IDERA

IDERA ER/Studio Data Architect
Quick Start Guide

Version 20.2.x
Published May 2024

May 1, 2024

Contents

Before you start

Introducing ER/Studio Data Architect

ER/Studio Data Architect is a visual modeling application used for platform-independent logical data architecture analysis and design, in addition to platform-specific physical database design and construction. The powerful, multi-level design environment addresses the everyday needs of database administrators, developers, and data architects who build and maintain large, complex database applications and strive to consolidate, report, and re-use metadata across the enterprise.

The progressive interface and simplicity of ER/Studio Data Architect is designed to effectively address the ease-of-use issues which have plagued data modeling and CASE tools for the past decade and more. The application equips you to create, understand, and manage the life-cycle of mission-critical database designs and business metadata within the enterprise.

ER/Studio Data Architect is rich and customizable, offering strong logical design capabilities like:

- The ability to spawn many physical designs from a corporate logical design.
- Bi-directional model comparison and information synchronization.
- Visual-Basic for Applications API for product customization.
- Powerful DDL reverse engineering and generation.
- Metadata import and export capabilities.
- Data lineage documentation.
- Sophisticated XML, HTML, and RTF-based documentation and reporting facilities.

With Team Server collaboration increases

- Central repository for models
- Check in/out of models or model fragments
- Version control

And with Team Server Core a web-based portal allows a wider audience to exploit the valuable knowledge in the models

- Publish models
- Business Glossary to support Data Governance
- Data Catalogs to allow users to find data
- Reporting and Analysis to publish useful reports on data
- Collaboration tools to allow the extended team to interact with models
- Subscription to get notified with data assets change

Product benefits by audience

- **Data Modelers and Data Architects**: ER/Studio Data Architect is critical for organizations concerned with eliminating data redundancy, creating an enterprise view of data assets, and assisting development with making informed decisions about how best to reuse elements pre-defined by the enterprise. Its powerful logical (non-database or technology specific) analysis and design environment helps to normalize and create an enterprise view of the objects concerning the data managed by an organization. More importantly, it can communicate this quickly through powerful reporting and metadata exchange mechanisms throughout the enterprise.

- **Database Administrators and Database Developers**: Managing databases is incredibly difficult without a blueprint or roadmap to understand important object dependencies. ER/Studio Data Architect's round-trip engineering capabilities including database reverse-engineering provide database administrators (DBAs) or developers with important *physical* data models in seconds. These models are powerful and efficient change management platforms, allowing users to update a model with the required changes necessary at the database and automatically generate DBMS-specific, syntactically correct, alteration or database DDL.

- **Business and IT Managers**: ER/Studio Data Architect's robust reporting facilities allow you to deliver critical information about designs to the enterprise in seconds. This heavily leveraged and beneficial capability of ER/Studio Data Architect allows users to provide, in literally seconds, clear, easily navigable and safe-to-distribute documentation about a database or enterprise data model to those who need to review it.

- **Data Stewards and Data Governance Teams**: Allow stewards to manage a Business Glossary of Business Terms and relate them to data models to ask the important questions "what information is important to us?" and "where is that information?"

What's new

For the latest updates on new features, fixed issues, and known issues in this release, please see the online version at http://docwiki.embarcadero.com/ERStudioDA/202/en/What%27s_New.

Installing IDERA ER/Studio Data Architect

This section will help you ensure your environment meets the technical requirements and helps you install ER/Studio Data Architect.

You can download a 14-day trial version of ER/Studio Data Architect, from https://www.idera.com/products/er-studio/data-architect. IDERA will send you a license registration code by email.

System requirements

Review the requirements before you install ER/Studio Data Architect. Adhering to these requirements optimizes the performance.

ER/Studio Data Architect Requirements		
Hardware	Processor	Dual core or higher
	RAM	4 GB
	Disk Space	2.5 GB
	Operating System	Any of the following: • Windows 7 • Windows 8 • Windows 8.1 • Windows 10 • Windows 11
Software	To use an Open Database Connectivity (ODBC) database as a data source, install the ODBC driver on your client computer. To connect to the DBMS using native connectivity, install the native client libraries available from the DBMS vendor.	
Privileges	Local administrator privileges (required for installation only.) Appropriate permissions are required to reverse or forward engineer directly to or from a database.	

Installing ER/Studio Data Architect

Tip	We recommend to have installed most current version of both, ER/Studio Team Server and ER/Studio Data Architect, to ensure compatibility.

Note	It is not possible to install the x64 bit version where a x32 bit version is also installed. You will need to remove any previously installed x32 bit versions of the software.

Log on to Windows with local administrator privileges.

1. Start the ER/Studio Data Architect installation program.

2. Walk through the installation wizard following the on-screen prompts.

   The following provides additional information for some options that you can check after the installation:

   o You can change the notation setting. Go **Tools > Options** from the ER/Studio Data Architect Main menu, and then click the Logical or Physical tab where you can choose the notation type.

   o By default, ER/Studio Data Architect installs shared directories and application files such as the DatatypeMapping, Macros, and Sample Models directories in the following location: C:\ProgramData\IDERA\ERStudioDA_x.x. If you prefer to store these files in another directory after installation, follow the procedure in Changing the Location of Shared Files.

   o For step-by-step instructions on how to use ER/Studio Data Architect, see the Tutorials section.

Additional Notes

For silent installations, launch the installer with commands adding –q to the installer:

```
installer.exe -q
```

Also, you can see a percentage bar instead of –q if you set a –passive:

```
installer.exe -passive
```

Upgrading from previous versions

Upgrade checklist

- Download ER/Studio Enterprise Upgrade. To download the most current release, go to https://license.embarcadero.com/edn/#login.

- Review the ER/Studio ReadMe. For the most current installation and usage information, see the ReadMe at http://docs.embarcadero.com/products/er_studio/ERDA202/ReadMe.htm.

- In case you need to upgrade Team Server/Repository to match the features available in your version of ER/Studio Data Architect, see the Team Server and Repository Installation and Upgrade Guide located at http://docwiki.embarcadero.com/TeamServer/ERStudio/202/en/Upgrading_Team_Server_and_Repository.

Note	Beginning with v18.5, ER/Studio Data Architect retains settings from previous versions.

WARNING	If you plan to use sample macros, first make a copy of the samples, and then store the copy in a folder separate from the ER/Studio installation folder to avoid overwriting your custom macros when upgrading to later versions of ER/Studio. Make sure that you make changes only to the copy of the sample macros.

Note	This procedure is for ER/Studio Enterprise installations only, which include the Repository. When you upgrade ER/Studio Data Architect, you must also upgrade ER/Studio Repository to correspond with the same build date as ER/Studio Data Architect.

License Enforcement: When working with the Repository, all users must use the same license type. ER/Studio now enforces this rule by marking the Repository as an Enterprise Team Edition Repository when a user with an Enterprise Team Edition license logs in. Subsequent log in attempts by users with a non-Team Edition license are denied.

Preparing to upgrade the Repository

User preparations

1. Ensure you have backed up all your .dm1 files.

2. Check in all your diagrams and objects. If you try to check in a file from the previous version with the same name as a current file, the current file will become corrupted.

3. Rename all local copies of Repository diagrams in the active file directory as follows:

   a. Choose **Repository > Options**.

   b. Make note of the Active File Directory.

   c. Switch to the Windows Explorer and rename all the diagrams in the Active File Directory. Alternatively, you can create a backup folder and move all dm1 files into the backup folder.

WARNING	Do not delete the RepoMachineToken.ini file in your Active File Directory.

Important notes

- Always back up the Repository database before upgrading.

- You can install both the current version and an earlier version of ER/Studio Data Architect on the same workstation.

See also

- [Release Notes](#)
- [Installing ER/Studio Data Architect](#)
- [Licensing](#)

Tutorials

The tutorials are intended to help you get started using IDERA's data modeling and database design solution, ER/Studio Data Architect.

After completing these tutorials, you'll have the foundation you need to explore the many features and benefits of ER/Studio Data Architect including how to:

- Create a new data model.
- Work with logical and physical diagrams.
- Leverage productivity-focused features such as powerful reporting engines.
- Use common tasks and commands to make you more productive.

The tutorials are divided into different sessions. Do them all at once or complete them individually as your time permits.

Choose a topic to work with:

- Getting started
- Logical and physical modeling
- Using data dictionary domains to populate new entity
- Establishing relationships between entities
- Creating and working with submodels
- Working with Business Data Objects
- Generating physical models from a logical model
- Working with Hierarchical Models
- Denormalizing the physical model
- How an entity maps to the physical model
- Documenting an existing database
- Documenting data lineage
- Creating a data flow
- Creating a data movement rule
- Defining external source and target systems
- Creating a data lineage and transformation visualization
- Diagram navigation and aesthetics
- Data Source Mapping

Once you have started, you can click Help from the Main menu to find many additional resources that complement and build on many of the activities shown in this brief guide. Or visit the Contents section of http://docwiki.embarcadero.com.

Tutorials: Getting started

Interface features

ER/Studio Data Architect has an enhanced user interface that is similar to Microsoft Windows with intuitive icons.

The following image names and describes the functionality of some key elements of the ER/Studio Data Architect user interface.

For information on enhanced UI features, please see the corresponding numbers below:
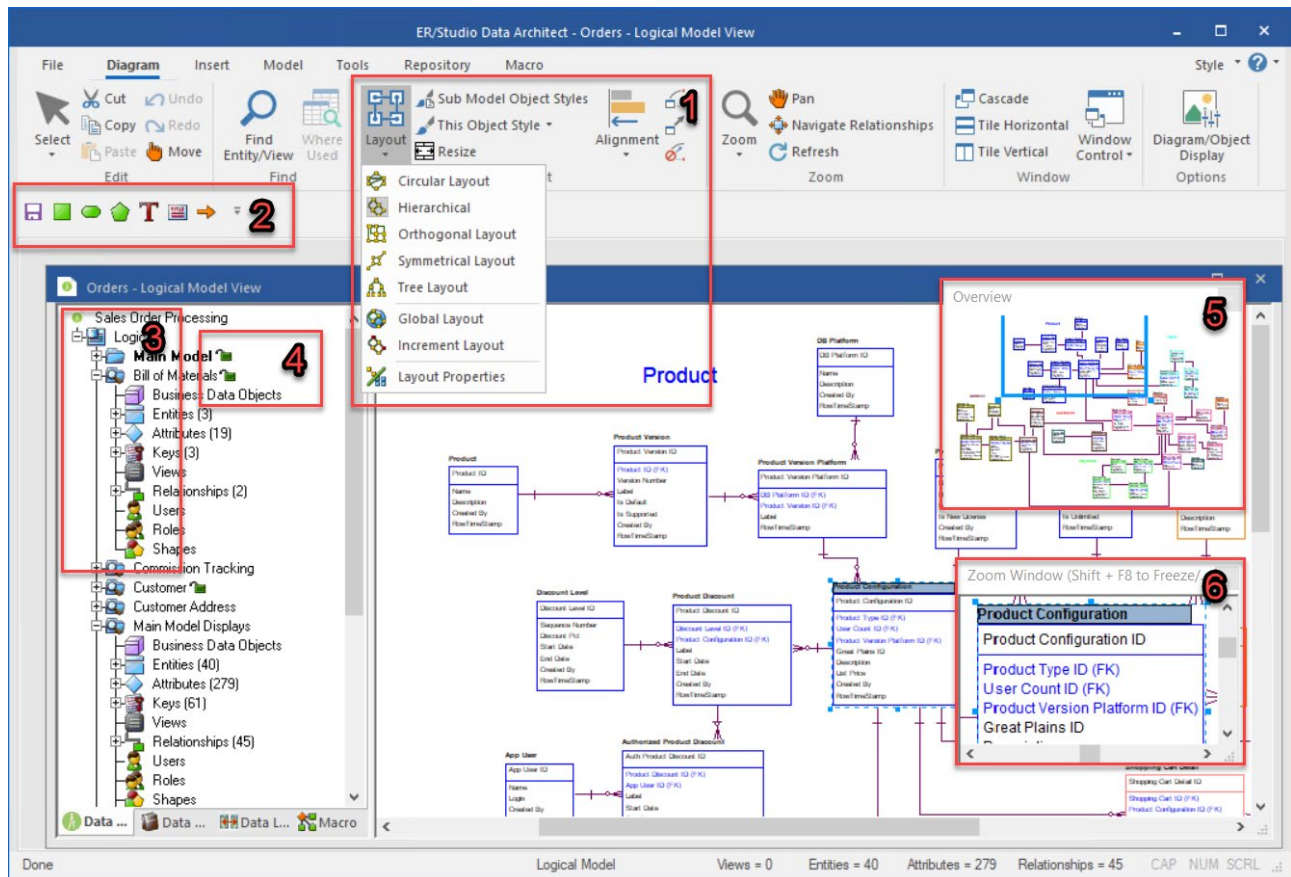
1. Sophisticated diagram auto-layout tools provide single-click clean-up of diagram objects.

2. Design elements and other often-used functions are dockable for quick access.

3. Data Model Explorer displays information about logical and physical models, submodels, and nested submodels.

4. Repository object status icons display real-time user access information.

5. The Overview Window lets you navigate large Data Models.

6. The Zoom Window helps you focus on the details of a specific area of a large, reduced diagram.

Starting to data model with ER/Studio Data Architect

1. On the **Windows Start > Programs** menu, choose **IDERA > ER/Studio Data Architect**.

2. Choose **File > New**. As you can see in the Create a New Model dialog, there are a number of ways to begin modeling with ER/Studio Data Architect:

   a. Build a new design from the ground up by drawing a new data model.

   b. Build a data model from an existing database through live reverse engineering.

   c. Import designs from other modeling products such as erwin or SQL files.

Tip	You can select an initial layout style for your model before the SQL import takes place.

3. For this example, select **Draw a new data model**, and then select **Relational** for the model type before clicking **OK**.

ER/Studio Data Architect now resembles the following image:
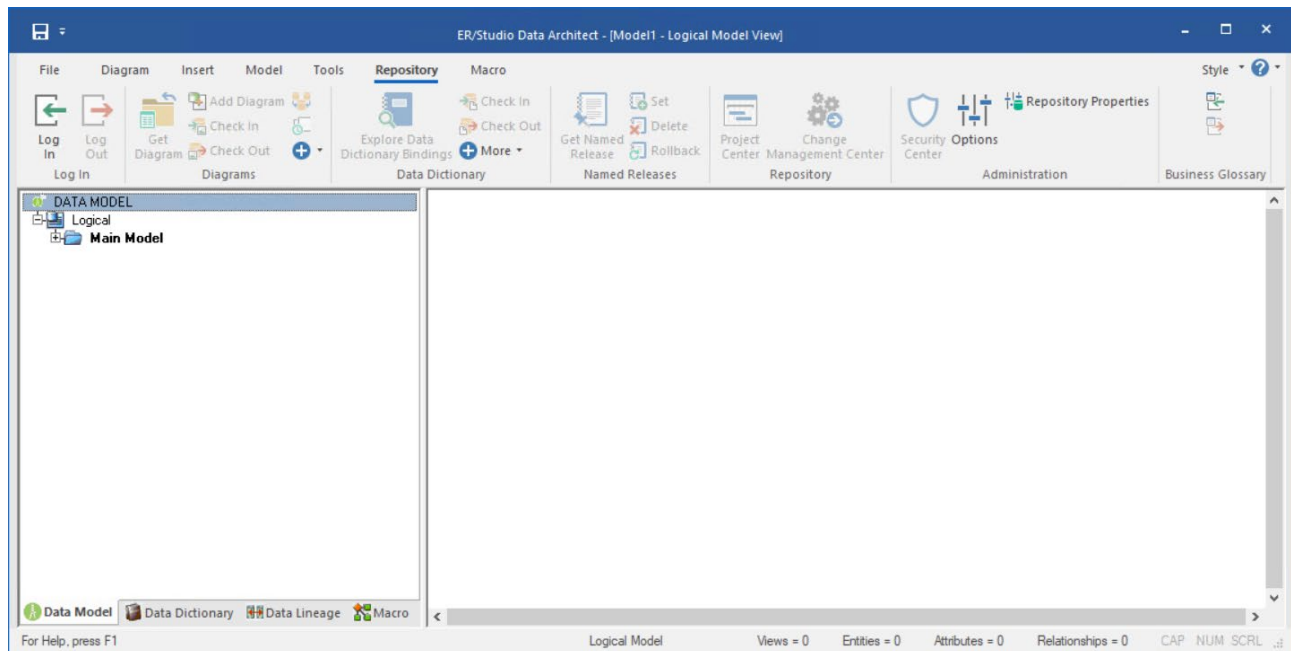
Tutorials: Logical and physical modeling

ER/Studio Data Architect supports both logical (non-DBMS or technology-specific) modeling and physical (DBMS-specific) modeling. ER/Studio Data Architect is designed to allow organizations the flexibility to analyze and design a business problem or application logically and generate as many different physical interpretations from the logical model as required. You can generate multiple physical from the logical model for the same DBMS (for example, Oracle) or other DBMSs (such as Oracle, SQL Server and DB2). Generating logical and physical models are discussed in detail in the following sessions.

- Using Data Dictionary Domains to Populate New Entity

- Establishing Relationships Between Entities

- Creating and Working with Submodels

- Working with Business Data Objects

- Generating Physical Models from a Logical Model

- Denormalizing the Physical Model

- Finding out How an Entity Maps to the Physical Model

# Tutorials: Using data dictionary domains to populate new entity

As instructed in Getting Started, you have chosen to draw a new data model to begin a logical model from the ground up. Before we begin to add entities, we will populate ER/Studio Data Architect with some sample domains.

What is a Domain? Domains are simply re-usable Attributes/Columns and are valuable tools in establishing standards. They allow data modelers to create a data element once (such as an ID field you require all of your entities to leverage as its primary key) which has the same data type, definition, rule, and constraint no matter where the data element is distributed and bound. Read more about Domains in ER/Studio Data Architect User Guide.

1. Choose **Tools > Import Data Dictionary**. The Import Data Dictionary Options dialog box appears.

2. Next to the File Location box, click the ellipsis and browse to the Sample Models folder, which is located at C:\Program Files\IDERA\ERStudioDA_X.X\Sample Models.

3. Double-click the **Orders.dm1** sample model so that it populates the dialog box, and then click **OK**. This model contains a pre-populated, sample data dictionary.

| Note	Under **Resolve Imported Objects with Duplicate Names**, you can choose between the options to determine how the dictionary objects are imported. This is important when importing into a diagram that already has dictionary objects in it.

Once opened, you will see that the ER/Studio Data Architect Model Explorer has automatically switched to the Data Dictionary tab to allow immediate drag-and-drop access to domains.
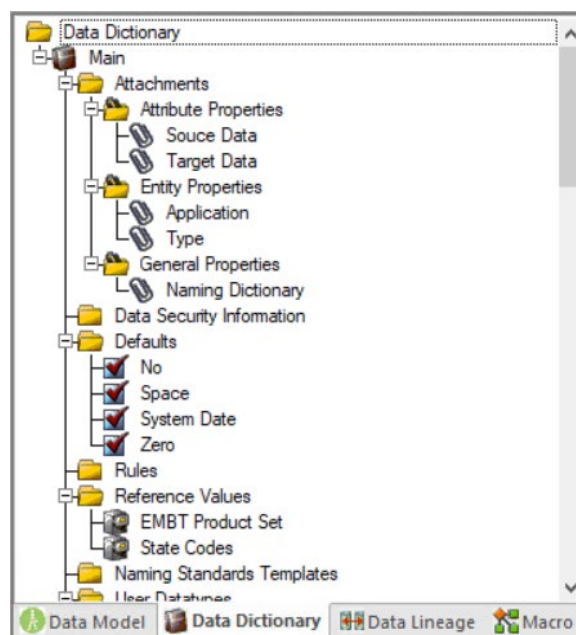
4. Now, to add an entity to the Data Model Window, click the **Entity** tool on the **Insert** ribbon, and then click in the Data Model Window to drop the entity.

Tip	The cursor will change to an entity symbol once the *Entity* tool is clicked, so you can click to drop as many entities on the Data Model Window as you want.

5. Right-click to return your mouse to the selection cursor.

6. Right-click the entity and click **Edit Entity** to open the Entity Editor. In the **Entity Name** field, type *Customer*, replacing the default entity name, and then click **OK**.

7. In the Domains folder of the Data Dictionary tab, locate the ID domain in the Numeric Domains folder.
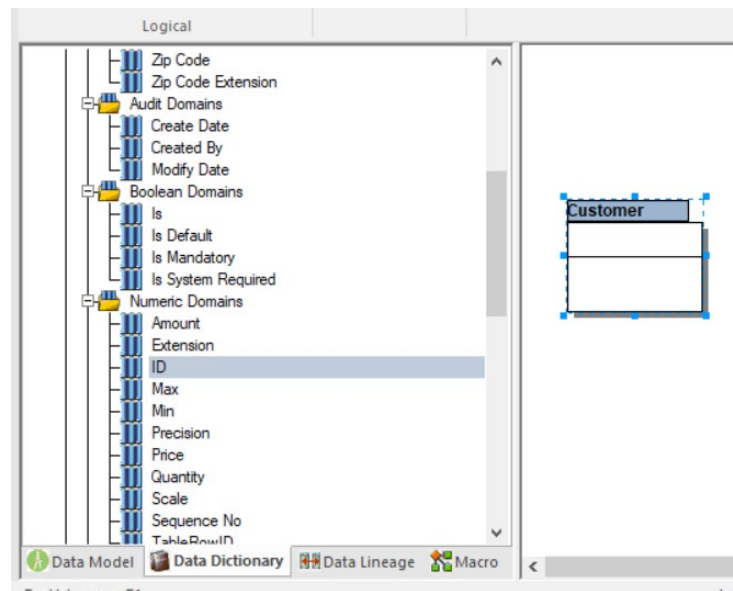
8. Click the ID domain (do not release your mouse), drag it onto the Customer entity, and then release it just below the entity's name, which is the entity's Primary Key field.

Tip	You can edit or rename an entity and insert, edit or rename attributes by holding down the Shift key and then clicking the name or attribute. Pressing the Tab key cycles between the entity's name, and primary key and non-primary key fields. After naming the field, press Return to insert a new field.

Tip	If you need to zoom in on the entity to read the entity name and attributes, press F8 to view the Zoom Window, then use the scroll bars of the Zoom Window to center the entity on the Data Model Window. Press Shift while rolling the mouse wheel forward to increase the view magnification. You can use the Pan tool to reposition the view to better see the entity.

9. Edit the entity again and in the Entity Editor, click **ID**, the name of the attribute we just created from the ID domain, and change its Attribute Name and Default Column Name properties to **CustomerID**, as seen in the illustration above. Then click **OK**.

10. Repeat the process in steps 7 and 8 to populate the Customer entity with the following domains:

    o **Name** and **Phone** from the **Property Domains** folder

    o **Address**, **City**, **State**, **Zip Code** from the **Address Domains** folder.

11. Drop another entity on the Data Model Window and call it **Order**.

12. Drag the ID domain onto the Order entity's Primary Key field; change the Attribute and Column name to **OrderID**.

13. Right-click the **Order** entity and select **Comments**.

14. Enter some text in the **Comments** editor, click **Add**, and then click **OK**. You can add comments to any database object in the same manner. This feature is useful when you want to provide feedback on a model to other users.

15. Save your data model. We will use it in the next session of this tutorial.

Tutorials: Establishing relationships between entities

1. On the Insert ribbon, click **Relationships > Non-Identifying, Mandatory Relationship**.
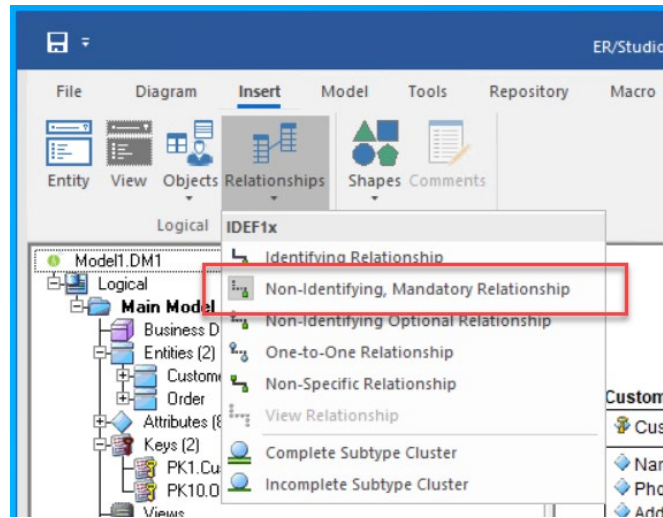
Tip	You can change the model notation by choosing **Model > Model Options**, and then selecting another notation option in the Notation area.

2. To establish a relationship between Customer and Order, click the parent entity, **Customer** and then click the child entity, **Order**.

Note	ER/Studio Data Architect supports sound design practices by automatically propagating the primary key, from parent to child entities. If there are candidate alternate keys that you want to propagate to the child, choose all available parent entity keys in the **Parent Key** list in the Relationship Editor. Deleting a relationship removes a non-native propagated attribute. However, if you want to keep the child columns of the relationship or foreign constraint, when you delete the relationship check the **Make Foreign Keys Native** option. In this case, if the relationship between Customer and Order is deleted, the CustomerID remains in the Order entity.

3. You can edit the relationship properties by right-clicking the relationship, and then selecting **Edit**. You can set a variety of information about the relationship in Properties, or you can forward and reverse phrases. To set phrases, click **Phrases**, and then set the forward and reverse relationship phrases.

Note	ER/Studio Data Architect lays out the phrases as sentences to confirm the context of the relationship.
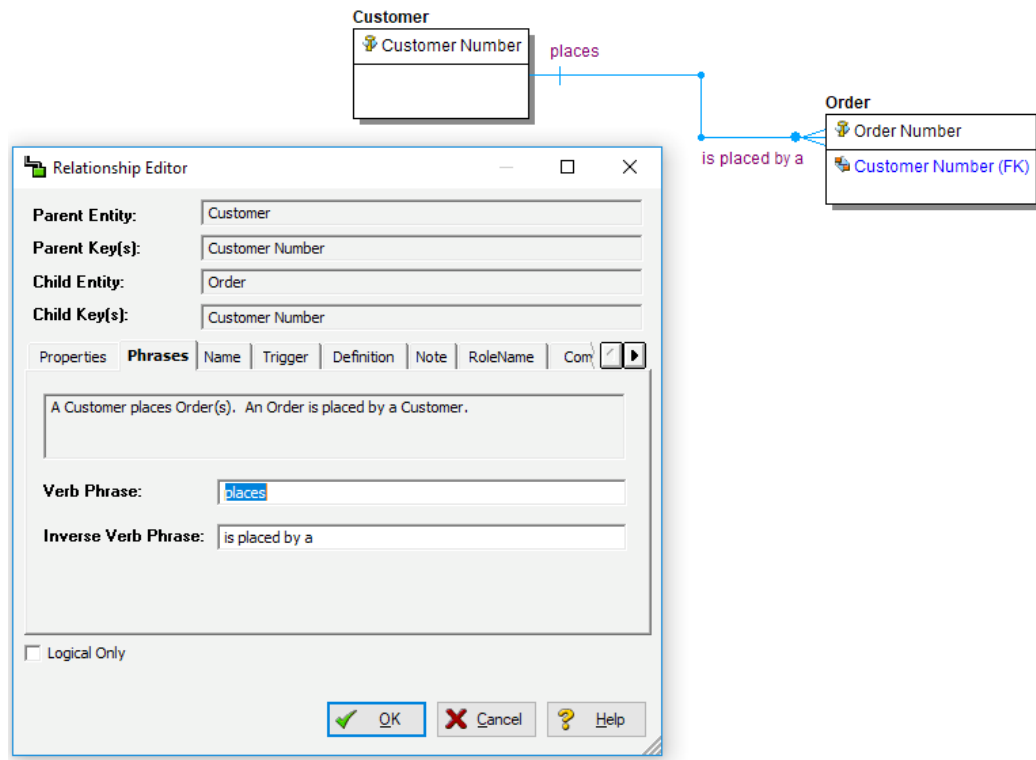
4. Create two relationships between Order and Address. Where you have multiple relationships between entities, the propagated foreign key is identical. When you create a second relationship between two entities, ER/Studio Data Architect opens a dialog to ask you to set role names which will generate unique foreign keys.

5. Set the Logical Rolename and Physical Name to *Shipping Address*.

Tutorials: Creating and working with submodels

Now that you have a general understanding of how to build logical models from the ground up in ER/Studio Data Architect, it is important to understand how to work with an important navigation feature of ER/Studio Data Architect, called submodels. Submodels and nested submodels are designed to break down large, complicated views of a data model in order to focus on a specific area. An important aspect of submodels to understand is that any changes made in the submodel, other than layout, color, display settings, notation or similar items which are unique to the submodel, will occur automatically in the Main Model view. In other words, change or add an attribute to an object in a submodel and the change is automatically propagated to its Main Model counterpart. Close the current sample model, and open a more mature model. Use this exercise to learn more about submodeling.

1. Choose **File > Open**.

2. Select **Orders.dm1**, and then click **Open**.

3. To preserve this sample model for future use, choose **File > Save As** and then save the Orders.dm1 file with a new name.

   In this exercise, we will modify this new model, so be sure to open the newly-named file rather than Orders.dm1 before continuing.

4. Collapse the folders in the Data Model tab of the Data Model Explorer to look like the image below:
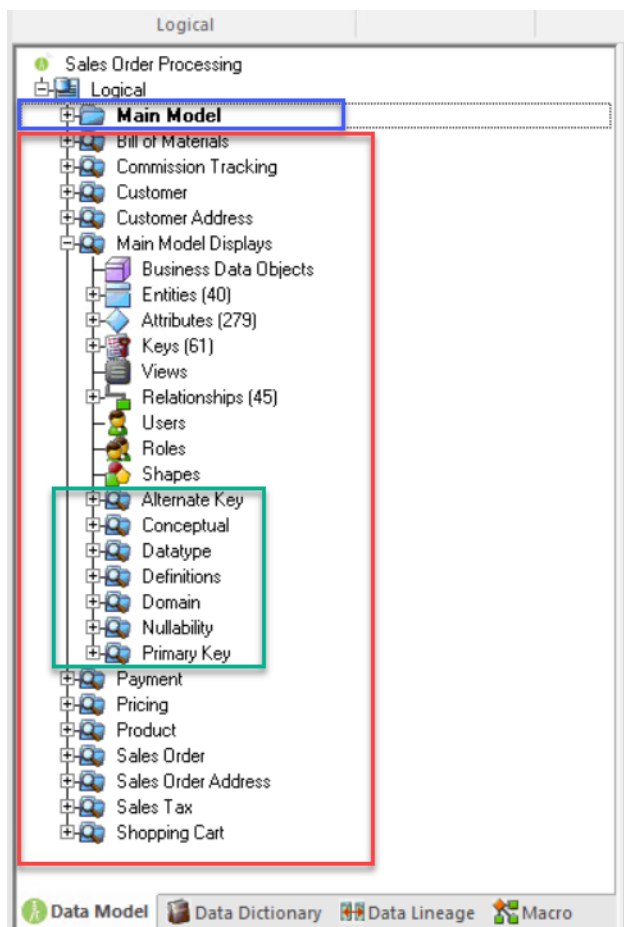
In the sample model, there are no physical models. This model includes several submodel folders that help to describe the logical model:

- o Main Model – This is the entire collection of all logical objects in the sample file. Note the absence of the magnifying glass on the folder icon which designates it as the main model.

- o Bill of Materials through Shopping Cart – These are submodels, which are smaller collections of entities derived from the Main Model that help to describe specific areas of the Main Model free from other entities.

- o Alternate Key through Primary Key – These are nested submodels, which can go 'n' levels deep and are literally submodels of submodels.

Feel free to explore. Click the plus sign (+) to expand these folders.

Let's create a submodel with all the objects related to the sample file Address components.

5. To make a new submodel, navigate to the **Main Model** and with the CTRL key depressed, click the objects in the Data Model Explorer, as seen in the following image.
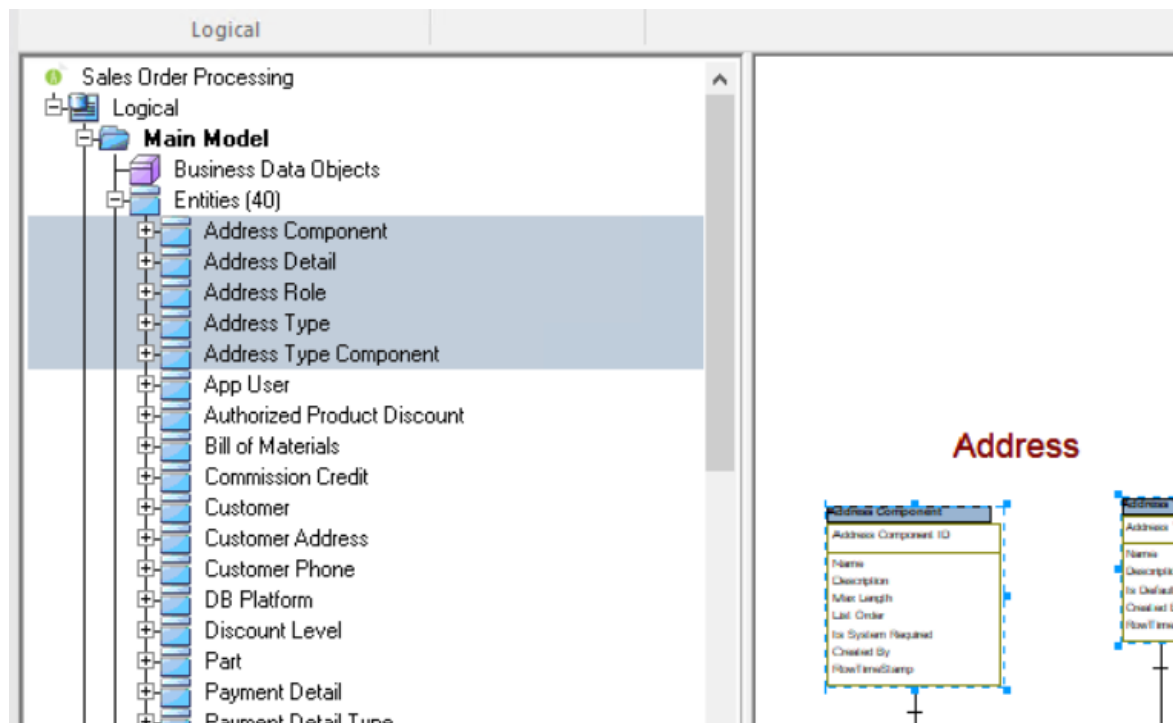

Note	Any objects selected in the Data Model Explorer also are selected on the Data Model Window. You can also select entities by lassoing them on the Data Model Window.

6. With the entities selected, choose **Model > Create Submodel**.
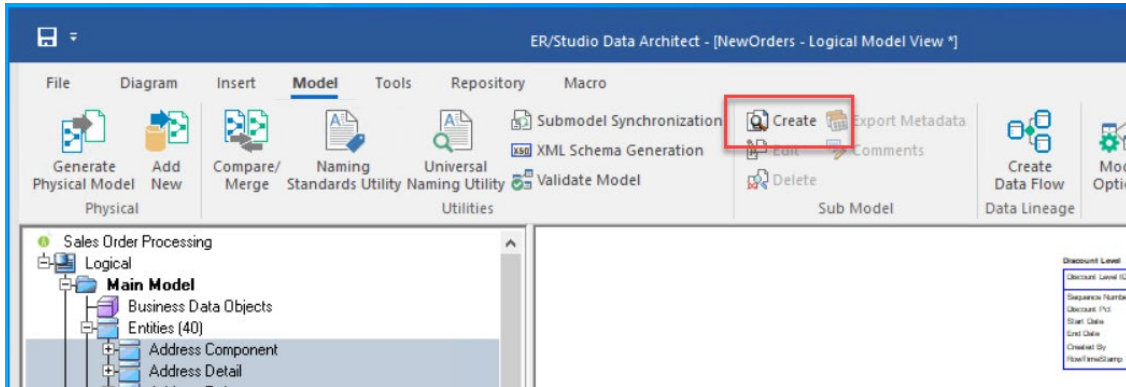
7. In the Create Submodel dialog box, type **Address Components** as the name for the submodel, and then click **OK**.

   ER/Studio Data Architect creates the Address Components submodel.
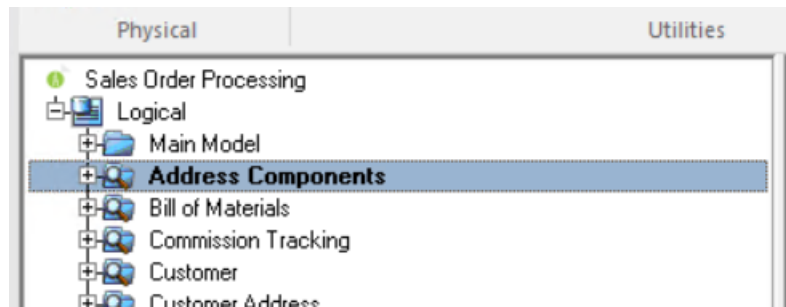

What do the results look like and how do I navigate to the submodel?

Once created, you will see the new submodel listed in the Data Model Explorer, denoted as a submodel by the magnifying glass over its folder, as in the case with Bill of Materials and the other submodels.

Tutorials: Working with Business Data Objects

The logical model's primary purpose is to communicate the content of existing data assets and requirements for new data assets with the consumers of data assets. The entity object does this as well, but has the job of an intermediary with the physical data model. We also have the Business Data Object (BDO). This object type allows us to group entities that cover a particular topic or business concept to make the logical model easier to understand. Use this exercise to learn more about Business Data Objects.

1. In the Logical model, look at the group of entities around the text "Customer." These entities all apply to the concept of Customer and can be grouped into a BDO.
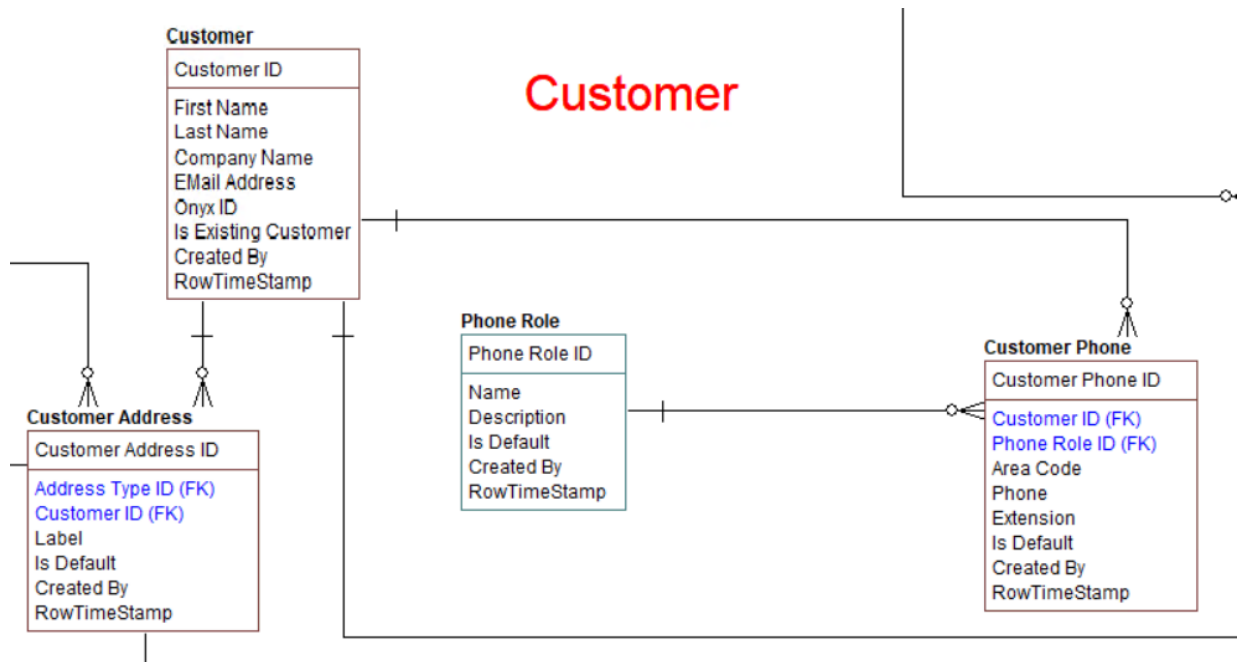
2. Create a Business Data Object by accessing the **Insert** ribbon and clicking **Objects > Business Data Object**, and then clicking in the model area to place the BDO. Be sure to put the object near the Customer information for ease of grouping in the next step.

3. Rename the BDO to "Customer" and then resize the BDO so that it covers the following entities:

   a. **Customer**

   b. **Customer Phone**

   c. **Phone Role**

   d. **Customer Address**

4. Select these entities inside the BDO, and then right click **Add to Business Data Object**.
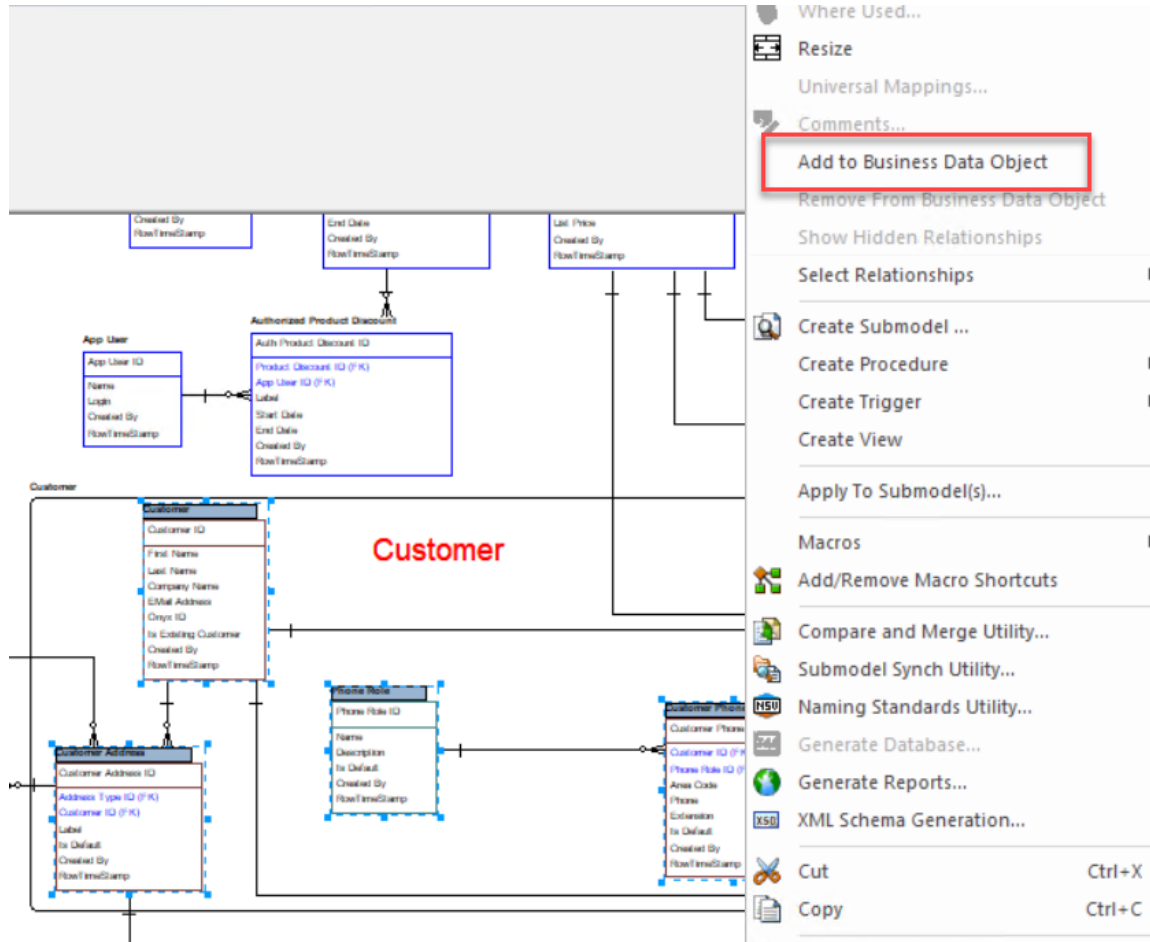
5. Open the Business Data Object Editor by double-clicking the BDO. Notice that all the entities are included in the list.

6. Select the **Customer** entity, and then click **Set Anchor**. This states that the primary entity for the BDO "Customer" is the entity "Customer."

7. *Optional*. Set a definition for the BDO as well as a physical name.

8. *Optional*. Add attachments.

Tips	• You can simplify the diagram by double clicking the minus sign to hide the entity content.
	• Whenever you move the BDO, the entities move with it.
	• You also can see the BDO and its contents listed in the tree view on the left.

Tutorials: Generating physical models from a logical model

ER/Studio Data Architect can generate as many physical models from a single logical model as desired. There are many ways to leverage multiple physical models in ER/Studio Data Architect to help the design process. Examples of how multiple physical models are used are:

- **Managing change in an existing application**: Maintain independent development, test, and production physical model diagrams that represent specific databases.

- **Migrating database applications**: Use ER/Studio Data Architect as an analysis and design hub for migrating database applications. Manage a physical model of the legacy source database application in addition to its new target physical model, which can be for an entirely new DBMS than originally maintained in the legacy database.

Let's generate a new physical model from a logical model in order to build a database. We will use the sample model.

1. Open your version of the Orders.DM1 sample model.

Tip	Use the steps shown in the first session to do so.

2. Select the **Main Model**, and then on the Model ribbon, click **Generate Physical Model**.

   ER/Studio Data Architect invokes a step-by-step wizard to walk you through the process of generating a DBMS-specific physical model.
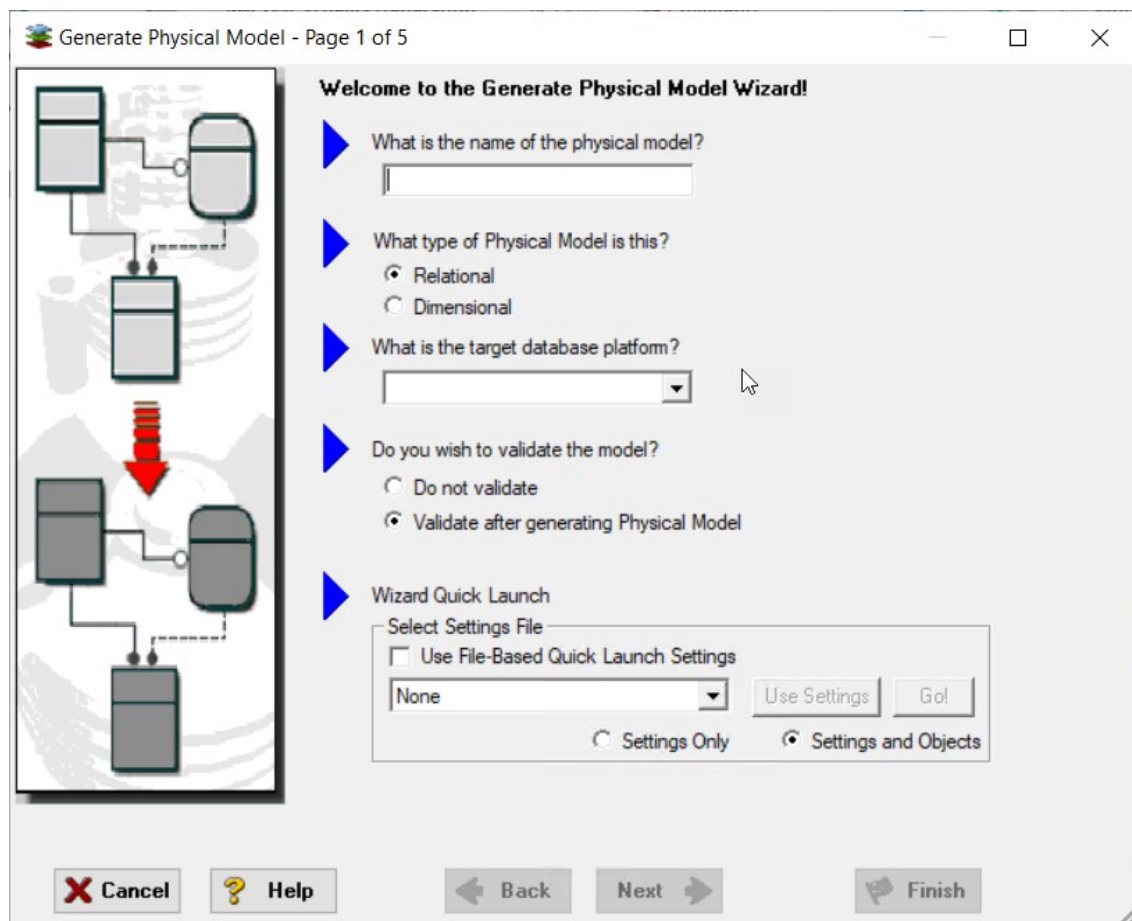
3. Name the new physical model, **DB2 Physical Model**, and then select **DB2 UDB for OS/390 11.x** as the target DBMS to generate.

4. Continue through the Generate Physical Model wizard, which prompts several questions about how you want your physical model generated.

Note	The wizard prompts you to customize items such as individual object selection, index assignment, default storage parameters, resolution of many-to-many relationships that may be in the logical model, and naming conventions. In Step 3 of the wizard, you can add prefixes or suffixes to the object name. For example, it can be useful to add an indicator of the object type in the name. A DBMS-specific validation check is also provided in this wizard.
Tip	The Quick Launch can store common settings so that an operation can be reused on this model or on any other models. You can reuse the settings on another model by choosing the **Use File-Based Quick Launch Settings** option when saving the Quick Launch information on the last page of the wizard.

5. To generate the new Physical Model, on the last page of the wizard, click **Finish**.

Now that a physical model is generated from the logical model, feel free to navigate to specific objects via the Data Model Explorer, such as the CUSTMR table selected here. Double-click and view the physical details of the object such as DDL, Indexes, Partitions, and Storage.


Tutorials: Working with Hierarchical Models

ER/Studio models information using Entity Relation Diagrams. You can convert these diagrams directly into relational models that represent relational databases. Data assets such as MongoDB, JSON, and Google BigQuery have hierarchical structures. These hierarchical structures also employ denormalization to improve performance with repeated substructures.

Data Architect converts relational logical models to denormalized hierarchical structures and vice versa using feature wizards including Generate Database, Reverse Engineer, Import from File, and Compare/Merge. To make this process repeatable, logical data model are enhanced to include additional information to help with this process.

Specifying Containment in the Logical Model

In the following logical data model diagram, notice that some lines display **Parent contains Child** or **Child contains Parent**. This allows you to specify how objects will be embedded in any hierarchical physical models you generate from it. This also drives both the Generate Physical Model and Compare/Merge wizards.
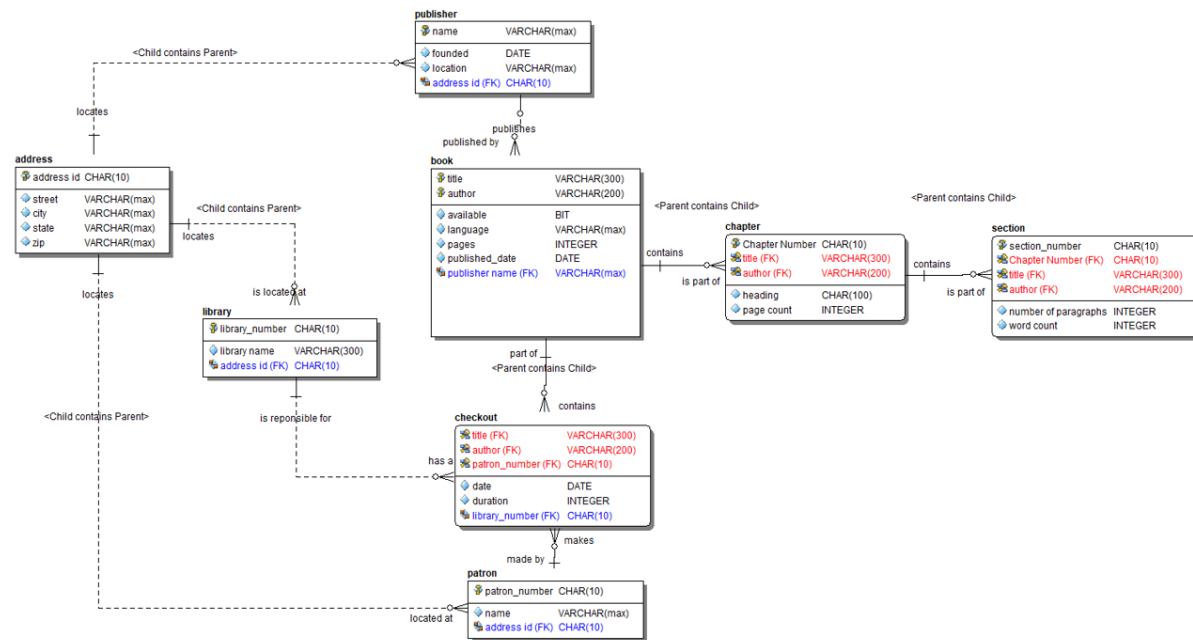
You can set two properties in the Relationship Editor to control this relationship.

- **Containing**. Select this option to contain one object within another as a Nested Object.

- **Parent is Container**. Select this object to denote the direction of containment.
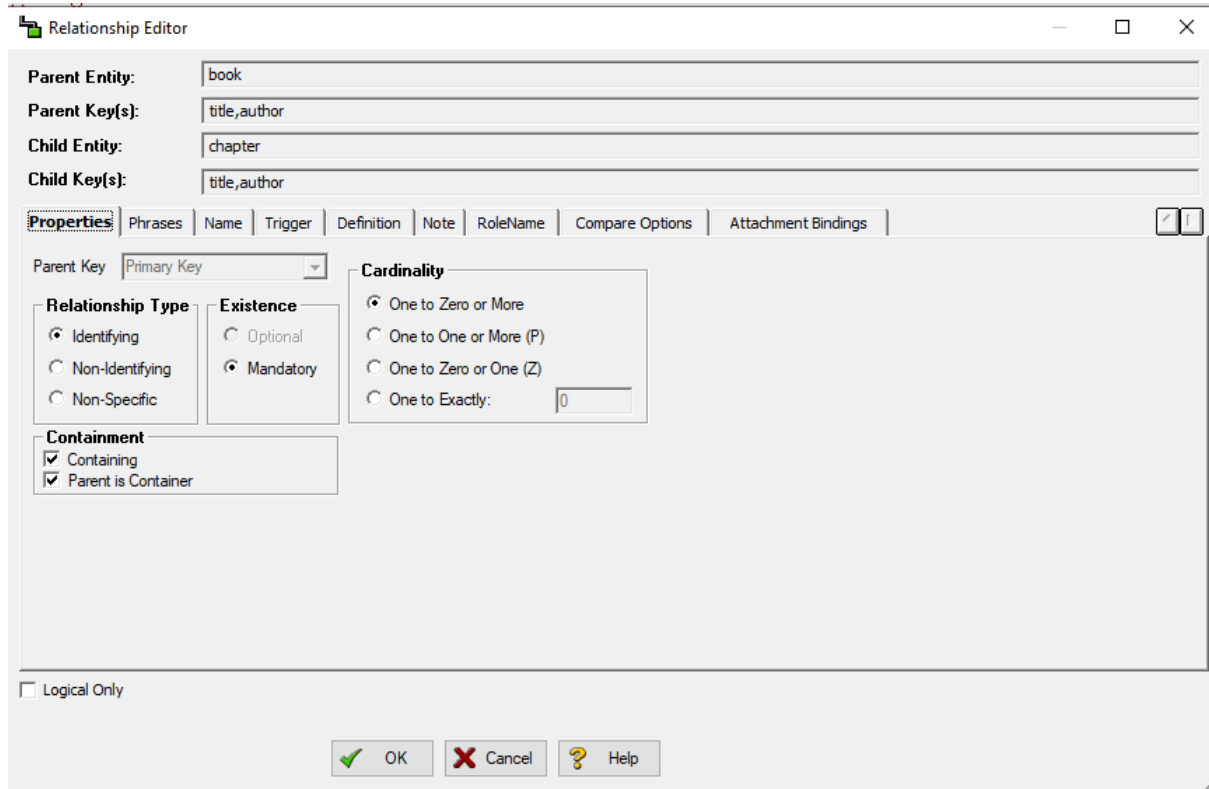
Generating the Physical Model (MongoDB)

The Generate Database wizard differs for each target platform, but for MongoDB models, there are the options as shown in the following image.
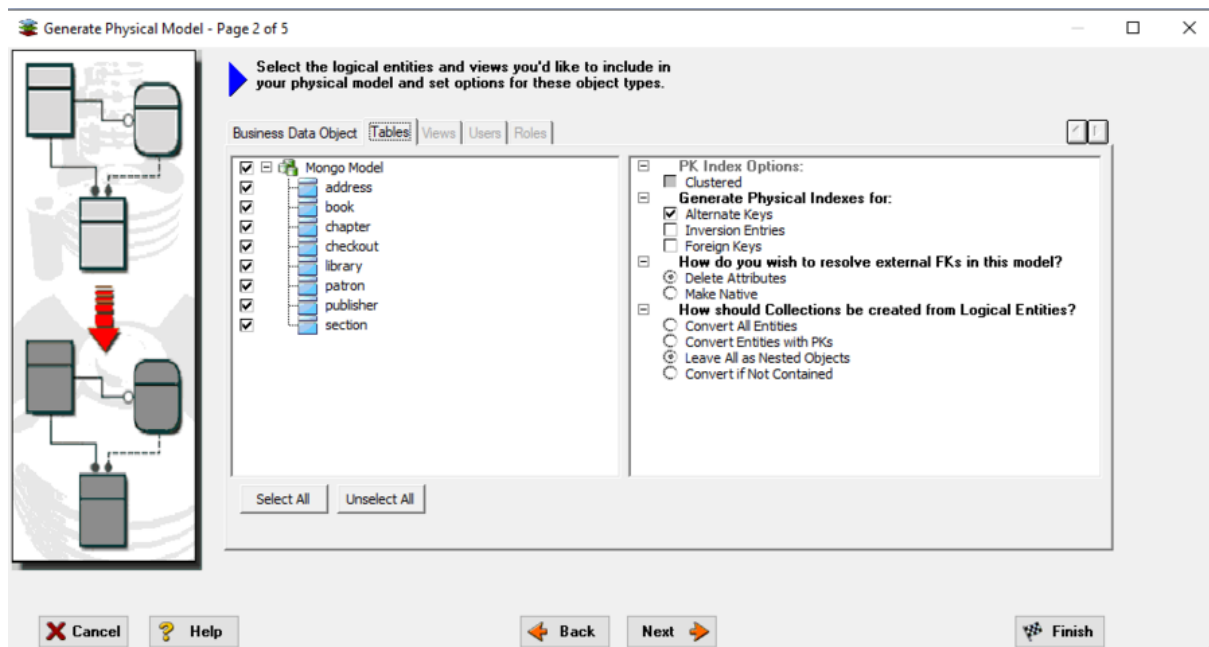
You can remove propagated Foreign Key properties from generated objects or make them as native fields, which may be useful later to connect objects together elsewhere via code.

The **How should Collections be created from Logical Entities** option lets you choose how you want to create Collections. The following options are available:

- **Convert All Entities**. Use this option to make all logical entities into Collections.
- **Convert Entities with PKs**. Use this option to make all logical entities with primary keys into Collections.
- **Leave All as Nested Objects**. Use this option to make all logical entities into Nested Objects.
- **Convert if Not Contained**. Use this option to take a rules-based approach as detailed below.

Understanding the Convert if Not Contained option

If you select **Convert if Not Contained**, the generator wizard uses the following steps:

1. For each entity in the logical model, create an object. Primary key attributes become standard fields.
2. Logical relationship lines generate either a referencing relationship or a containment relationship.
3. For each generated object, the wizard reviews whether the object has a containing relationship from any other object. If it does, it becomes a Nested Object. If not, it becomes a top-level Collection.
4. All Collections are given a surrogate primary key "_id".
5. For referencing relationships, primary keys are propagated.

For the above logical model, we would generate the following model using this logic.
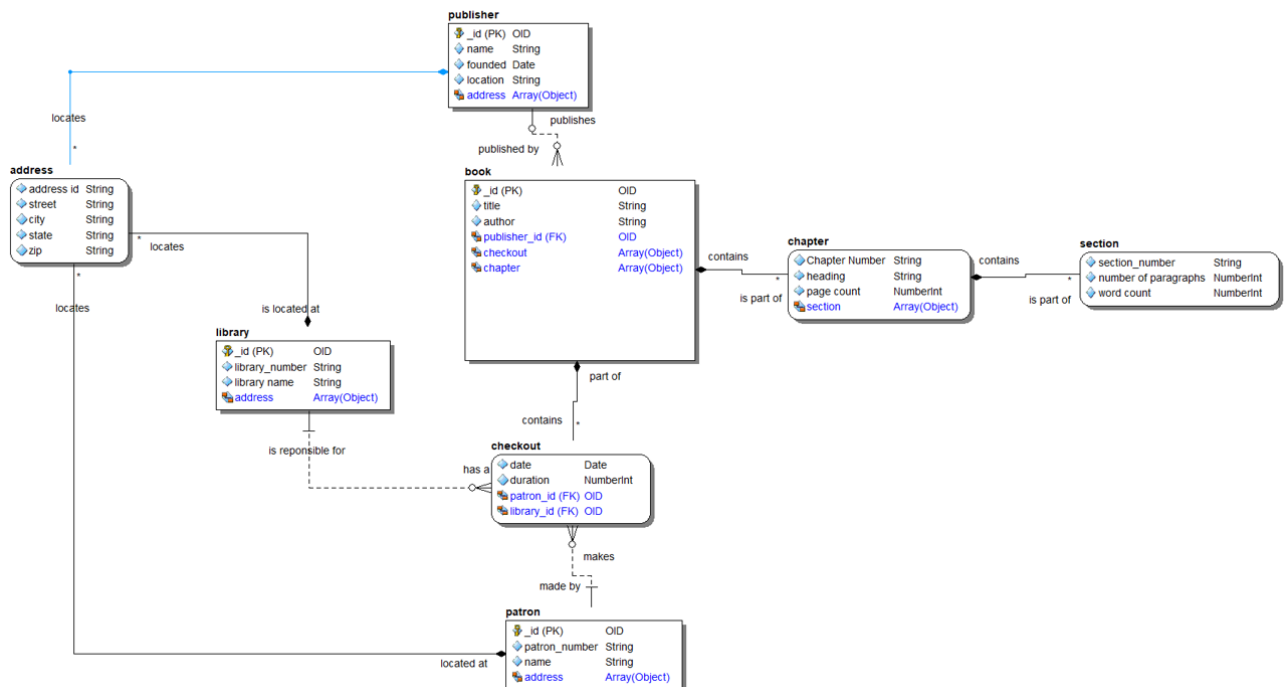
Notice the following traits in the above model:

- Note the borders of the objects. Collections have square corners and Nested Objects have rounded corners.
- We created four Collections: **book**, **patron**, **library**, and **publisher**. The other entities have generated Nested Objects.
- The entities **section** and **chapter** are nested inside **book**.
- **address** is fully denormalized as nested objects inside **publisher**, **library**, and **book**.
- **checkout** is nested inside **book**, but retains references to the Collections **library** and **patron**.

Understanding why you didn't get what you wanted

Once you generate your physical models, you will see a structure of objects with Collections and Nested Objects based on the rules used by the wizard. If you don't get what you expect, you have the following options:

1. Modify the Containment properties in the logical model and use Compare/Merge between the logical and physical to propagate the changes.
2. Modify the Physical model directly. Use the Switch Object Class tool on the Insert tab of the ribbon bar to convert a Collection to a Nested Object and vice versa. You can then propagate changes back to the logical model.

Using the rolled-up view

The object view is useful to understand the reuse of objects and to edit the cardinality to Nested Objects. You can also select a "rolled up" rendering of the model that makes it easier to visualize the hierarchical nature of the model.

To access this view

1. On the Diagram tab of the ribbon bar, select **Diagram/Display Options**.
2. On the Object tab, select **Roll Up Contained Objects**.

The MongoDB model we generated above now looks like this:
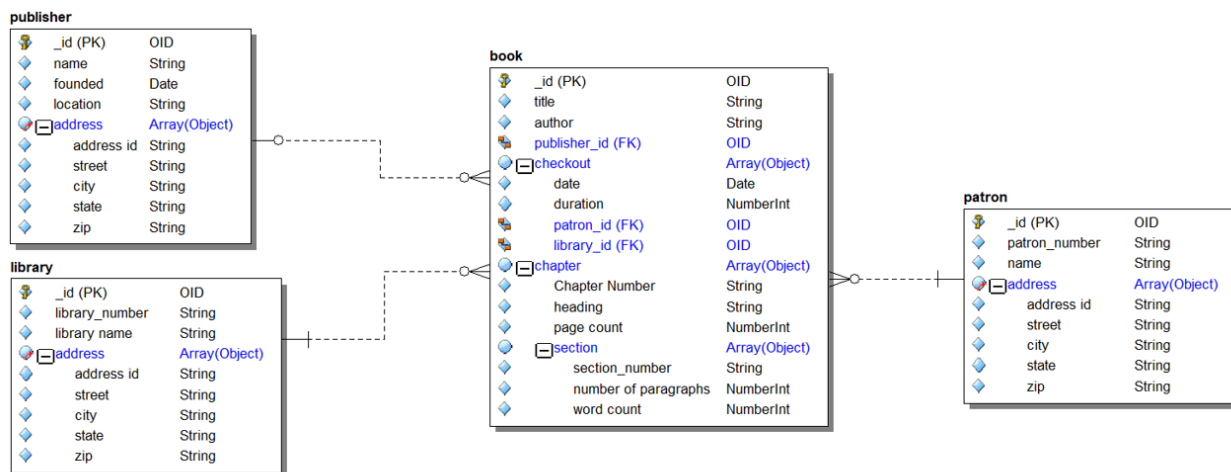
Notice that the Nested Objects are rolled up into the parents.

Rapidly creating a Hierarchical Physical Model

You may want to bypass the process of creating a logical model and jump straight to creating a physical model. You can use the compare/merge functionality to later create your logical model.

To rapidly create a Hierarchical Physical Model

1. Create a new model, and then add an entity into the logical model.

2. Generate a Physical model for the MongoDB platform.

3. Insert objects onto the physical model directly using the Insert tab on the ribbon bar.

4. Insert fields directly into the objects using the keyboard by pressing Shift while right-clicking inside of the object. This action opens a text box.

5. Type the name of the field, and then press the Enter key. A dialog appears allowing you to select the data type of the field.

6. Select string, and then press Enter to create the field. A new text box appears.

7. Type a name, and then press Enter.

8. In the datatype selection box, select **Object**, and then select the array checkbox.

9. Press Enter. A dialog appears allowing you to create a new reusable Nested Object.

10. Click **Yes**.

11. In the next new field, specify a name for the new field but before you press Enter, press the Tab key. This action indents the field, making a field of the object you just created. Now press Enter, and then specify the data type of the field.

As you build up your model, you can move or copy fields across the model using the Move tool to move fields or with the Control key to copy fields.

Tutorials: Denormalizing the physical model

ER/Studio Data Architect comes equipped with denormalization wizards to help you optimize the physical design once the physical model is generated. The wizards help automate the process and keep the ties between the physical tables and the logical entities.
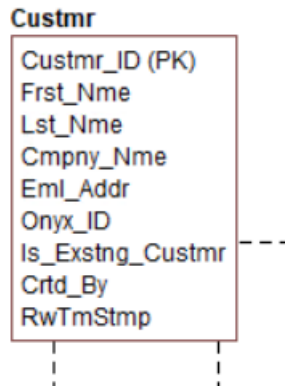
The active, denormalization wizards available depend on which tables are selected in the physical model when you select **Denormalization Mapping**. For example, if two tables that are related to each other are selected, the valid operations would be **Rollups** or **Rolldowns**.
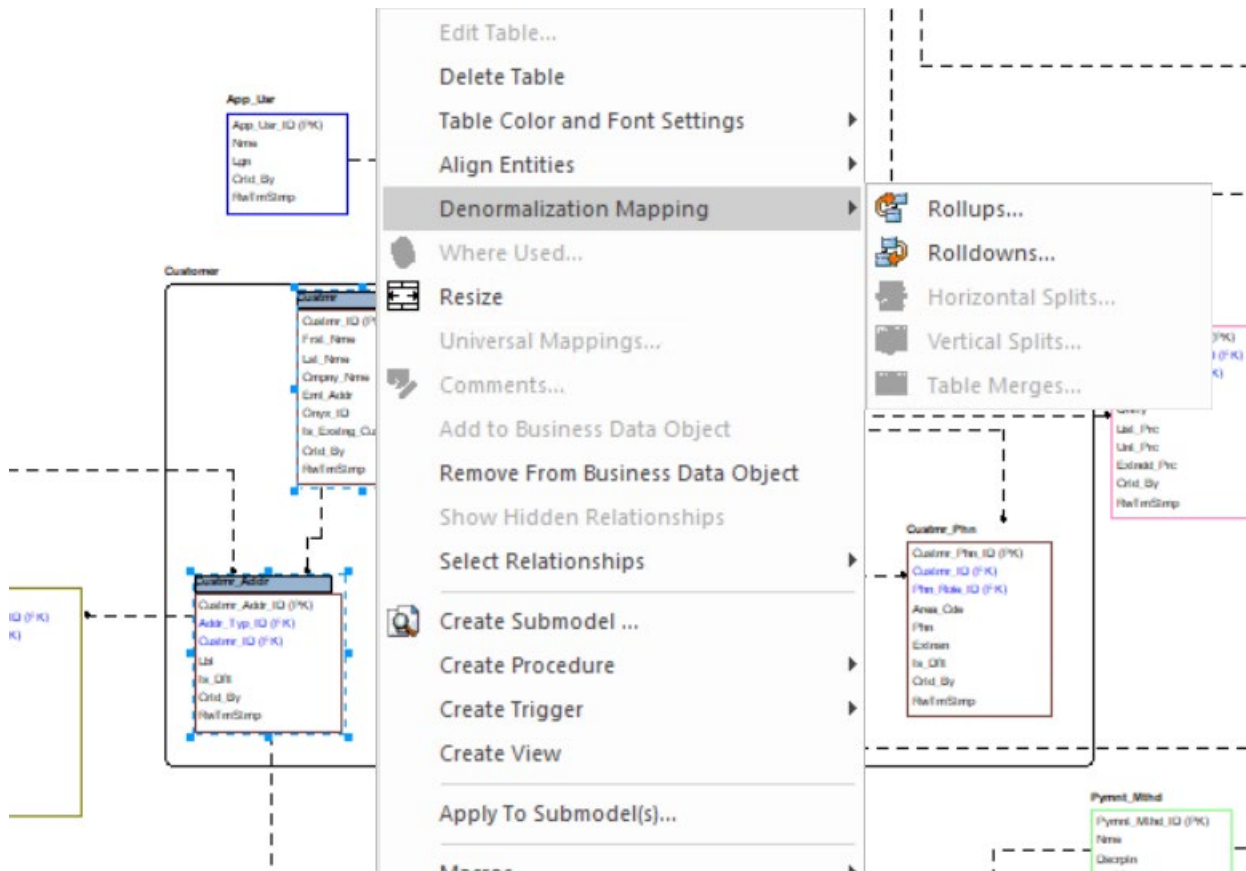
When only one table is selected, the Denormalization Mapping options **Horizontal Splits** and **Vertical Splits** are available for you to split the table accordingly. The **Table Merges** option is available when two unrelated tables are selected.

Let's walk through an example of a denormalization operation using the generated physical model in a previous session of this tutorial. We may want to reduce the overhead on the Custmr table by splitting it into two physical tables, Custmr_East and Custmr_West. Splitting the table can reduce the query time and provide opportunities to store the tables at different locations which could further reduce lookup time.

Before the operation, the Custmr table looks like this:

Custmr

Custmr_ID (PK)
Frst_Nme
Lst_Nme
Cmpny_Nme
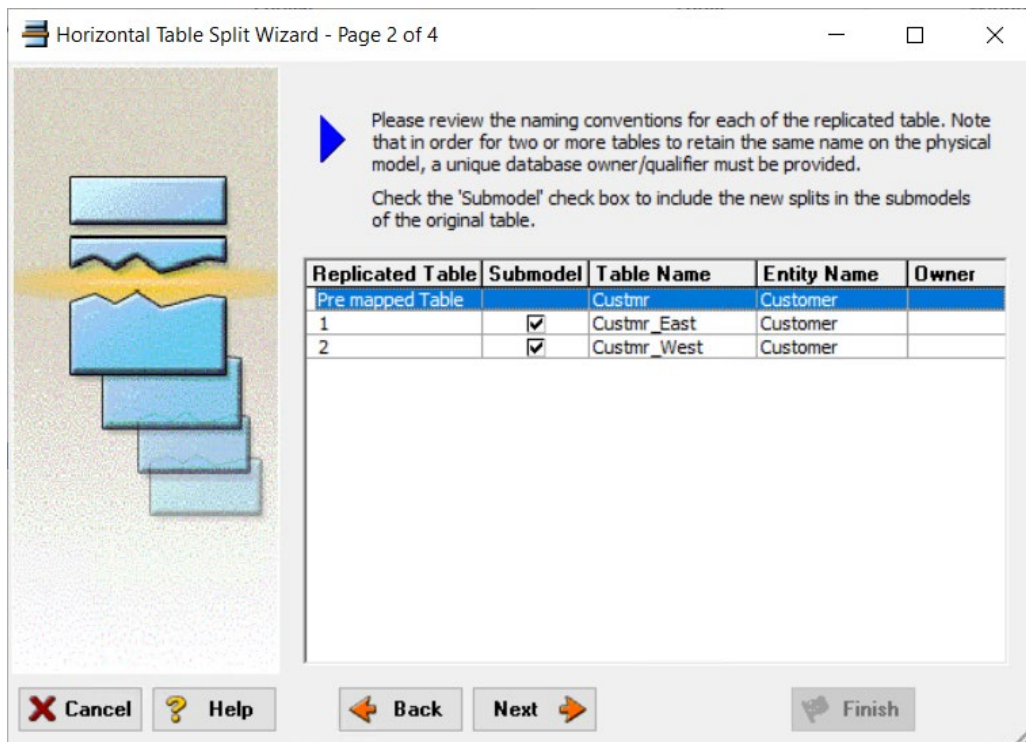Eml_Addr
Onyx_ID
Is_Exstng_Custmr
Crtd_By
RwTmStmp

1. Open the Orders1.dm1 model you modified and saved in the first session.

2. In the Data Model Explorer, right-click the **Custmr** table in the Physical Model.

3. Choose **Denormalization Mapping > Horizontal Splits**.

   Notice that since only Custmr is selected, the only possible mappings are vertical and horizontal splits.
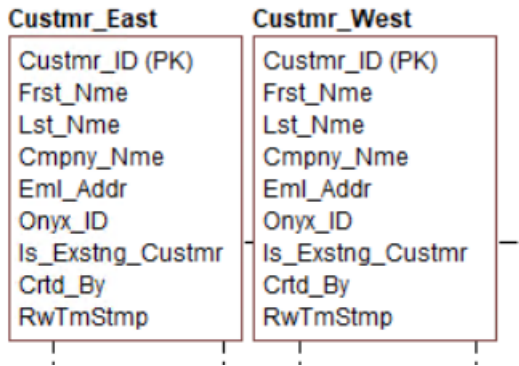
   The Horizontal Table Split Wizard launches.

4. On Page 1 of the wizard, type **2** for the number of splits, and then click **Next**.

5. On Page 2 of the wizard, rename Custmr_1 and Custmr_2 to **Custmr_East** and **Custmr_West**, and then click **Next**.

Horizontal Table Split Wizard - Page 2 of 4   —   □   ×

Please review the naming conventions for each of the replicated table. Note that in order for two or more tables to retain the same name on the physical model, a unique database owner/qualifier must be provided.

Check the 'Submodel' check box to include the new splits in the submodels of the original table.

Replicated Table	Submodel	Table Name	Entity Name	Owner
Pre mapped Table		Custmr	Customer	
1	✔	Custmr_East	Customer	
2	✔	Custmr_West	Customer	

✗ Cancel   ? Help   ← Back   Next →   ♥ Finish

6. On Page 3 of the wizard, we will keep all of the relationships, so click **Next**.

7. On Page 4 of the wizard, verify the name and definition for the denormalization operation, and then click **Finish**.

Finished! The Custmr table now is split into two physical tables that look like this:

Custmr_East

Custmr_ID (PK)
Frst_Nme
Lst_Nme
Cmpny_Nme
Eml_Addr
Onyx_ID
Is_Exstng_Custmr
Crtd_By
RwTmStmp

Custmr_West

Custmr_ID (PK)
Frst_Nme
Lst_Nme
Cmpny_Nme
Eml_Addr
Onyx_ID
Is_Exstng_Custmr
Crtd_By
RwTmStmp

Notice that the two tables are identical except for the name. You can selectively choose which attributes are included in the resultant tables by using a vertical split.

The denormalization mapping is stored with each of the tables.
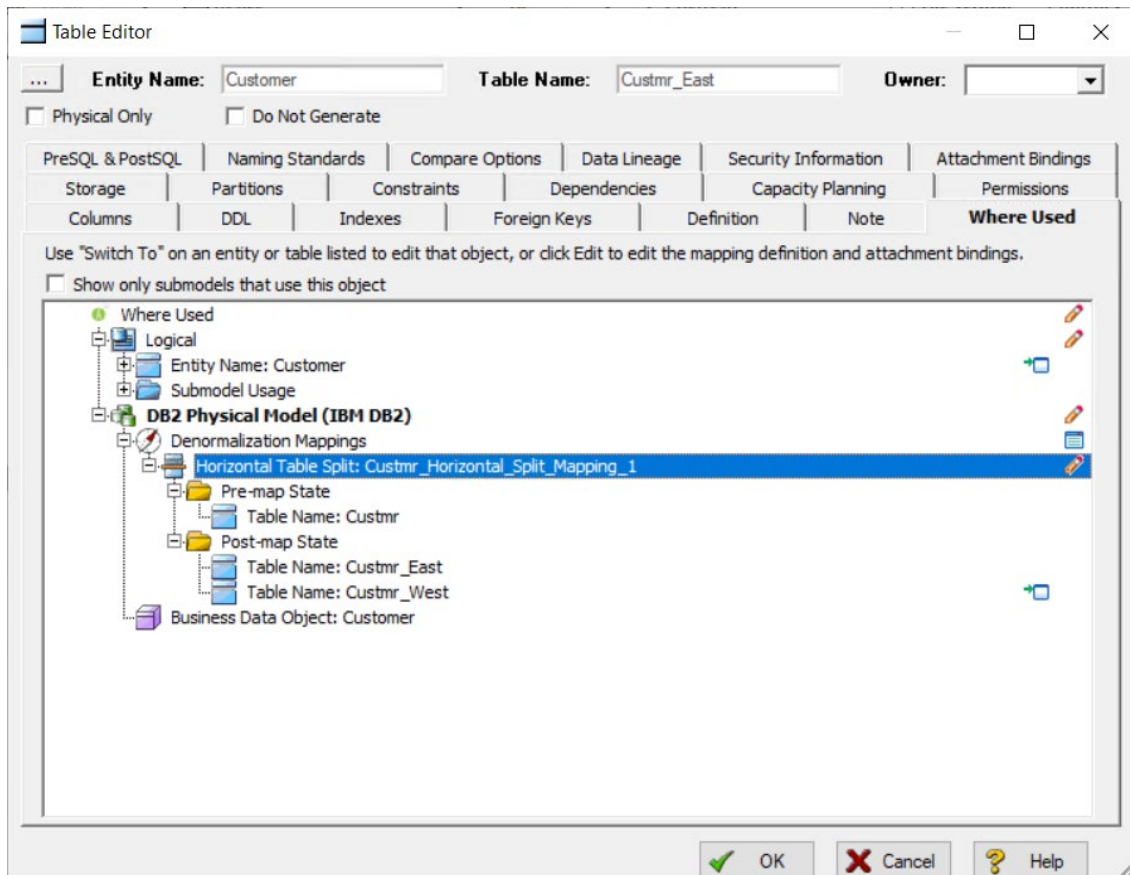
You can use the denormalization information to undo the operation or see the history of what happened. ER/Studio Data Architect tracks the before and after states of these operations. This comes in handy in the next section where we discuss the Where Used analysis that can be performed between the logical and physical models.

Tutorials: How an entity maps to the physical model

Now that we performed a denormalization operation, the logical entity, Customer, essentially has become two physical tables, Custmr_East and Custmr_West. The ties between the logical and physical models are not lost. ER/Studio Data Architect allows you to see what Customer in the logical model maps to in the DB2 physical model.

Let's take a look at the Customer entity in the logical model.

1. In the Data Model Explorer, go back to the Customer entity in the Logical model.

2. To start the Entity Editor, double-click the **Customer** entity, and then click the **Where Used** tab.
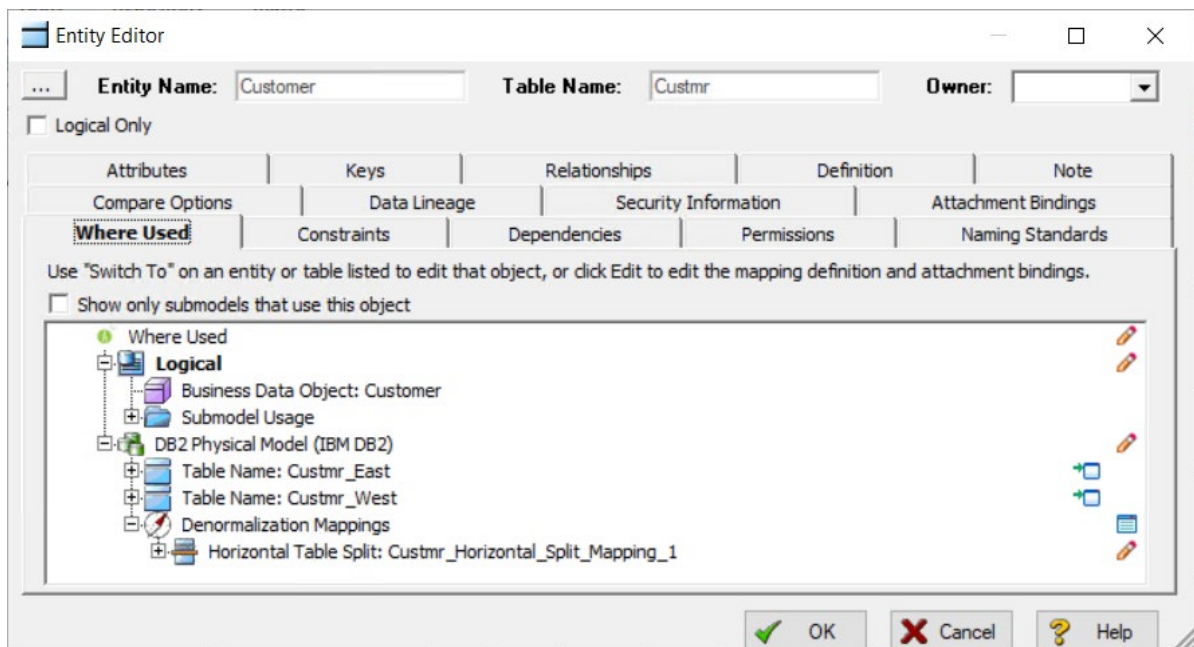
In the expanded tree, you can see the lineage of what happened to the object. Notice that Custmr_East and Custmr_West are listed as physical implementations of the Customer entity. The denormalization mapping object shows how the end result was achieved. The Where Used tab displays the submodel usage of a particular entity within the logical or physical model, which allows you to see which business areas the entity belongs to.

Note	Where Used information is also available for attributes and columns.

Conclusion

In this session, you have seen how incredibly quick and easy it is to:

- Build a logical data model from scratch.
- Create a new submodel view to understand how to model on specific parts of a larger Main Model.
- Generate a physical model from a logical database in preparation for building a new database.
- Denormalize objects in the physical model.
- View the mappings between the logical and physical models using the Where Used tab.

Tutorials: Documenting an existing database

One of ER/Studio Data Architect's most powerful applications is that of a documentation generator to communicate complex databases and associated metadata to the Enterprise. ER/Studio Data Architect is equipped with extensive report generation capabilities:

- **HTML Report Generation**: Instantaneous generation of an HTML-based Web site designed to provide simple navigability through data models and model metadata using standard browsers such as Google Chrome or Mozilla Firefox.

- **RTF Report Generation**: Instantaneous documentation generation compatible with applications like Microsoft Word.

In the exercise below, we will reverse-engineer an existing database and generate an HTML report for distribution and navigation to those who depend upon the information about the data model, but who may not be permitted to connect to the database for security or organizational reasons.

Prerequisite

This exercise assumes that you can connect to an existing database in order to document it. If you cannot connect to an existing database, you can still generate documentation from the installed sample models; skip steps 1 through step 8 below which relate to in reverse-engineering and begin at step 9 after opening a sample model included with ER/Studio Data Architect.

1. Choose **File > New**. The Create a New Model window appears.

2. Select **Reverse-engineer an existing database**, and then click **Login**.
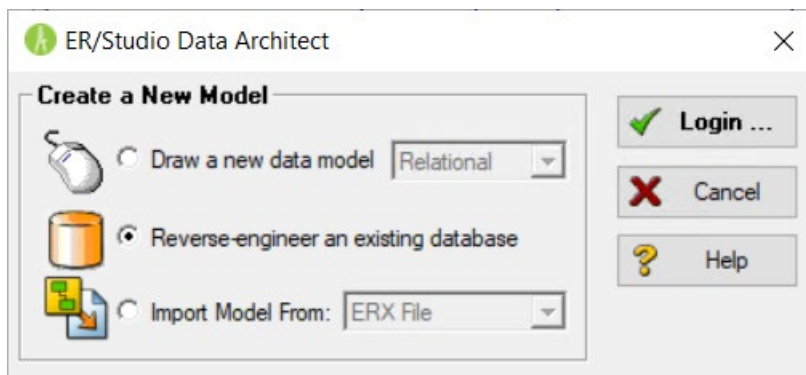
- o   You can reverse engineer the database from either an ODBC datasource or via Native RDBMS client connectivity. In this example, we use Native Connectivity to Microsoft SQL Server.
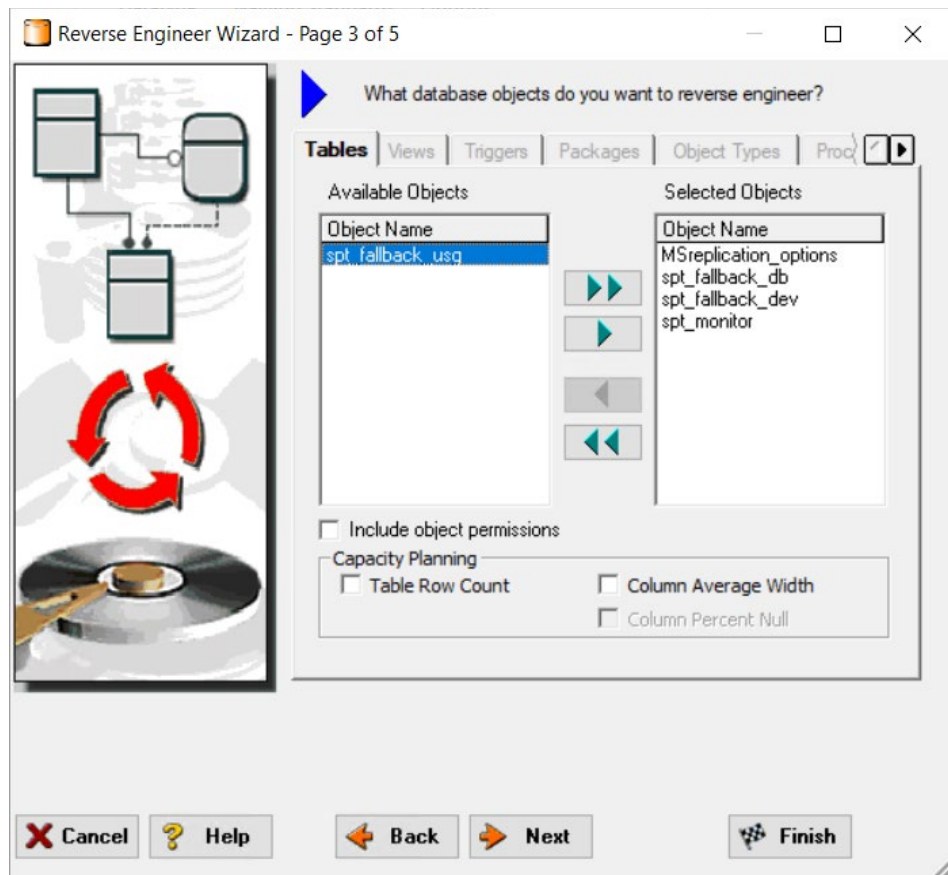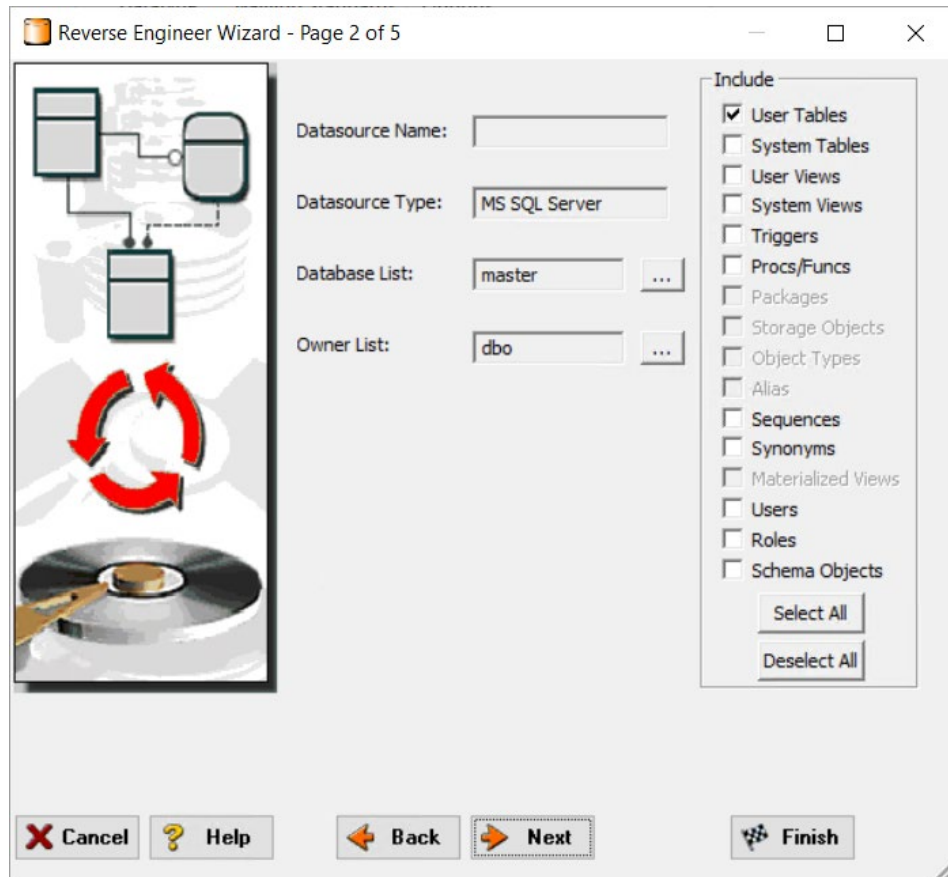
The Reverse Engineer Wizard appears.

---

3. Type the relevant connectivity information such as the data source name, user name, and password, and then click **Next**.
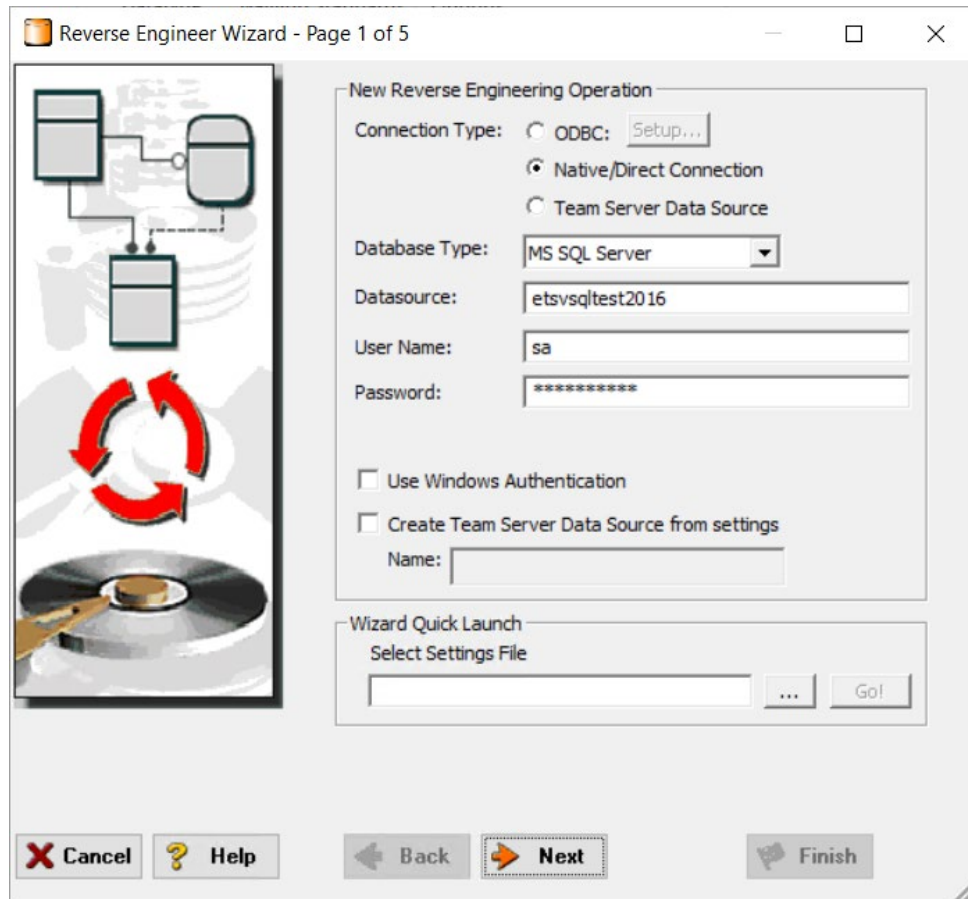
4. Walk through the Reverse Engineer Wizard selecting the objects, options, and layout preferences for the model.

Note	A new feature using ER/Studio Team Server data sources is available in the Reverse Engineer Wizard. Please see ER/Studio Team Server documentation for more details.

Datasource Name:

Datasource Type: MS SQL Server

Database List: master ...

Owner List: dbo ...

Include
- ☑ User Tables
- ☐ System Tables
- ☐ User Views
- ☐ System Views
- ☐ Triggers
- ☐ Procs/Funcs
- ☐ Packages
- ☐ Storage Objects
- ☐ Object Types
- ☐ Alias
- ☐ Sequences
- ☐ Synonyms
- ☐ Materialized Views
- ☐ Users
- ☐ Roles
- ☐ Schema Objects
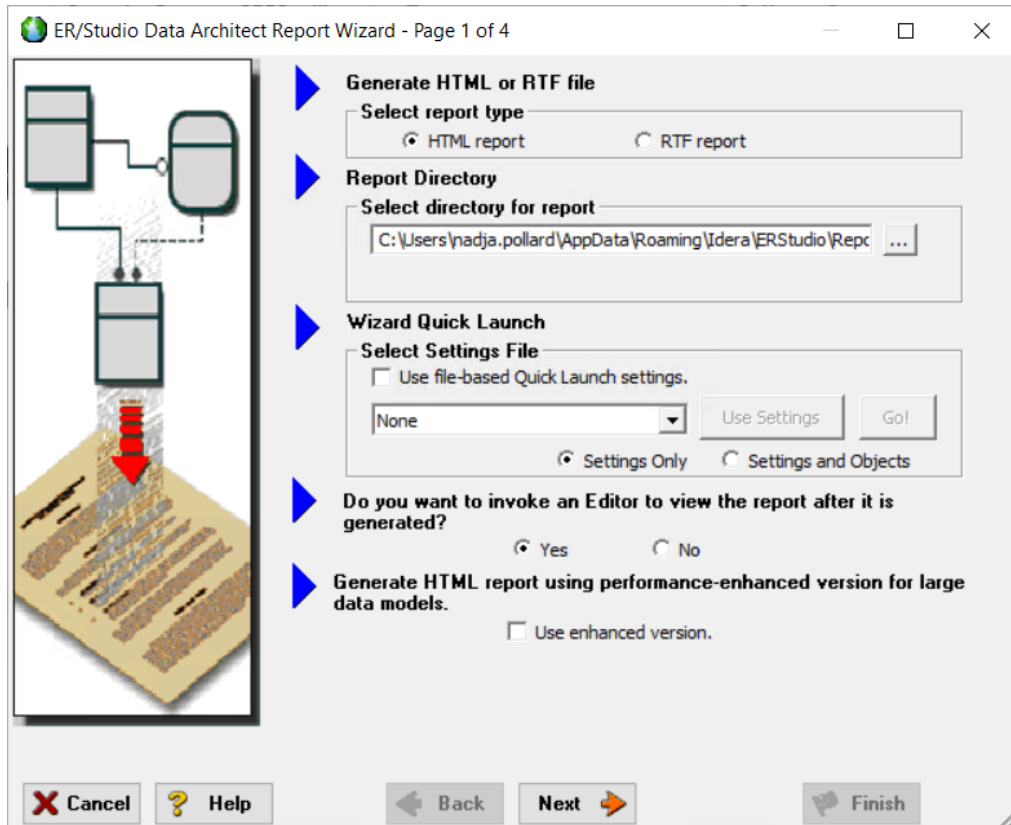
Select All

Deselect All

✗ Cancel | ? Help | ← Back | → Next | Finish

What database objects do you want to reverse engineer?

Tables | Views | Triggers | Packages | Object Types | Proc ◄ ►

Available Objects

Object Name
spt_fallback_usg

Selected Objects

Object Name
MSreplication_options
spt_fallback_db
spt_fallback_dev
spt_monitor

☐ Include object permissions

Capacity Planning
- ☐ Table Row Count
- ☐ Column Average Width
- ☐ Column Percent Null

✗ Cancel | ? Help | ← Back | → Next | Finish

5. Continue through the wizard to select layout styles and other preferences.
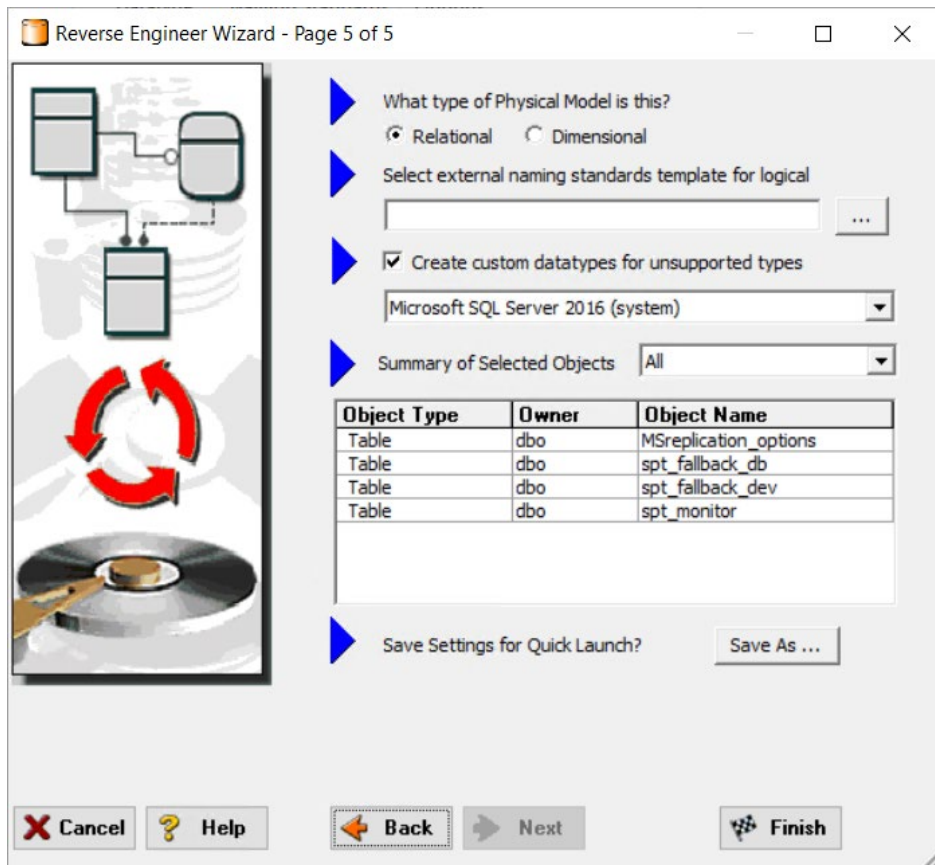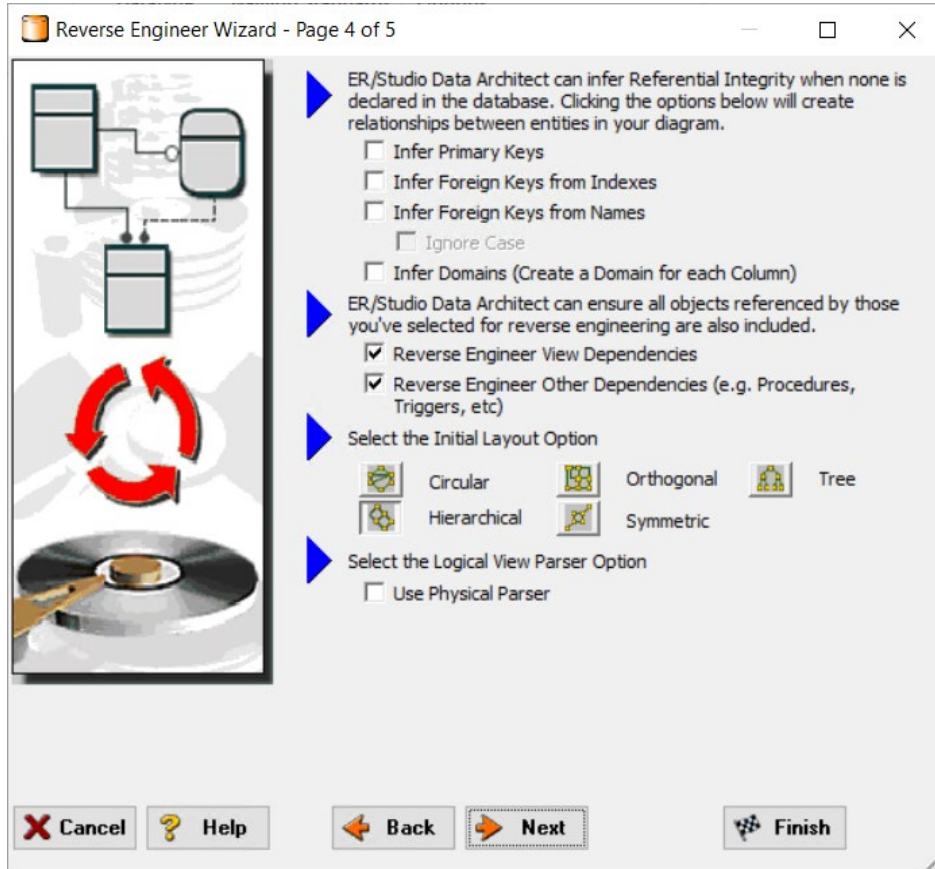
6. Click **Finish** and ER/Studio Data Architect reverse engineers your database.

   Once reverse engineering of your database is complete, we will generate a complete HTML report of the database for others in your organization to review.

7. In the Data Model Explorer, select the **Physical Main Model**.

8. Choose **Tools > Generate Reports**.

9. On the first page of the wizard, select **HTML** for the report type.

10. On page 2, click **Select All** in both areas of the Diagram tab.
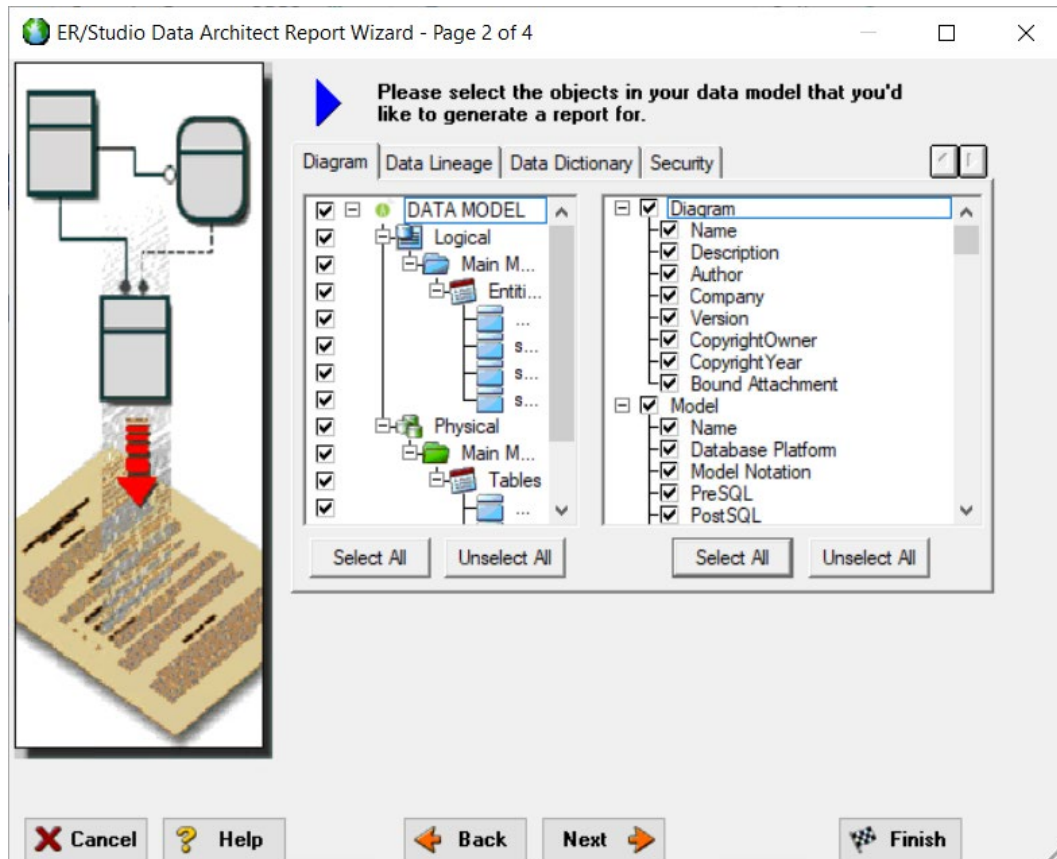
11. Click **Select All** in both areas of the Data Dictionary and the Procedures tabs, and then click **Next**.

Note	The tabs available on page 2 depend on what objects are supported by the database platform of the selected model. Some databases support more database objects than Microsoft SQL Server, such as Oracle and IBM DB2 LUW for which there are also tabs on this page for procedures, functions, triggers, packages, and tablespaces. If the model was previously denormalized, a Denormalization Mappings tab would also appear.

12. On page 3, click **Select All** in the Submodel Image Options area.

Tip	In Logo and Link Options you can choose to replace the ER/Studio Data Architect default IDERA logo in favor of your own corporate logo (and Hyperlink).
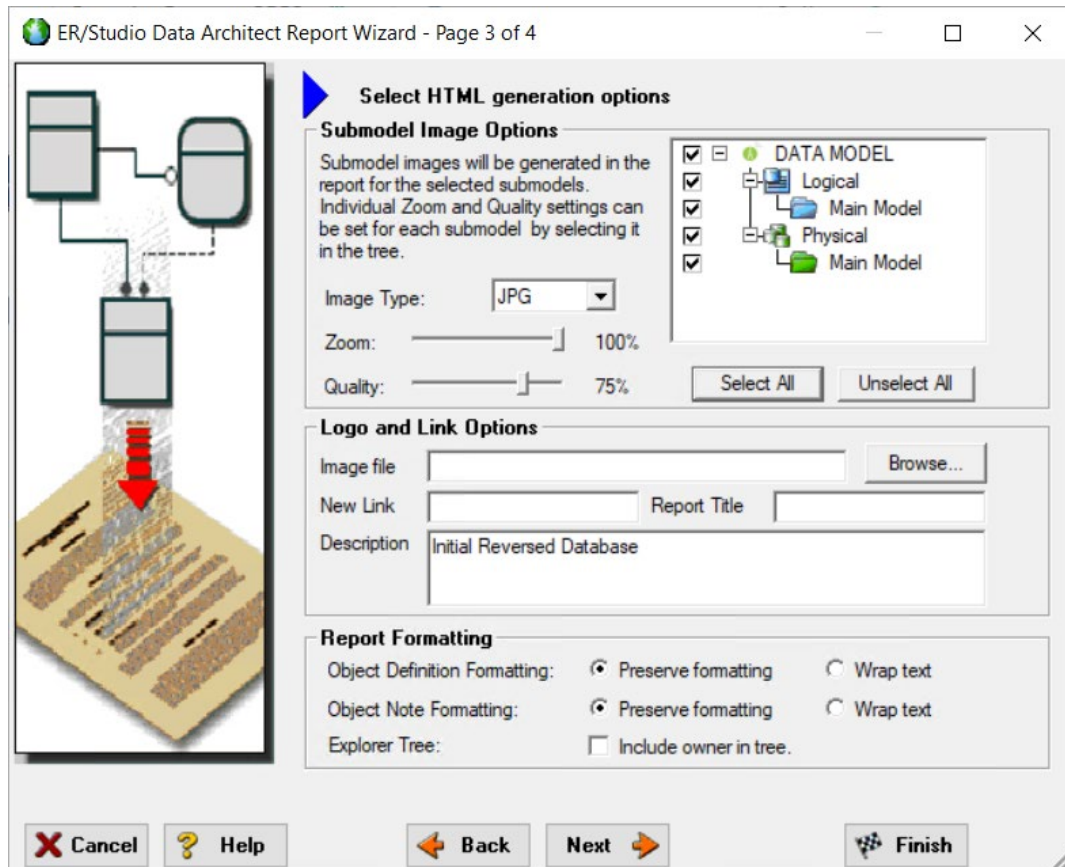
Because HTML formatting can be included in object definitions, you can also choose to preserve the formatting specified on the Definitions tab of the various object editors.

13. Click **Next** to advance to the last page, and then click **Finish**.

ER/Studio Data Architect then begins the report publication process and launches the default browser so you can review the report.

14. Finished!

Start navigating the report via your browser. Navigation will perform exactly as it does when you are using ER/Studio Data Architect. Expand the tree to find Model Image and click on it (see below). You will see a read-only version of your data model (as seen below). Use the Explorer to navigate to any metadata you want or select the entities and relationships in the model image to jump to their information.

Conclusion

In this session, you learned how to:

- Connect to and reverse-engineer an existing database with ER/Studio Data Architect.

- Document a database in seconds by using ER/Studio Data Architect's automatic HTML documentation publication facility.

Tutorials: Documenting data lineage

The Data Lineage feature of ER/Studio Data Architect enables you to document the movement of data from point A to point B (and any intermediate steps in between). This movement is sometimes referred to as Extraction, Transformation and Load (ETL). Points A and B can be anything from flat files, high-end databases such as Oracle and DB2, XML, Access databases, and Excel worksheets. This is sometimes referred to as source and target mapping. A model produced in ER/Studio can represent any point along the way. Data Architects need the ability to specify the *source* or *target* of data down to the column-level. Along with the metadata that defines the source and target mapping are rules for how the data is manipulated along the way.

The next section will help you document the data lineage of your systems.

It is comprised of the following tasks which correspond to the general ETL workflow:

- [Creating a Data Flow](#)
- [Creating a Data Movement Rule](#)
- [Defining External Source and Target Systems](#)
- [Creating a Data Lineage and Transformation Visualization](#)

Tutorials: Creating a data flow

The Data Flow organizes and encapsulates one data transformation and the source tables and columns used in the transformation to produce the target data. Multi-tiered mappings are possible and there can be multiple transformations involving different columns between two tables as illustrated below.
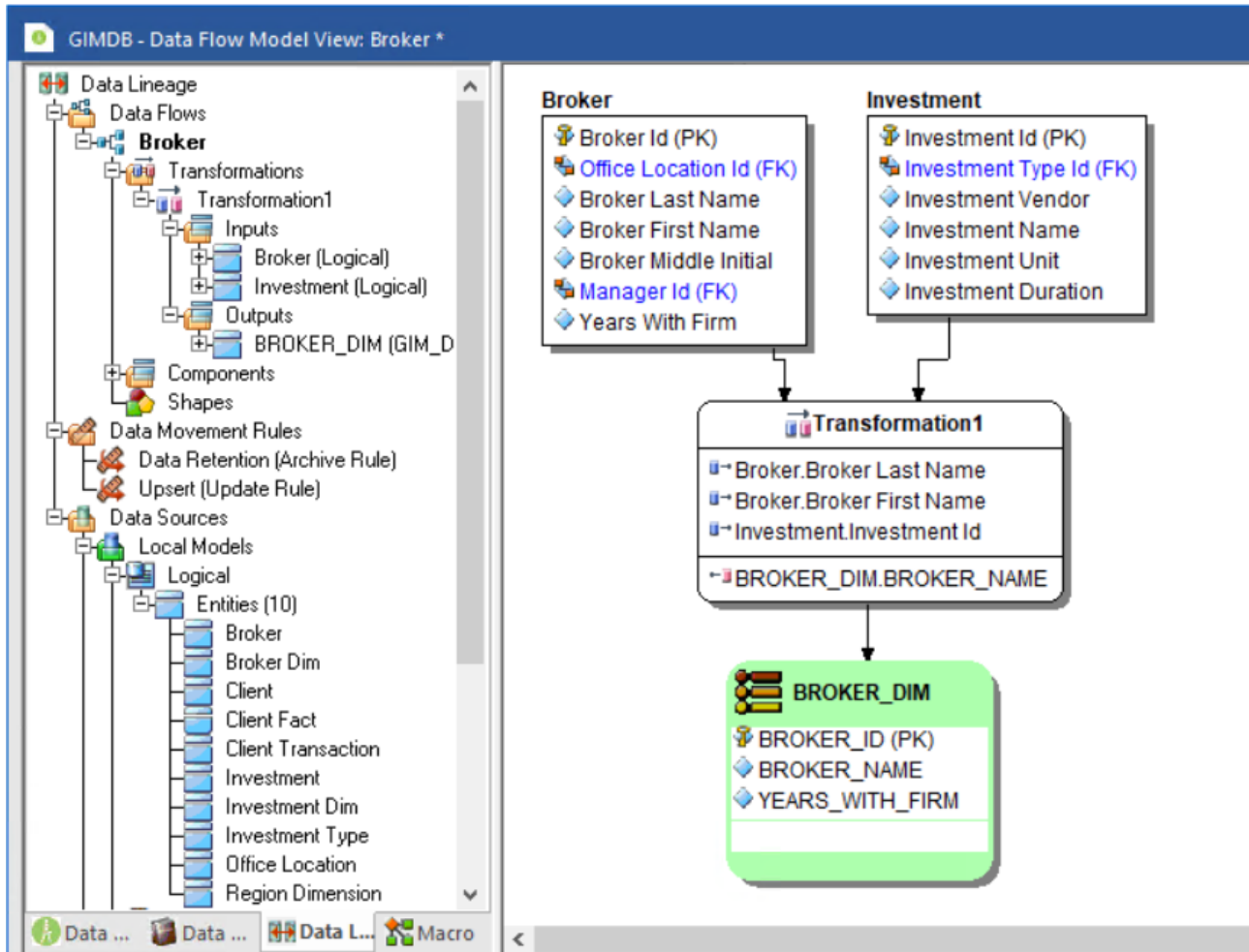
Create a data lineage data flow

1. Choose **File > Open** and select the **GIMB.DM1** diagram in the Sample Models directory.

2. Click the Data Lineage tab at the bottom of the application window. You are prompted to create a Data Lineage Data Flow.
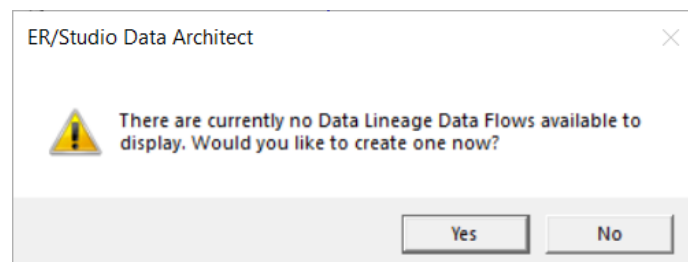
3. Click **Yes**.

   If this is not the first time you click the Data Lineage tab after opening a diagram, from the Data Lineage Explorer, right-click the **Data Flows** node, and then click **Create Data Flow**.

   If you would like to keep your version of GIMB.DM1 unchanged, '*save as*' a different filename before proceeding past the Undo/Redo warning dialog.

4. Enter a data lineage data flow name, and then click **OK**.
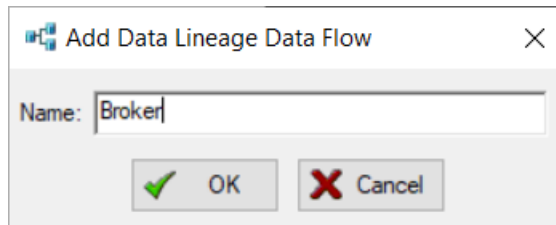
| Note	The name that appears in the diagram title tab at the top of the application window is appended with: *data flow* name, when you click a task in the Data Lineage explorer, such as *GIMDB.DM1 - Data Flow Model View: Broker**.

The Data Flow is created.

Tutorials: Creating a data movement rule

Data Movement rules describe how source and target tables and entities are related. You can relate source data to one or more tables and entities in the same model, the active diagram, or to tables imported from external systems. The rules defined here are used at the table level on the Data Lineage tab of the entity and table editors.

Create a data movement rule

1. On the Data Lineage tab, right-click **Data Movement Rules**, and then choose **New Data Movement Rule**.
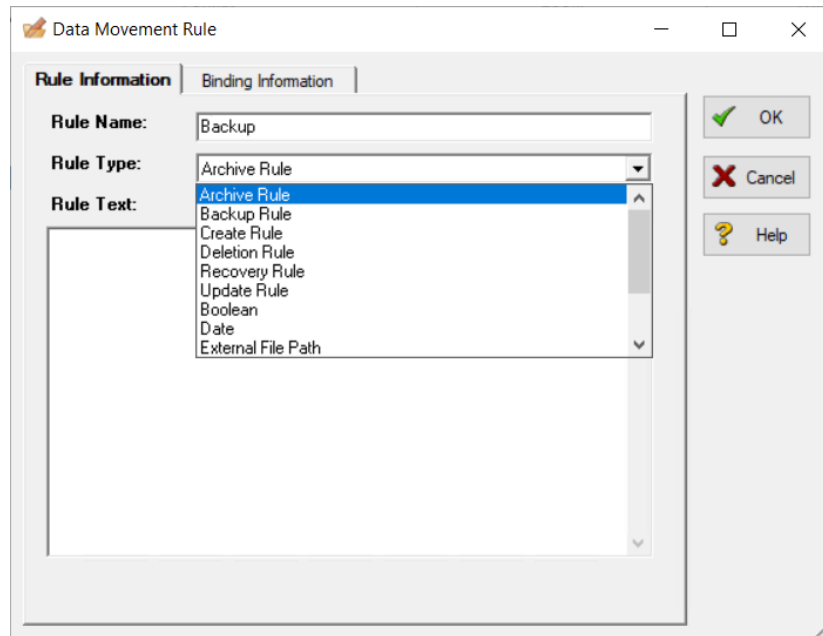
2. Complete the Data Movement Rule editor as required

   o **Rule Name**: Enter a name that indicates the operation and objects acted on, depending on the specifics of your binding definition.

   o **Rule Type**: Select a generic movement rule type that best describes the data movement.

   o **Rule Text**: Document your data movement plan here, perhaps adding instructions or contingency plans.

Tip	Once created, you can edit the Data Movement rule by double-clicking it to launch the Data Movement Rule editor.

3. Click **OK** to exit the editor.

Binding Information tab

Select the object classes and/or specific objects to which you want to bind this attachment. You can override this setting using the Data Lineage tab of the entity or table editor.

Tutorials: Defining external source and target systems

Data sources can originate from models in the active diagram (local models) or from external sources that are either imported into the active diagram or created on the Data Lineage tab. A data source can be imported from *.dm1* files, *.dt1* files, database or from SQL files, flat files, and other common application files. The following describes how to import metadata from an external source.

Note	Source data imported through the Data Lineage tab only includes information such as table and column name, datatype, nullability, primary key, and column definitions. To obtain more details, reverse engineer the database or import it into ER/Studio Data Architect using the Metadata Wizard.

Import external source or target data

1. From the Data Lineage tab, expand the **Data Sources** node.

2. Right-click **Other Sources**, and then choose **Import New Source**.
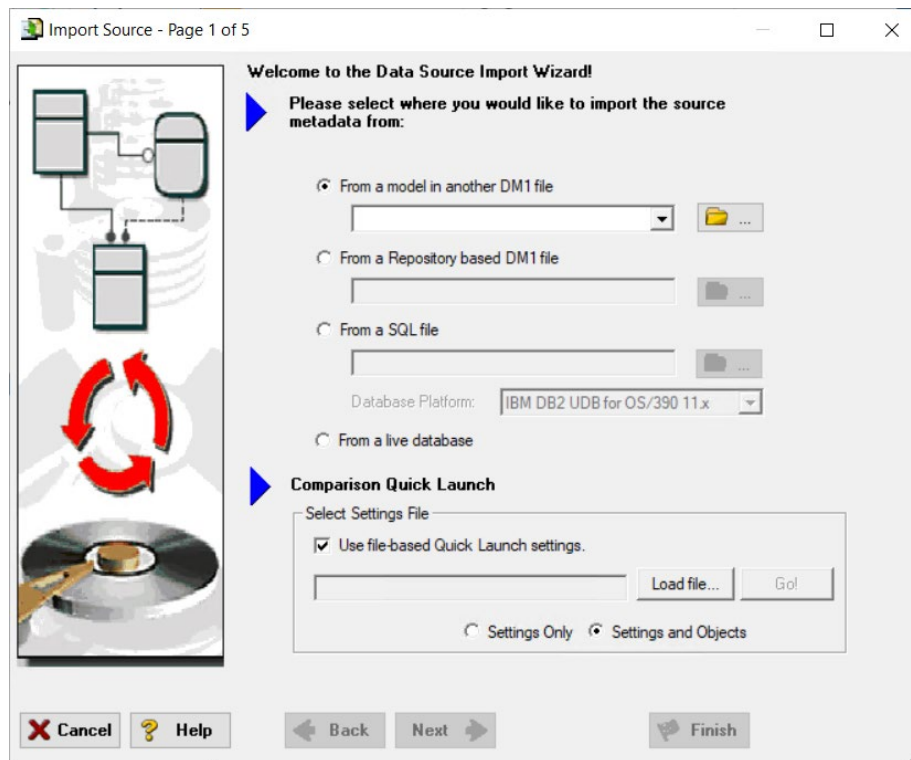
3. Complete the Import Source wizard as required, and then click **Finish** to import the source.

   The new source will appear under the Other Sources node.

The following describes options that require additional explanation:

Page 1: Please select where you would like to import the source metadata from

- o **From a Repository based DM1 file**: Lets you obtain source from data models and Named Releases managed within the ER/Studio Repository. When you select this option, ER/Studio Data Architect opens the Repository Operation Status dialog box and the Get From Repository dialog box. This process connects to the current Repository Server defined in the Repository settings. The Import Source wizard automatically gets the diagram.

- o **From an SQL file**: ER/Studio Data Architect imports the SQL file.

- o **From a live database**: If you select this option, a page appears where you can select the database and connection type. The connection type can be either ODBC or Native/Direct Connection. For information about connecting to databases, including troubleshooting information, see Connecting to Database Sources and Targets.

- o **Comparison Quick Launch**: The Compare Quick Launch data is saved as an *.rvo file. For information on using the *Quick Launch* option in the wizard, see Saving and Using Quick Launch Settings.

Page 5. Results

- o **Current and Target Model Display Grid**: Between the Source and Target models is a Resolution column. The default merge decision is Merge the data into the new source file. You can click on any item in the Resolution column to enable the decision list. If you want to change the decision, click the list and then click the new resolution. When you change the default resolution of an object, the decisions of their dependent properties and objects are automatically updated. You can also click the category folders, like the Tables Resolution column to change all the decisions for all the underlying objects in that object category. And, you can use the CTRL key to select multiple items, and then right click to enable the decision list.

- o **SQL Difference**: To enable the SQL Difference utility, select any difference that is a long text field, such as a Definition, Note, or DDL, and then click SQL Difference to view the differences between the SQL of the models. This utility only allows you to view the differences; difference resolutions are performed on the Results page of the Compare and Merge Utility.

- o **Filter Report on Results**: Create a report of the source content and your chosen resolutions. You can choose to create an HTML or an RTF report.

Tip	You can modify the default display using the options at the bottom of the page.

# Tutorials: Creating a data lineage and transformation visualization

1. To create the data source or transformation input tables, expand **Data Sources > Local Models > Logical > Entities**, and then drag and drop the Broker and Investment tables onto the Data Lineage window.

2. To create the data target or transformation output tables, navigate to **Data Sources > Local Models > GIM_DW**, and then drag and drop the Broker_DIM table onto the Data Lineage window.

3. To obtain the Transformation Insertion tool, right-click an empty space in the Data Lineage window, and then click **Insert Transformation**.

4. To insert the transformation, click in the Data Lineage window between the source and target data sources, and then right-click to drop the Transformation Insertion tool.

5. Reposition and resize the transformation object to suit your needs.

6. Right-click an empty space of the Data Lineage window, and then click **Insert Data Stream**.

Tip	Transformation and Data Flow tools are also available on the toolbar. Hover the mouse over the tools to find the tool you need.

7. Click an input and then click the transformation object. Repeat as many times as necessary to link all the inputs to the transformation object.

8. Click the transformation object and then click an output.
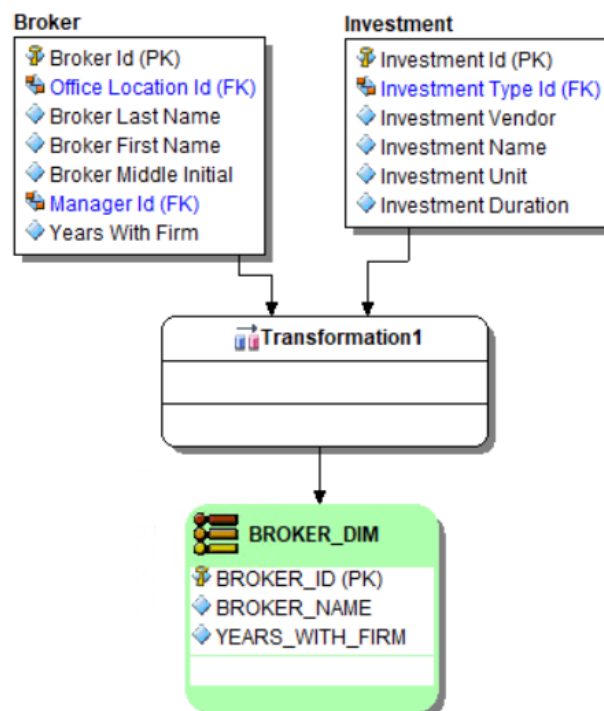
9. To define which columns should be used in the transformation and any transformation rules, double-click the new transformation to open the Transformation Editor.

10. Complete the Transformation Editor as required, and then click **OK** to exit the editor.
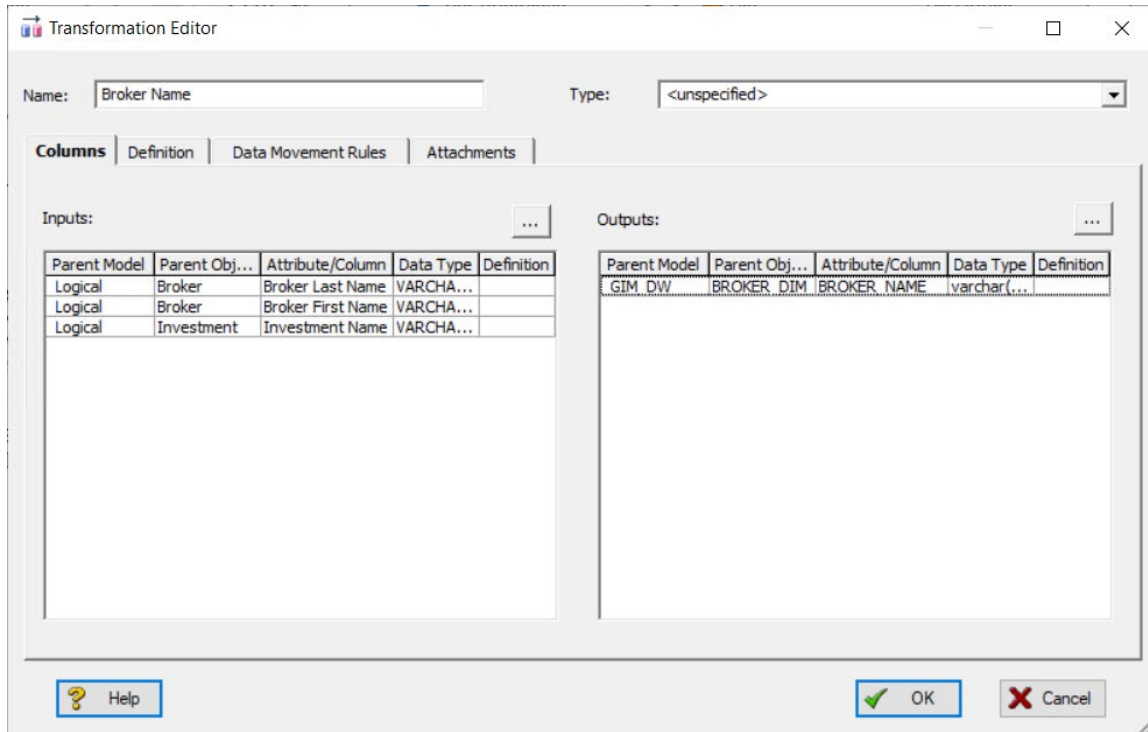
11. Finished! Now you can more easily share your ideas with your colleagues.

Tip	Once the Data Flow is created, you can double-click it to change its name, or double click a transformation or component to change its properties.

The following describes options in the Transformation Editor that require additional explanation, for more options that are not required for this tutorial, please read the User Guide.

Columns tab

- **Inputs**: Click the ellipsis (**...**) button to choose the inputs to be transformed in this task.

- **Outputs**: Click the ellipsis (**...**) button to choose the outputs resulting from the transformation.

Definition tab

- **Business**: Describe the transformation for your audience.

- **Code**: Enter the code that will perform the transformation, such as a SELECT statement, or a VBBasic or Java Script function or procedure.

Data Movement Rules tab

These are the rules from the Data Movement Rules node of the Data Lineage Explorer.

Note	You can delete or edit an input or output column by double-clicking the transformation in the Data Lineage window, clicking the ellipsis in the Transformation Editor and then deselecting the column you want to remove.

Attachments tab

Bind an external piece of information or attachment to the transformation. You can also remove an attachment from an object, override the default value of an attachment, or change the position of a bound attachment.

To override the value of the attachment you have moved to the Selected Attachments grid, double-click the Value field of the target attachment. ER/Studio Data Architect opens the Value Override Editor or a list, depending on the attachment datatype. Attachments are created in the Attachments folder of the Data Dictionary and must be applied to the default before they will display on this tab.

# Tutorials: Diagram navigation and aesthetics

To assist with the creation of presentation-quality diagrams that are easy to navigate and are aesthetically pleasing, ER/Studio Data Architect offers progressive diagram Auto Layout and Navigation utilities that also help you to clean up complex diagrams. Modelers should spend time solving complex database or business data model problems, not forcing boxes and lines to look a certain way.

Navigating the diagram

To demonstrate some of ER/Studio Data Architect's layout and navigation utilities, we will import a sample SQL script provided with ER/Studio Data Architect.

1. Close any files you have open.

2. Choose **File > New**.

3. Select **Import Model From > SQL File**, and then click **Import**.

Note	The ERX File choice enables you to import erwin ERX files. Selecting From External Metadata launches the MetaWizard to import from alternative sources.
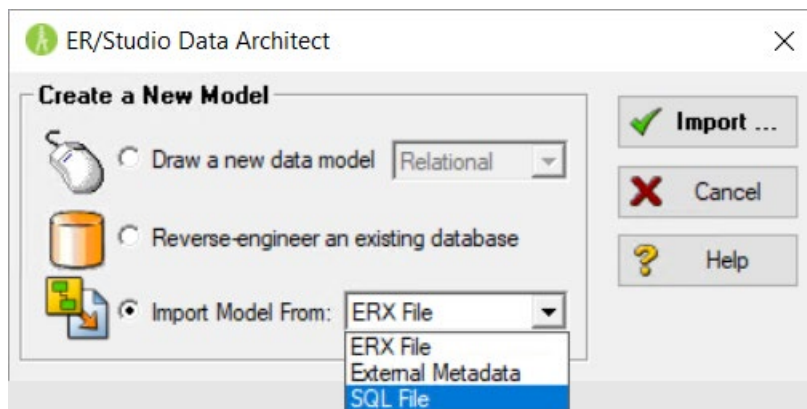
    The Import Database SQL File dialog appears.

4. To the right of **Select a Database SQL File**, click the folder icon, and then find **Sample DDL (DB2 7.x for OS390).SQL** (located at C:\Program Files\IDERA\ERStudio Data Architect x.x\SQLCode\Sample DDL (DB2 7.x for OS390).SQL), and then click **Open**.

5. In the **Select the target database platform** list, click **IBM DB2 UDB for OS /390 x.x**, and then click **OK**.
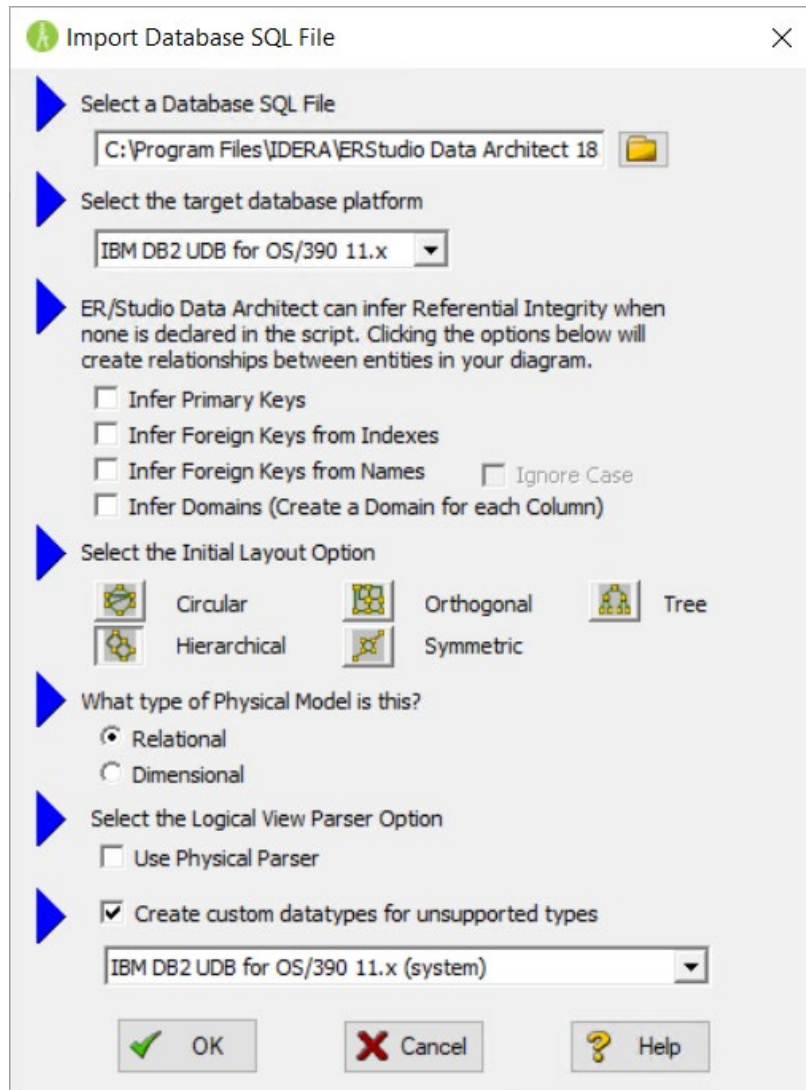
Finished! Once the SQL Script is finished importing, the following items will assist you in leveraging a variety of Auto Layout and Navigation Features.

- **Diagram > Layout**. Use any of the Auto Layout styles, such as hierarchical, orthogonal, and symmetrical, to change the layout of the diagram with the click of a button. The auto layout styles are all entirely customizable styles. You can also customize the diagram layout via the **Layout Properties** option. In addition, you can have multiple layout styles in a model. Select an object, and then choose the layout style. If you don't have any objects selected when you click the layout button, the layout style chosen is applied to the entire model.

- **Tools > Overview Window**. Use this window as a thumbnail of your model to pan the entire model or zoom in and out. It can also pan and zoom the diagram if grabbed or sized.

- **Tools > Zoom Window**. Use this window as a magnifying glass to enlarge any diagram objects under your mouse cursor. You can also press SHIFT+F8 to freeze the zoom window to keep a single object frozen while you continue to pan around the diagram. If the Zoom window is not already visible, press F8 to activate it.

- **Data Model Explorer**: Click any object in the Data Model explorer and it is automatically selected in the diagram and focused in both the Zoom Window and Overview window.

Diagram aesthetics

One of the tremendous benefits of building data models is the wide range of audiences that can realize value from them. Part of this relies on what information is displayed in the diagram. Depending on the audience you may want to limit or expand what is displayed. For example, developers may benefit from looking at a model that displays data type, null option, and unique and non-unique index information, while business analysts may just need the entity name and the definition. ER/Studio Data Architect offers many display properties that can be customized exactly for this purpose.

Continuing with the previous section, we will use the DB2 model that was built to demonstrate some of the ways to customize the appearance of the model.
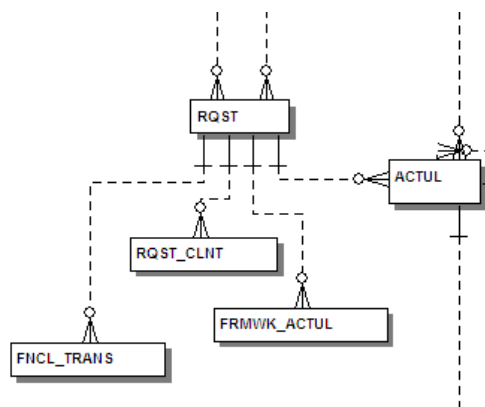
We will use the Diagram and Object Display Options dialog on the Diagram toolbar to further customize the view of the logical and physical models.

Setting the logical model display

1. Select the Logical model and then on the Diagram ribbon, click the **Diagram and Object Display Options** tool.
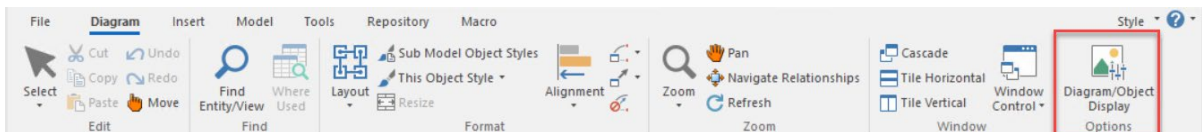
2. In the Diagram and Object Display Options dialog, click the Entity tab, select **Entity** in the **Display Level** area, and then click **OK**.
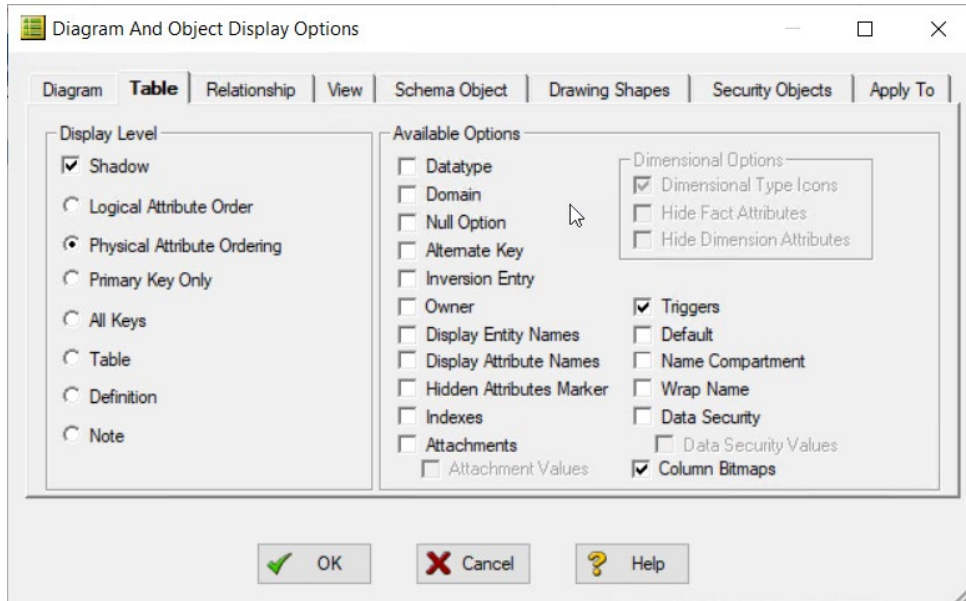

Note	Only entity names are displayed for each entity. You may also want to re- layout the diagram since the entity sizes have changed.

Setting the physical model display

1. Select the Physical model, and then choose **Diagram > Diagram and Object Display Options**.

2. In the Diagram and Object Display Options dialog, click the Table tab.

3. In the Display Level area, select **Physical Attribute Ordering**.

4. In the Available Options area, select the specific properties you want to display, and then click **OK**. The model now displays more details for the physical model.
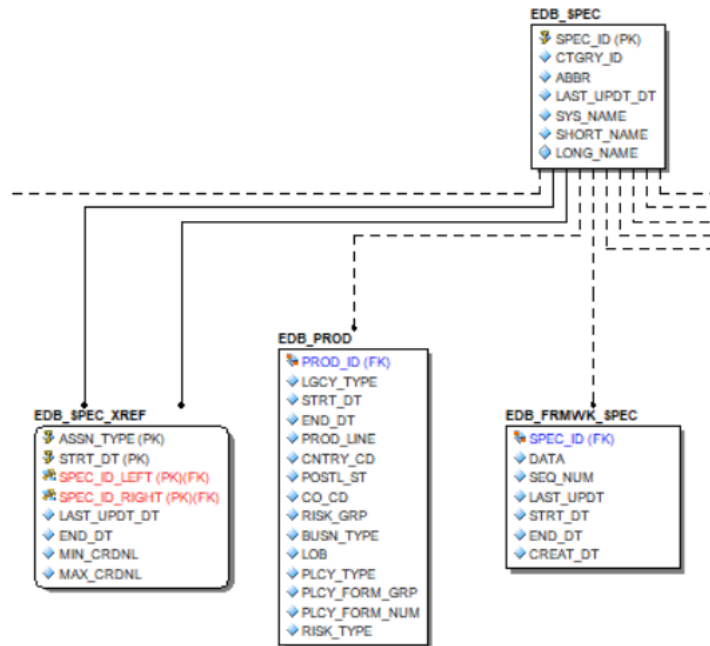
| Note | Because the sizes of the objects changed, you may want to change the model layout using one of the advanced layout engines. |

Tutorials: Data Source Mapping

Steps for using the Data Source Mapping:

1. Create or open a diagram

2. Connect and Publish on Team Server

3. Log in to Team Server from ER/Studio Data Architect:

4. To log in to Team Server, choose **Repository > Team Server Log In**.

5. Complete the log in dialog and then click **OK**.

6. Right click on the physical model that you want to map and select **Data Source Mapping** option.

7. Relate the model with the data source.

8. Edit any table on the physical model related and go to Where use tab. Or right click on the table and select **Where use** option.

9. The Where Used tab/dialog shows the data sources related. These are potential database instances that could be impacted if the table changes.

Conclusion

In this session, you have learned how to:

- Import an SQL file and allow ER/Studio Data Architect to automatically create a diagram.
- Use a variety of auto layout and navigation tools to enhance the aesthetic experience of the diagram and to improve the data model navigability.
- Customize the display of both the logical and physical models.

Thank you for completing the tutorials section. Please visit the User Guide for more examples and information on using ER/Studio Data Architect.

See Also

- Wiki Home
- User Guide