embarcadero®

Product Documentation

ERStudio Data Architect

Quick Start Guide

Version 2016.0.3

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance, and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, Embarcadero® RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder®, and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

May, 2016

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** has been 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:

- o The ability to spawn many physical designs from a corporate logical design.

- o Bi-directional model comparison and information synchronization.

- o Visual-Basic for Applications API for product customization.

- o Powerful DDL reverse engineering and generation.

- o Metadata import and export capabilities.

- o Data lineage documentation.

- o Sophisticated XML, HTML, and RTF-based documentation and reporting facilities.

Notice for Developer Edition Users

The Developer Edition of ER/Studio Data Architect provides the user with all the standard functionality of ER/Studio except for the following limitations:

- o No access to the ER/Studio Repository and ER/Studio Team Server.

- o Limited import and export functionality; the Import and Export Metadata Model Bridges are not supported.

o   Limited database support. For a list of supported database platforms, see [Supported Database Platforms](#).

Important Name Change Notice

The following describes recent product names changes in the ER/Studio family

Old Name	New Name	Short Form of New Name
ER/Studio	ER/Studio Data Architect	ER/Studio DA
ER/Studio Repository	ER/Studio Repository	Repository
Embarcadero Connect	ER/Studio Team Server	ER/Studio TS
EA/Studio	ER/Studio Business Architect	ER/Studio BA

Product Benefits by Audience

o   **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.

o   **Database Administrators and Database Developers**: Managing databases can be 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 can be used as powerful and efficient change management platforms, allowing users to update a model with the required changes which need to be implemented at the database and automatically generate DBMS-specific, syntactically correct, alteration or database DDL.

o   **Business and IT Managers**: ER/Studio Data Architect's robust reporting facilities allow critical information about designs to the enterprise to be delivered in seconds. This heavily leveraged and beneficial capability of ER/Studio DA 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.

What's New

ER/Studio Data Architect 2016 (16.0.3) release notes updated May 27th, 2016.

Release 2016 features summary:

Business Data Objects

- o You can now group your entities/tables into Business Data Objects. These objects allow you to better describe your business concepts by grouping entities or tables that share a common relationship in to a visible container.

Model/Submodel NSTs

- o You can now bind Naming Standards Templates to Models and Submodels.

Windows 10 Support

- o You can now install your favorite Data Modeling software on your newest OS.

Teradata Support

- o We now support Teradata 15.x.

Teradata Enhancements

- o Period Datatypes.
- o Temporal Datatypes.
  - ▪ AS VALIDTIME.
  - ▪ AS TRANSACTIONTIME.
- o Teradata JSON Support.

SQL Azure Support

- o Microsoft SQL Azure is now fully supported for all operations.

Installing Embarcadero 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 _http://www.embarcadero.com/downloads_. Embarcadero 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

	Processor	Dual Core or higher
	RAM	2 GB
Hardware	Disk Space	2.5 Gb
	Operating System	Any of the following: • Windows Vista, 7, 8, 8.1 & 10 • Windows Server 2008, 2008R2 & 2012
Software		To use an Open Data Base 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). DBMS privileges to create objects on the server: EXECUTE permission as the user needs to execute stored procedures, for example Create_Temp_Tables.

Installing ER/Studio Data Architect

Tip: We recommend to have installed most current version of both, **ER/Studio Repository** 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 onscreen prompts.

The following provides additional information for some options that are not self-explanatory:

o *Default Notation*: Choose the notation type you want **ER/Studio Data Architect** to use by default to display the model.

o *IDEF1X*: Data modeling technique used by many branches of the United States Federal Government.

o *IE (Martin/Finkelstein)*: Information Engineering (IE) as developed by Martin and later revised by Finkelstein.

o *IE (Crow's Feet)*: Uses IE notation and represents relationships with connecting lines between entities, and symbols at the ends of those lines to represent the cardinality of the relationship. This is the notation used in Oracle texts and in other applications such as Visio. Foreign keys are displayed in entity boxes.

o *IE (Hide Foreign Keys)*: Uses standard IE notation, but foreign keys are not displayed in entity boxes.

Note: You can change the notation setting at a later time. To change the notation, choose **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 *Shared Application Data Location*: 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\Embarcadero\ERStudio_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.

For information on installing ER/Studio with Embarcadero AppWave, please see the *AppWave Admin Quick Start* and the *AppWave Administrator Guide* at http://docwiki.embarcadero.com.

Upgrading From Previous Versions

Use the following checklist to ensure ER/Studio Repository is correctly upgraded.

Upgrade Checklist

o   Download ER/Studio Enterprise Upgrade. To download the most current release, go to embarcadero.com.

o   Review the ER/Studio ReadMe. For the most current installation and usage information, see the ReadMe at embarcadero.com.

o   Prepare to upgrade. For what to do before you upgrade to the latest version, see Preparing to Upgrade the Repository.

o   In case you need to ER/Studio Team Server to match the features available in your version of ER/Studio Data Architect, see ER/Studio Team Server Upgrade Notes.

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.

Important: To ensure compatibility with ER/Studio Repository synchronization, this version of ER/Studio Data Architect requires at least ER/Studio Repository version 7.0 for all Repository operations.

License Enforcement - When working with the Repository, all users must use the same license type. ER/Studio now enforces this 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 will not be allowed.

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:

    1. Choose **Repository > Options.**

    2. Make note of the Active File Directory.

    3. Switch to the Windows Explorer and rename all the diagrams in the Active File Directory.

Administrator Preparations

1. Ensure no one is logged into the Repository.

2. Backup the current Repository database.

3. Restart the Repository host computer.

Note: If you are upgrading from a Repository version prior to 4.0, you can delete the C:\Program Files\Embarcadero\Repository\Data directory. This directory is not used with Repository versions 4.0 and later.

Upgrades from ER/Studio Repository 6.x to 7.0

1. Prepare the Repository for upgrade

2. Log on to Windows with local administrator privileges.

3. Start the *Repository* installation program and select *Repair Repository Services*. The installation program starts the following services:

▪ Embarcadero Repository Communication Server.

▪ Embarcadero Repository Database Server.

▪ Embarcadero Repository Event and Dispatch Server.

The Repository services and your existing database will be upgraded to the current version.

Upgrades from ER/Studio Repository prior to 6

The Repository upgrade must be done incrementally from one major version to the next until the target version is reached. You cannot skip major versions during the Repository

---

upgrade as the installer for each major version only contains the database changes between the new version and the previous one.

When upgrading from earlier versions, such as from 5.5 to Repository 6.6, you must first upgrade the Repository to version 6.0, update the database, uninstall version 6.0, then upgrade to 6.6, and update the database. Once the Repository upgrade is complete and you have installed the latest version of ER/Studio Data Architect, then you should upgrade your diagrams with the latest version of ER/Studio DA.

Consult Embarcadero Technical Support when upgrading from versions prior to 4.5.

Upgrading from 6.5/6.6/6.7 to 7.0 does not require the Diagram Upgrade step.

Important Notes

o   Always backup the Repository database before upgrading.

o   You can install ER/Studio Data Architect 9.x and an earlier ER/Studio Data Architect version on the same workstation.

See Also

o   [Release Notes](#)

o   [Installing Embarcadero ER/Studio Data Architect](#)

o   [Licensing](#)

Tutorials

The tutorials are intended to help you get started using Embarcadero'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. You will have learned how to do the following:

o Create a new data model.

o Work with logical and physical diagrams.

o Leverage productivity-focused features such as powerful reporting engines.

o Use common tasks and commands to make you more productive.

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

You can use these basic tutorials as a road map of product highlights, but also to help you find your own path in exploring **ER/Studio Data Architect**.

Choose a topic to work with:

o [Getting Started](#)

o [Logical and Physical Modeling](#)

o [Using Data Dictionary Domains to Populate New Entity](#)

o [Establishing Relationships Between Entities](#)

o [Creating and Working with Submodels](#)

o [Generating Physical Models from a Logical Model](#)

o [Denormalizing the Physical Model](#)

o [Finding out How an Entity Maps to the Physical Model](#)

o [Documenting an Existing Database](#)

o [Documenting Data Lineage](#)

o [Creating a Data Flow](#)

o [Creating a Data Movement Rule](#)

o [Defining External Source and Target Systems](#)

- o [Creating a Data Lineage and Transformation Visualization](#)

- o [Diagram Navigation and Aesthetics](#)

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 graphic below 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 or mouseover the image:
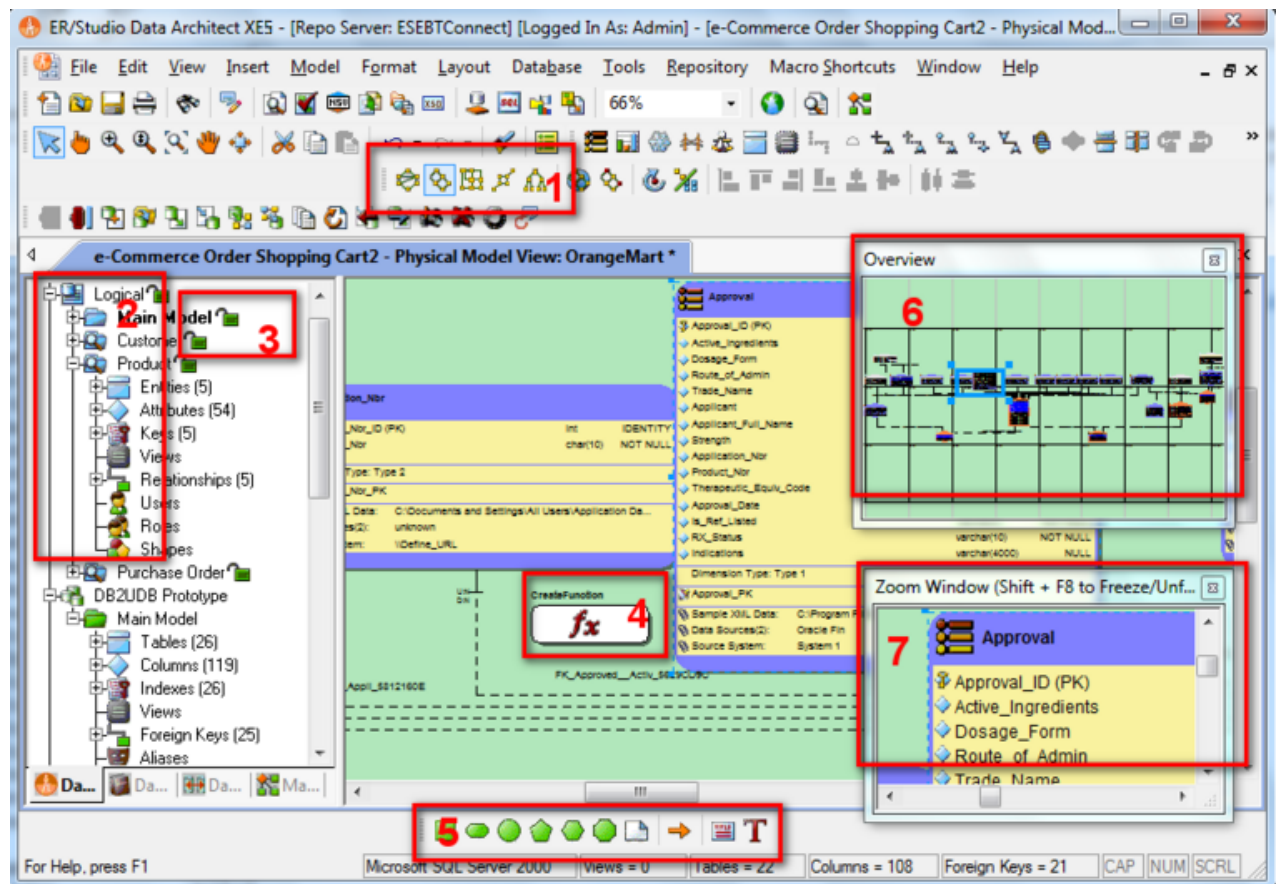
1. Sophisticated diagram auto-layout tools provide single-click clean-up of diagram objects.

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

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

4. Complex schema objects like functions can be displayed to illustrate dependencies.

5. Toolbars are dockable anywhere in the ER Studio DA application window.

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

7. 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

On the **Windows Start > Programs** menu, choose **Embarcadero > ERStudio Data Architect.**

1. Choose File > New > Draw a new data model.

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:

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

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

o 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.

o Ensure *Relational* is selected for the new model type and then click *OK*.

After selecting **Draw a new data model** and clicking **OK**, ER/Studio Data Architect will resemble the image below:
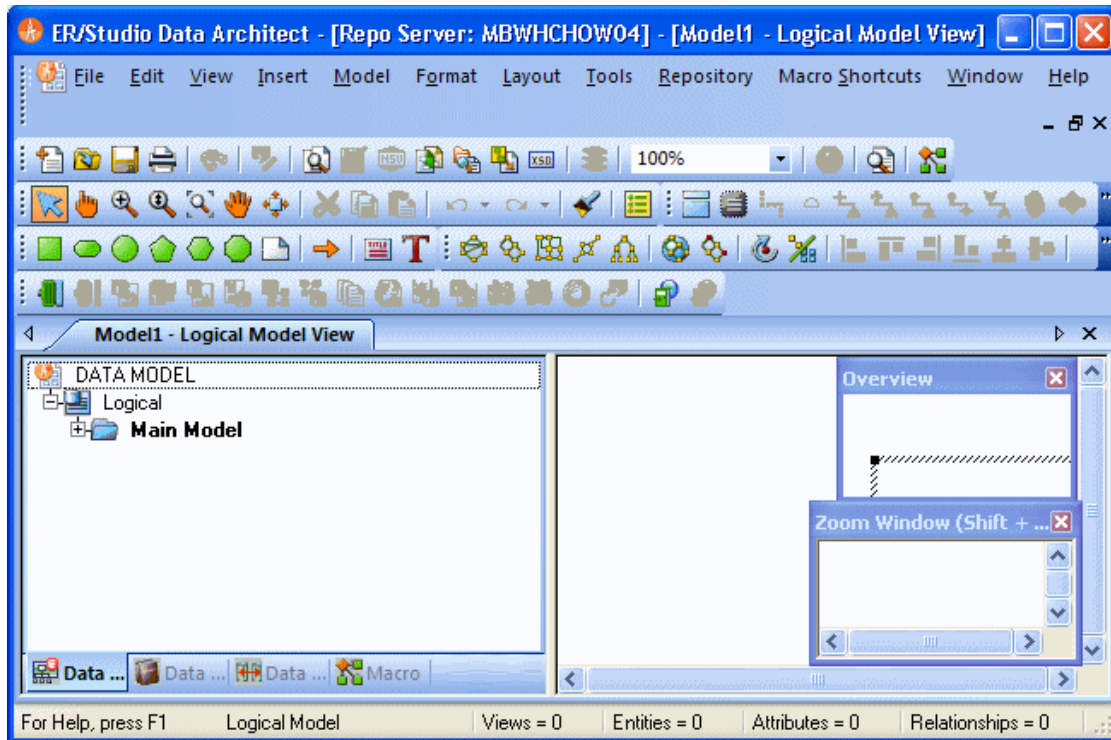
Tutorials: Logical and Physical Modeling

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. Multiple physical models can be generated 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 will be discussed in detail in the following sessions.

- o [Using Data Dictionary Domains to Populate New Entity](#)

- o [Establishing Relationships Between Entities](#)

- o [Creating and Working with Submodels](#)

- o [Generating Physical Models from a Logical Model](#)

- o [Denormalizing the Physical Model](#)

- o [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 File > Import Data Dictionary.**

2. Next to the **File Location** box, click the ellipsis and browse to the **Sample Models** folder, which is located at:

   ▪ **For Windows XP (No longer supported)**:
   C:\Documents and Settings\All Users\Application Data\Embarcadero\ERStudioDA_X.X\Sample Models

   ▪ **For Windows Vista and later versions**:
   C:\ProgramData\Embarcadero\ERStudioDA_X.X\Sample Models

3. Double-click the *Orders.dm1* sample model 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 a couple of 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.
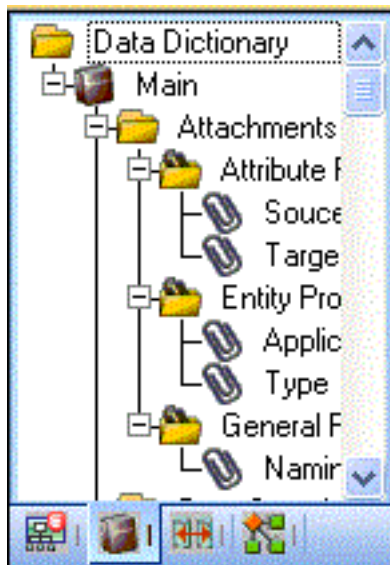
4. Now, to add an entity to the Data Model Window, click the *Entity* tool on the *Modeling Toolbar*,
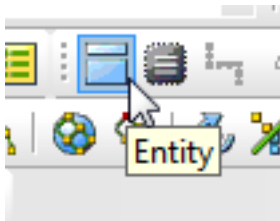

and then click in the Data Model Window to drop the entity.

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 open the *Entity Editor*. In the entity name field, type *Customer*, replacing the default entity name, Entity1 and then click **OK**..

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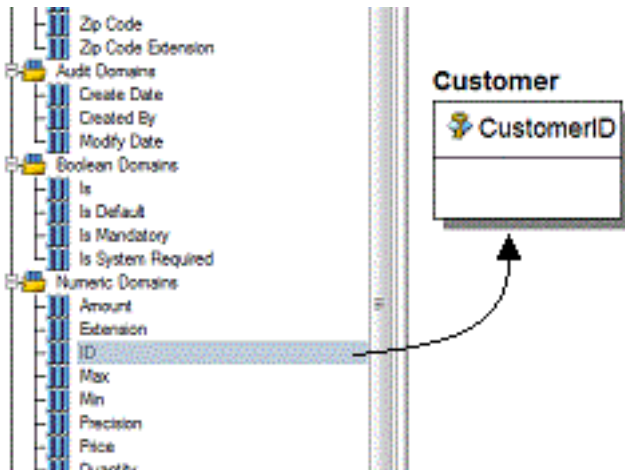
8. Click the **ID** domain (do not release your mouse), drag it onto the **Customer** entity, and then release it just below the entity's name, which is the entity's Primary Key field.

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. In the entity, 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 step 7 and step 8 to populate the Customer entity with the following domains:

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

- **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 can be very 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 Modeling toolbar, click the **Non-Identifying, Mandatory Relationship tool.**
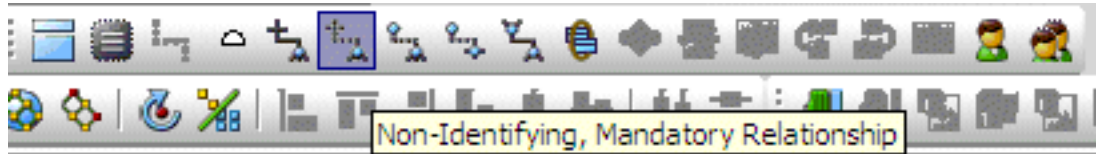
Note: The screen shots of this tutorial were taken using a model where the notation was set to IE (Crow's Feet). Depending upon the notation your model is currently set to, the icons for each relationship will be slightly different.

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 should also be propagated to the child, in the *Relationship Editor* you can choose all available parent entity keys in the Parent Key list. Deleting a relationship will remove 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 you can check the Make Foreign Keys Native option. In this case, if the relationship between Customer and Order is deleted, the CustomerID will be left in the Order entity.

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 can be 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 be modifying this model.

4. Collapse the folders in the *Data Model* tab of the Data Model Explorer to look like the image below:
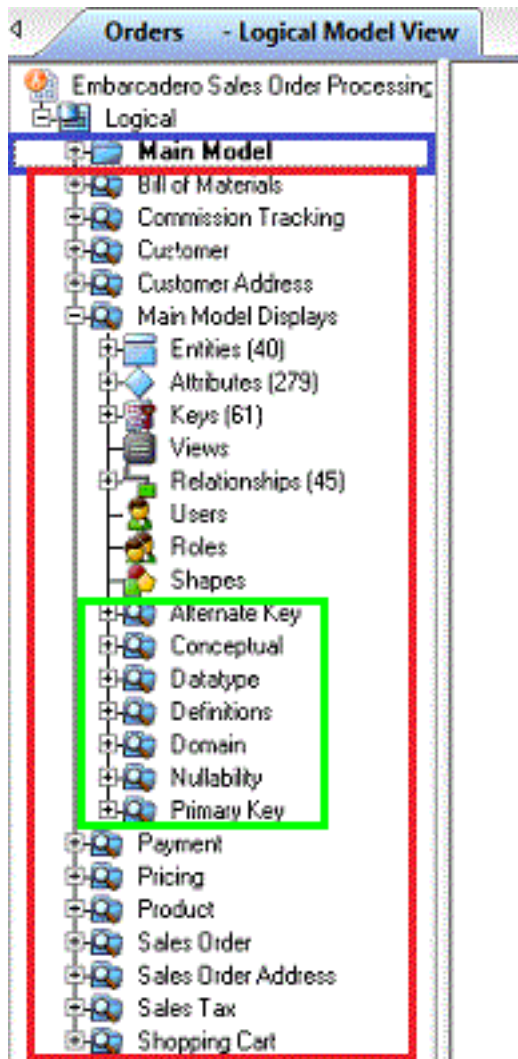
In the *Orders.DM1* sample model, there are no physical models. This model includes several submodel folders that help to describe the logical model:

- **Main Model** – This is the entire collection of all logical objects in the Orders.DM1 file. Note the absence of the magnifying glass on the folder icon which designates it as the main model.

- **Bill of Materials** throughShopping 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.

- **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 *Orders.DM1* Address components.

---

5. To make a new submodel, navigate to Logical Main model and with the **CTRL** key depressed, click the objects in the Data Model Explorer, as seen in the image below.
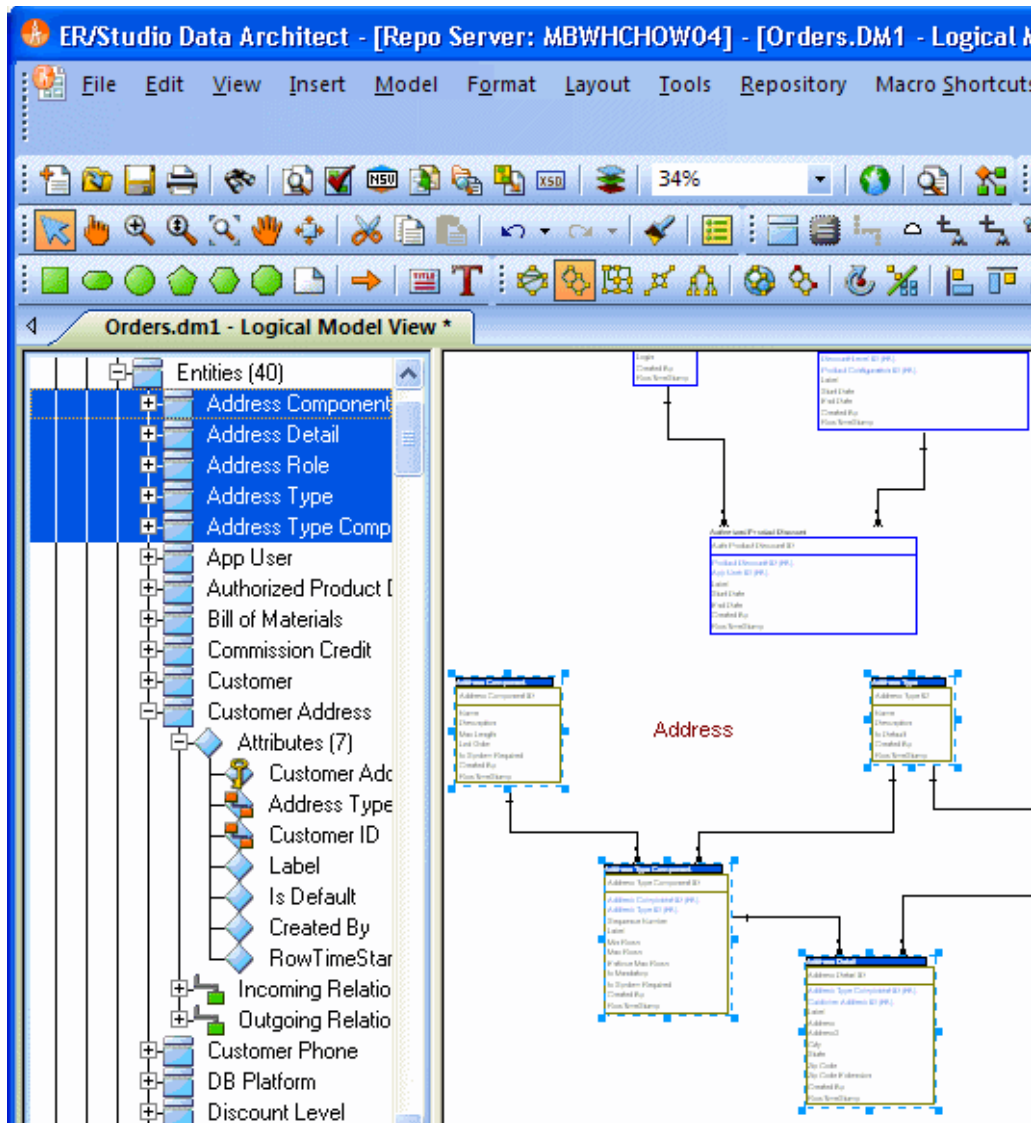
Note: Any objects selected in the Data Model Explorer will also be selected on the Data Model Window as well (also seen here). You can also select entities by lassoing them on the Data Model Window.

6. With the entities selected, choose **Model > Create Submodel**.
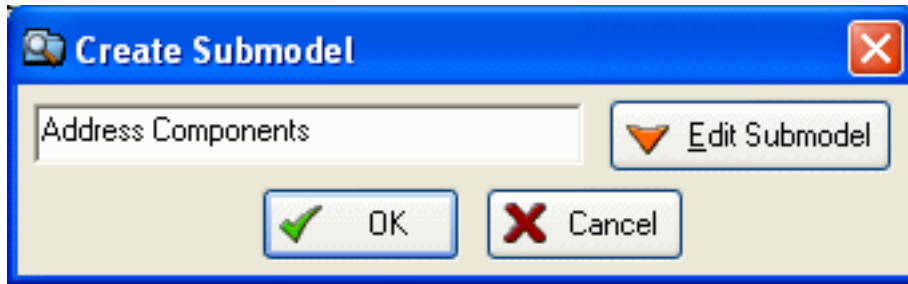
7. Below **Create Submodel**, enter **Address Components** as the name for the submodel.

8. Click **OK**.

ER/Studio Data Architect creates the **Address Components** submodel.
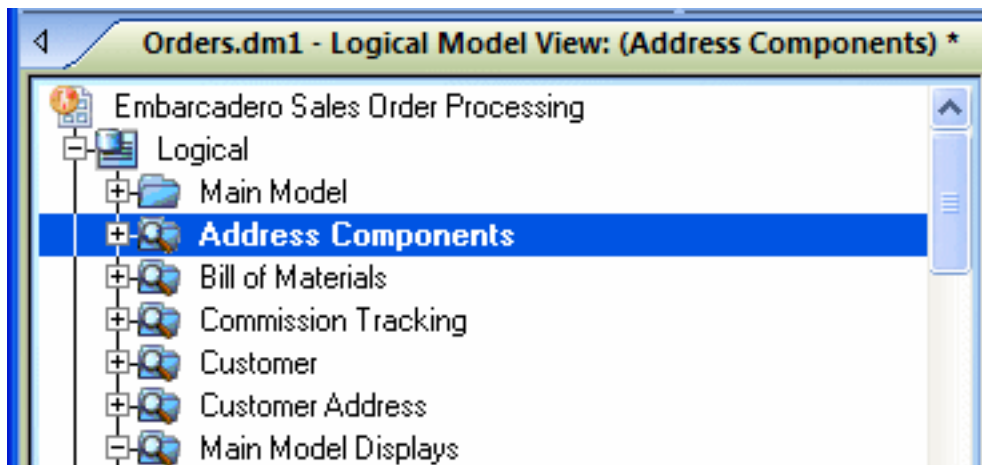
What do the results look like and how do I navigate to the submodel?

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: 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:

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

- o   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 *Orders.DM1* sample model.

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

Tip: Use the steps shown in the last session to do so.

2.   Select the **Main Model** and then click **Model > 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.
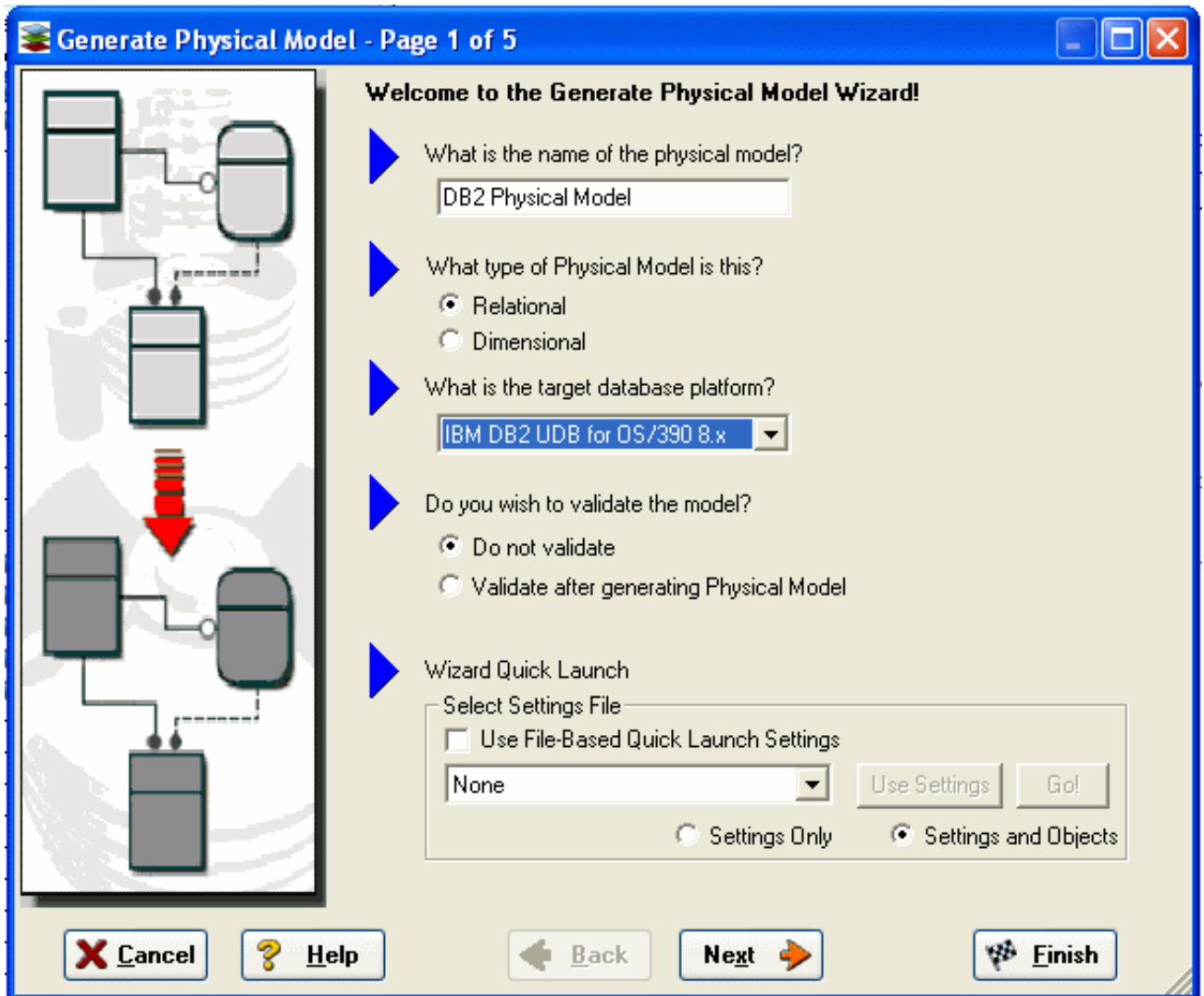
3. Name the new physical model, **DB2 Physical Model** and then select **DB2 UDB for OS/390 8.x** as the target DBMS to generate.

4. Continue through the **Generate Physical Model** wizard, which prompts several questions about how you want your physical model to be 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 Setting* 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 has been 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: 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.
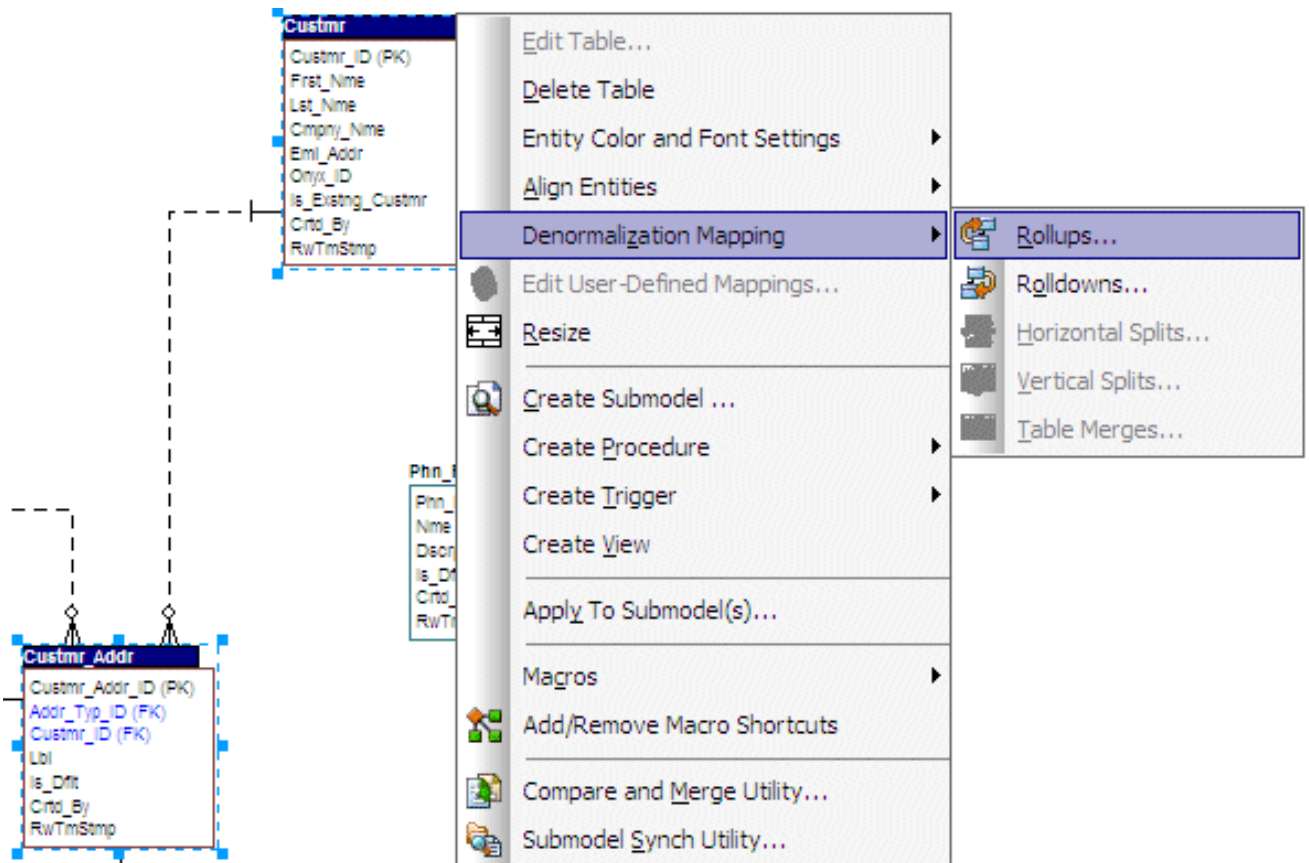
When only one table is selected, the option to split the tables becomes available. The Table Merge option is available when two unrelated tables are selected.

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.
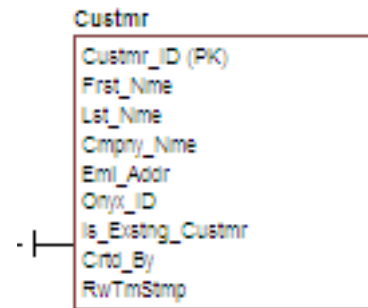
Before the operation, the Custmr table should look like:

1. Open the Orders1.dm1 model you modified and saved in the last 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, type *2* for the number of splits.

5. On Page 2, rename Custmr1 and 2 to *Custmr_East* and *Custmr_West* respectively.
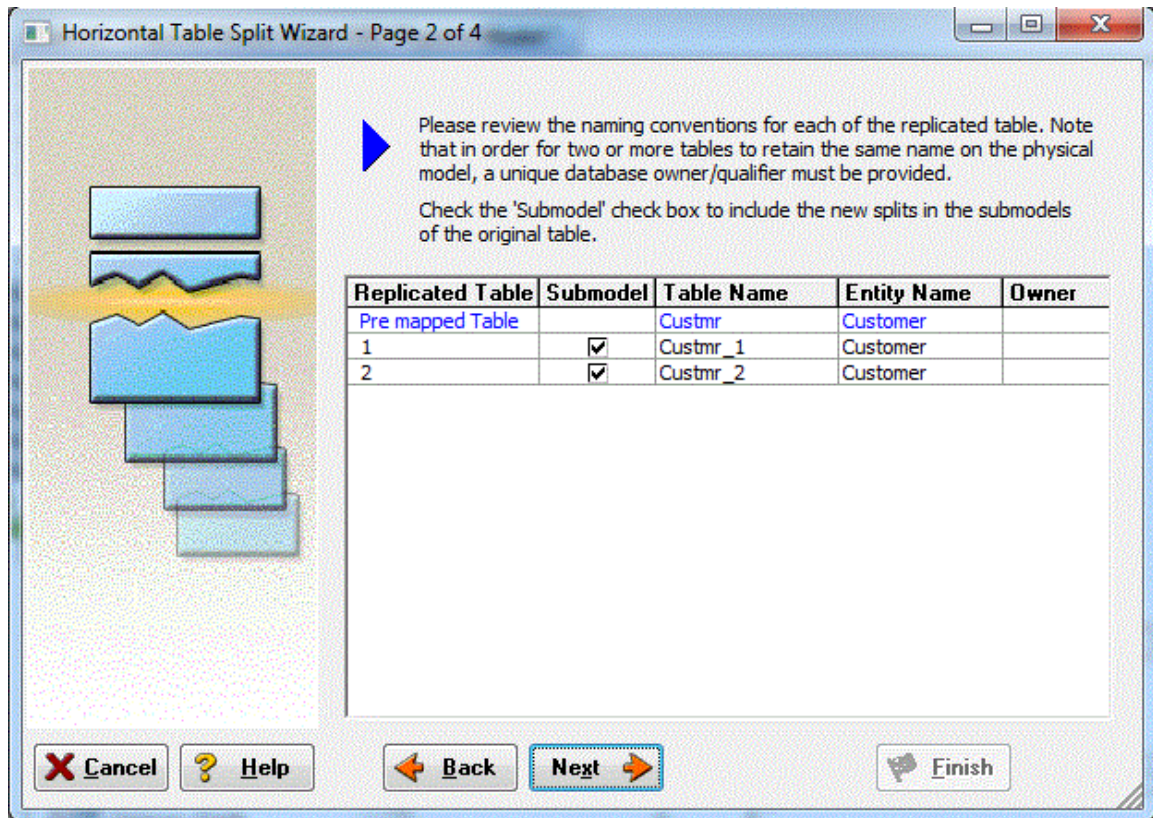
6. On Page 3, click **Next**.

We will keep all the relationships.

7. On Page 4, type a name and definition for the denormalization operation, and then click **Finish**.

Finished! After the split the Custmr table will be 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

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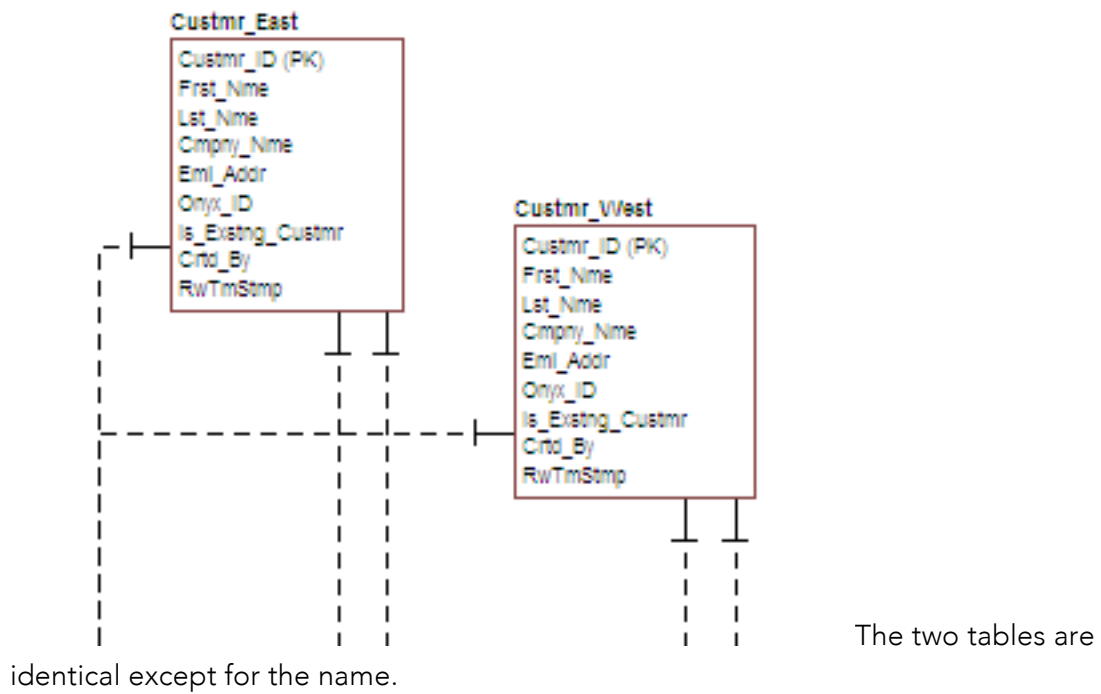
You can selectively choose which attributes are included in the resultant tables by using a vertical split.

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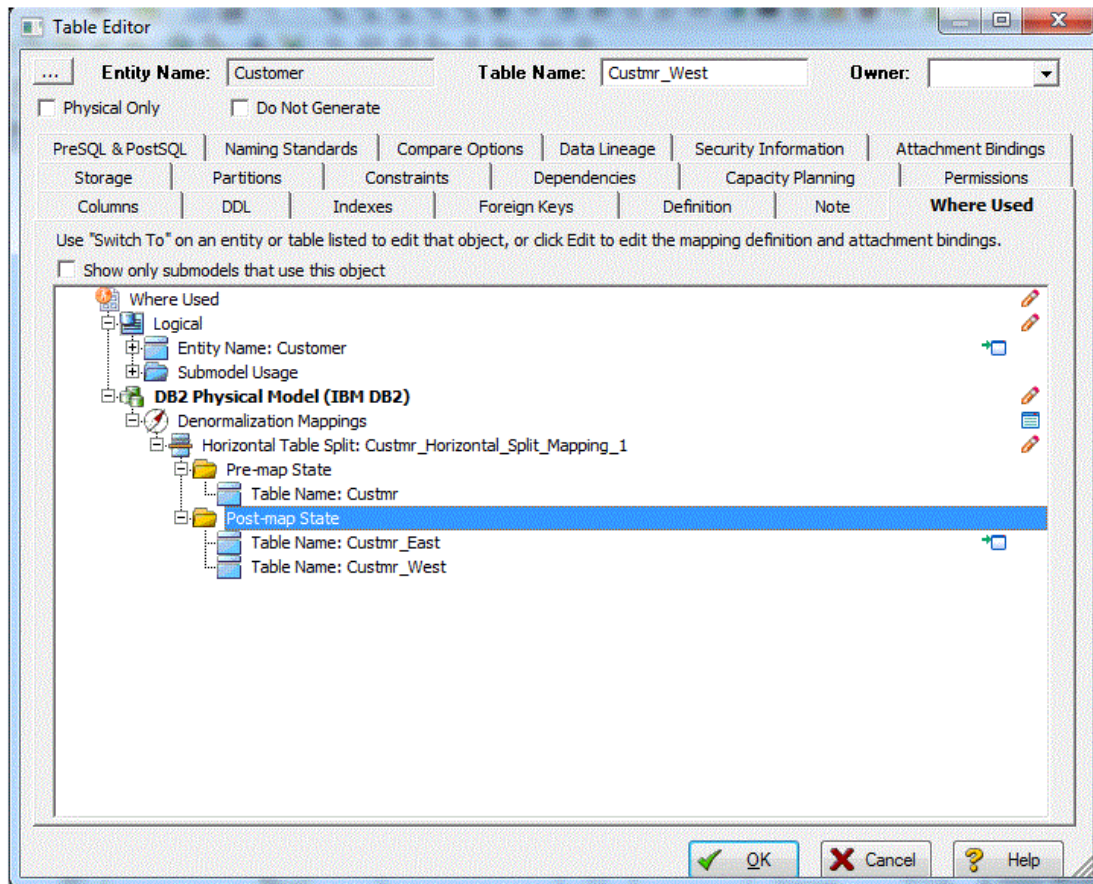
You can use the denormalization information to undo the operation or see the history of what happened. ER/Studio Data Architect tracks the before and after states of these operations. This comes in handy in the next section where we discuss the **Where Used** analysis that can be performed between the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

Tutorials: Finding out How an Entity Maps to the Physical Model

Now that we have 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, navigate back to the Customer entity in the Logical model.

2. To start the Entity Editor, double-click the **Customer** entity.

3. Click the **Where Used** tab.
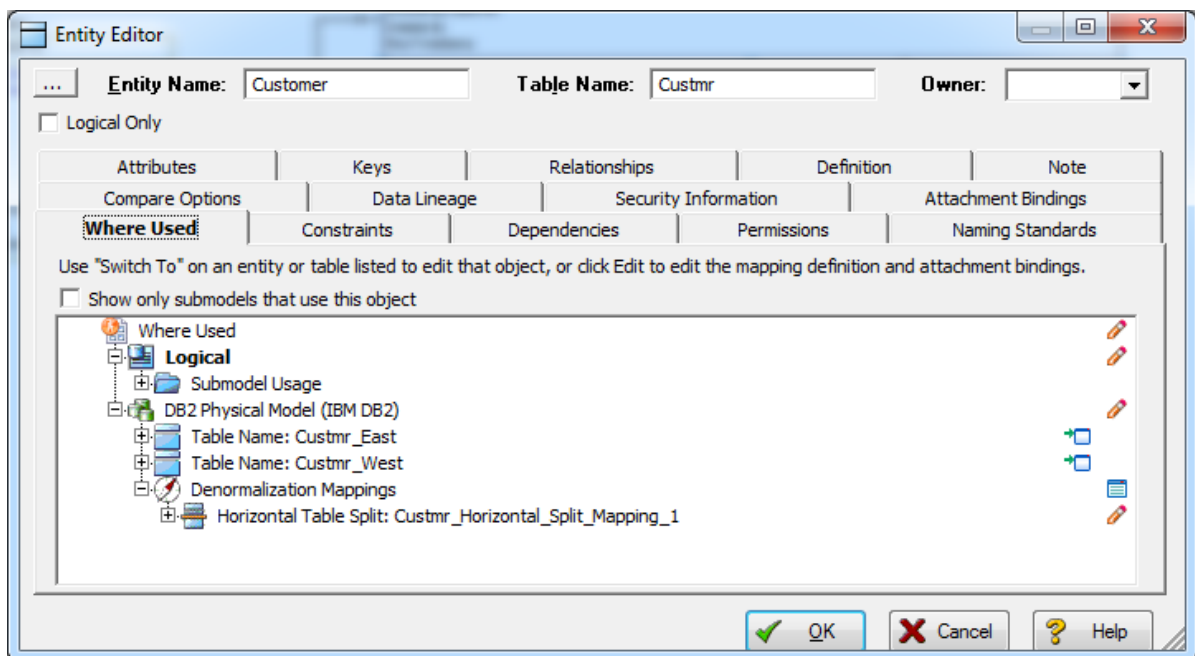
Once the tree is expanded, you can see the lineage of what has happened to the object. Notice that *Custmr_East* and *Custmr_West* are listed as physical implementations of the *Customer* entity. The denormalization mapping object shows how the end result was achieved.

The **Where Used** tab also displays the submodel usage of a particular entity within the logical or physical model. This 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:

- o   Build a logical data model from scratch.

- o   Create a new submodel view to understand how to model on specific parts of a larger Main Model.

- o   Generate a physical model from a logical database in preparation for building a new database.

- o   Denormalize objects in the physical model.

- o   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:

- o **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.

- o **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.

PRE-REQUISITE:

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.
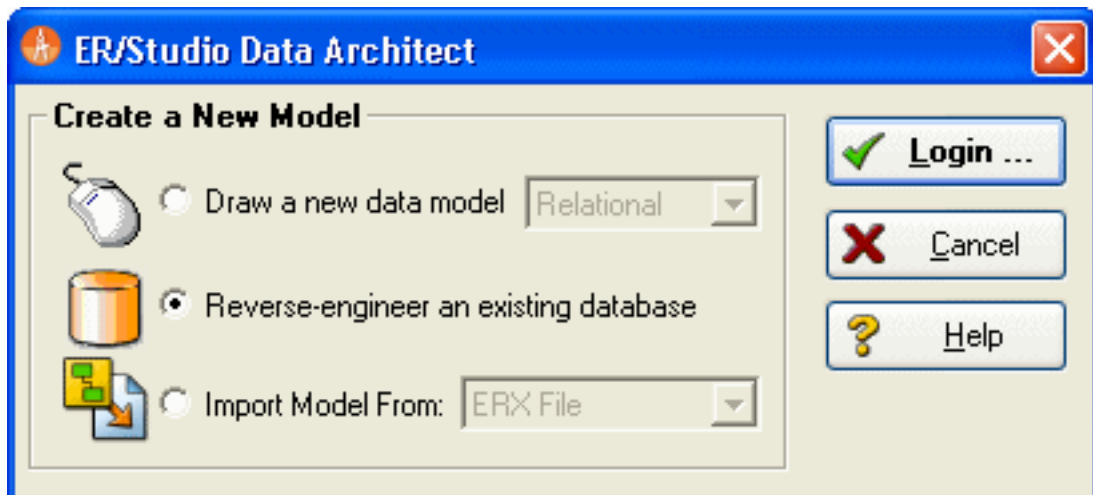
2. Select *Reverse-engineer an existing database*.

3. Click *Login*.

▪ You can reverse engineer the database from either an ODBC datasource or via Native RDBMS client connectivity. In this example, Native Connectivity to Microsoft SQL Server will be demonstrated.

4. Type the relevant connectivity information such as the data source name, and user name and password and then click *Next*.
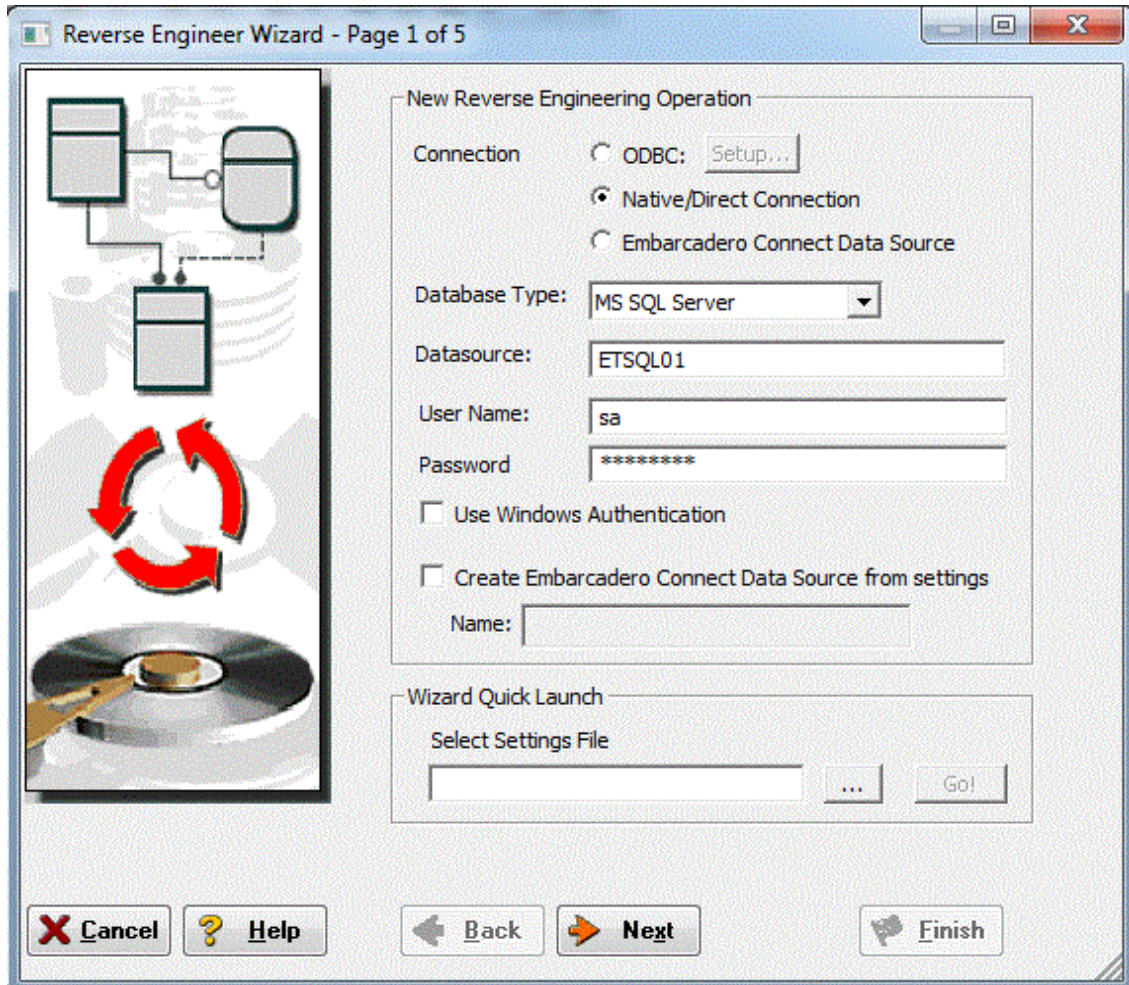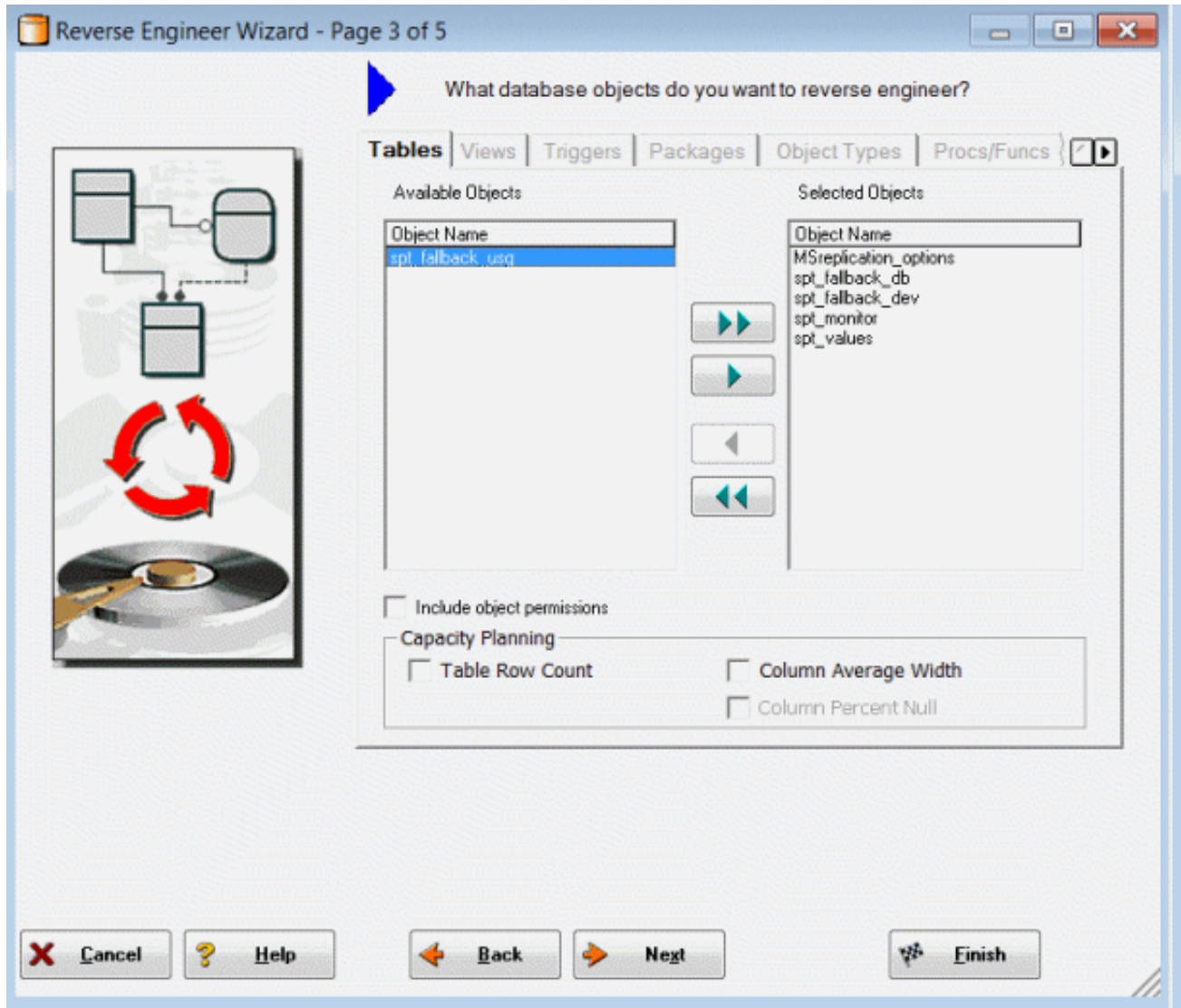

5. Walk through the *Reverse Engineer Wizard* selecting the objects, options, and layout preferences for the model.

Note: A new feature using ER/Studio Team Server data sources is available in the Reverse Engineer Wizard. Please see ER/Studio TS documentation for more details.
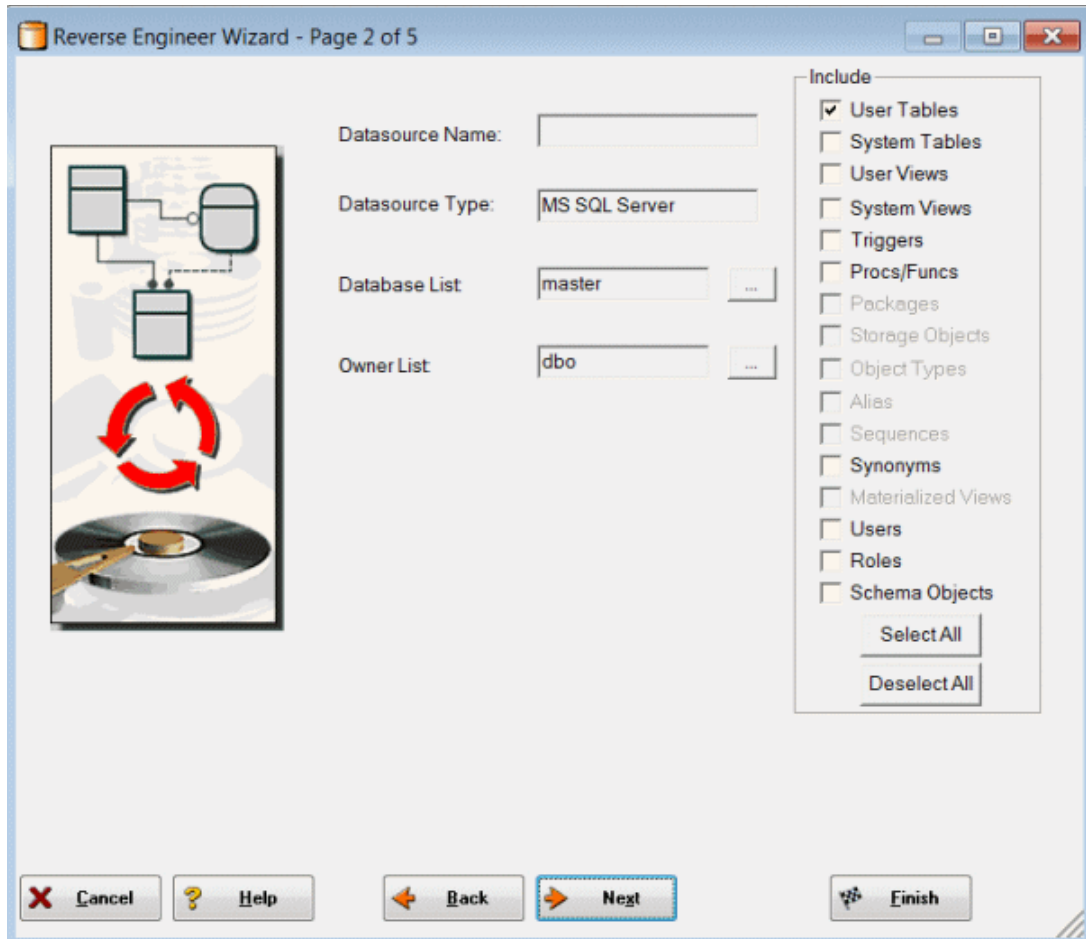
6. Continue through the wizard to select layout styles and other preferences.

**Reverse Engineer Wizard - Page 4 of 5**

ER/Studio Data Architect can infer Referential Integrity when none is declared in the database. Clicking the options below will create relationships between entities in your diagram.

☐ Infer Primary Keys
☐ Infer Foreign Keys from Indexes
☐ Infer Foreign Keys from Names
    ☐ Ignore Case
☐ Infer Domains (Create a Domain for each Column)

ER/Studio Data Architect can ensure all objects referenced by those you've selected for reverse engineering are also included.

☑ Reverse Engineer View Dependencies
☑ Reverse Engineer Other Dependencies (e.g. Procedures, Triggers, etc)
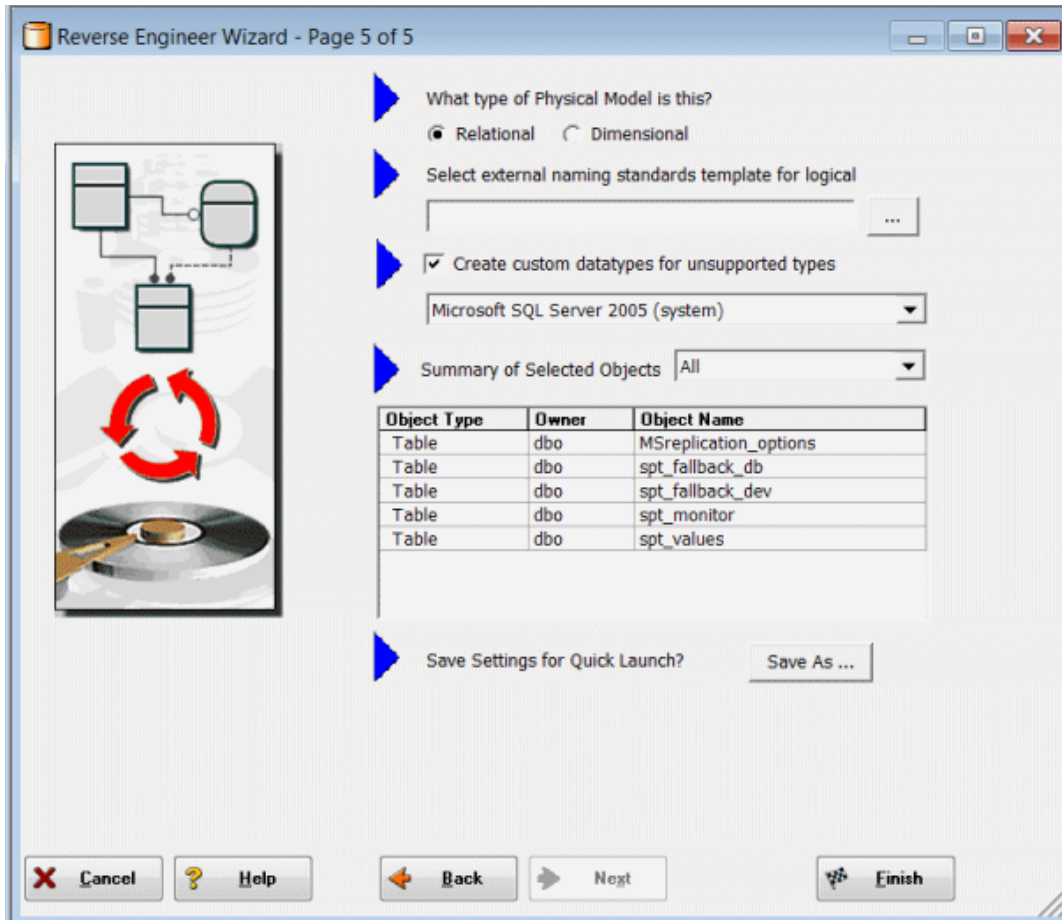
Select the Initial Layout Option

Circular   Orthogonal   Tree
Hierarchical   Symmetric

Select the Logical View Parser Option
☐ Use Physical Parser
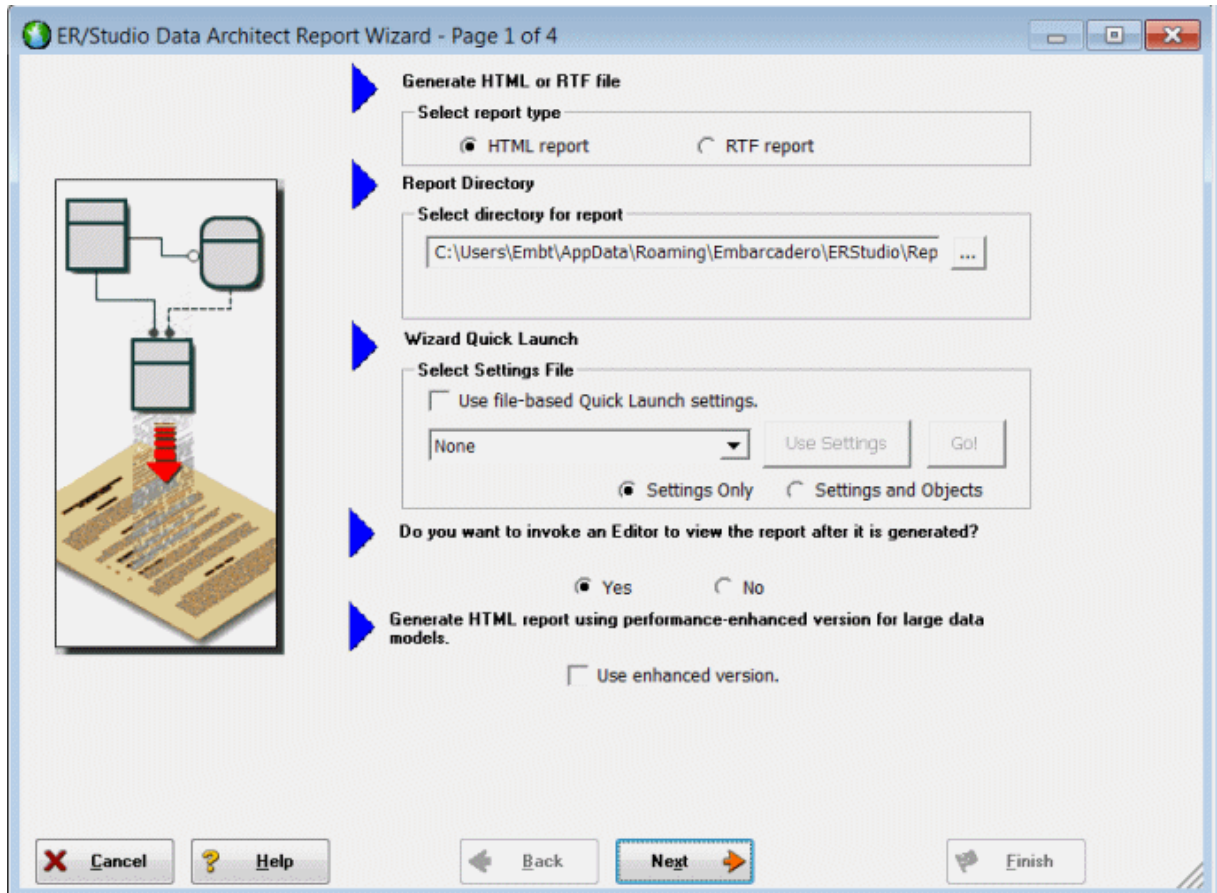
✕ Cancel   ? Help   ← Back   → Next   Finish

7. 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.

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

9. Choose **Tools > Generate Reports**.

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

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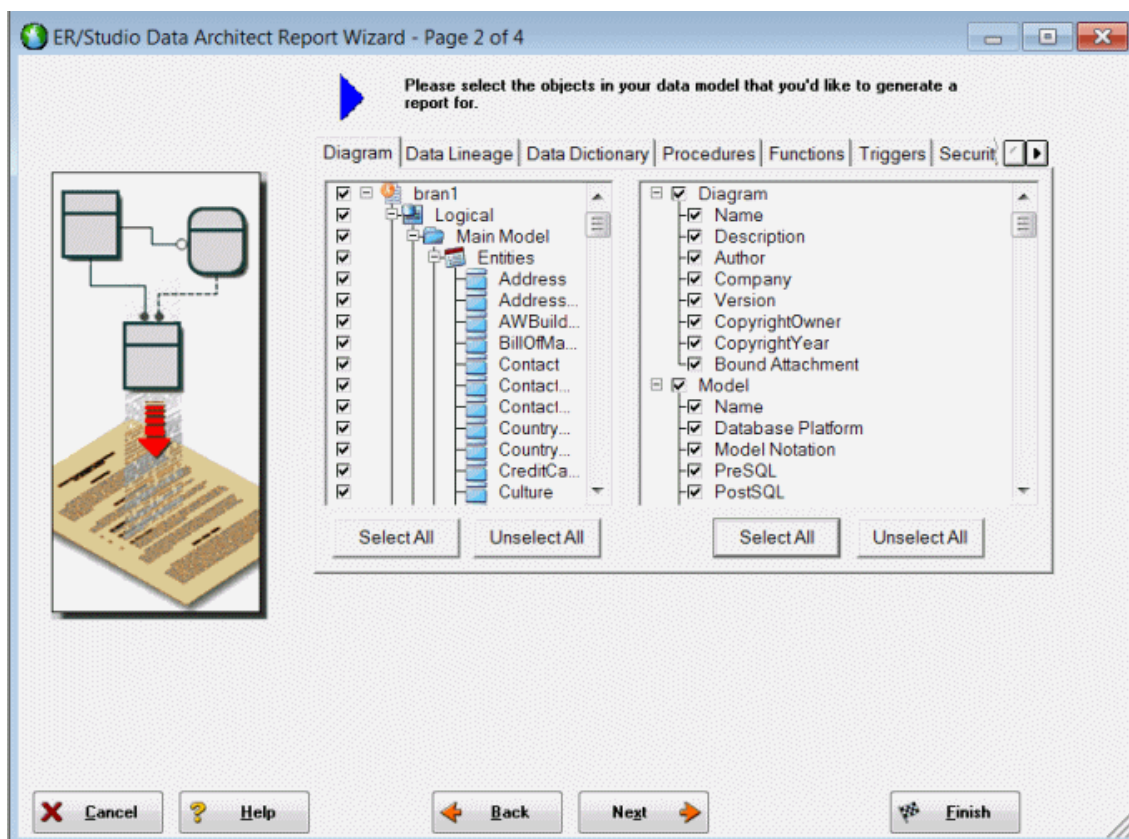
12. Click **Select All** in both areas of the **Data Dictionary** and the **Procedures** tabs, and then click **Next**.

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 2005, such as Oracle11g and IBM DB2 LUW 9.x 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.

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

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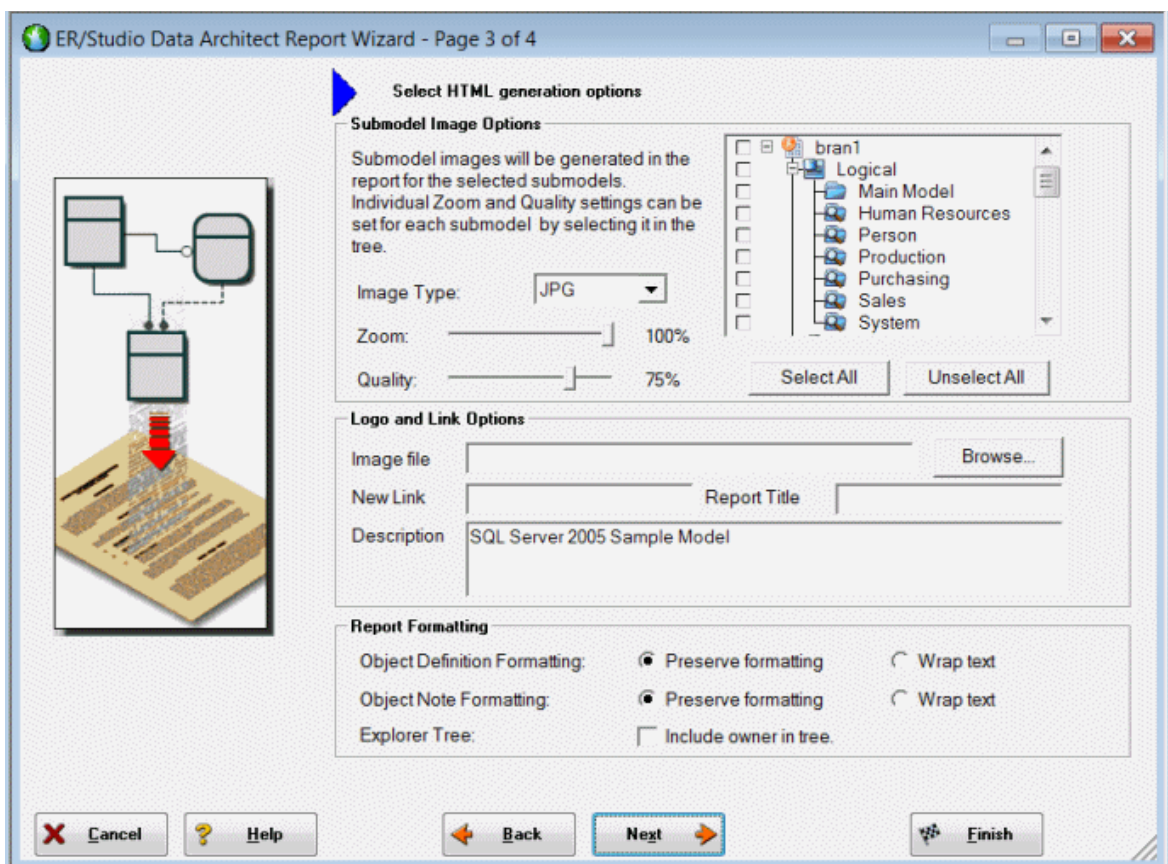
Because HTML formatting can be included in object definitions, you can also choose to preserve the formatting specified on the *Definitions* tab of the various object editors.

14. Click **Next** to advance to Page 4 of 4, 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.

15. 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:

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

o 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 ERStudio 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:

- o [Creating a Data Flow](#)

- o [Creating a Data Movement Rule](#)

- o [Defining External Source and Target Systems](#)

- o [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.
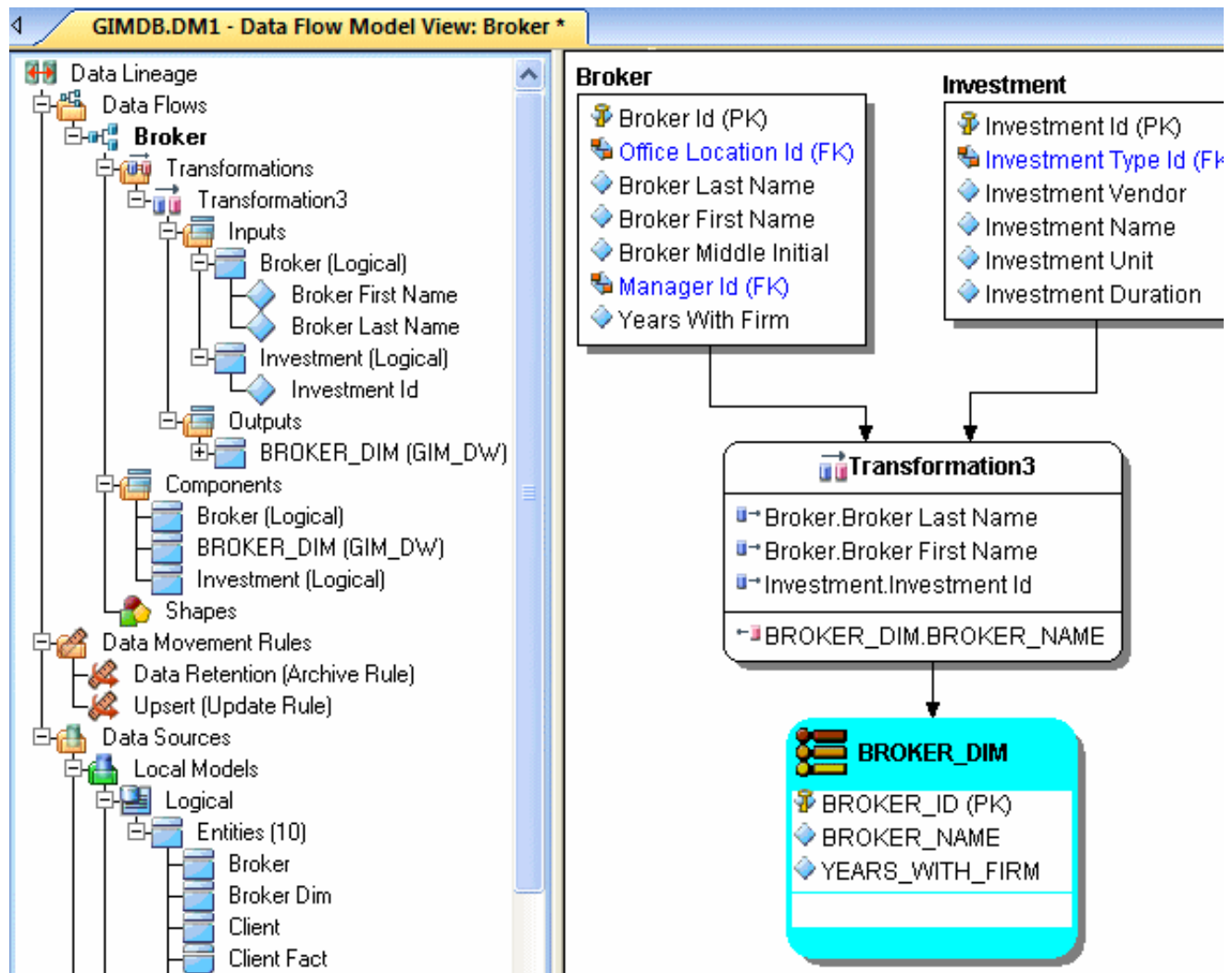
Create a Data Lineage Data Flow

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*.

---
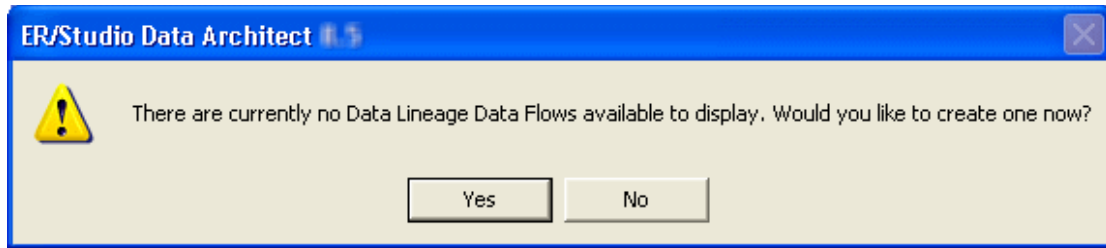
3. Click *Yes*.

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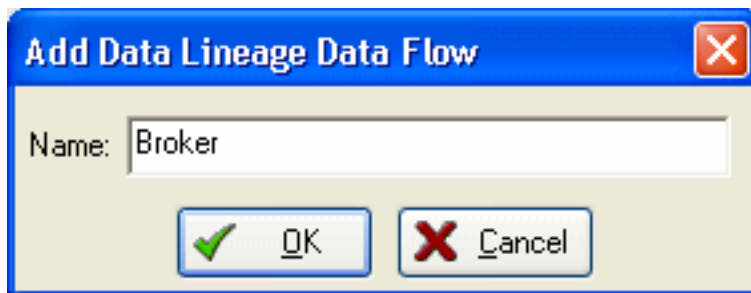
Note: The name that appears in the diagram title tab at the top of the application window is appended with*: data flow* name, when you click a task in the *Data Lineage* explorer, such as *GIMDB.DM1 - Data Flow Model View: Broker**.
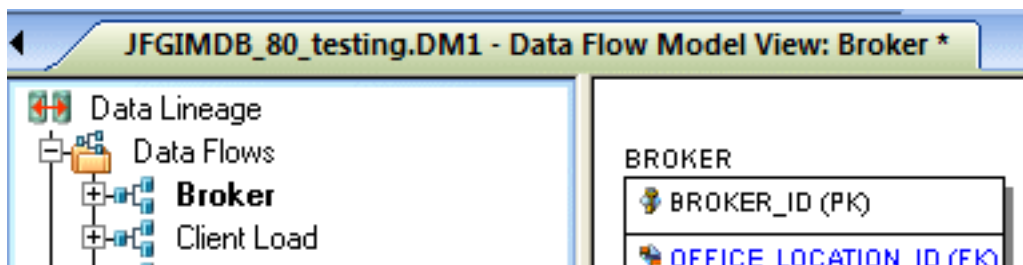

The Data Flow has been created.

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 choose New Data Movement Rule.
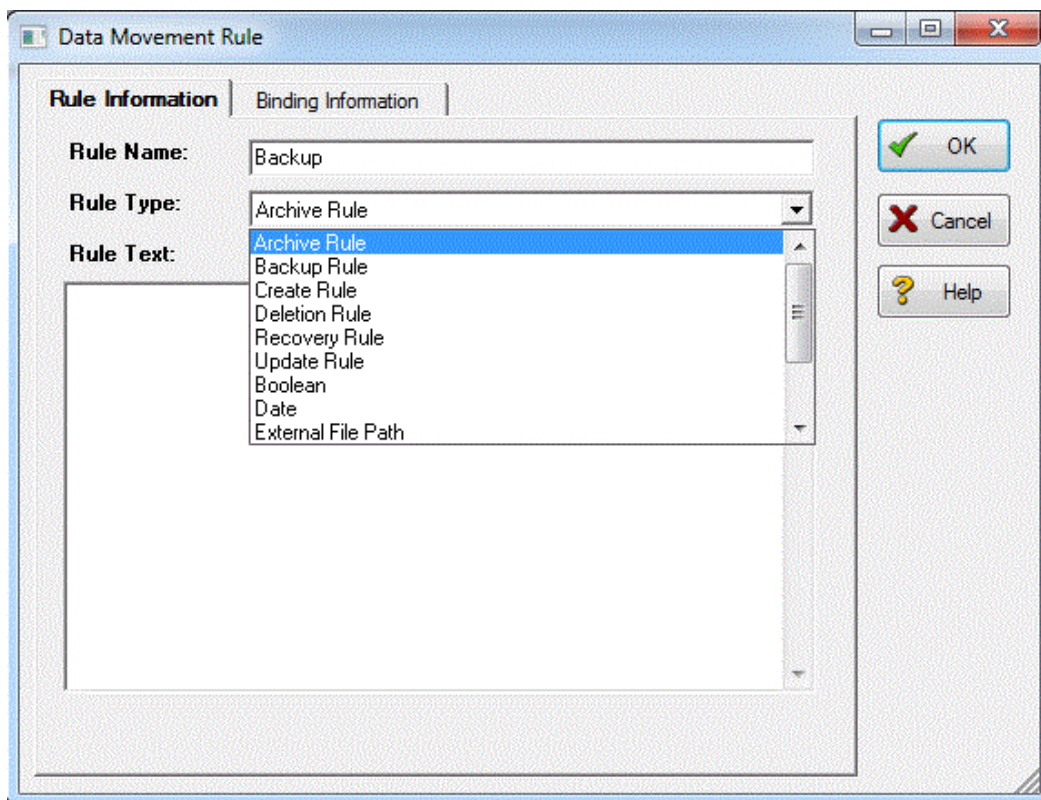
2. Complete the **Data Movement Rule** editor as required and then click **OK** to exit the editor.

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

The following describes options that require additional explanation:

Rule Information tab

- 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.

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, nillability, 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 choose **Import New Source**.
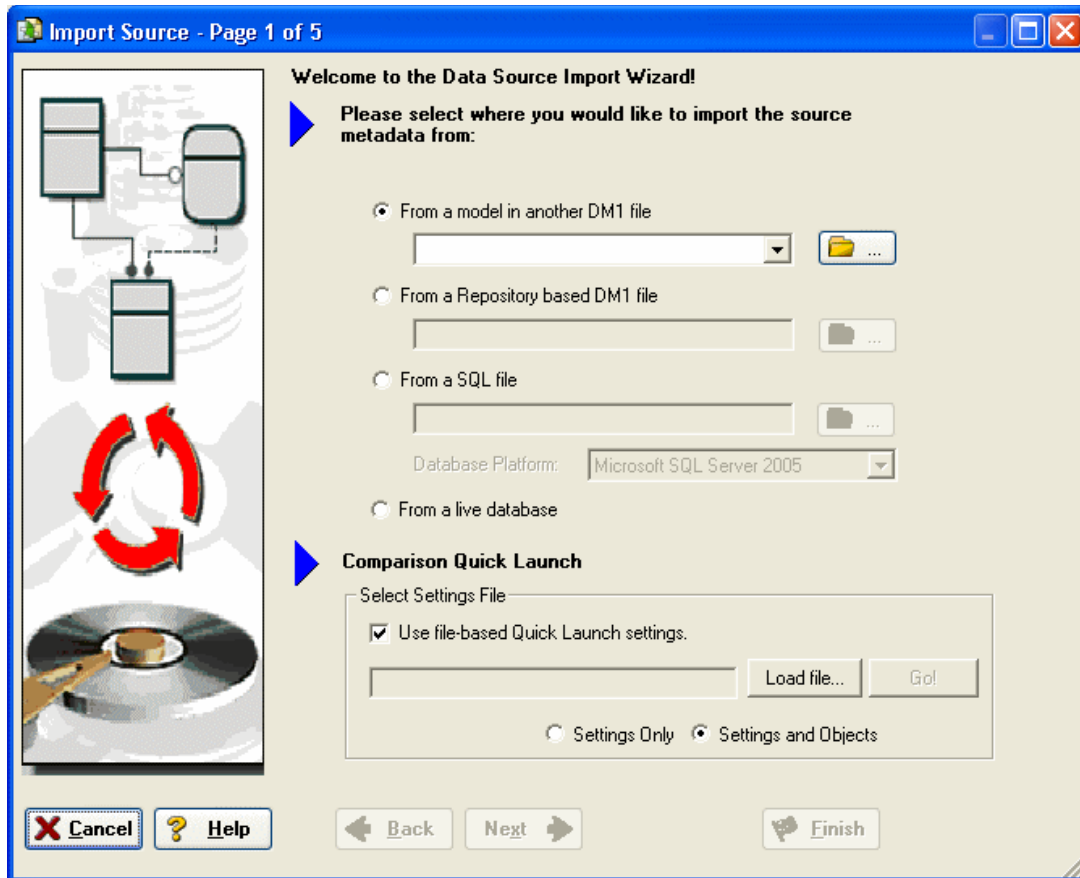
3. Complete the **Import Source** wizard as required and then click **Finish** to import the source.

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.

o **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** table onto the Data Lineage window.
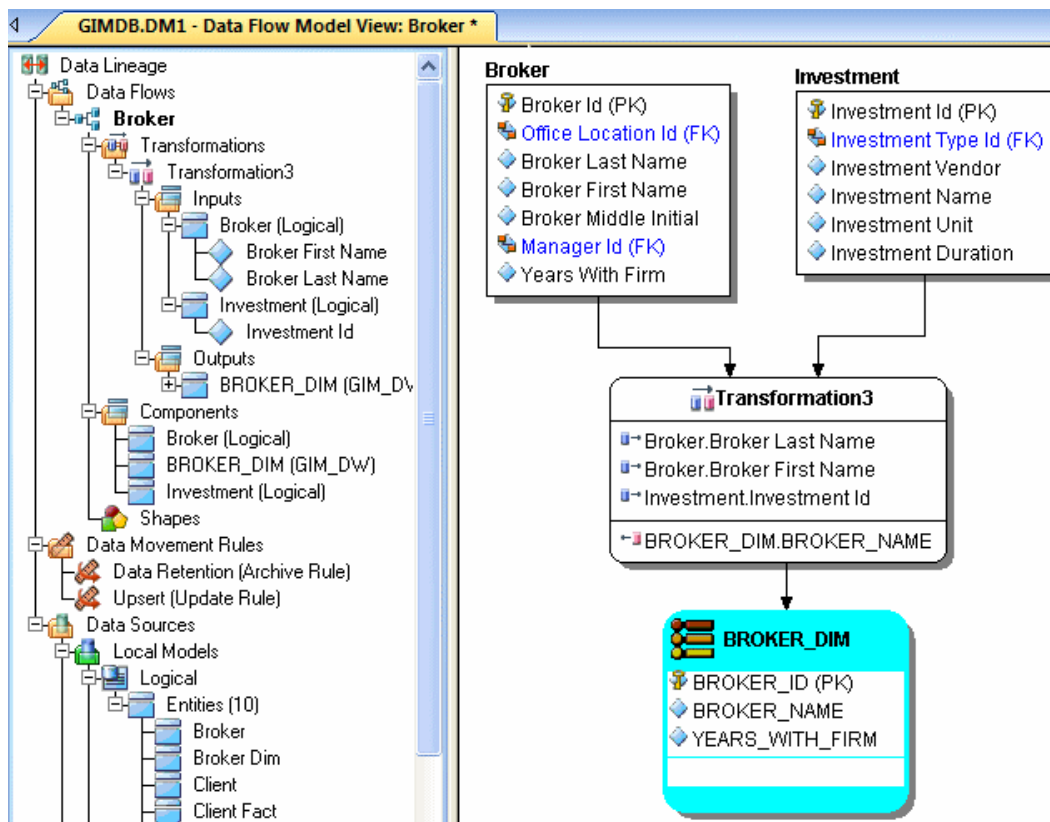

3. To obtain the Transformation Insertion tool, right-click an empty space in the **Data Lineage** window and then click **Insert Transformation**.

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.
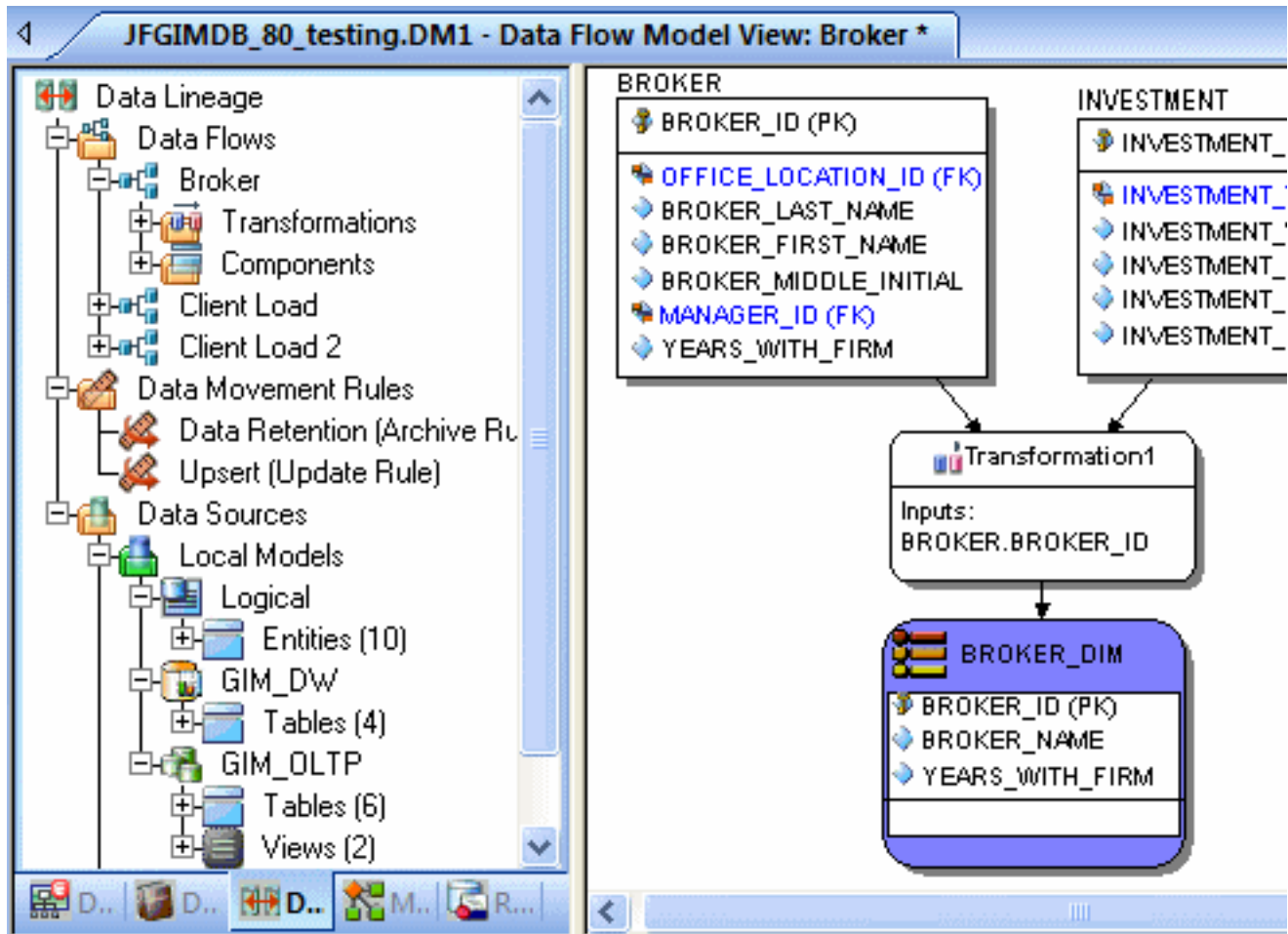
Tip: If the Inputs and Outputs do not display in the diagram as they do in the illustration above, choose **View > Diagram and Object Display Options > Transformation** and then click *Inputs and Outputs Columns*.

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. You are done! 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

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

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

Definition tab

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

o **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.
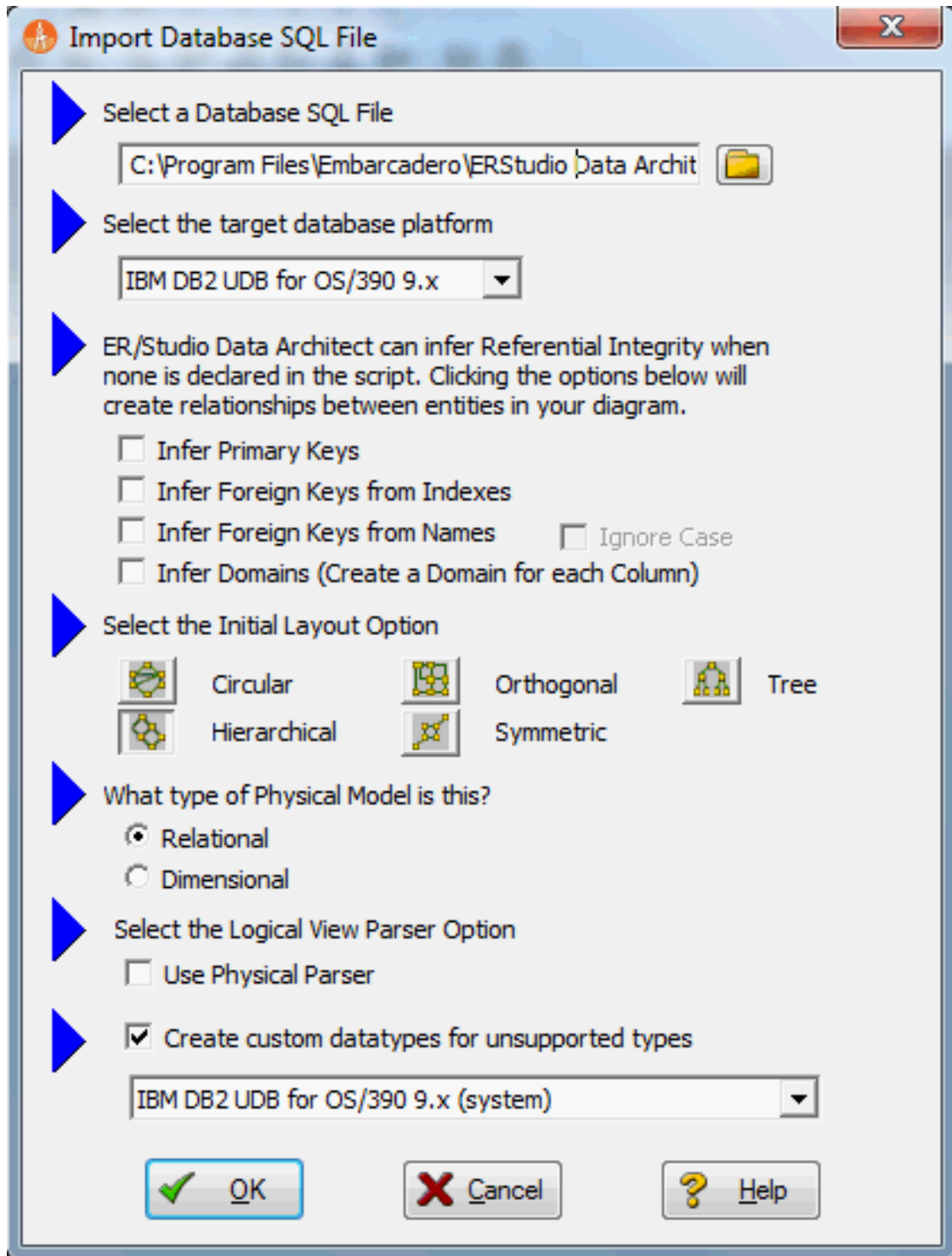
2.   Choose **File > New**.
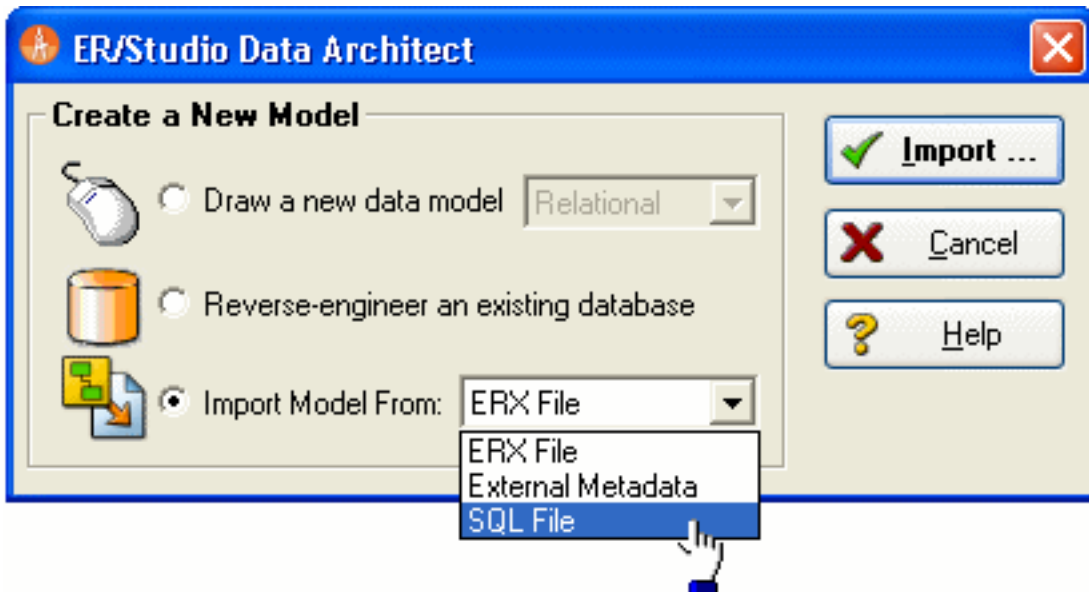
3.   Select **Import Model From:** and then in the import list, choose **SQL File**, then click **Import**.

Note: The ERX File choice enables you to import ERwin 3.5.2 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, click **Sample DDL (DB2 7.x for OS390).SQL**, and then click **Open**.

▪ The full path to this file is:
For Windows XP (No longer supported):
C:\Documents and Settings\<user>\Application

Data\Embarcadero\ERStudio\SQLCode

For Windows Vista and later versions:

C:\Program Files\Embarcadero\ERStudio Data Architect X.X\SQLCode .

Depending on your version, if you are running the 32 bit version on 64 bit Windows, the path may be, C:Program File (x86)...

5. In the **Select the target database platform** list, click **IBM DB2 UDB for OS /390 9.x**.

6. Click **OK**.

Finished! Once the SQL Script is finished importing (as depicted below) the following items will assist you in leveraging a variety of Auto Layout and Navigation Features.
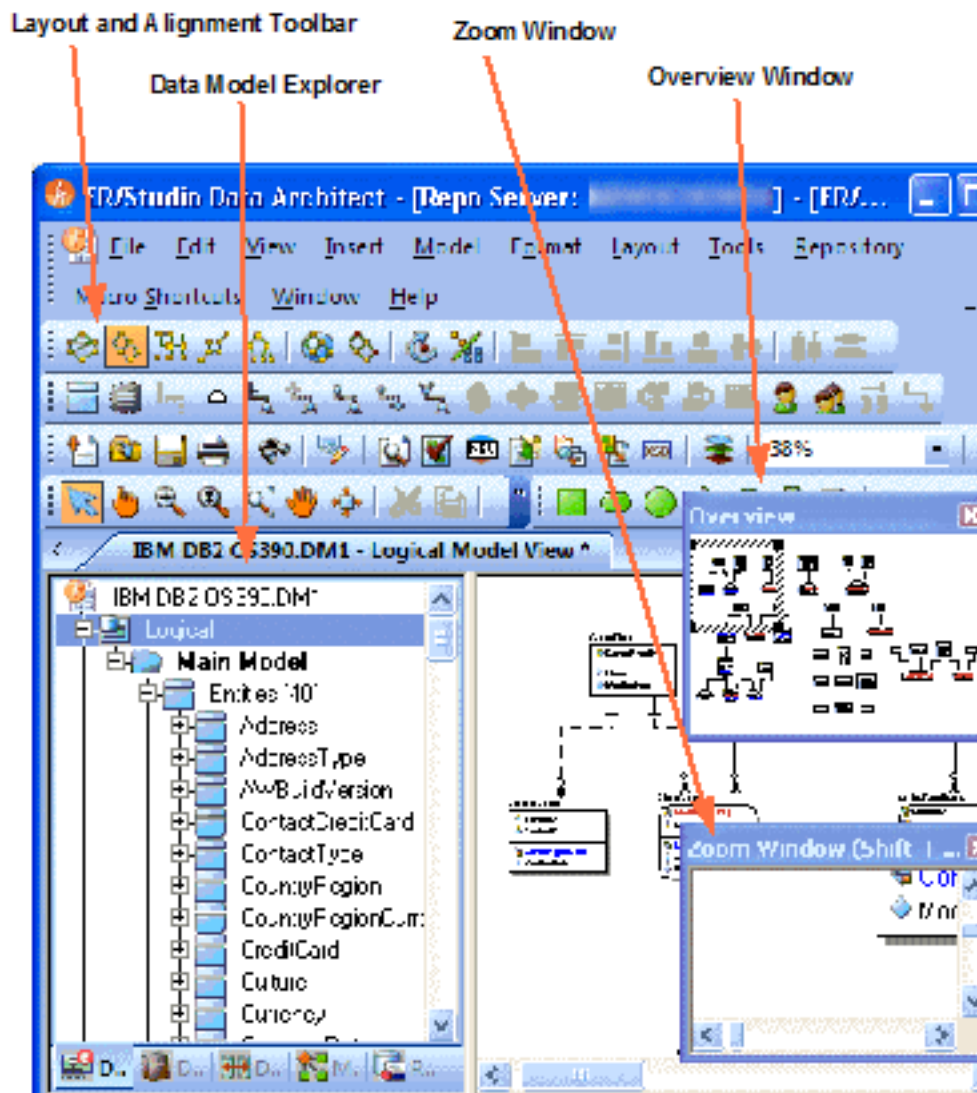
o **Layout and Alignment Toolbar**: Use any of the four Auto Layout styles to change the layout of the diagram with the click of a button. The auto layout styles are all entirely customizable styles. You can also customize the diagram layout via the Layout Properties pages that can be launched by clicking Layout > Layout Properties. In addition, you can have multiple layout styles in a model. Select some 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 will be applied to the entire model.

o **Data Model Explorer**: Click on any object in the **Data Model** explorer and it will automatically be selected in the diagram and focused in both the *Zoom Window* and *Overview* window.

o **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. If the **Overview** window is not already visible, press **F9** to activate it.

o **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.

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.
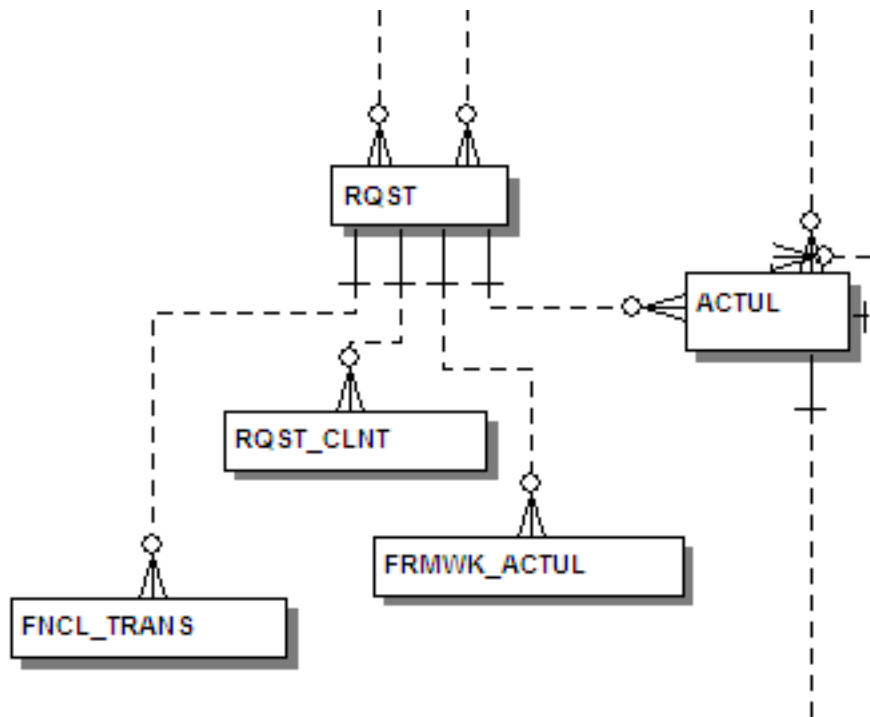
o You can use the Colors & Fonts tool to customize the look and feel further of each model.

Setting the Logical Model Display

1. Select the *logical model* and then on the Diagram Toolbar, click the *Diagram and Object Display Options* tool.

2. In the *Diagram and Object Display Options* dialog, click the *Entity* tab, and then in the *Display Level* area, select *Entity*.

3. 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 **View > Diagram and Object Display Options**.

2. In the **Diagram And Object Display Options** dialog, click the **Table** tab.
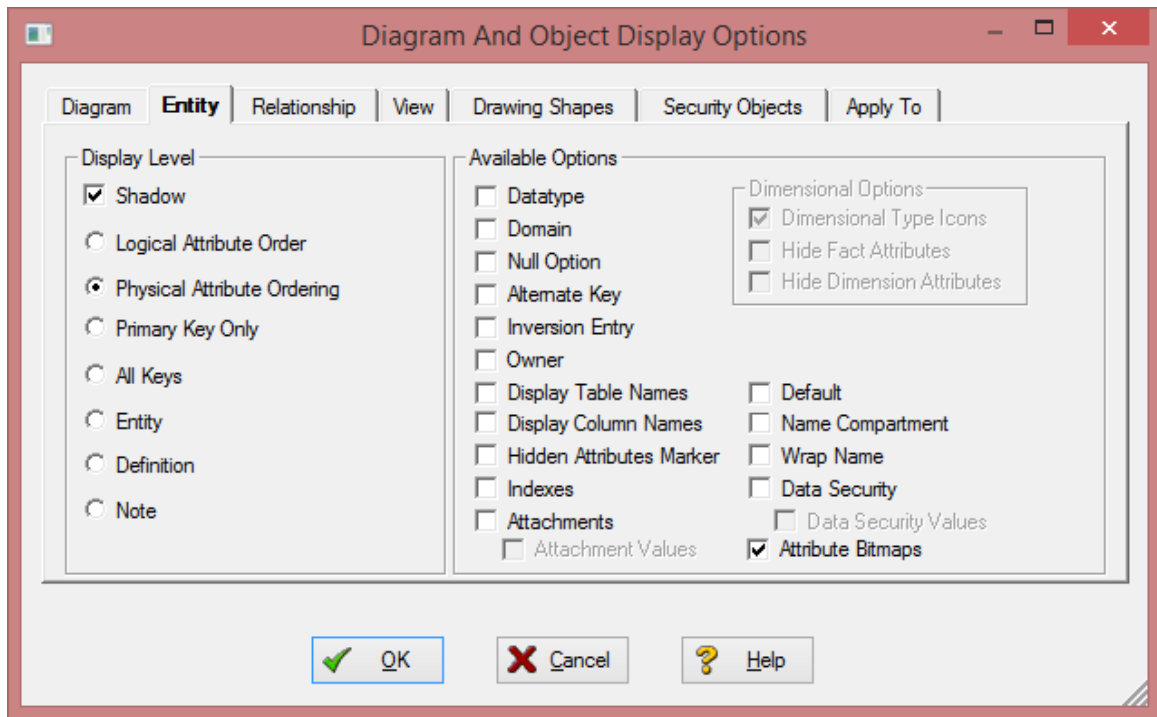
3. In the **Display Level** area, select **Physical Attribute Ordering**.

4. In the **Available Options** area, select the specific properties you want to display.

5. Click **OK**.

The model should now display more details for the physical model, as seen below.
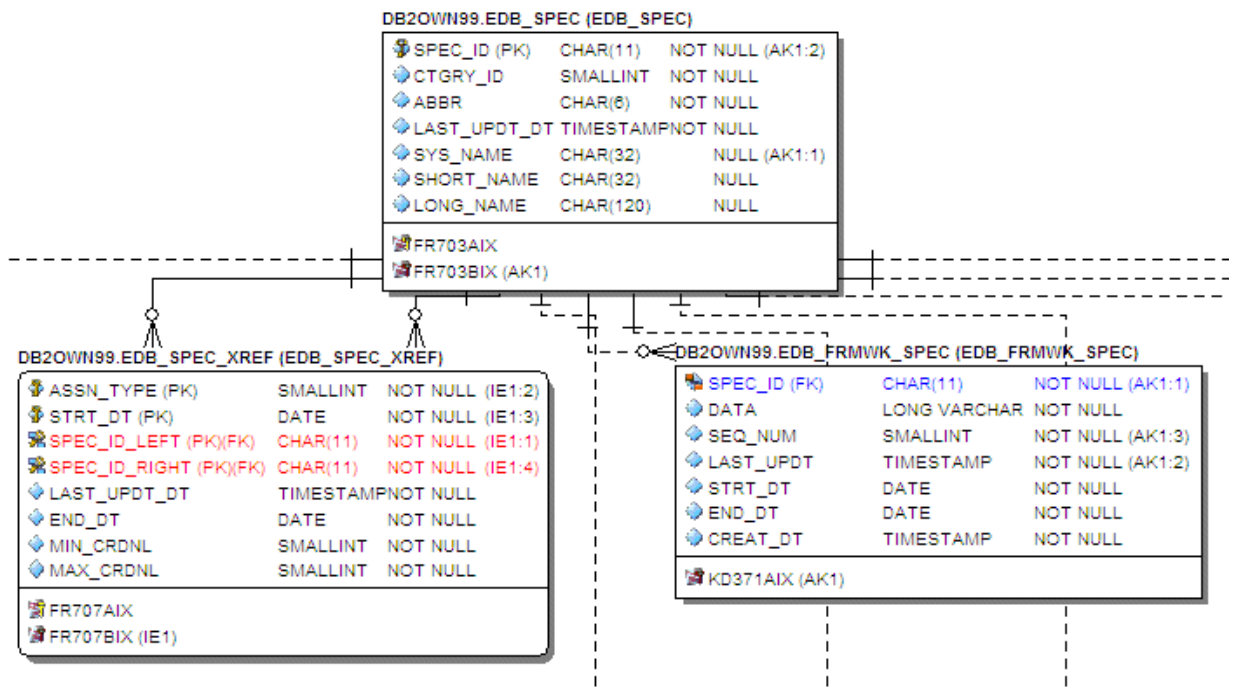
Note: Since the sizes of the objects changed, you may want to change the model layout using one of the ER/Studio Data Architect advanced layout engines. You can also customize the default display properties for new models by choosing **Tools > Options**, and then selecting the desired options on the **Display** tab.

Conclusion

In this session, you have learned how to:

o   Import an SQL file and allow ER/Studio Data Architect to automatically create a diagram.

o   Use a variety of auto layout and navigation tools to enhance the aesthetic experience of the diagram and to improve the data model navigability.

o   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

o   [Wiki Home](#)

o   [User Guide](#)