

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.

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 <u>erwin support</u> 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 <u>erwin Data Modeler</u>, 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 <u>http://erwin.com/products</u>.

Provide Feedback

If you have comments or questions, or feedback about erwin product documentation, you can send a message to <u>techpubs@erwin.com</u>.

erwin Data Modeler News and Events

Visit <u>www.erwin.com</u> 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

Legal Notices	
Contents	5
Introduction	7
erwin Mart Administrator UI Facelift	
NoSQL Modeling	9
MongoDB Support	
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	
Reverse Engineering Models	23
Forward Engineering Models	
Comparing Changes using Complete Compare	
JSON and AVRO Support	
Reverse Engineering Models - JSON and AVRO	
Forward Engineering Models - JSON and AVRO	
Oracle Support Summary	
Microsoft SQL Server Support	
Microsoft Azure SQL Server Support	
MySQL Support	
Data Vault 2.0 Support	74

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

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 <u>MongoDB 4.x</u> 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 Lit- erals	Date and Time	Other
double	 string 	• date	 object
 binary 		 timestamp 	 array
• int			• null
 integer 			 objectId
 boolean 			 regex
• minKey			• code
• maxKey			
 long 			
decimal			

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 	

	• inet	• string
	• int	 integer
	• smallint	 integer
	• text	• string
	• time	• integer, string
	• timestamp	• integer, string
	• timeuuid	• uuid
	• tinyint	 integer
	• uuid	• uuid
	• varint	 integer
Collection	• list	
	• map	
	• set	
Tuple	• tuple	

Couchbase Support

erwin Data Modeler (DM) now supports <u>Couchbase 6.x</u> 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:

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.

deler Target Server				\times
SQL Server	~	SQL Server Version	2016/2017	~
		Default	t SQL Server Datatype	
		char(18)	~
		Defa	ult Non-Key Null Option <u>O</u> T NULL	
			JLL	
			OK Cancel	
	deler Target Server	deler Target Server	deler Target Server SQL Server Version Default Char(Defa O NG O NG O NG	deler Target Server SQL Server Version 2016/2017 Default SQL Server Datatype char(18) Char(18) Default Non-Key Null Option NQT NULL NULL OK Cancel

The erwin Data Modeler -- Target Server screen appears.

3. In the **Database** drop-down list, select MongoDB.

By default, the Auto Denormalization check box is selected. Keep it selected.

erwin Data Mo	odeler Target Server			\times
Database:	MongoDB v	MongoDB Version	4.x	
🗹 Auto Den	ormalization	Default N String Default	NongoDB Datatype Non-Key Null Option T NULL	~
			OK Cancel	

4. Click **OK**.

The conversion process starts.

Embedding	x
Processing Data : (6) / (24) - (25)%	
Cancel	

Once the conversion is complete, the existing model in 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:

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.



 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 o	create a new derived model Compare Level: Unknown
	New Model Type O Logical O Physical O Logical/Physical
<u>Overview</u>	Create Using Template:
Source Model	Blank Logical/Physical Model
Target Model	
Type Selection	
Object Selection	Remove Browse File System Browse Mart
<u>Naming Standards</u>	Creates a new model with both logical and physical levels (erwin DM classic) and default settings.
	Target Database
	Database: SQL Server \checkmark Version: 2016/2017 \checkmark
	Auto Denormalization Auto Normalization Relationships
	< Back Next > Derive Close Help

3. 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 c	reate a new derived model Compare Level: Unknown
	New Model Type O Logical O Physical O Logical/Physical
<u>Overview</u>	Create Using Template:
Source Model	Blank Logical/Physical Model
Target Model	
Type Selection	
Object Selection	Remove Browse File System Browse Mart
<u>Naming Standards</u>	Creates a new model with both logical and physical levels (erwin DM classic) and default settings.
	Target Database
	Database: MongoDB V Version: 4.x V
	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.

Derive Model				
Derive Model Type Selection Please select the types of obje	ects to be Derived	Compar Logical i	e Level: and Physical	
	Filter Level	Physical Level	Data	base Level
<u>Overview</u>	Option Set: Advanced	Default Option Set	~ 🗁	0 🎽 🗍
<u>Source Model</u> <u>Target Model</u>	⊡	ies tion Defaults		^
Type Selection Object Selection	■ ✓ III Data Mu ■ ✓ III Data So ■ ✓ III Data So ■ ✓ Ø Data Va	ovement Rule ource Object ault 2.0 wilt Attributes		
<u>Naming Standards</u>	Data Va D	our Attributes pe Standard Trigger Template Value Table		
		ed Notes		*
	< <u>B</u> ack <u>N</u> ext	: > Derive	Close	Help

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.

Derive Model					
Source Model Object Selection Compare Level: Please select the objects you want to linked to the derived model Logical and Physical					
Overview Source Model Target Model Type Selection Object Selection Naming Standards	Choose Objects Using Sets		Selected Objects:	2021 R1 ations ype Standard It Trigger Ter It Values ns s/Tables grams s g Standards Setup onships ct Areas es r Templates ds	
	< <u>B</u> ack <u>N</u> ext >	Deriv	e Close	Help	

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

8. Click **Derive**.

The model derivation process starts.

Embedding	x
Processing Data : (13) / (24) - (54)%	
Cancel	

Once the conversion is complete, the existing model in 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:

 In erwin Data Modeler (DM), click Actions > Reverse Engineer. The New Model screen appears.

New Model ×
Туре ————
○ Logical ○ Physical ● Logical/Physical ○ Match template
Target Server
Match template target server
Database: MongoDB V Version: 4.x
Template
<default> 🗸 💆 💆</default>
Preserve the template <u>b</u> inding
Next Cancel

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

The Reverse Engineer Process Wizard appears.

	Reverse Engineer From
Connection	Database O Script File
Database	File: Browse
Collection	Overview
Option Set	Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineer from the physical
Detail Options	database.
Scheduler	Reverse engineering is the process of generating the physical database schema from a physical model. You can customize the Reverse Engineer for your target server using features in the Detail options.
	The Option Set page lets you customize the Reverse Engineer option set.
	The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page also
	lets you use Complete Compare and Martfor the target location.
	Ites you use Complete Compare and Martfor the target location. The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.

- 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.

Reverse	Engineer From		
0			
⊖ <u>D</u> ata	base	Script File	
<u>F</u> ile:			Browse

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

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

Reverse Engineer Pro	cess Wizard					\times
Reverse Engineer Conn This page provides a Reve	ection Information erse Engineer Connect	tion Information.				
Overview	Data <u>b</u> ase	MongoDB 4.x				~
Connection	Authentication	Database Authentication				\sim
<u>Database</u>	<u>U</u> ser Name					
Collection	Pass <u>w</u> ord					
Option Set		Parameters	Value			_
Detail Options	Connection Meth	od	CONNECTION STRING			\sim
Scheduler	Connection String	g:	mongodb+srv:/	-	'gra	.m
	Recent Connection	Connect ns: db:((MongoDB 4.x)	Disconnect			
	< <u>E</u>	Back Next >	ОКС	ancel	H	elp

6. Click **Connect**.

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

7. Click Next.

Reverse Engineer Pr	rocess Wizard			—		>
abase Selection Op is page allows the use	ptions er to select available database li:	st.				
Overview	System Objects					
Connection	Available Databases:	Enter filter text	Selected Datab	bases:		
Database	Mongo_Collation_01	(4)				
Collection		omment (3)				
Option Set	financial (5) MonceDR View 01 (1)	0				
Detail Options	newDB (2)	IJ				
Scheduler	sample_airbnb (2)					
	sample_analytics (3)	, ,				
	sample_geospatial (1))				
	sample_supplies (1)					
	sample_training (12)					
	sample_training_test	(1)				
	sample_weatherdata	(1)				
	test (3)					
	Items: (15)		Items: (0)			
	< Back	Next >	OK	Cancel	н	elp

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 🗭.



Available Databases:	Enter filter text]	Selected Databases:
_Mongo_Collation_01	. (4)		financial (5)
_Mongo_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collation_Collat	omment (3)		MongoDB_View_01 (1)
dbname (3)			
newDB (2)		۵	
📋 sample_airbnb (2)			
sample_analytics (3)			

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.

Reverse Engineer Procession	cess Wizard			_	
Collection Selection Opt This page allows the user t	ions to select available collection list.				
Overview	Document Count v		🗹 Deep Search	Sampling Sequence	ORandom
Connection	Available Collections: En	nter filter text	Selected Collection	IS:	
Database	book2 (1)				
Collection	collection_view_001 (2)				
Option Set	item2 (1)				
Detail Options	newproduct (1)				
Scheduler					
]		
		為			
		~			
	Items: (2)		Items: (0)		
	< <u>B</u> ack N	<u>V</u> ext >	ОК	Cancel	Help

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 🗭.

Available Collections:	Enter filter text	
book2 (1) collection_View_001 (2) employee (0) item2 (1) newProduct (1) newProduct2 (1)		(-)

This moves the selected collections under Selected Collections.

Available Collections:	Enter filter text	Selected Collections:		
newProduct2 (1)		Image: A start of the start	book2 (1) collection_View_001 (2) employee (0) item2 (1) newProduct (1)	

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.

Reverse Engineer Pro	cess Wizard			×
Reverse Engineer Optio This page allows the user	n Set to select the Reverse Engineer Option Set.			
Overview Connection	Items to Reverse Engineer Option Set: Open Save Save As			
Database	Default Option Set - Database			\sim
Collection				
Option Set	· ···································			
Detail Options				
Scheduler				
	< <u>B</u> ack <u>Next</u> OK C	ancel	He	elp

13. Click Next.

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

Reverse Engineer Pro	cess Wizard	_		×
Reverse Engineer Detai This page provides a Deta	il Options il Options of the Reverse Engineer Process.			
Overview Connection	NSM Options Glossary CSV File: Browse Reverse Engineer			
Database	System Objects			
Option Set				
Detail Options	O Owners (comma separated):			
Scheduler	Infer Primary Keys Indexes Relations Indexes Names Case Conversion of Physical Names None Olower OUPPER Force Case Conversion of Logical Names None Olower OUPPER OMixed Include Generated Triggers			
	< Back Next > OK	Cancel	He	elp

14. Click **OK**.

The reverse engineering process starts.

Reverse Engineer from Database	_		×
Reverse Engineer			
Retrieving View Objects			
		Cancel	

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:

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.

Forward Engineer Sc	chema Generation Wizard — 🗆 🔿
Schema Generation Ov This page provides an ov	rerview rerview of the Forward Engineer Schema Generation.
Overview	Overview
	Welcome to the Forward Engineer Wizard. This wizard lets you forward engineer a model.
Collection Filter	Forward engineering is the process of generating the physical database schema from a physical model. You use the
	Schema Generation wizard to forward engineer a model and generate the schema. The schema that you generate includes all options that are supported in yourtarget server. You can customize the generated schema for your target server using features in the Schema Generation wizard.
	The Option Selection page lets you customize the Schema Generation option set and the database template.
	The Summary page lets you review the selected Schema Generation options in a hierarchical tree structure. The page also lets you enter a comment for the current Schema Generation option set.
	The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.
	The Collection Filter page lets you select a subset of the tables in a model for forward engineering.
	The Preview page lets you preview the schema DDL script generated for the current model.
	· · · · · · · · · · · · · · · · · · ·
	< <u>Back</u> <u>N</u> ext > Generate OK Cancel Help

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.

Forward Engineer Sch	iema Generation Wiza	ırd					_ D	×
Schema Generation Opt	tions to change the Forward I	Engineer Schema	Generation Option	1.				
	<u>-</u>							
Overview	Option Set:	Default NoSQL	Schema Generation	~	Open	Save	Save A	s
Option Selection		Manage DD fat				-		
Collection Filter	Database Template:	MongoDB.tet			Browse	<u>E</u> dit	Rese	,t
Preview	General Syntax Op	otion	_					
	Use DB	Comments	🗹 Blank Value					
	- Collection Syntax (Ontion						
			Incort	C Schomo	Validation			
	Vicieate	⊡ brop	∑Insert		Validadori			
	View Syntax Optio	n						
	Create	Drop						
	Index Syntax Opti	on						
	Create	Drop						
	User Syntax Optio	n						
	Create	Drop						
		< <u>B</u> ack	: <u>N</u> ext >	Genera	ote OK	Cano	el	Help

4. Click Next.

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

Forward Engineer Schema Generation Wizard				
Schema Generation Col This page allows the user	llections Filter to change the filter on the Forward Engineer Schema Generation Collections.			
Overview Option Selection Collection Filter Preview	Image: Second state sta			
	< <u>B</u> ack <u>N</u> ext > Generate OK (Cancel	He	lp

5. Select the collections that you want to forward engineer.
6. Click **Preview** to view the schema script.

Forward Engineer Sc	hema Generation Wizard	_		Х
Forward Engineer Sche This page provides a prev	ma Generation Preview view of the Forward Engineer Schema Generation.			
Overview	🖹 🏯 🚊 🌉 🧤 🐼 🍞 🗆 Auto Error Check			
	Viewer Text			
Collection Filter	1 use eMoviesTechPubs;			^
Preview	<pre>2 db.createCollection("MOVIE"); 3 db.MOVIE.insert(4 (5 "movie_number": 0, 6 "movie_title": "", 7 "movie_director": "", 8 "description": "", 9 "star_l_name": "", 10 "rating": "", 11 "star_2_name": "", 12 "genre": "", 13 "rental_rate": 0,</pre>			
	14 "movie_url": "", 15 "movie_clip": BinData(0," "), 16日 "MOVIE_STORE": [17 { 18 "movie_number": 0, 19 "store_number": 0 20 } 21], 22 "MOVIE_COPY": [23 {			
	24 "general_condition": "", 25 "movie_format": "", 26 "mo_co_num": 0, 27 "movie_number": 0.			
	283- "MO RENT REC" · I		-	¥
	<		3	•
	< Back Next > Generate OK Ca	ncel	н	elp

Use the following options:

- Auto Error Check: Select this option to enable auto error check by the forward engineering wizard.
- Error Check (¹): Use this option to run an error check. Based on the results, you can correct the generated script.
- **Text Options** (We): Use this option to configure the preview text editor's look and feel, such as window, font, syntax color settings. For more information,

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:

Open your MongoDB model in erwin Data Modeler (DM).
 Note: Ensure that you are in the Physical mode.

eMoviesDoc.erwin 🗙		*	Objects Count	μ×
		RATING	Name: MODEL_ Type: Logical/F View Mode: Physical	6 'hysical
id UST_number CUST_address CUST_dity CUST_first_name CUST_first_name CUST_last_name	CUST_CREDIT _id CUST_number oredit_card oredit_card_exp status_code	stars review	Subject Areas: 0 Collections: 5 Fields: 213	Inde Rela
CUST_zip_oode email PAYMENT [0] payment_transaction_number payment_amount payment_ataus ocheck_bank_number ocheck_number epay_vendor_number epay_acount_number oredit_oard_exp oredit_oard_exp oredit_oard_yppe	■ PAYMENT ▼ [0] payment_transaction_number payment_date payment_date payment_date payment_date payment_status check_bank_number check_number epay_scoout_number oredit_oard_number oredit_oard_number oredit_card_ppe EMP_number customer_no-	id movie_number movie_title movie_director description star_1_name rating star_2_name genre rental_rate movie_url movie_url movie_url movie_tip ■ MOVIE_STORE ▼[0]	Style: Pie Style (3D)	1
< ER_Diagram_118		>	Properties Objects Cou	nt

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.

Right Model Selection			
Select the Right Model Please select a model for the ri	ight side of complete co	Co mpare. Da	mpare Level: tabase
	Load From File	🔿 Database / Script	◯Mart
<u>Overview</u>			Load
Left Model	Open Models in Me	emory:	
Right Model	Open Models	Location	
Type Selection	Model_6	C: \Users \admin \erv	vin, Inc\Technical Publication
Left Object Selection			
Right Object Selection			
Advanced Options			
	Set selected m	odel as read only	
Load Session Save Session	< <u>B</u> ack	Vext > Compare	Close Help

3. Click **Database/Script**.

By default, the Allow Demand Loading option is selected.

Right Model Selection				×
Select the Right Model Please select a model for the righ	t side of complete co	mpare.	Compare Level: Database	
<u>Overview</u>	Load From	● Database / Script ☑ Allow Demand Li	⊖ Ma pading	t Load
Left Model Right Model Type Selection Left Object Selection Right Object Selection Advanced Options	Open Models in Me Open Models Model_6	c: Users admin (erwin, Inc\Technical	Publication
Load Session Save Session	< <u>B</u> ack <u>!</u>	Vext > Compare	Close	Help

4. Click Load.

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

New Model X
Туре
○ Logical ○ Physical ● Logical/Physical ○ Match template
Target Server
Match template target server
Database: MongoDB V Version: 4.x
Template
<default> 🗸 💾 🚰</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.

Reverse Engineer Pr	rocess Wizard – 🗆 >
everse Engineer Pro This page provides an o	cess Overview verview of the Reverse Engineer Process.
Overview	Reverse Engineer From
Connection	Database O Script File
Database	Eile:
Collection	Overview
Option Set	
Detail Options	database.
Scheduler	Reverse engineering is the process of generating the physical database schema from a physical model. You can customize the Reverse Engineer for your target serverusing features in the Detail options.
	The Option Set page lets you customize the Reverse Engineer option set.
	The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page also lets you use Complete Compare and Martfor the target location.
	The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.
	The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering.
	The Preview page lets you preview the schema DDL script generated for the current model.
	< Back Next > OK Cancel Help

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.

Reverse Engineer Proc	cess Wizard					×
Database Selection Opti This page allows the user t	ions to select available database li	ist.				
Overview	System Objects					
Connection	Available Databases:	Enter filter text	Selected Databases:			
Database	Mongo_Collation_0	1 (4) omment (3)				
Collection	dbname (3)					
Option Set	eMoviesDoc (4))				
Detail Options	financial (5)	, ,				
Scheduler	MongoDB_View_01(newDB (2) sample_airbnb (2) sample_analytics (3) sample_geospatial (1 sample_mflix (5) sample_supplies (1) sample_training_test sample_weatherdata test (3)	1) 1) ; (1) a (1)				
	Items: (17)		Items: (0)			
	< <u>B</u> ack	Next >	ОК	Cancel	He	lp

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

This moves the selected databases under Selected Databases.

Reverse Engineer Provide the second secon	ocess Wizard Hions r to select available database list.				×
Overview	System Objects	Selected Databases:			
Database Collection <u>Option Set</u> Detail Options Scheduler	 _Mongo_Collation_01 (4) _Mongo_Collation_Comment (3) dbname (3) eMoviesTechPubs (4) financial (5) MongoDB_View_01 (1) newDB (2) sample_airbnb (2) sample_analytics (3) sample_geospatial (1) sample_urflix (5) sample_training (12) sample_training_test (1) sample_training_test (1) test (3) 	eMoviesDoc (4)			
	Items: (16) < <u>B</u> ack <u>N</u> ext >	Items: (1)	ancel	He	٩þ

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).

Reverse Engineer Pro Collection Selection Opt This page allows the user	cess Wizard t ions to select available collection lis	t.			
Overview	Document Count ~	4	Deep Sear	Sampling ch	◯ Random
Connection	Available Collections:	Enter filter text	Selected Colle	ctions:	
Database			CUST (2)		
Collection			CUST_CF	EDIT (2)	
Option Set				2)	
Detail Options			4		
Scheduler					
	Items: (0)		Items: (4)		
	< <u>B</u> ack	<u>N</u> ext >	ОК	Cancel	Help

- 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 righ	t side of complete	compare.	Comp Data	oare Level: base	
<u>Overview</u>	Load From	● Datab ☑ All	oase / Script ow Demand Loadir	⊖ Mai	rt Load
Left Model	Open Models in I	Memory:			
Right Model	Open Models	Loc	ation ud.mongodb:(: (Mongol)B 4.x)
Type Selection	Model_6	C:\	Users \admin \erwin	, Inc\Technical	Publication
Left Object Selection Right Object Selection					
Advanced Options					
	Set selected	model as read o	only		
Load Session Save Session	< <u>B</u> ack	<u>N</u> ext >	Compare	Close	Help

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

Type Selection					×
Complete Compare Type Selection Please select the types of objects	on to be compared in co	mplete compar	Com re. Data	pare Level: Ibase	
	Compare Level	□P	hysical Level	🗹 Data	abase Level
<u>Overview</u>	Option Set: Speed	Option Set		~ 🗁	🗋 🏝 🔳
Left Model Right Model Type Selection Left Object Selection Right Object Selection Advanced Options	Image: Constraint of the second se	pperties llection tabase lationship er Id w			
Load Session Save Session	< <u>B</u> ack <u>N</u>	ext >	Compare	Close	Help

For example, the following image shows the default options.

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

Left Model Object Selection			×
Left Model Object Selection Please select the objects you want	t to compare from left model.	Compare Level: Database	
Overview Left Model Right Model Type Selection Left Object Selection Right Object Selection Advanced Options	Choose Objects Using Sets	Selected Objects:	
Load Session Save Session	< <u>B</u> ack <u>N</u> ext >	Compare Close He	elp

For example, the following image shows the default options.

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

Right Model Object Selection				×
Right Model Object Selection Please select the objects you want	to compare from the right model.	Co Da	mpare Level: tabase	
Overview Left Model Right Model Type Selection Left Object Selection Right Object Selection Advanced Options	Choose Objects Using Sets	Selec	ted Objects:	ns
Load Session Save Session	< <u>B</u> ack <u>N</u> ext >	Compare	Close	Help

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.

Resolve Differences				– 🗆 X
<				
🖬 🗟 🎒 Gi				
🖻 🛃 📣 🙆 💪				
= 🗟 ≢ 🕃 =3				
🔳 🖻 🏛 🗸				
Object View	1	Model_6 - C:\Users\admin\erwin, Inc\T		14 - cloud.mongodb:(avinashm):
🗆 🗋 Model	Model	_6	\Rightarrow	Model_14
🗆 🚞 Collections			<¢¢	
E Collection	≢ CUST			CUST
E Collection	≢ CUST	CREDIT	\$	CUST_CREDIT
E Collection	≢ MOVIE	E	$\langle \downarrow \downarrow \rangle$	MOVIE
Collection	≢ RATIN	IG	$\langle \downarrow \downarrow \rangle$	
🗆 🚞 Fields			$\langle \downarrow \downarrow \rangle$	
III Field	≠ reviev	v		
🔚 Field	≢ stars			
🕀 🎆 Collection	STOR	E		STORE
Property View	® ≢id.str	Model_6 - C:\Users\admin\erwin, Inc\T		Model_14 - cloud.mongodb:(avinashm):
🗄 📆 Field Order	≢_id,st	ore_number,store_manager,store_addr	¢¢	

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 b.

This launches the Forward Engineering Alter Script Generation Wizard.

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

	Optio <u>n</u> Set:	Select the optio	ins for DB Sync	~	Open	<u>S</u> ave	Save <u>A</u> s	
	Database <u>T</u> emplate:	MongoDB.fet			Browse	<u>E</u> dit	Re <u>s</u> et	
Preview	General Syntax Op	ition						
	Use DB	Comments	🗹 Blank Value					
	Collection Syntax (Option						
	Create	Drop	✓ Insert	Schema	Validation			
	View Syntax Option	n						
	Create	Drop						
	Index Syntax Option	on						
	Create	Drop						
	User Syntax Option	n						
	Create	Drop						

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

Forward Engineer Alter Script Schema Generation Wizard	_		Х
Schema Generation Collections Filter This page allows the user to change the filter on the Forward Engineer Schema Generation Collections.			
Overview Option Selection Freview Preview Preview Selected, 5 shown Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Preview Previ			
< <u>B</u> ack <u>N</u> ext > Generate OK	Cancel	He	elp

- 20. Click **Preview** to view and verify the alter script.
- 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.
- 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 <u>JSON</u> and <u>AVRO</u> file formats. The following table lists the supported objects and data types for each format:

File	Objects	Data Types
Format	•	
JSON	JSON Objects	 Object
	Fields	Array
	Relationships	Integer
		Null
		String
		Number
		Boolean
AVRO	Records	Array
	Fields	Boolean
	Relationships	Union
		Map
		• int
		Double
		 Object
		String
		Byte
		enum
		Fixed
		Long

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:

 In erwin Data Modeler (DM), click Actions > Reverse Engineer. The New Model screen appears.

New Model X	
Type Logical Physical Match template	
Target Server	
Match template target server	
Database: JSON V Version: 1.x V	
Template	
<default></default>	
Preserve the template binding	
Next Cancel	/

2. Click Logical/Physical and set Database to JSON.

3. Click Next.

The Reverse Engineer Process Wizard appears.

Reverse Engineering N Overview Reverse Engineering option	Nizard — 🗆 🗙
Overview	Reverse Engineer From
Detailed Options Scheduler	Database Json Data Schema File Eile: C:\Users\ArunnShampath\erwin, Inc\Tedt Browse
	Overview Image: Control of the Reverse Engineering Wizard Walks you through the process of reverse engineering a model from a physical database. You can customize the operation for a target database of your choice using the several option sections available. Detailed Options: Use this section to configure detailed reverse engineering options, ruba is NSM options, objects to be reverse engineered, primary keys and relations inference, physical and logical name case conversions, and trigger indusion. Scheduler: Use this section to configure the model and mart location for the reverse engineering job being scheduled.
	< Back Next > OK Cancel Help

- 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 Glossary CSV File: Browse Reverse Engineer System Objects Records/Views Owned By Image: All Options Current User Owners (comma separated): Infer Primary Keys Indexes Relations Names Case Conversion of Physical Names Infer Image: One Owner Owner OUPPER Force Case Conversion of Logical Names Image: One Owner Include Generated Triggers			
< Back Next > OK	Cancel	Н	ielp

6. Click **OK**.

The reverse engineering process starts.

Reverse Engine	er From Script		×
Script Name:	C:\Users\admin\erwin, Inc\Technical Publications Team - erwin Data Modeler and	Ma	
🔍 Reverse E	ngineer		
Cancel	Building		

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:

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.

orward Engineer So		
page provides an ov	rerview erview of the Forward Engineer Schema Generation.	
Overview		
Option Selection	Overview	
	Welcome to the Forward Engineer Wizard. This wizard lets you forward engineer a model.	
Table Filter Preview	Forward engineering is the process of generating the physical database schema from a physical model. You use the Schema Generation wizard to forward engineer a model and generate the schema. The schema that you generate includes all options that are supported in yourtarget server. You can customize the generated schema for your target server using features in the Schema Generation wizard.	ie
	The Option Selection page lets you customize the Schema Generation option set and the database template.	
	The Summary page lets you review the selected Schema Generation options in a hierarchical tree structure. The page also lets you enter a commentfor the current Schema Generation option set.	
	The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.	
	The Table Filter page lets you select a subset of the tables in a model for forward engineering.	
	The Preview page lets you preview the schema DDL script generated for the current model.	
		-

3. Click Option Selection.

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

Forward Engineer Sch	ema Generation Wiza	rd				-		×
Schema Generation Opt This page allows the user t	ions to change the Forward E	ngineer Schema	Generation Options					
Overview	Optio <u>n</u> Set:	Default NoSQL	Schema Generation	~	Open	<u>S</u> ave	Save <u>A</u> s	
Option Selection	Database Template:	JsonDB.fet			Browse	Edit	Reset	
<u>Table Filter</u>	General Syntax Op	tion						
Preview	Use DB	Comments	Blank Value					
	Collection Syntax C	Option						
	Create	Drop	Insert	Schema	Validation			
		< <u>B</u> ack	<u>N</u> ext >	Genera	te OK	Cance	el He	lp

4. Click Next.

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

Forward Engineer Sc Schema Generation Ta This page allows the user	hema Generation Wizard bles Filter to change the filter on the Forward Engineer Schema Generation Tables.	_		×
Overview Option Selection Table Filter Preview	Image: Construct of the system Image: Construct of the system			
	16 selected, 16 shown Apply selection of Tables to T < Back	ablespace Cancel	s and Dat	tabases elp

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

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

Forward Engineer Sc	nema Generation Wizard			×
Forward Engineer Sche This page provides a prev	na Generation Preview iew of the Forward Engineer Schema Generation.			
Overview	🛅 🖽 📑 🌉 🍉 Ď 🗆 Auto Error Check			
Option Selection	Viewer Text			
Table Filter	1 /* [JSON Object:academic catalogs 15.0] */			^
	3 "title": "", 4 "description": "", 5 "type": ", 6 "properties": { 7 "metadata": { 8 "title": "", 9 "description": "", 10 "type": "", 110 "properties": { 120 "createdBy": { 13 "title": "", 14 "description": "", 15 "type": "" 16 }, 170 "createdOn": { 18 "title": "", 19 "description": "",			
	19 "description": "", 20 "oneOf": [21 { 22 "type": "", 23 "format": "", 24 "pattern": "", 25 }, 26 { 27 "type": "", 28 "maxLength": 0		3	~
	< <u>B</u> ack <u>N</u> ext > Generate OK Car	icel	Н	elp

Use the following options:

- Auto Error Check: Select this option to enable auto error check by the forward engineering wizard.
- Error Check (¹): Use this option to run an error check. Based on the results, you can correct the generated script.
- **Text Options** (WA): Use this option to configure the preview text editor's look and feel, such as window, font, syntax color settings. For more information,

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.

Save Forward Eng	gineering Script X
Data <u>b</u> ase	JSON
<u>S</u> et Path	C:\Users\admin\OneDrive - erwin, Inc\Documents\M
Generate File O	otions ultiple Files
	<u>S</u> ave <u>C</u> lose

- 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 <u>Oracle 12c R2, 18c, 19c, and 21c</u> 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
 BINARY	• CHAR	• DATE	 JSON*
DOUBLE	• CHAR()	INTERVAL	 ANYDATA
 BINARY	• CHARVARVING()	DAY TO	ANYDATASET
FLOAT		SECOND	ANYTYPE
• DEC	CHARACTER	INTERVAL	BEILE
• DEC()	CHARACTER()	YEAR TO	BLOD
• DEC()	CHARACTERVARYING	MONTH	BLOB
5 DLC(,)	()	 TIMESTAMP 	LONGRAW
 DECIMAL 	CLOB	 TIMESTAMP 	ORDAUDIO
 DECIMAL() 	DBURITYPE	WITH LOCAL	ORDDICOM
 DECIMAL(,) 		TIMEZONE	
DOUBLE	• URITIPE	TIMESTAMP	• ORDDOC
PRECISION	HTTPURITYPE	TIMEZONE	ORDIMAGE
REAL	 JSON 		ORDVIDEO
• ΕΙΟΑΤ	• JSON()	()	• RAW()
	LONG	 TIMESTAMP 	SDO GEOMETRY
• FLOAT()		() WITH	
• INT	• NATIONAL CHAR	LOCAL	• SDO_GLORASTER
• INTEGER	NATIONAL CHAR	TIMEZONE	SI_AVERAGECOLOR
NUMBER	VARTING()	 TIMESTAMP 	• SI_COLOR
	NATIONAL CHAR()	() WITH	

• NUMBER()	NATIONAL	TIMEZONE	• SI_
• NUMBER(,)	CHARACTER		COLORHISTOGRAM
NUMERIC	NATIONAL CHARACTER		SI_FEATURELIST
• NUMERIC()	VARYING()		 SI_ POSITIONALCOLOR
 NUMERIC (,) 	 NATIONAL CHARACTER() 		SI_STILLIMAGE
ROWID	NCHAR		• SI_TEXTURE
SMALLINT	 NCHAR VARYING() 		
	• NCHAR()		
	NCLOB		
	• NVARCHAR2()		
	UROWID		
	• UROWID()		
	• VARCHAR()		
	• VARCHAR2()		
	XDBURITYPE		
	• XMLTYPE		

*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 <u>SQL Server support sum</u>mary.

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.

erwin Data Modeler Target Server				×
Database: SQL Server	~	SQL Server Version 20	019	~
✓ Auto Normalization ✓ Relationships		Default S 20 char(18 20 Null Op 0 NOT NU 0 NULL	012 014 016/2017 019 zure	
		OK	К	Cancel

The following table lists the supported objects:

Supported Objects					
Always Encrypted	Resource Pools*				
Кеуѕ	Schemas				
Application Roles	Sequences				
 Assemblies* 	 Server Audits* 				
Asymmetric Keys	Server Audit Spe-				
Certificates	cification*				
Credentials*	Statistics				
Database Roles	Stored Procedures				
Databases	Symmetric Keys				
Database Triggers	• Spatial Indexes (Table)				
External Data	Synonyms				

Source	Tables
 External File Format* 	 ColumnStore Indexes (Table)
External Library*	• XML Indexes (Table)
External Table	 Indexes (Table)
• Full-Text Catalogs	Table Triggers
 Full-Text Indexes (Table) 	TriggersUser Ids
• Full-Text Stoplists	Views
Functions	View Indexes
Logins	View Triggers
 Partition Func- tions 	XML Schema Col- lections
 Partition Schemes 	

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

The following table lists the supported data types:

Exact Numerics	Approx- imate	Date and Time	Char- acter	Unicode Char-	Binary Strings	Geo Types	Others
	Numer-		Strings	acter			
	ics			Strings			
 bigin- 	• fl-	 date 	• ch-	• nc-	• bin-	 rowver- 	•
t	0- at	• dat-	ar	har	ary	sion	CHA- R
• num-	at	etime-	• va-	• nv-	• var-	 hier- 	VAR-
eric	• re-	offset	rc-	arc-	bin-	archyid	YING
• bit	al	• dat-	h- ar	har	ary	 uniquei- den- 	• CHA-
• smal-		etimez	• te-	ext	age	tifier	RAC-

lint	 smalld- 	xt		• sql_vari-	TER
 deci- mal small- mon- 	ate- time • dat- etime			antxmlgeo- metry	 CHA- RAC- TER VAR- YING
• int • tiny- int	• time			 geo- graphy 	• NATI- ONA- L CHA-
• mon- ey					VAR- YING
					• NATI- ONA- L CHA- R
					 NATI- ONA- L CHA- RAC- TER
					• NATI- ONA- L 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
 TINYINT 	CHAR	 DATE 	Geometry Type
 SMALLINT 	 VARCHAR 	• TIME	POINT
MEDIUMINT	 BINARY 	 DATETIME 	LINESTRING
• INT,	CHAR BYTE	TIMESTAMP	 POLYGON
INTEGER		• YEAR	MULTIPOINT
BIGINT DECIMAL	VARBINARY	MULTILINESTRING MULTIPOLYGON	
--------------------	------------------------------	----------------------------------	
• DECIMAL, DEC,	TINYBLOB	GEOMETRYCOLLECTION	
NUMERIC,	BLOB	GEOMETRY	
FLOAT	BLOB and		
• DOUBLE,	TEXT Data		
DOUBLE	Types		
PRECISION, REAL	MEDIUMBLOB		
• BIT	LONGBLOB		
	 TINYTEXT 		
	• TEXT		
	MEDIUMTEXT		
	LONGTEXT		
	• JSON Data		
	Туре		
	• ENUM		
	• Set Data Type		

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 PI Defaults History Options Definition LIDD History Notes Extended Notes	
Medel Leference inc	
Name EMOVIES 2021 P1	
Author erwinTeam	
Type: Logical / Physical Target Server And Version: SQL Server 2019	
Notation	
	~
	
Modeling Features	
Is Dimensional	
Data Valit 2.0	
Close C	ancel
	etails

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-

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

Objects Count		4 ×
۵		
Name: EMOVIES 2	2021 R1	
Type: Logical/Phy	ysical	
View Mode: Physical		
Subject Areas: 4	Indexes: 30	Views: 2
Tables: 9	Relationships: 20	MaterializedView: 0
Columns: 75	Sub-Categories: 0	Databases: 0
Style: Pie Style (3D)	Pallete: Illustration	▼ Legend
Properties Objects Count		· · · · · · · · · · · · · · · · · · ·

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		ų×
H 9+ 6 7 5		
▲ General		^
Name	EMOVIES 2021 R1	
Author	erwinTeam	
Definition		
Туре	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	\checkmark	
Unique Names	Not Unique	
Is NSM Option From AMT		
▲ 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 <u>Object Browser</u> topic.

bject Browser									- 🗆	×
₽ 🛃		Enter Search	Text	Tables: (26)	Views: ((4) Materialized V	iews: (0) 📄	B #	€	
BookAut	h		^	Summary				1		^
🗄 🔚 Stor_Nan	n			Tables		26		2 CREATE	TABLE BookAut	ch
🗄 🔚 Purchase	Ordr			Columns		138		3 (
🗄 🔚 Rylty				Views		4		4 Aut	n_ia	
🗄 🔚 Disc				Materialized V	iews	0		6 PRIMAR	RY KEY (Auth 1	Id.1
🗄 🔚 Job				Indexes		54		7 FOREIGN	KEY (Auth Ic	d)
🗄 🔚 Publshr_L	ogo			Index Member	rs	73		8 FOREIGN	KEY (Book Id	a) i
🗄 🔚 Emp				Relationships		41		9);	-	
🗈 🔚 Auth				Domains		22		10		
🗉 📄 Publshr								11 COMMENT	C ON TABLE Boo	o kAı
🗈 🔚 Book				Current Subject Area				12 12 COMMENT	ON COLUMN R	V
🗄 🔚 Ordr_ltm	1		~	Current Subject	LLAIEd		<	TS: COMMENT		>
🏂 Index 🛛	🎽 T 🛆	🥔 Index	🛃 Orde	er 🛛 🌌 Owner	. ^	😏 Relationsh	ᇕ Type 🛛 🖂	🧝 Parent	🥩 Child	^
UPKCL_taind PK	<	Auth_Id	Ascendin	g BookAuth		R/53	Identifying	Ordr_Ship	Bk_Ordr	
UPKCL_taind PK	(Book_Id	Ascendin	g BookAuth		R/44	Derived	Pmt	Payment View	- 1
auidind IF		Auth_Id	Ascendin	g BookAuth		R/38	Subtype	Pmt	Personal_Chk	
titleidind IF		Book_Id	Ascendin	g BookAuth		R/45	Derived	Personal_Chk	Payment View	
UPK_storeid PK	ς	Stor_Id	Ascendin	g Stor_Nam		R/39	Subtype	Pmt Manu Orda	Mony_Ordr	
VIFIStore IF	,	Rgn_Id	Ascendin	g Stor_Ivam		R/43	Nep Identifying	Mony_Orar	Payment view	
VIE2Durcha IE		Oust Id	Ascendin	g Purchase		P/40	Subtype	Dmt	Ord_Ork	
XIE 1Purcha IF		Stor Id	Ascendin	g Purchase		R/42	Derived	Crd. Card	Payment View	
XPKRovalty PK	<	Rvltv Id	Ascendin	a Rvltv	~	R/47	Derived	Cust	Payment View	~
ndexes: (54) Inde	x members: ((73)				Relationships: (41)		Export •	Properties	ose

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.

Auto Normalization	Auto Denormalization	
Source:	<u>T</u> arget:	~
🤧 Collection 🛛 🖓 Database 🛆	Source:	
	Collection Database A	
Relationship Type: Auto	Embedding Type: Embed as Auto Cascading All Euceel: 0 Auto Cleanup Cascade Help	~

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 <u>Reverse</u> <u>Engineering</u> and <u>Forward Engineering</u> topics.

Overview Reverse Engineer From Connection Database Database Overview Collection Overview Option Set Detail Options Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target serverusing features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Option Set page lets you customize the Reverse Engineer option set. The Option Set page lets you customize the Reverse Engineer option set. The Option Set page lets you customize the Reverse Engineer option set. The Collection Filter page lets you select asubset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model	rse Engineer Pro	cess Wizard — 🗌
Overview Reverse Engineer From Database © gatabase Collection Overview Option Set Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineerfrom the pd database. Scheduler Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineerfrom the pd database. Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target server using features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. The group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select asubset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model	Engineer Proce e provides an ove	ess Overview erview of the Reverse Engineer Process.
Connection Database Collection Option Set Detail Options Scheduler Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineer from the pl database. Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target server using features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Override page lets you enter the owner override name of an object group. Th group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select asubset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model.	rview	Reverse Engineer From
Database Elle: Browse Collection Overview Overview Option Set Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineerfrom the pl database. Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target serverusing features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. Th group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets youpreview the schema DDL script generated for the current model	nection	Database O Script File
Collection Overview Option Set Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineerfrom the pl database. Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target serverusing features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. Th group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select asubset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model		Eile: Browse
Option Set Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineerfrom the pl database. Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target serverusing features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. Th group and name you indicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model	ction	Overview
Detail Options database. Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target server using features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. Th group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model		Welcome to the Reverse Engineer Wizard. This wizard lets you reverse engineer from the physical
Scheduler Reverse engineering is the process of generating the physical database schema from a phy model. You can customize the Reverse Engineer for your target serverusing features in the options. The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. Th group and name you indicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model	il Options	database.
 The Option Set page lets you customize the Reverse Engineer option set. The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor thetarget location. The Owner Override page lets you enter the owner override name of an object group. The group and name youindicate override the owner name assignment for that object group in model. The Collection Filter page lets you select asubset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current model. 	duler	Reverse engineering is the process of generating the physical database schema from a physical model. You can customize the Reverse Engineer for your target serverusing features in the Detail options.
The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page lets you use Complete Compare and Martfor the target location. The Owner Override page lets you enter the owner override name of an object group. The group and name you indicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current mode		The Option Set page lets you customize the Reverse Engineer option set.
The Owner Override page lets you enter the owner override name of an object group. Th group and name you indicate override the owner name assignment for that object group in model. The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current mode		The Scheduler page lets you customize the Reverse Engineer Scheduler options. The page also lets you use Complete Compare and Martfor the target location.
The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering. The Preview page lets you preview the schema DDL script generated for the current mode		The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.
The Preview page lets youpreview the schema DDL script generated for the current mode		The Collection Filter page lets you select a subset of the tables in a model for Reverse engineering.
		The Preview page lets you preview the schema DDL script generated for the current model.
< Back Next > OK Cancel		< Back Next > OK Cancel H

	📕 Forward Engineer Schema Generation Wizard 🛛 📃								
Sch Th	Schema Generation Overview This page provides an overview of the Forward Engineer Schema Generation.								
	Overview	Overview							
		Welcome to the Forward Engineer Wizard. This wizard lets you forward engineer a model.	<u></u>						
		Forward engineering is the process of generating the physical database schema from a physical model. You use the Schema Generation wizard to forward engineer a model and generate the schema. The schema bat you generate includes all options that are supported in your target server. You can customize the generated schema for your target serve using fastures in the Schema Generation wizard to forward engineer a model and the schema for your target server. You can customize the generated schema for your target server.	ier						
		The Option Selection page lets you customize the Schema Generation option set and the database template.							
		The Summary page lets you review the selected Schema Generation options in a hierarchical tree structure. The page also lets you enter a comment for the current Schema Generation option set.							
		The Owner Override page lets you enter the owner override name of an object group. The object group and name you indicate override the owner name assignment for that object group in the model.							
		The Table Filter page lets you select a subset of the tables in a model for forward engineering.							
		The Preview page lets you preview the schema DDL script generated for the current model.							
			\sim						
		< Back Next > Generate OK Cancel	Help						

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).

Options							×
General XML Diagnostic	s Reporting	Mart					
Messages							-
<u>R</u> eset all messages							
File Locations							-
Default model location:	C:\Users\admi	n\OneD	rive - erwin,	Inc\D	ocuments∤My	Browse	
Default template location:	C:\Users\admi	n\OneD	rive - erwin,	Inc\D	ocuments∤My	Browse	
Transaction log location:	C:\Users\admi	n\AppD	ata \Local \Ter	mp		Bro <u>w</u> se	
Diagram Diagram Suppress diagram tooltips Validate previous version metadata in model							
Enforce Relationship Nullability Rules							
Supertype-Subtype Transformation							
Quick Complete Compare							
Help Source							-
Use <u>o</u> nline help) Use <u>l</u> ocal help	Þ					
			OK		Cancel	Help	

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 namein 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]][/[defaultauthdb][?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 <u>Database Connection Parameters</u> topic.