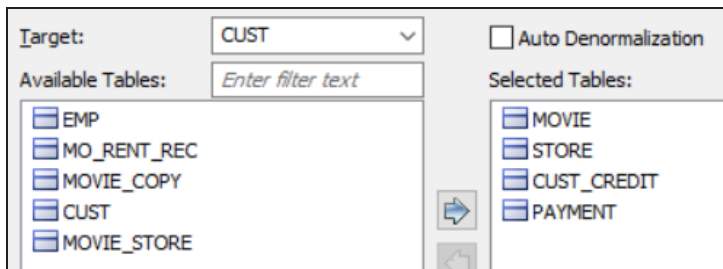
Select **Auto Denormalization** option to merge tables with the target automatically. This embeds the tables in the model with one-one relationships as User Defined Type and one-to-many relationships as normal columns.

## Denormalizing NoSQL Database Models

2. Under **Available Tables**, select one or more tables to merge. Then, click .



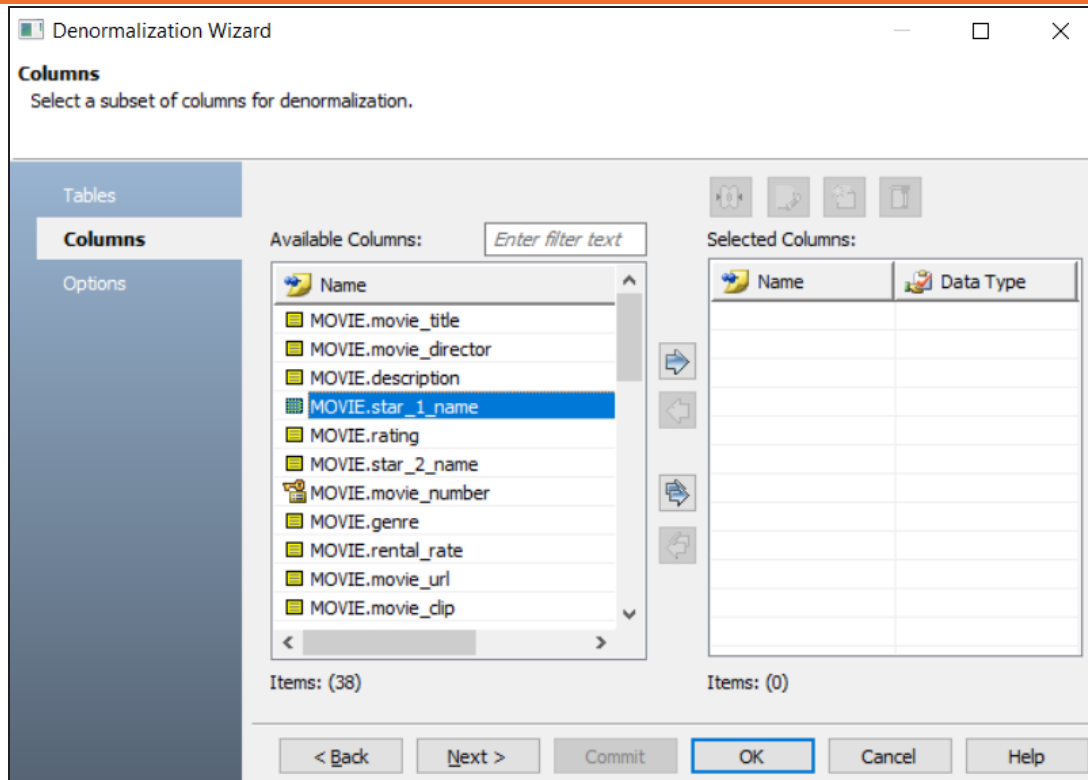
This moves the selected tables under Selected Tables.




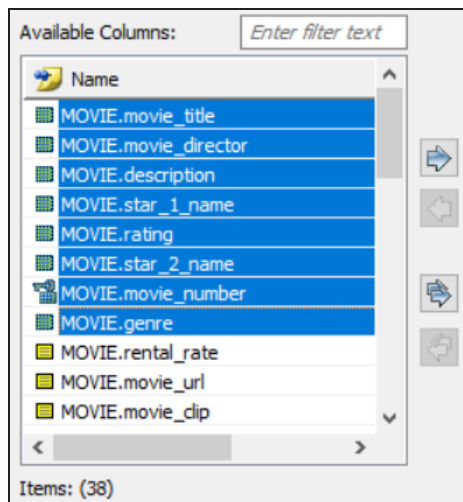
3. Click **Next**.

The Column section appears. It displays a list of available columns.

## Denormalizing NoSQL Database Models

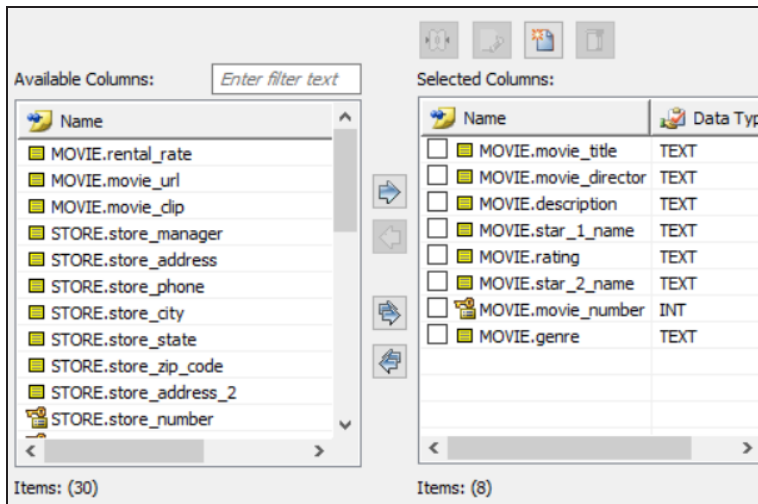


4. Under **Available Columns**, select the that you want to merge. Then, click .



## Denormalizing NoSQL Database Models

This moves the selected databases under Selected Columns.



Once you have added the selected columns, you can use any of the following options:

### Merge (🔗)

Use this option to merge the selected columns and create a new column under Selected Columns.

### Update (🔧)

Use this option to edit column details such as column name, domain parent, and data type for a selected column.

### New (📄)

Use this option to add a new column under Selected Columns.

### Delete (🗑️)

Use this option to delete the selected columns.

5. Click **Next**.

The Options section appears.

6. Select an **Embedding Type**.

You can select the following embedding options:

## Denormalizing NoSQL Database Models

---

- **Embed as Auto:** Use this option to embed tables through an auto-mechanism based on one-to-many and one-to-one relationships.
  - **Embed as Normal:** Use this option to embed collections using normal column styles.
  - **Embed as UDT:** Use this option to embed collections using a User Defined Type (UDT) styles.
7. Select **Relationships** option to include table relationships to the model.
  8. Select **Cascading** options to determine how multiple collections are merged into a single collection.  
Use the following cascading options:
    - **All:** Use this option to denormalize all relationship levels in a collection into a single collection.
    - **Levels:** Use this option to specify the number of levels up to which collections are denormalized into one collection. For example, if you set Level to 1, all the collections up to level 1 in the relationship hierarchy are denormalized into a single collection.
    - **Auto Cleanup:** Use the option to delete the source collection after denormalization.
  9. Click **OK**.

The denormalization process starts and displays objects based on the denormalization option.

Alternatively, you can click **Commit** to apply changes to the model without exiting the Denormalization Wizard.



# Project Explorer

Using the Project Explorer feature in erwin Data Modeler, you can now bring multiple data models in your enterprise applications together and manage them as a single project. All models and objects in the project are stored at one location of your choice. You can also choose to save the entire project in erwin Mart, which enables online team collaboration, data governance, and consistency for the project.

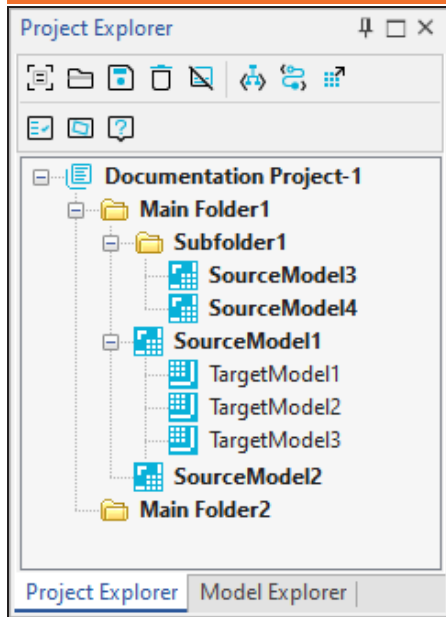
Projects eliminate the need for loading and managing required models separately. Instead, it provides you with a single, unified project metadata file using which you can launch, organize, and manage project objects.

## Project Explorer Pane

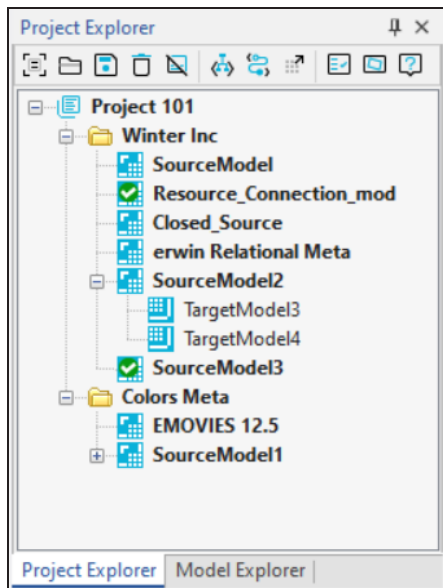
The Project Explorer can be located on the erwin DM user interface as a dedicated pane as well as a ribbon tab. The Project Explorer pane provides an organized view of your project and the objects in it. All physical and logical models in projects are arranged as Source (Parent) models and Target (Child) models. There are no constraints on the number of models in a project. You can have multiple source models in the project, and multiple target models under each source model. You can further organize your models using folders and sub-folders.

By default, this pane opens on the left-side of the application. You can also access it via **View > Panes > Project Explorer Pane** on the ribbon.

## Project Explorer



The models that have connection parameters configured for reverse engineering are indicated using a green check mark. The below screenshot displays the a green check mark on the modes with configured connection parameters.



### Project Explorer Options

The Project Explorer provides you with the following options on the toolbar and ribbon:

**New Project** 

Use this option to create projects. For more information, refer to the [Creating Projects](#) topic.

**Open Project** 

Use this option to open existing projects.

**Save Project** 

Use this option to save the project updates.

**Delete** 

Use this option to delete the selected objects in projects.

**Close Project** 

Use this option to close projects.

**Schema Script** 

Use this option to generate forward engineering schema scripts for models in projects.

**Alter Script** 

Use this option to generate alter scripts for models in projects.

**Save to Mart** 

Use this option to save projects to mart.

**Report** 

Use this option to generate project reports. For more information, refer to the [Generating Project Reports](#) topic.

**Properties** 

## Project Explorer

---

Use this option to edit the project or object properties.

## Managing Projects

Using the Project Explorer, you can perform the following actions to manage your projects:

- Create folders
- Create models
- Reverse engineer models
- Create target models
- Derive target models
- Sync target models with source model
- Rename objects
- Delete objects

For more information, refer to the [Using Project Explorer](#) topic.

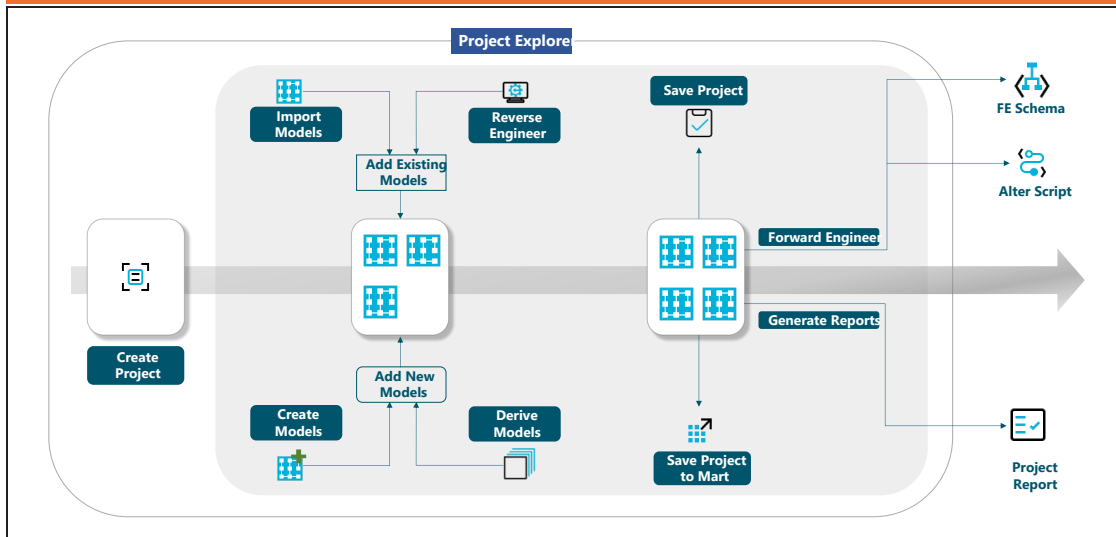
# Using Project Explorer

The Project Explorer provides an end-to-end solution to create and control a project, add objects to it, manage and modify these objects, and generate detailed project reports.

Creating and using projects in erwin DM generally involves the following steps:

- Creating projects
- Creating folders
- Importing models
- Creating models
- Reverse engineering models
- Creating target models
- Deriving models
- Managing projects
  - Renaming objects
  - Deleting objects
  - Saving projects
- Forward engineering models
- Generating project reports

## Using Project Explorer



To understand the feature further, consider two separate use cases of projects. Note that these use cases need not be mutually exclusive and may often coincide in real life usage.

### Use Case 1: Create Projects Using Existing Mart Models

In this example, assume that you have one logical model, SourceModel1, stored in your mart. You aim to derive a set of logical and physical models from it, arranged in a hierarchy of source and target models, and manage them together in erwin DM.


Normally, all derived models are saved separately from their source models, and when you need to work on them, you'll open them individually. erwin DM displays the opened models on the window. However, erwin DM cannot display the multi-level hierarchy of the source and target models. Additionally, managing large numbers of models on DM screen is hard and time-consuming.

Using projects, you can overcome these challenges. The Project Explorer pane arranges all models in a project in a comprehensible tree structure, where target models are arranged under their respective source models. There are multiple levels of source-target branches to help you understand the hierarchy of these models. Also, using the tree structure in the Project Explorer pane, you can identify and manage the required models.

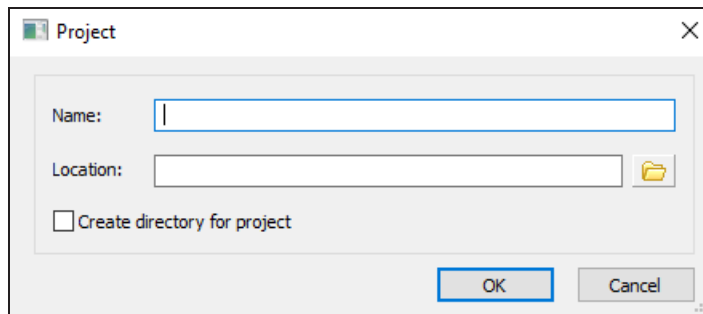
To start, create a project in erwin DM Project Explorer.

### Creating Projects

To create a project, follow these steps:

1. In the Project Explorer, click **New Project** .

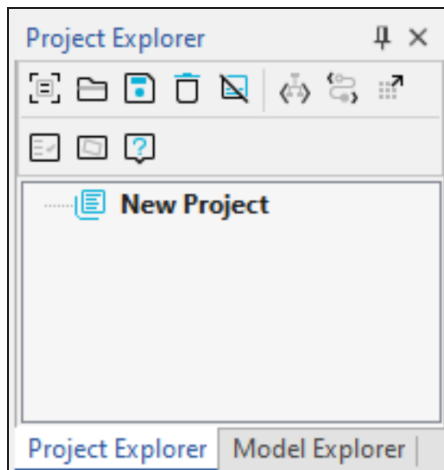
The Project window appears.



2. Enter a name and location for the project.
3. Select **Create directory for project**, if you want to create a new directory in the selected project location.
4. Click **OK**.

The new project is created.

When you create a project, it automatically opens in the Project Explorer pane.



Initially, the explorer pane is blank because the project does not contain any objects in it.

## Using Project Explorer

---

Before populating the project with models, you must create a folder to contain them.

You can use more folders to organize your models. This is especially useful if your project contains a significant number of models.

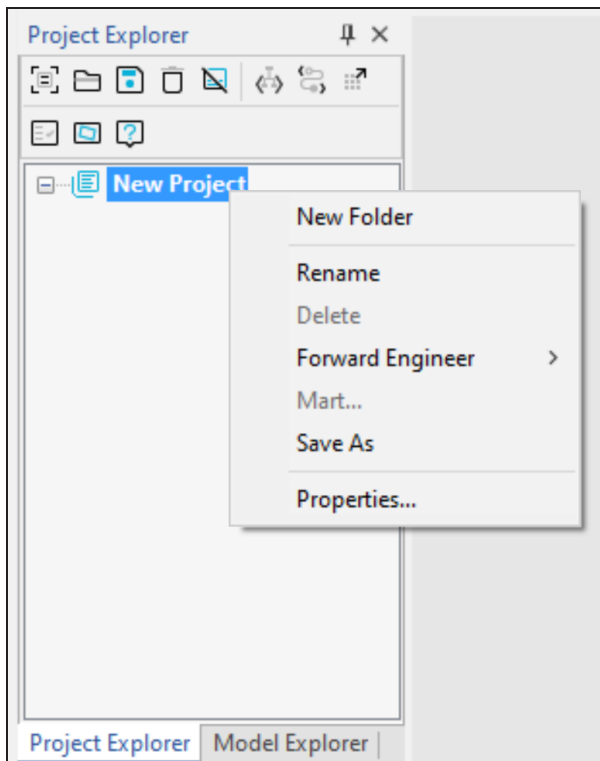
Start by creating a folder to contain the model, SourceModel1 (Logical).

## Creating Folders

You can add any number of folders and subfolders in a project, but you cannot create more than one level of subfolders in a folder.

To create a new folder, follow these steps:

1. In the Project Explorer, right-click the project name.



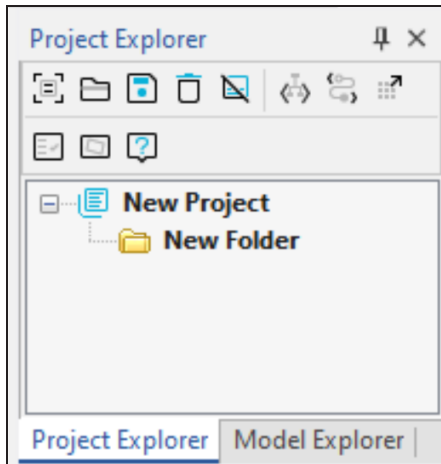
2. Click **New Folder**.
3. Enter a name for the folder, and press **Enter**.



## Using Project Explorer

---

The new folder is added to the project. You can see the new folder under the project in the Project Explorer pane.



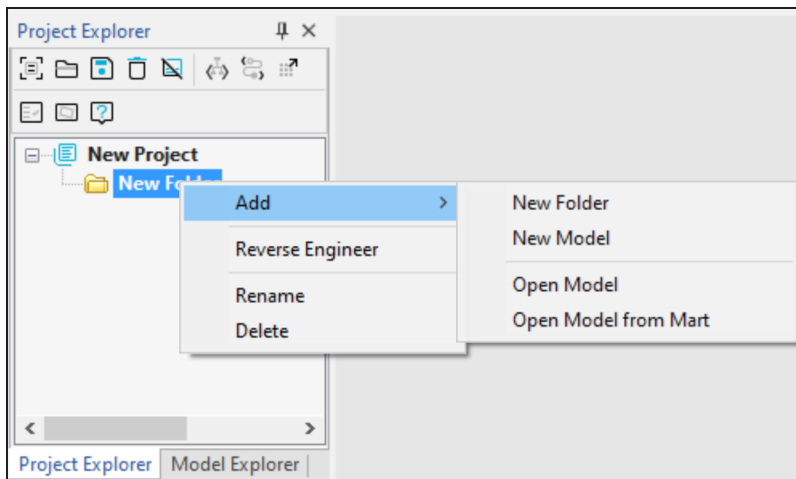
Now, you can import the model, SourceModel1, into the folder.

## Importing Models from Mart

To import a model from mart, you must first connect to erwin Mart.

Once you are connected, to import the model, follow these steps:

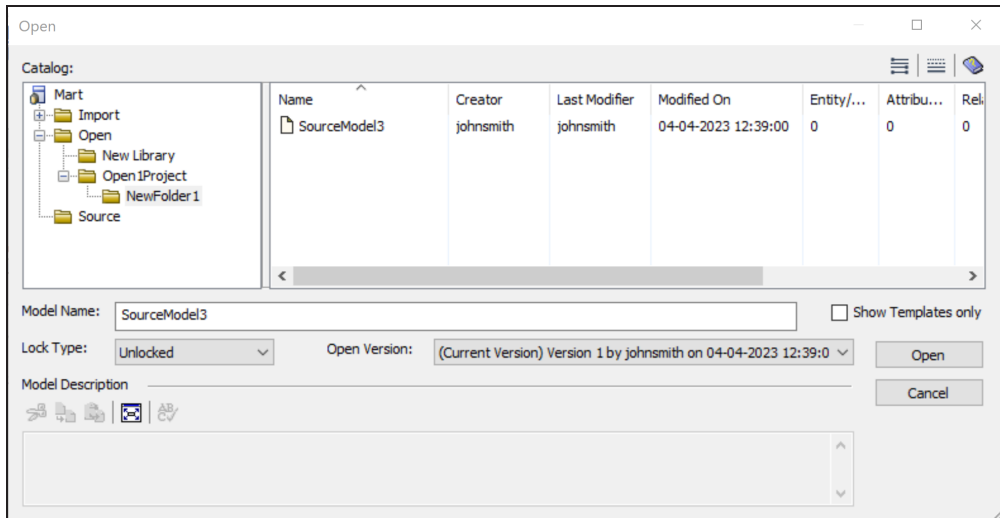
1. Right-click the folder that you created.



2. Click **Add > Open Model from Mart**.

## Using Project Explorer

A mart catalogs window appears.



3. Select the model file.
4. Click **Open**.

The model is imported to the project. You can find SourceModel in the folder that you created.

Once you have your source model imported into your project, you can derive more logical and physical models from it.

## Deriving Target Models

You can derive any model type from an existing model. Some of the more common derive scenarios are:

- Multiple physical models from a logical model. You can use a single conceptual logical model to derive a physical model for different target servers.
- A generic physical model from a logical model. A generic physical model is a model in which you specify DBMS-independent design decisions, and indicate ODBC/generic as the target server.
- A physical model from a logical model. Your new physical model can be customized for a target database and version, and to enforce naming standards.

## Using Project Explorer

- A logical model from a logical model. For example, you can derive a new logical model based on a subject area from the source model.

The model that you open before you use the wizard is known as the source model. The source model contains all model objects that you can include in a derived, or target model.

For more information, refer to the [Design Layers and Model Derivation](#) topic in the Online Help.

Start by deriving a physical model, TargetModel1, and a logical model, TargetModel2, from the source mart model.

## Deriving Physical Models

To derive a physical model from SourceModel1, follow these steps:

1. Right-click SourceModel1.
2. Click **DL-Derive**.

The Derive Model window appears.

Derive Model

**Select the Target Model**  
Please select the options to create a new derived model

Compare Level: Unknown

[Overview](#)  
[Source Model](#)  
**[Target Model](#)**  
[Type Selection](#)  
[Object Selection](#)  
[Naming Standards](#)

New Model Type  
☐ Logical ☐ Physical ☒ Logical/Physical

Create Using Template:  
Blank Logical/Physical Model

Remove Browse File System... Browse Mart...

Creates a new model with both logical and physical levels (erwin DM classic) and default settings.

Target Database  
Database: MongoDB Version: 4.x

☒ Advanced Denormalization

< Back Next > Derive Close Help

## Using Project Explorer

---

3. On the **Target Model** tab, under New Model Type, select the model type Physical or Logical/Physical.
4. Select the required database and database version.
5. Enable **Advanced Denormalization**, if required.
6. Enter appropriate values on the other tabs.
7. Click **Derive**.
8. If you have enabled Advanced Denormalization, the Denormalization wizard appears. You can select the objects such as tables and columns for denormalization and click **OK**.

Once you have completed the denormalization process, the new model is added to the project.

You can find the derived model under SourceModel1. By default, erwin DM provides a name for the derived model. You can rename the model to TargetModel1 using the Rename option in Project Explorer.

## Deriving Logical Models

To derive a logical model from SourceModel1, follow these steps:

1. Right-click SourceModel1.
  2. Click **DL-Derive**.
- The Derive Model window appears.
3. On the **Target Model** tab, under New Model Type, select the model type Logical or Logical/Physical.
  4. Enter appropriate values on the other tabs.
  5. Click **Derive**.

The new derived model is added to the project.

You can find the derived model under SourceModel1. By default, erwin DM provides a name for the derived model. You can rename the model to TargetModel2 using the Rename option in Project Explorer.

## Using Project Explorer

---

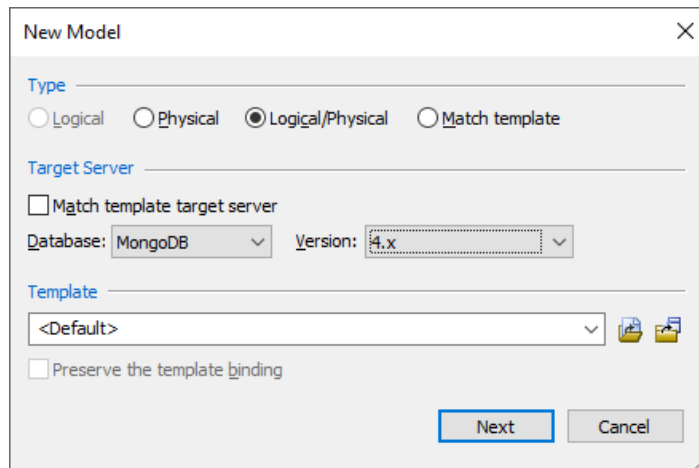
In addition to importing and deriving models, you can also reverse engineer database scripts to add models to projects.

## Reverse Engineering Models

To reverse engineer a model from a database script, follow these steps:

1. Right-click the folder in which you want to add the reverse engineered model.
2. Click **Reverse Engineer**.

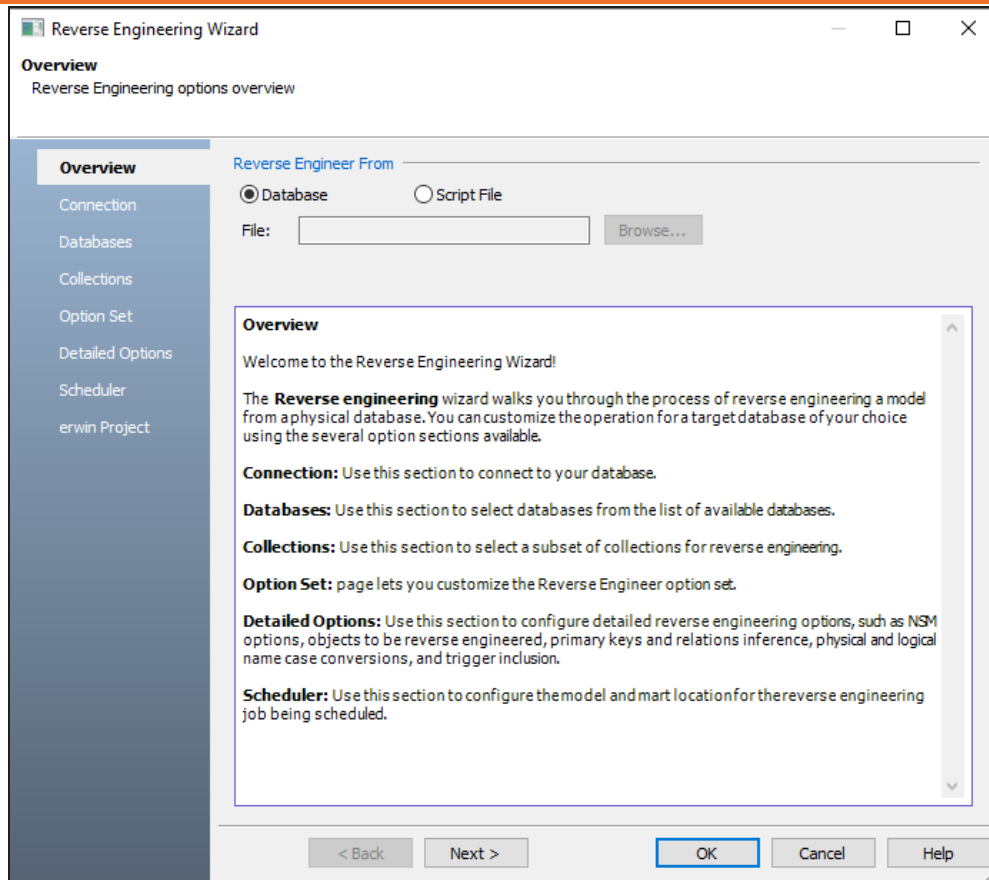
The New Model window appears.



3. Select the type of the model. Available options are Physical, Logical/Physical, and Match Template.
4. Select the Target Server database and version.
5. Click **Next**.

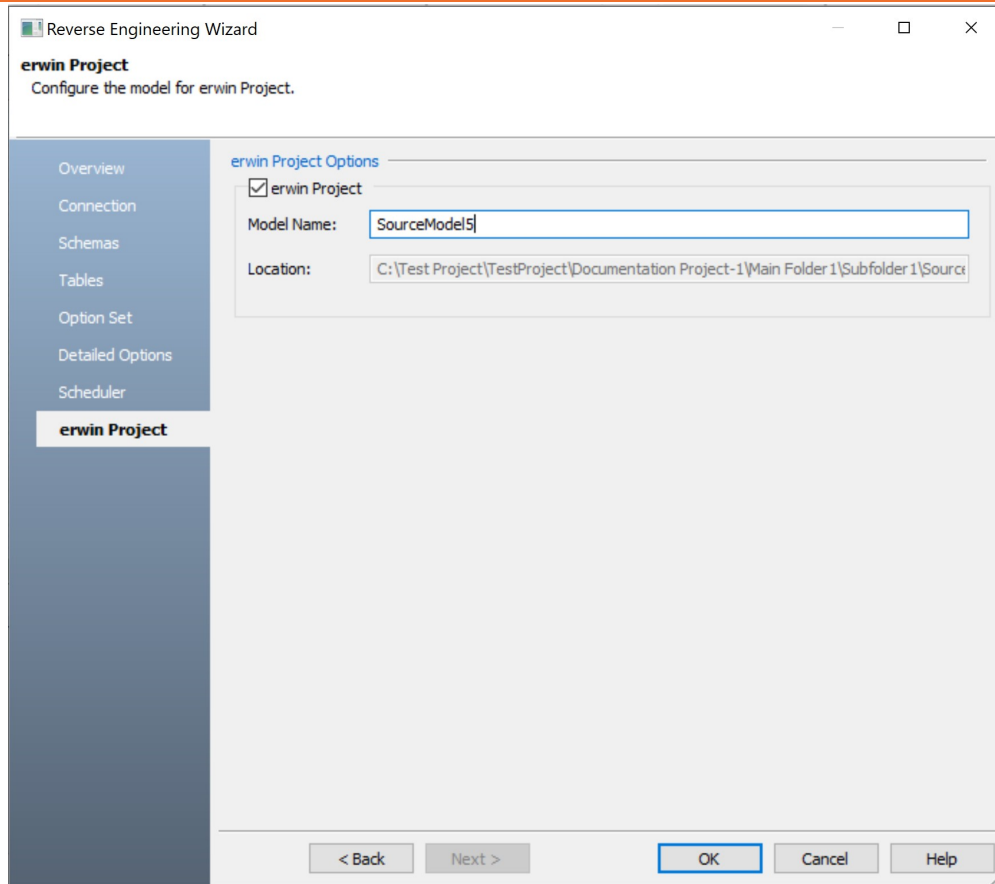
The Reverse Engineering Wizard appears.

## Using Project Explorer



6. On the **Overview** tab, browse and select the script file from your storage.
7. Enter appropriate values in the fields for the other tabs.
8. On the **erwin Project** tab, select **erwin Project** to specify the model belongs to a project.

## Using Project Explorer



9. Enter a name for the model.

10. Click **OK**.

The model is added to the project. You can find the reverse engineered model in the folder you selected.

Now that you have the project and the models ready, you can start working on the models as required.

At any point of time, you can perform the following actions on the project, using the Project Explorer:

- Rename an object
- Delete an object
- Save the project

## Using Project Explorer

---

- Forward engineer models in the project
- Generate project report

To read more about managing projects, refer to the [Managing Projects](#) section.

## Use Case 2: Create Projects Using New and Local Models


In this example, assume that you have two models from a business application stored in your local system; SourceModel1, a logical model, and SourceModel2, a physical model. To work on the application, you need to open them separately from their respective locations. To collaborate with your team, you need to upload them to Mart. You also must save each model every time you finish working.

However, using projects, you can reduce the effort required to manage these models separately. You can also organize these models at a single location and add more models and objects to the project as required. Apart from this, you can simply save the project to your mart.

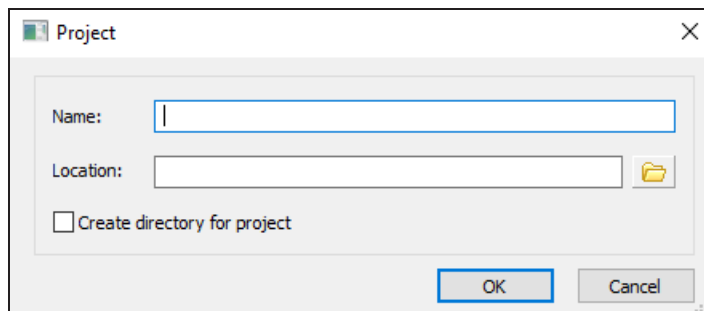
To start, create a project in erwin DM Project Explorer.

## Creating Projects

To create a project, follow these steps:

1. In the Project Explorer, click **New Project** .

The Project window appears.



2. Enter a name and location for the project.



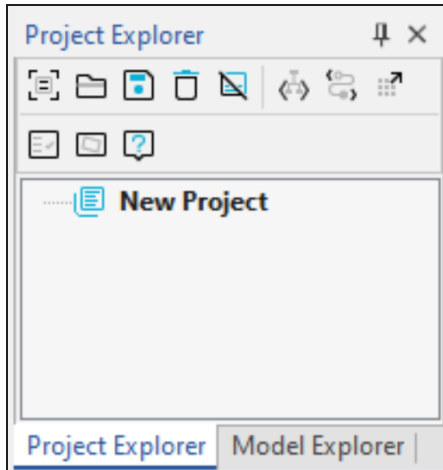
## Using Project Explorer

---

3. Select **Create directory for project**, if you want to create a new directory in the selected project location.
4. Click **OK**.

The new project is created.

When you create a project, it automatically opens in the Project Explorer pane.



Initially, the explorer pane is blank because the project does not contain any objects in it.

Before populating the project with models, you must create a folder to contain them.

You can use more folders to organize your models. This is especially useful if your project contains a significant number of models.

Start by creating a folder to contain the models, SourceModel1 (Physical) and SourceModel2 (Logical).

## Creating Folders

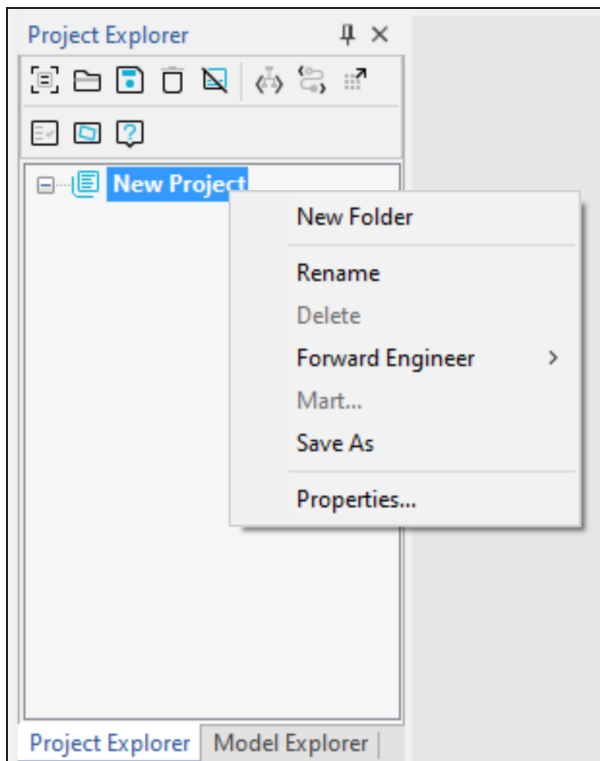
You can add any number of folders and subfolders in a project, but you cannot create more than one level of subfolders in a folder.

To create a new folder, follow these steps:

## Using Project Explorer

---

1. In the Project Explorer, right-click the project name.

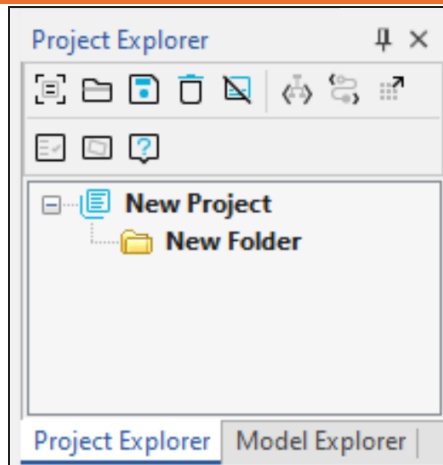


2. Click **New Folder**.
3. Enter a name for the folder, and press **Enter**.

The new folder is added to the project. You can see the new folder under the project in the Project Explorer pane.

## Using Project Explorer

---



Now, you can import the two models (SourceModel1 and SourceModel2) into the folder.

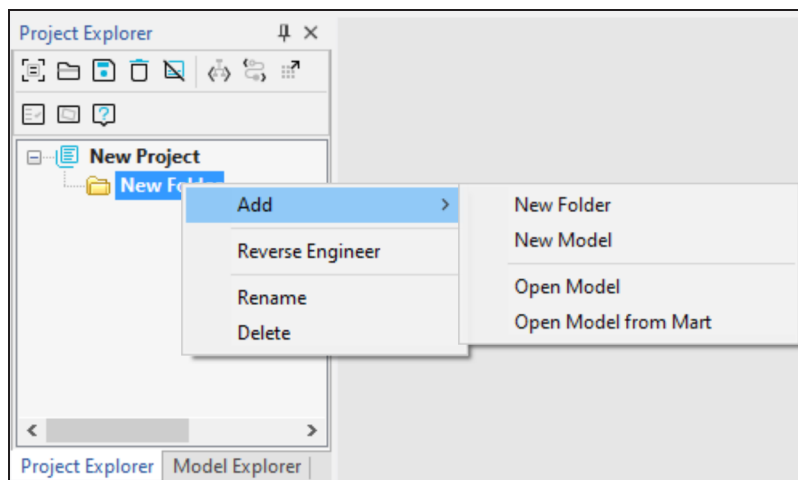
## Importing Models

You can import models from your local storage.

In this example, SourceModel1 and SourceModel2 are in a local folder.

To import models from local folders, follow these steps:

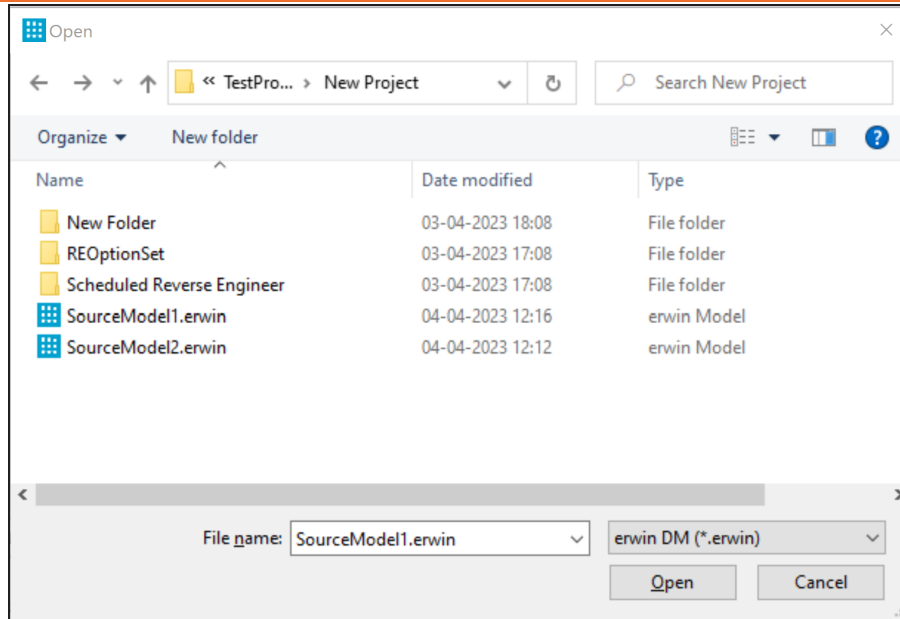
1. Right-click the folder that you created.



2. Click **Add > Open Model**.

A file explorer window appears.

## Using Project Explorer



3. Select the model files, SourceModel1.erwin and SourceModel2.erwin.
4. Click **Open**.

The two models are imported to the project. You can find SourceModel1 and SourceModel2 in the folder that you created.

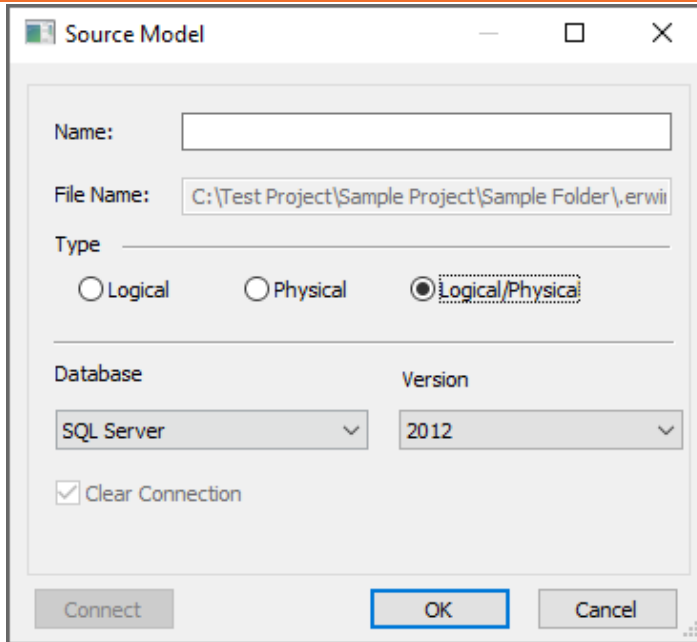
In addition to importing existing models, you can also create new models in the project. For this example, create a new model in the folder that you created.

## Creating Models

To create a new model, follow these steps:

1. Right-click the folder that you created.
2. Click **Add > New Model**.

The Source Model window appears.



3. Enter the model's name.
4. Select the model type. You can create a Logical model, a Physical model, or a Logical/Physical model.
5. Select the database and database version for the model. This is available only if you select Physical or Logical/Physical model types.
6. Click **OK**.

The model is added to the project in the folder that you created.

In addition to importing and deriving models, you can also reverse engineer database scripts to add models to projects.

## Reverse Engineering Models

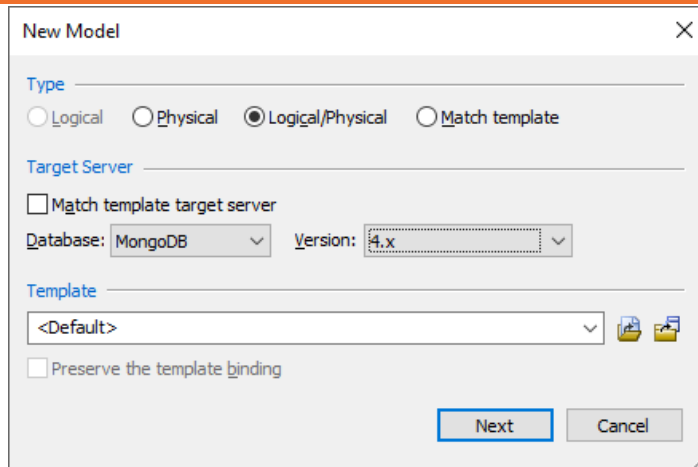
To reverse engineer a model from a database script, follow these steps:

1. Right-click the folder in which you want to add the reverse engineered model.
2. Click **Reverse Engineer**.

The New Model window appears.

## Using Project Explorer

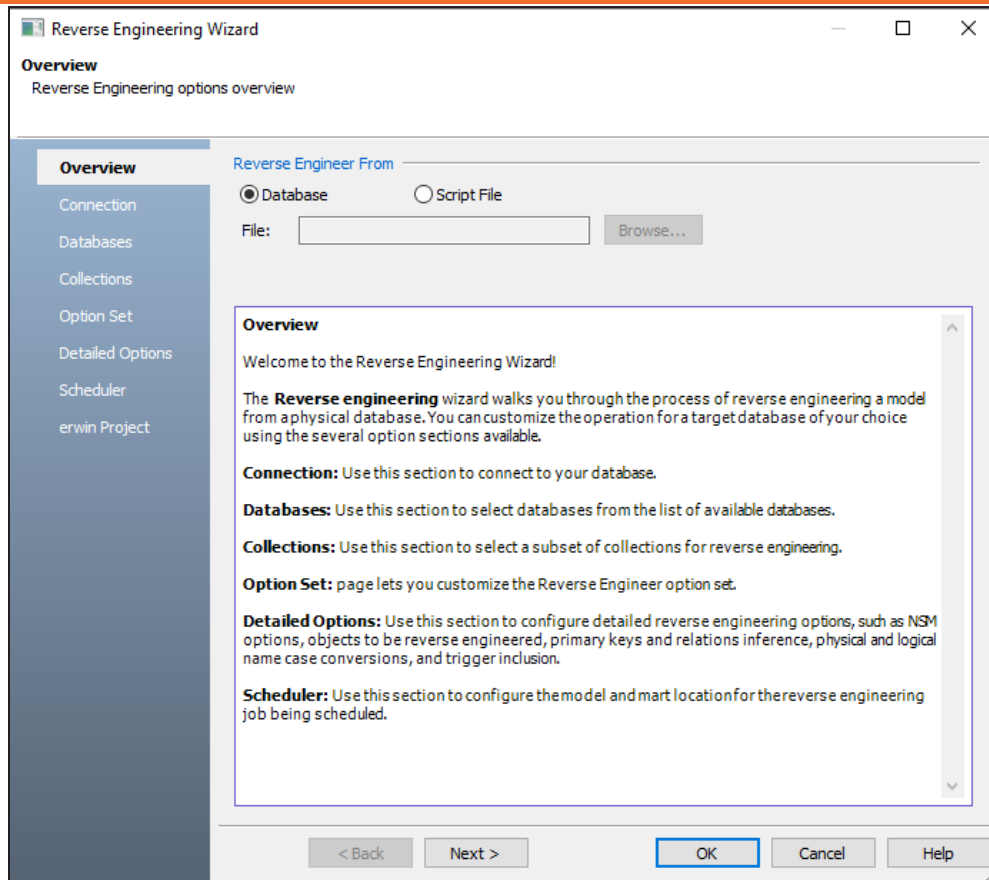
---



3. Select the type of the model. Available options are Physical, Logical/Physical, and Match Template.
4. Select the Target Server database and version.
5. Click **Next**.

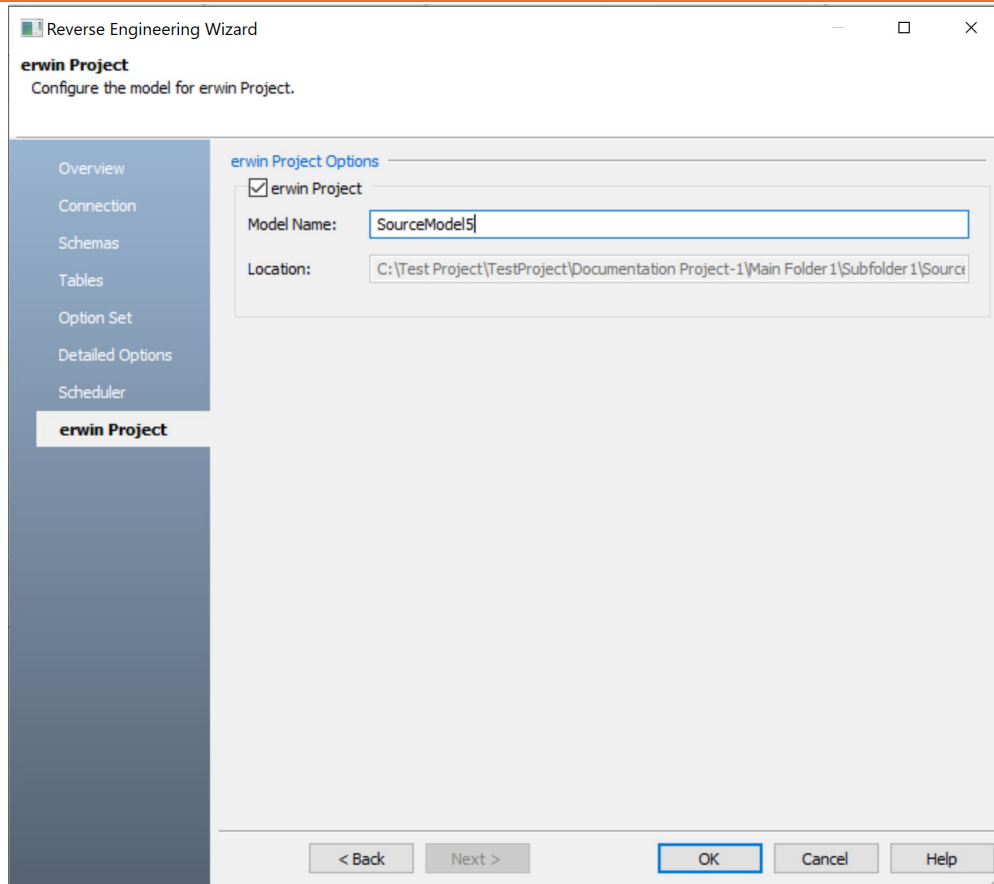
The Reverse Engineering Wizard appears.

## Using Project Explorer



6. On the **Overview** tab, browse and select the script file from your storage.
7. Enter appropriate values in the fields for the other tabs.
8. On the **erwin Project** tab, select **erwin Project** to specify the model belongs to a project.

## Using Project Explorer



9. Enter a name for the model.

10. Click **OK**.

The model is added to the project. You can find the reverse engineered model in the folder you selected.

## Creating Target Models

In addition to deriving models, you can create target models using the following methods:

- Add new model
- Add new model with Auto Sync
- Open model as target



## Using Project Explorer

---

A target model may or may not inherit the objects and attributes of its source model. You can also enable synchronization of a source model with its target, so that every change made to the source is replicated to its targets.

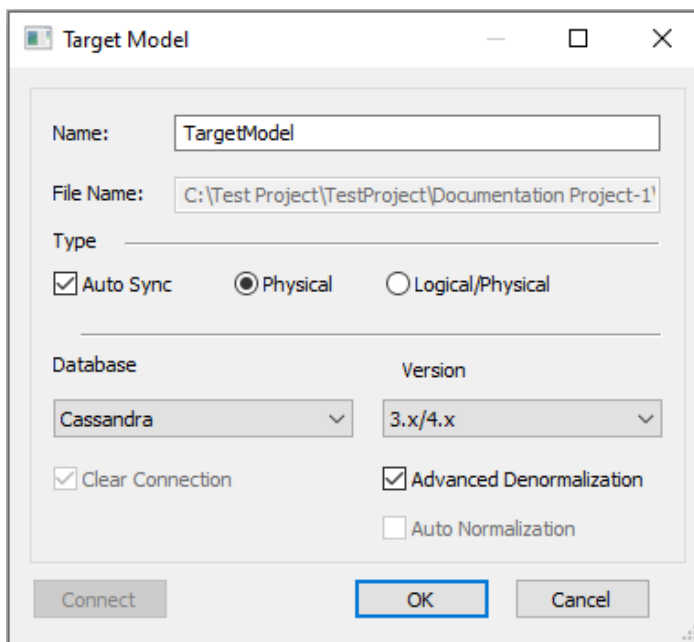
In this example, we will create a target model, TargetModel1, from SourceModel1.

TargetModel1 will be a denormalized NoSQL replica of the SourceModel1.

To create a target model from SourceModel1, follow these steps:

1. Right-click SourceModel1.
2. Click **Add > New Model**.

The Target Model window appears.



3. Enter the name TargetModel1 for the target model.
4. (Optional) Select **Auto Sync** to enable syncing the source model with the target model.
5. Select the required database and version for the target model.
6. (Optional) Select **Advanced Denormalization** and **Auto Normalization**, if required.
7. Click **OK**.

## Using Project Explorer

---

8. If you have selected Advanced Denormalization, the Denormalization wizard appears. You can select the objects such as tables and columns for denormalization and click **OK**.

Once you have completed the denormalization process, the new model is added to the project. You can find TargetModel1 under the source model you selected.

Now that you have the project and the models ready, you can start working on the models as required.

At any point of time, you can perform the following actions on the project, using the Project Explorer:

- Rename an object
- Delete an object
- Save the project
- Forward engineer models in the project
- Generate project report

To read more about managing projects, refer to the [Managing Projects](#) section.

## Managing Projects

You can use the following option in Project Explorer to manage projects.

### Renaming Objects

You can change the name of the project or any object in it.

To rename the model, follow these steps:

1. Right-click a model.
2. Click **Rename**.
3. Enter the new name.
4. Press **Enter**.

The new name is applied to the model.

### Deleting Objects

To delete the model, follow these steps:

1. Right-click the model.
2. Click **Delete**.
3. Click **OK** to confirm.


Alternatively, select the model and click **Delete**  on the Project Explorer.

The selected model is deleted from the project.

Deleting an object is an irreversible action. You cannot retrieve an object once it is deleted from the project.

While working on the project, ensure that you save your changes at regular intervals and before closing erwin DM to avoid losing the progress.

### Saving Project


To save a project, click **Save**  on the Project Explorer. This saves all your progress to the specified project folder.

You can also choose to save the project to erwin Mart. Saving to Mart ensures security of the data and enables team collaboration.

To save a project to Mart, you must first connect to erwin Mart.

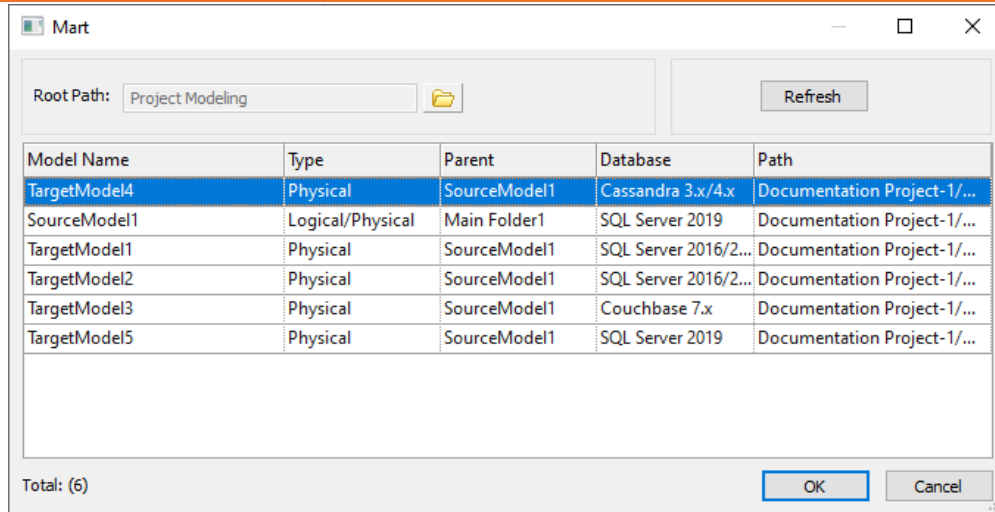
Checked-out mart models in projects are always saved to their original library.

Once you are connected, to save the project to Mart, follow these steps:

1. In the Project Explorer, click **Save to Mart** .

The Mart window appears.

## Using Project Explorer



2. Specify a root path for the project and click **OK**.


The project and models are now saved to Mart. You can access the project or any model in it any time from your mart.


Now that you have added multiple models into the project, you can forward engineer the models and generate schema scripts.


You can generate schema scripts and alter scripts for the models in the project using the Forward Engineering feature in Project Explorer. In this example, we will forward engineer all the models in the project, generate alter scripts, and save the script files in your local folder.

## Forward Engineering Models

To forward engineer models and generate schema and alter scripts, follow these steps:

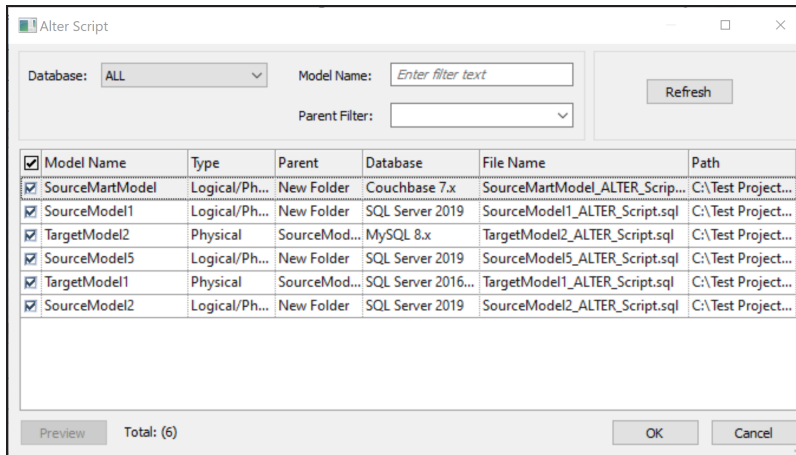
1. To generate a schema script for the models in the project, click **Schema Script**  in Project Explorer. Alternatively, you can right-click the project and click **Forward Engineer > Schema Script**.

To generate a schema script for the models in the project, click **Alter Script**  in Project Explorer. Alternatively, right-click the project and click **Forward Engineer > Alter Script**.

For this example, click **Alter Script**  to generate alter scripts for the models.

## Using Project Explorer

The Alter Script window appears.



2. To filter models from the list, enter appropriate values in the following filters and click **Refresh**.

- Database
- Model Name
- Parent Filter

As we are generating alter scripts for all models in the project, select **All** in the **Database** drop-down, click **Refresh** and select all models.

3. (Optional) To preview the alter script for a model, select the model and click **Preview**.
4. Click **OK**.

Alter scripts are generated for all the models in your project. You can find the alter scripts in the Forward Engineering folder in your local project folder.

Finally, you can generate reports for all or selected models in the project using the Report option in Project Explorer. The project report contains details like model names, model types, parent models, database types, and model file path in your local or online folder.

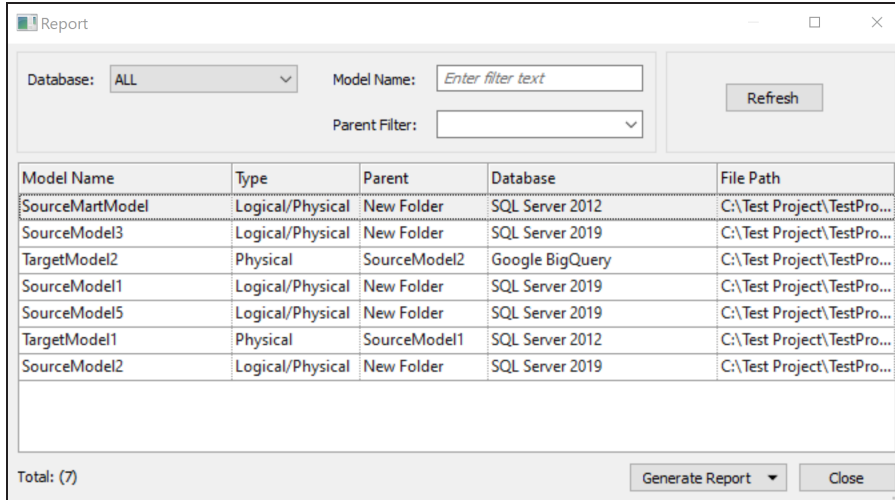
## Generating Project Reports

In this example, we will generate reports of all SQL Server database models in HTML format. To generate project reports, follow these steps:

## Using Project Explorer

1. In the Project Explorer, click **Report** .

The Report window appears.



The Report window displays a table of models and filters. The filters include Database (set to ALL), Model Name (with a placeholder 'Enter filter text'), and Parent Filter. The table lists models with columns for Model Name, Type, Parent, Database, and File Path. At the bottom, there is a 'Total: (7)' label, a 'Generate Report' button with a dropdown arrow, and a 'Close' button.

| Model Name      | Type             | Parent       | Database        | File Path                  |
|-----------------|------------------|--------------|-----------------|----------------------------|
| SourceMartModel | Logical/Physical | New Folder   | SQL Server 2012 | C:\Test Project\TestPro... |
| SourceModel3    | Logical/Physical | New Folder   | SQL Server 2019 | C:\Test Project\TestPro... |
| TargetModel2    | Physical         | SourceModel2 | Google BigQuery | C:\Test Project\TestPro... |
| SourceModel1    | Logical/Physical | New Folder   | SQL Server 2019 | C:\Test Project\TestPro... |
| SourceModel5    | Logical/Physical | New Folder   | SQL Server 2019 | C:\Test Project\TestPro... |
| TargetModel1    | Physical         | SourceModel1 | SQL Server 2012 | C:\Test Project\TestPro... |
| SourceModel2    | Logical/Physical | New Folder   | SQL Server 2019 | C:\Test Project\TestPro... |

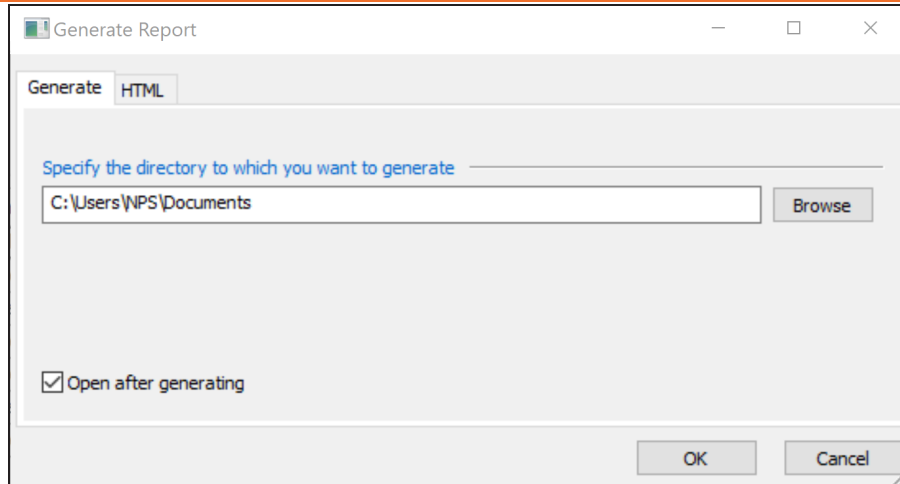
2. To filter models from the list, enter appropriate values in the following filters and click **Refresh**.
  - Database
  - Model Name
  - Parent Filter.

As we are generating reports for all SQL Server database models, select **SQL Server** in the **Database** drop-down, click **Refresh**, and select all available models.

3. Click **Generate Report** and select **HTML** from the drop-down.

The Generate Report window appears.

## Using Project Explorer



4. On the **Generate** tab, specify a location for the project report.
5. (Optional) You can customize the report further using the **HTML** tab.
6. Click **OK**.

Project Report is generated for the selected the models. You can find the report for all SQL Server models in the directory location you specified.

| Project Explorer Report |                 |                  |              |                 |   |
|-------------------------|-----------------|------------------|--------------|-----------------|---|
| Definition :            |                 |                  |              |                 |   |
| Project Explorer Report |                 |                  |              |                 |   |
| #                       | Model Name      | Type             | Parent       | Database        | File Path   |
| 1                       | SourceMartModel | Logical/Physical | New Folder   | SQL Server 2012 | C:\New Project\New Folder\SourceModel4.erwin      |
| 2                       | SourceModel3    | Logical/Physical | New Folder   | SQL Server 2019 | C:\New Project\New Folder\CheckoutTestModel.erwin |
| 3                       | SourceModel1    | Logical/Physical | New Folder   | SQL Server 2019 | C:\New Project\New Folder\SourceModel1.erwin      |
| 4                       | SourceModel5    | Logical/Physical | New Folder   | SQL Server 2019 | C:\New Project\New Folder\SourceModel5.erwin      |
| 5                       | TargetModel1    | Physical         | SourceModel1 | SQL Server 2012 | C:\New Project\New Folder\TargetModel1.erwin      |
| 6                       | SourceModel2    | Logical/Physical | New Folder   | SQL Server 2019 | C:\New Project\New Folder\SourceModel2.erwin      |
| Total : 6               |                 |                  |              |                 |   |

## Productivity and UI Enhancements

Several additions and enhancements have been implemented to improve erwin Data Modeler's (DM) productivity and usage experience. These enhancements are:

- [Copy Family](#)
- [Demand Loading](#)
- **Scheduler Update**
  - Support for projects has been added to erwin DM Scheduler. You can now schedule reverse engineering jobs for all the required models in a project using erwin DM Scheduler.
- **Project Explorer**
  - Project Explorer now displays models with configured connection parameters for reverse engineering with a green check mark.
- **Custom Option Set**
  - You can now create and use custom option sets for Complete Compare in erwin DM Scheduler.
- **Recent Files Update**
  - The Recent Files list in the erwin DM Welcome screen now displays the latest version of the Mart models instead of the version that was last accessed.
- **Exclude UDP**
  - You can now choose to exclude User Defined Properties (UDP) during Complete Compare to accelerate the process.
- **SDI Indication on Model Diagram**
  - For columns or attributes labelled SDI, you can now choose to display an SDI indication on the model diagram via diagram properties. For more information, refer to the [Set Table Display Preferences](#) and [Set Entity Display Preferences](#) topics.



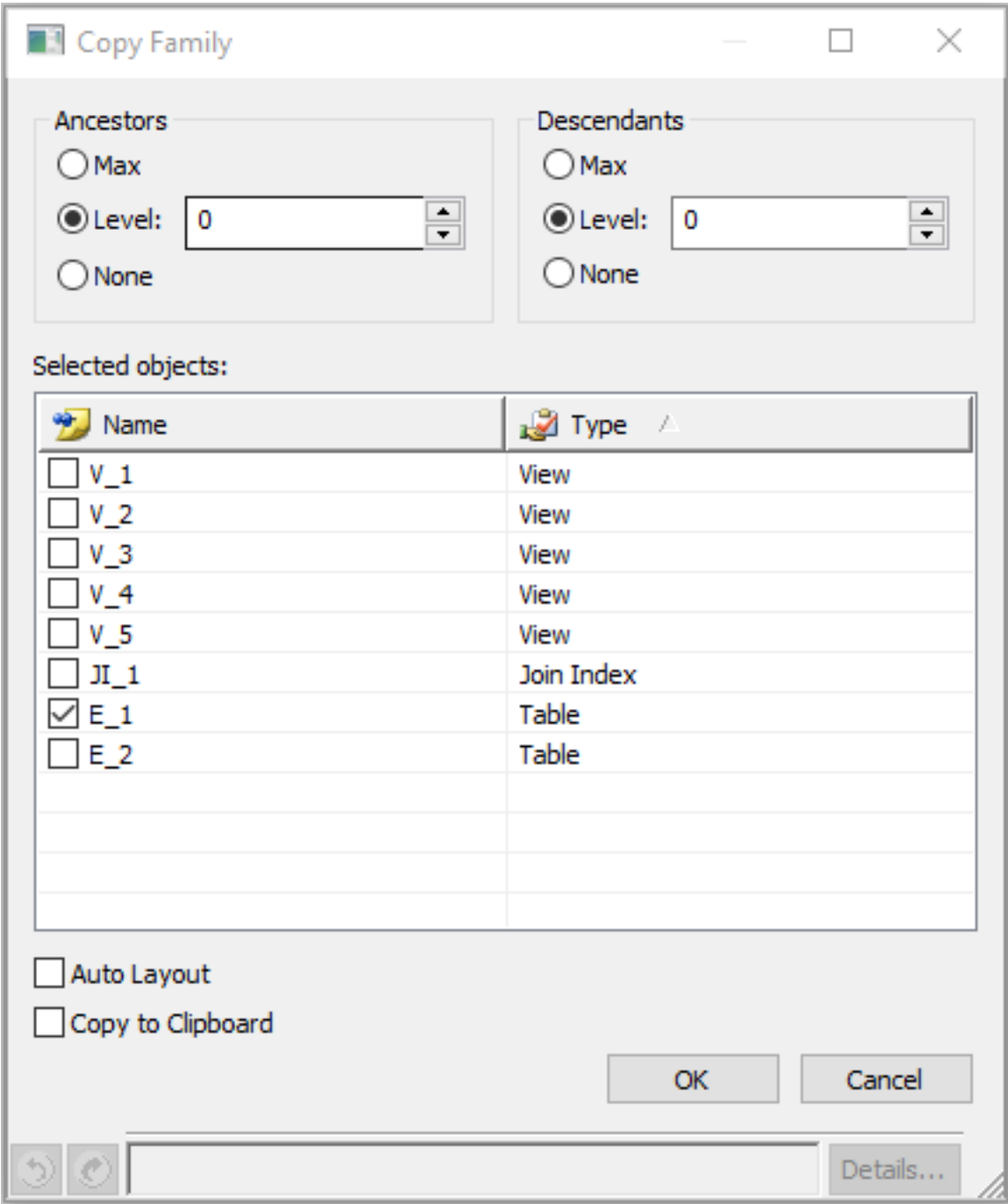
## Copy Family

You can create new Entity Relationship (ER) diagrams from parts of existing model object families using the Copy Family feature. You can select the number of levels of objects in the family to be copied, and create a new diagram with the new family or paste the selected family to another existing ER diagram.

### To create a new diagram from an existing family

1. Open a diagram and select an object that belongs to the required object family.
2. On the **Home** ribbon, click **Copy Family**.

The Copy Family window appears.



## Productivity and UI Enhancements

---

3. Specify the required number of levels of ancestors and descendants in the family with respect to the selected object.
4. Select **Auto Layout**, if you want erwin DM to automatically generate a layout for the family in the new diagram.
5. Select **Copy to Clipboard**, if you want to keep the object family in the clipboard.
6. Click **OK**.

A new ER diagram is created with the selected objects in the family.

If you have selected Copy to Clipboard, you can also paste the selected objects into any existing or new ER diagram.

## Demand Loading

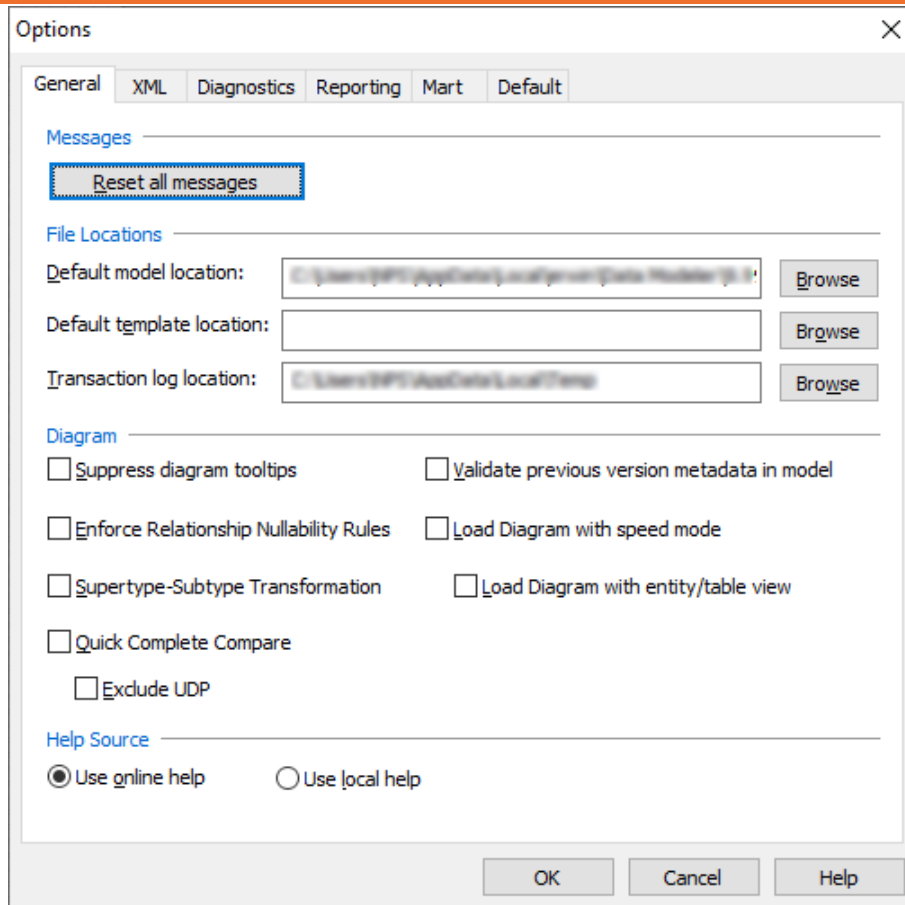
Demand Loading reduces the time required to open projects. When you open a project, erwin DM loads all the models in it, which may increase the loading time, whereas Demand Loading opens only the project metadata and leaves the models in it unloaded. You can load required models manually.

For example, if a project contains multiple large models, opening them simultaneously will require more time. If you want to open a single model in the project and work on it, you can select Demand Loading and open the project. This instantly opens the project metadata in the Project Explorer. You can then select the required model to open it.

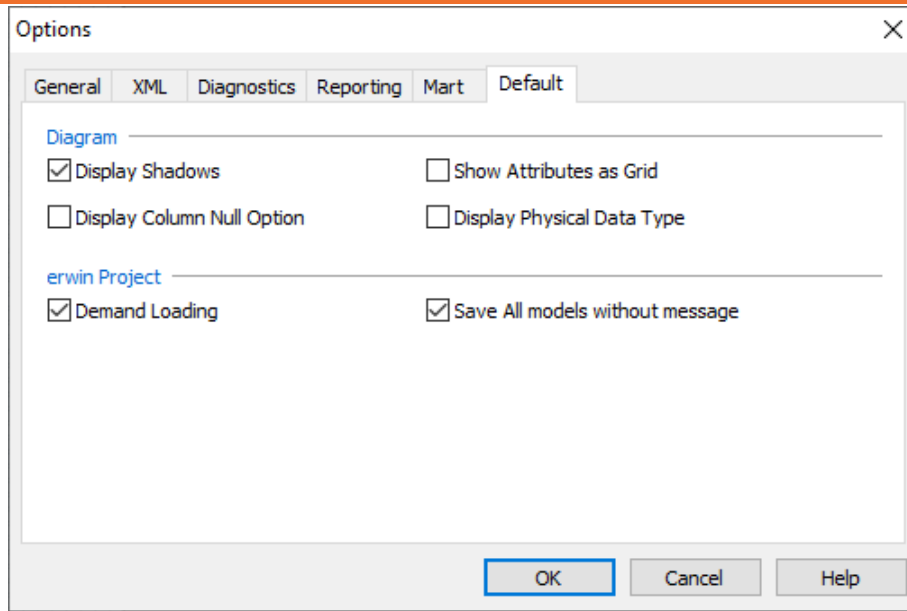
To use Demand Loading, follow these steps:

1. Click **Tools > Options**.

The Options screen appears.



2. On the Default tab, select **Demand Loading**.



3. Click **OK**.

Demand Loading is enabled. erwin DM will load only the metadata when you open a new project.

## erwin ER360

erwin ER360 provides unified and central visuals of your erwin data models to help you manage, review, and govern your on-premise or SaaS 'On Cloud Mart' data models. It provides self-service access to visualization and detailed metadata classification to encourage business teams, governance teams, and other stakeholders to collaborate and increase data literacy. For more information, refer to [erwin ER360 Online Help](#).

## DM Connect for DI

The DM Connect for DI feature has been upgraded to support:

- [Importing one or more erwin Data Intelligence \(erwin DI\) environments](#) to erwin Data Modeler (erwin DM)
- [Multiple erwin DI configurations](#)
- erwin DI v13.0

## erwin Mart Portal Enhancements

erwin Mart Portal has undergone several enhancements as follows:

- Email addresses or user name, individually as well as together are supported as your default selector instead of only username. You can configure this under settings to be displayed across erwin Mart Portal and erwin ER360.
- The catalog tree displays a Sensitive Data Indicator, SDI tag for models that contain sensitive data.
- Ability to control the level of permission and assign a user or a group to a profile, library, or model has been added.
- View as a permission at library level has been added.
- Using the Project Explorer in erwin DM, you can save projects and their models to mart under catalogs.
- You can generate Personal Access Tokens for REST API via Settings.
- User device tracking for logs is supported.

## Enterprise Modeling Compliance

A new module, Enterprise Modeling Compliance has been added in erwin Mart Portal. Enterprise Modeling Compliance (EMC) provides you (non-admin user) with a mechanism to apply metadata standards across your mart catalogs and maintain data model quality at the enterprise level.

## DM Connect for DI

---

To use EMC, you need to be well-versed with business requirements so that you can create and use a combination of policies and rules on your data models. This enables you to implement high compliance standards in your modeling practice.

Rules define a set of protocols to be followed in an organization for model compliance. You can create rules or use default rules to be followed throughout the model and populates errors or warnings if not followed.

Now, these rules are applied to the default policies or user-defined policies, which you can create. Policies are globally defined set of principles. Using policies, you can maintain business standards across your organization. Policies are applied to the objects such as attributes, columns, entities, tables, domains, and relationships, which are available in the models.

Based on the policy applied to the model, job can be executed and generate comprehensive reports with pass, error, and warning statistics. If the job result got failed, you can go to the model and improve the data quality for the objects to pass the job result.

The following sections walk you through these features:

- [Policies](#)
- [Rules](#)
- [Reports](#)

## Policies

Policies are globally defined set of principles. Using policies, you can maintain business standards across your organization. It includes the rules applied to objects, passing criteria with errors and warnings. For example, the below policy, Definition, a default policy, includes rules applied to attribute, domain, entity, and relationship. Here, pass criteria is 10% error and 15% warning.

## DM Connect for DI

Policies Rules Generate Report View Reports

Comment  
Comments 2  
**Definition**  
Illustration  
Logical Name  
Physical Name  
Quest

**Details**

Name  
Definition

Description  
Validate Definition rules

Pass Criteria Errors (10%)  
Warnings (15%)

**Rules**

Select Object Type  
Attribute Entity Domain Relationship

Select Object Type

☐ Comment should be more than 5 words  
☒ Definition does not contain bad characters  
☒ Definition is populated  
☒ Definition less than 255  
☒ Logical and physical definitions are the same  
☐ Logical Name does not contain bad characters

## Rules

Rules define a set of protocols to be followed in an organization. You can create rules for default and user defined policies. It includes validation action with the value to validate. For example, the below rule, Definition is populated, is applied to definition. This means, If definitions available in models do not populate, the errors are generated.

Policies Rules Generate Report View Reports

Comment does not contain bad characters  
Comment is populated  
Comment must not have star  
Comment should be more than 5 words  
Definition does not contain bad characters  
**Definition is populated**  
Definition less than 255  
Logical and physical definitions are the same  
Logical Name does not contain bad characters  
Logical Name does not contain underscore  
Logical Name is not same as Definition  
Logical Name is not same as Physical Name

**Details**

Name  
Definition is populated

Description  
Verify object definition is populated

Select Property/Object  
Definition

Select Validate Action  
is populated

Value  
N/A

Warning ☒ Error



## Reports

When you run the job, policy is applied to the catalog and based on applied policies and rules, comprehensive reports are generated with pass, error, and warning statistics. For example, the below report, Comments for GBQ, has passed with 59% having 41% errors, 0% warnings. You can see catalog name is Google BigQuery, object type is entity/table and rules applied to the catalog are, Comment does not contain bad characters and Comment is populated.

| Policies Rules Generate Report View Reports |                                |      |                                      |  |
|---|--------------------------------|------|--------------------------------------|--|
| <input type="checkbox"/>                    | Amazon Keyspaces               | Fail | Pass: 69%, Errors: 31%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Amazon Keyspaces_1666183066624 | Fail | Pass: 69%, Errors: 31%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Amazon monthly                 | Fail | Pass: 75%, Errors: 25%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Amazon monthly_1666845515894   | Fail | Pass: 75%, Errors: 25%, Warnings: 0% |  |
| <input checked="" type="checkbox"/>         | Comments for GBQ               | Pass | Pass: 59%, Errors: 41%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Comments for GBQ_1667470340128 | Pass | Pass: 59%, Errors: 41%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Keyspaces                      | Pass | Pass: 0%, Errors: 100%, Warnings: 0% |  |
| <input type="checkbox"/>                    | Illustration_1668405521609     |      | 0%                                   |  |

| Details         |              |        |   |                      |
|-----------------|--------------|--------|---|----------------------|
| Catalog         | Object Type  | Name   | Comment does not contain bad characters | Comment is populated |
| Google BigQuery | Entity/Table | Tab4   | Pass                                    | Error                |
| Google BigQuery | Entity/Table | Tab5   | Pass                                    | Error                |
| Google BigQuery | Entity/Table | Tab6   | Pass                                    | Error                |
| Google BigQuery | Entity/Table | Tab7   | Pass                                    | Error                |
| Google BigQuery | Entity/Table | Tab8   | Pass                                    | Error                |
| Google BigQuery | Entity/Table | Tab10  | Pass                                    | Pass                 |
| Google BigQuery | Entity/Table | Tab11_ | Pass                                    | Pass                 |
| Google BigQuery | Entity/Table | _Tab12 | Pass                                    | Pass                 |
| Google BigQuery | Entity/Table | _Tab12 | Pass                                    | Pass                 |
| Google BigQuery | Entity/Table | Tab13  | Pass                                    | Pass                 |

To create polices, rules, and reports, refer [Using Enterprise Modeling Compliance](#) topic.

## Using Enterprise Modeling Compliance

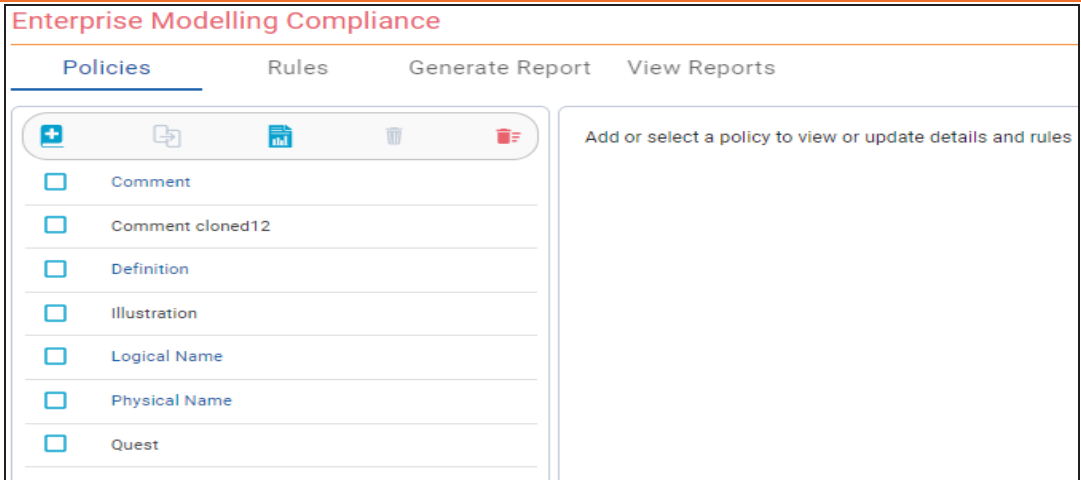
This topic explains how you can use polices, rules, and reports for model compliance.

Let's create a rule, Physical name does not contain star. This rule is applicable to each physical names of the catalog.

To create a rule, follow these steps:

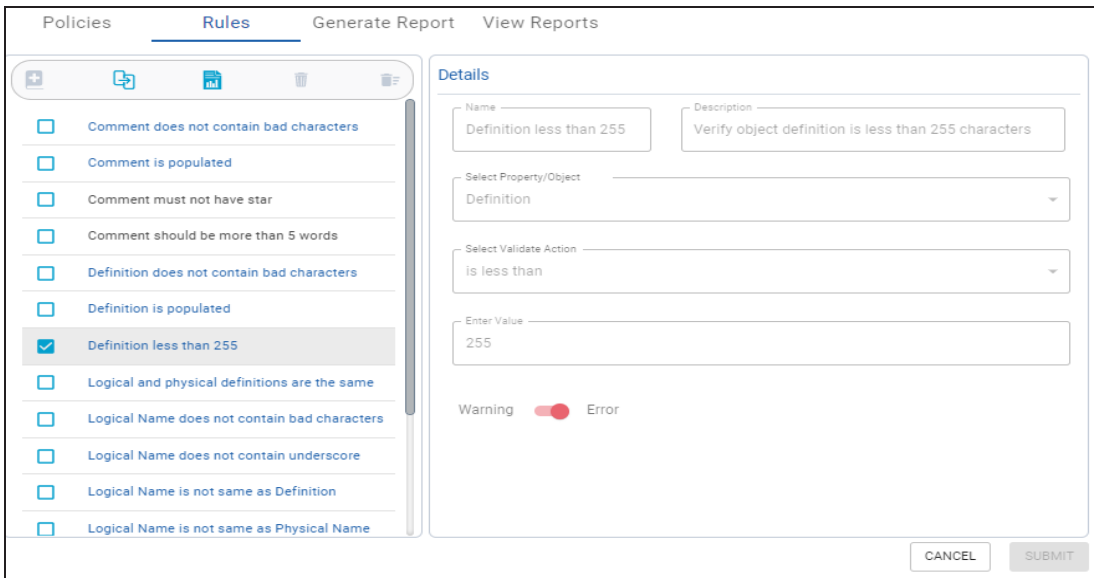
1. Go to **Application Menu > EMC**.

The Enterprise Modeling Compliance page appears.



2. On the **Rules** tab, click .

The Details section appears.



3. Enter appropriate values in the fields. Refer to the following table for field descriptions.

| Field                  | Description  |
|------------------------|--|
| Name                   | Specifies the name of the rule   |
| Description            | Specifies the description and purpose of the rule  |
| Select Property/Object | Specifies the property or object to which the rule is applicable                                       |
| Select Validate Action | Specifies the condition for validation. Based on the selected object type, available options may vary. |
| Enter Value            | Specifies the value for validation   |
| Warning/Error          | Specifies whether failing the rule populates results in a warning or error                             |

4. Click **Submit**.

The rule, Physical name does not contain star, is created and added to the list of rules. In the list of rules, user-defined rules are displayed with the black font.

The screenshot shows the DM Connect for DI interface. On the left, a list of rules is displayed, with 'Comment does not contain star' selected. On the right, the 'Details' panel for this rule is shown. The 'Name' field contains 'Comment does not contain star' and the 'Description' field contains 'Physical names cannot have stars'. The 'Select Property/Object' dropdown is set to 'Physical Name'. The 'Select Validate Action' dropdown is set to 'does not contain'. The 'Enter Value' field contains '\*'. At the bottom, the 'Warning/Error' toggle is set to 'Warning'.

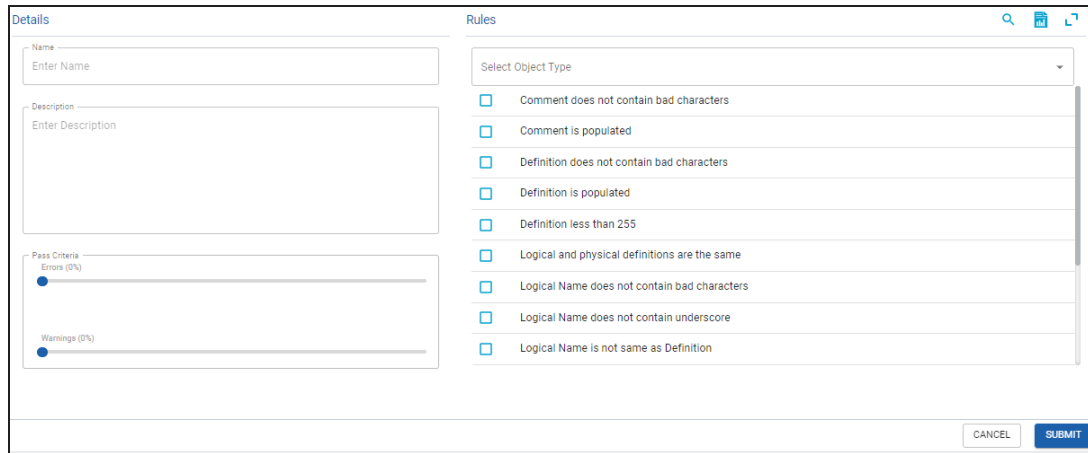
Once the rule is created, you can apply this rule to the objects or properties while creating policies.

Now, let's create a policy, Physical name without star, having the above rule and apply it to tables, domains, and relationships of the model.

To create a policy, follow these steps:

1. On the Policies tab, click .

The Details and Rules section appears.



The screenshot shows a form with two main sections: 'Details' and 'Rules'. The 'Details' section on the left contains four fields: 'Name' (with placeholder 'Enter Name'), 'Description' (with placeholder 'Enter Description'), 'Pass Criteria' (with 'Errors (0%)' and 'Warnings (0%)' progress bars), and 'Select Object Type' (a dropdown menu). The 'Rules' section on the right lists ten rules, each with a checkbox: 'Comment does not contain bad characters', 'Comment is populated', 'Definition does not contain bad characters', 'Definition is populated', 'Definition less than 255', 'Logical and physical definitions are the same', 'Logical Name does not contain bad characters', 'Logical Name does not contain underscore', 'Logical Name is not same as Definition', and 'Logical Name is not same as Definition'. At the bottom right, there are 'CANCEL' and 'SUBMIT' buttons.

2. Enter appropriate values in the fields. Refer to the following table for field descriptions.

| Field              | Description   |
|--------------------|---|
| Name               | Specifies the name of the policy  |
| Description        | Specifies the description and purpose of the policy   |
| Pass Criteria      | Specifies the pass criteria in percentage for errors and warnings   |
| Select Object Type | Specifies the object types from the drop-down list. Based on the selected object types, available rules appear. Select the required rules to apply to the policy. |

3. Click **Submit**.

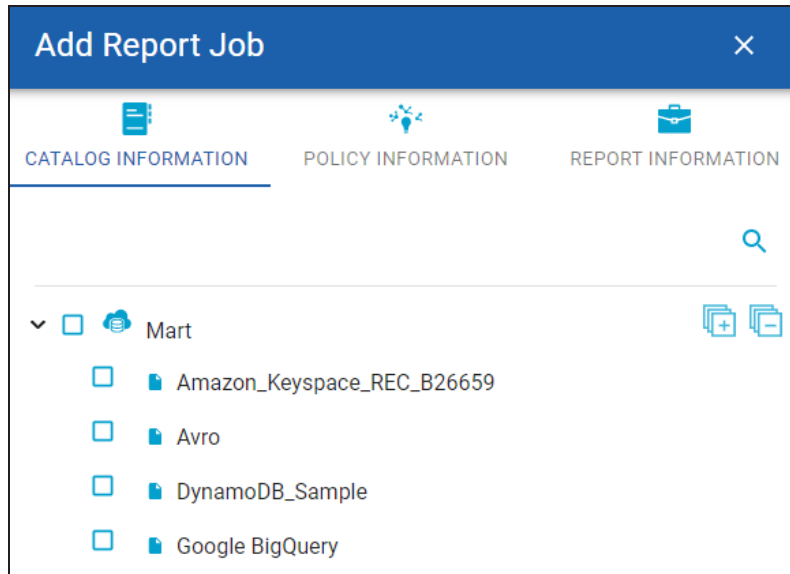
The policy, Physical Name without Star, is created and added to the list of policies. In the list of policies, user-defined policies are displayed with the black font.

Now, you can apply this policy to the model to check the data quality. Also, you can use default rules and policies for model compliance.

To apply a policy, Physical Name without Star, to the Model- Avro, do the following:

1. On the **Generate Reports** tab, click .

The Add Report Job window opens.



2. On the Catalog Information tab, select the catalog- Avro, for which you want to add the job, and click **Next**.

The Policy Information tab opens.

**Add Report Job** ×

CATALOG INFORMATION    **POLICY INFORMATION**    REPORT INFORMATION

Select Policy ▼

< PREVIOUS    NEXT >

3. Select the policy- Physical Name without Star, which you want to apply for the job, and click **Next**.

The Report Information tab opens.

The screenshot shows the 'Add Report Job' form with the 'REPORT INFORMATION' tab selected. The form includes the following fields and controls:

- Report Name:** A text input field with the placeholder 'Enter Report Name'.
- Scheduled On:** A date and time picker showing '2022/10/27 09:55 AM' with a calendar icon.
- Job Interval:** A dropdown menu with a red border, currently showing 'Job Interval'.
- Notify Me:** A checkbox that is currently unchecked.
- Notification Email:** A text input field.
- CC List:** A text area for additional email addresses.
- Run Now:** A checkbox that is currently unchecked.
- Navigation:** At the bottom right, there is a '< PREVIOUS' button and a 'SUBMIT' button.

4. On the Report Information tab, enter the following details:

**Report Name**

Specifies a name for the report.

**Scheduled On**

Specifies the date and time at which the job must run.

**Job Interval**

## DM Connect for DI

Specifies the required frequency at which the job must run. You can set the job to run once or recur daily, weekly, monthly, or yearly.

### Notify Me

Specifies whether a notification is sent to you when the job status changes.

This enables the Notification Email and CC List fields.

### Notification Email

Specifies the email address at which you want to receive the notification.

### CC List

Specifies a list of email addresses, separated by a semi-colon, that must receive the job notification

### Run Now

Specifies whether to run the job immediately.

#### 5. Click **Submit**.

The job is added to the calendar for execution with its Job State set to Scheduled.

Now, you can view the job report. It shows name, overall results along with the percentage of pass, error, and warnings.

#### 1. Click **View Reports**.

+

📄

↺

🗑️

📄

| Name   | Result | Statistics                                 |
|--|--------|--|
| <input type="checkbox"/> Amazon Keyspaces                  | Fail   | Pass: 69%,<br>Errors: 31%,<br>Warnings: 0% |
| <input type="checkbox"/> Amazon<br>Keyspaces_1666183066624 | Fail   | Pass: 69%,<br>Errors: 31%,<br>Warnings: 0% |
| <input type="checkbox"/> Amazon<br>monthly_1666845515894   | Fail   | Pass: 75%,<br>Errors: 25%,<br>Warnings: 0% |
| <input type="checkbox"/> Keyspaces Illustration            | Fail   | Pass: 0%, Errors:<br>100%, Warnings:<br>0% |

Details

Select a valid report to view report details

#### 2. Select a report to view detailed information.



## DM Connect for DI

| Details                    |              |  |  |   |  |  |
|----------------------------|--------------|--|--|---|--|--|
| Catalog                    | Object Type  | Name   | Logical Name does not contain underscore | Logical Name is not same as Physical Name | Logical Name does not contain bad characters | Logical Name is not same as Definition |
| Amazon_Keyspace_REC_B26659 | Entity/Table | all_data_types                                   | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | t_0123456789123456789_abcdefghijklmnopqrstuvwxyz | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | table123   | Pass                                     | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | tbl_48_char_123456789_abcdefghijklmnopqrstuvwxyz | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | cyclingtab                                       | Pass                                     | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | test_999   | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | my_table   | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | my_table2  | Error                                    | Error                                     | Pass   | Pass                                   |
| Amazon_Keyspace_REC_B26659 | Entity/Table | employee_information                             | Error                                    | Error                                     | Pass   | Pass                                   |

If the errors are more than 20% and warnings are more than 10%, you can go to the model and remove the stars (\*) for model compliance and increase the data quality.

## Enterprise Glossary

A new module, Enterprise Glossary has been added to erwin Mart Portal. Enterprise glossary provides you with a mechanism to implement business terms, vocabularies, ontologies, and standards. Thus enabling you to build and govern an enterprise-level standard for your mart catalog assets.

Using Enterprise Glossary, you can create and associate your business terms and definitions used in different areas of your business departments or domains.

Here, enterprise glossaries work as containers for business terms. Therefore, before creating business terms, you need to create a glossary for them. You can also create sub-glossaries to group business terms further.

Once business terms are added to the glossary, it is then associated to the catalogs.

After association, you can map business terms to the objects available in the catalogs and generate association and mapping reports. Glossary Associations Report includes details such as catalog name and number of unmapped objects. A Mapping report displays details such as catalog name, object type, object name, and mapped term.

To create glossaries, business terms, associations, mappings, and reports, refer to the [Using Enterprise Glossary](#) topic.

## Using Enterprise Glossary

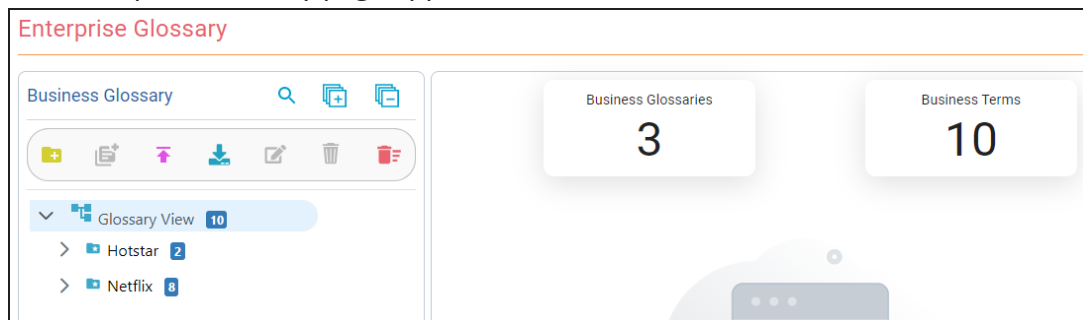
This topic explains how you can create and use glossaries, business terms, associations, mappings, and reports.

For example, assume you have a project called YouTube. Start by creating a glossary called YouTube to contain all the business terms used in the YouTube project.

To create a glossary, follow these steps:

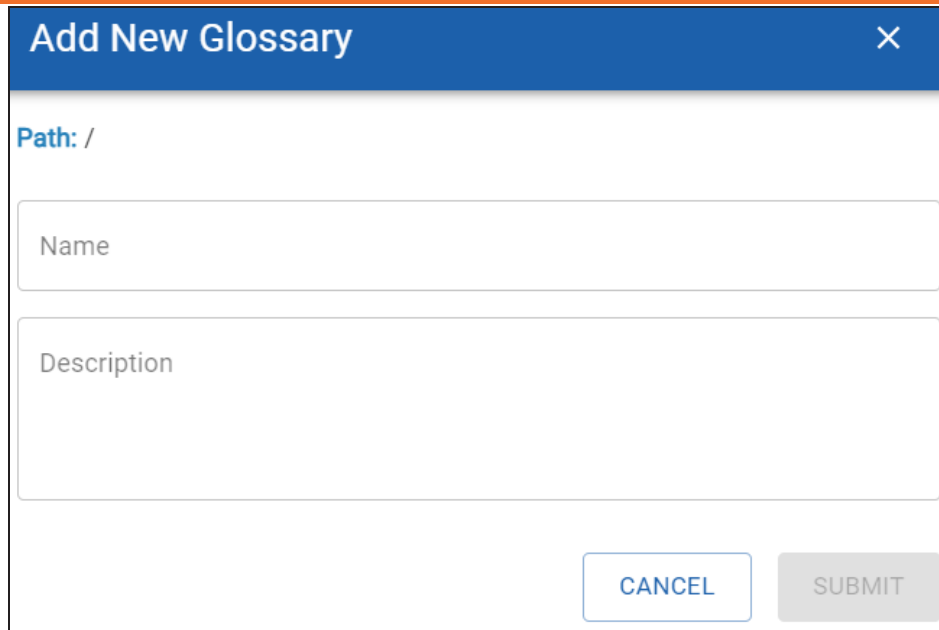
1. Go to **Application Menu > Enterprise Glossary**.

The Enterprise Glossary page appears.



2. Select the Glossary View node and click .

The Add New Glossary page appears.

A dialog box titled "Add New Glossary" with a close button (X) in the top right corner. Below the title bar, there is a "Path: /" label. The main area contains two text input fields: "Name" and "Description". At the bottom right, there are two buttons: "CANCEL" and "SUBMIT".

**Add New Glossary** ×

Path: /

Name

Description

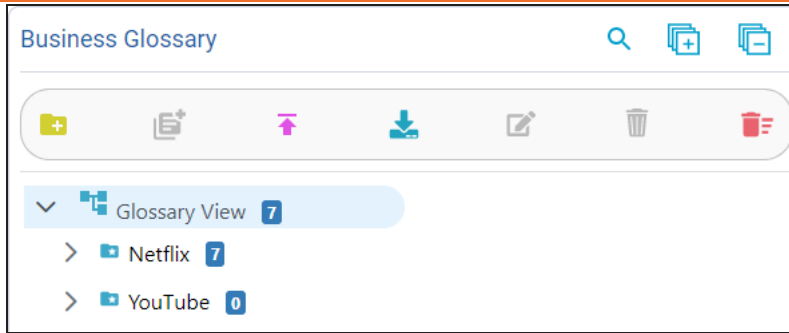
CANCEL SUBMIT

3. Enter appropriate values in the fields. Refer to the following table for field descriptions.

| Field       | Description   |
|-------------|---|
| Name        | Specifies the name of the glossary<br>For example, YouTube.   |
| Description | Specifies the description and purpose of the glossary<br>For example, YouTube contains information such as Users, Account details, Videos, Playlists. |

4. Click **Submit**.


The YouTube glossary is created and added to the list of glossaries.



Once the glossary is created, you can use this glossary as a container and add all business terms which are used in YouTube project.

Now, let's create a business term, Videos, in the YouTube glossary.

To create a business term, follow these steps:

1. Select the glossary, YouTube.
2. Click .

The Add New Business Terms page appears. You can see YouTube as the path because the new business term is being created in the glossary, YouTube.

**Add New Business Term** X

Path: YouTube

Business Term

Definition

Description

☐ SDI

CANCEL SUBMIT

3. Enter appropriate values in the fields. Refer to the following table for field descriptions.



| Field                 | Description  |
|-----------------------|--|
| Business Term         | Specifies the name of the business term.<br>For example, Videos.   |
| Definition            | Specifies the definition of the business term.<br>For example: Videos available in the YouTube.  |
| Description           | Specifies the description about the business term.<br>For example: Videos is used for posting, Adds, movies, songs available in YouTube. |
| Sensitive Data Indic- | Specifies whether the business term is sensitive.  |

## DM Connect for DI

|                    |  |
|--------------------|--|
| ator (SDI)         |  |
| SDI Classification | Specifies the sensitivity classification of the business term.<br><br>This option is available only when SDI is selected.  |
| SDI Description    | Specifies the description about the sensitivity classification.<br><br>This option is available only when SDI is selected. |

### 4. Click **Submit**.

The business term, Videos, is created and added to the list of terms.

| TERMS                           |                                 |  |                                     |                       |         |   |
|---------------------------------|---------------------------------|--|-------------------------------------|-----------------------|---------|---|
| ASSOCIATIONS                    |                                 |  |                                     |                       |         |   |
| MAPPINGS                        |                                 |  |                                     |                       |         |   |
| REPORTS                         |                                 |  |                                     |                       |         |   |
| Definition, Desc...             |                                 |  |                                     |                       |         |   |
| <input type="checkbox"/> Name   | Definition                      | Description                                      | SDI                                 | SDI Classification    | Path    | Actions   |
| <input type="checkbox"/> Videos | Videos available in the YouTube | Videos is used for posting, Adds, movies, son... | <input checked="" type="checkbox"/> | Classification Public | YouTube |   |

Now, the glossary, YouTube, with one business term, Videos, is ready. Let's associate YouTube glossary to an existing catalog, DynamoDB.

1. Select the glossary, YouTube.
2. On the Associations tab, select the DynamoDB catalog.

TERMS

ASSOCIATIONS

MAPPINGS

REPORTS

Catalogs

3. Click .

The YouTube glossary is now associated to the DynamoDB catalog.

After the association of YouTube to DynamoDB, you can map the business term, Videos to the objects available in the DynamoDB catalog.

Let's map the term, Videos, to the table- MusicCollection, available in the DynamoDB model.

1. On the **Mappings** tab, select a model, DynamoDB, to which you want to map business terms of the YouTube glossary.

The Model properties appears.

2. Click .

3. In the Glossary Terms drop down, select the business term, Videos.



The screenshot shows a table with two columns: 'Table/Entity' and 'Glossary Terms'. The 'Table/Entity' column contains the text 'MusicCollection'. The 'Glossary Terms' column contains a dropdown menu with 'videos' selected. To the right of the dropdown is a blue button with a white 'x' icon. To the far right of the table is a blue button with a white checkmark icon and a white 'x' icon.

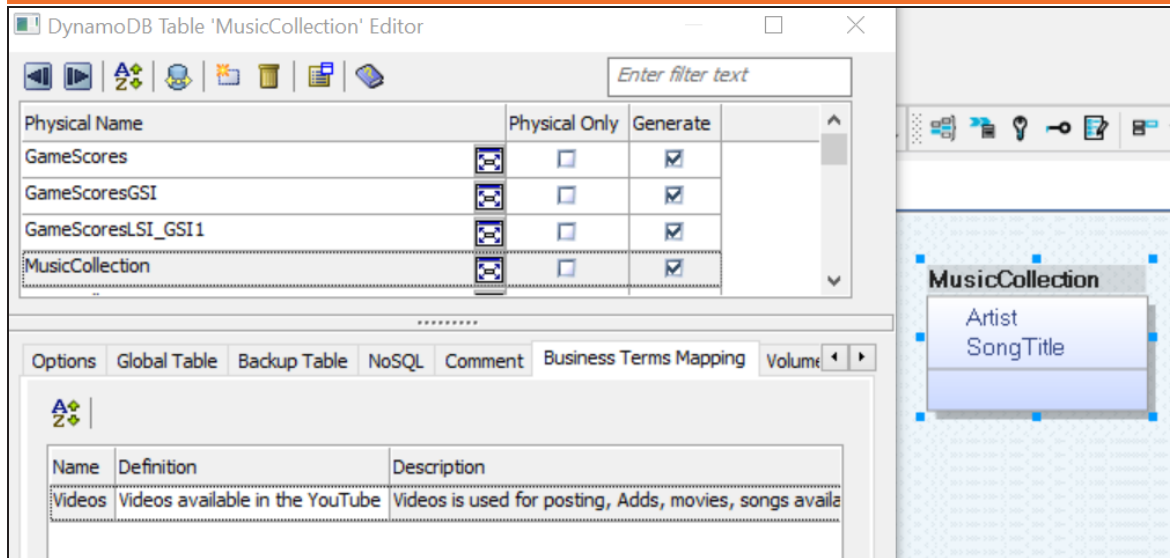
4. Click **Save**.

The business term, Videos is mapped to the table, MusicCollection.

Now, on the **Reports** tab, you can view associations and mappings done on the DynamoDB model.

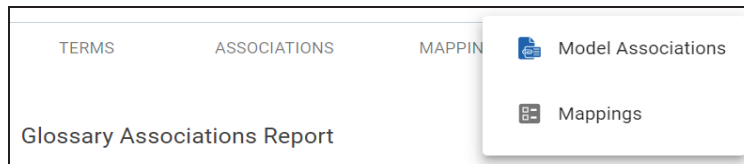
You can also see the mapping you have created in erwin Data Modeler. Go to <procedure> and you can find the mapping of the object, MusicCollection to the business term, Videos in DynamoDB.

## DM Connect for DI



To view associations reports, follow these steps:

1. Click the **Reports** tab.



2. Click **Associations**.

TERMS ASSOCIATIONS MAPPINGS REPORTS

Glossary Associations Report

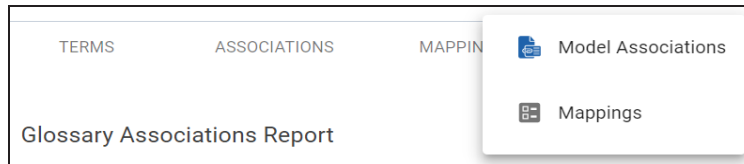
| Name                     | Not Mapped Objects | Modified By | Modified On         |
|--------------------------|--------------------|-------------|---------------------|
| DynamoDB (Mart/DynamoDB) | 78                 | George      | 2022-11-30 08:37:16 |

The Glossary Associations Report appears. It displays details such as model names and number of unmapped objects. Here, the report indicates that there are 78 unmapped objects in the selected model, DynamoDB.

To view mappings reports, follow these steps:



1. Click the **Reports** tab.



2. Click **Mappings**.
3. Select DynamoDB model.

The Business Term Mappings report appears. From the report, you can see that the object, MusicCollection, is mapped to the business term, Videos.

| Business Term Mappings   |                 |                  |             |
|--------------------------|-----------------|------------------|-------------|
| Catalog Name             | Object Name     | Object Type      | Mapped Term |
| DynamoDB (Mart/DynamoDB) | MusicCollection | Table/Entity     | videos      |
| DynamoDB (Mart/DynamoDB) | MusicPut        | Table/Entity     | videos      |
| DynamoDB (Mart/DynamoDB) | SongTitle       | Column/Attribute | videos      |

## Setting up Unity Catalogs in Databricks

You can set up unity catalogs in Databricks and reverse engineer them into erwin Data Modeler.

This process involves the following steps:

- [Creating User Cluster](#)
- [Creating Access Token](#)
- [Accessing the JDBC String](#)

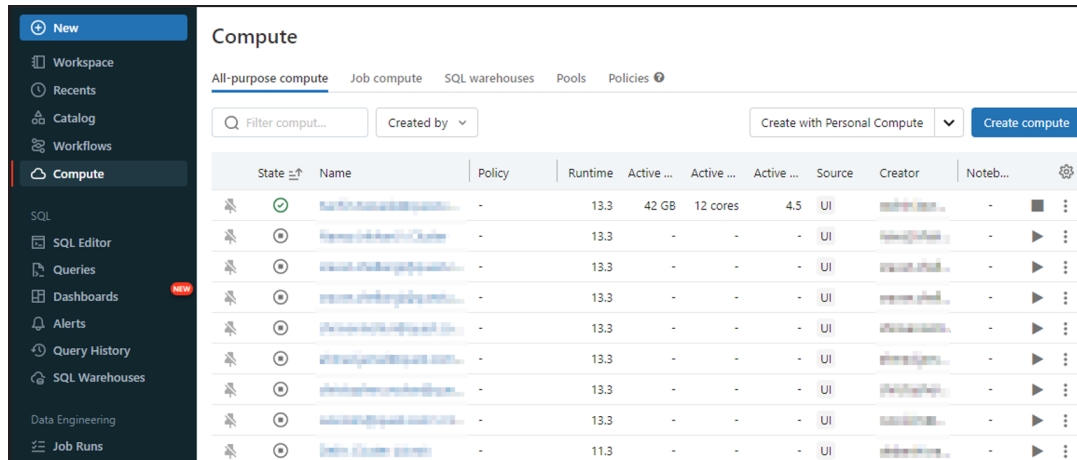
### Creating User Cluster

Creating user cluster enables you to access unity catalogs in Databricks.

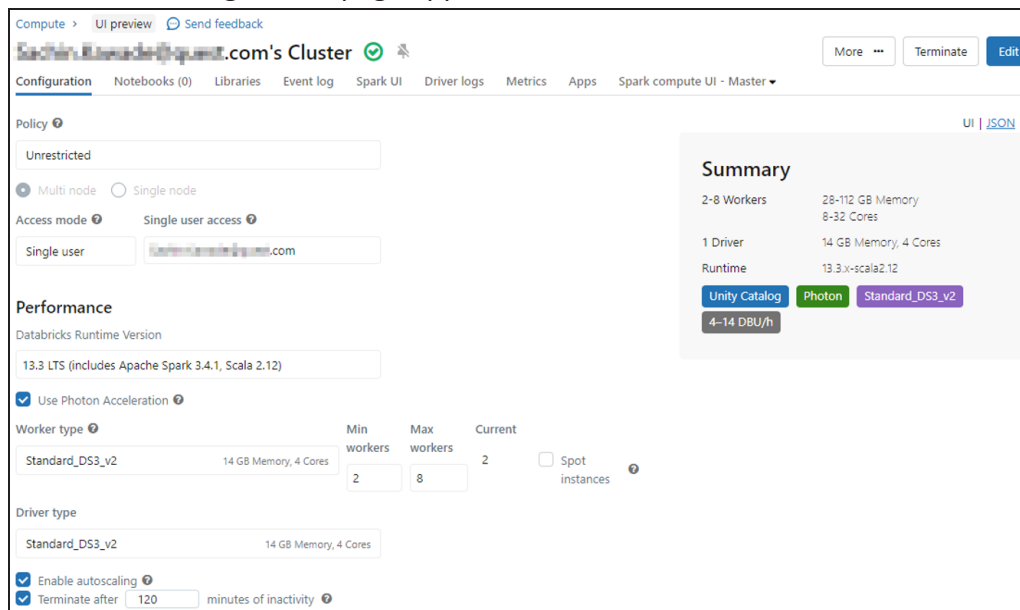
To create a user cluster, follow these steps:

## DM Connect for DI

1. Log in to the Microsoft Azure Databricks portal using your Databricks account.
2. On the left side navigation panel, go to **Compute** tab, and click **Create compute**.



The cluster configuration page appears.



3. In the configuration page, configure the parameters based on the following table.

| Fields      | Parameter   | Additional Information                             |
|-------------|-------------|--|
| Access mode | Single user | Selecting the Single user option will enable Unity |

| Fields                     | Parameter                        | Additional Information            |
|----------------------------|----------------------------------|-----------------------------------|
|                            |                                  | Catalog creation in the database. |
| Databricks Runtime Version | 13.3 LTS, 12.2 LTS, and 11.3 LTS |                                   |

4. Click **Create compute**.

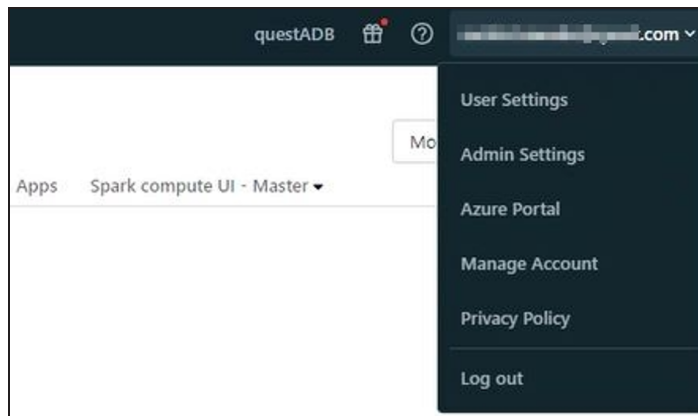
The user cluster is created.

## Creating Access Token

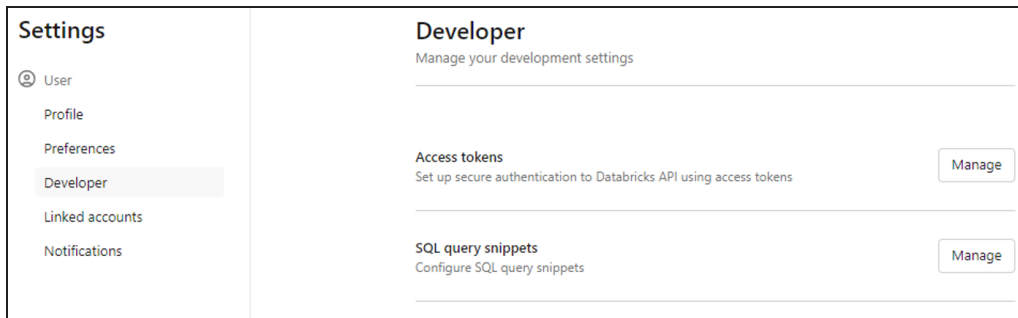
Once you have created a user cluster, you can generate the access token. You can use this access token to get access to the database using erwin DM.

To generate the access token, follow these steps:

1. In the right-top corner, click the email address, and select **User Settings** option.

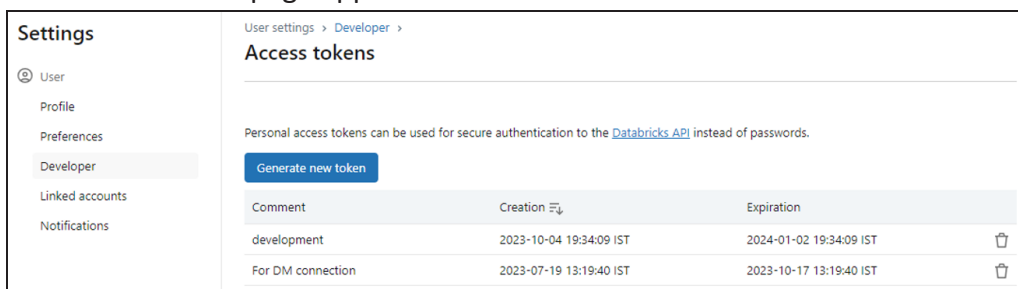


The Settings page appears.



2. Go to **Developer > Access tokens > Manage**.

The Access tokens page appears.



3. Click **Generate new token**.

The Generate new token page appears.

The screenshot shows a 'Generate new token' dialog box. It has a title bar with a close button. Inside, there is a 'Comment' section with a text input field containing 'What's this token for?'. Below that is a 'Lifetime (days)' section with a dropdown menu showing '90'. At the bottom right are 'Cancel' and 'Generate' buttons.

4. Add comments and set an expiry for the token. Then, click Generate.

The token is generated and this expires after certain days. After expiry, you can generate another token using the above steps.

## Accessing the JDBC String

## DM Connect for DI

Once the access token is generated, use the following steps to run the user cluster and copy the JDBC string.

1. Go to **Compute** tab and locate the user cluster that you created.
2. Click ► to run the cluster.
3. Once the user cluster starts, open the cluster, and expand the **Advanced options**.
4. Go to the **JDBC/ODBC** tab, copy the JDBC string from the **JDBC URL** section. This string can be used to connect to the Databricks from erwin DM for reverse engineering. For more information about the connection details, refer to the [Database Connection Parameters](#) topic.

▼ Advanced options

Azure Data Lake Storage credential passthrough ⓘ  
☐ Enable credential passthrough for user-level data access

Spark   Logging   Init Scripts   **JDBC/ODBC**   Permissions

Server Hostname

Port

Protocol

HTTP Path

JDBC URL   2.6.25 or later   ▼ ⓘ

Databricks supports drivers released within the last two years. [Download drivers here.](#)

```
jdbc:databricks://adb-5019866263455093.13.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/5019866263455093/0925-140915-ol47lgxh;AuthMech=3;UID=token;PWD=<personal-access-token>
```

When you copy the JDBC string, ensure that you copy the string up to "AuthMech=3".

