



## **erwin Data Modeler**

### **Feature Tour**

**Release 2021 R1**

## 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 © 2021 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 click **Sign in** or **Sign up** to 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).

### erwin Data Modeler News and Events

Visit [www.erwin.com](http://www.erwin.com) 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

---

<b>Legal Notices</b>	<b>2</b>
<b>Contents</b>	<b>5</b>
<b>Introduction</b>	<b>7</b>
<b>erwin Mart Administrator UI Facelift</b>	<b>8</b>
<b>NoSQL Modeling</b>	<b>9</b>
MongoDB Support	10
Cassandra Support	11
Couchbase Support	13
Migrating Relational Models to NoSQL Models	14
Migration by Changing the Target Database	14
Migration by Deriving a Model	16
Reverse Engineering Models	23
Forward Engineering Models	33
Comparing Changes using Complete Compare	38
<b>JSON and AVRO Support</b>	<b>54</b>
Reverse Engineering Models - JSON and AVRO	55
Forward Engineering Models - JSON and AVRO	59
<b>Oracle Support Summary</b>	<b>65</b>
<b>Microsoft SQL Server Support</b>	<b>68</b>
<b>Microsoft Azure SQL Server Support</b>	<b>69</b>
<b>MySQL Support</b>	<b>72</b>
<b>Data Vault 2.0 Support</b>	<b>74</b>

---

<b>Productivity and UI Enhancements .....</b>	<b>76</b>
Welcome Page .....	76
Objects Count Pane .....	77
Properties Pane .....	78
Object Browser .....	79
Normalization and Denormalization .....	80
Reverse Engineering and Forward Engineering Wizard Redesign .....	81
Improved Speed Mode .....	83
<b>Database Connectivity .....</b>	<b>84</b>

## 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) 2021 R1 release.

The features and enhancements introduced in this release are:

- [erwin Mart Administrator UI Facelift](#)
- [NoSQL Modeling](#)
- [JSON and AVRO Support](#)
- [Oracle 12c R2, 18c,19c, and 21c](#)
- [Microsoft SQL Server 2019](#)
- [Microsoft Azure SQL](#)
- [MySQL](#)
- [Data Vault 2.0 Support](#)
- [Productivity and UI Enhancements](#)
- [Database Connectivity](#)

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

## erwin Mart Administrator UI Facelift

erwin Mart Administrator now comes with a brand new UI that follows Google's Material Design principles. The redesigned UI offers an improved user experience with its modern look and feel, dark and light modes, and graphical buttons and icons.

Apart from the overall facelift, the wiki-like editable Home page lets you add information, such as key text, process diagrams, important hyperlinks, resources, and much more. Also, the configurable Dashboard let's you add and view a pictorial presentation of your data and actions on the Mart. You can add charts for your data footprint, model overview and history, profile data, and session overview.

For more information, refer to the [erwin Mart Online Help](#).



## NoSQL Modeling

Along with relational databases, erwin Data Modeler (DM) now supports the following NoSQL, non-relational databases as target databases:

- [MongoDB 4.x](#)
- [Cassandra 3.x](#)
- [Couchbase 6.x](#)

These NoSQL databases support all the erwin DM features and functions. The following sections will take you through these features with MongoDB database as an example:

- [Migrating a relational model to NoSQL model](#)
- [Reverse engineering models from database and script](#)
- [Forward engineering models to database](#)
- [Comparing changes using Complete Compare](#)

## MongoDB Support

erwin Data Modeler (DM) now supports [MongoDB 4.x](#) as a target database. This implementation supports the following objects:

- Databases
- Collection
  - Collation
- Index
- Relationships
- User IDs
  - Roles
- View

The following table lists 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>

## Cassandra Support

erwin Data Modeler (DM) now supports [Cassandra 3.x/4.x](#) as a target database. This implementation supports the following objects:

- Aggregate
- Function
- Keyspace
- Materialized View
  - Materialized View Column
- Role
- Table
  - Table Column
  - Index
- User Type

The following table lists the supported data types:

Category	Data Type	Supported Constants
Native	• ascii	• string
	• bigint	• integer
	• blob	• blob
	• boolean	• boolean
	• counter	• integer
	• date	• integer, string
	• decimal	• integer, float
	• double	• integer, float
	• float	• integer, float

	<ul style="list-style-type: none"> <li>• inet</li> <li>• int</li> <li>• smallint</li> <li>• text</li> <li>• time</li> <li>• timestamp</li> <li>• timeuuid</li> <li>• tinyint</li> <li>• uuid</li> <li>• varint</li> </ul>	<ul style="list-style-type: none"> <li>• string</li> <li>• integer</li> <li>• integer</li> <li>• string</li> <li>• integer, string</li> <li>• integer, string</li> <li>• uuid</li> <li>• integer</li> <li>• uuid</li> <li>• integer</li> </ul>
Collection	<ul style="list-style-type: none"> <li>• list</li> <li>• map</li> <li>• set</li> </ul>	
Tuple	<ul style="list-style-type: none"> <li>• tuple</li> </ul>	

## Couchbase Support

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

- Bucket
- Document
  - Field
- Full Text Index
- Global Index
- User ID
- View

Following are the supported data types:

- MISSING
- NULL
- BOOLEAN
- NUMBER
- STRING
- ARRAY
- OBJECT
- BINARY

# Migrating Relational Models to NoSQL Models

You can convert and migrate your relational models to NoSQL models in two ways:

- [Changing the target database](#)
- [Deriving a model](#)

This topic walks you through the steps to migrate a SQL Server model to a MongoDB model. Similarly, you can migrate your relational models to Cassandra and Couchbase models.

**Note:** Ensure that you keep a backup of your original models.

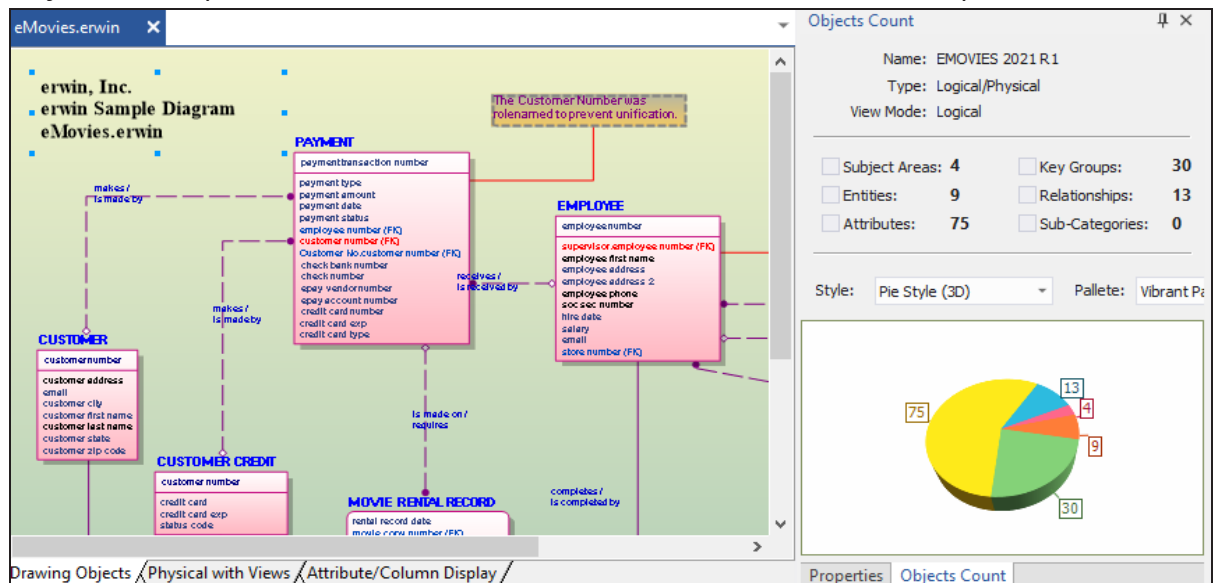
## Migration by Changing the Target Database

To migrate by changing the target database, follow these steps:

1. Open your relational model in erwin Data Modeler (DM).

**Note:** Ensure that you are in the Physical mode.

For example, the following image uses the sample eMovies.erwin model. In the **Objects Count** pane, note the number of tables, columns, and relationships.



2. On the ribbon, click **Actions > Target Database** or on the status bar, click the database name.

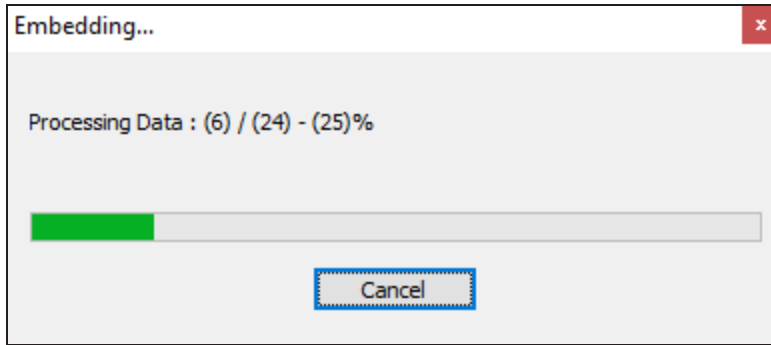
The erwin Data Modeler -- Target Server screen appears.

The screenshot shows the 'erwin Data Modeler -- Target Server' dialog box. It has a title bar with a close button (X). Inside, there are two dropdown menus at the top: 'Database:' set to 'SQL Server' and 'SQL Server Version' set to '2016/2017'. Below these, there is a section for 'Default SQL Server Datatype' with a dropdown set to 'char(18)'. Underneath that is a section for 'Default Non-Key Null Option' with two radio buttons: 'NOT NULL' (unselected) and 'NULL' (selected). At the bottom right are 'OK' and 'Cancel' buttons.

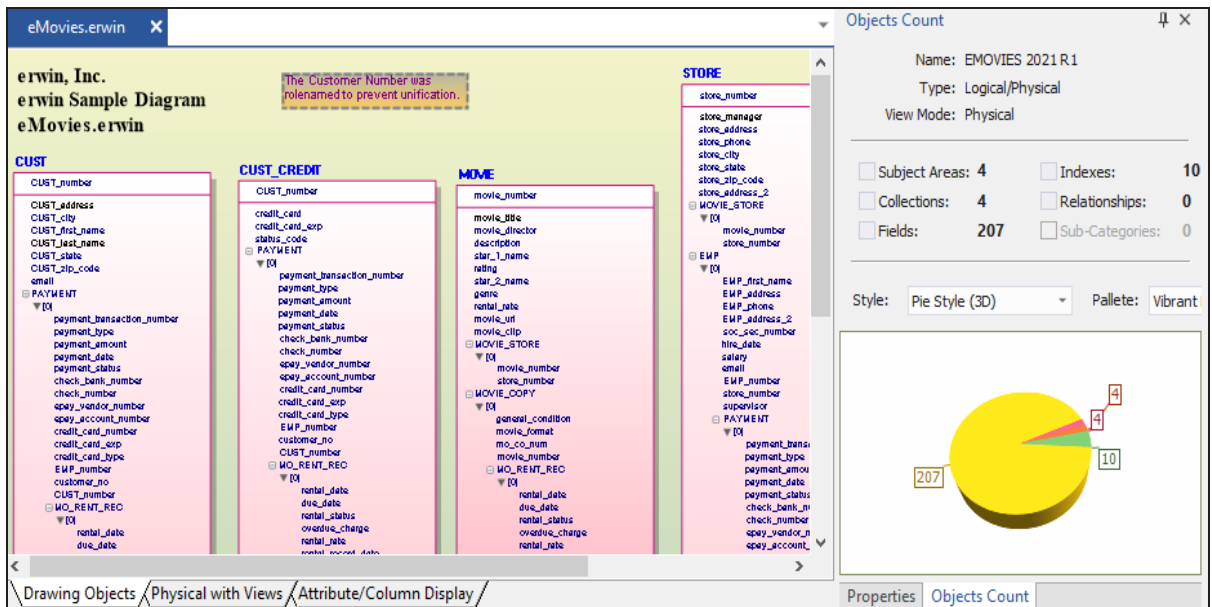
3. In the **Database** drop-down list, select MongoDB.  
By default, the Auto Denormalization check box is selected. Keep it selected.

The screenshot shows the 'erwin Data Modeler -- Target Server' dialog box with 'MongoDB' selected in the 'Database:' dropdown. The 'MongoDB Version' dropdown is set to '4.x'. The 'Auto Denormalization' checkbox is checked. The 'Default MongoDB Datatype' dropdown is set to 'String'. The 'Default Non-Key Null Option' section has a 'NOT NULL' checkbox which is unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

4. Click **OK**.  
The conversion process starts.



Once the conversion is complete, the existing model is migrated to a NoSQL database.



In the **Objects Count** pane, note that instead of tables and columns, we now have collections and fields. Also, the Relationships count has changed to 0. The migration process converts and merges multiple tables, columns, and relationships to the NoSQL format according to the database that you select.

**Note:** This migration method overwrites the existing model once you save it. Hence, we recommend that you keep a backup of your original model.

## Migration by Deriving a Model

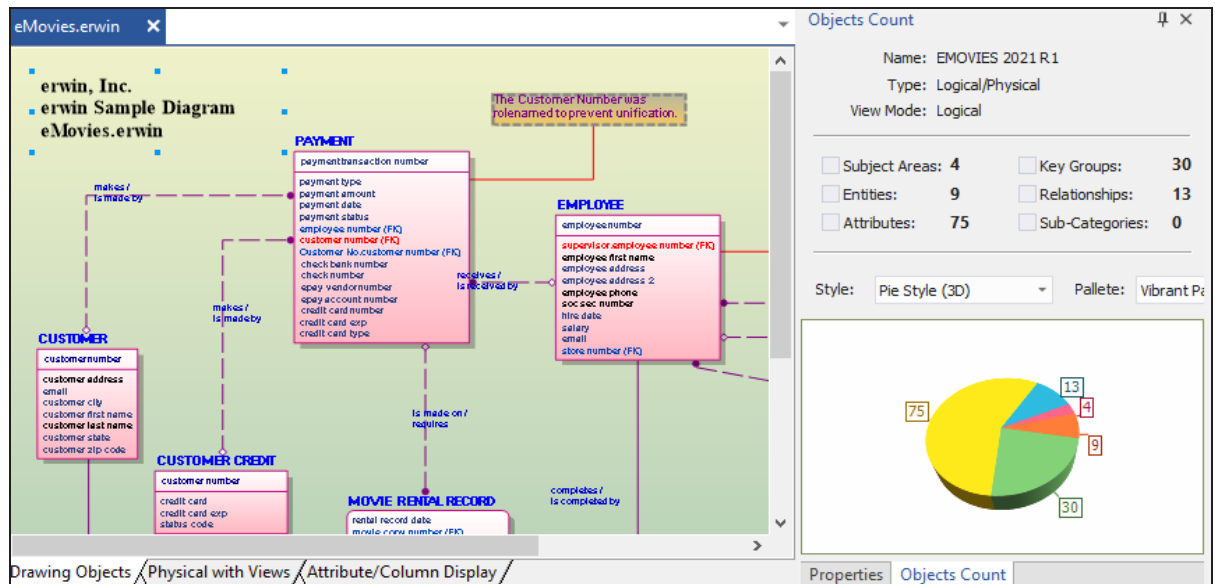
To migrate by deriving a model, follow these steps:



1. Open your relational model in erwin Data Modeler (DM).

**Note:** Ensure that you are in the Physical mode.

For example, the following image uses the sample eMovies.erwin model. In the **Objects Count** pane, note the number of tables, columns, and relationships.



2. On the ribbon, click **Actions > Design Layers > Derive New Model**.

The Derive Model screen appears. By default, the Source Model is set to your current model.

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: SQL Server
Version: 2016/2017

☐ Auto Denormalization
☐ Auto Normalization
☐ Relationships

< Back
Next >
Derive
Close
Help

- In the **Database** drop-down list, select **MongoDB**.  
By default, the Auto Denormalization check box is selected. Keep it selected.

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

☒ Auto Denormalization

☐ Auto Normalization

☐ Relationships

< Back

Next >

Derive

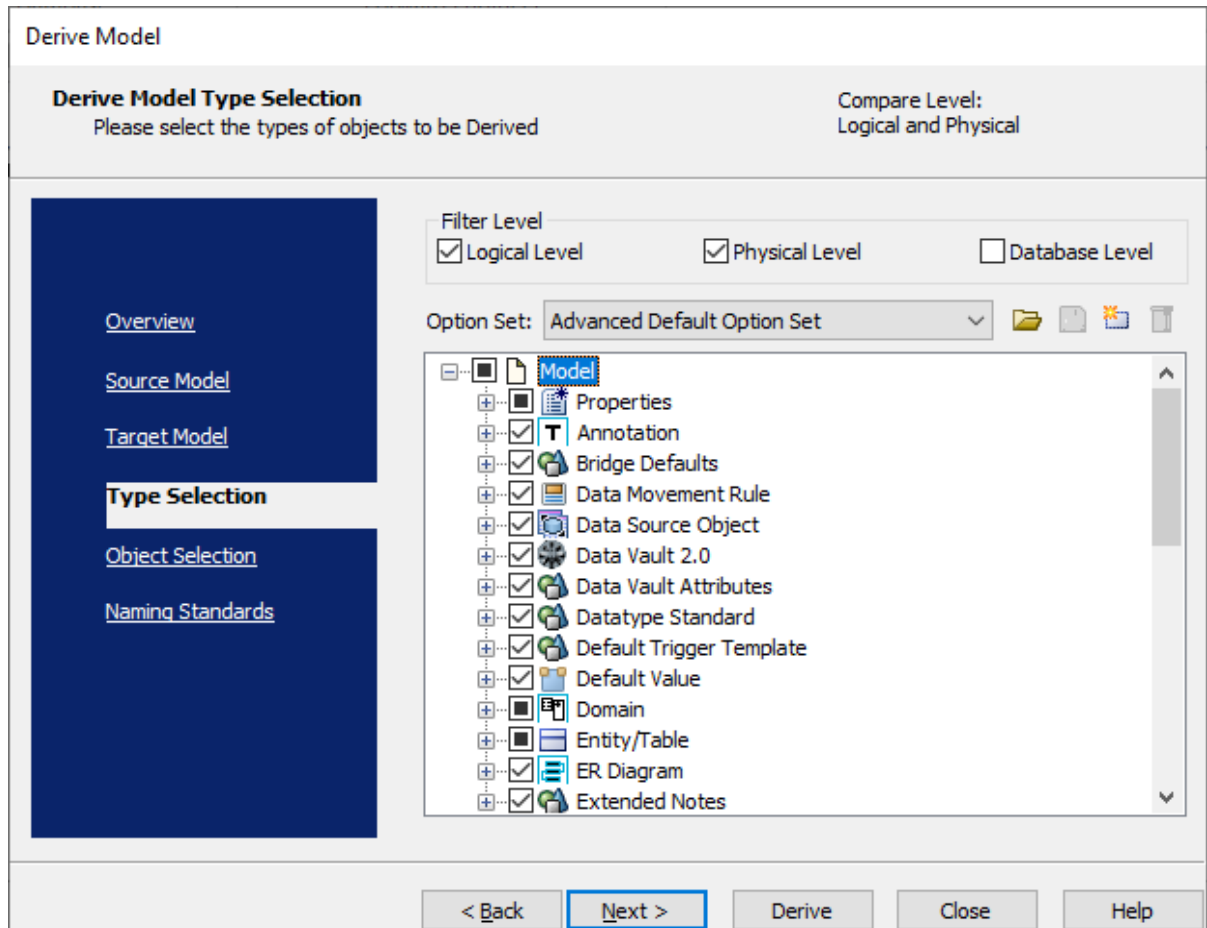
Close

Help

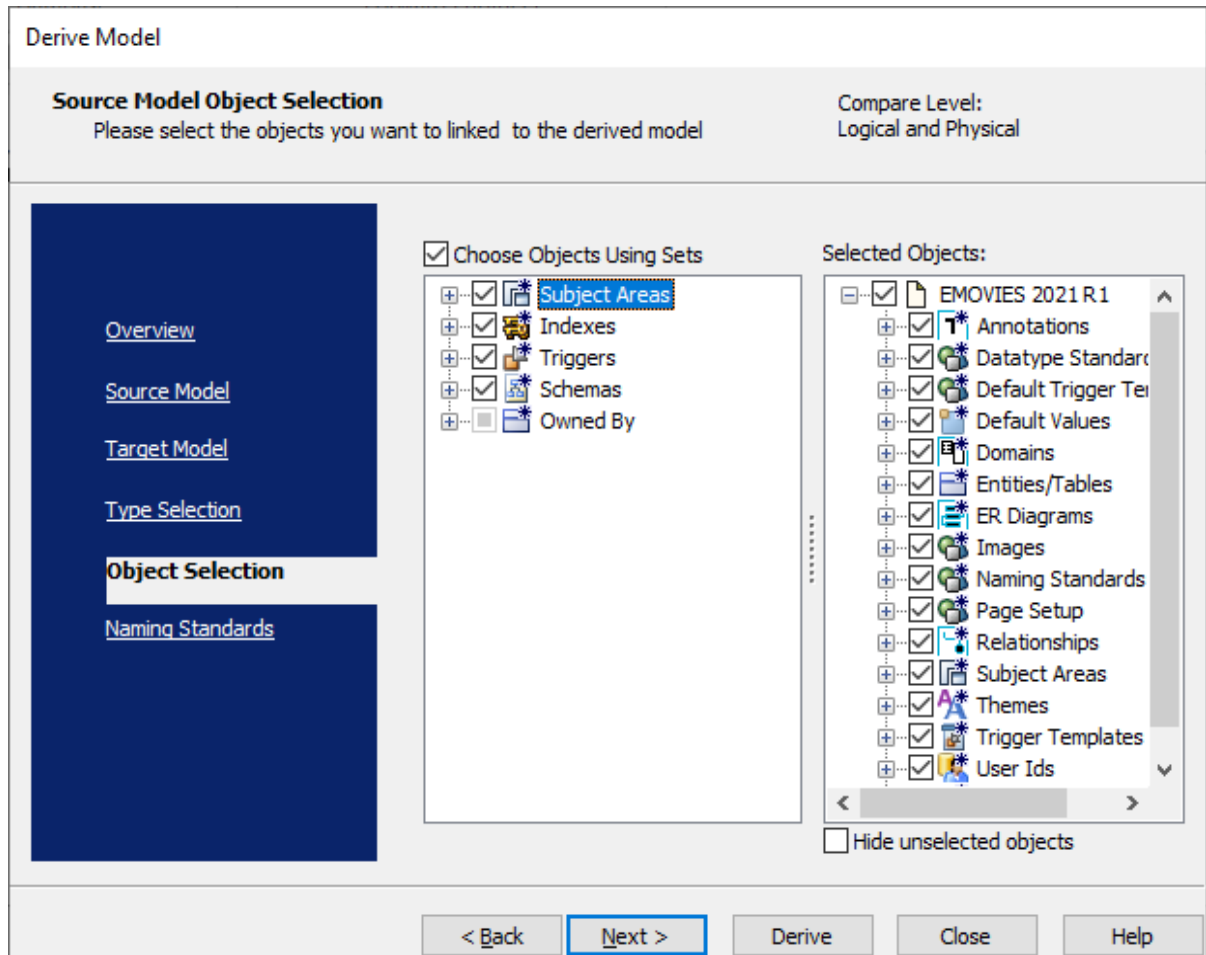
4. Click **Next**.

**Note:** If the Type Resolution screen appears, click **Finish**.

The Type Selection section appears.



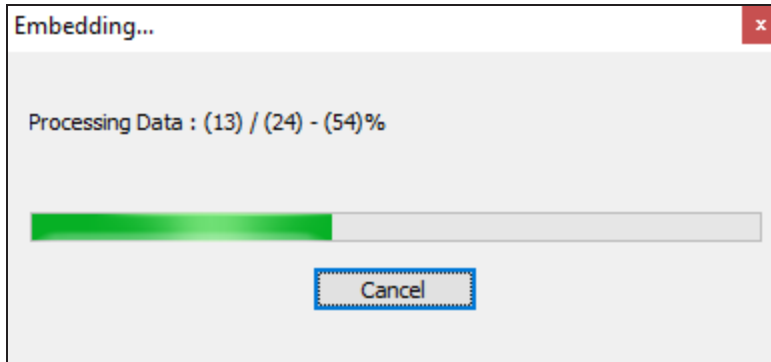
5. Select the types of objects that you want to derive into the target MongoDB model.
6. Click **Next**.  
The Object Selection section appears. Based on the object types you selected in step 5, it displays a list of objects.



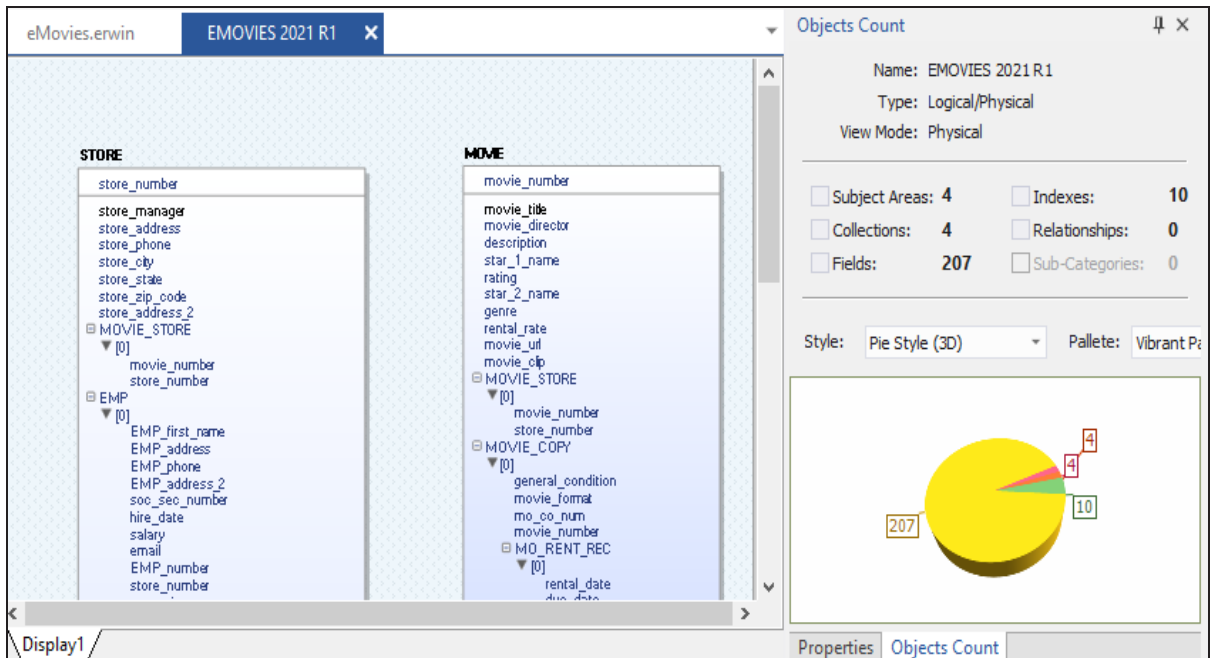
7. Select the objects that you want to derive into the target MongoDB model.

8. Click **Derive**.

The model derivation process starts.



Once the conversion is complete, the existing model is migrated to a NoSQL database.



In the **Objects Count** pane, note that instead of tables and columns, we now have collections and fields. Also, the Relationships count has changed to 0. The migration process converts and merges multiple tables, columns, and relationships to the NoSQL format according to the database that you select.

## Reverse Engineering Models

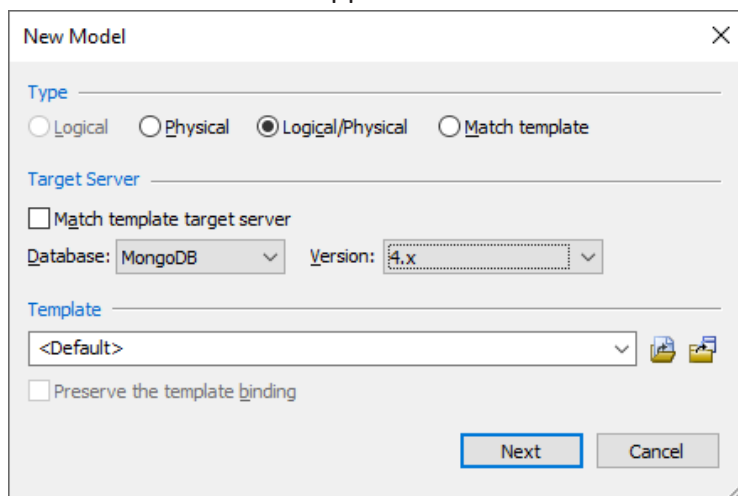
You can create a data model from a database or a script using the Reverse Engineering process.

This topic walks you through the steps to reverse engineer a MongoDB model. Similarly, you can reverse engineer a model from your Cassandra Keyspace and Couchbase Bucket.

To reverse engineer a model:

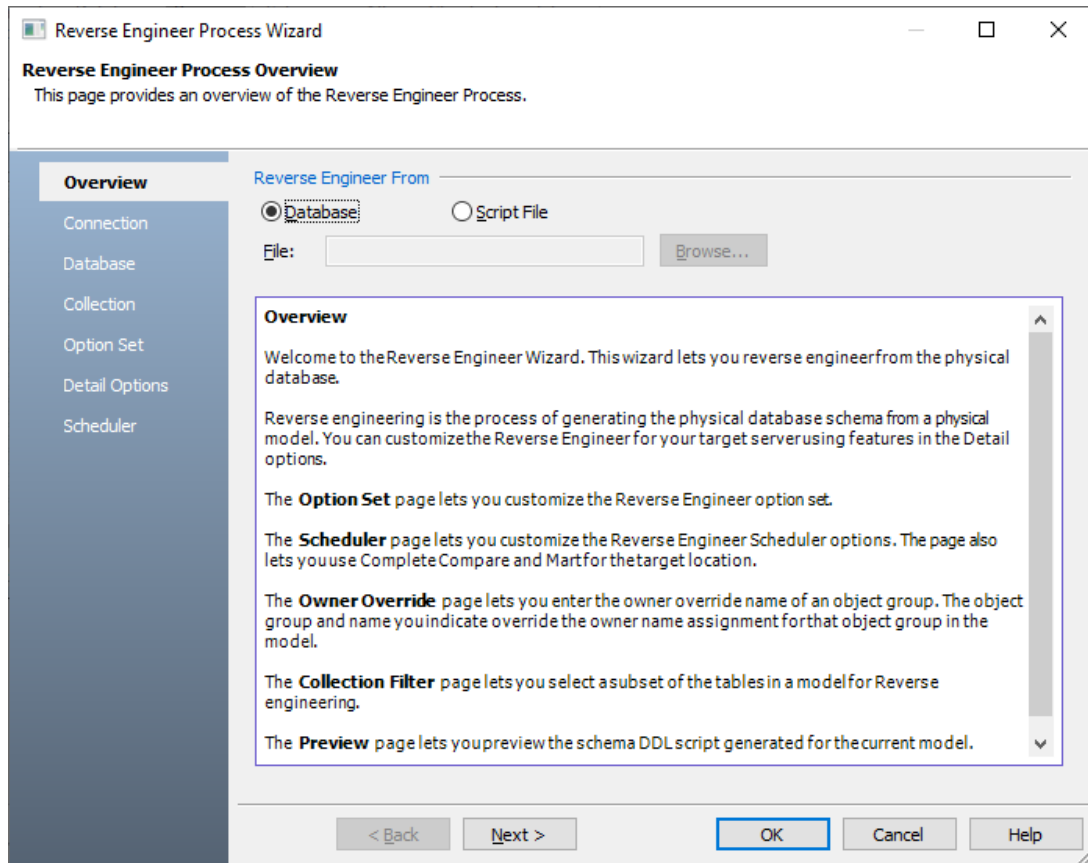
1. In erwin Data Modeler (DM), click **Actions > Reverse Engineer**.

The New Model screen appears.



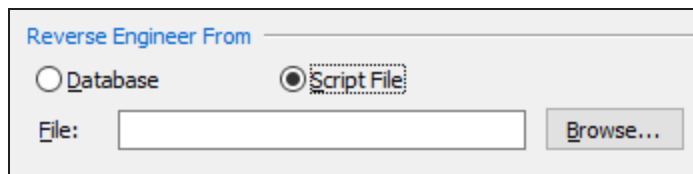
2. Click **Logical/Physical** and set **Database** to MongoDB.
3. Click **Next**.

The Reverse Engineer Process Wizard appears.



4. Click one of the following options:

- **Database:** Use this option to reverse engineer a model from your database.
- **Script File:** Use this option to reverse engineer a model from a script. Selecting this option enables the File field. Click **Browse** and select the necessary script file.



**Note:** If you click **Script File**, jump to step 8 below and ensure that Document Count or Document % is not set to zero (0).

5. Click **Next**.



The Connection section appears. Use this section to connect to the database from which you want to reverse engineer the model. You can connect to the database directly or using a connection string. The following table explains the connection parameters:

Connection Method	Parameters/Values
Connection String	<p>Specify the MongoDB Connection String.</p> <p>For example: <code>mongodb+srv://&lt;abcd&gt;:****@&lt;xyz&gt;.mongodb.net/test?retryWrites=true&amp;w=majority</code></p> <p>Replace &lt;abcd&gt; with your username and &lt;xyz&gt; with host name. The host name parameter would change based on your MongoDB deployment; standalone, replica set, or a sharded cluster.</p>
Direct	<p>Specify the host name and port number of your MongoDB deployment. Also, specify the database that you want to connect to.</p>

In the following image, for example, the connection is being established using a connection string.

The screenshot shows the 'Reverse Engineer Process Wizard' window, specifically the 'Reverse Engineer Connection Information' tab. The window has a sidebar on the left with options: Overview, **Connection**, Database, Collection, Option Set, Detail Options, and Scheduler. The main area contains the following fields and controls:

- Database:** A dropdown menu set to 'MongoDB 4.x'.
- Authentication:** A dropdown menu set to 'Database Authentication'.
- User Name:** An empty text input field.
- Password:** An empty text input field.
- Parameters and Value Table:**

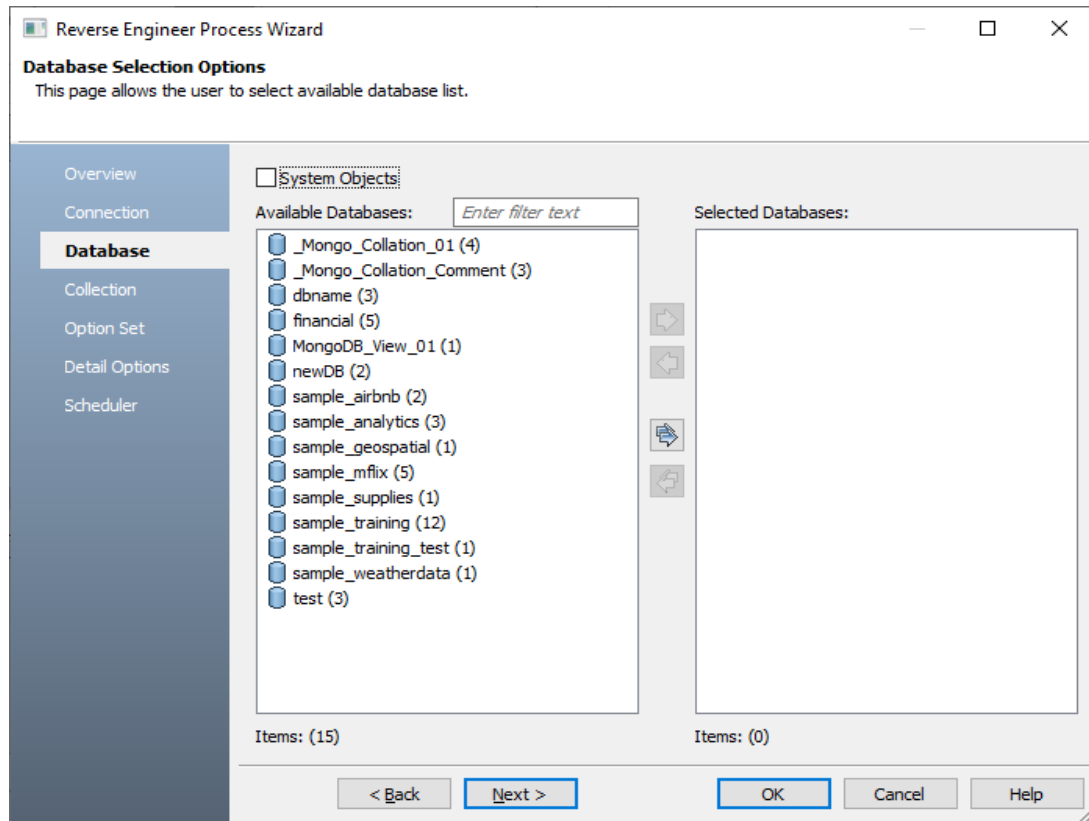
Parameters	Value
Connection Method	CONNECTION STRING
Connection String:	mongodb+srv://[redacted]:[redacted]@gra.m
- Connect/Disconnect Buttons:** Two buttons located below the table.
- Recent Connections:** A list box containing one entry: 'cloud.mongodb:([redacted]) (MongoDB 4.x)'. It has a scrollbar at the bottom.
- Navigation Buttons:** '< Back', 'Next >', 'OK' (highlighted with a blue border), 'Cancel', and 'Help'.


6. Click **Connect**.

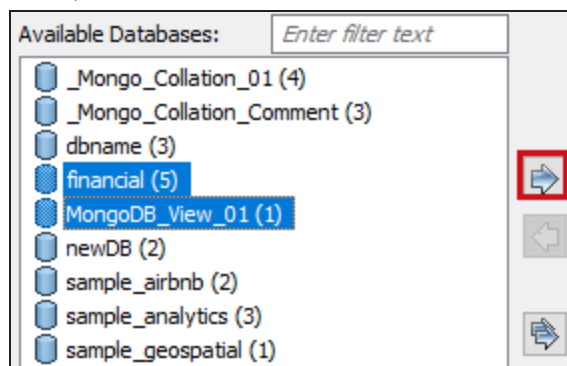
On successful connection, your connection information is displayed under Recent Connections.

7. Click **Next**.

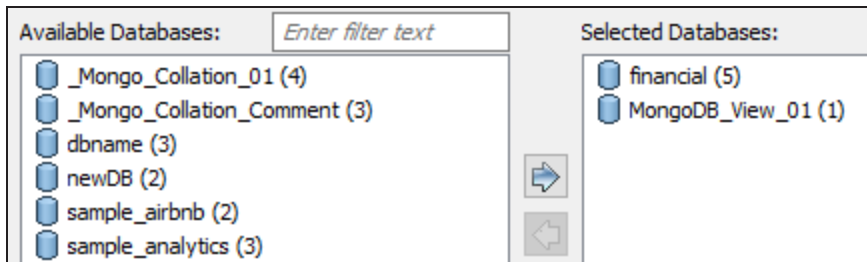
The Database section appears. It displays a list of available databases.



8. Under **Available Databases**, select the databases that you want to reverse engineer. Then, click .

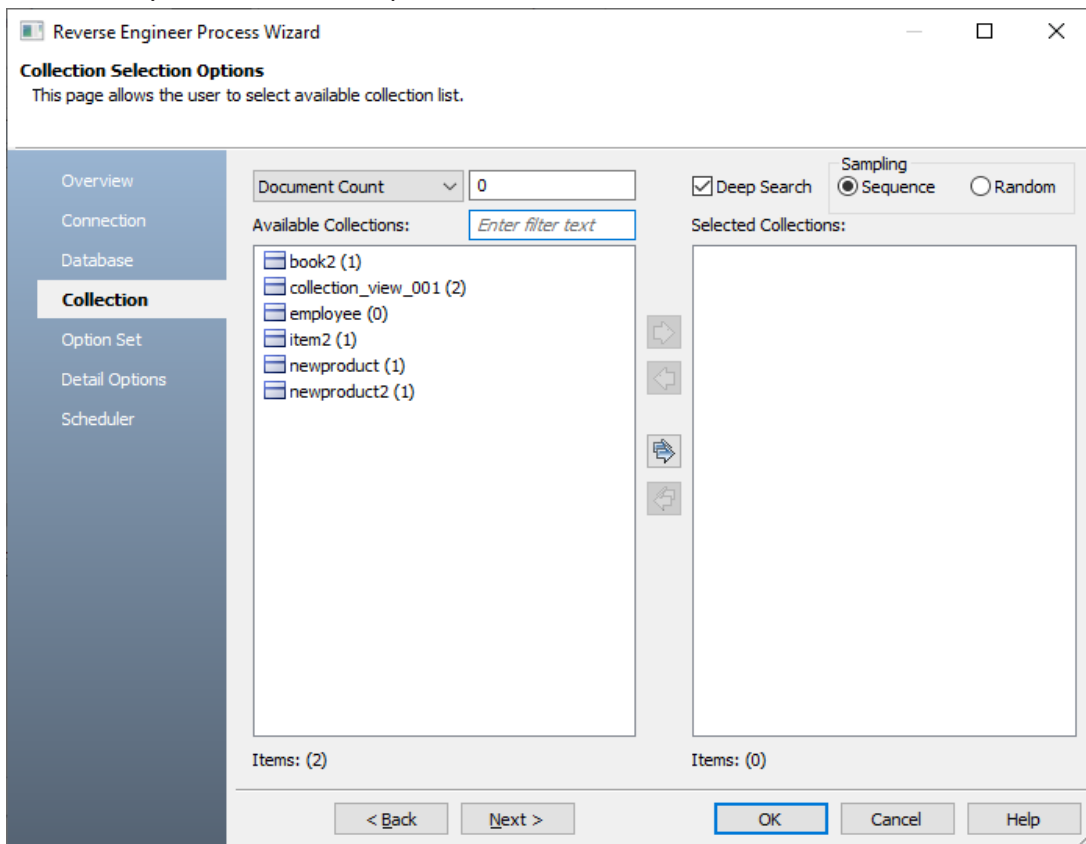


This moves the selected databases under Selected Databases.




9. Click **Next**.

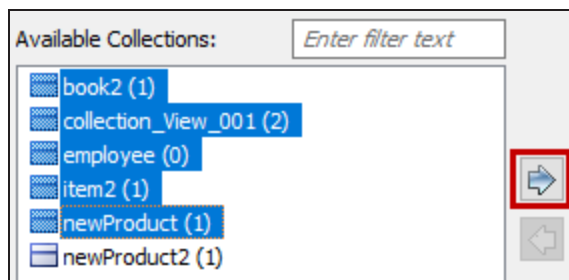
The Collection section appears. It displays a list of available collections in the databases that you selected in step 8.



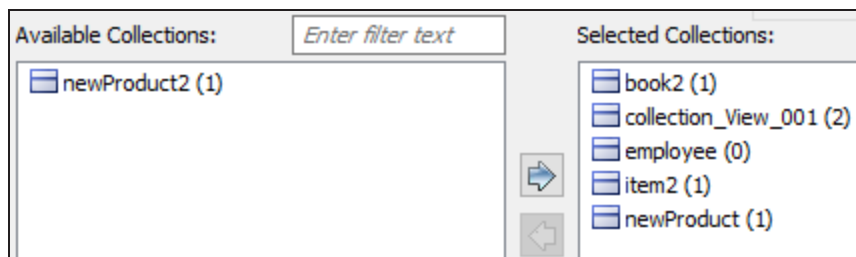
10. Use the following options:

- **Document Count/Document (%)**: Use this option to specify the number of documents or percentage of total records that the newly generated model schema would contain.
- **Deep Search**: Use this option to specify whether the deep search algorithm is used to retrieve the right samples for schema generation.
- **Sampling**: Use the Sequence or Random sampling methods to sample records in the selected collections. Sampling enables you to retrieve right estimates for accurate collection schema generation.

11. Under **Available Collections**, select the collections that you want to reverse engineer. Then, click .

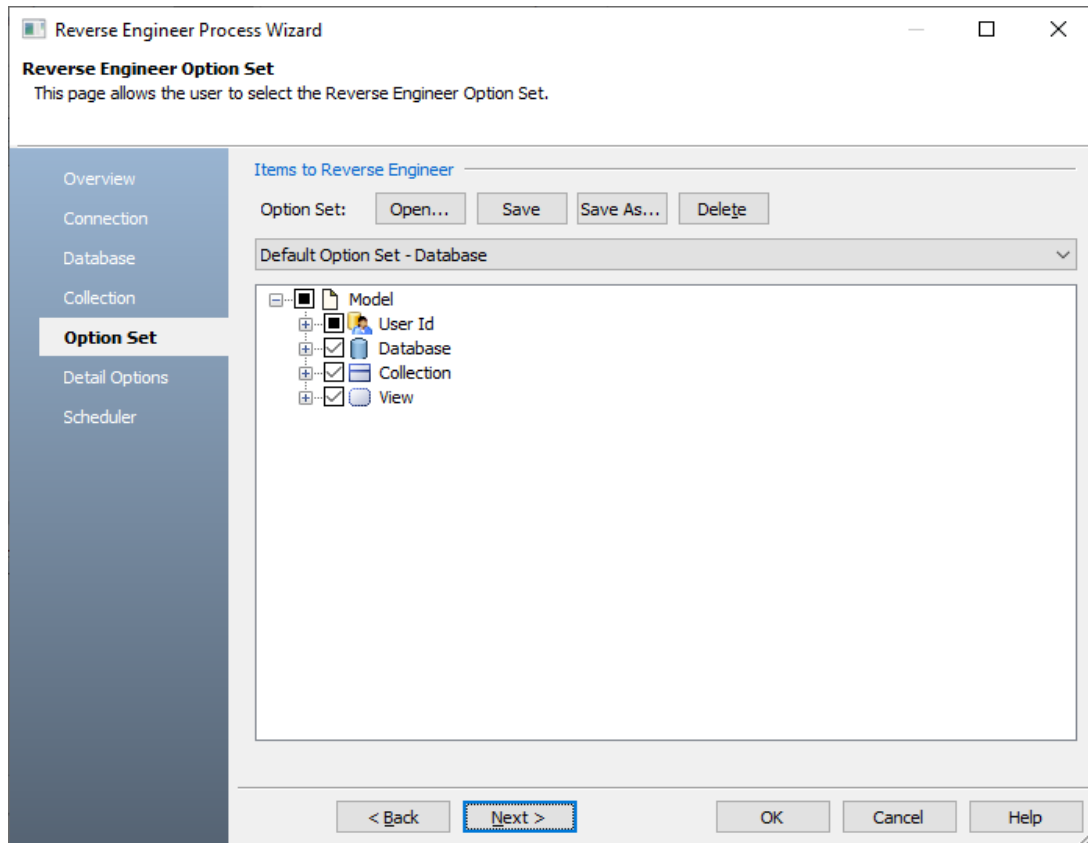


This moves the selected collections under Selected Collections.



12. Click **Next**.

The Option Set section appears. It displays the default option set. You can either use the default or a custom option set.



13. Click **Next**.

The Detail Options section appears. Set up appropriate options based on your requirement.

Reverse Engineer Process Wizard

**Reverse Engineer Detail Options**  
This page provides a Detail Options of the Reverse Engineer Process.

Overview  
Connection  
Database  
Collection  
Option Set  
**Detail Options**  
Scheduler

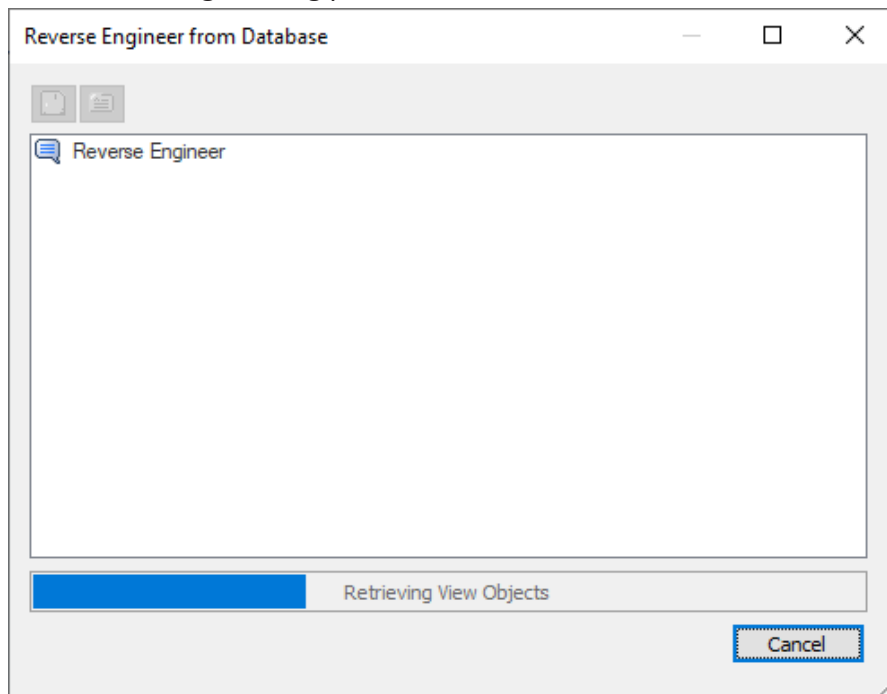
**NSM Options**  
Glossary CSV File:

**Reverse Engineer**  
☐ System Objects  
Collections/Views Owned By  
☒ All ☐ Current User  
☐ Owners (comma separated):   
**Infer**  
☐ Primary Keys ☐ Relations  
From  
☐ Indexes ☐ Names  
**Case Conversion of Physical Names**  
☒ None ☐ lower ☐ UPPER ☐ Force  
**Case Conversion of Logical Names**  
☒ None ☐ lower ☐ UPPER ☐ Mixed  
☐ Include Generated Triggers

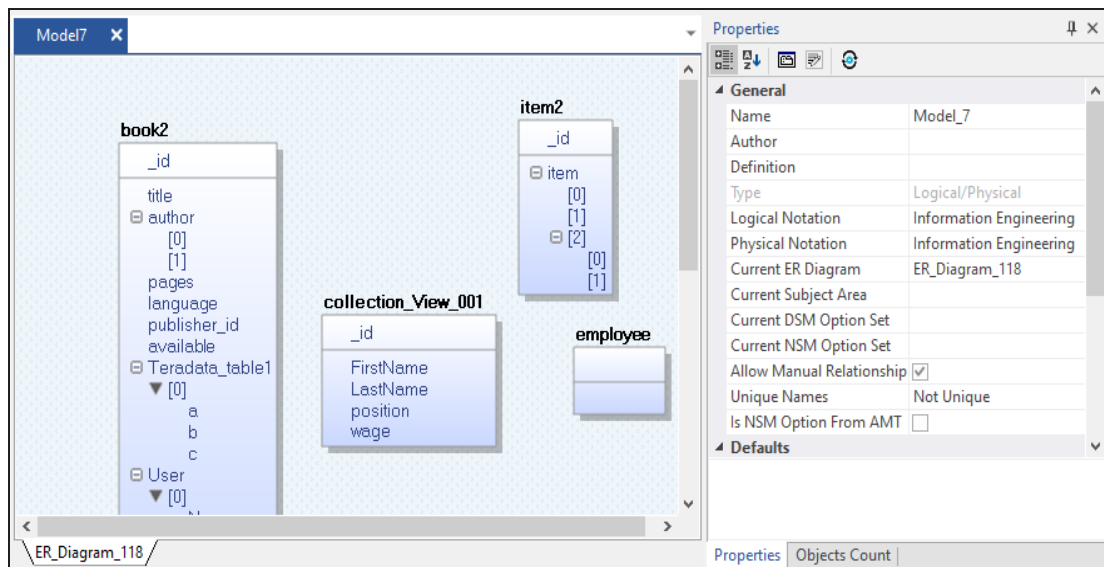
< Back **Next >** OK Cancel Help

14. Click **OK**.

The reverse engineering process starts.



Once the process is complete, based on your selections, a schema is generated and a model is created.





## Forward Engineering Models

You can generate a physical database schema from a physical model using the Forward Engineering process.

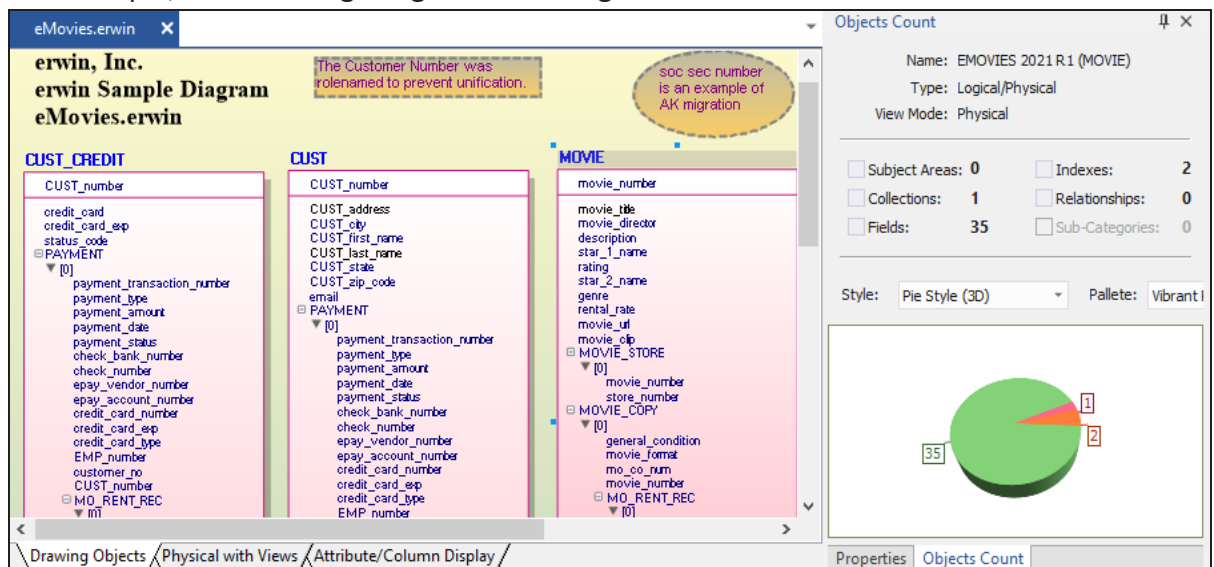
This topic walks you through the steps to forward engineer a MongoDB model. Similarly, you can forward engineer a model to your Cassandra Keyspace and Couchbase Bucket.

To forward engineer a model:

1. Open your MongoDB model in erwin Data Modeler (DM).

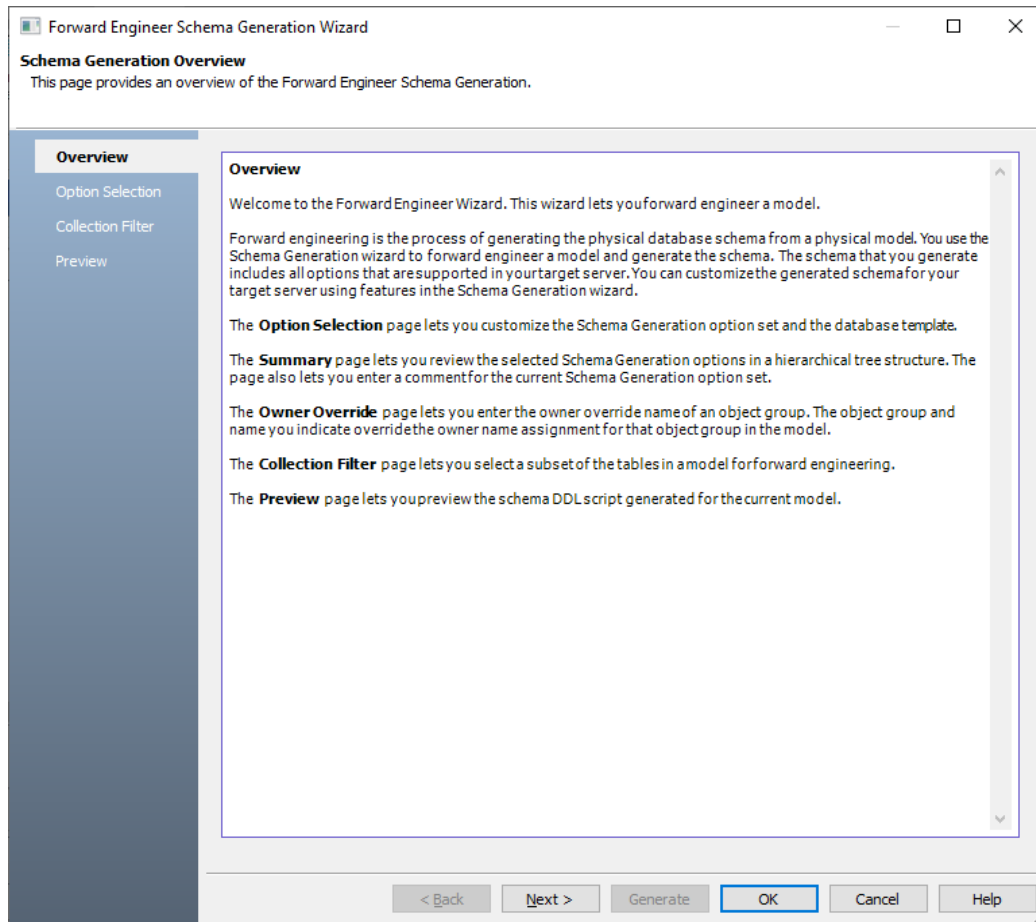
**Note:** Ensure that you are in the Physical mode.

For example, the following image uses a MongoDB model with two collections.



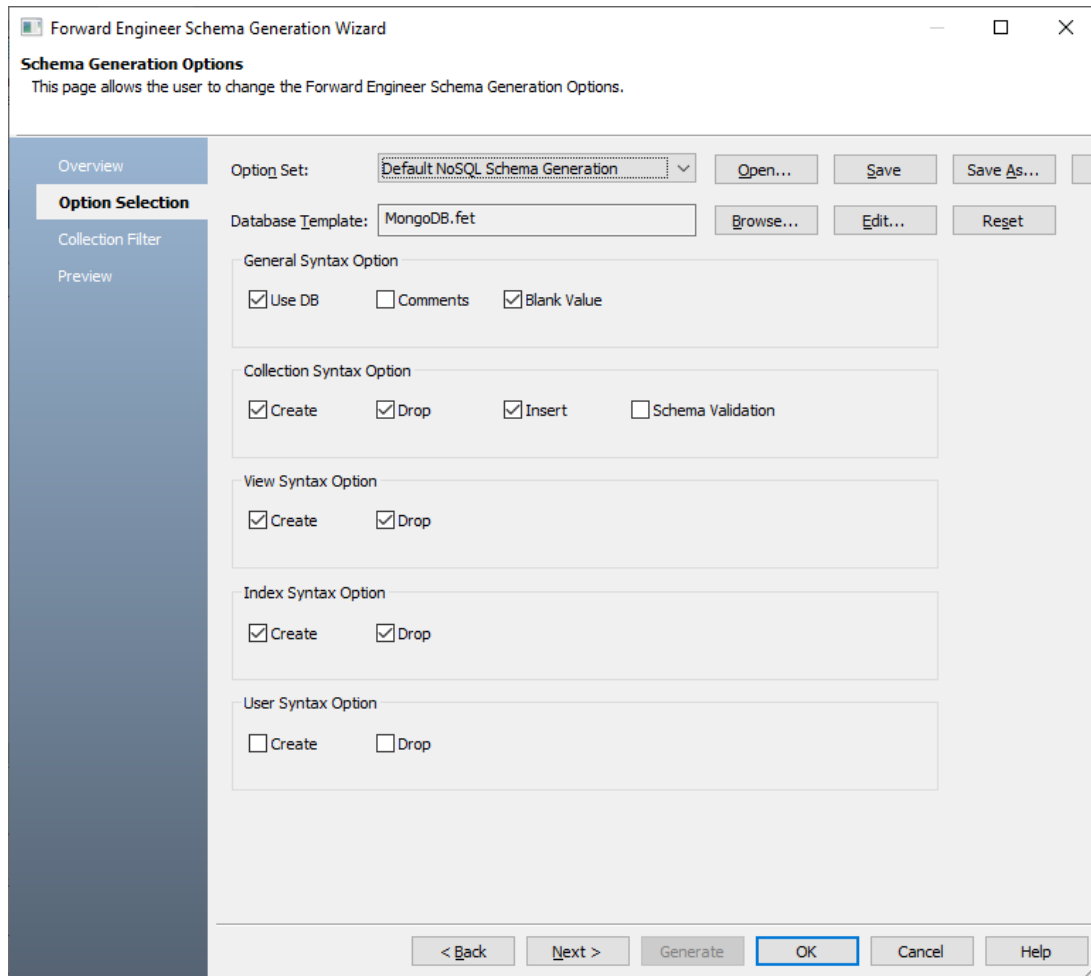
2. Click **Actions > Schema**.

The Forward Engineer Schema Generation Wizard appears.



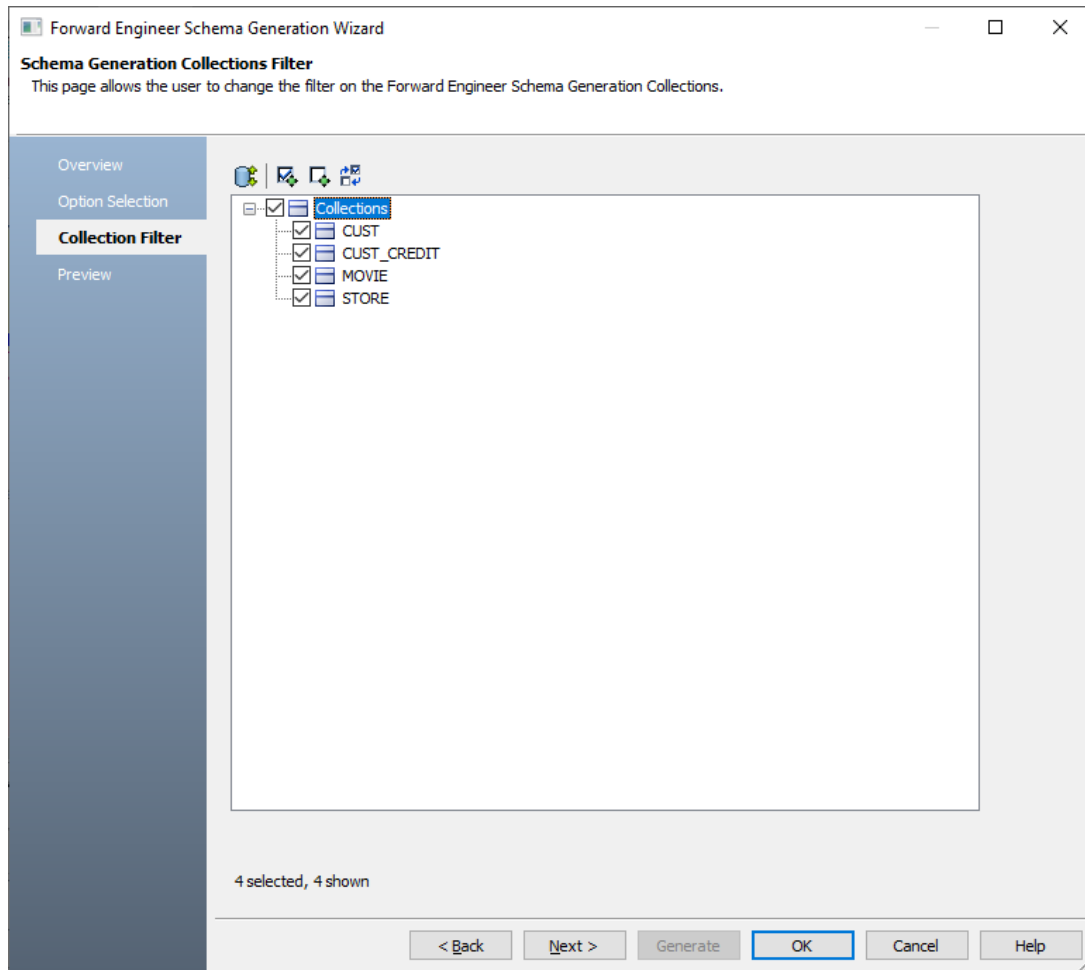
3. Click **Option Selection**.

The Option Selection section displays the default option set. Clear the **Drop** check boxes and select other syntax check boxes as required.



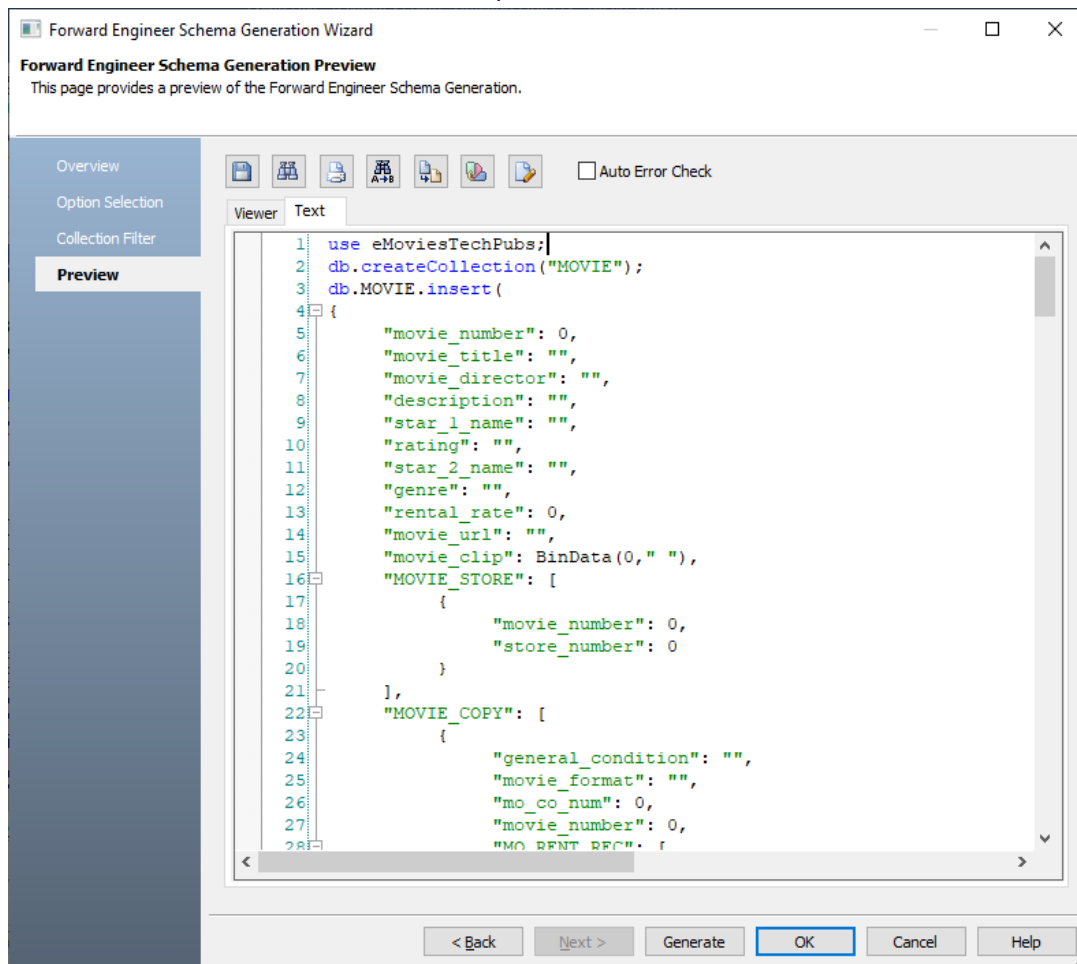
4. Click **Next**.

The Collection Filter section appears. It displays a list of collections available in your model.




5. Select the collections that you want to forward engineer.

6. Click **Preview** to view the schema script.



Use the following options:

- **Auto Error Check:** Select this option to enable auto error check by the forward engineering wizard.
- **Error Check** (

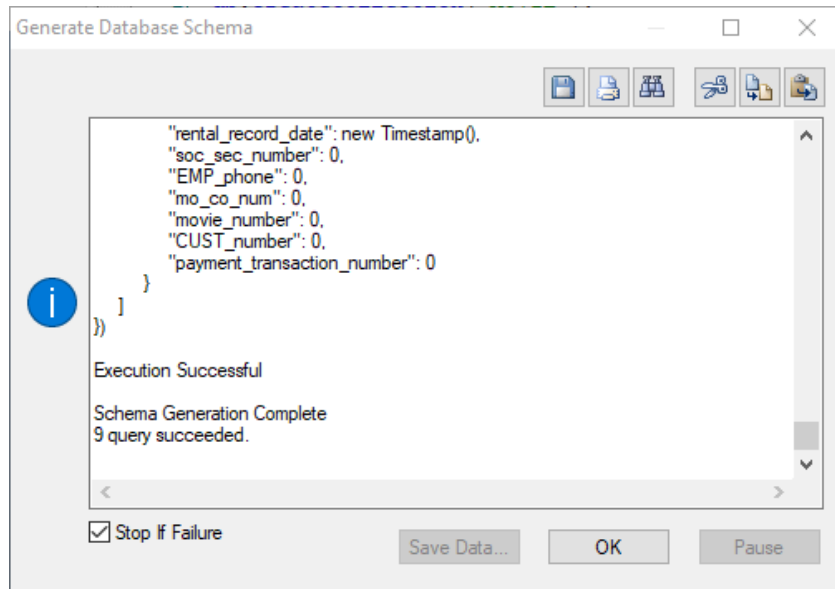
37

refer to the Forward Engineering Wizard - Preview Editor topic.

- **Save** (📁): Use this option to save the generated script in the JSON or BSON format.

7. Click **Generate**.

The forward engineering process starts. The script generates your physical database schema. You can access your database and verify the newly generated schema.



## Comparing Changes using Complete Compare

You can compare your model with database, script, or another local model to check for differences using the Complete Compare wizard. Based on the results, you can then resolve or merge differences. Thus, maintaining a consistent model and database.

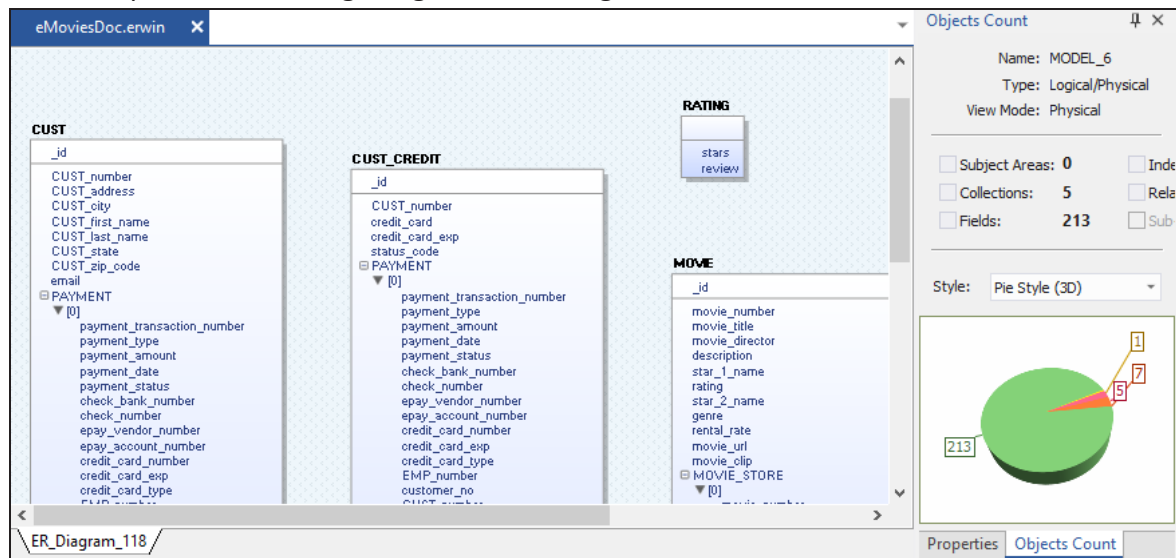
This topic walks you through the steps to compare a MongoDB model with database. Similarly, you can compare your Cassandra and Couchbase models.

To compare models with database:

1. Open your MongoDB model in erwin Data Modeler (DM).

**Note:** Ensure that you are in the Physical mode.

For example, the following image uses a MongoDB model with two collections.



## 2. Click **Actions > Complete Compare**.

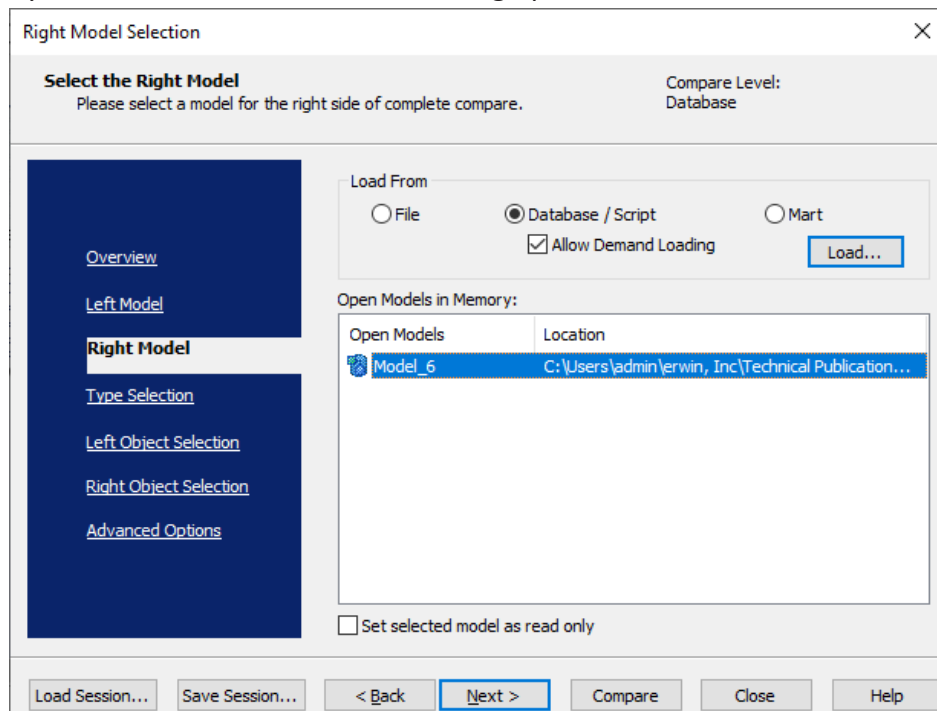
By default, the Complete Compare wizard assigns the open model as the Left Model. Hence, the Right Model section appears.

The Right Model Selection dialog box is shown. It has a title bar "Right Model Selection" and a close button. The main area is titled "Select the Right Model" and contains the text "Please select a model for the right side of complete compare." and "Compare Level: Database". On the left, there is a navigation pane with links: Overview, Left Model, Right Model (selected), Type Selection, Left Object Selection, Right Object Selection, and Advanced Options. In the center, there is a "Load From" section with radio buttons for File (selected), Database / Script, and Mart, and a "Load..." button. Below this is an "Open Models in Memory:" section with a table showing the open models. The table has two columns: "Open Models" and "Location". The first row shows "Model\_6" and "C:\Users\admin\erwin, Inc\Technical Publication...". At the bottom, there is a checkbox "Set selected model as read only" which is unchecked. The bottom of the dialog has buttons: Load Session..., Save Session..., < Back, Next > (highlighted), Compare, Close, and Help.

Open Models	Location
Model_6	C:\Users\admin\erwin, Inc\Technical Publication...

3. Click **Database/Script**.

By default, the Allow Demand Loading option is selected.



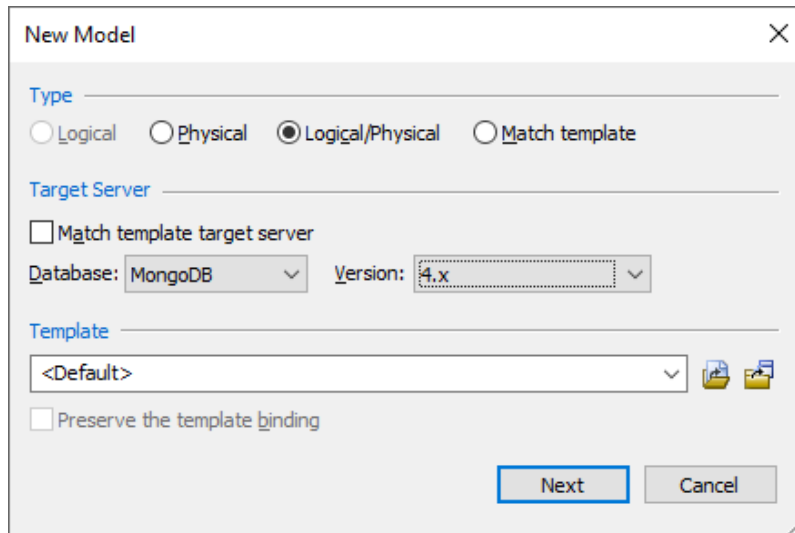
The dialog box is titled "Right Model Selection" and has a close button (X) in the top right corner. It contains a section titled "Select the Right Model" with the instruction "Please select a model for the right side of complete compare." and a "Compare Level: Database" label. On the left is a dark blue sidebar with links: Overview, Left Model, Right Model (highlighted), Type Selection, Left Object Selection, Right Object Selection, and Advanced Options. The main area has a "Load From" section with radio buttons for File, Database / Script (selected), and Mart, and a checked checkbox for "Allow Demand Loading" with a "Load..." button. Below is a table titled "Open Models in Memory:" with columns "Open Models" and "Location". The table contains one row: "Model\_6" and "C:\Users\admin\erwin, Inc\Technical Publication...". At the bottom is a checkbox "Set selected model as read only" and a row of buttons: Load Session..., Save Session..., < Back, Next > (highlighted), Compare, Close, and Help.

Open Models	Location
Model_6	C:\Users\admin\erwin, Inc\Technical Publication...

4. Click **Load**.

The New Model dialog box appears. This starts the reverse engineering process to pull a model from the database to compare.





The image shows a 'New Model' dialog box with a close button (X) in the top right corner. It is divided into three sections: 'Type', 'Target Server', and 'Template'. The 'Type' section has four radio buttons: 'Logical', 'Physical', 'Logical/Physical' (which is selected), and 'Match template'. The 'Target Server' section has a checkbox 'Match template target server' which is unchecked, followed by 'Database:' with a dropdown menu showing 'MongoDB' and 'Version:' with a dropdown menu showing '4.x'. The 'Template' section has a dropdown menu showing '<Default>' and two folder icons to its right. Below this is a checkbox 'Preserve the template binding' which is unchecked. At the bottom right are 'Next' and 'Cancel' buttons. The 'Next' button is highlighted with a blue border.

**New Model**

Type

☐ Logical ☐ Physical ☒ Logical/Physical ☐ Match template

Target Server

☐ Match template target server

Database: MongoDB Version: 4.x

Template

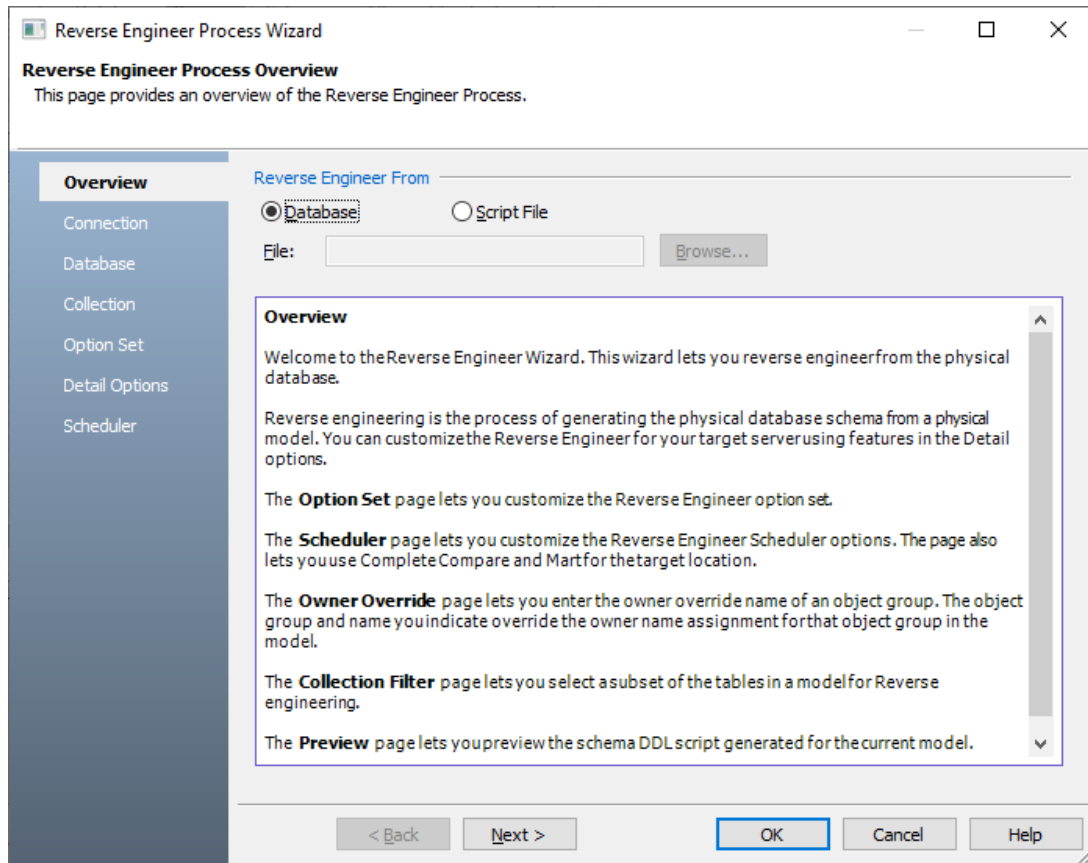
<Default>

☐ Preserve the template binding

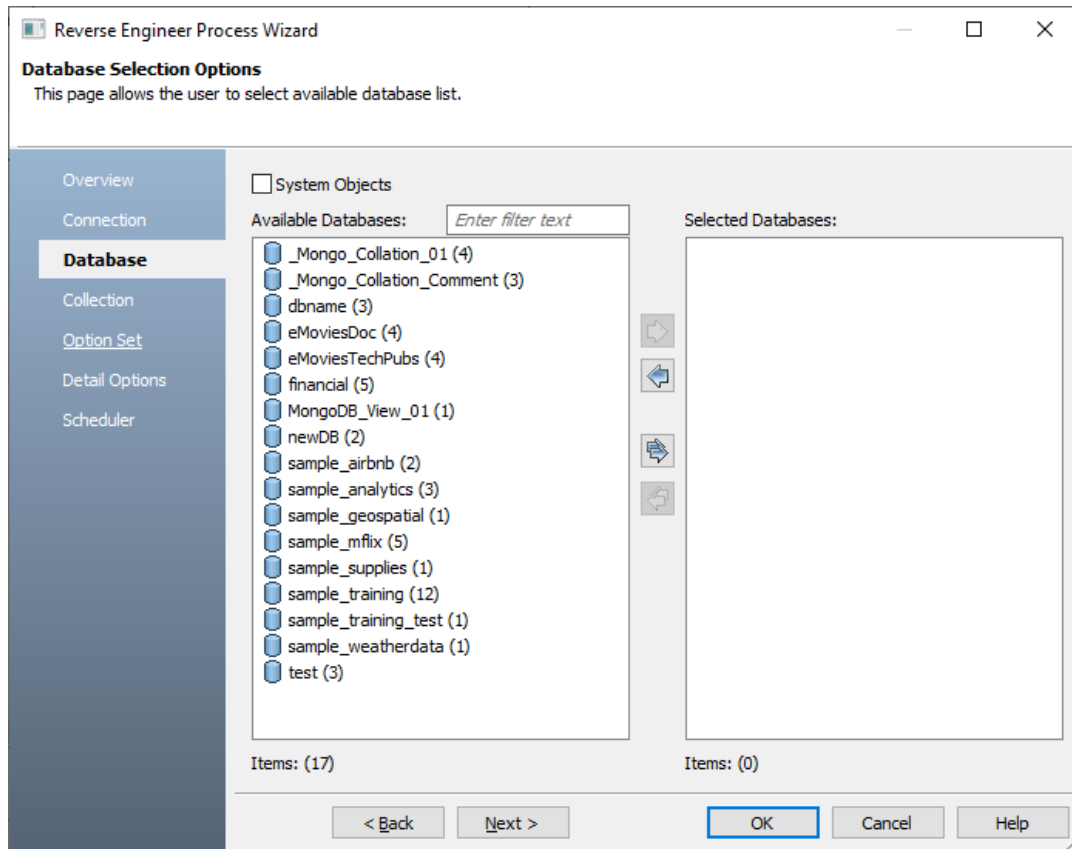
Next Cancel


5. Ensure that the Database is set to the correct one. In this case, MongoDB. Then, click **Next**.

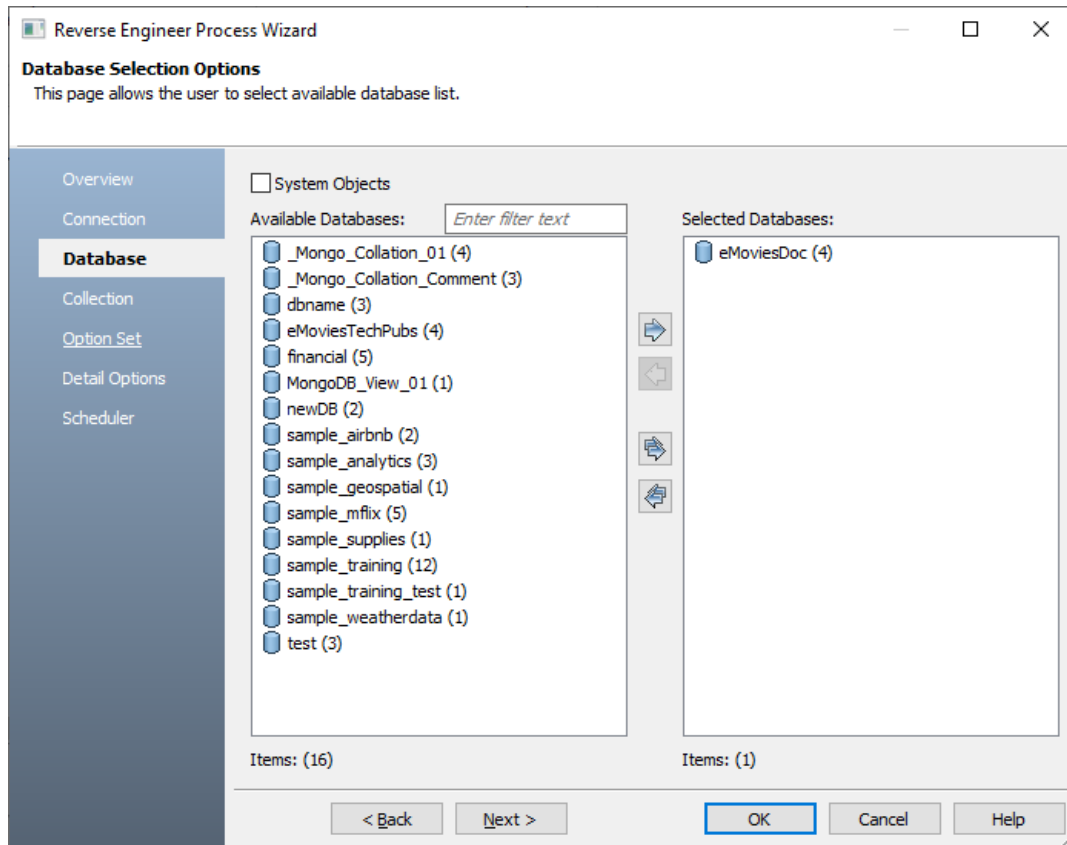
The Reverse Engineer Process Wizard appears.



6. Click **Database**. Then, click **Next**.  
The Connection section appears. Use this section to connect to the database from which you want to [reverse engineer the model](#).
7. After connection is established, click **Next**.  
The Database section appears. It displays a list of available databases.

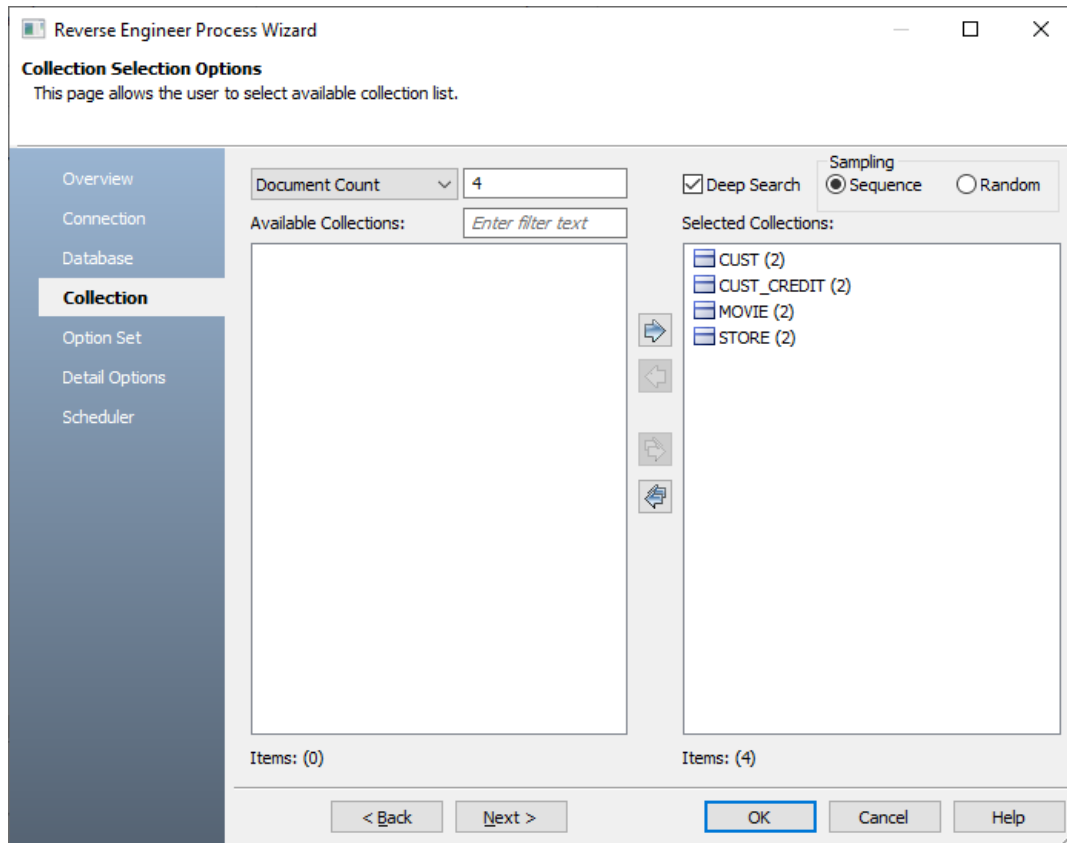


8. Under **Available Databases**, select the databases that you want to reverse engineer. Then, click . This moves the selected databases under Selected Databases.



9. Click **Next** and in the Collection section, click .

This selects all the available collections. Also, ensure that the Document Count/Document % is not set to zero (0).



10. Click **Next** and in the Option Set section, keep the default configuration.
11. Click **Next** and in the Detail Options section, keep the default configuration.
12. Click **OK**.

The reverse engineering process starts. Once the process is complete, the Right Model is set to the one that you reverse engineered.

Right Model Selection

Select the Right Model

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

Compare Level:

Database

Overview

Left Model

Right Model

Type Selection

Left Object Selection

Right Object Selection

Advanced Options

Load From

File

Database / Script

Mart

Allow Demand Loading

Load...

Open Models in Memory:

Open Models	Location
Model_14	cloud.mongodb:(...): (MongoDB 4.x)
Model_6	C:\Users\admin\erwin, Inc\Technical Publication...

☐ Set selected model as read only

Load Session...

Save Session...

< Back

Next >

Compare

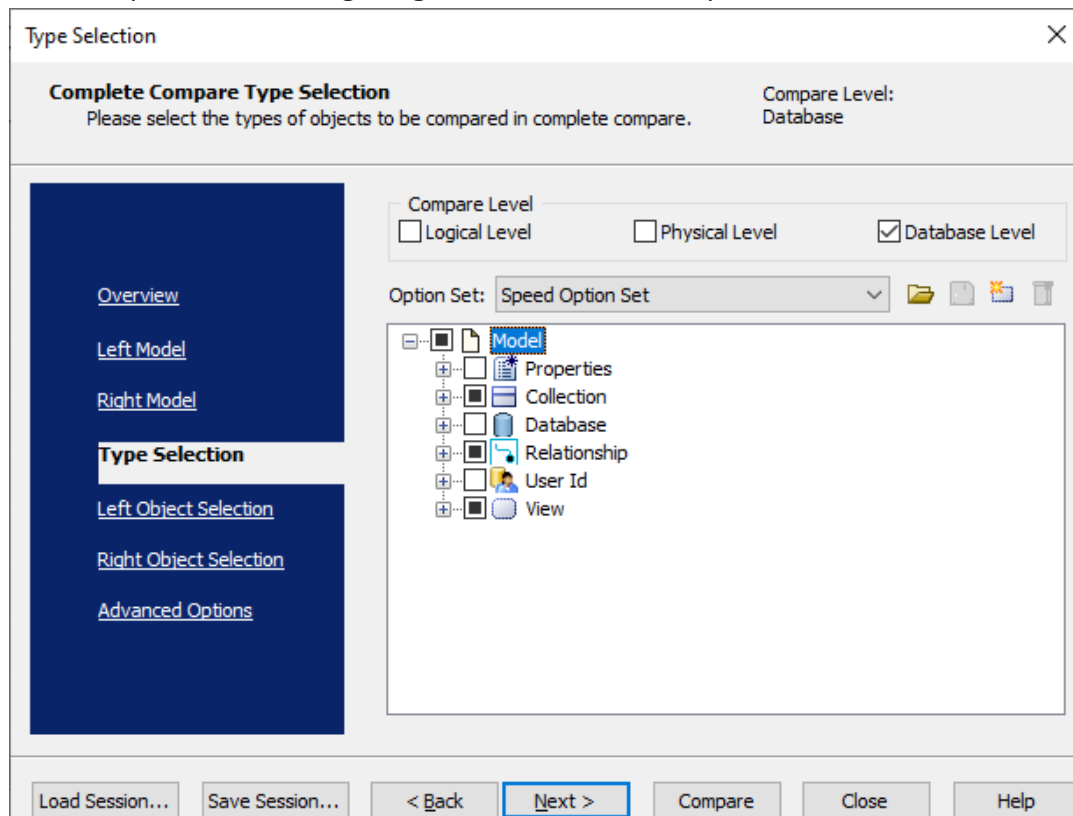
Close

Help

13. Click **Next** and in the Type Selection section, select the appropriate options.

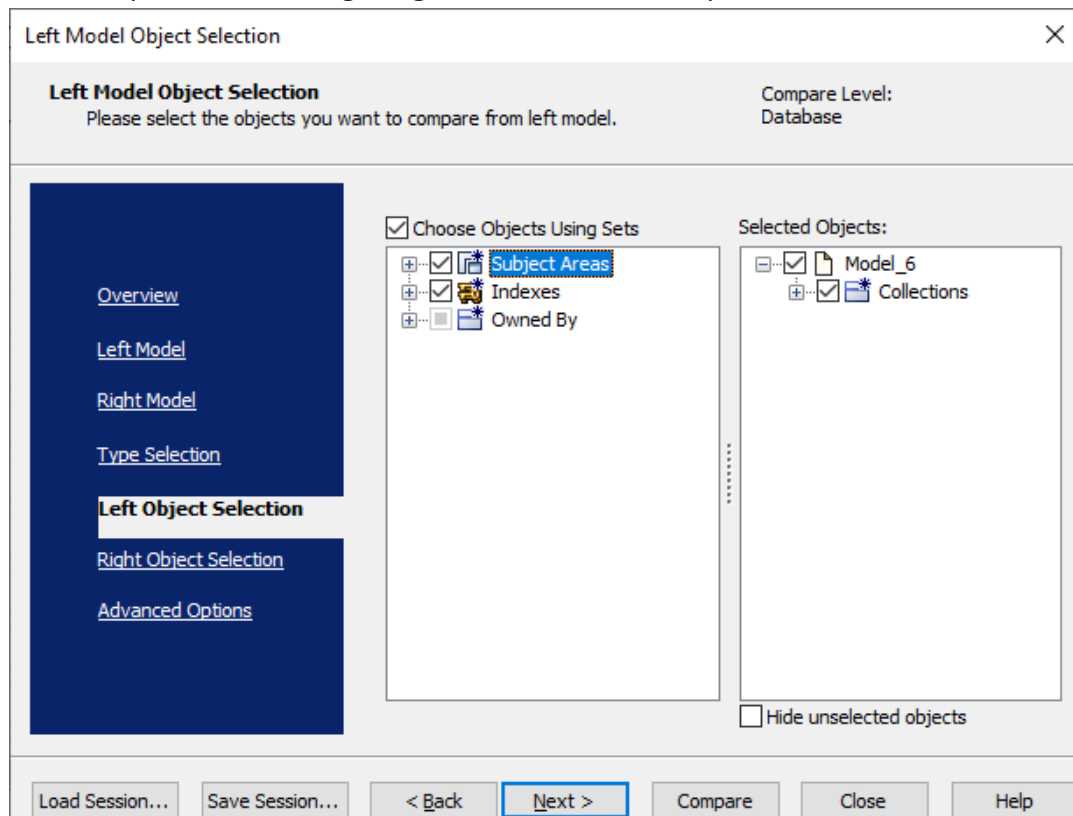
46

For example, the following image shows the default options.



14. Click **Next** and in the Left Object Selection section, select the appropriate options.

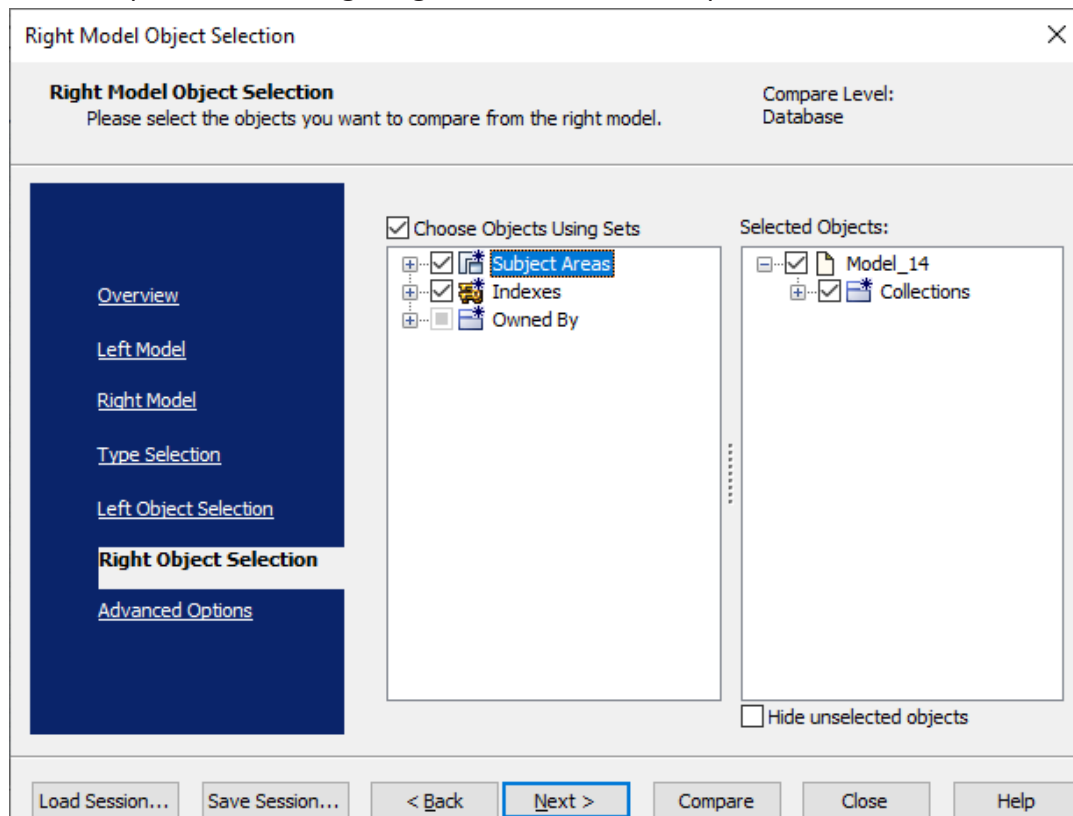
For example, the following image shows the default options.



15. Click **Next** and in the Right Object Selection section, select the appropriate options.



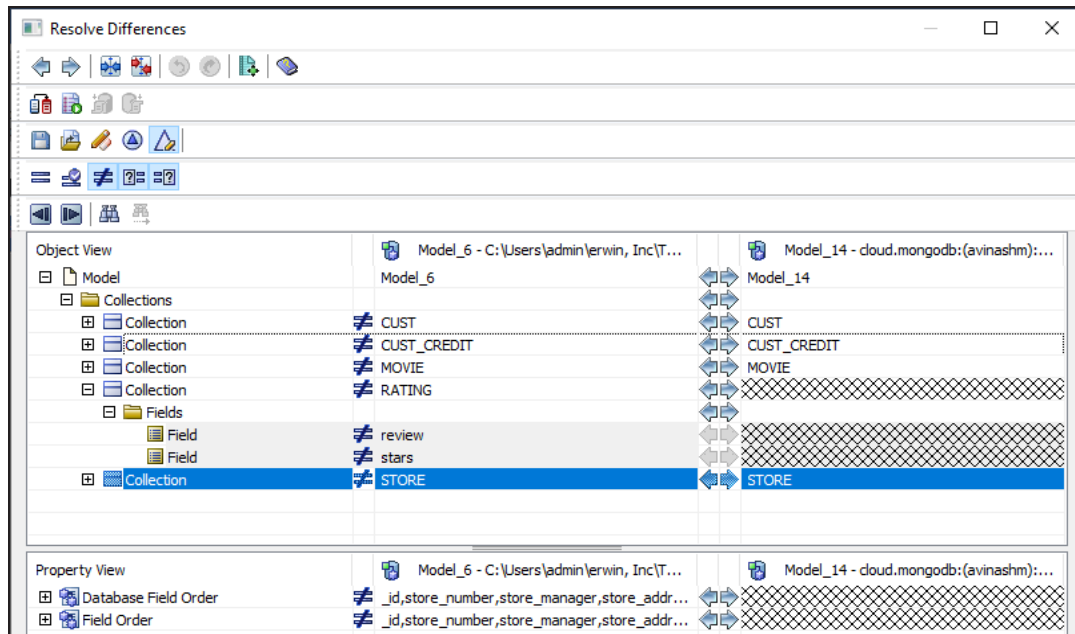
For example, the following image shows the default options.



16. Click **Compare**.

The comparison process runs, and the Resolve Differences dialog box appears. It displays the differences between your model and database.

For example, the following image shows that the Rating collection is available in your model but not in the database.



Select the Rating collection and click . This will move the Rating collection to the right model (from the database). Similarly, resolve other differences.

17. As differences were moved to the right model, click . This launches the Forward Engineering Alter Script Generation Wizard.

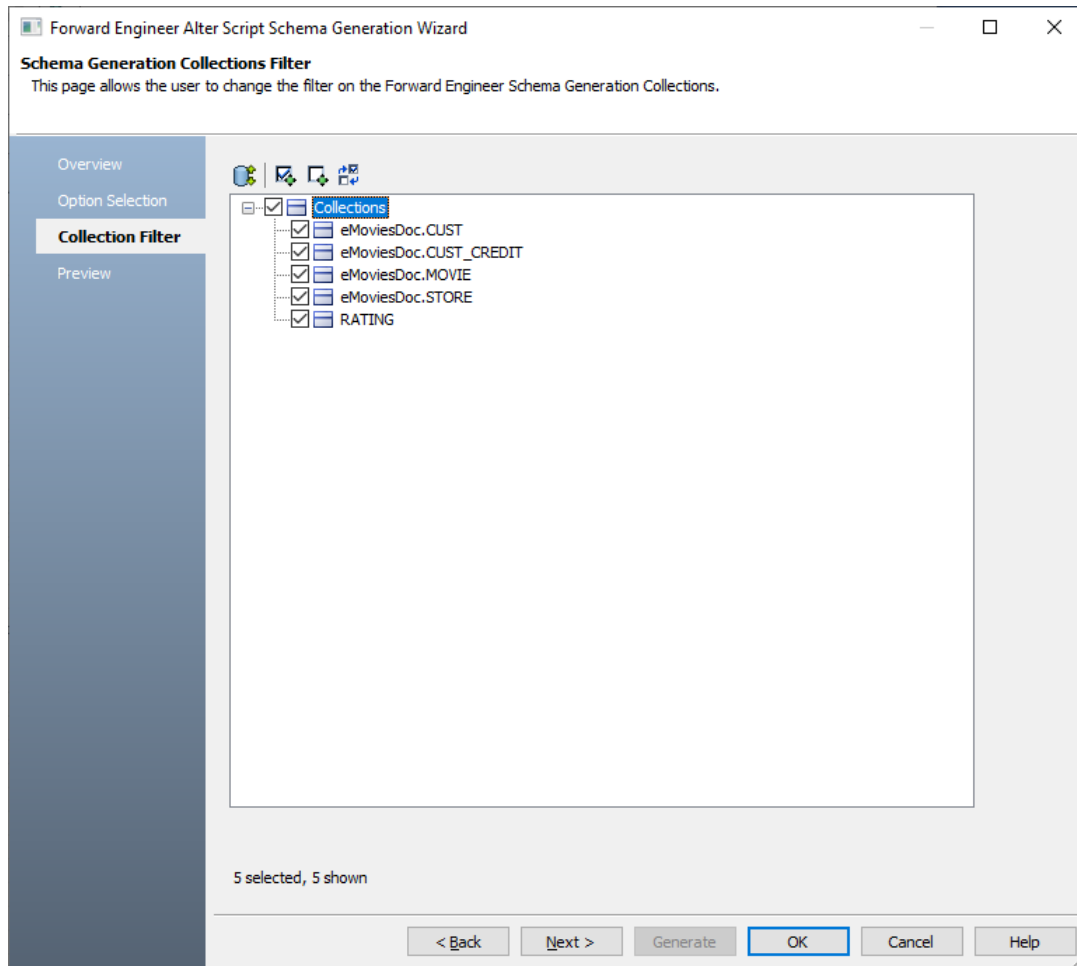
18. Click **Option Selection** and clear all the **Drop** check boxes.

The screenshot shows the 'Forward Engineer Alter Script Schema Generation Wizard' window, specifically the 'Schema Generation Options' tab. The left sidebar has four items: 'Overview', 'Option Selection' (which is highlighted), 'Collection Filter', and 'Preview'. The main area of the dialog is titled 'Schema Generation Options' and includes a subtitle: 'This page allows the user to change the Forward Engineer Schema Generation Options.' Below this, there are several sections for configuring options:

- Option Set:** A dropdown menu showing 'Select the options for DB Sync'. To its right are buttons for 'Open...', 'Save', and 'Save As...'.
- Database Template:** A text field containing 'MongoDB.fet'. To its right are buttons for 'Browse...', 'Edit...', and 'Reset'.
- General Syntax Option:** A section containing three checkboxes: 'Use DB' (unchecked), 'Comments' (unchecked), and 'Blank Value' (checked).
- Collection Syntax Option:** A section containing four checkboxes: 'Create' (checked), 'Drop' (unchecked), 'Insert' (checked), and 'Schema Validation' (unchecked).
- View Syntax Option:** A section containing two checkboxes: 'Create' (checked) and 'Drop' (unchecked).
- Index Syntax Option:** A section containing two checkboxes: 'Create' (checked) and 'Drop' (unchecked).
- User Syntax Option:** A section containing two checkboxes: 'Create' (unchecked) and 'Drop' (unchecked).

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Generate', 'OK' (which is highlighted with a blue border), 'Cancel', and 'Help'.

19. Click **Collection Filter** and select or verify the collections to be included on the forward engineering script.



20. Click **Preview** to view and verify the alter script.
21. Click **Generate** and connect to your MongoDB database.  
The forward engineering process starts. The script generates your physical database schema. You can access your database and verify the newly generated schema.
22. Click **OK**. Then click **Finish**.  
This closes the Resolve Differences dialog box and displays the Complete Compare wizard.
23. Click **Close**.



## JSON and AVRO Support

erwin Data Modeler (DM) now includes modeling support for [JSON](#) and [AVRO](#) file formats. The following table lists the supported objects and data types for each format:

File Format	Objects	Data Types
JSON	<ul style="list-style-type: none"><li>▪ JSON Objects</li><li>▪ Fields</li><li>▪ Relationships</li></ul>	<ul style="list-style-type: none"><li>▪ Object</li><li>▪ Array</li><li>▪ Integer</li><li>▪ Null</li><li>▪ String</li><li>▪ Number</li><li>▪ Boolean</li></ul>
AVRO	<ul style="list-style-type: none"><li>▪ Records</li><li>▪ Fields</li><li>▪ Relationships</li></ul>	<ul style="list-style-type: none"><li>▪ Array</li><li>▪ Boolean</li><li>▪ Union</li><li>▪ Map</li><li>▪ int</li><li>▪ Double</li><li>▪ Object</li><li>▪ String</li><li>▪ Byte</li><li>▪ enum</li><li>▪ Fixed</li><li>▪ Long</li></ul>

Similar to relational or NoSQL databases, JSON and AVRO as target databases support:

- [Reverse engineering models from scripts](#)
- [Forward engineering models](#)

## Reverse Engineering Models - JSON and AVRO

You can create a data model from JSON and AVRO scripts using the Reverse Engineering process.

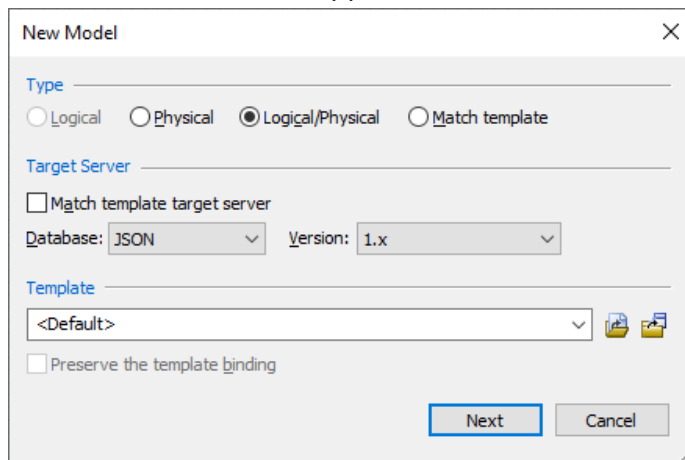
**Note:** For reverse engineering German language JSON scripts, ensure the script Encoding is set to Convert to ANSI.

This topic walks you through the steps to reverse engineer a JSON model from a script file. Similarly, you can reverse engineer a model from your AVRO script file.

To reverse engineer a model:

1. In erwin Data Modeler (DM), click **Actions > Reverse Engineer**.

The New Model screen appears.

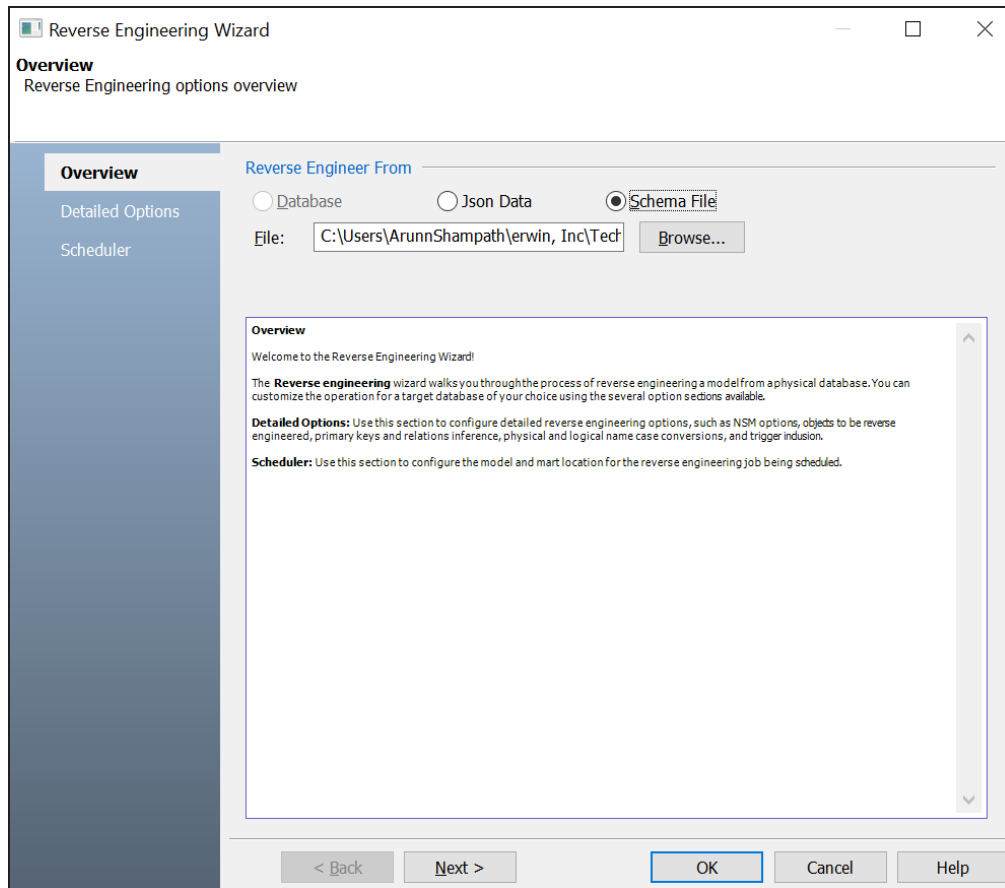


The screenshot shows the 'New Model' dialog box with the following settings:

- Type:** Logical (unselected), Physical (unselected), Logical/Physical (selected), Match template (unselected).
- Target Server:** Match template target server (checked).
- Database:** JSON (selected in dropdown).
- Version:** 1.x (selected in dropdown).
- Template:** <Default> (selected in dropdown).
- Buttons:** Next (highlighted with a blue border), Cancel.

2. Click **Logical/Physical** and set **Database** to JSON.
3. Click **Next**.

The Reverse Engineer Process Wizard appears.



4. Select **Json Data** or **Schema File** format option. Then, click **Browse** and select one or multiple script files.
5. Click **Next**.  
The Detail Options section appears. Set up appropriate options based on your requirement.



Reverse Engineer Process Wizard

**Reverse Engineer Detail Options**  
This page provides a Detail Options of the Reverse Engineer Process.

Overview  
**Detail Options**  
Scheduler

NSM Options  
Glossary CSV File:

Reverse Engineer

☐ System Objects

Records/Views Owned By  
☒ All ☐ Current User  
☐ Owners (comma separated):

Infer  
☐ Primary Keys ☐ Relations  
From  
☐ Indexes ☐ Names

Case Conversion of Physical Names  
☒ None ☐ lower ☐ UPPER ☐ Force

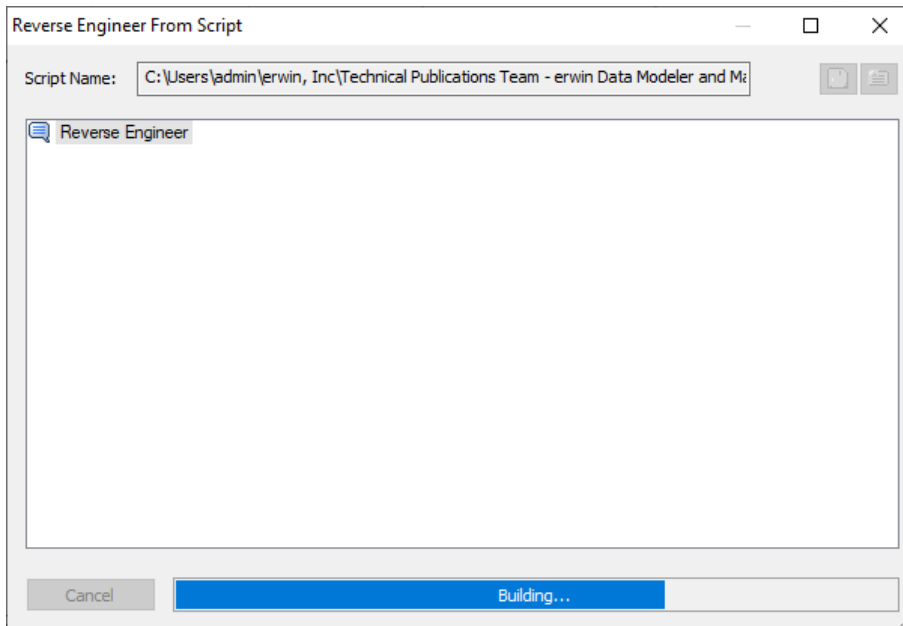
Case Conversion of Logical Names  
☒ None ☐ lower ☐ UPPER ☐ Mixed

☒ Include Generated Triggers

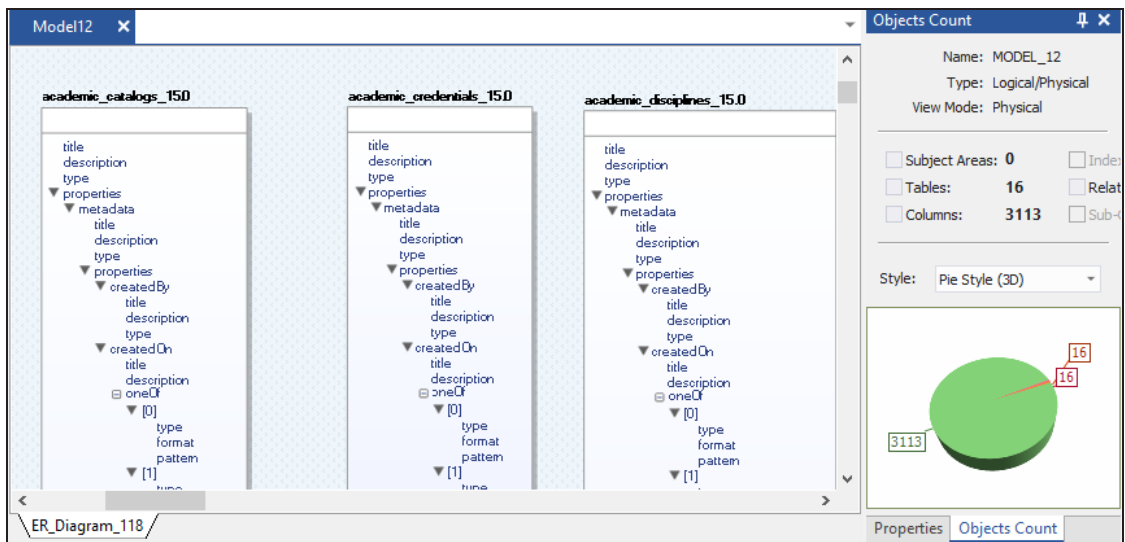
< Back Next > OK Cancel Help

6. Click **OK**.

The reverse engineering process starts.



Once the process is complete, based on your selections, a schema is generated, and a model is created.



## Forward Engineering Models - JSON and AVRO

You can generate a physical schema from a physical model using the Forward Engineering process and then, save it in the JSON and AVRO file formats.

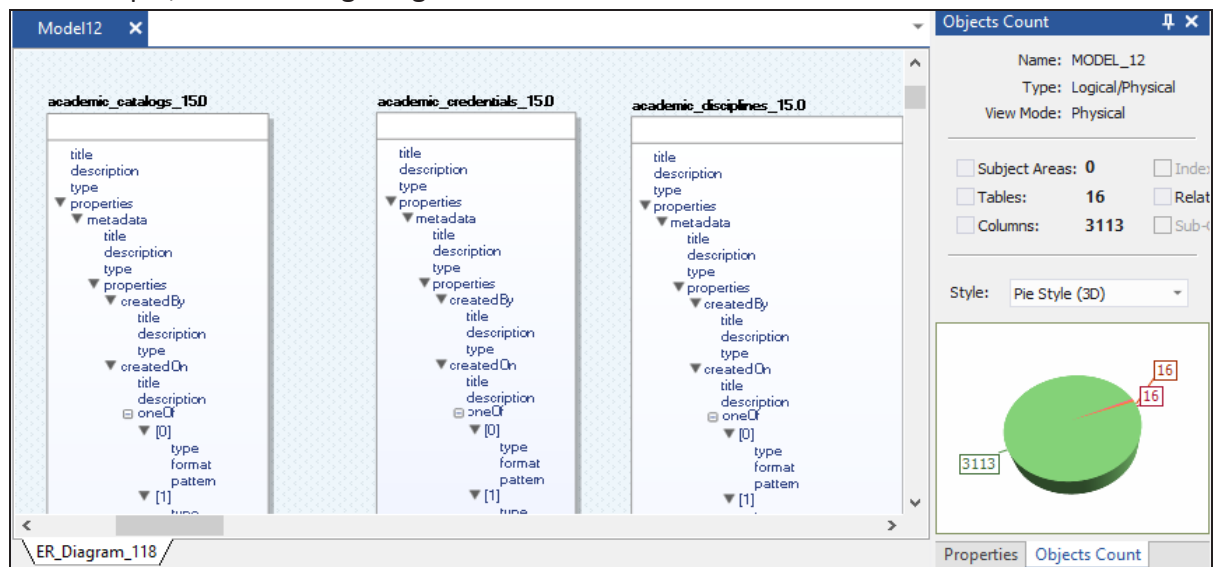
This topic walks you through the steps to forward engineer a JSON model. Similarly, you can forward engineer an AVRO model.

To forward engineer a model:

1. Open your JSON model in erwin Data Modeler (DM).

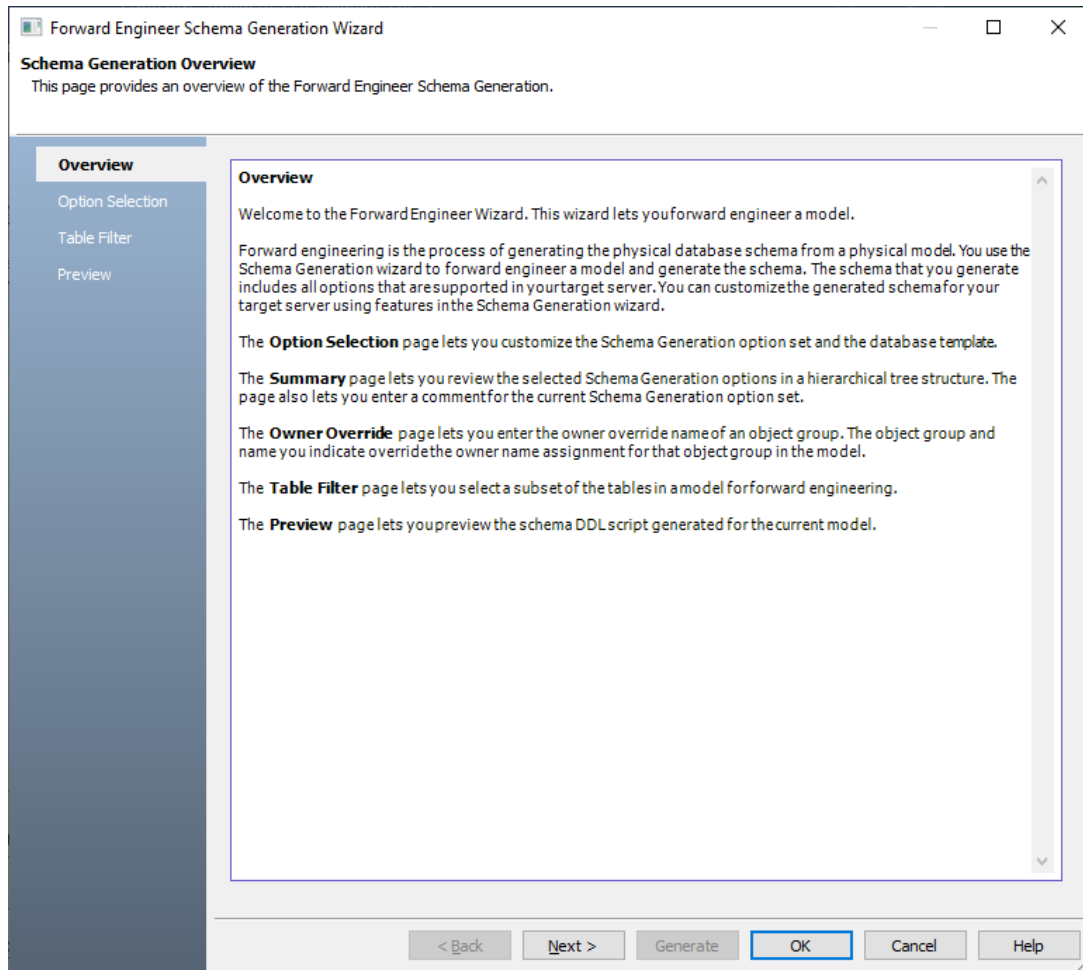
**Note:** Ensure that you are in the Physical mode.

For example, the following image uses a JSON model with 16 tables.



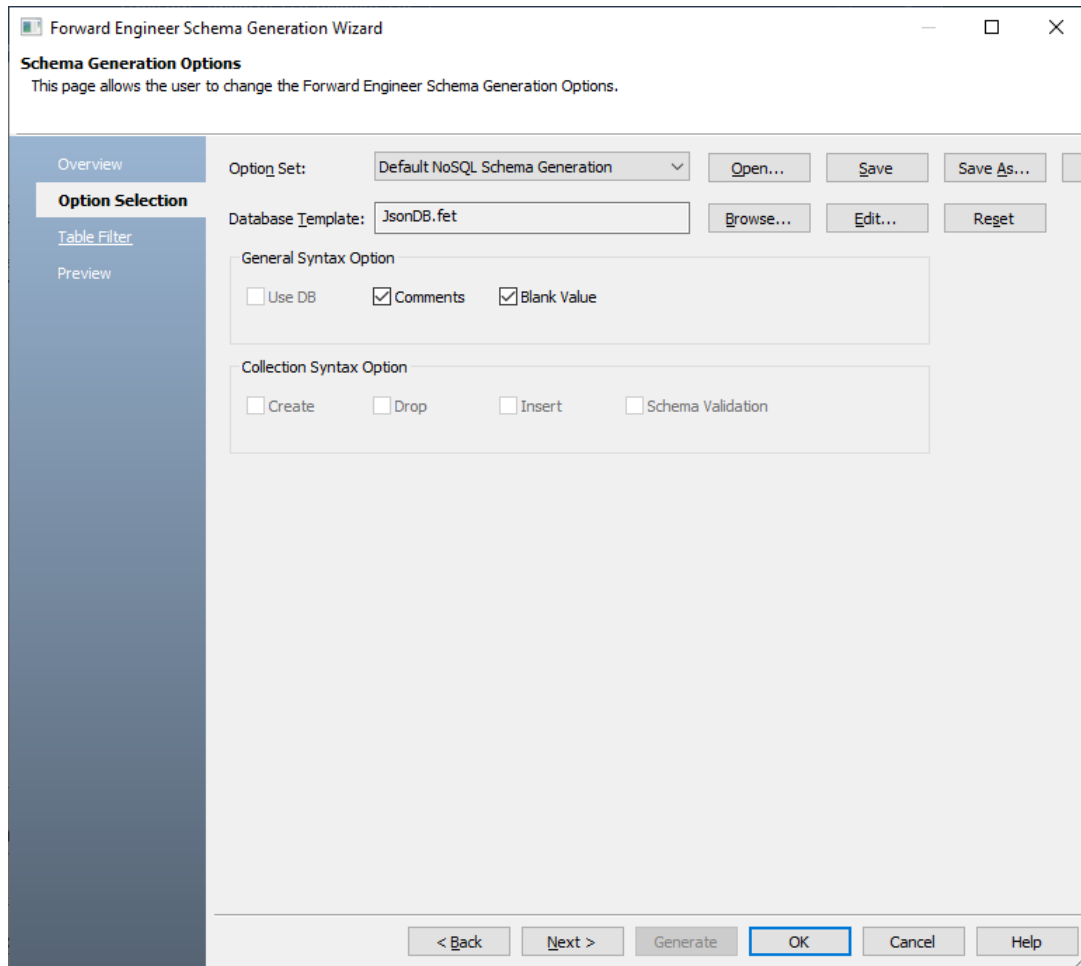
2. Click **Actions > Schema**.

The Forward Engineer Schema Generation Wizard appears.



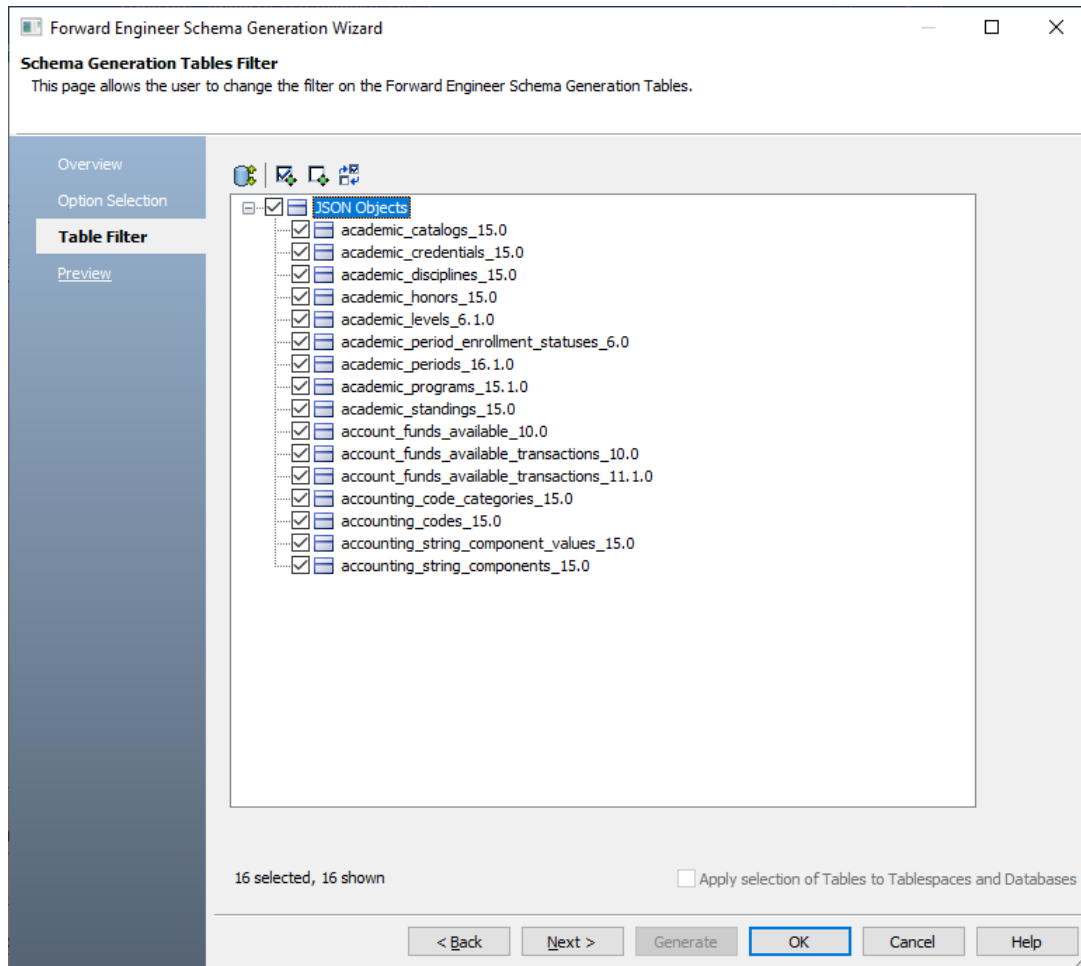
3. Click **Option Selection**.

The Option Selection section displays the default option set. Select appropriate syntax options.



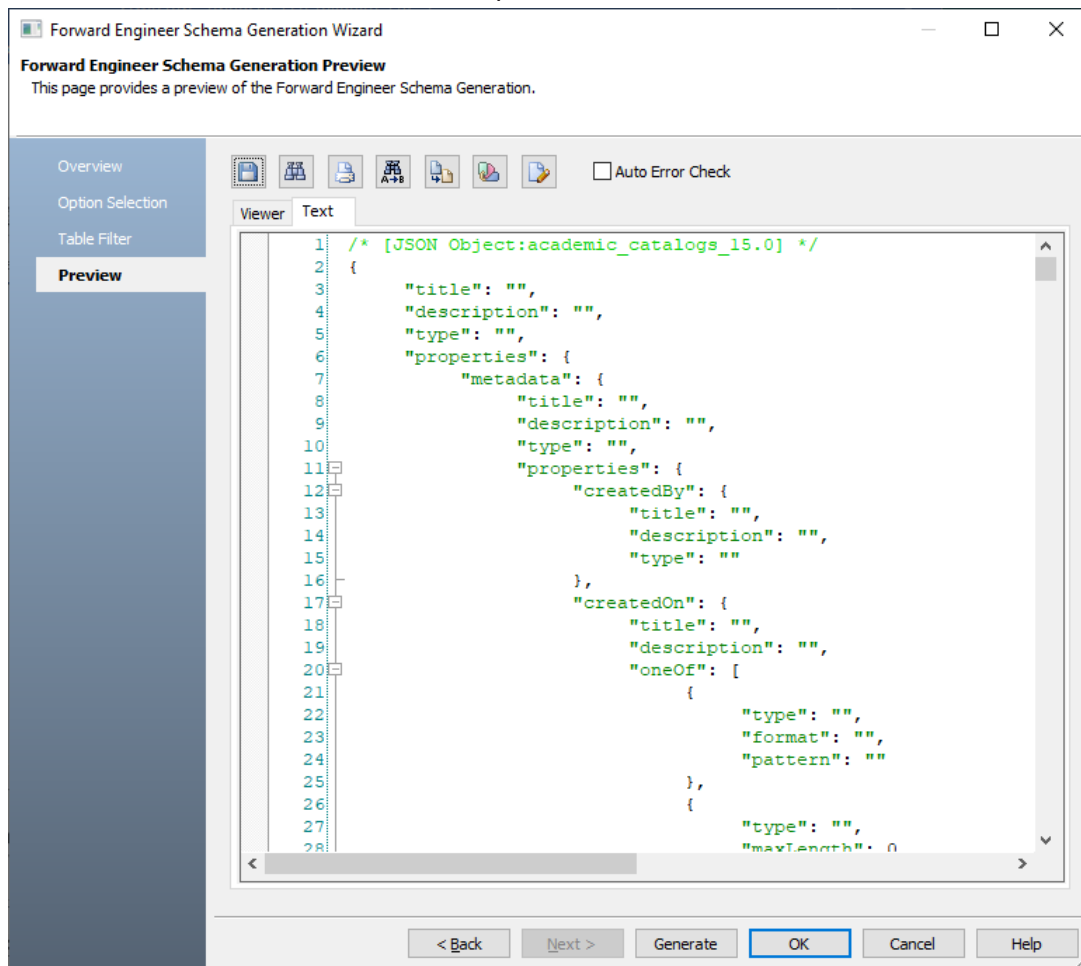
4. Click **Next**.

The Table Filter section appears. It displays a list of tables (JSON objects) available in your model.




5. Select the tables (JSON objects) that you want to forward engineer.

6. Click **Preview** to view the schema script.



Use the following options:

- **Auto Error Check:** Select this option to enable auto error check by the forward engineering wizard.
- **Error Check** (

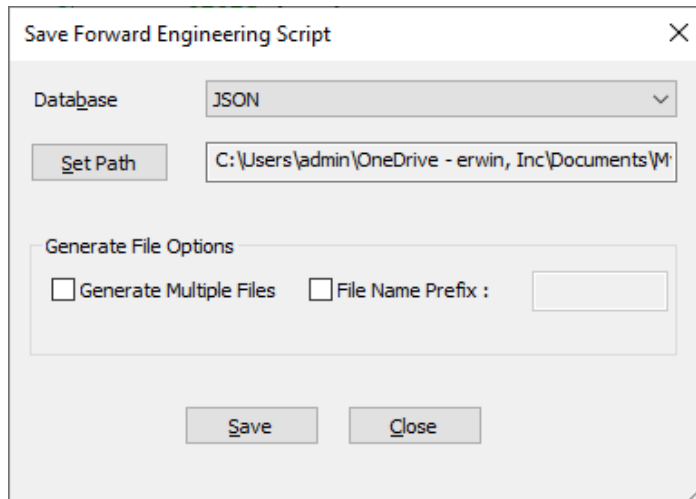
63

refer to the Forward Engineering Wizard - Preview Editor topic.

- **Save** (📁): Use this option to save the generated script.

7. Click **Generate**.

The following screen appears.



The screenshot shows a dialog box titled "Save Forward Engineering Script" with a close button (X) in the top right corner. Inside the dialog, there is a "Database" dropdown menu set to "JSON". Below it is a "Set Path" button and a text field containing the path "C:\Users\admin\OneDrive - erwin, Inc\Documents\M". Underneath these is a section titled "Generate File Options" which contains two checkboxes: "Generate Multiple Files" and "File Name Prefix :". The "File Name Prefix" checkbox is followed by an empty text input field. At the bottom of the dialog are two buttons: "Save" and "Close".

8. Use the following options:

- **Set Path:** Use this option to set the location to save the script file.
- **Generate Multiple Files:** By default, a single script file is created. Select this option to save the script into multiple files by objects.
- **File Name Prefix:** Select this option to add a script file name. Enter a file name. If this option is not selected, the script file is saved with a default name (Erwin\_FE\_Script.json).

9. Click **Save**.

Your script file is saved at the configured location. You can open it in any text editor and verify.



## Oracle Support Summary

erwin Data Modeler (DM) now supports [Oracle 12c R2, 18c, 19c, and 21c](#) as target databases. This implementation supports the following objects:

- Cluster
- Column
- Comment
- Context
- Database
- Database Link
- Directory
- Disk Group
- Function
- Index Editor for Clusters
- Index Editor for Materialized Views
- Index Editor for Tables
- Library
- Materialized Views
- Materialized View Log
- Package
- Package Body
- Pre and Post Scripts
- Rollback Segment
- Sequence
- Stored Procedure
- Synonym

- Table
- Tablespace
- Tablespace Group
- Trigger
- Views

The following table lists the supported data types:

Numeric	String Literals	Date and Time	Other
<ul style="list-style-type: none"> <li>• BINARY_DOUBLE</li> <li>• BINARY_FLOAT</li> <li>• DEC</li> <li>• DEC()</li> <li>• DEC(,)</li> <li>• DECIMAL</li> <li>• DECIMAL()</li> <li>• DECIMAL(,)</li> <li>• DOUBLE PRECISION</li> <li>• REAL</li> <li>• FLOAT</li> <li>• FLOAT()</li> <li>• INT</li> <li>• INTEGER</li> <li>• NUMBER</li> </ul>	<ul style="list-style-type: none"> <li>• CHAR</li> <li>• CHAR()</li> <li>• CHARVARYING()</li> <li>• CHARACTER</li> <li>• CHARACTER()</li> <li>• CHARACTERVARYING()</li> <li>• CLOB</li> <li>• DBURITYPE</li> <li>• URITYPE</li> <li>• HTTPURITYPE</li> <li>• JSON</li> <li>• JSON()</li> <li>• LONG</li> <li>• NATIONAL CHAR</li> <li>• NATIONAL CHAR VARYING()</li> <li>• NATIONAL CHAR()</li> </ul>	<ul style="list-style-type: none"> <li>• DATE</li> <li>• INTERVAL DAY TO SECOND</li> <li>• INTERVAL YEAR TO MONTH</li> <li>• TIMESTAMP</li> <li>• TIMESTAMP WITH LOCAL TIMEZONE</li> <li>• TIMESTAMP WITH TIMEZONE</li> <li>• TIMESTAMP()</li> <li>• TIMESTAMP() WITH LOCAL TIMEZONE</li> <li>• TIMESTAMP() WITH</li> </ul>	<ul style="list-style-type: none"> <li>• JSON*</li> <li>• ANYDATA</li> <li>• ANYDATASET</li> <li>• ANYTYPE</li> <li>• BFILE</li> <li>• BLOB</li> <li>• LONGRAW</li> <li>• ORDAUDIO</li> <li>• ORDDICOM</li> <li>• ORDDOC</li> <li>• ORDIMAGE</li> <li>• ORDVIDEO</li> <li>• RAW()</li> <li>• SDO_GEOMETRY</li> <li>• SDO_GEORASTER</li> <li>• SI_AVERAGECOLOR</li> <li>• SI_COLOR</li> </ul>

<ul style="list-style-type: none"> <li>• NUMBER()</li> <li>• NUMBER(,)</li> <li>• NUMERIC</li> <li>• NUMERIC()</li> <li>• NUMERIC (,)</li> <li>• ROWID</li> <li>• SMALLINT</li> </ul>	<ul style="list-style-type: none"> <li>• NATIONAL CHARACTER</li> <li>• NATIONAL CHARACTER VARYING()</li> <li>• NATIONAL CHARACTER()</li> <li>• NCHAR</li> <li>• NCHAR VARYING()</li> <li>• NCHAR()</li> <li>• NCLOB</li> <li>• NVARCHAR2()</li> <li>• UROWID</li> <li>• UROWID()</li> <li>• VARCHAR()</li> <li>• VARCHAR2()</li> <li>• XDBURITYPE</li> <li>• XMLTYPE</li> </ul>	TIMEZONE	<ul style="list-style-type: none"> <li>• SI_COLORHISTOGRAM</li> <li>• SI_FEATURELIST</li> <li>• SI_POSITIONALCOLOR</li> <li>• SI_STILLIMAGE</li> <li>• SI_TEXTURE</li> </ul>
---	---	----------	--

\*This datatype is supported only for Oracle 21c.

## Microsoft SQL Server Support

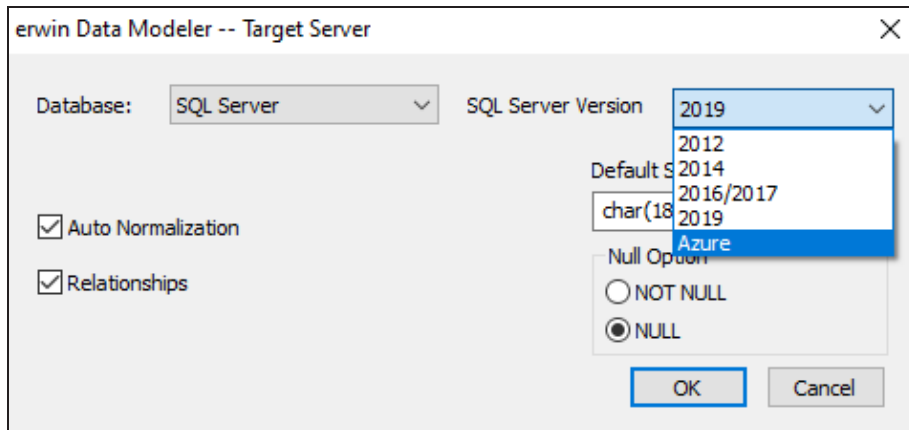
Support for Microsoft SQL Server 2019 as a target database has been enhanced to implement the following objects:

- External Library
- External Language
- External Data Source
- External File Format
- External Table
- Statistics

For detailed information on supported objects and data types, refer [SQL Server support summary](#).

## Microsoft Azure SQL Server Support

Microsoft Azure SQL support in erwin DM has been revamped. It is now supported on top of Microsoft SQL Server to leverage common functionality. On the New Model and Target Database dialog boxes, you can find Azure under SQL Server Version drop-down list.



The following table lists the supported objects:

Supported Objects	
<ul style="list-style-type: none"><li>• Always Encrypted Keys</li><li>• Application Roles</li><li>• Assemblies*</li><li>• Asymmetric Keys</li><li>• Certificates</li><li>• Credentials*</li><li>• Database Roles</li><li>• Databases</li><li>• Database Triggers</li><li>• External Data</li></ul>	<ul style="list-style-type: none"><li>• Resource Pools*</li><li>• Schemas</li><li>• Sequences</li><li>• Server Audits*</li><li>• Server Audit Specification*</li><li>• Statistics</li><li>• Stored Procedures</li><li>• Symmetric Keys</li><li>• Spatial Indexes (Table)</li><li>• Synonyms</li></ul>

Source	<ul style="list-style-type: none"> <li>• Tables</li> </ul>
<ul style="list-style-type: none"> <li>• External File Format*</li> </ul>	<ul style="list-style-type: none"> <li>• ColumnStore Indexes (Table)</li> </ul>
<ul style="list-style-type: none"> <li>• External Library*</li> </ul>	<ul style="list-style-type: none"> <li>• XML Indexes (Table)</li> </ul>
<ul style="list-style-type: none"> <li>• External Table</li> </ul>	<ul style="list-style-type: none"> <li>• Indexes (Table)</li> </ul>
<ul style="list-style-type: none"> <li>• Full-Text Catalogs</li> </ul>	<ul style="list-style-type: none"> <li>• Table Triggers</li> </ul>
<ul style="list-style-type: none"> <li>• Full-Text Indexes (Table)</li> </ul>	<ul style="list-style-type: none"> <li>• Triggers</li> </ul>
<ul style="list-style-type: none"> <li>• Full-Text Stoplists</li> </ul>	<ul style="list-style-type: none"> <li>• User Ids</li> </ul>
<ul style="list-style-type: none"> <li>• Functions</li> </ul>	<ul style="list-style-type: none"> <li>• Views</li> </ul>
<ul style="list-style-type: none"> <li>• Logins</li> </ul>	<ul style="list-style-type: none"> <li>• View Indexes</li> </ul>
<ul style="list-style-type: none"> <li>• Partition Functions</li> </ul>	<ul style="list-style-type: none"> <li>• View Triggers</li> </ul>
<ul style="list-style-type: none"> <li>• Partition Schemes</li> </ul>	<ul style="list-style-type: none"> <li>• XML Schema Collections</li> </ul>

\* These objects are supported only for Azure SQL Managed Instance.

The following table lists the supported data types:

Exact Numerics	Approximate Numerics	Date and Time	Character Strings	Unicode Character Strings	Binary Strings	Geo Types	Others
<ul style="list-style-type: none"> <li>• bigint</li> <li>• numeric</li> <li>• bit</li> <li>• smallint</li> </ul>	<ul style="list-style-type: none"> <li>• float</li> <li>• real</li> </ul>	<ul style="list-style-type: none"> <li>• date</li> <li>• datetimeoffset</li> <li>• datetime2</li> </ul>	<ul style="list-style-type: none"> <li>• char</li> <li>• varchar</li> <li>• text</li> </ul>	<ul style="list-style-type: none"> <li>• nchar</li> <li>• nvarchar</li> <li>• ntext</li> </ul>	<ul style="list-style-type: none"> <li>• binary</li> <li>• varbinary</li> <li>• image</li> </ul>	<ul style="list-style-type: none"> <li>• rowversion</li> <li>• hierarchyid</li> <li>• uniqueidentifier</li> </ul>	<ul style="list-style-type: none"> <li>• CHAR</li> <li>• VARCHAR</li> <li>• CHARACTER</li> </ul>

lint • decimal • small-money • int • tiny-int • money		• small-date-time • date-time • time	xt			• sql_variant • xml • geometry • geography	TER • CHARACTER VARYING • NATIONAL CHARACTER VARYING • NATIONAL CHARACTER • NATIONAL CHARACTER • NATIONAL TEXT
--	--	--	----	--	--	---	---

## MySQL Support

erwin Data Modeler (DM) now supports [MySQL 8.x](#) as a target database. This implementation supports the following objects:

- Database
- Event
- Function
- Function\_UDF
- Logfile Group
- Server
- Spatial Ref System
- Stored Procedure
- Table
  - Index
  - Table Column
- Tablespace
- Trigger
- User ID
- Validation Rule
- View
  - View Column

The following table lists the supported data types:

Numeric	String Literals	Date and Time	Other
<ul style="list-style-type: none"><li>• TINYINT</li><li>• SMALLINT</li><li>• MEDIUMINT</li><li>• INT,</li><li>• INTEGER</li></ul>	<ul style="list-style-type: none"><li>• CHAR</li><li>• VARCHAR</li><li>• BINARY</li><li>• CHAR BYTE</li></ul>	<ul style="list-style-type: none"><li>• DATE</li><li>• TIME</li><li>• DATETIME</li><li>• TIMESTAMP</li><li>• YEAR</li></ul>	<ul style="list-style-type: none"><li>• Geometry Type</li><li>• POINT</li><li>• LINESTRING</li><li>• POLYGON</li><li>• MULTIPOINT</li></ul>



<ul style="list-style-type: none"> <li>• BIGINT</li> <li>• DECIMAL, DEC, NUMERIC, FIXED</li> <li>• FLOAT</li> <li>• DOUBLE, DOUBLE PRECISION, REAL</li> <li>• BIT</li> </ul>	<ul style="list-style-type: none"> <li>• VARBINARY</li> <li>• TINYBLOB</li> <li>• BLOB</li> <li>• BLOB and TEXT Data Types</li> <li>• MEDIUMBLOB</li> <li>• LONGBLOB</li> <li>• TINYTEXT</li> <li>• TEXT</li> <li>• MEDIUMTEXT</li> <li>• LONGTEXT</li> <li>• JSON Data Type</li> <li>• ENUM</li> <li>• Set Data Type</li> </ul>		<ul style="list-style-type: none"> <li>• MULTILINESTRING</li> <li>• MULTIPOLYGON</li> <li>• GEOMETRYCOLLECTION</li> <li>• GEOMETRY</li> </ul>
--	--	--	---

**Note:**

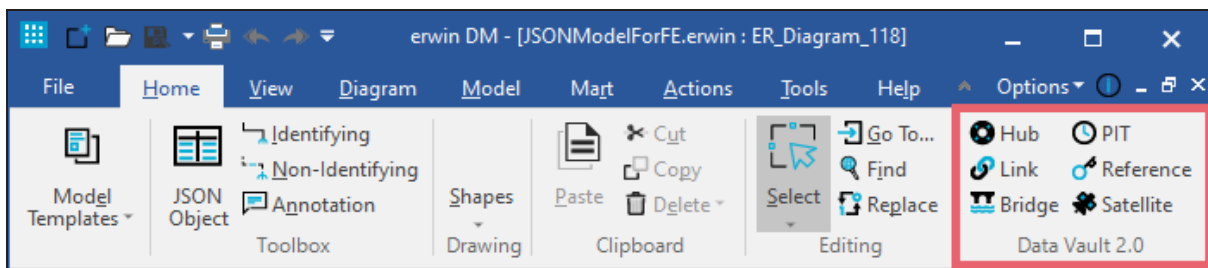
- Refer to MySQL database documentation for detailed information on specific MySQL objects and properties. erwin® Data Modeler documentation for the property editors provides brief descriptions of the controls on each dialog box and tab, which you can use as a point of reference while working with database design features.
- As a best practice, use the MySQL ANSI ODBC driver for Reverse Engineering from Database (REDB) while using erwin® Data Modeler

## Data Vault 2.0 Support

erwin Data Modeler (DM) now Data Vault 2.0 as a modeling technique across all target databases. This implementation supports the following Data Vault 2.0 components by default through API:

- Hub
- Link
- Satellite
- Reference
- PIT
- Bridge

These components are available on Home tab of the ribbon.



To enable Data Vault 2.0 on your model, follow these steps:

1. Right-click the model and click **Properties**.
2. On the **Model Editor > General** tab, select the **Data Vault 2.0** check box.

Model 'EMOVIES 2021 R1' Editor

General Defaults RI Defaults History Options Definition UDP History Notes Extended Notes

**Model Information**

Name	EMOVIES 2021 R1
Author	erwinTeam

Type: Logical / Physical Target Server And Version: SQL Server 2019

**Notation**

Logical Notation	INDFF1v
Physical Notation	INDFF1v

**Modeling Features**

Is Dimensional	<input type="checkbox"/>
Data Vault 2.0	<input type="checkbox"/>
Data Movement	<input type="checkbox"/>

Close Cancel

Details...

Once enabled, Data Vault 2.0 components are available via the Model Explorer. You can now [convert your model to a Data Vault model](#).

You can also create custom components and apply them to tables. However, these custom components do not appear on the ribbon.

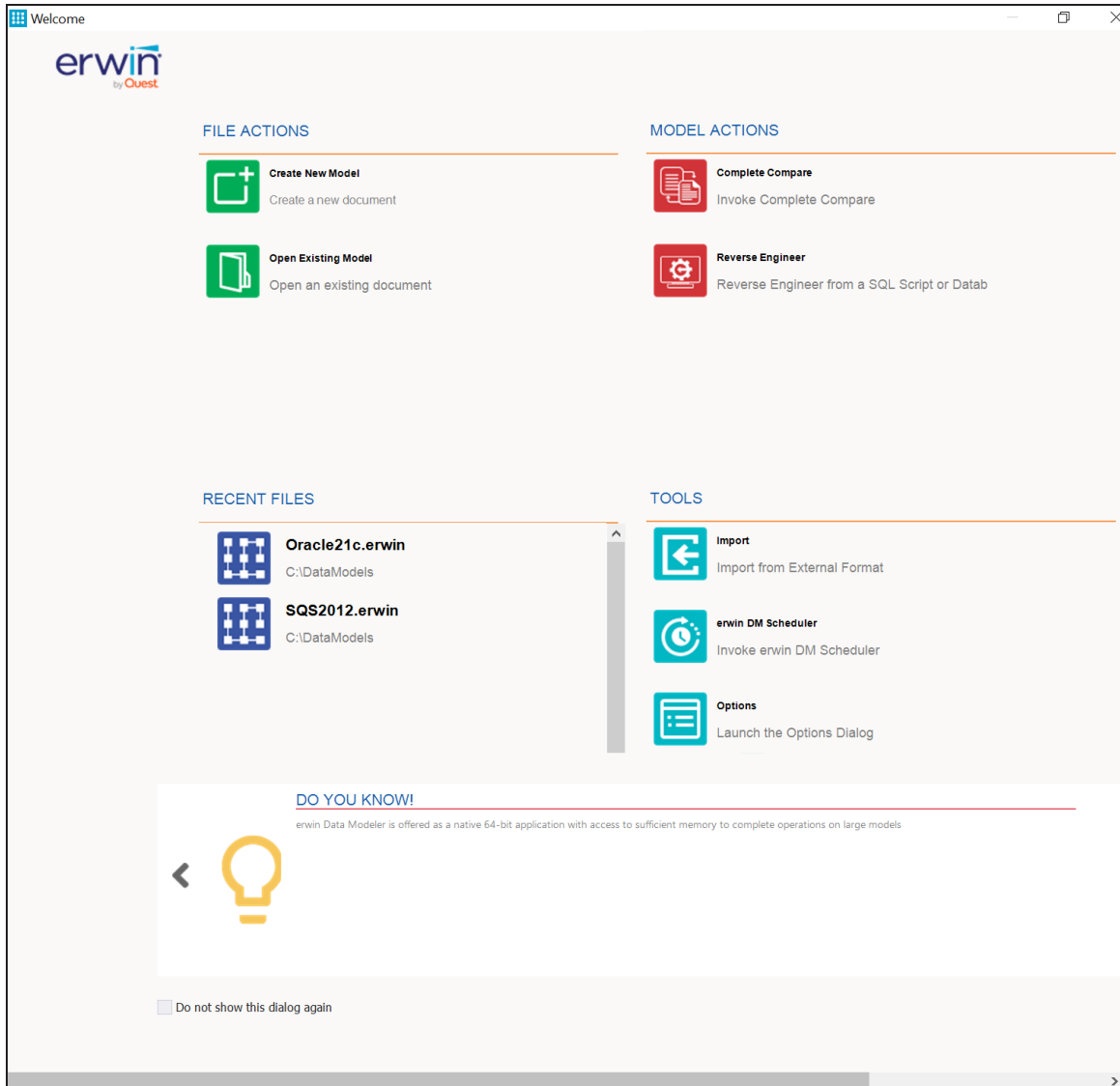
## Productivity and UI Enhancements

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

- [Welcome Page](#)
- [Objects Count Pane](#)
- [Properties Pane](#)
- [Object Browser](#)
- [Normalization and Denormalization](#)
- [Reverse Engineering and Forward Engineering Wizard Redesign](#)
- [Improved Speed Mode](#)

### Welcome Page

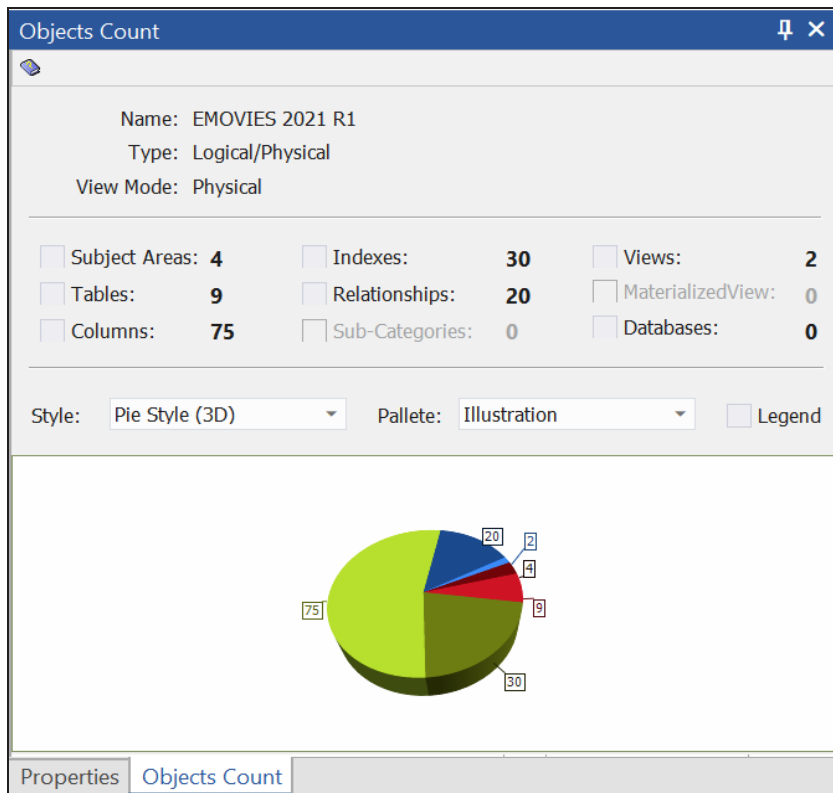
The Welcome page is a starter page that helps new users to get started with erwin DM. It appears when you launch erwin DM and is also accessible via **Help > Welcome**. It contains shortcuts to key actions that are performed frequently, such as opening models or creating new ones, running reverse engineering or complete compare wizards, and connecting to the Mart. Apart from these, the Welcome page provides access to recently used files, erwin DM Tools, Technical Support, and Help links.



## Objects Count Pane

As data models become complex and large, it becomes necessary to get a snapshot of the objects within models. To facilitate this, erwin DM now includes an Objects Count pane. This pane displays information about a selected model and a count of all the objects present in it. Also, it displays a snapshot of this information in the pictorial format, which you can cus-

customize using the Style and Palette options. By default, this pane opens on the right-side of the application. You can also access it via **View > Panes > Objects Count Pane**.



## Properties Pane

While working on data models, accessing the property editors to view or edit the model and its object's properties can get tedious and slow you down. To address this, erwin DM now includes a Properties pane. This pane enables you to view and edit the selected object's properties along with the model diagram, side-by-side. By default, this pane opens on the right-side of the application. You can also access it via **View > Panes > Properties Pane** on the ribbon.

Properties

General

Name	EMOVIES 2021 R1
Author	erwinTeam
Definition	
Type	Logical/Physical
Logical Notation	IDEF1x
Physical Notation	IDEF1x
Current ER Diagram	Drawing Objects
Current Subject Area	
Current DSM Option Set	eMovies
Current NSM Option Set	eMovies.nsm
Allow Manual Relationship Layout	<input checked="" type="checkbox"/>
Unique Names	Not Unique
Is NSM Option From AMT	<input type="checkbox"/>

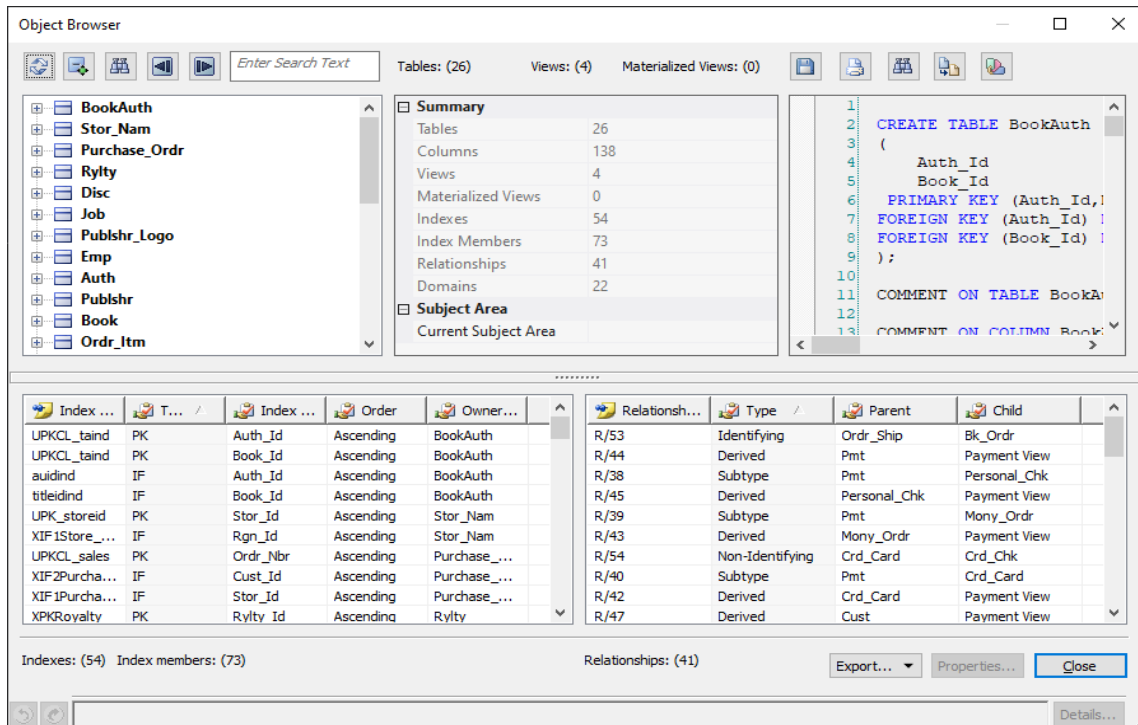
Defaults

Properties | Objects Count

## Object Browser

The Object Browser is a one-stop location where you can view tables, views, materialized views, indexes, relationships, and the complete model's or specific table's DDL. You can export this information as a report in CSV, HTML, or PDF formats. To access the Object

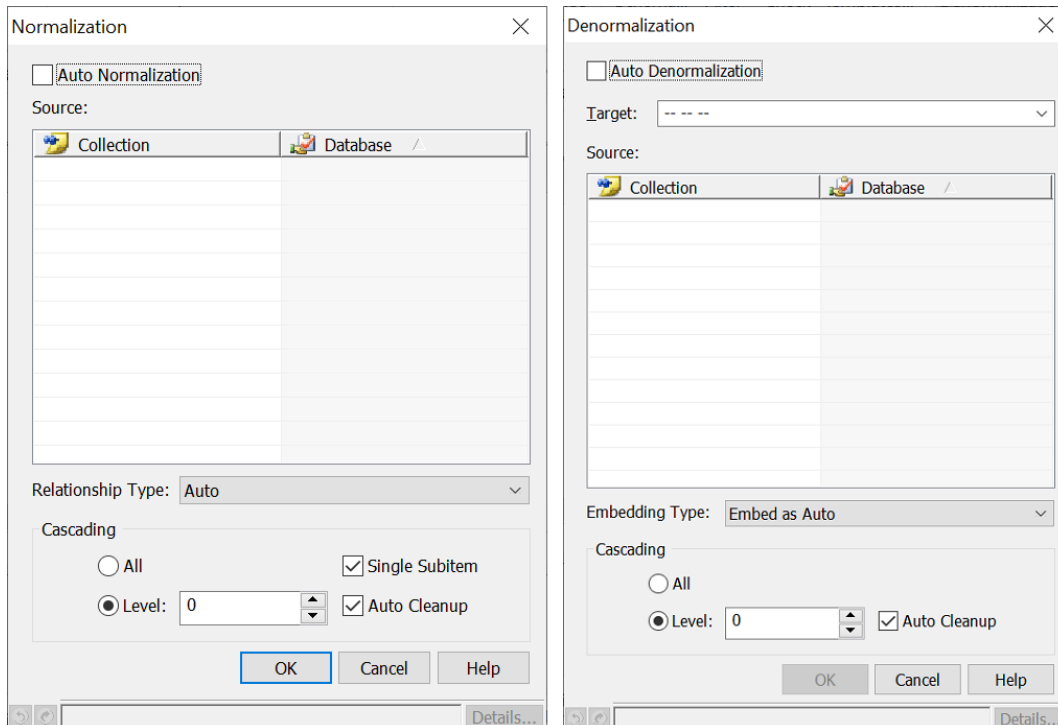
Browser, on the **Properties** pane, click  or on the ribbon, click **Tools > Object Browser**. For more information, refer to the [Object Browser](#) topic.



## Normalization and Denormalization

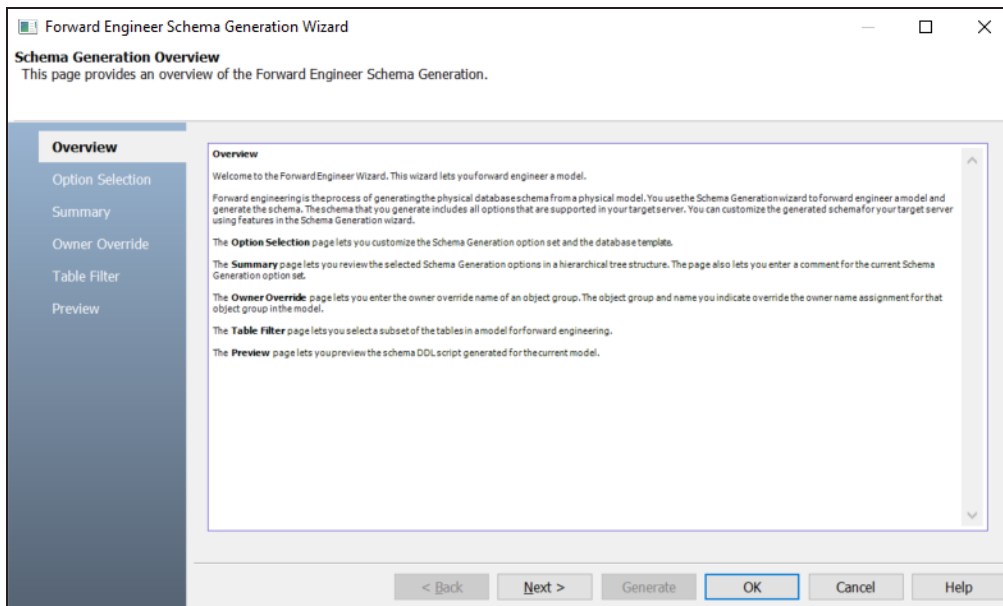
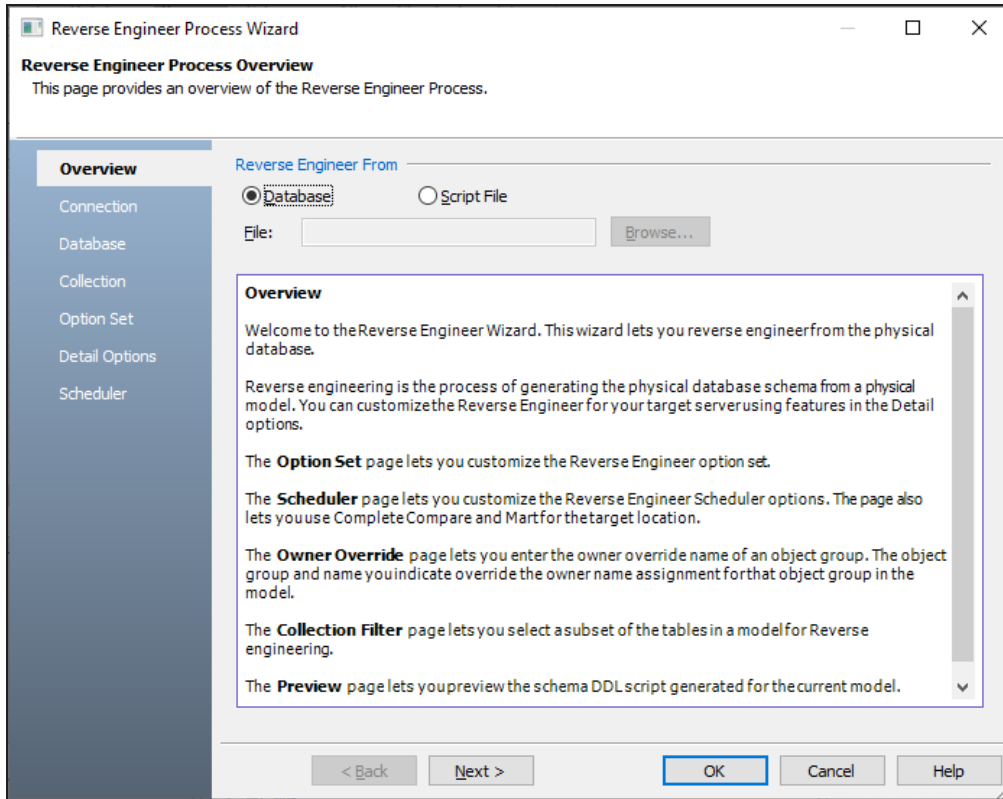
The Normalization and Denormalization features enable you to define relationships in a NoSQL model. Normalization splits the fields in a collection into multiple collections based on the selected relationship type. Whereas Denormalization embeds multiple collections into a single collection based on the selected embedding type. To access these features, on the ribbon, click **Actions**. Then, click **Normalization** or **Denormalization**. For more information, refer to the [Defining Relationships Using Embedding Method](#) topic.





## Reverse Engineering and Forward Engineering Wizard Redesign

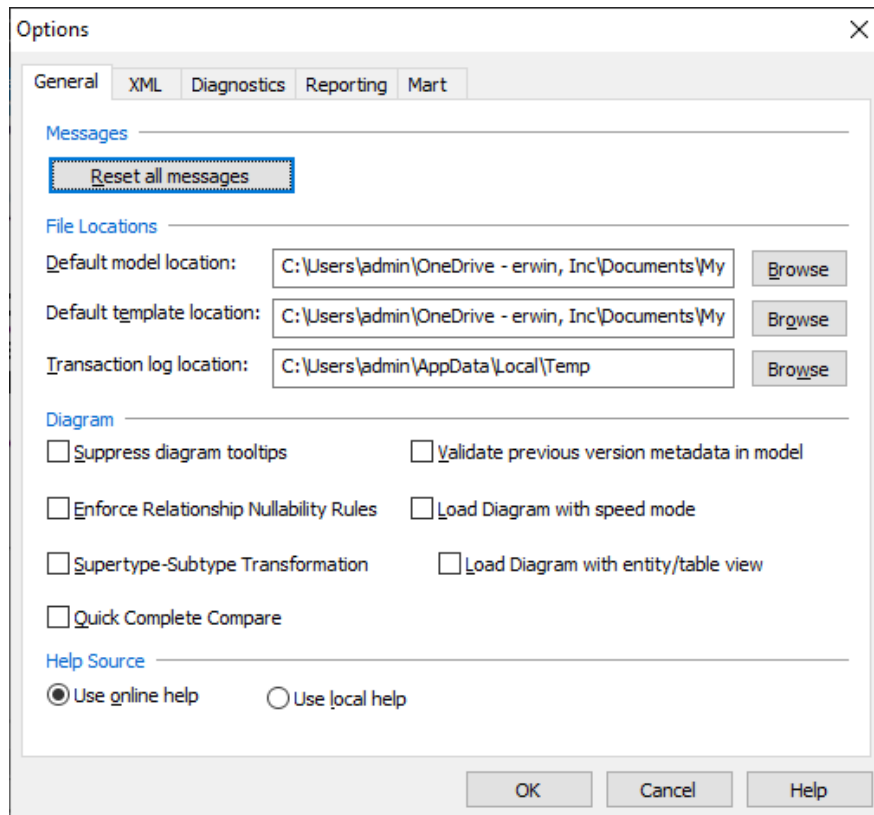
The Reverse Engineering and Forward Engineering wizards have been redesigned for better arrangement of properties and ease of use. For more information, refer to the [Reverse Engineering](#) and [Forward Engineering](#) topics.



## Improved Speed Mode

The Load Diagram with speed mode option now provides another option, Load Diagram with entity/table view. This option improves the model load performance in case of large models significantly. It does so by rendering the model in the simplest way possible, with only the entities or tables, and their relationships. Also, by default, it disables the PK-FK highlight feature (**Display Diagram Highlight** option on Model Editor).

This option is available on **Tools > Options** dialog box.



## Database Connectivity

You can now connect to the following databases from erwin Data Modeler (DM) without using a client software.

- Relational Databases:
  - Oracle
  - SQL Server
  - Azure SQL
  - Azure Synapse
  - MySQL
  - MariaDB
  - SAP ASE
  - Snowflake
- NoSQL Databases:
  - Cassandra
  - Couchbase
  - MongoDB

To connect to these databases without a client software, use the following database connection parameters:

### Oracle

#### Instance

Specifies the JDBC instance to which you want to connect.

For a cloud-based connected, the instance name is as follows:

*TNS\_ADMIN=<Path of unzipped cloud wallet file>*

For example, *TNS\_ADMIN=C:\\Users\\MyUser\\Wallet\_DBTEST*

Note: Ensure that you have downloaded, saved, and unzipped the cloud wallet file.

### Connection String

Specifies the connection string based on your JDBC instance in the following format:

*[jdbc:oracle:thin:@//<servername>:1521/](#)*

For example, *JDBC:ORACLE:thin:@//localhost:1521/*

For a cloud instance, the connection string is as follows:

*[jdbc:oracle:thin:@<dbname\\_priority>?](#)*

For example, *jdbc:oracle:thin:@dbtest\_medium?*

## SQL Server

### Connection Type

Specifies the type of connection you want to use. Select *Use Native Connection* to connect using the API provided by the SQL Server Native client software.

Select *Use ODBC Data Source* to connect using the ODBC data source that you have defined. Select *Use JDBC Connection* to connect using JDBC.

### Instance

Specifies the JDBC instance to which you want to connect.

### Database

Specifies the name of the database that you want to connect to.

### Connection String

Specifies the connection string based on your JDBC instance and SQL Server database name in the following format:

*[jdbc:sqlserver://<servername>:1433=<SqlDBname>](#)*

For example, *jdbc:sqlserver://localhost:1433*

## Azure SQL

### Connection Type

Specifies the type of connection you want to use. Select *Use Native Connection* to connect using the API provided by the SQL Server Native client software. Select *Use ODBC Data* to connect using the ODBC data source that you have defined. Select *Use JDBC Connection* to connect using JDBC.

### Instance

Specifies the JDBC instance to which you want to connect.

### Database

Specifies the name of the database that you want to connect to.

### Connection String

Specifies the connection string based on your JDBC instance and SQL Server database name in the following format:

*jdbc:sqlserver://<servername>:<port>*

For example, *jdbc:sqlserver://localhost:1433*

### Connect to Managed Instance

Specifies whether the connection should be to an Azure SQL Managed Instance.

## Azure Synapse

### Connection Method

Specifies the type of connection you want to use. Select **JDBC** to connect using JDBC.

### Connection String

Specifies the connection string based on your JDBC instance and Azure Synapse database name in the following format:

*jdbc:sqlserver://<server name>.sql.azuresynapse.net:1433;*

For example, *jdbc:sqlserver://localhost.sql.azuresynapse.net:1433;*

### Database

Specifies the name of the database that you want to connect to.

## MySQL

### Hostname/IP

Specifies the hostname or IP address of the server where your database is hosted.

### Port

Specifies the port configured for your database.

### Database

Specifies the name of the database to which you want to connect.

**Note:** For the JDBC connection to work seamlessly, ensure that you download the required JDBC driver and rename it to `mysql-connector-java-8.0.22.jar`.

## MariaDB

### Connection String

Specifies the connection string in the following format:

*`jdbc:mariadb`*

### Hostname/IP

Specifies the hostname or IP address of the server where your database is hosted.

### Port

Specifies the port configured for your database.

### Database

Specifies the name of the database to which you want to connect.

**Note:** For the JDBC connection to work seamlessly, ensure that you download the required JDBC driver and rename it to `mariadb-java-client-2.6.1.jar`.

## SAP ASE

**Connection Method**

Specifies the type of connection you want to use. Select *Use JDBC Connection* to connect using JDBC.

**Server**

Specifies the name of the server where the database is installed.

**Database**

Specifies the name of the database that you want to connect to.

**Connection String**

Specifies the connection string based on your JDBC instance and SAP ASE database name in the following format:

*jdbc:sybase:Tds:<servername>:5000*

For example, *jdbc:sybase:Tds:localhost:5000*

**Snowflake****Connection String**

Specifies the connection string based on your JDBC instance and Snowflake database name. For example, *jdbc:snowflake://<account\_name>.snowflakecomputing.com/*

Connection String is a mandatory parameter.

**Database**

Specifies the name of the database that you want to connect to.

Database is a mandatory parameter.

**Warehouse**

Specifies the name of the warehouse that you want to connect to.

Warehouse is a mandatory parameter.

**Role**

Specifies the role that you want to use to connect to the database.

Role is a mandatory parameter.



## Schema

Specifies the name of the schema that you want to connect to.

## Cassandra

### Connection Method

Specifies the type of connection you want to use. Select *Direct* to connect to connect to your cluster directly. Select *Connection String* to connect to your cluster using a connection string.

### Hostname/IP

Specifies the hostname or IP address of the server where your cluster is hosted.

### Port

Specifies the port configured for your cluster.

### Connection String

Specifies the path to the secure connect ZIP file in the following format:

*C:\<file name>.zip*

For example, *C:\TempCass\secure-connect-testdb.zip*

## Couchbase

### Connection Method

Specifies the type of connection you want to use. Select *Direct* to connect to connect to your bucket directly. Select *Connection String* to connect to your bucket using a connection string.

### Hostname/IP

Specifies the hostname or IP address of the server where your bucket is hosted.

### Port

Specifies the port configured for your bucket.

**Bucket**

Specifies the name of the bucket to which you want to connect.

**SSL Certificate Path**

Specifies the path to the SSL certificate, if you have one. You can leave this field blank.

**Connection String**

Specifies the connection string in the following format:

*couchbases://<database server>/<bucket>?ssl=no\_verify*

For example, *couchbases://server1.dp.cloud.couchbase.com/testbucket?ssl=no\_verify*

**MongoDB****Connection Method**

Specifies the type of connection you want to use. Select *Direct* to connect to connect to your database directly. Select *Connection String* to connect to your database using a connection string.

**Hostname/IP**

Specifies the hostname or IP address of the server where your database is hosted.

**Port**

Specifies the port configured for your database.

**Database**

Specifies the name of the database to which you want to connect.

**Connection String**

Specifies the connection string in the following format:

*mongodb://[username:password@]host1[:port1][,...hostN[:portN]][/[defaultau-  
thdb][?options]]*

For example, *mongodb+srv://myusername:\*\*\*\**

*@cluster0.v7gra.mongodb.net/test?retryWrites=true&w=majority*

For more information on database connection parameters, refer to the [Database Connection Parameters](#) topic.