

**Contents {ewc HLP25632, HLP256\_Tile, banner.bmp}**


This page is for [testing](#) only.

**[Start tutorial](#)**

## Tutorial Menu

# ERwin Tutorial

Click on the blue labels to find out about the activity.  
Click a Try It button to perform the activity using ERwin.

1 About This Tutorial		6 Relationships <a href="#">Try It</a>
2 ERwin Workplace Overview <a href="#">Try It</a>		7 Display Levels <a href="#">Try It</a>
3 Reverse Engineering <a href="#">Try It</a>		8 Subject Areas <a href="#">Try It</a>
4 Entities and Attributes <a href="#">Try It</a>		9 Notation Preferences <a href="#">Try It</a>
5 Domains <a href="#">Try It</a>		10 Fonts and Colors <a href="#">Try It</a>

[More...](#)









Copyright 1998 Logic Works, Inc.


{button Exit Tutorial,CW('menu')}

## Tutorial Menu 2

***ERwin Tutorial***

Click on the blue labels to find out about the activity.  
Click a Try It button to perform the activity using ERwin.

<b>11</b> <i>Physical Model</i>			<b>15</b> <i>Complete Compare</i>	
<b>12</b> <i>Views</i>			<b>16</b> <i>Model Printouts</i>	
<b>13</b> <i>Indexes</i>			<b>17</b> <i>Reports</i>	
<b>14</b> <i>Forward Engineering</i>			<b>18</b> <i>Tutorial Summary</i>	

 *Back...*

Copyright 1998 Logic Works, Inc.

{button Exit Tutorial,CW('menu')}

## Path

The default ERwin program installation directory is:

**C:\Program Files\Logic Works\ERwin35\**

The tutorial models will be located in:

**C:\Program Files\Logic Works\ERwin35\Tutorial\**

You may have installed ERwin to a different location. In that case, the tutorial models will be located in:

(installation drive:\directory path)\**Tutorial\**

## Read-Only

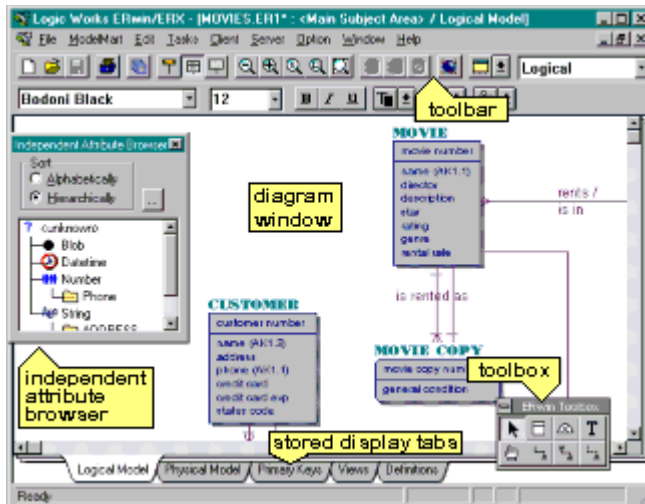
The tutorial models are read-only files. This message box is a reminder that you will not be able to update the files with any changes you make. You may, however, save your changed models to a different file using **File|Save As**.

## Arranging the Workspace

**Hint:** While you are working through a Try It, if your workspace is too crowded, here are a few tips:

- Change your display setting resolution to a minimum of 800 X 600 pixels; the higher the number, the more room you'll have in the ERwin workplace.
- Use CTRL+B (toggle) to temporarily hide the Independent Attribute Browser or resize it by dragging its sides.
- Use CTRL+T (toggle) to temporarily hide the Toolbox.
- Drag one or more objects to another location in the ERwin workplace.
- Use the zoom icons on the toolbar to change the magnification of the diagram.

## Activity 2: ERwin Workplace Overview {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

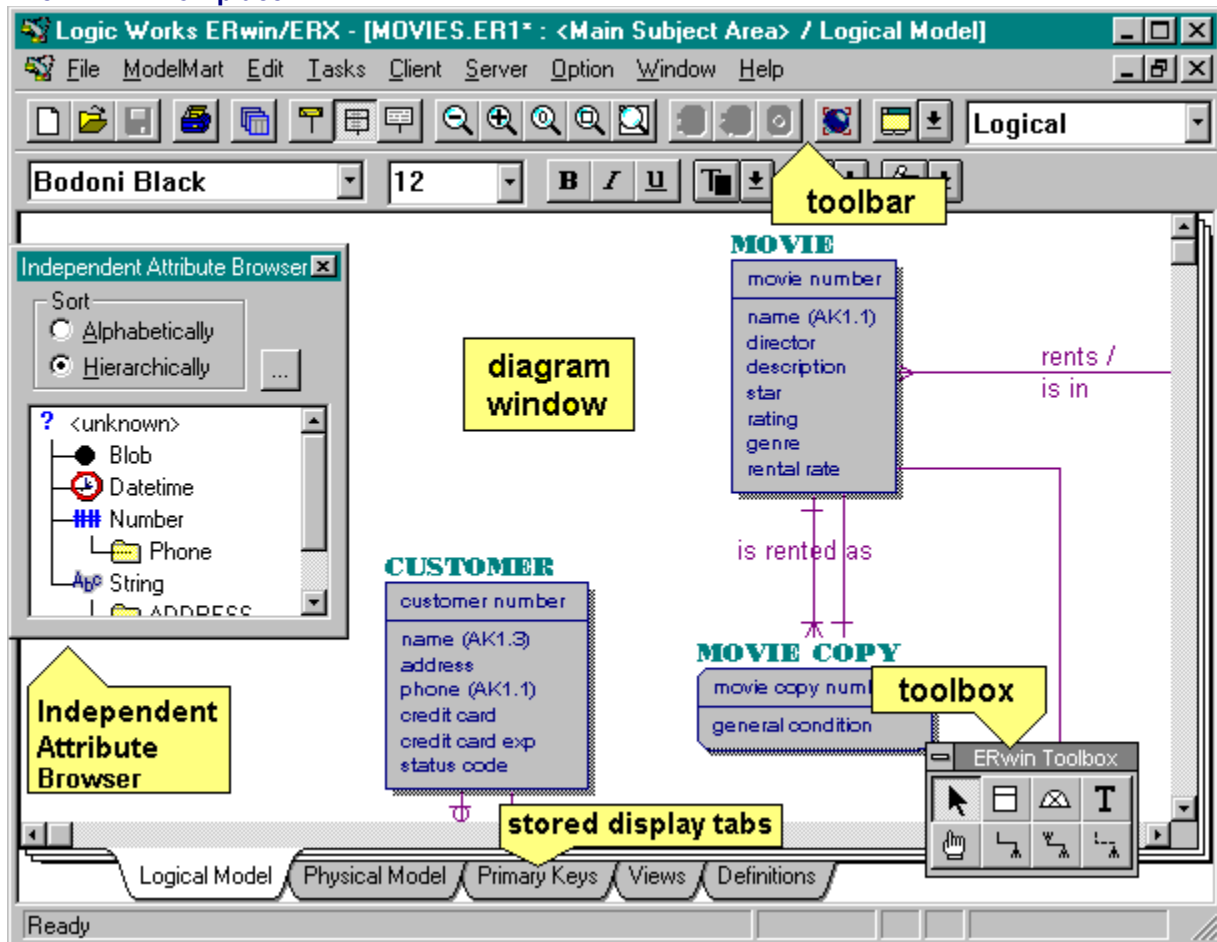
### Welcome to the ERwin Workplace

The ERwin workplace resembles the desktop of many Windows-based drawing and painting tools. It includes a:

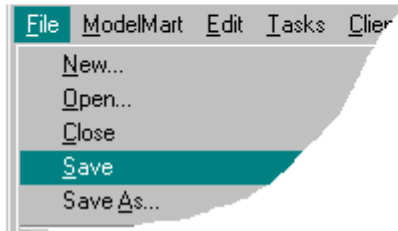
- Drawing area, known as the diagram window
- Toolbar
- Toolbox
- Independent Attribute and Independent Column Browser



## The ERwin Workplace



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on buttons for definitions.

### The ERwin Toolbar

The ERwin toolbar is located near the top of the ERwin window, and it contains a group of buttons that you use to perform common tasks. Each toolbar button is a shortcut that you can use instead of selecting a menu option. For example, you can use the Save toolbar button instead of choosing Save on the File menu.

The availability of the toolbar buttons varies, depending on the object that you have selected and whether you are working in the logical or physical model. You can view a description of each toolbar button by placing the mouse cursor on each button.

To see a description for a toolbar button in the figure on this page, simply click on the buttons.





## The Toolbar



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Model Option List

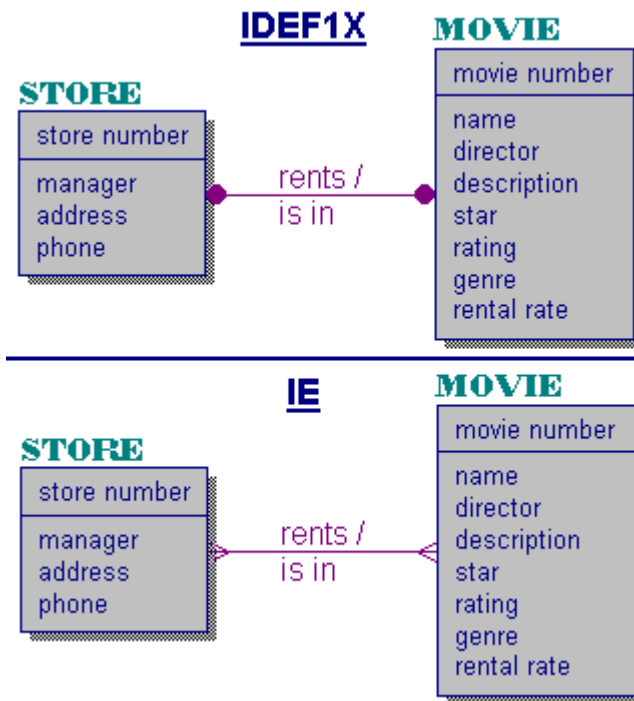
You can easily switch between the logical model and the physical model by selecting the model type from option list on the ERwin toolbar.

- The logical model supports the needs of the **data modeler** who must represent business information and define business rules.
- The physical model supports the needs of the **database administrator**, who focuses on the physical implementation of the model in a database.

You will learn more about the model types in the activities that follow.



Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



**Notation Preferences**

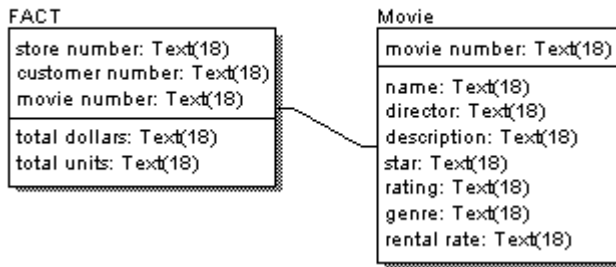
When you work in the logical or physical model, you can choose whether ERwin uses **Integration DEFINition** for Information Modeling (IDEF1X) notation or **Information Engineering** (IE) notation.

The IDEF1X and IE notations use different symbols to represent entity and table relationships.

The example to your left illustrates the two notations for the logical model.



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

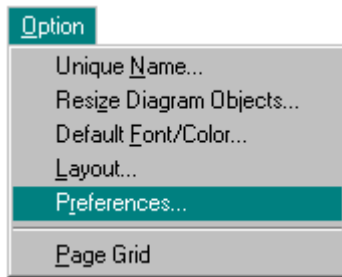


### Dimensional Model

ERwin also includes an additional notation for the physical model that can be used to design a [dimensional model](#).



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



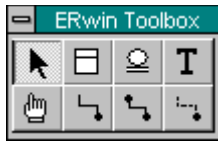
### Options menu- Preferences

The default notation that ERwin uses for both the logical and physical model is IDEF1X, but you can switch to IE notation at any time in either model.

To change the notation, select **Preferences** from the **Options** menu.



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### The ERwin Toolbox

By default, the ERwin toolbox is located in the diagram window. The ERwin toolbox contains a group of tools that you use to perform common tasks such as creating a table/entity or creating relationships.

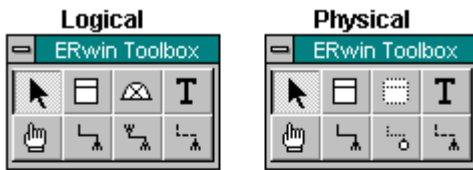
You can display or hide the toolbox by using CTRL+T as a toggle.

To see a description for a toolbox tool in the diagram to your left, simply click on the button.

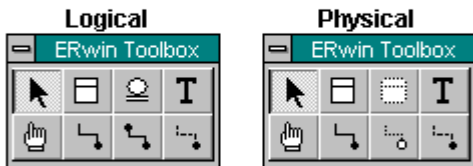


## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

### IE Notation



### IDEF1X Notation



### Dimensional Modeling Notation

Logical

N/A

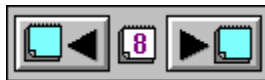
Physical



### **Toolbox Changes**

The toolbox changes when you switch from **Information Engineering (IE)** to **Integration DEFINition** for information modeling (IDEF1X) notation and some of the tools change for logical and physical model. A different toolbox is available for the [Dimensional Model](#).

For more information on each toolbox tool, see ERwin's Online Help.

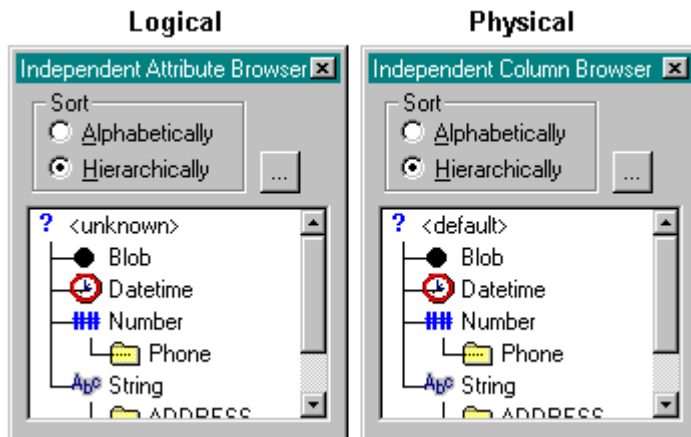


## **Dimensional Model**

A physical model used to design data warehouses.



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

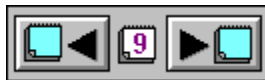


### Independent Attribute or Independent Column Browser

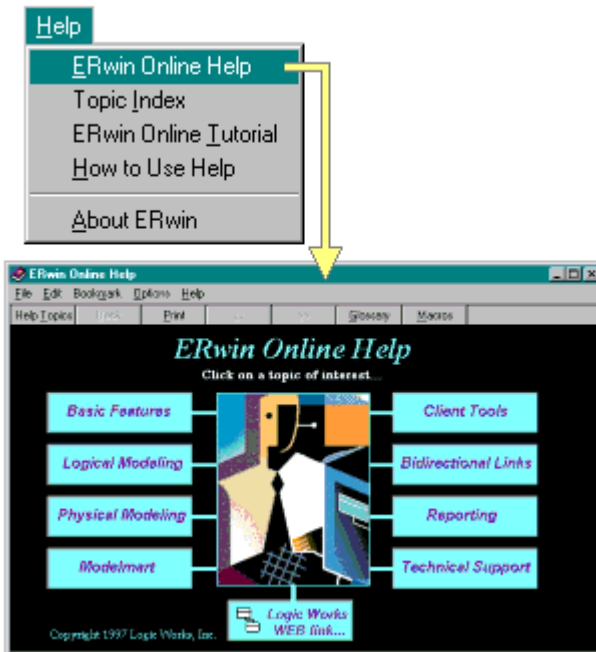
The **Independent Attribute Browser** contains a list of reusable independent attributes that you can place in any entity in your model.

Similarly, the **Independent Column Browser** contains reusable independent columns.

The browser automatically appears when you open ERwin. Use CTL+B as a toggle to show or hide the browser.



## Activity 2: ERwin Workplace Overview, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

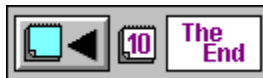


### ERwin's Online Help

When you have a question while working in ERwin, help is only a click away. To access **ERwin's Online Help**, click the Help menu on the ERwin menu bar then click one of the options to find a particular topic.

You can also click F1 to view context-sensitive help for the current dialog or highlighted menu option.

To practice displaying the toolbar and switching between models, click the **Try It** button below.



Try It

Next  
Activity

### Workplace Overview Try It

In this exercise, you will practice displaying the toolbar and switching between models.

To begin, you will open a model which has been provided for this tutorial activity.

Click **Next Step >** to continue.

Cancel

Next Step ►

### STEP 1: Open the sample model

■ Select the model

path\02-tutor.er1

from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

Check that **Logical** model is displayed in the option list on the ERwin toolbar.



You may need to rearrange the objects in the workplace...

Click **Next Step** >

◀ **Prev Step**

**Next Step** ▶

## STEP 2: Display and hide the toolbar

Click on the Window menu. Notice whether **Toolbar** is checked or not.

Click on **Toolbar** to toggle it on or off.  
Toggle it on before continuing.

Click **Next Step >**



Next Step ►



### STEP 3: Switch to the physical model

Select the **Physical** model option from the list in the ERwin toolbar.

Select the **Logical** model option from the list in the ERwin toolbar.

Notice how the model changes as you switch between models. For example, the logical names switch to physical names and the Independent Attribute Browser switches to the Independent Column Browser (if the Browser is *not* visible, press Ctrl+B.)

Click **Next Step >**



Next Step ►

#### STEP 4: Change the notation

Choose **Preferences...** on the Option menu.

Click the **Methodology** tab.

Click the **IDEF1X button** in the Logical Notation group box.



Click **OK**.



Click **Next Step >**

**Next Step ▶**

#### STEP 4: Close the Model


Click  below to verify that you have performed the activity correctly, then click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save**  
**as .ER1 file** then click **OK**.

To close without saving, select **Close**  
**without saving** then click **OK**.



Click  below to close this Try It card.

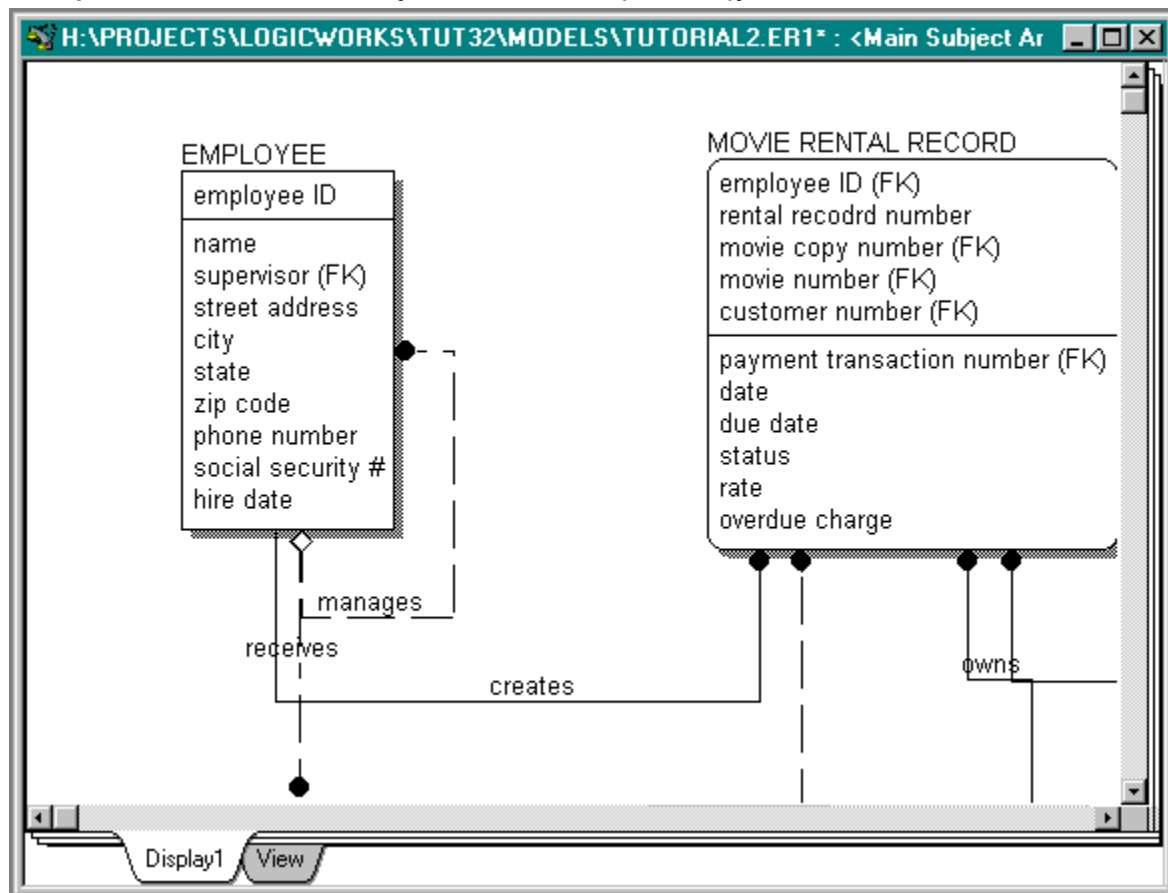
- The End -

 Prev Step

 Done

 Check

Workplace Overview Check {button Close,CW("check")}



**New Diagram Button**

Displays a dialog to let you choose a template from which to start a new ERwin diagram.

### **Open Diagram Button**

Displays a dialog to let you open an existing ERwin diagram.

**Save Diagram Button**

Displays a dialog to let you save changes to the current diagram.

**Print Diagram Button**

Displays a dialog to let you print all or part of the current diagram.

### **Report Browser Button**

Opens the Report Browser.

**Entity/Table Display Level Button**

In the logical model, displays only the name of each entity inside an entity box. In the physical model, displays only the name of each table inside a table box.



### **Attribute/Column Display Level Button**

In the logical model, displays the attributes for each entity, with the primary key attributes above the line and the non-key attributes below the line. In the physical model, displays the column name and datatype for each table. Datatype only displays *if* the Column Datatype option is enabled.

**Entity/Table Definition Display Level Button**

In the logical model, displays the definition for each entity. In the physical model, displays the comment for each table.

### **Zoom Out Button**

Decreases the magnification in increments of 25 percent.

**Zoom In Button**

Increases the magnification in increments of 25 percent.

**No Magnification Button**

Resets the magnification level to 100 percent.

**Fit Model Button**

Fits the entire diagram in the diagram window.

### Select Rectangle to Fit Button

Lets you focus on a specific area of the diagram.

**Forward Engineer Button**

Opens the <Database> Server Schema Generation Report dialog to let you create a new database schema from an ERwin data model or save the schema DDL script as an ASCII text file. Available only for the physical model.



### **Complete Compare Button**

In the Physical Model, opens the Complete Compare - Set Options dialog to let you compare your data model with an existing database, schema script file, ERwin data model, or ModelMart diagram.

### **Select Target Server Button**

In the Physical Model, opens the ERwin/ERX - Target Server dialog to let you select a target database management system (DBMS).

**ModelMart Button**

Displays or hides the ModelMart toolbar.

**Create Subject Area Button**

Opens the Subject Area Editor to let you select or change subject area members for your model.

### Select Subject Area List

Lets you select and switch to a subject area in the current diagram.

### **Logical/Physical Model Option List**

Lets you switch between the logical model and the physical model. The Dimensional option indicates that DM (Dimensional Model) notation is selected for the modeling notation for the Physical Model.

**Select Button**

Lets you select objects for editing. You cannot use the Select button to select attributes for editing. You can select entities and relationships for general diagram editing tasks.

### **Independent Entity Button**

Inserts a new independent entity when you click on the diagram.



### **Complete Subtype Button**

Inserts a complete subtype relationship when you click on a parent then a child entity.

### **Text Block Button**

Inserts a text block when you click on the diagram.

### **Attribute Manipulation Button**

Copies or moves attributes within the same entity.

### **Identifying Relationship Button**

Inserts an identifying relationship when you click on the parent then the child entity.

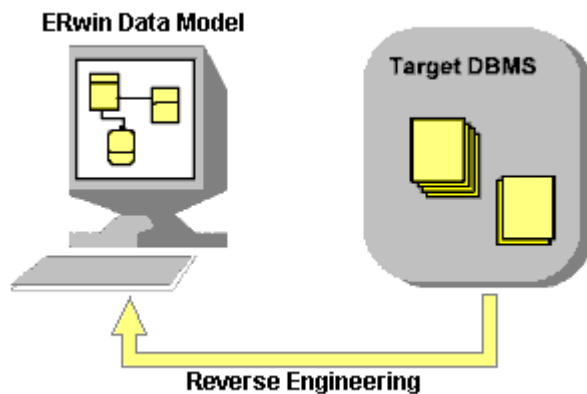
### **Many-to-Many Relationship Button**

Inserts a many-to-many relationship when you click on the first entity in the relationship, then the next entity.

### **Non-identifying Relationship Button**

Inserts a non-identifying relationship when you click on the parent then the child entity.

**Activity 3: Create an ERwin Data Model by Reverse Engineering {ewc HLP25632, HLP256\_Tile, banner.bmp}**



**What is Reverse Engineering?**

ERwin lets you quickly create a data model by **reverse engineering** an existing physical database.

During reverse engineering, ERwin first captures the information in your database or script file, including:

- Tables
- Columns
- Relationships
- Triggers
- Stored procedures
- Validation rules
- Physical storage properties

ERwin then automatically creates a physical model in your diagram based on this information.



### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Refining the Data Model

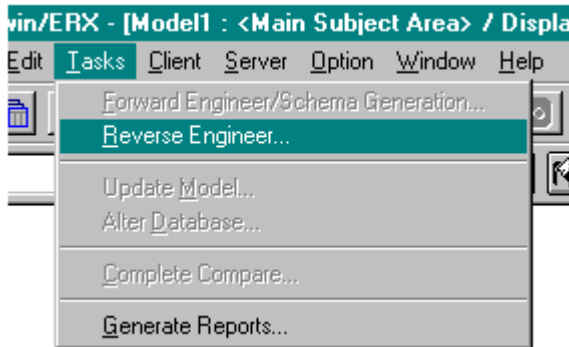
After you create a data model by reverse engineering, you can use the ERwin tools and editors to:

- Add new database objects
- Create system documentation
- Redesign the database structure based on changing requirements





### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



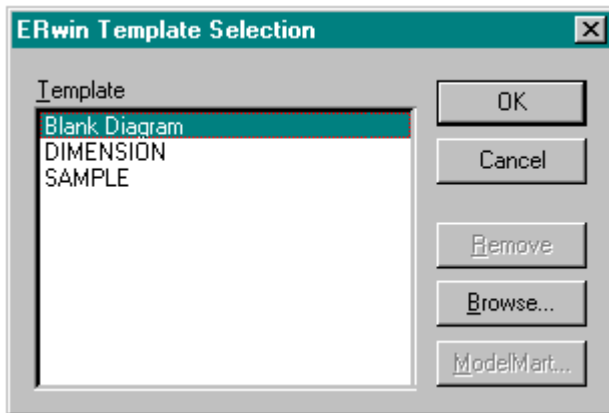
#### Task Menu - Reverse Engineer

In a few minutes, you will try this activity on a sample database. But first, let's take a quick look at the main menus and dialogs you will use.

To reverse engineer, you will choose **Reverse Engineer** on the Tasks menu. If the options on this menu are not available (dimmed), you will need to switch to the Physical Model using the Logical Model/Physical Model option on the toolbar.



### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Choosing a Template

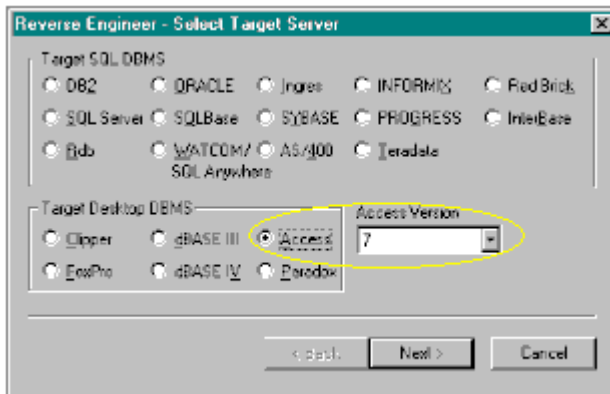
Next, you will select a template.

A template stores diagram settings such as background color, foreign-key color, shadow offset, and ERwin objects that are automatically applied to each diagram based on the template.

Templates provide the ideal way of maintaining a consistent look and feel across models.



### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the picture for a closer view.

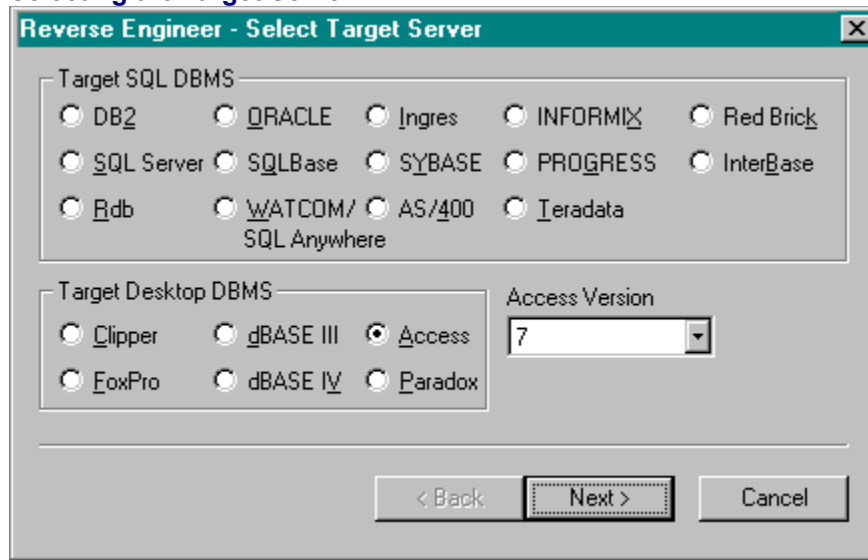
#### Selecting the Target Server

Now, you are ready to select the target server that contains the database that you want to reverse engineer. You will also choose the version in the Version list box, if applicable.

ERwin reads and interprets the information in your target server during reverse engineering.



### Selecting the Target Server



The dialog box is titled "Reverse Engineer - Select Target Server" and features a close button (X) in the top right corner. It is divided into two main sections: "Target SQL DBMS" and "Target Desktop DBMS".

**Target SQL DBMS**

This section contains a grid of radio buttons for selecting a SQL database system:

- ☐ DB2
- ☐ ORACLE
- ☐ Ingres
- ☐ INFORMIX
- ☐ Red Brick
- ☐ SQL Server
- ☐ SQLBase
- ☐ SYBASE
- ☐ PROGRESS
- ☐ InterBase
- ☐ Rdb
- ☐ WATCOM/SQL Anywhere
- ☐ AS/400
- ☐ Ieradata

**Target Desktop DBMS**

This section contains a grid of radio buttons for selecting a desktop database system:

- ☐ Clipper
- ☐ dBASE III
- ☒ Access
- ☐ FoxPro
- ☐ dBASE IV
- ☐ Paradox

**Access Version**

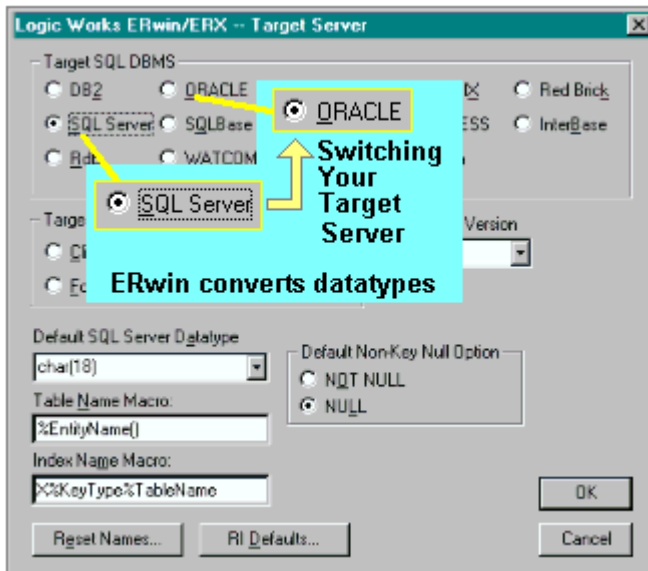
To the right of the desktop DBMS options is a label "Access Version" followed by a dropdown menu currently showing the value "7".

**Navigation Buttons**

At the bottom of the dialog are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a thick border.



### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



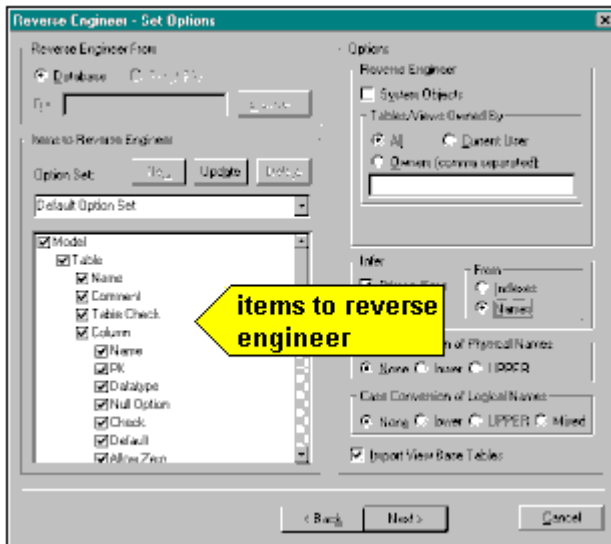
#### Switching the Target Server

When you are modeling, there may be times when you want to change the target DBMS for the active model from one server to a different one. For example, you may want to port an existing database to a different DBMS.

You can use ERwin to perform most of the conversion for you simply by reverse engineering the database and then *changing* the target server.



### Activity 3: Reverse Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the picture for a closer view.

#### Setting Options for Reverse Engineering

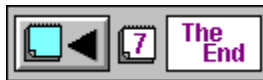
The final step in reverse engineering is:

- Choosing the database objects and properties that you want to import
- Setting case conversion options
- Deciding whether or not you want to infer primary keys or relationships from indexes

This list of items will vary depending on the database you are reverse engineering.

It's that easy! And, when you are done reverse engineering, ERwin asks if you want to use the automatic layout feature or if you want to arrange the diagram objects yourself.

To reverse engineer a sample database, click the **Try It** button below.





## Setting Options for Reverse Engineering

**Reverse Engineer - Set Options**

Reverse Engineer From  
☒ Database ☐ Script File  
File:

Items to Reverse Engineer  
Option Set:     
Default Option Set:

☒ Model  
    ☒ Table  
        ☒ Name  
        ☒ Comment  
        ☒ Table Check  
        ☒ Column  
            ☒ Name  
            ☒ PK  
            ☒ Datatype  
            ☒ Null Option  
            ☒ Check  
            ☒ Unknown  
            ☒ Unknown

Options  
Reverse Engineer  
☐ System Objects  
Tables/Views Owned By  
☒ All ☐ Current User  
☐ Owners (comma separated):

Infer  
☒ Primary Keys ☐ Indexes  
☒ Relations ☒ Names

Case Conversion of Physical Names  
☒ None ☐ lower ☐ UPPER

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

☒ Import View Base Tables

< Back    Next >    Cancel

## Reverse Engineering Try It

During this exercise you will create a data model by reverse engineering a sample database.

Click **Next Step >** to continue.



## STEP 1: Select the Reverse Engineer task

On the **Tasks** menu, choose **Reverse Engineer**.

*The Template Selection dialog is displayed.*

Click **Next Step >**





## STEP 2: Select a template

Select **Blank Diagram** on the **Template Selection** dialog.

*For more information about this dialog box, press F1.*

In the dialog, click **Next >**.

Click **Next Step >**





### STEP 3: Select the target server

Check that the target server -  
**Access** is selected.

Check that 97 is typed in the **Access  
Version** text box.

*For more information about this  
dialog, press F1.*

Click **Next >**.

Click **Next Step >**



#### STEP 4: Set options

Press **Next >** to accept the default options.

Click **Next Step >**





## STEP 5: Connect to the database

In the **Access Connection** box,  
check that `admin` is the username.

A password is not required.

Click on the Database box.

Click the **Browse** button and select  
path**reverse.mdb**


Click **Connect**.

Click **Next Step >**



## STEP 6: Examine the model

Initially the tables may appear empty. But, text appears as you zoom into the model. That's because ERwin is attempting to show the entire model and reduced the model to the lowest level of magnification.


Click on the zoom icon  until the model comes into focus.

Click **Next Step** >





## **STEP 7: Close the New Model**

Click  below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

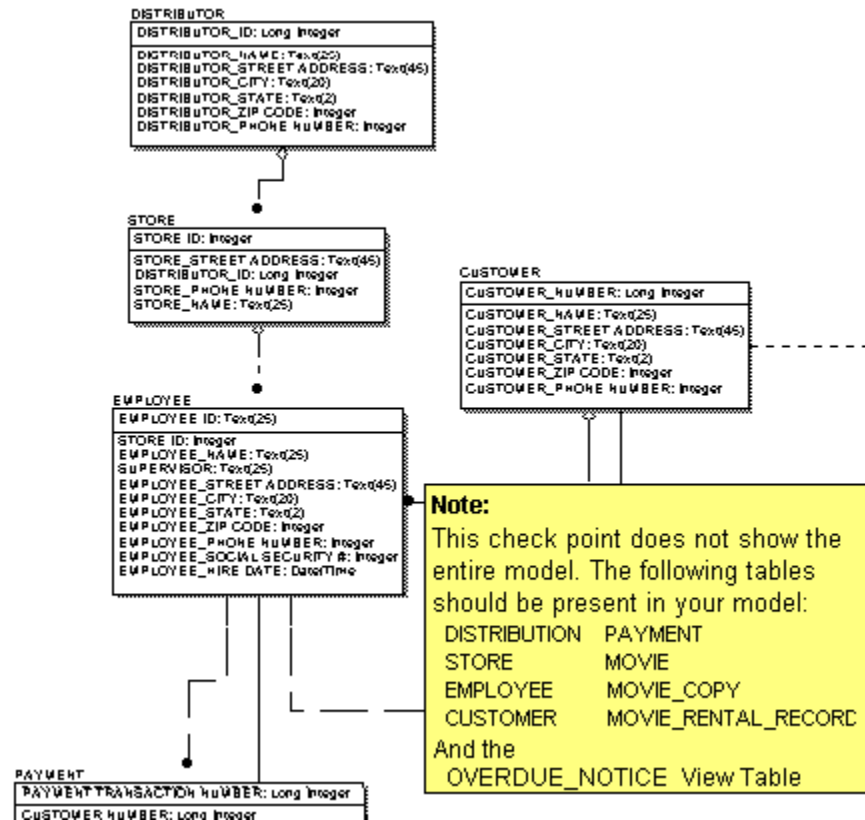
Click below to close this Try It card.

**- The End -**

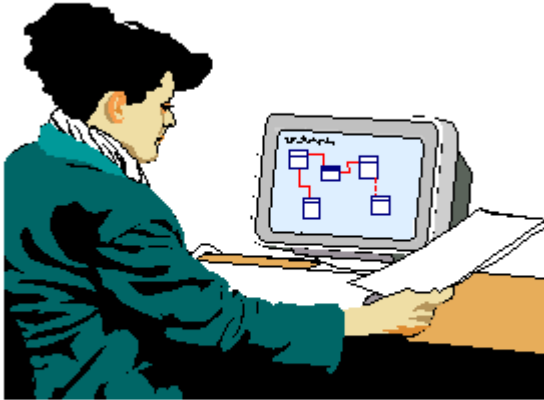




## Reverse Engineering Check {button Close,CW("check")}





## About This Tutorial {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Welcome to the ERwin Tutorial

This ERwin Tutorial is designed to help you become familiar with ERwin, the award winning data modeling tool. By showing you how to create a data model for a sample database, you can quickly learn how to use ERwin for real database design and schema generation.

Click  below to go to the next page.  
To return to the tutorial menu, click .



About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



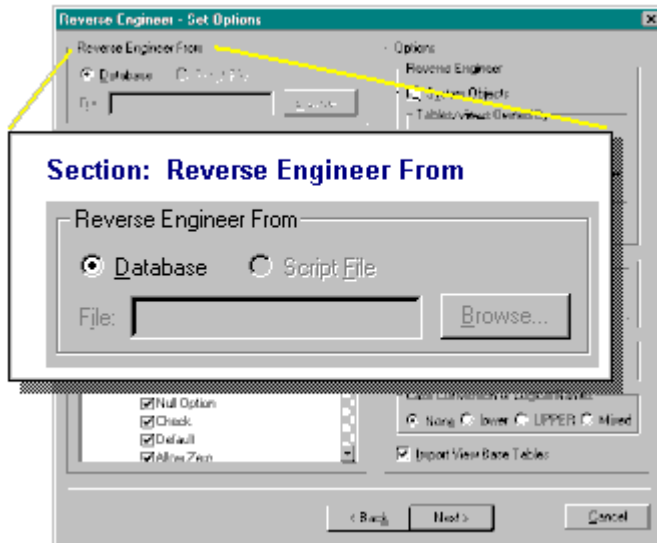
### Tutorial Format

This tutorial is organized by typical ERwin modeling activities. Each activity builds upon the knowledge learned in the previous activity, so we recommend you complete them in the presented sequence.





## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Activity Overview

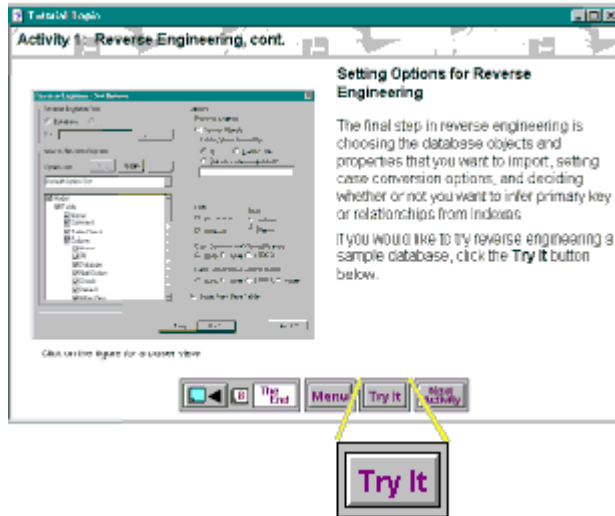
For those of you who are new to data modeling, each activity begins with an introduction covering important concepts and terminology.

Each overview also includes some of the menus and dialogs you will see while performing these activities.

To get a closer look at some of these screens, you can simply click on the area to see a zoomed-in view.



## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

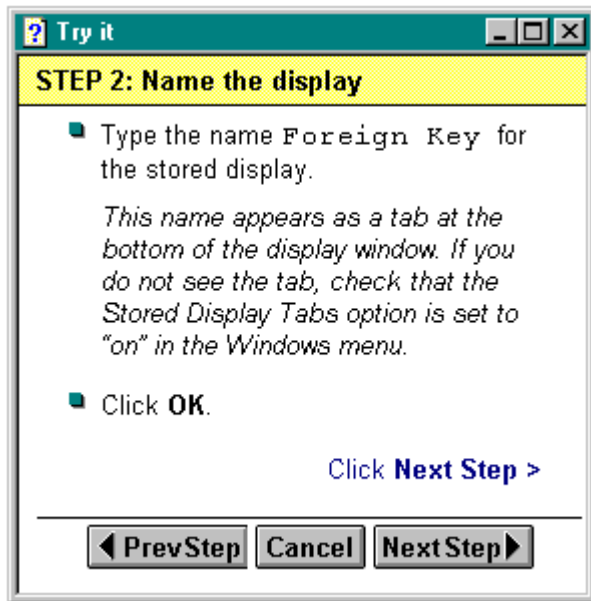


### Try It

Once you have reviewed the overview topics, you are ready to try the activity on the sample data model provided with this tutorial. This sample model represents a database that could be used to manage information for a video rental store.



## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Cue Cards

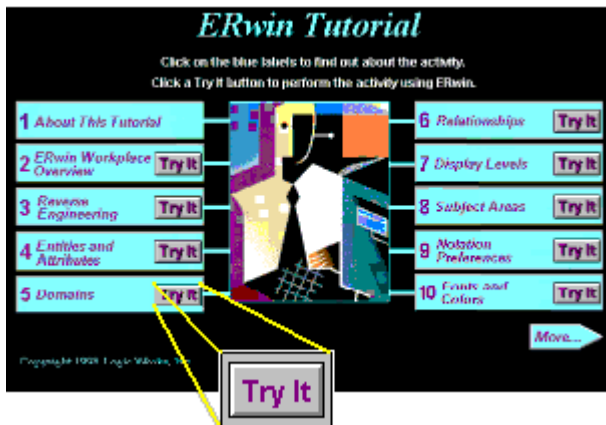
When you click Try It, the tutorial presents step-by-step cue cards to guide you through completing an activity using the sample model.

To help you quickly perform these steps, these cards use some special typographical conventions:

- Text that you type is displayed in a Courier font. For example:  
Type the name **Foreign Key** for...
- System responses and special instructions are displayed in italics. For example:  
*This name appears as a tab...*
- Buttons, menus, and options you choose or click are bolded. For example:  
Click **OK**.



## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



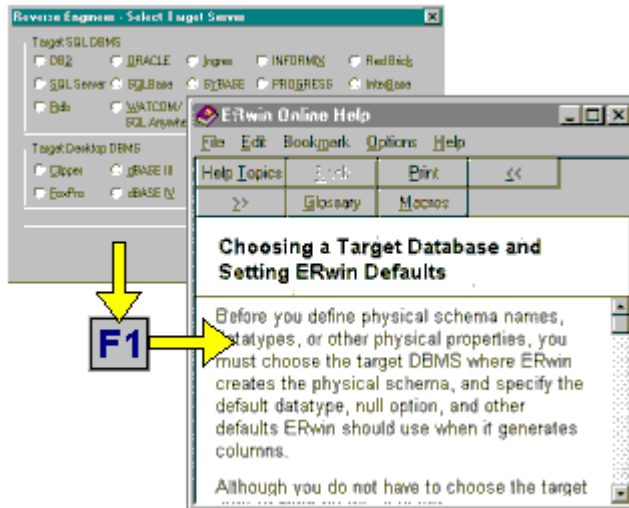
### Experienced Users

If you are already familiar with data modeling, you may want to go directly to the practice exercises (Try It) from the Main Tutorial Menu.





## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

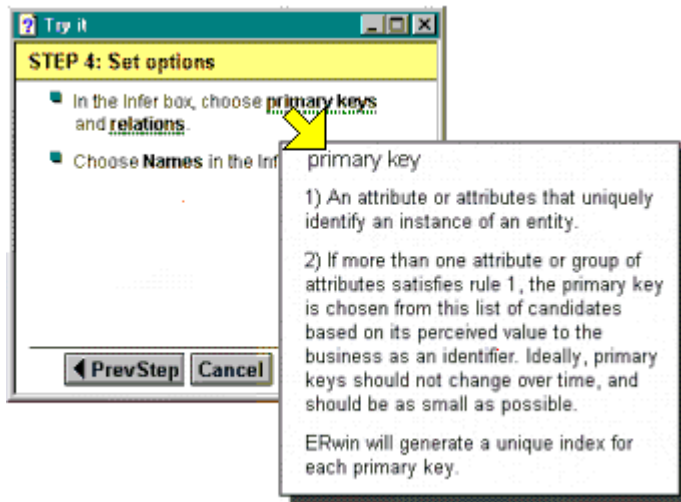


### Getting More Information

If you need more detailed information during a step, you can press F1 to view context-sensitive help for a particular dialog. This context-sensitive help is also available while designing your own data model.



## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



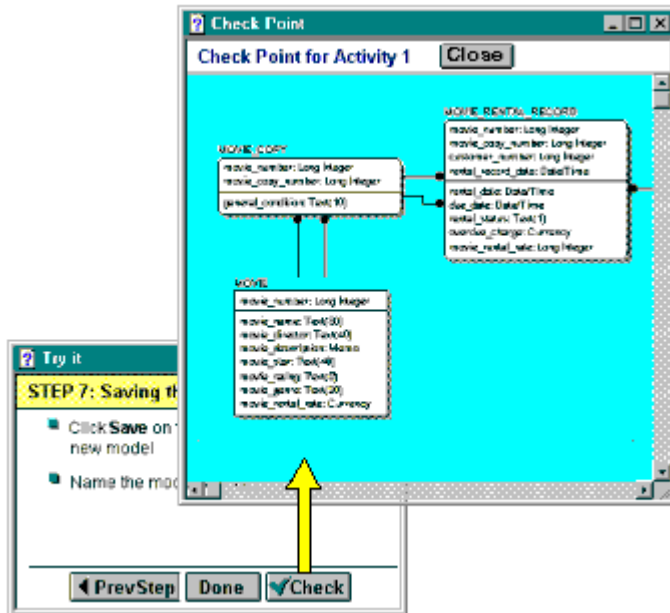
### Glossary Terms

You may, from time to time, come across a term that is unfamiliar to you. Don't worry. If the word is underlined, simply click on it to get a brief definition.

For example, clicking on the term *primary keys* in the figure to the left, displays the term in a popup window.



## About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Checking Your Work

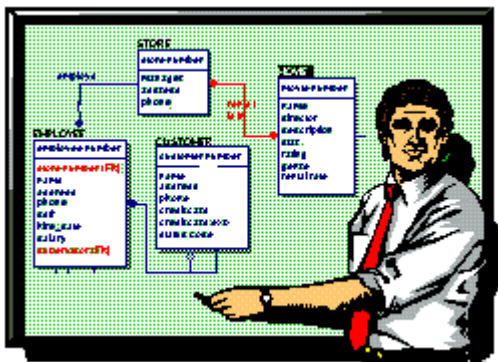
At the end of each Try It, you can verify that you have performed each activity correctly by simply clicking on the **Check** button. A Check Point sample is displayed to show you how your data model should look if you followed the tutorial steps exactly.

To return to the Try It cue card, simply click on the **Close** button.

To return to the Activity Overview, simply click on the **Done** button on the cue card.



**About This Tutorial, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



## In-depth Training

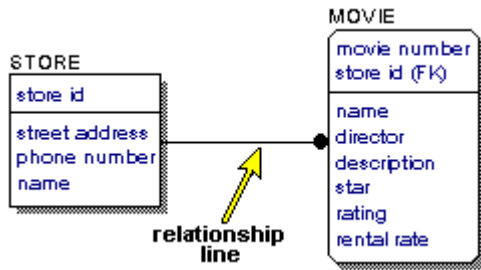
In addition to this tutorial, Logic Works provides a complete curriculum of classroom training to assist you in becoming a proficient ERwin user.

Check our web site **www.logicworks.com**  
(Education/Consulting section) for an up-to-  
date workshop schedule.





## Activity 6: Creating a Relationship {ewc HLP25632, HLP256\_Tile, banner.bmp}



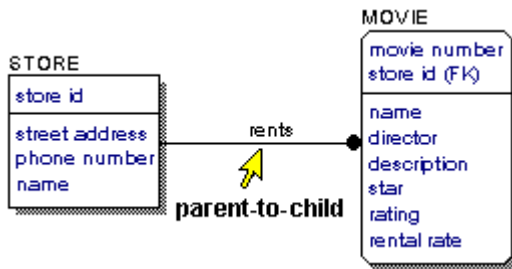
### Entity Relationships

In an entity-relationship diagram, a **relationship** shows an association between two entities. ERwin represents a relationship as a line connecting two entities. Depending on the notation you choose, the symbols at either end of the line may change.

The diagram to your left shows the relationship between the STORE entity and the MOVIE entity using Integration DEFINition (IDEF1X) notation.



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Verb Phrases

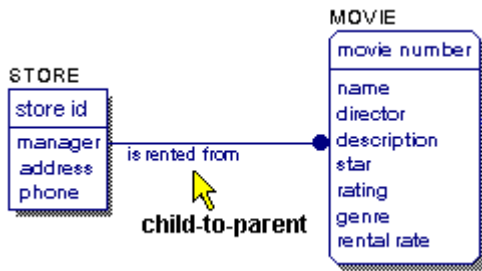
A **verb phrase** describes how two entities are related. In ERwin, you can create two types of verb phrases: parent- to-child and child-to-parent.

A **parent-to-child** verb phrase describes how the parent is related to the child. This relationship is designated by a dot at the end of the relationship line in Integration DEfinition (IDEF1X) notation.

For example, in the example to the left, the verb phrase <rents> describes the relationship between the parent entity STORE and the child entity MOVIE. You can read the relationship as “a store <rents> one or more MOVIES.”



**Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



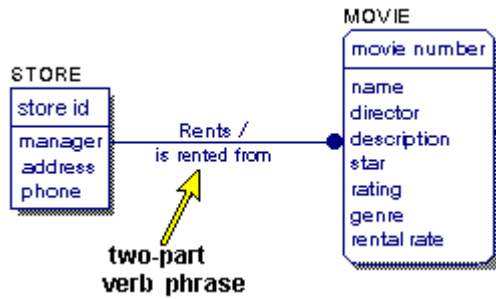
**Child-to-Parent Verb Phrase**

A **child-to-parent** verb phrase describes how a child entity is related to a parent entity.

In the example on the left, the verb phrase <is rented from> describes the relationship between the child entity MOVIE and the parent entity STORE. You read this relationship as "A MOVIE <is rented from> a STORE."



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Two-Part Verb Phrase

If you enter both a parent-to-child and child-to-parent verb phrase, you have created a **two-part verb phrase**.

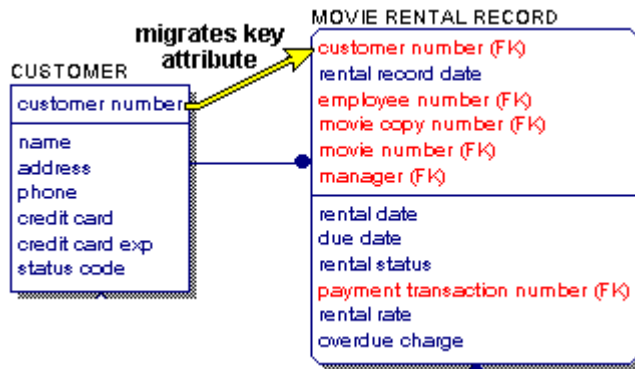
When you view the relationships that you create, ERwin displays a two-part verb phrase near the middle of the relationship line with the two phrases joined by a forward slash symbol (e.g. Rents/is rented from). You read the relationship to your left as "a STORE rents a MOVIE."

In order to see the verb phrase, the Display Option called "Display Verb Phrase" must be "on". We will discuss Display Options in more detail in Activity 7.





## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### What is a Foreign Key?

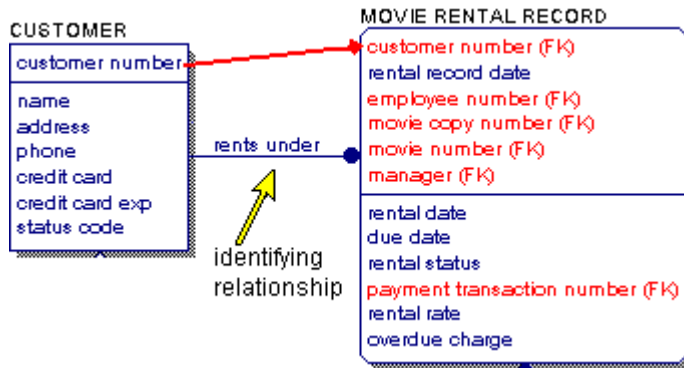
When you create a relationship between entities, ERwin automatically migrates the primary key attribute(s) of the parent entity, to the child entity.

The name of the migrated key attribute, also known as a **foreign key**, is indicated by the designator **(FK)**. This migration occurs automatically in the physical model.

In order to see the foreign keys in the logical model, the Display Options called "Show Migrated Keys" and "FK Designator" must be "on." We will discuss Display Options in more detail in Activity 7.



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Two Types of Relationship

You can draw two types of relationships: **Identifying** and **Non-Identifying**.

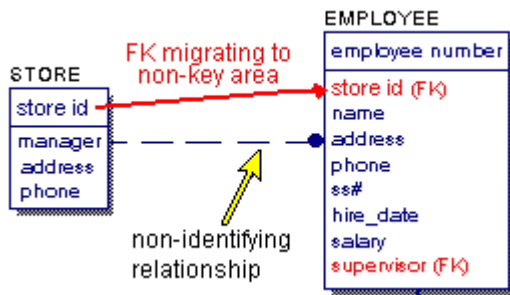
In an **identifying relationship**, the FK migrates above the line and becomes part of the primary key of the child entity.

In ERwin, a solid line represents an identifying relationship.

For example, CUSTOMER "rents under" MOVIE RENTAL RECORD. In this relationship, you need to know which customer rented the movie in order to uniquely identify MOVIE RENTAL RECORD.



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



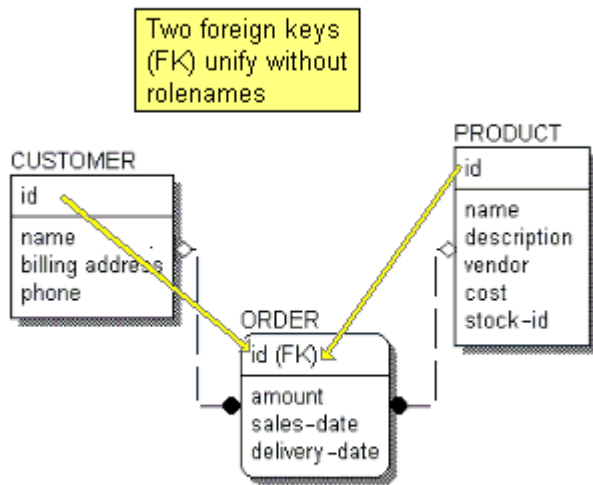
### Non-Identifying Relationship

In a **non-identifying relationship**, the foreign key migrates below the line and becomes a non-key attribute in the child entity. In ERwin, a dashed line represents a non-identifying relationship.

For example, the **STORE** "employs" **EMPLOYEE**. In this relationship, an employee can be uniquely identified *without* knowing the store number.



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Unification

Earlier you read that when you create a relationship, the primary key attribute of the parent entity migrates to the child entity as a foreign key attribute. If the foreign key attribute has the same name as an owned attribute in the child entity, ERwin automatically unifies the two instances into one attribute because it assumes that they are the same attribute. The process of combining or unifying identical attributes in an entity is called **unification**.

In the example on the left, the CUSTOMER, PRODUCT, and ORDER entities have a primary key called "id." Both CUSTOMER and PRODUCT entity have a non-identifying relationship with the ORDER entity. When the CUSTOMER primary key "id" migrates to the ORDER entity, ERwin unifies the two instances into one foreign key attribute. Similarly, it unifies the "id" that migrates from the PRODUCT entity.





## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

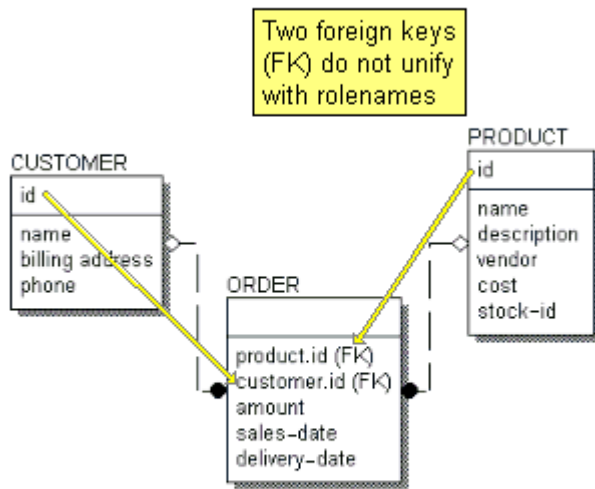
### Using Rolenames

Sometimes, unification is not the desired behavior. You may want to maintain a distinction between multiple attributes with the same name in an entity because each serves a different purpose.

For example, in order to track references to customer and product for each order, you need to include all of the “id’s” in the ORDER entity as separate attributes.

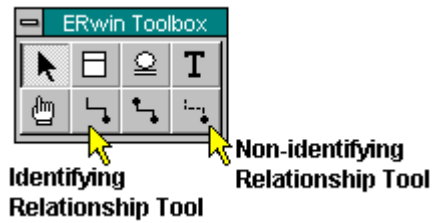
In this situation, you can assign a **rolename**, which is basically an alias that you assign to a foreign key. In the example to the left, both customer.id and product.id in the ORDER entity represents foreign keys that have been assigned a rolename.

A rolename has two names separated by a period. The first name is the alias (the rolename you assign) and the second is the basename (the name of the primary key in the parent entity).





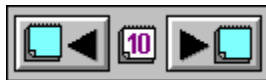
## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Relationship Tools

In a few minutes, you will create an identifying relationship using the sample model. But first, let's take a quick look at the tools and dialog you will see.

To create a relationship, you will start by selecting either the Identifying or Non-Identifying Relationship tool from the ERwin toolbox.



## Activity 6: Creating a Relationship, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

Relationship Editor

Relationship: STORE employs EMPLOYEE

**Rolename**

New... Delete

General Definition Rolename/RI Actions UDP

Verb Phrase

Parent-to-Child: employs **Parent-to-child verb phrase**

Child-to-Parent: works at **Child-to-parent verb phrase**

Relationship Cardinality

Summary: One-to-One-or-More (P)

Cardinality

☐ Zero, One or More

☒ One or More (P)

☐ Zero or One

☐ Exactly:

Relationship Type

☐ Identifying ☐ Non-Identifying

☐ Nulls ☐ Nulls Allowed ☒ No Nulls

☐ Logical Only

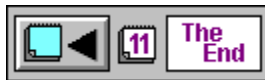
OK Cancel

### Relationship Editor

Once you have created a relationship, you can add the verb phrases and assign a [rolename](#) using the **Relationship Editor**.

To create a relationship, click the **Try It** button below.

Click the figure for a closer view.





## The Relationship Editor

Relationship: STORE employs EMPLOYEE

New... Delete

General | Definition | Rolename/RI Actions | UDP

Verb Phrase

Parent-to-Child: employs

Child-to-Parent: works at

Relationship Cardinality

Summary: One-to-One-or-More (P)

Cardinality

☐ Zero, One or More

☒ One or More (P)

☐ Zero or One (Z)

☐ Exactly:

Relationship Type

☐ Identifying

☒ Non-Identifying

Nulls

☐ Nulls Allowed

☒ No Nulls

☐ Logical Only

OK Cancel

## **Rolename**

Rolenames are helpful when distinguishing between migrated keys that have the same name as existing attributes. Without rolenames, the attributes unify and will be generated as a single column in your database.

## Relationships Try It

In this exercise, you will create a non-identifying relationship with a verb phrase as well as review the rolename feature.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\06-tutor.er1](#)  
from the **File|Open...** menu.



Click **OK** on the “read-only” message box.


Check that **Logical** model is displayed in the option list on the ERwin toolbar.

You may need to rearrange the  
objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Select relationship tool

In the ERwin toolbox, click the **Non-Identifying Relationship** tool .

*The cursor changes shape to identify the tool you selected.*

Click **Next Step** >





**STEP 2: Draw the relationship line between the two entities**

Click the **STORE** entity (parent).

Click the **EMPLOYEE** entity (child).

*Because this is a non-identifying relationship, ERwin draws a dashed line.*

Click **Next Step >**







### STEP 3: Display the Relationship Editor

Click the **Select** tool in the ERwin Toolbox.

Click the relationship line just created.

Position cursor directly over the selection bar (small black square) and click the right mouse button.

*ERwin displays the relationship shortcut menu.*

On the shortcut menu, choose **Relationship Editor**.

Click **Next Step >**





#### STEP 4: Add verb phrases

In the **General** Tab, replace the default text with `employs` in the Parent-to-Child Phrase box.

Click **OK**.

*ERwin closes the Relationship Editor and adds the verb phrase to the relationship line.*

If you do not see the verb phrase you need to turn on the verb phrase display option. To set this option:

Right-click on a blank part of the diagram, then choose **Display Options/Relationships** then **Verb Phrases** from the short-cut menu.  
[Click Next Step >](#)







## STEP 5: Look at rolename for employee ID

Double-click on the recursive relationship line for **Employees**.  
(HINT: relationship line has the verb phrase “manages”)

*ERwin opens the Relationship Editor.*

Click the **Rolename/RI Actions** tab.

Notice that supervisor has been  
assigned as the rolename for  
employee ID.

Click **Cancel**.

To see the rolename in the entity, you must turn on the rolename display option.

Click **Next Step >**



## STEP 6: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

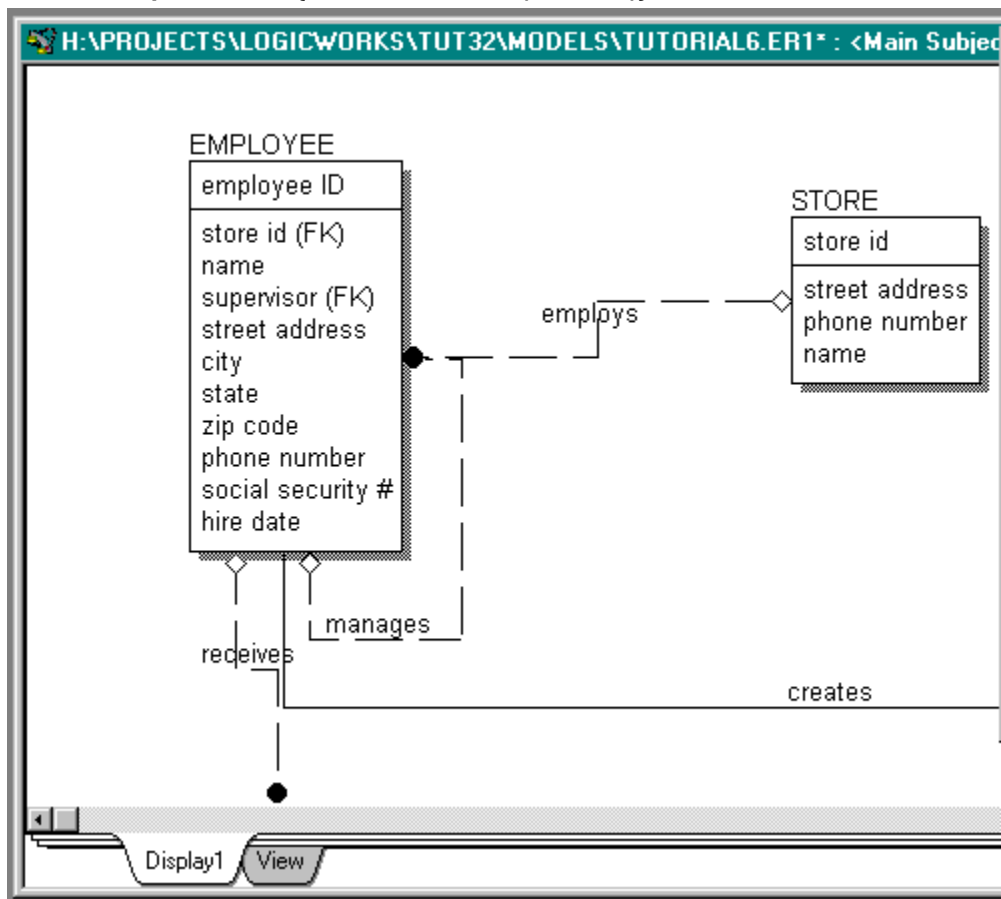
**- The End -**



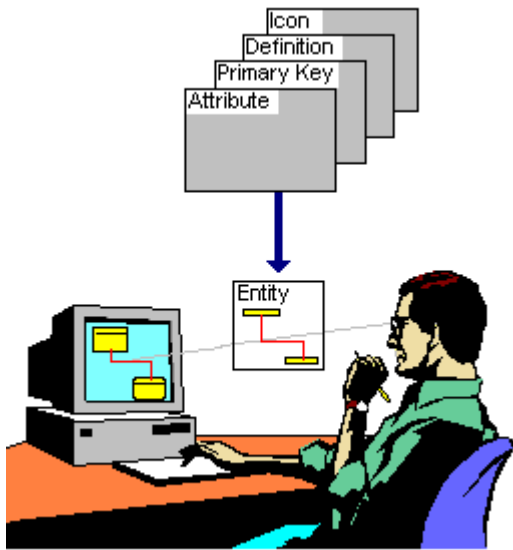




Relationships Check {button Close,CW("check")}



## Activity 7: Displaying Information in Your Data Model {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Display Levels for the Logical Model

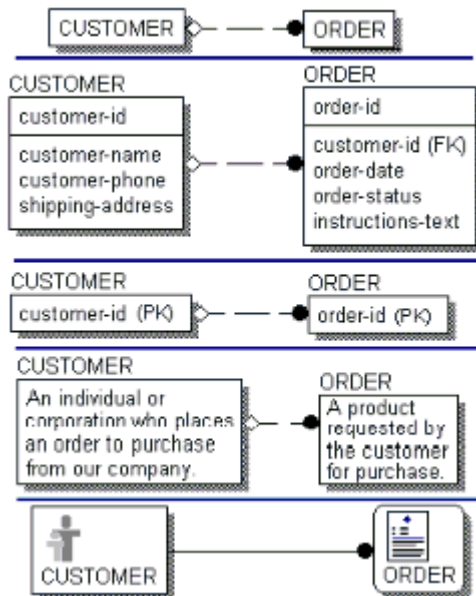
ERwin provides you with five different display levels to view the logical information about a data model. These levels are useful for showing different types of information in the logical model. The five levels are:

- Entity
- Attribute
- Primary Key
- Definition
- Icon

ERwin provides two sets of display levels, one for the logical model and one for the physical model. In this activity, we will focus on the logical model.



## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Display Level Examples

- [Entity level](#)
- [Attribute level](#)
- [Primary Key level](#)
- [Definition level](#)
- [Icon level](#)



### **Entity level**

Displays the name of each entity in a data model inside an entity box. No other information is displayed for the entity.



### **Attribute level**

Displays the attributes for each entity in a data model, with the primary key attributes above the line and the non-key attributes below the line in an entity box.

### **Primary Key level**

Displays the primary key attributes (those found above the line in an entity box) for each entity in a data model. The attributes below the line are hidden, but display when you switch back to the Attribute Display Level and you can also see them in the Attribute Editor.

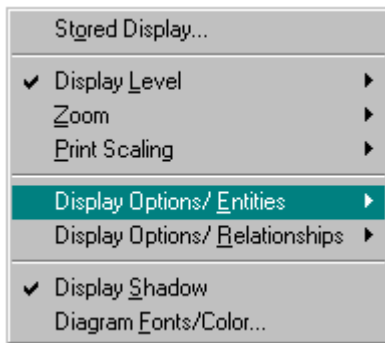
**Definition level**

Displays the definition for each entity in a data model.

### **Icon level**

If you have assigned a large bitmap to an entity, the bitmap is displayed when you choose Icon Level. You can assign a different bitmap to each entity in the Icon tab of the Entity Editor.

## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Display options

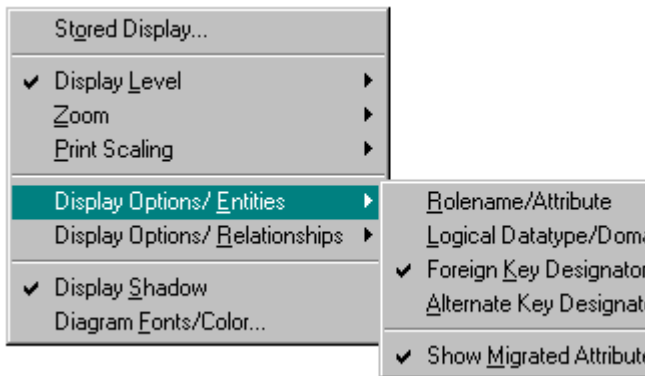
ERwin also provides you with display options for entities and their relationships. These options are useful for showing different types of information in the logical model.

- Entity display options
- Relationship display options

You can use the shortcut menu shown to the left, to set these options.



## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Entity display options

There are five display options that you can use to view the logical information about the **entities** in a data model, including:

- [Rolename/Attribute](#)
- [Logical Datatype/Domain](#)
- [Foreign Key Designator](#) (FK)
- [Alternate Key Designator](#) (AK)
- [Show Migrated Attributes](#)





### **Rolename/Attribute**

Display the basename and the rolename for the attributes that have been assigned a rolename. If you do not display this option, basenames will *not* display.

### **Logical Datatype/Domain**

Display the logical datatype or domain for the attributes in a data model.

### **Foreign Key Designator (FK)**

Display the foreign key designator (FK) after the attribute name if either the “Foreign Key Designator” is selected as a display option or if the “Show Migrated Attributes” option is selected in the Stored Display Editor.

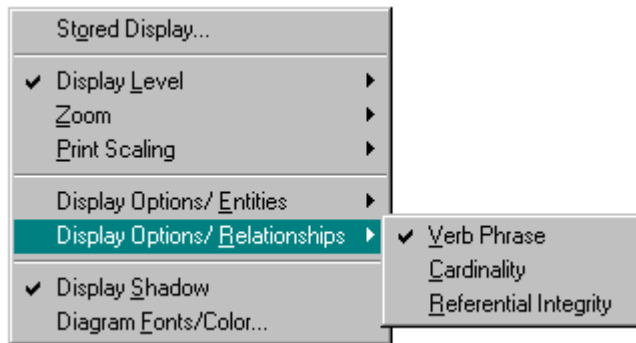
**Alternate Key Designator (AK)**

Display the alternate key (AK) designator and inversion entry (IE) designator for the attributes in an entity.

**Show Migrated Attributes**

Display the primary key attributes that have migrated to associated entities as foreign key attributes.

## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Relationship display options

There are three display options that you can use to view the logical information about the relationships in a data model, including:

- [Verb Phrase](#)
- [Cardinality](#)
- [Referential Integrity](#)



### **Verb Phrase**

Display verb phrases for the relationships in a data model.



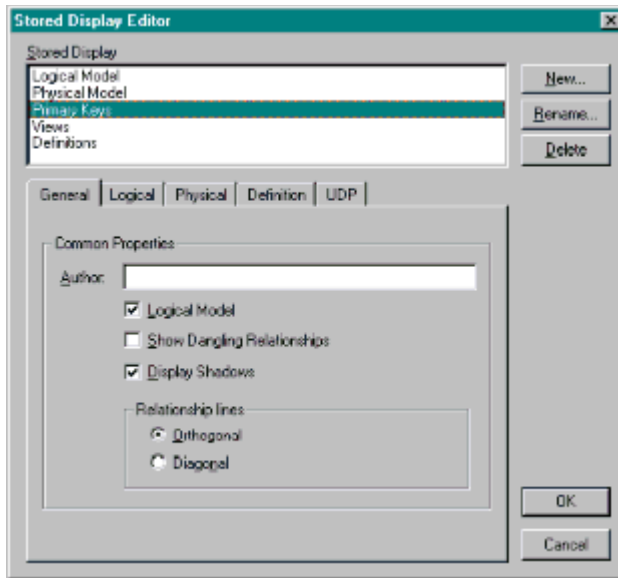
## Cardinality

Display cardinality symbols for the relationships in a data model. Cardinality shows the ratio of instances of a parent entity to a child entity. A parent entity may be related to a zero one, or more instances (or an exact number of instances of a child entity.) Each type of cardinality is represented by a symbol.

## **Referential Integrity**

Display referential integrity symbols for the relationships in a data model. Referential integrity enforces business rules that define actions that a DBMS should take when a row of data in a table is inserted, updated, or deleted.

## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Stored Displays

ERwin lets you create your own customized displays. Using the **Stored Display Editor**, you can change the display options and display levels for the current diagram. Then you can save the display settings and assign a name to the stored display.

This feature is particularly helpful when you need to look at your model for different purposes. For example, you may want to quickly check all the entities or show entities using icons for management or non-technical audiences.

Any changes you make to the graphical appearance or layout of a stored display do *not* affect any other stored display in the diagram.



## Stored Displays

The image shows a software window titled "Stored Display Editor" with a close button (X) in the top right corner. The window is divided into several sections. At the top, under the heading "Stored Display", there is a list box containing five items: "Logical Model", "Physical Model", "Primary Keys" (which is highlighted with a blue selection bar), "Views", and "Definitions". To the right of this list are three buttons: "New...", "Rename...", and "Delete". Below the list is a tabbed interface with five tabs: "General", "Logical", "Physical", "Definition", and "UDP". The "General" tab is currently selected. Inside the "General" tab, there is a section titled "Common Properties" which contains a text input field labeled "Author:". Below this are three checked checkboxes: "Logical Model", "Show Dangling Relationships", and "Display Shadows". Further down is a section titled "Relationship lines" containing two radio button options: "Orthogonal" (which is selected) and "Diagonal". At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Stored Display Editor

Stored Display

- Logical Model
- Physical Model
- Primary Keys
- Views
- Definitions

New...  
Rename...  
Delete

General Logical Physical Definition UDP

Common Properties

Author:

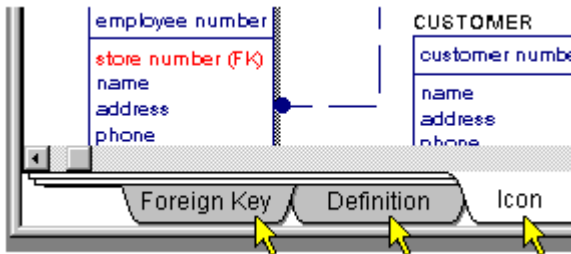
☒ Logical Model  
☐ Show Dangling Relationships  
☒ Display Shadows

Relationship lines

☒ Orthogonal  
☐ Diagonal

OK  
Cancel

## Activity 7: Displaying Information, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

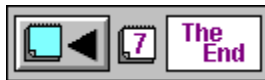


### Stored Display Tabs

By default, the first Stored Display is automatically created by ERwin and assigned the name Display1. For each stored display you create, ERwin adds a tab to the bottom of the diagram window. You must set the Stored Display Tabs option to "on" in the Window menu to see these tabs.

When you save a data model, ERwin saves all stored displays that are associated with the data model in an .er1 file, which is ERwin's native file format.

To create a stored display, click the **Try It** below.





## Displaying Information Try It

In this exercise, you will practice creating a new stored display.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\07-tutor.er1](#)  
from the **File|Open...** menu.



Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Open the New Store Display Dialog

Right-click on a blank area of the diagram, then choose **Stored Display...** from the shortcut menu.

Click the **New...** button to open the New Stored Display dialog.

Click **Next Step >** to continue.





## STEP 2: Name the display

Type the name `Foreign Key` for the stored display.

Click **OK**.

Click **Next Step >**







### STEP 3: Choose a display level

Click the **Logical** Tab.

Select **Attribute** level.

*For more information about this Editor, position cursor in the editor and press F1.*

Click **Next Step >**





#### STEP 4: Choose entity options

Click the **Foreign Key Designator** checkbox on.

Click the **Show Migrated Attributes** checkbox on.

Click **OK**.

*A tab labeled Foreign Key appears at the bottom of the display window. If you do not see the tab, check that the Stored Display Tabs option is set to "on" in the Windows menu.*

Click **Next Step >**





## STEP 5: Close the Model

Click below to verify that you have performed the activity correctly, then

click .



On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

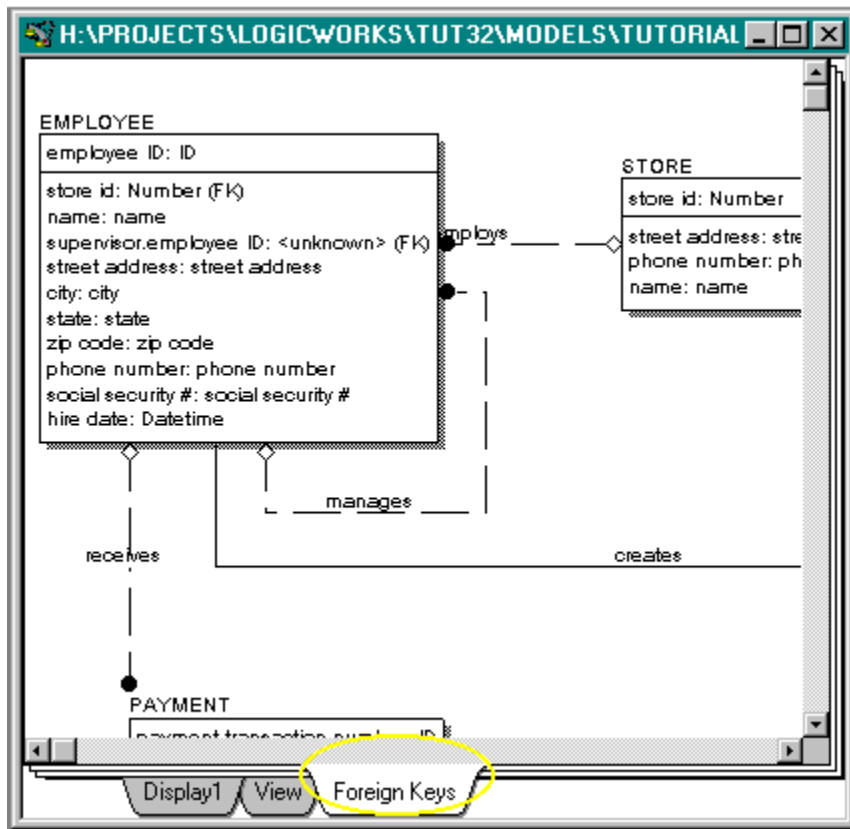
Click below to close this Try It card.

**- The End -**

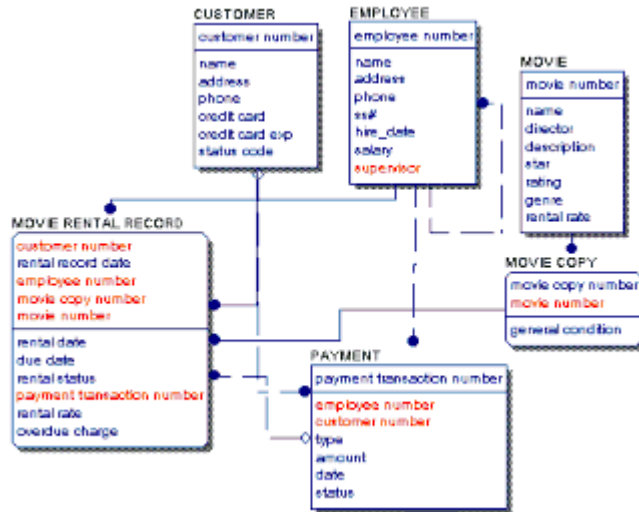




Displaying Information Check {button Close,CW("check")}



#### Activity 4: Creating Entities and Attributes {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### What is an Entity?

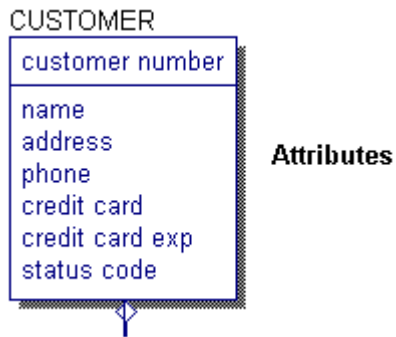
An **entity** is a person, place, or thing about which an organization maintains information. For example, in our sample model we have the following entities:

- EMPLOYEE
- CUSTOMER
- MOVIE
- MOVIE COPY
- MOVIE RENTAL RECORD
- PAYMENT

Each entity usually corresponds to a database table.



#### Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### What is an Attribute?

Once you have discovered the people, places, and events that define the entities in your model, you can begin to define the attributes for each entity.

For example, once you create the CUSTOMER entity, you can begin to define the individual pieces of information you want to track for each customer, including:

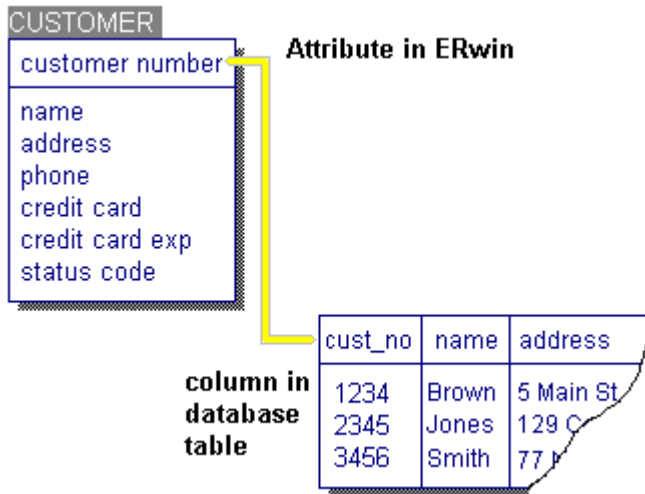
- Name
- Address
- Phone number

Each of these pieces of information is saved in ERwin as an **attribute** of the CUSTOMER entity.





#### Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

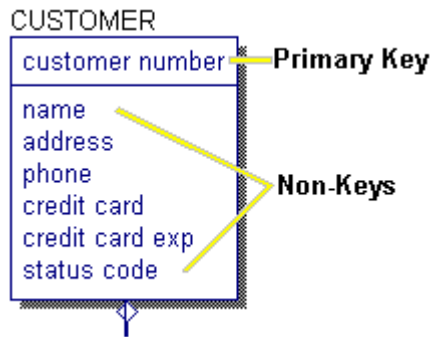


#### Attributes and Columns

In the logical model, an attribute usually corresponds to a column in a table. For example, the CUSTOMER entity contains the *customer number* attribute. The *customer number* attribute becomes the *cust\_no* column in a database.



**Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



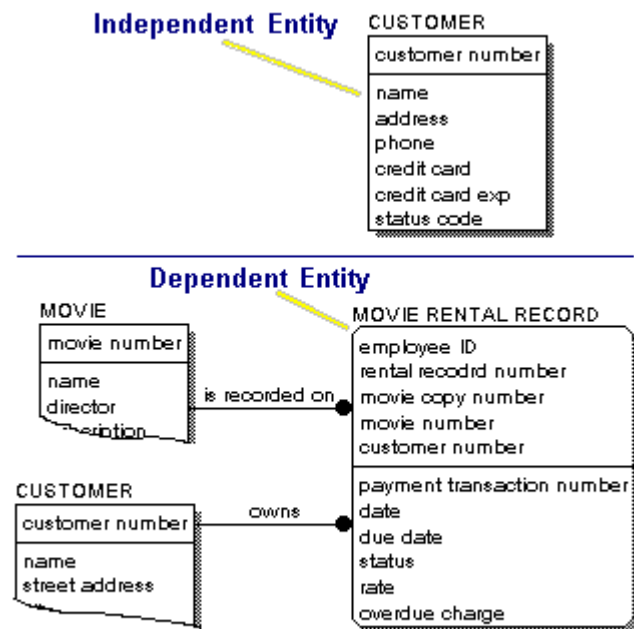
**Primary Key and Non-Key Areas**

In ERwin, an entity is drawn as a box with a horizontal line near the top of the box. In an entity box, the area above the line is called the **primary key** area and contains the primary key attributes.

The **non-key** area, which contains attributes that are *not* the primary key, is below the line.



#### Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Two Types of Entities

An ERwin logical model can contain two types of entities: **independent** and **dependent**.

An **independent entity** is an entity whose instances can be uniquely identified without determining its relationship to another entity. It is represented as a box with square corners.

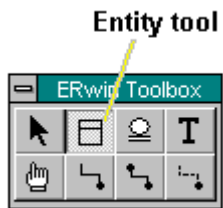
In the example to your left, the CUSTOMER entity is autonomous of any other entities in the data model.

A **dependent entity** is an entity whose instances *cannot* be uniquely identified without determining its relationship to another entity or entities. It is represented as a box with rounded corners.

For example, the MOVIE RENTAL RECORD entity depends upon the CUSTOMER and MOVIE entities for its identity, which means that its primary key attributes have migrated from two other entities-CUSTOMER and MOVIE.



**Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



**Entity Tool**

ERwin includes a single tool in the ERwin toolbox for creating both independent and dependent entities.

The type of entity is determined by the type of relationship in which it is involved.

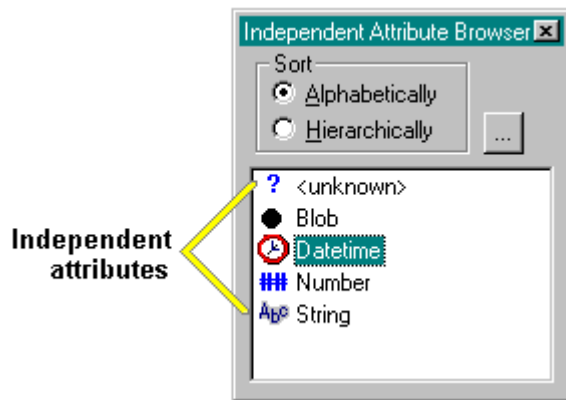




### **Relationship definition**

A relationship is an association between two entities and is represented by a solid or dotted line. We will discuss relationships in more detail in Activity 6.

**Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



### Adding Attributes

To speed construction of your model and to promote consistency, ERwin provides **independent attributes**, a shortcut method for adding attributes. Rather than adding an attribute and assigning properties in a 2-step process each time you add an attribute, ERwin provides predefined, “reusable” attributes that work like a template.

Using a drag-and-drop operation from the Independent Attribute browser, you can add an attribute by simply selecting from the Independent Attribute list and dropping it on the entity.

This list includes five independent attributes that are available with every model. You can also add your own user-defined independent attributes.

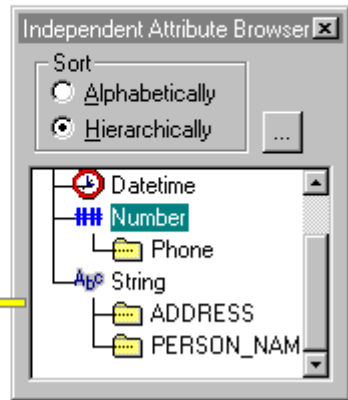


#### Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

##### MOVIE



independent  
attribute  
properties  
are inherited



#### Inheriting Properties

The new attribute inherits all the properties of the selected independent attribute. You will have the option to change the name and modify the attribute properties after the drag and drop operation.

You can also add your own user-defined independent attributes. We will discuss this in more detail in the next activity called Defining Domains.



#### Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

Select Tool



Attribute Manipulation Tool



#### Moving Entities and Attributes

ERwin includes two tools in the ERwin toolbox for moving entities and attributes.

The *Select* tool is used to move the entire entity.

The *Attribute Manipulation tool* is used to move an attribute within and between entities.





**Activity 4: Creating Entities and Attributes, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**

**On-Diagram Editing**

While you are still in the design phase, the names of entities and attributes may change frequently. ERwin lets you use **on-diagram editing** to name objects directly *without* opening an editor.

The edit box, shown on the left, is where you will type the object name.

If you would like to try adding a logical entity on the sample model, click the **Try It** button below.





## Entities and Attributes Try It

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\04-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “[read-only](#)” message box.


Check that **Logical** model is displayed in the option list on the ERwin toolbar.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Select the entity tool

In the ERwin toolbox, click the **Entity** tool .

The Entity tool creates an independent entity with square corners.

[Click \*\*Next Step\*\* >](#)







### identifying relationship

In an **identifying relationship**, the parent entity's primary attributes FK migrates above the line and becomes part of the primary key of the child entity. In ERwin, a solid line represents an identifying relationship.

## STEP 2: Draw the new entity


Click anywhere on the diagram.

*ERwin draws the new entity and labels it E/n, where E stands for entity and n is a unique number.*

Click **Next Step >**



**STEP 3: Type the name of the entity**

Click the Select tool  in the toolbox.

Click on the new entity.

When an edit box appears, type  
VIDEO CENTER.

Press SHIFT+ENTER to close the  
edit window for the entity name.

Click **Next Step >**







#### STEP 4: Add primary key attribute

Select **number** from the list displayed in the Logical Independent Attribute Browser dialog.

**Tip:** If you do *not* see the Independent Attribute Browser, click on the diagram then press CTRL+B to display it.

Drag and drop the **number**

independent attribute into the primary key area of the new VIDEO CENTER entity.

Click **Next Step >**



## STEP 5: Add non-key attributes

Drag and drop **street address** into the Non-Key area.

*Notice that a horizontal line appears to indicate where the attribute will be placed.*

Drag and drop **phone number** into the Non-Key area.

Click **Next Step** >



## STEP 6: Change the entity name

Select the **VIDEO CENTER** entity using the **Select** tool.

Click the name VIDEO CENTER, wait briefly, then click again.

Use the BACKSPACE or DELETE key  
to erase the name.

Type the new name **STORE**.  
**Tip:** You can press the ESC (Escape)  
key to cancel a change.  
[Click \*\*Next Step\*\* >](#)







## STEP 7: Change the attribute name

Select the **STORE** entity using the **Select** tool.

Tab to the **Number** attribute.

Use the DELETE or BACKSPACE key  
to erase the old name.

Type `store-id`.

Press SHIFT + ENTER to save your changes.

**Tip:** You can press the ESC (Escape) key to cancel a change.

Click **Next Step >**



## STEP 8: Move the entity

Click the **STORE** entity and drag it to the left, then to the right.

Click **Next Step >**





## STEP 9: Move the attribute

Click the **Attribute Manipulation** tool



Click the attribute **Street Address** in the STORE entity and drag it so that it is the last attribute in the list.

Click **Next Step >**





## STEP 10: Close the model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

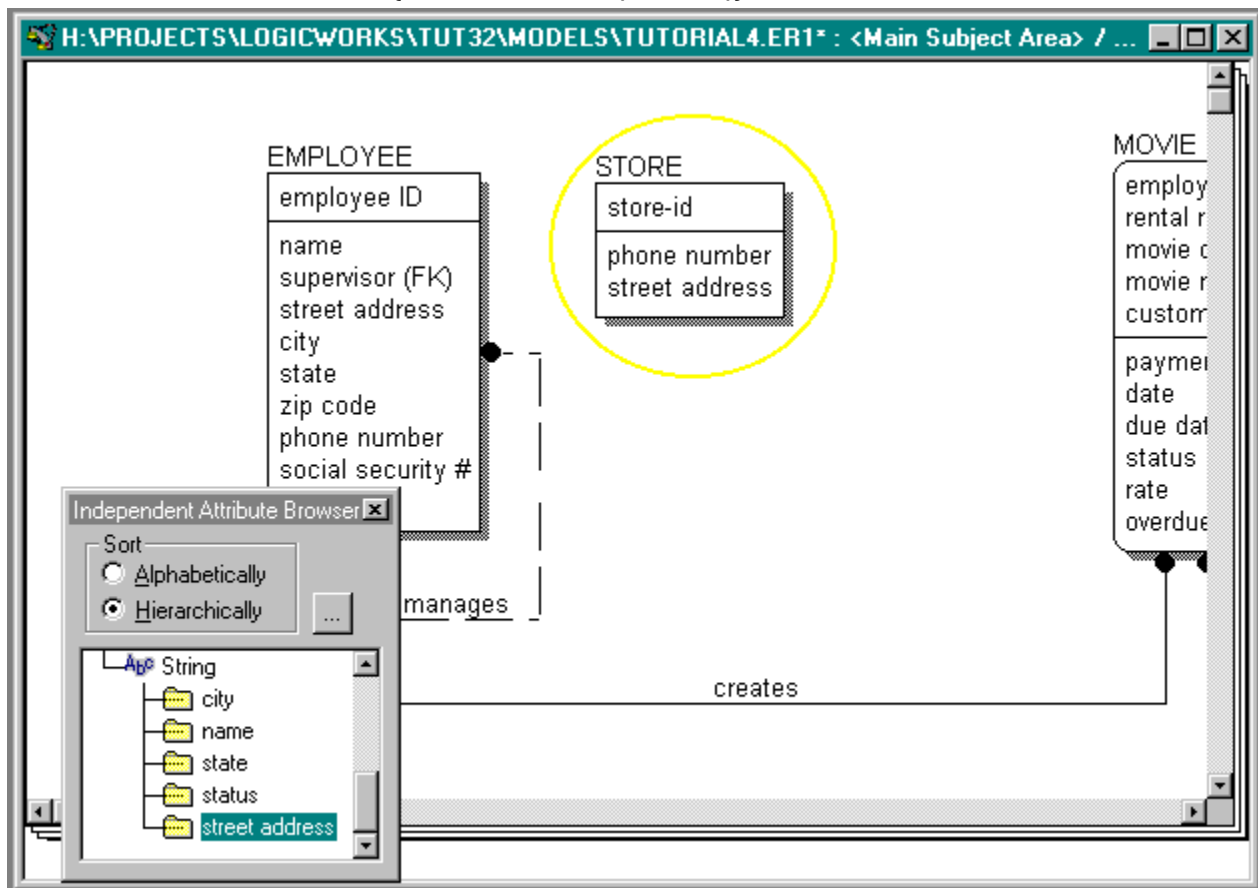
**- The End -**



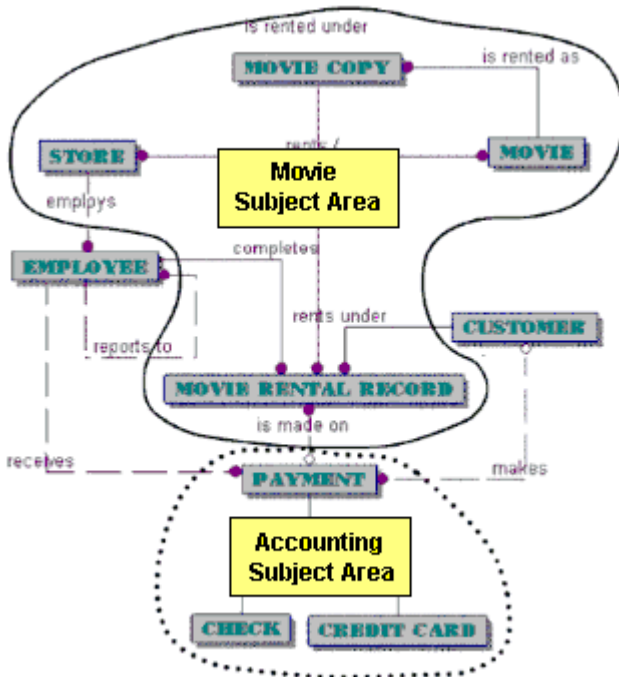




Entities and Attributes Check {button Close,CW("check")}



## Activity 8: Creating Subject Areas {ewc HLP25632, HLP256\_Tile, banner.bmp}



### What is a Subject Area?

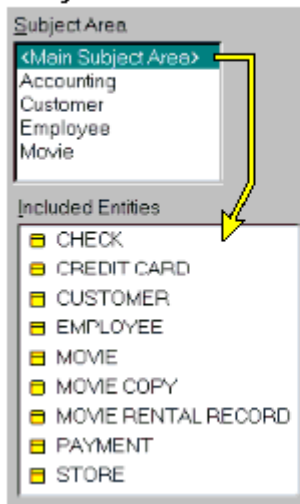
A **subject area** is a subset of objects taken from the whole pool of objects in your diagram. You can create multiple subject areas in your diagram. Typically, you create a subject area to:

- Help you manage a large diagram
- Reduce the number of objects that you work with
- Focus on a particular business function

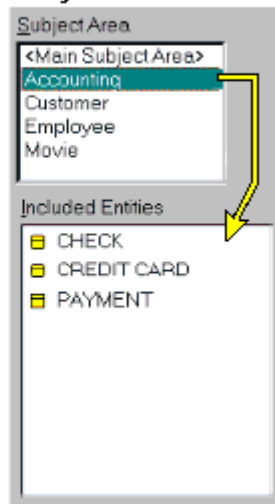


## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

### Main Subject Area



### Accounting Subject Area



### Main Subject Area

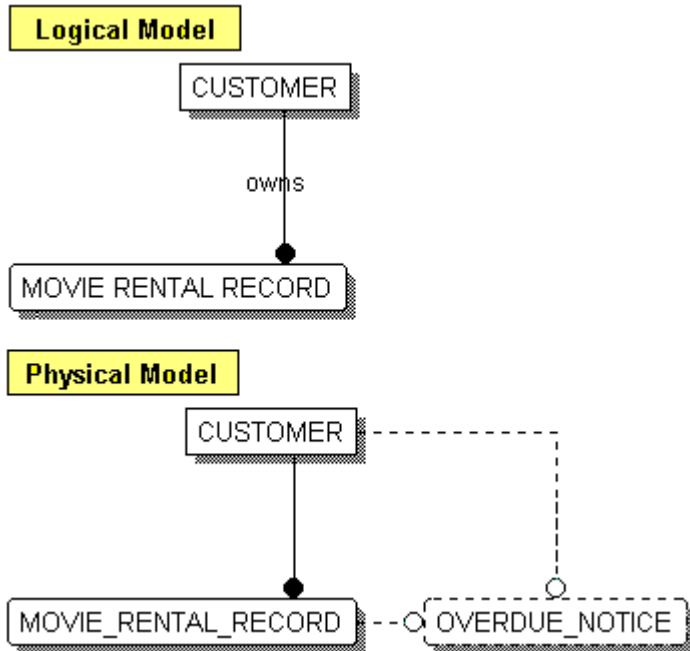
By default, a new data model includes one subject area called the **Main Subject Area**. It includes all the objects (entities, tables, views and text blocks in a data model).

ERwin lets you create one or more specialized subject areas that show specific business functions or tasks. Typically, objects in a specialized subject area relate to a specific business unit such as finance, marketing, or manufacturing.

In the example to your left, the **Main Subject Area** includes entities and relationships for all facets of the business. In addition there are four subject areas, called **Accounting**, **Customer**, **Employee**, and **Movie**, which include only those objects that relate to those specific functions of the video store business.



## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Logical and Physical Subject Areas

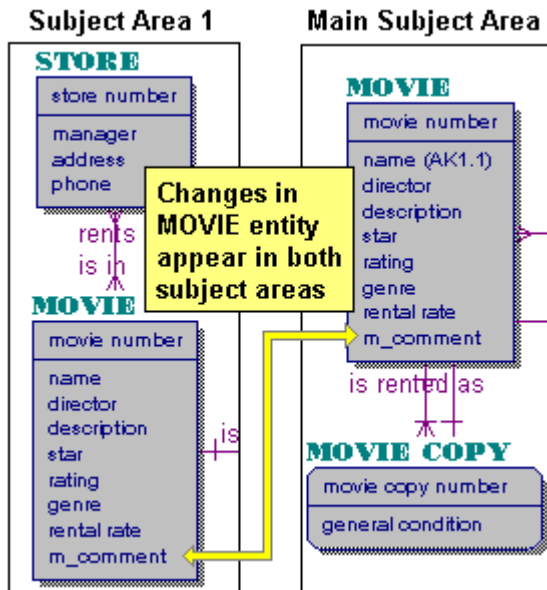
When you create a subject area for a logical model, ERwin automatically creates a corresponding subject area for the physical model and vice-versa.

If you include or remove members from a subject area, the membership change is reflected in the corresponding physical or logical model, but will *not* affect the membership of any other subject area.

However, if you delete or change an object in a subject area, the change ripples through *all* subject areas that contain that object.



## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Subject Areas are NOT copies...

It is important to understand that the subject areas are *not* copies of the data model, but are dynamic subsets of the data model.

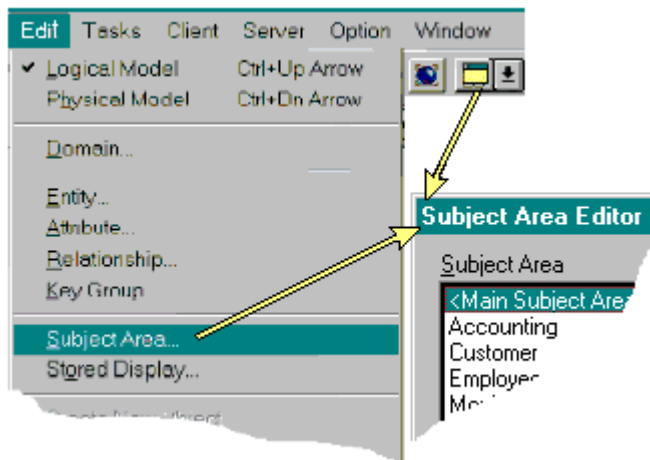
In other words, if you add objects in a subject area, such as an entity and a relationship, those objects are added to the current subject area and the Main Subject Area.

If you add an attribute (or column) to an existing entity, it is added to the entity (or table) in every subject area in which it is a member.






## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



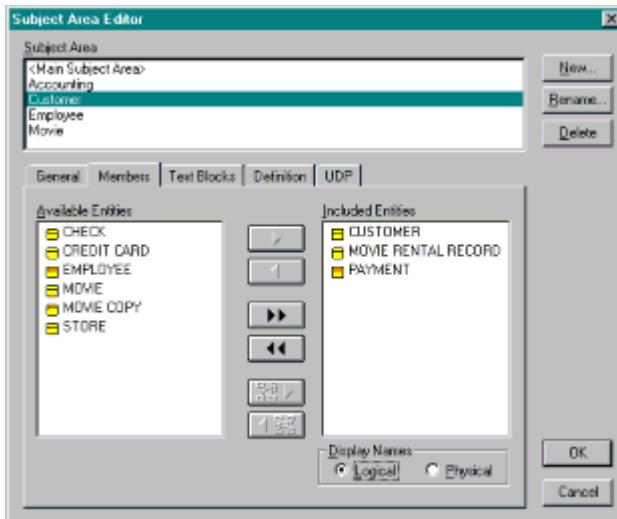
### Edit menu - Subject Area

Soon, you will create a subject area for the sample model, but first let's take a look at the menus and dialogs you will use.

To create a subject area you will select Subject Area from the Edit menu or click the  button on the toolbar.



## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Subject Area Editor

Using the Subject Area Editor you can:

- create, modify, or delete subject areas in the ERwin diagram
- view all of the objects that are included in a subject area in the model

Adding objects or members to a subject area is done by simply moving them from one list to the other in the Members tab.



## Subject Area Editor

Subject Area Editor

Subject Area

<Main Subject Area>

Accounting

Customer

Employee

Movie

New...

Rename...

Delete

General

Members

Text Blocks

Definition

UDP

Available Entities

CHECK

CREDIT CARD

EMPLOYEE

MOVIE

MOVIE COPY

STORE

↓

↑

▶▶

◀◀

⌵

⌶

Included Entities

CUSTOMER

MOVIE RENTAL RECORD

PAYMENT

Display Names

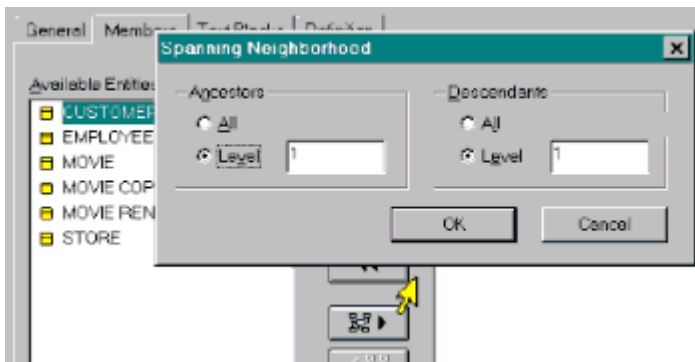
☒ Logical

☐ Physical


OK

Cancel

## Activity 8: Creating Subject Areas, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Spanning Neighborhoods

A quick way to add members to a subject area is to click on the  button to open the **Spanning Neighborhood** dialog.

In this dialog, you can specify the level of ancestors and descendants you want to automatically include in the subject area. Ancestors are the parents and grandparents of an entity. Descendants are the children and grandchildren of an entity.

To create a subject area, click on the **Try It** button below.







### Creating Subject Areas Try It

In this exercise you will create a new subject area.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\08-tutor.er1](#)  
from the **File|Open...** menu.


Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Open the Subject Area dialog

Click the  on the ERwin toolbar or choose **Subject Area** from the Edit menu.

Check that **<Main Subject Area>** is highlighted.

Click **Next Step >**





## STEP 2: Name the new subject area

Click the **New...** button.

Enter the name **Movie Inventory** in the New dialog.

Click **OK**.

Click **Next Step >**





### STEP 3: Add general information

In the **General** tab, type your name in the Author text box.

Click **Next Step >**



#### STEP 4: Select the members

Click the **Members** tab to specify the objects (entities, tables or views) in the subject area.


Check that **Logical** is selected in the Display Names group box.


Click **Next Step >**





**STEP 5: Move the members to the Included list**

Click  to move all objects to the Available list.

Select **MOVIE** from the Available list, and click the  button to move it to the Included list.

Repeat for **MOVIE COPY**, **MOVIE RENTAL RECORD**, and **STORE**.

Click **OK**.

*ERwin automatically switches to the new subject area when you close the Subject Area Editor.*

Click **Next Step >**







## Step 6: Selecting a Subject Area

Click the  on the toolbar.

Select **Main Subject Area** from the Subject Area list.

Select **Movie Inventory** from the  
Subject Area list.

Click **Next Step >**



## **STEP 7: Close the model**

Click below to verify that you have performed the activity correctly, then

click

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

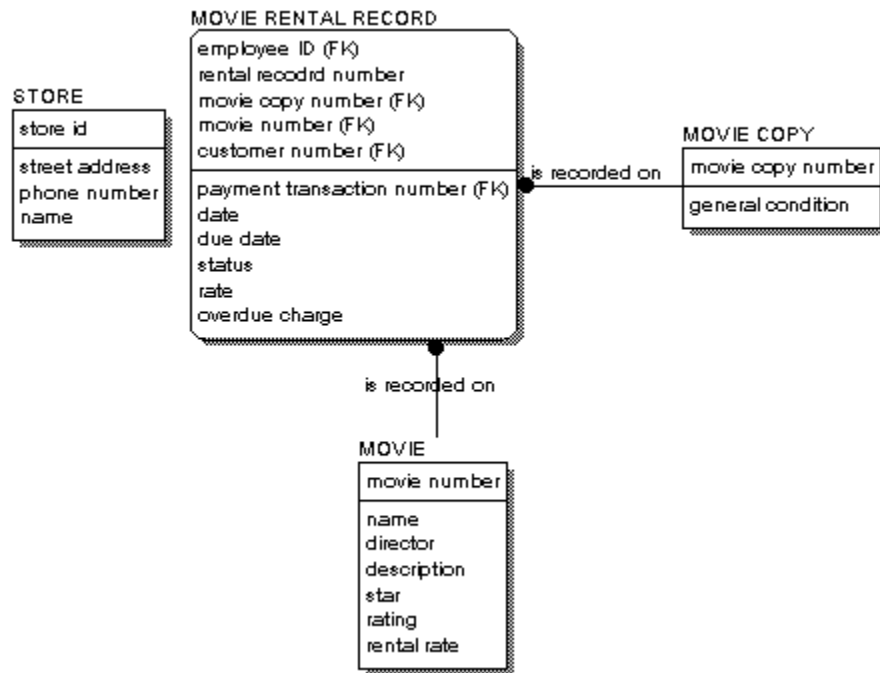
**- The End -**







Creating Subject Areas Check {button Close,CW("check")}



## Activity 9: Changing the Notation Preferences {ewc HLP25632, HLP256\_Tile, banner.bmp}

### Logical

#### IDEF1X



#### IE



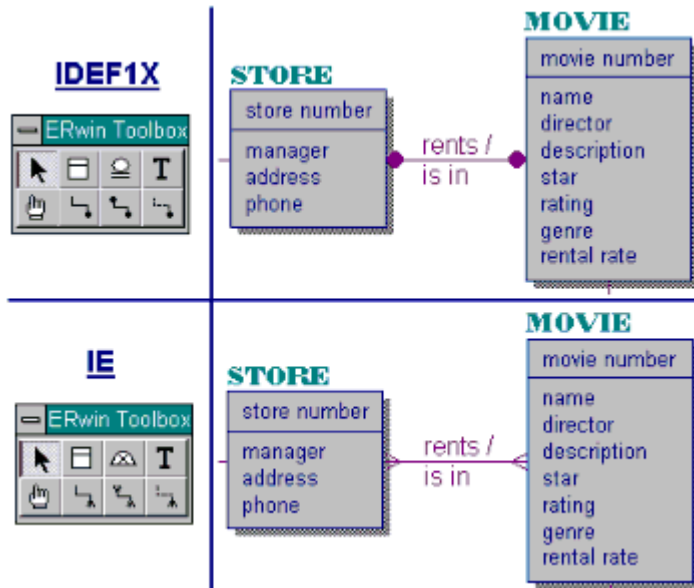
### Notation Preferences

When you work in the logical or the physical model, you can choose whether ERwin uses **Integration DEFinition** for Information Modeling (IDEF1X) notation or **Information Engineering** (IE) notation.

The default notation that ERwin uses for both the logical and physical model is IDEF1X, but you can switch to IE notation at any time in either model.



**Activity 9: Changing the Notation Preferences, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



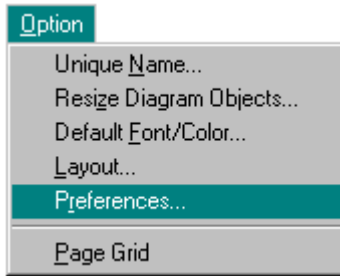
**Different Symbols**

The IDEF1X and IE notations use different symbols to represent:

- Entity and table relationships
- Cardinality, which identifies how each instance of a parent relates to a child entity or table



**Activity 9: Changing the Notation Preferences, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



**Options Menu - Preferences**

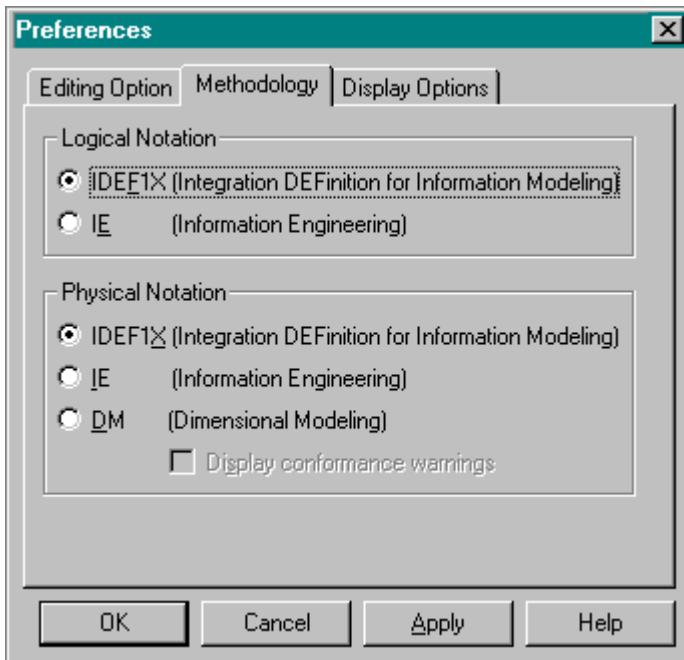
Soon, you will try changing the notation. But first, let's look at the menus and dialog boxes you will use for this task.

To change the notation, select **P**references from the **O**ption menu.





**Activity 9: Changing the Notation Preferences, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



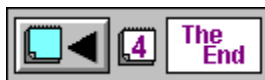
### Methodology Tab

The **Methodology** tab in the Preference dialog includes options that let you choose whether ERwin uses IDEF1X or IE notation for the logical or physical model.

You may want to choose a different notation to help distinguish the logical and physical models.

NOTE: Another option is available for the Dimensional model.

To change modeling notation preferences, click on the **Try It** button below.





### Notation Preferences Try It

In this exercise you will change notations.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\09-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

Check that **Logical** model is displayed in the option list on the ERwin toolbar.

You may need to rearrange the  
objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Open the Preference dialog

Choose **Preferences...** on the Option menu.

Click the **Methodology** tab.

Click **Next Step >**







## STEP 2: Change the notation

Select **IE** in the Logical Notation box.

Select **IE** in the Physical Notation box, if it is not already selected.

Click **Apply** to save the preferences  
before moving to a different tab.

Click **OK**.

Click **Next Step >**





### **STEP 3: Close the model**

Click below to verify that you have performed the activity correctly, then

click

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.



To close without saving, select **Close without saving** then click **OK**.

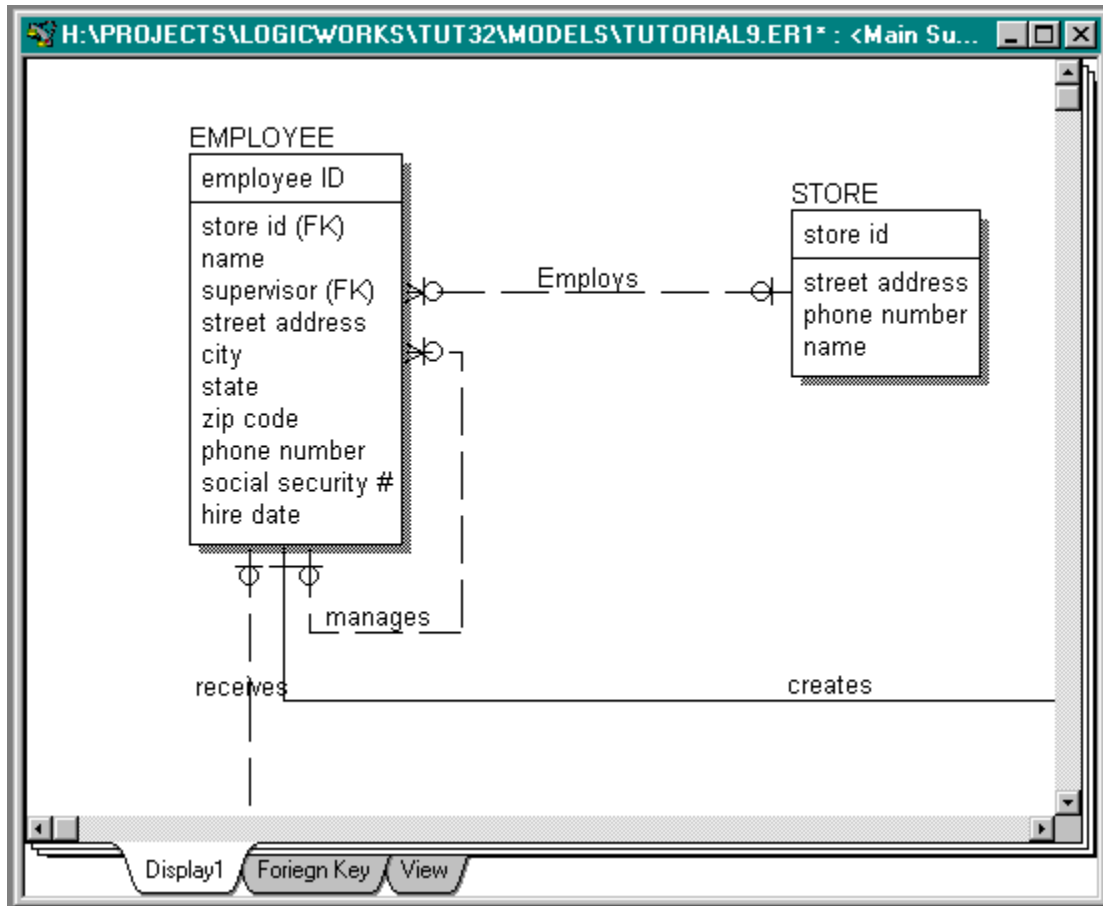
Click below to close this Try It card.

**- The End -**





Notation Preferences Check {button Close,CW("check")}



## Activity 10: Changing Fonts and Colors {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Changing Fonts and Colors

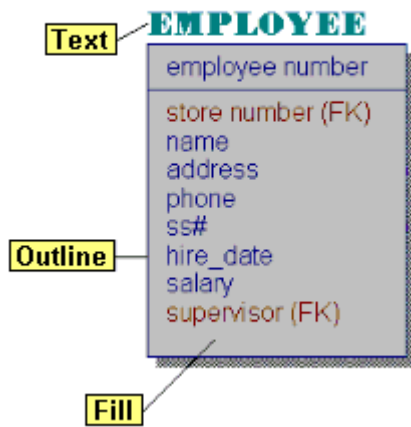
When you design an ERwin diagram, you can use ERwin's font and color features to enhance its appearance and make it easier to view and understand.

For example, if you want to prepare a diagram for analysis and presentation, you might want to:

- Enlarge the font size for all entity names so that they are easier to read.
- Change the color of all foreign keys to red, so that they are distinguishable from other attributes or columns.
- Use different fonts and/or colors to identify new objects from old objects in your model.



## Activity 10: Changing Fonts and Colors, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Three Types of Graphic Features

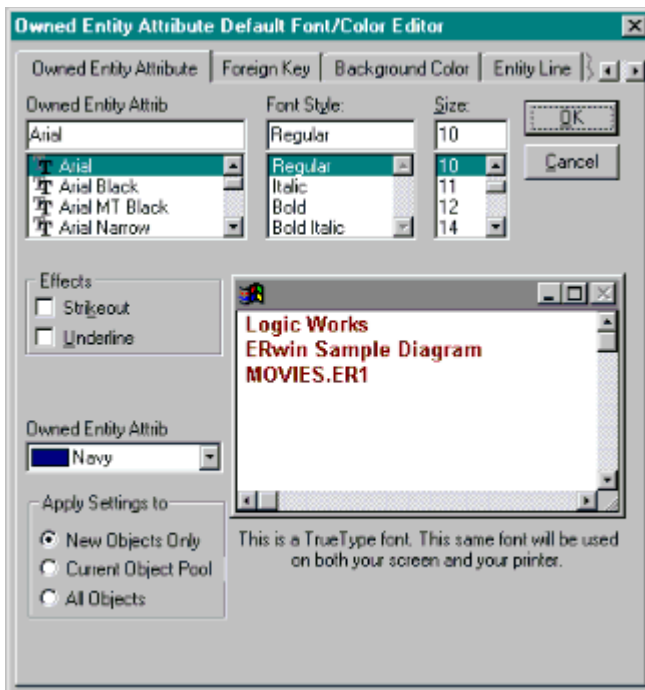
An ERwin diagram consists of three types of graphic features: **text**, **fill**, and **outlines**. You can enhance these features by changing fonts and colors.

In addition to these graphic elements, there are many object types you can globally change such as Foreign Keys, Entity Names, and Owned Keys.





## Activity 10: Changing Fonts and Colors, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Default Fonts/Colors

When you add an object to a diagram, ERwin automatically assigns it the default font and color for that object type. If you want to customize the colors, you can assign a different default color scheme using the **Default Font/Color Editor**.



## Default Font/Color Editor

Owned Entity Attribute Default Font/Color Editor

Owned Entity Attribute Foreign Key Background Color Entity Line

Owned Entity Attrib Font Style: Size: OK Cancel

Arial Regular 10

Arial Black Regular 10

Arial MT Black Italic 11

Arial Narrow Bold 12

Bold Italic 14

Effects

☐ Strikeout

☐ Underline

Owned Entity Attrib

Navy

Apply Settings to

☒ New Objects Only

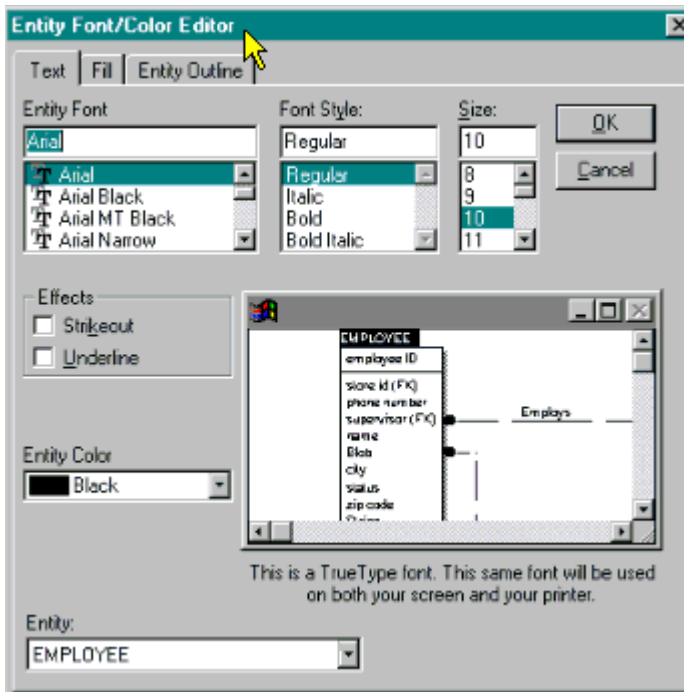
☐ Current Object Pool

☐ All Objects

Logic Works  
ERwin Sample Diagram  
MOVIES.ER1

This is a TrueType font. This same font will be used on both your screen and your printer.

## Activity 10: Changing Fonts and Colors, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Changing Individual Objects

You can also change the color or font of **individual objects** in your diagram using the <Object> Font/Color Editors or the ERwin Font and Color Toolbar.



## Changing Individual Objects

**Entity Font/Color Editor**

Text | Fill | Entity Outline

Entity Font: **Arial**

Font Style: Regular

Size: 10

OK

Cancel

Effects:

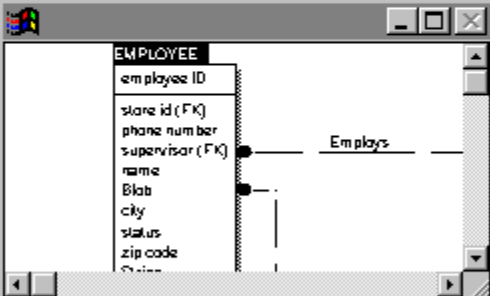
☐ Strikethrough

☐ Underline

Entity Color: Black

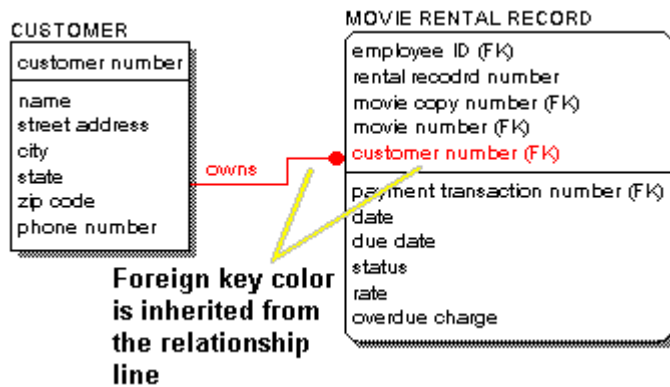
Entity: EMPLOYEE

This is a TrueType font. This same font will be used on both your screen and your printer.



The diagram shows an entity named 'EMPLOYEE' with attributes: employee ID, store id (FK), phone number, supervisor (FK), name, Blab, city, status, zip code, and a partial attribute. A relationship named 'Employs' is shown connecting the 'EMPLOYEE' entity to itself.

## Activity 10: Changing Fonts and Colors, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Objects Can Inherit Color

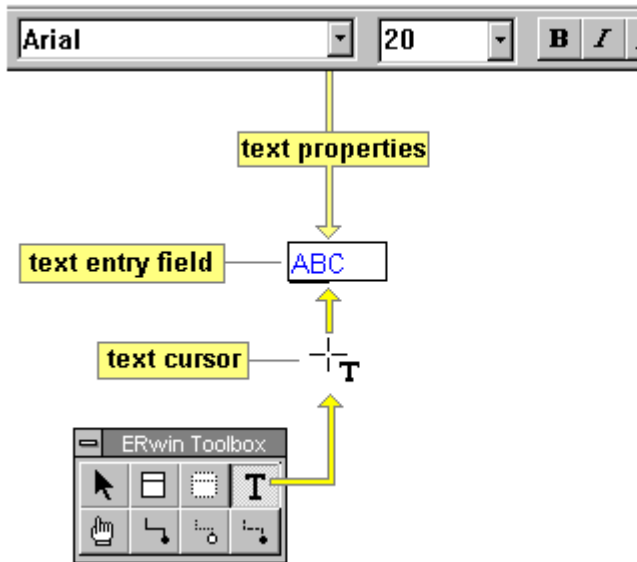
The color of an object can be related to the color it inherits from other objects in the diagram. For example, you can choose to have a foreign key attribute or column, inherit its font and/or color from its parent primary key or relationship.

You can also change the color of foreign keys so that you can easily identify owned primary key columns or attributes from foreign key columns or attributes in the diagram.





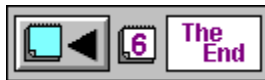
## Activity 10: Changing Fonts and Colors, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Text Blocks

You might also want to add text blocks to an ERwin diagram to identify or clarify model elements. Text blocks can include any text you choose, including diagram version numbers, notes, titles, and captions.

To change the color and fonts in the sample model, click the **Try It** button below.





## Fonts and Colors Try It

In this exercise, you will practice changing the colors of a text box and an entity outline. You will also change the colors of the foreign keys and set the stored display option to see the foreign keys.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\10-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.




## STEP 1: Change color of a text block

Select the **Store** entity.

*The entity name appears in a box with white text in a black background.*

If the Font and Color Toolbar is not on the screen, click on the **Font and Color Toolbar** option on the Window

menu.

Click the select Text Color arrow  
 to see color options and then  
select the dark green color square.

Click **Next Step** >





## STEP 2: Change color of an entity outline

Click the right mouse button on the **STORE** entity.

On the shortcut menu, click **Object  
Font/Color...**

Click the **Entity Outline** tab to open the Entity Line Color editor.

In the **Entity Line Color** list box, select **Purple** and then click **OK**.


Click **Next Step >**





### STEP 3: Change color of all foreign keys

On the Option menu, click **Default Font/Color**.

Click on the tab scrolling tool  at the top right corner of the dialog until you see the **Foreign Key** tab, then click on the tab.

Under Apply Settings to, click **All Objects**.

In the **FK Entity Attribute** list box near the bottom of the dialog, click **Red**.

Click **OK**.

Click **Next Step >**





#### **STEP 4: View foreign keys**

You must set the Store Display options to see the foreign keys.

Click the right mouse on the **diagram background**.

Select **Stored Display...** on the shortcut menu.

On the **Logical** tab, check **Foreign Key Designator** in the Entity Options box.

Repeat and select **Show Migrated Attributes** then click **OK**.

[Click Next Step >](#)





## STEP 5: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

**- The End -**

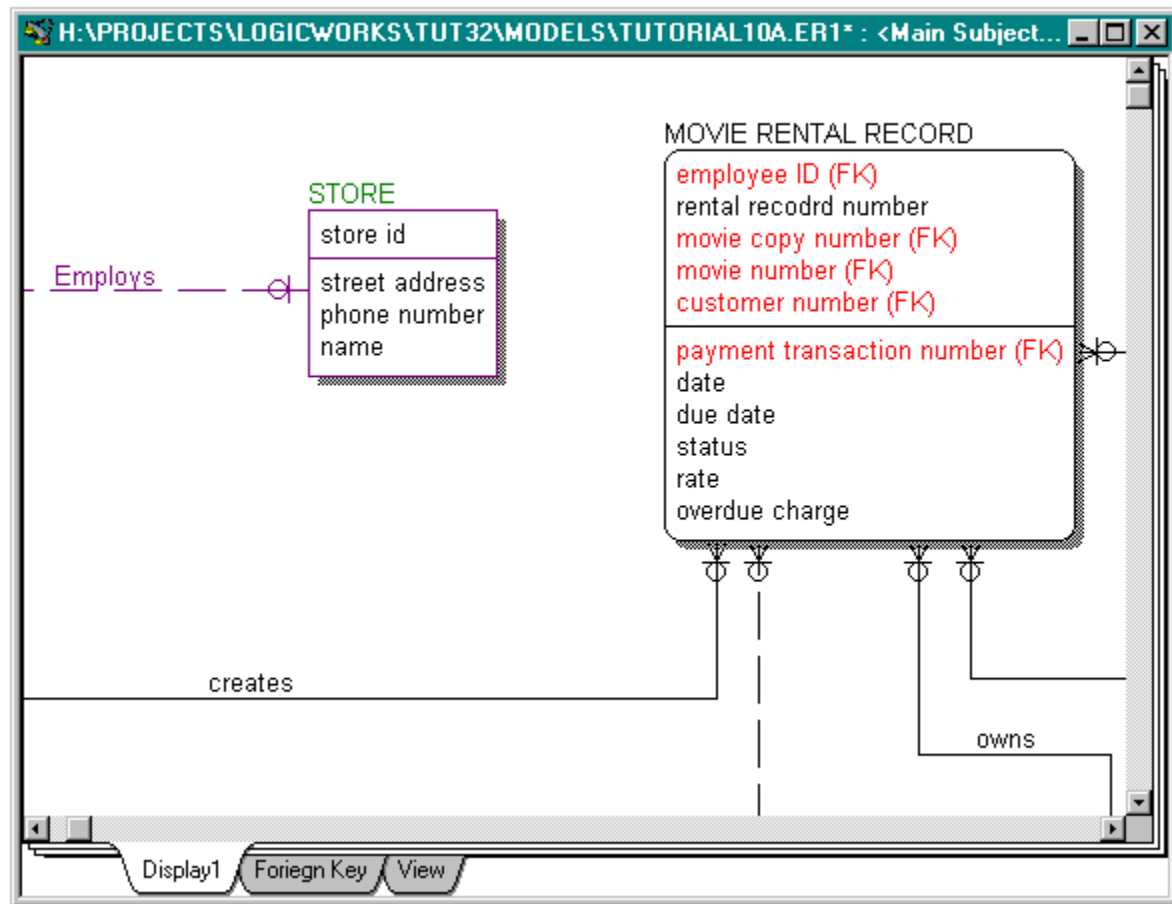






Fonts and Colors Check

{button Close,CW("check")}



## Activity 11: Switching to the Physical Model {ewc HLP25632, HLP256\_Tile, banner.bmp}



data modeler



database administrator

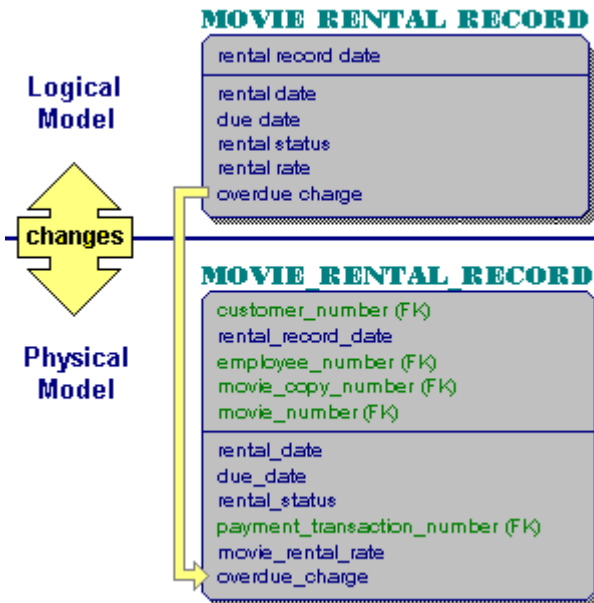
### Logical and Physical Models

In ERwin, you can include both a logical and a physical model in a single ERwin diagram:

- The logical model supports the needs of the **data modeler**, who captures business information and business rules.
- The physical model supports the needs of the **database administrator**, who focuses on the physical implementation of the model in a database.



## Activity 11: Switching to the Physical Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



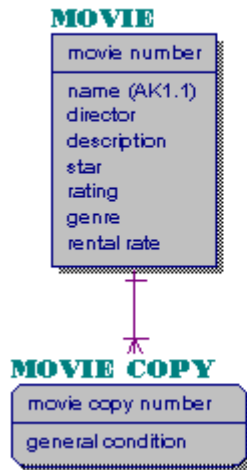
### Relationship between the Models

By default, the logical model is closely related to the physical model. If you make a change in the logical model, the change is automatically reflected in the physical model, and vice-versa.

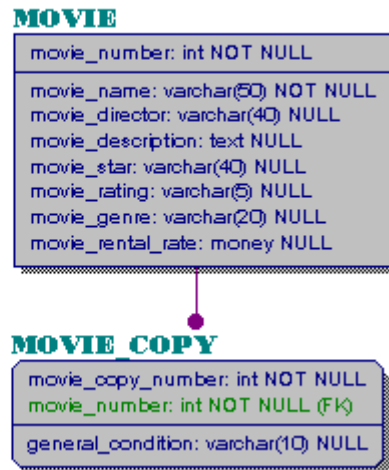
In the example to your left, if you add an attribute such as overdue charge to an entity, it also appears in the corresponding table in the physical model. By default, the physical model always displays the migrated keys.

## Activity 11: Switching to the Physical Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

### Logical Model



### Physical Model



### Same but Different...

The logical and physical models can look very different, depending on the:

- notation selected (IDEF1X, IE, or [DM](#) )
- display options selected
- modeling decisions to mark diagram objects “logical only” or “physical only”

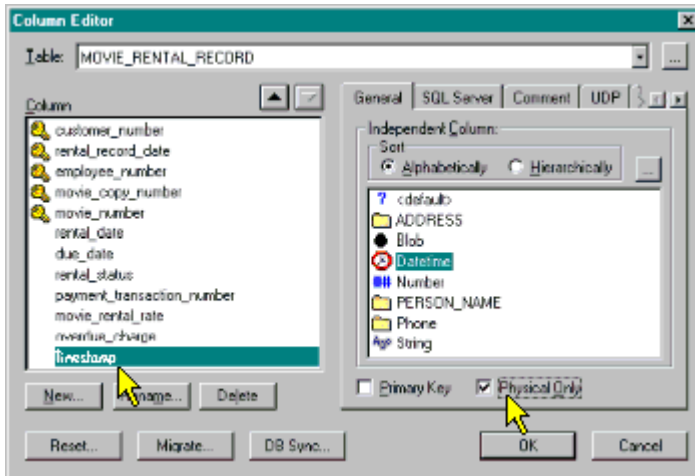
In the example on the left, the logical model is in the IE notation and display preferences are set to hide migrated attributes.

In contrast, the corresponding physical model has display preferences set to show datatype, null option, and foreign key designator.

**DM (Dimensional Model)**

A methodology for graphically depicting Fact, Dimension, and Outrigger tables in a star schema. This methodology is typically used to diagram dimensional models and data warehouses.

### Activity 11: Switching to the Physical Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

#### “Physical Only” Flag

You can use ERwin to create a logical model that is different from, but related to a physical model.

For example, in the physical model you can flag certain objects like “timestamp” in the physical model as “physical only” so that they do *not* appear in the corresponding logical model, and vice-versa.



## Column Editor

**Column Editor** [X]

Table:

Column

- customer\_number
- rental\_record\_date
- employee\_number
- movie\_copy\_number
- movie\_number
- rental\_date
- due\_date
- rental\_status
- payment\_transaction\_number
- movie\_rental\_rate
- overdue\_charge
- timestamp

General | SQL Server | Comment | UDP

Independent Column:

Sort

☒ Alphabetically ☐ Hierarchically

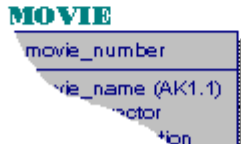
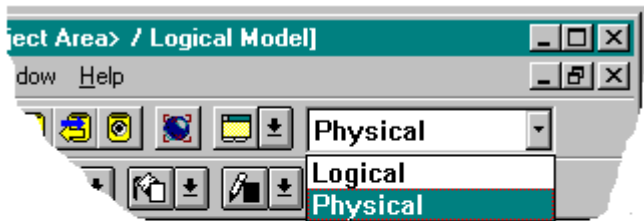
- <default>
- ADDRESS
- Blob
- Datetime
- Number
- PERSON\_NAME
- Phone
- String

☐ Primary Key ☒ Physical Only

Reset... Migrate... DB Sync... OK Cancel



**Activity 11: Switching to the Physical Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**

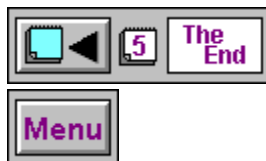


### ERwin Toolbar

You can easily switch between the logical to the physical model or vice-versa simply by selecting the Logical/Physical Model option list on the ERwin toolbar.

ERwin identifies the physical model as Dimensional in the toolbar when you use DM (Dimensional modeling) notation.

To try switching to the physical model, click the **Try It** button below.





### Physical Model Try It

In this exercise, you will practice switching to the physical model and then set a column to appear only in the physical model.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\11-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Switch to the physical model

Select the **Physical** model option from the list in the ERwin toolbar.

Click **Next Step >**



## STEP 2: Add a new column

If the Independent Column Browser is not visible, press CTRL+B.

Select **Datetime** from the list on the Independent Column Browser.



Drag Datetime from the browser and drop it into the **MOVIE** table between the DESCRIPTION and MOVIE\_STAR columns.

Click **Next Step >**



### STEP 3: Open the Column Editor

Right-click on the **MOVIE** table, then click **Column Editor** on the shortcut menu.

Click **MOVIE\_DATETIME** in the Column list.

Click **Next Step >**





#### STEP 4: Flag column as “physical only”

Click the General tab. At the bottom, check the “Physical Only” box.

Click **OK**.

Click **Next Step >**







## STEP 5: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

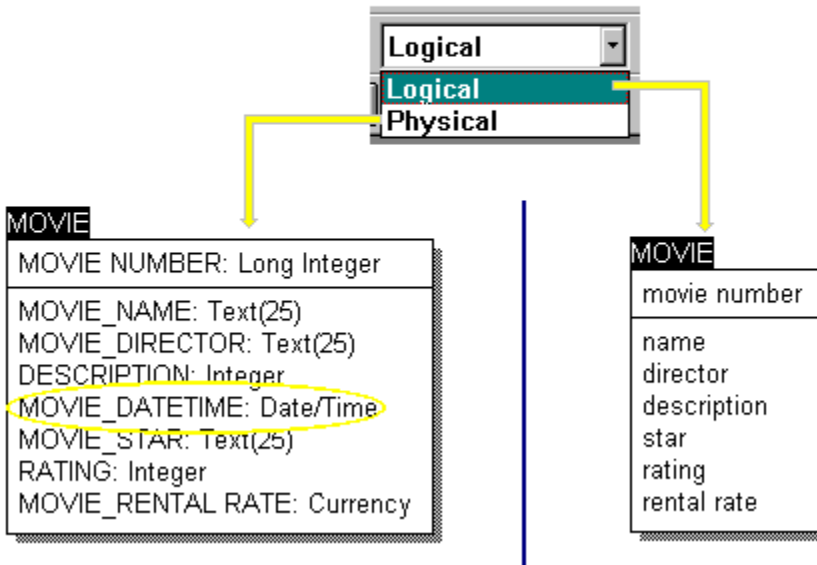
**- The End -**





### Physical Model Check {button Close,CW("check")}

When you switch between the physical and logical models, MOVIE\_DATETIME should only appear in the physical model as shown below.



## Activity 12: Using Views {ewc HLP25632, HLP256\_Tile, banner.bmp}

**Base Table**

Emp_ID	F_Name	L_Name	Phone	Salary	Hire_Date
1	Donna	Smith	555-1389	29,500	1/15/91
2	Gary	Jones	555-2009	34,700	6/23/94
3	Lin	Peters	555-5835	41,250	8/2/89



**View Table**

F_Name	L_Name	Phone
Donna	Smith	555-1389
Gary	Jones	555-2009
Lin	Peters	555-5835

### What is a View?

A **database view** is a custom-tailored presentation of the data in one or more tables. A view can also be thought of as a “stored query.” The result of the view query is a virtual table.

To the database user, a view appears just like a real table with a set of named columns and rows of data. Unlike a table however, a view is not a permanently stored set of data values.



## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Representing Views

ERwin represents a view as a box with a dashed outline and a view relationship as a dashed line with a hollow circle.

Notice in the example to your left that the **RECEIPT** View was created using two tables:

- MOVIE\_RENTAL\_RECORD
- CUSTOMER

Views are physical objects and can *only* be created and seen in the physical model.





## Representing Views

### CUSTOMER

customer_number
customer_first_name
customer_last_name
customer_address_1
customer_address_2
customer_city
customer_state
customer_zip
customer_phone
customer_credit_card
customer_credit_card_exp
customer_status_code

### RECEIPT

customer_number: CUSTOMER.customer_number: int
rental_date: MOVIE_RENTAL_RECORD.rental_date: datetime
due_date: MOVIE_RENTAL_RECORD.due_date: datetime
movie_rental_rate: MOVIE_RENTAL_RECORD.movie_rental_rate: int

### MOVIE RENTAL RECORD

customer_number
rental_record_date
employee_number
movie_copy_number
movie_number
rental_date
due_date
rental_status
payment_transaction_number
movie_rental_rate
overdue_charge

## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

**Base Table**

Emp_ID	F_Name	L_Name	Phone	Salary	Hire_Date
1	Donna	Smith	555-1389	28500	14/05/91
2	Gary	Jones	555-2009	9500	01/02/94
3	Lin	Peters	555-5835	41250	08/27/89



**View Table**

F_Name	L_Name	Phone
Donna	Smith	555-1389
Gary	Jones	555-2009
Lin	Peters	555-5835

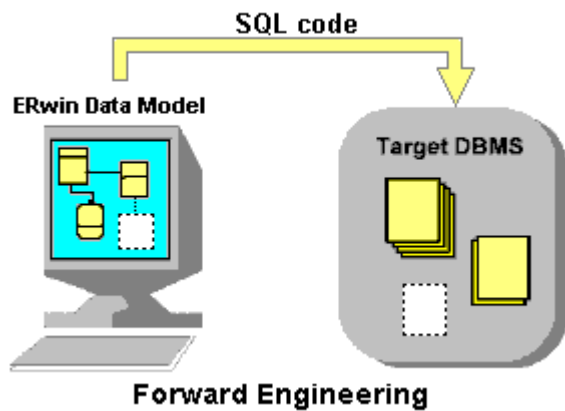
### Benefits of Using Views

Views are an important part of SQL. By using a view in your data model you can:

- Create your own customized presentation of the data stored in a database.
- Restrict access to data, allowing different users access to only certain rows or columns of a table.
- Simplify database access by creating customized structures tailored to the needs of individual users. A user can write a simple query on a view rather than a complex query on the actual database tables.



## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

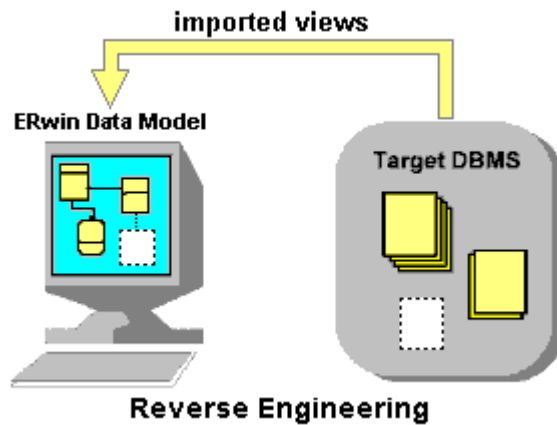


### Forward Engineering Views

When you **forward engineer** your data model to generate a database, ERwin automatically generates the SQL code defining the view and stores it in the database.



## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



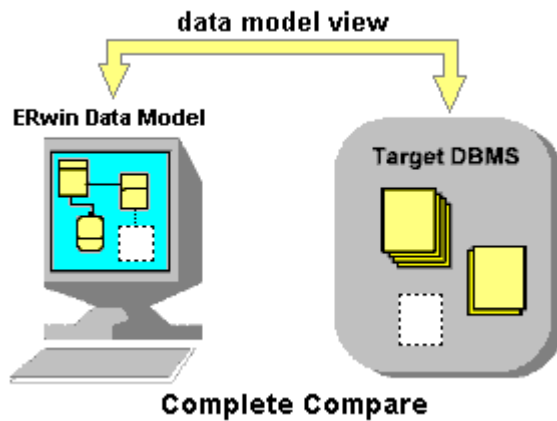
### Reverse Engineering Views

When you **reverse engineer** an existing database that includes one or more views, ERwin imports each view, parses the view syntax and, where possible, creates view relationships to the tables referenced by the view.





## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Complete Compare

Views are supported by ERwin's **complete compare** feature so that when you update a view either in your data model or in your database, you can easily keep your data model view specification in sync with your database view specification.



## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

View Table Tool



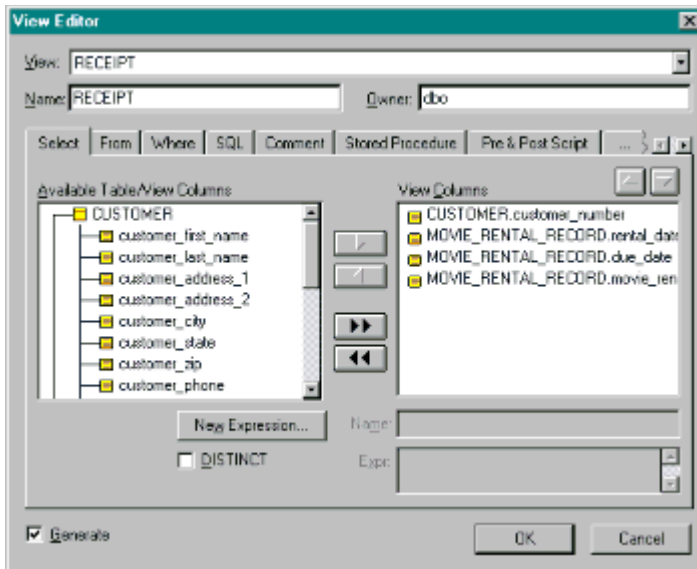
View Relationship Tool

### ERwin Toolbox

In the physical model, the ERwin toolbox contains the **View** and **View Relationship** tools.



## Activity 12: Using Views, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



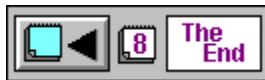
Click on the figure for a closer view.

### View Editor

You can use the View Editor to specify the properties of a view. You can:

- Add or remove view columns.
- Change the name of the view.
- Enter or edit the owner name for the selected view.
- Specify whether you want ERwin to generate the view SQL code during forward engineering or synchronization.

To look at different views in the sample model, click on the **Try It** button below.





## View Editor

**View Editor**

View: RECEIPT

Name: RECEIPT Owner: dbo

Select From Where SQL Comment Stored Procedure Pre & Post Script ...

Available Table/View Columns

- CUSTOMER
  - customer\_first\_name
  - customer\_last\_name
  - customer\_address\_1
  - customer\_address\_2
  - customer\_city
  - customer\_state
  - customer\_zip
  - customer\_phone

New Expression...

☐ DISTINCT

View Columns

- CUSTOMER.customer\_number
- MOVIE\_RENTAL\_RECORD.rental\_date
- MOVIE\_RENTAL\_RECORD.due\_date
- MOVIE\_RENTAL\_RECORD.movie\_ren

Name:

Expr:

☒ Generate

OK Cancel

## Using Views Try It

In this exercise, you will practice modifying an existing view in the sample model.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\12-tutor.er1](#)  
from the **File|Open...** menu.



Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Display the views for the model

Click on the **View** Display tab at the bottom of the form.

*The View tab represents a stored display designed to show all the views associated with the sample model.*

Click **Next Step >**





## STEP 2: Locate the views

Locate the **OVERDUE NOTICE** view.

*Notice that the Overview Notice view was created using the CUSTOMER and MOVIE RENTAL RECORD tables.*

Click **Next Step >**







### STEP 3: Create a View Column Expression

Right-click on the **Overdue\_Notice** view, then click **View Editor** on the shortcut menu.

Click **New Expression...**

Type `OVERDUE_CHARGE_RATE` in the Column box.

Type `movie_rental_rate*1.5` in the Expression box.

Click **OK**.

Click **OK** on the View Editor dialog.

This expression will calculate the  
overdue rate based on the movie rate  
time 1.5.

Click **Next Step** >



#### **STEP 4: Close the Model**

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

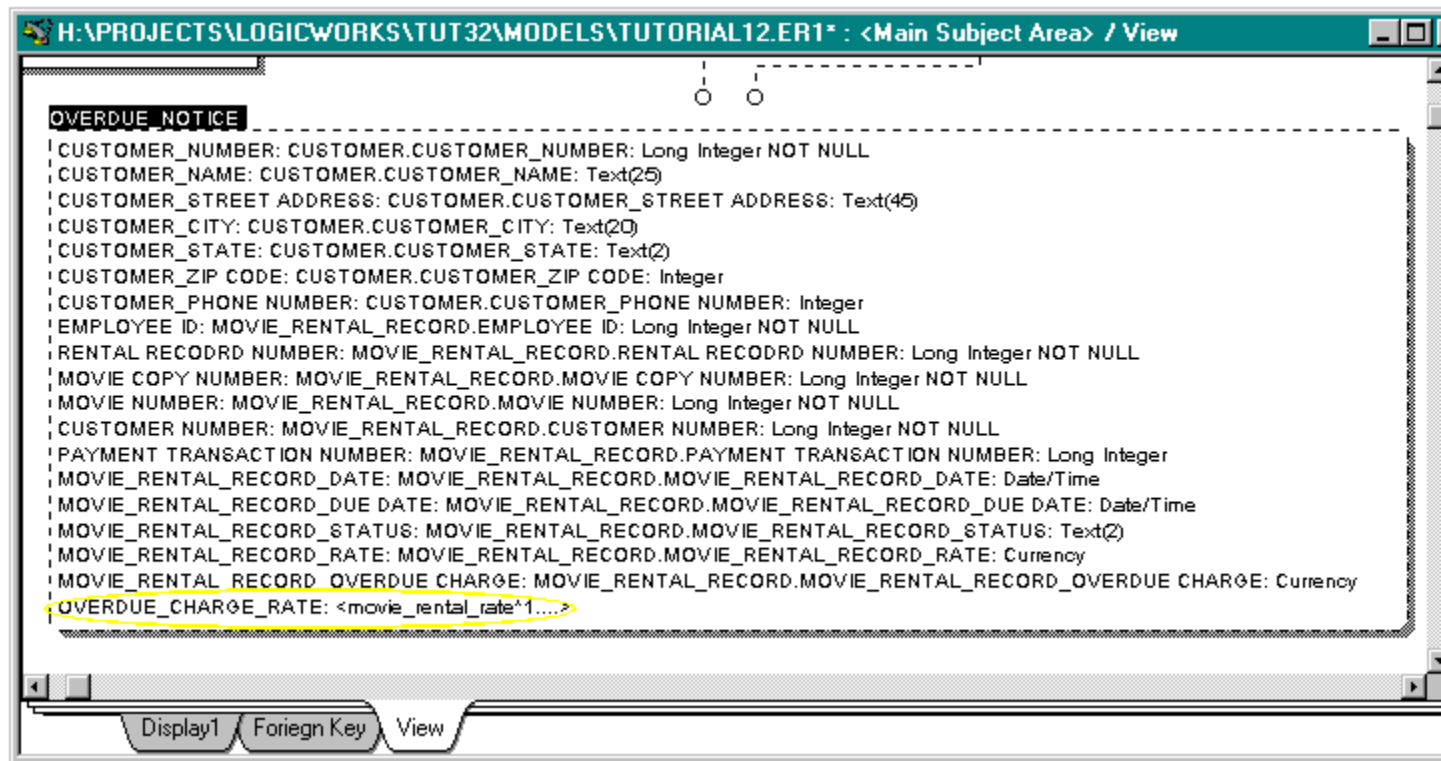
**- The End -**



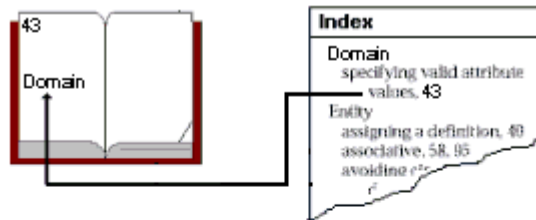




Using Views Check {button Close,CW("check")}



### Activity 13: Creating an Alternate Key Index for the STORE Table {ewc HLP25632, HLP256\_Tile, banner.bmp}



**Index**

Last Name	Phone	Rows
Smith	555-1234	2,5,10

**Movie Rental Table**

	CUST_NO	RENTAL_DATE	MOVIE_NO	
1				
2	58	01/21/97	123	
3				
4				
5	58	01/31/97	72	
6				
7				
8				
9	58	03/10/97	257	
10				

### Using Indexes

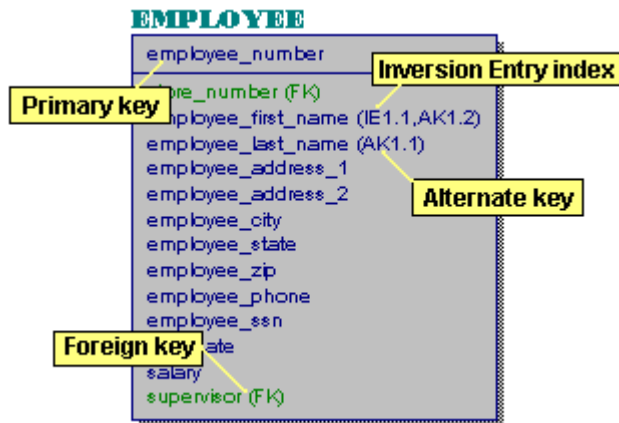
Just as the index in a book helps you to find information by quickly listing all of the pages where a particular topic is discussed in the book, an index in a database table points to all of the rows where a particular column value is stored.

You can use indexes to:

- Sequentially access the indexed file
- Directly access individual records in the indexed file based on a given value for the indexed field



**Activity 13: Creating an Alternate Key Index, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



### Four Types of Indexes

ERwin supports four types of indexes:

- [Primary key \(PK\) index](#)
- [Foreign key \(FK\) index](#)
- [Alternate key \(AK\) indexes](#)
- [Inversion entry \(IE\) indexes](#)



### Primary key (PK) index

A primary key (PK) index is an index on the primary key column(s) in a particular table.

#### Notes:

- You can *only* have one primary key index for each table. Each index can include multiple columns.
- A primary key index is unique, so the indexed columns cannot have duplicate values nor can they be null (empty).
- ERwin automatically creates a primary key index for each table that has one or more primary key columns.



### **Foreign key (FK) index**

A foreign key (FK) index is an index on one or more foreign key columns migrated through a single relationship to the child table.

ERwin automatically creates a foreign key index for each set of foreign key columns that are migrated through a relationship to a child table from its parent.

Each table can include zero, one, or more foreign key indexes, depending on the number of relationships in which the table participates as a child.

### **Alternate Key (AK) indexes**

A unique index other than the primary key index. Duplicate values in an AK index is *not* allowed.

### **Inversion Entry (IE) indexes**

A non-unique, or inversion entry (IE) index lets you quickly access data using values that are not unique, such as EMPLOYEE last name. Duplicate values in the IE index *are* allowed.

**Activity 13: Creating an Alternate Key Index, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**

**MOVIE**

```
movie_number: int
movie_name: varchar(50) (AK11)
movie_director: varchar(40)
movie_description: text
movie_star: varchar(40)
movie_rating: varchar(5)
movie_genre: varchar(20)
movie_rental_rate: money
```

Alternate key

**What is an Alternate Key Index?**

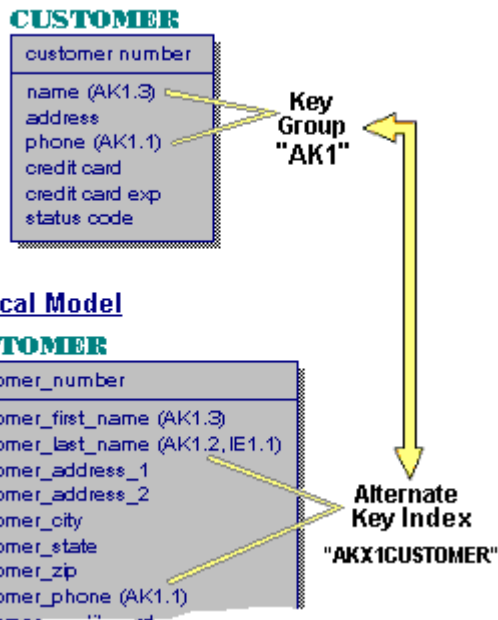
A unique, or alternate key (AK) index is an index on a set of columns in a particular table other than the primary key column set. Duplicate values in the indexed columns are *not* allowed.

In the example to the left, the movie-name column in the MOVIE table is an alternate key index. You can quickly find a movie by its name assuming that there are no duplicate movie names.



### Activity 13: Creating an Alternate Key Index, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

#### Logical Model

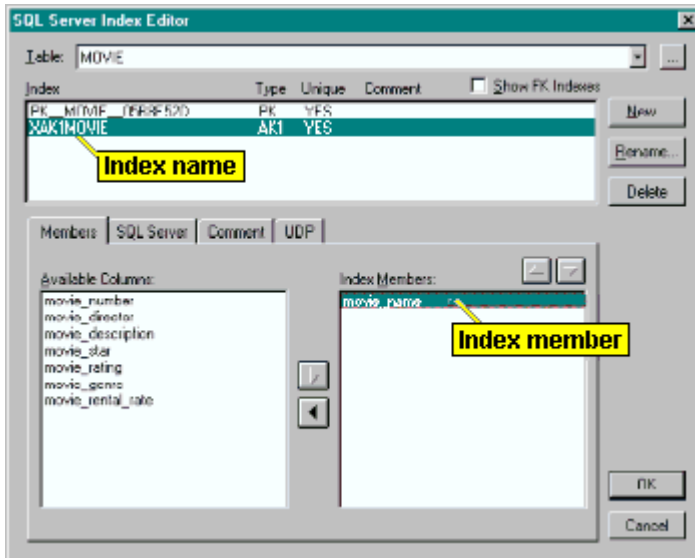


#### Logical Key Groups and Physical Indexes

At the logical level if you create an alternate key group that contains one or more attributes, ERwin automatically creates the corresponding unique index for those columns in the corresponding table.



### Activity 13: Creating an Alternate Key Index, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



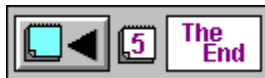
Click the figure for a closer view.

#### Index Editor

At the physical level, you can use the **Index Editor** to create an index on a table. The default name for a unique index is XAKnTableName, as specified by the Index Name Macro.

When you create an index you can use the “Physical Only” property so indexes do not appear in the logical model as key groups.

To create an alternate key index in the sample model, click the **Try It** button below.







## Index Editor

The screenshot shows the 'SQL Server Index Editor' window. At the top, the 'Table:' dropdown is set to 'MOVIE'. Below this is a table listing existing indexes. The first index is 'PK MOVIE\_05B8E52D' with type 'PK' and 'Unique' set to 'YES'. The second index, 'XAK1MOVIE', is selected and highlighted in blue, with type 'AK1' and 'Unique' set to 'YES'. To the right of the table are buttons for 'New...', 'Rename...', and 'Delete'. A checkbox labeled 'Show FK Indexes' is also present. Below the index table is a tabbed interface with 'Members', 'SQL Server', 'Comment', and 'UDP' tabs. The 'Members' tab is active, showing two panes: 'Available Columns:' on the left and 'Index Members:' on the right. The 'Available Columns:' pane lists 'movie\_number', 'movie\_director', 'movie\_description', 'movie\_star', 'movie\_rating', 'movie\_genre', and 'movie\_rental\_rate'. The 'Index Members:' pane contains 'movie\_name', which is selected. Between the panes are two arrow buttons for moving items. At the bottom right are 'OK' and 'Cancel' buttons.

Table: MOVIE

Index	Type	Unique	Comment
PK MOVIE_05B8E52D	PK	YES	
XAK1MOVIE	AK1	YES	

☐ Show FK Indexes

New...  
Rename...  
Delete

Members | SQL Server | Comment | UDP

Available Columns:

- movie\_number
- movie\_director
- movie\_description
- movie\_star
- movie\_rating
- movie\_genre
- movie\_rental\_rate

Index Members:

- movie\_name

OK  
Cancel

### Alternate Key Index Try It

In this exercise, you will create a new alternate key index.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\13-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Select the table

Right-click on **STORE** table, then choose **Access Index** on the shortcut menu.

“Access” is the name of the target server for this model.

Click **Next Step >**







## STEP 2: Create a new alternate key index

Click the **New...** button.

Check that the **Unique** option is selected.

If you clear the Unique check box,  
ERwin automatically creates an  
Inversion Entry index.

You can change the Key Group or  
Index names to match your company's  
naming conventions (e.g. AK1STORE).

Click **OK**.

Click **Next Step >**



### STEP 3: Choose column members

On the Members tab, select  
**Store\_PhoneNumber** in the

Available Columns list, then click to  
include in the Index Members list.

Click **OK**.

*ERwin creates an AK index for the corresponding column in the physical model.*

Click **Next Step >**



#### STEP 4: View AK Index

*The AK Designator option must be set to on to see the alternate key index.*

Right-click on blank area of the diagram, then click **Display Options/Tables** on the short-cut menu.



Select **Alternate Key Designator** to the **on** position.

Locate the AK Designator in the **STORE** entity.

Click **Next Step >**





## STEP 5: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

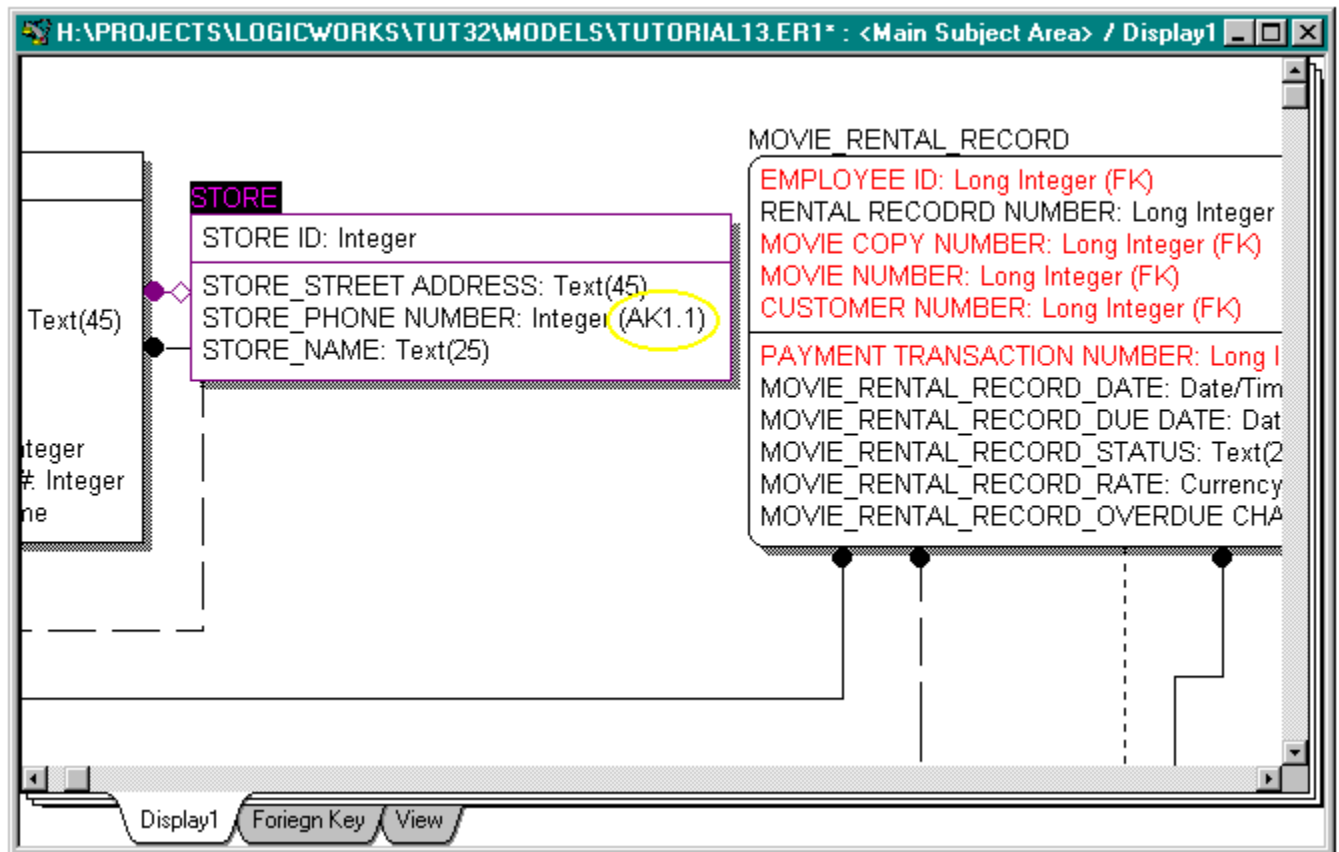
**- The End -**



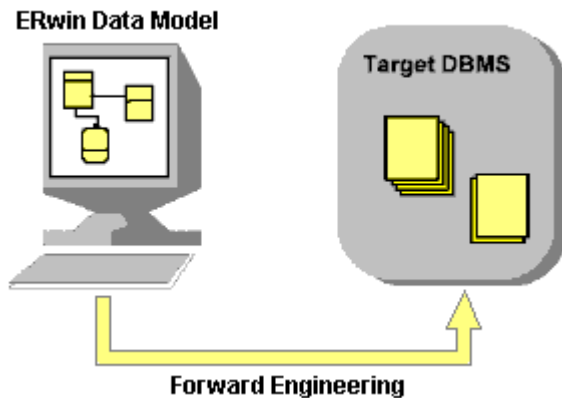




Alternate Key Index Check {button Close,CW("check")}



## Activity 14: Forward Engineering a Data Model {ewc HLP25632, HLP256\_Tile, banner.bmp}



### What is Forward Engineering?

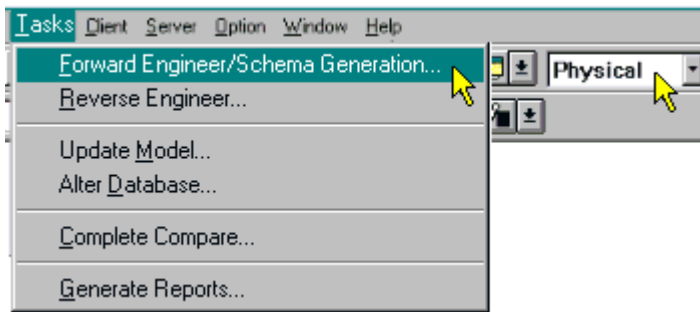
**Forward engineering** is a process that generates the physical database schema from the data model. You can use ERwin's forward engineering feature to design and create your database *without* writing a single SQL CREATE TABLE or CREATE INDEX statement.

When you generate a schema, you can choose to generate:

- Tables
- Triggers
- Stored procedures
- Indexes
- Constraints
- Physical storage objects
- Other database features supported by your target DBMS



**Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**

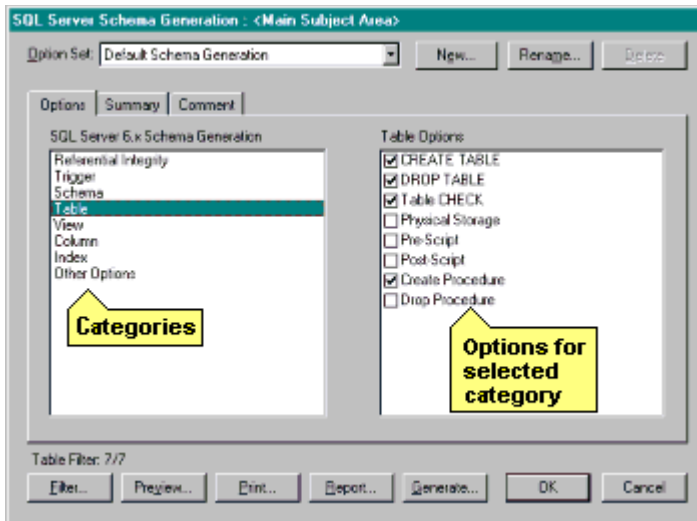


**Tasks Menu- Forward Engineer / Schema Generation**

When you are in the physical model, you can choose the Forward Engineer/Schema Generation option on the Tasks menu to create a database schema.



#### Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

#### Changing Schema Generation Options

ERwin lets you view and set schema generation options by category. In the Schema Generation Editor, the left panel lists all the categories and the right panel lists all the options for the selected category.

The target server you select determines which options appear in the Editor.

In the example to your left, the Drop Table option has been selected. This means the generated schema will include DROP TABLE statements for each table that you create.



## Schema Generation Options Window

**SQL Server Schema Generation : <Main Subject Area>**

Option Set: Default Schema Generation New... Rename... Delete

Options Summary Comment

SQL Server 6.x Schema Generation

- Referential Integrity
- Trigger
- Schema
- Table**
- View
- Column
- Index
- Other Options

Table Options

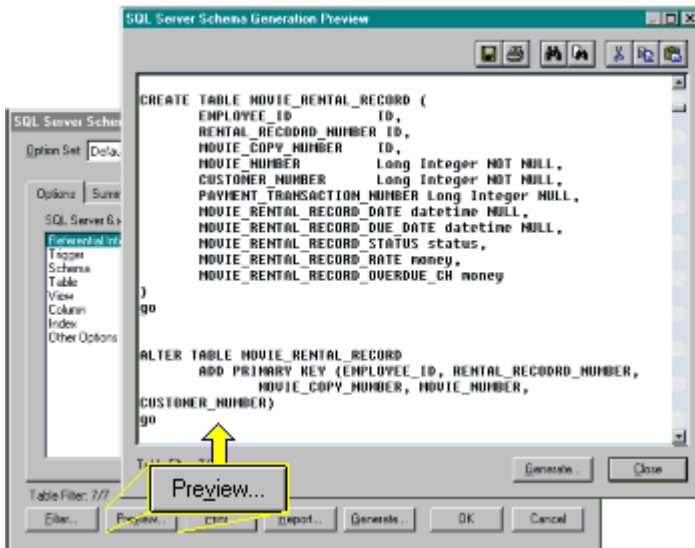
- ☒ CREATE TABLE
- ☒ DROP TABLE
- ☒ Table CHECK
- ☐ Physical Storage
- ☐ Pre-Script
- ☐ Post-Script
- ☒ Create Procedure
- ☐ Drop Procedure

Table Filter: 7/7

Filter... Preview... Print... Report... Generate... OK Cancel



#### Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

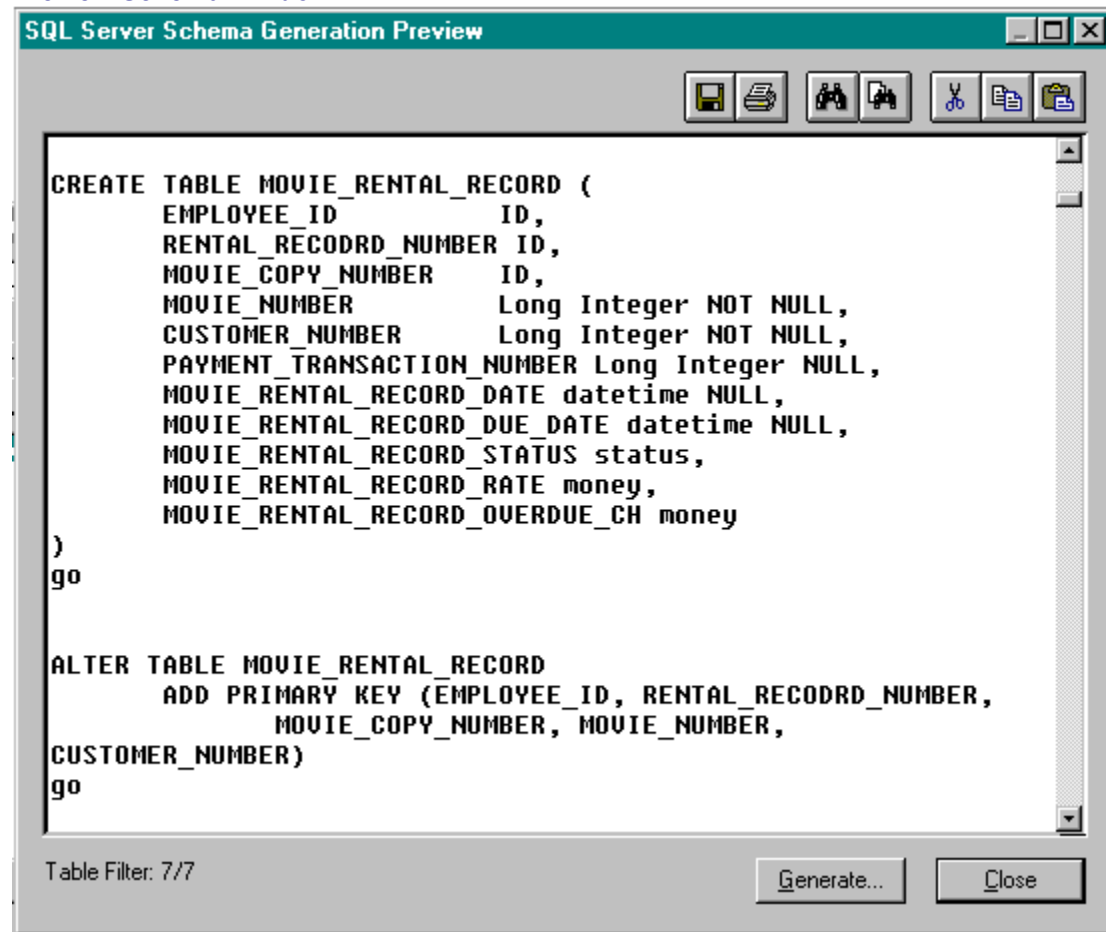
#### Preview Schema

When you choose Forward Engineering/Schema Generation on the Tasks menu, ERwin opens the Schema Generation Options dialog.

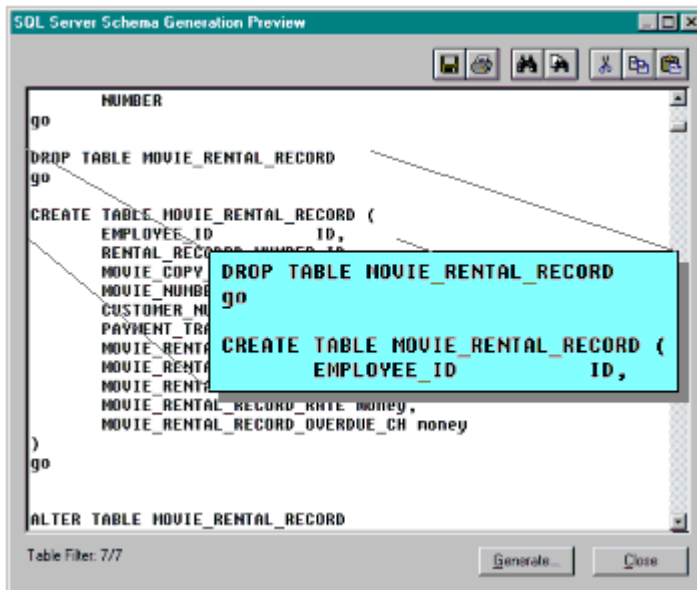
Simply click on the Preview button to view the generated schema.



## Preview Schema Window



## Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}

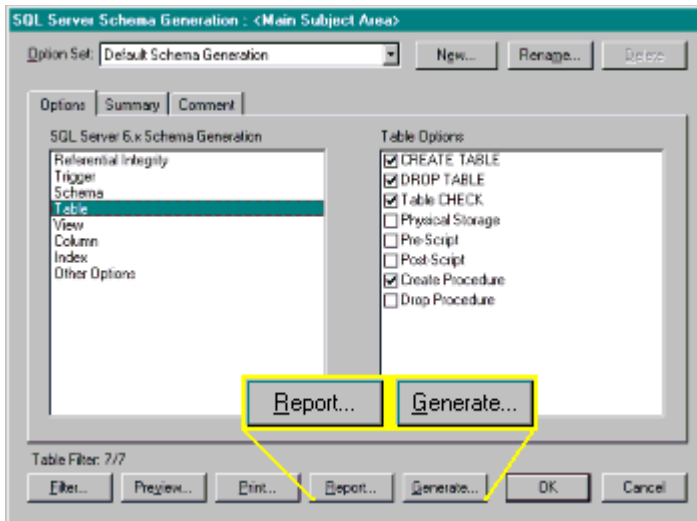


## Resulting Schema

When you preview the generated schema, you can see how your options affect the schema script. Because the Drop Table option was selected, the DROP TABLE statements appear in the script.



#### Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



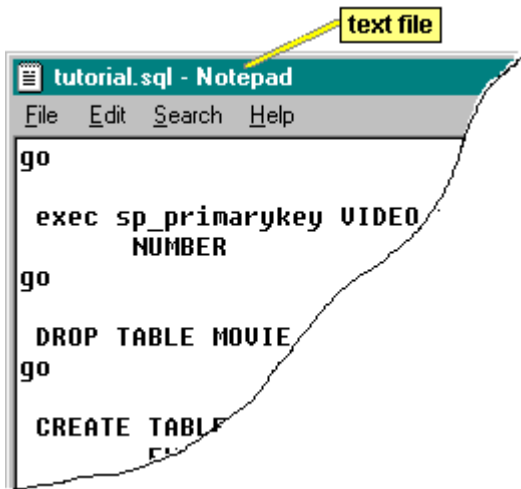
#### Two Options

Once you are satisfied with the content of the generated schema, ERwin gives you two choices. You can:

- Connect ERwin directly to the target server and generate the schema in one step by clicking on the **Generate** button.
- Save a SQL DDL (Data Definition Language) script as an ASCII text file by clicking on the **Report** button.



#### Activity 14: Forward Engineering, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Generating a Script

When you save a DDL script as a text file, you must run it as a separate step on the server to generate the schema. The resulting text file can be:

- Opened in Windows Notepad.
- Imported into a word processing application.
- Loaded by any utility used to interpret SQL scripts such as SPUFI for DB2, ISQL for SQL Server, SQL\*DBA for ORACLE, or SQLTalk for SQLBase.

To forward engineer a data model, click the **Try It** button below.







### Forward Engineering Try It

In this exercise, you will review a sample database schema that was previously generated from a data model.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\14-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

Check that **Physical** model is displayed in the option list on the ERwin toolbar.

You may need to rearrange the  
objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Choose the Forward Engineering task

On the **Tasks** menu, choose  
**Forward Engineering / Schema  
Generation...**

*The SQL Server Schema  
Generation dialog is displayed.*

If the option is unavailable (dimmed),  
check that you are in the Physical

model.

Click **Next Step** >





## STEP 2: Preview the generated schema

Click the **Preview** button at the bottom of the window.

You are looking at the SQL Server schema, although you will *not* be generating an SQL database as part of this exercise.

Using the scroll bar, review the schema.

Click the **Close** button.

Click **Next Step >**





### STEP 3: Change the schema options

Click on **Options** tab.

Select **Table** from the **Schema Generation** list box.

Verify that the **DROP TABLE** option is off.

Click **Next Step >**





#### STEP 4: Explore other options

Click on other Schema Generation categories and look at the options.

Click **Next Step** >



## STEP 5: View the Option Summary

Click the **Summary** tab.

*The summary tab lists the options you selected and can be modified, if necessary.*

Click **Next Step >**



## STEP 6: Save the Option Set

Click the **New...** button to save the schema generation options.

Type `Myoptions`.

Click **OK**.

*Myoptions is displayed in the  
Option Set list.*

Click **Next Step >**



## STEP 7: Close the Model

Click below to verify that you have performed the activity correctly, then

click .



Click **OK** on the SQL Server Schema Generation dialog to close it.

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save**  
**as .ER1 file** then click **OK**.

To close without saving, select **Close**  
**without saving** then click **OK**.

Click below to close this Try It card.

**- The End -**



Forward Engineering Check {button Close,CW("check")}

SQL Server Schema Generation : <Main Subject Area>

Option Set:

Options | Summary | Comment

SQL Server 6.x Schema Generation

☐ Schema

☒ Create Rule

☒ Create Default

☒ Create Procedure

☒ sp\_addtype

☐ Table

☒ CREATE TABLE

☒ Table CHECK

☒ Create Procedure

☐ View

☒ CREATE VIEW

☒ Create Procedure

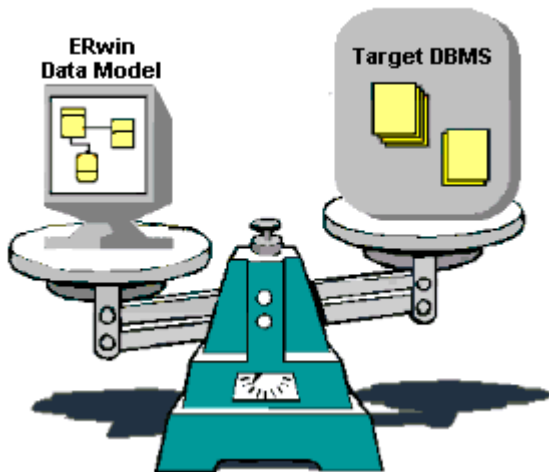
☐ Column

☒ Show Selected Only

☐ Edit Options

Table Filter: 8/8

**Activity 15: Synchronizing the Data Model with the Database {ewc HLP25632, HLP256\_Tile, banner.bmp}**



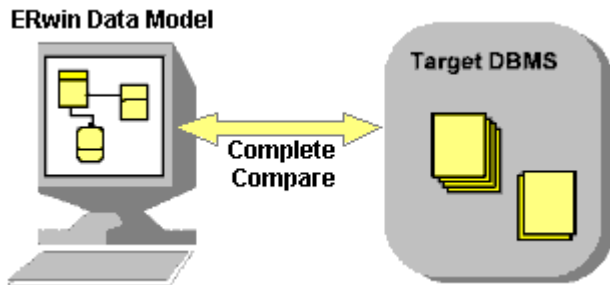
**Synchronizing Process**

Database environments are dynamic. No matter how well your original database is designed, you must be able to alter the database to support unforeseen needs and requirements as they arise.

The same is true for your ERwin data model. If the goals for the database expand, you must be able to manage the changes to the data model in a controlled way and keep the model in sync with the database.



**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



**Complete Compare**

By using ERwin's Complete Compare feature, you can update your physical database schema to reflect changes to the model and update the model to capture changes made directly to the database.

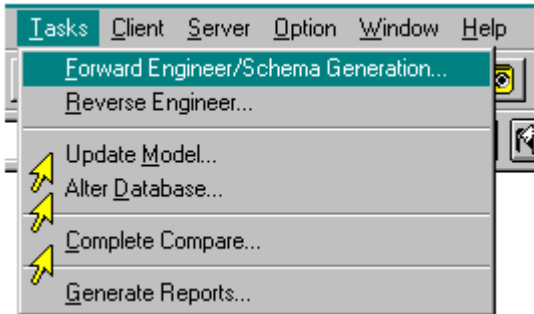
This synchronization process captures differences between an ERwin data model and a physical database, schema script file, ModelMart diagram, or even a different ERwin model.

ERwin detects the differences in tables and table properties, columns and column properties, views, triggers, defaults, stored procedures, and other database objects.





**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



### Three Methods to Update Differences

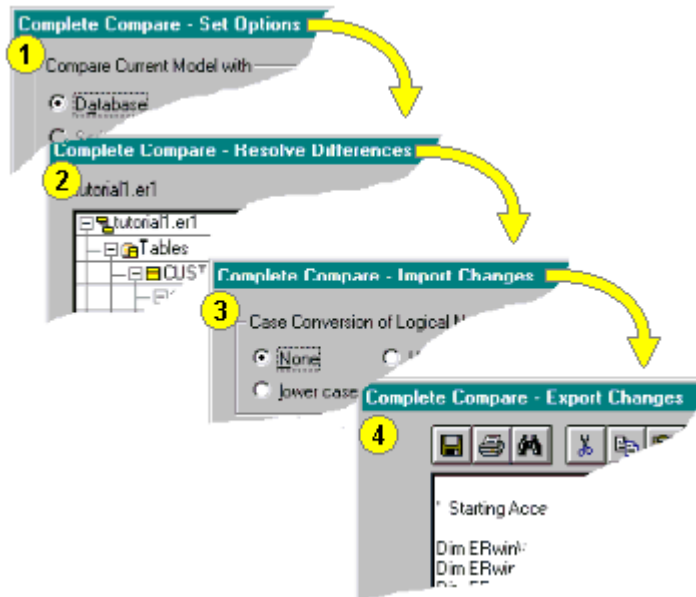
Once ERwin detects the differences, you have three methods to update the differences. You can:

- Update the current ERwin data model.
- Alter the database, either directly or by generating an alter script, which can be customized and used to update the database.
- Update both the database and the model at the same time.

The remainder of this activity will focus on updating both the database and data model using the Complete Compare feature.



**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



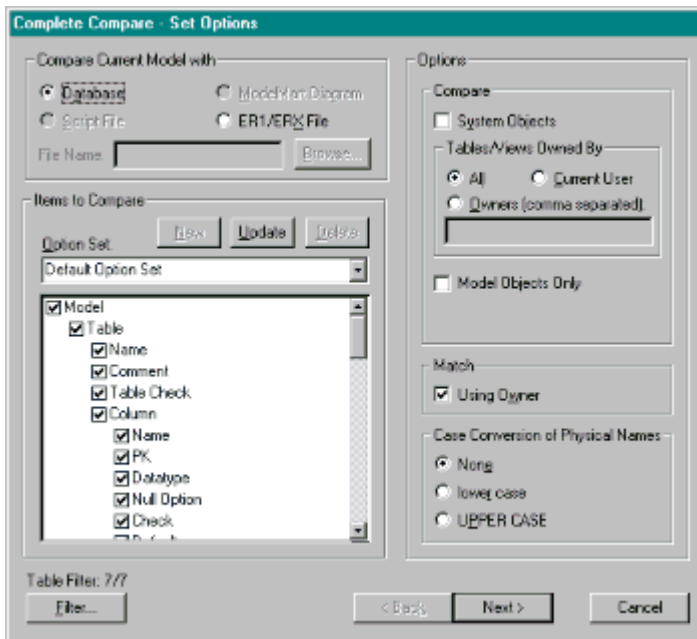
**Four Steps to Perform Complete Compare**

When you choose Complete Compare on the Tasks menu, ERwin displays a series of dialogs that guide you through the process of importing model information into ERwin and exporting changes to a database or script file. These steps include:

1. Setting complete compare options.
2. Resolving differences by selecting items you want to import, export, or ignore.
3. Importing changes into your model.
4. Exporting changes to a database or script file.



**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



Click on the figure for a closer view.

### Complete Options Dialog

When you choose Complete Compare on the Tasks menu, ERwin opens the Complete Compare - Set Options dialog.

This dialog is used to select items to compare and to set comparison options.



## Complete Compare Options Dialog

**Complete Compare - Set Options**

Compare Current Model with:

☒ Database ☐ ModelMart Diagram  
☐ Script File ☐ ER1/ERX File

File Name:

Items to Compare:

Option Set:

Option Set:

- ☒ Model
  - ☒ Table
    - ☒ Name
    - ☒ Comment
    - ☒ Table Check
    - ☒ Column
      - ☒ Name
      - ☒ PK
      - ☒ Datatype
      - ☒ Null Option
      - ☒ Check

Options:

Compare

☐ System Objects

Tables/Views Owned By:

☒ All ☐ Current User  
☐ Owners (comma separated):

☐ Model Objects Only

Match

☒ Using Owner

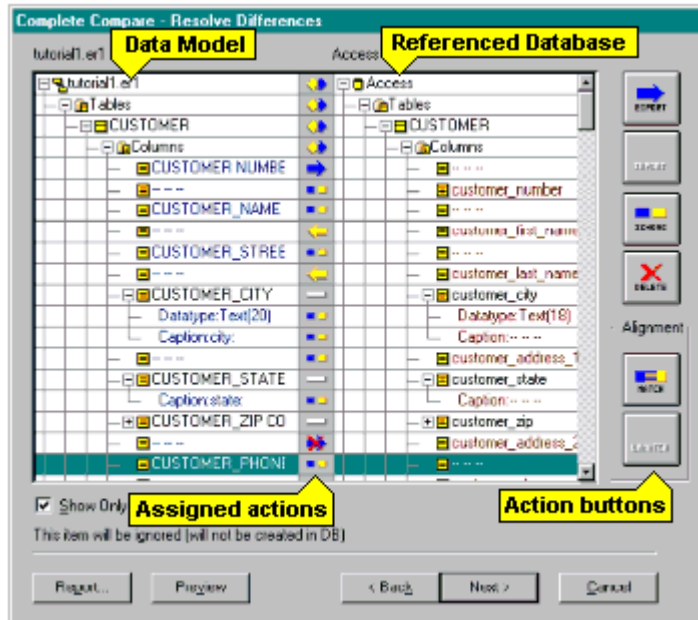
Case Conversion of Physical Names

☒ None ☐ lower case ☐ UPPER CASE

Table Filter: 7/7



**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



Click on the figure for a closer view.

### Resolving Differences Dialog

ERwin lists the differences between the active ERwin model on the left and the referenced database, script, or model on the right.

The middle column indicates how you intend to resolve those differences. The buttons to the right of the dialog are used to assign the appropriate actions.



## Resolving Differences Dialog

**Complete Compare - Resolve Differences**

tutorial1.er1      Access

tutorial1.er1		Access
Tables		Tables
CUSTOMER		CUSTOMER
Columns		Columns
CUSTOMER_NUMBE	→	---
---	→	customer_number
CUSTOMER_NAME	→	---
---	→	customer_first_name
CUSTOMER_STREE	→	---
---	→	customer_last_name
CUSTOMER_CITY	→	customer_city
Datatype:Text(20)	→	Datatype:Text(18)
Caption:city:	→	Caption:---
---	→	customer_address_1
CUSTOMER_STATE	→	customer_state
Caption:state:	→	Caption:---
CUSTOMER_ZIP CO	→	customer_zip
---	→	customer_address_2
CUSTOMER_PHONE	→	---

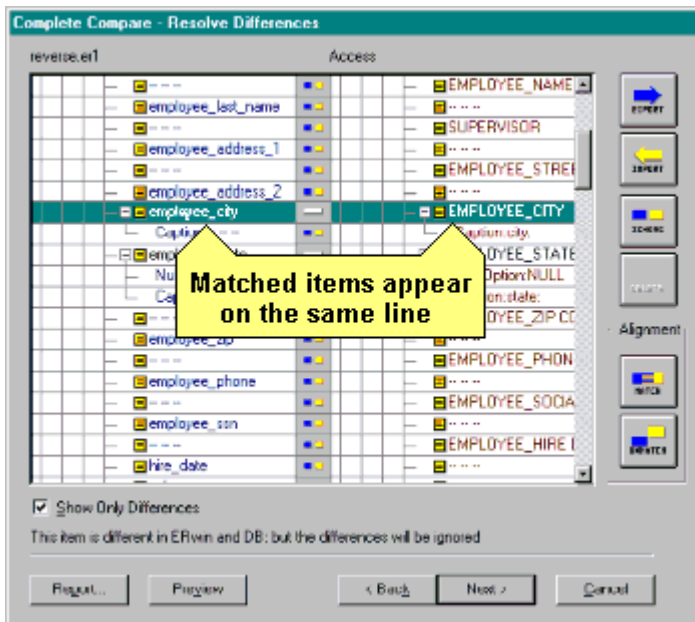
☒ Show Only Differences

This item will be ignored (will not be created in DB)

Report...    Preview    < Back    Next >    Cancel

EXPORT  
IMPORT  
IGNORE  
DELETE  
Alignment  
MATCH  
MISMATCH

**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



Click on the figure for a closer view.

### Items that Match

Items that ERwin matches automatically are displayed on the **same line** in the Resolve Differences dialog.

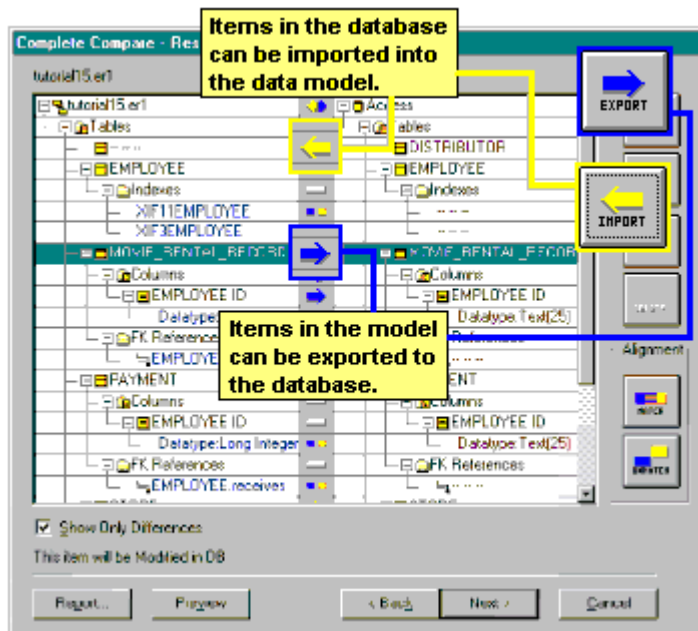
ERwin matches objects of the same type if they have the same name. You change the matching if you need to.



## Resolve Differences Dialog

[illegible]

**Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



**Identifying Items to Import and export**

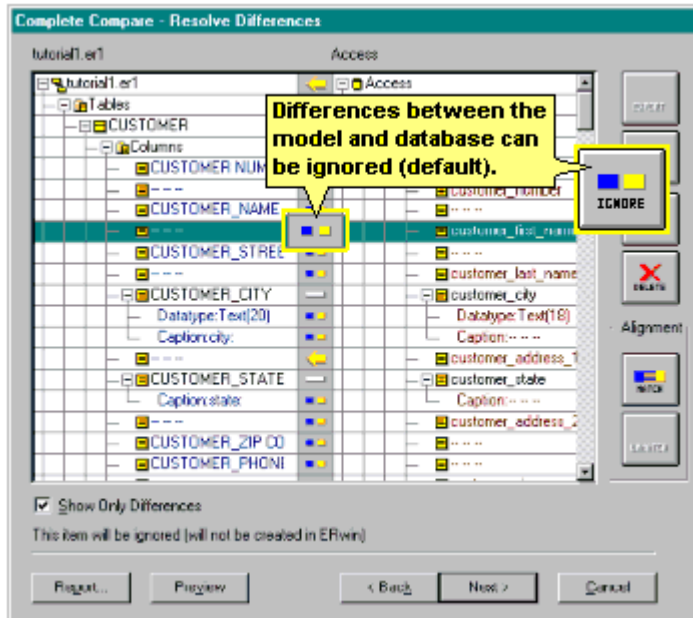
You can select items to import using the **Import** button.

You can select items to export using the **Export** button.





### Activity 15: Synchronizing the Data Model with the Database, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



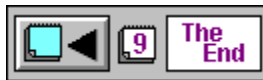
## Items Set to Ignore

If ERwin locates an item in the data model that is not in the source database or file, ERwin automatically sets the action for that item to **Ignore**.

You can also set the action to ignore if you want to:

- override a previous action you applied
- ignore some of an objects properties while importing and exporting others

To perform a Complete Compare on the sample model, click the **Try It** button below.





### Complete Compare Try It

In this two part exercise, you will practice

- 1) Importing items.
- 2) Exporting a column.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\15-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.



## STEP 1: Set Complete Compare Options

On the **Task** menu, choose **Complete Compare**.

Check that the **Compare Current Model with Database** is selected.

Check that **Default Option Set** is selected.

Click **Next>** on the Set Options dialog.

Click **Next Step >**







## STEP 2: Complete the Connection Box

In the **Access Connection box**,  
check that the username is `admin`.

Tab over the password field.

Click the **Browse** button and select  
path \compare.mdb

Click **Connect**.

Click **Next Step >**





### STEP 3: Select Items to Import

On the **Resolve Difference** dialog, select **DISTRIBUTOR** in the Access list.

Click on **Import** button.

Click **Next Step >**







## STEP 4: Import Changes into Your Model

Click **Next>** on the Resolve Differences Dialog.

*The Complete Compare - Import Changes dialog displays.*

*For more information about this dialog, press F1.*

Click **Start Import** button in the bottom right corner of the dialog.

Click **Next Step >**



## STEP 5: Review the Changes


Click the **View Results** button.

Click **Next Step >**



## **STEP 6: Finish Importing Items**

Click  below to verify that you have

performed the activity correctly, then click  .

In the **Import Summary** dialog, click the **Close** button.

In **Import Changes** dialog, click the **Finish** button.

Click **Yes** to layout your model.

**- Finished with Part 1 -**

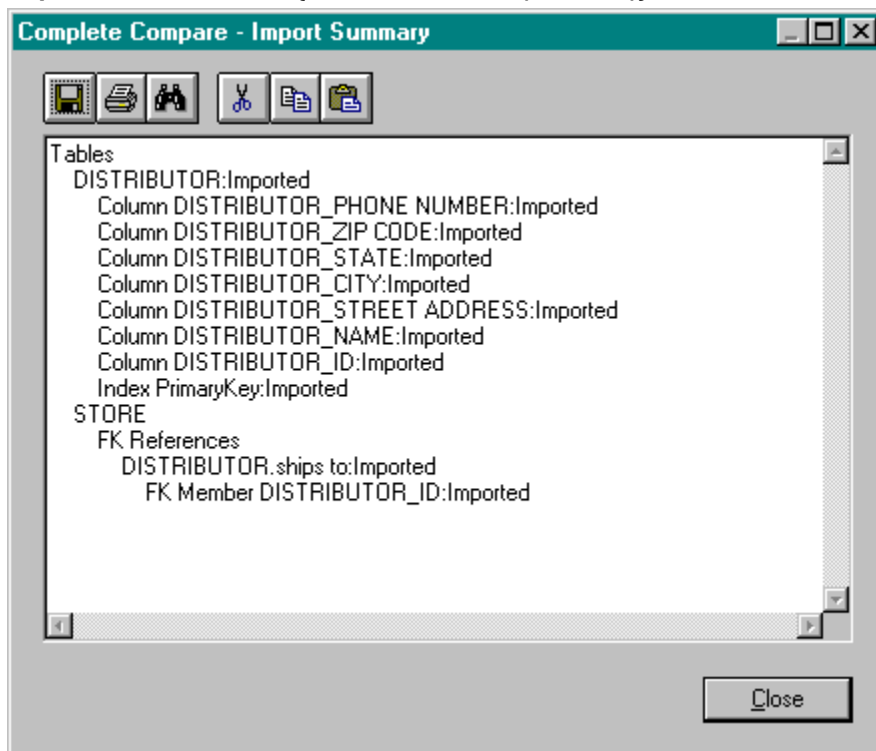
Click **Part 2 ▶** below to continue.





Part 2 ►

Import Items Check {button Close,CW("check")}



## Export a Column Try It

In this part of the Try It you will export a table.

Click **Next Step >** to continue.



## STEP 1: Set Complete Compare Options

On the **Task** menu, choose  
**Complete Compare**.

Check that **Compare Current  
Model with Database** is selected.

Check that **Default Option Set** is selected.

Click **Next>** on the Set Options dialog.

Click **Next Step >**







## STEP 2: Select Items to Export

On the Resolve Difference dialog, select the first Datatype under **Employee ID** from the model list.

Click on the **Export** button.

Click **Next Step >**





**STEP 3: Export changes to a script file.**

Click **Next>** on the Resolve Differences Dialog.

*The ALTER Script Warnings dialog appears.*

Click **Close**.

*The Complete Compare - Export*

*Changes dialog displays.*

*For more information about this  
dialog, press F1.*

Click **Next Step** >



## STEP 5: Review the Changes

Click **Start Export** button on the top right corner of the dialog.

*If an error condition occurs, processing will stop and the stop icon will display. Click the **Continue** button to proceed.*

Use the scrollbar to review the



change messages.

Click **Next Step** >



## **STEP 6 : Close the Model**

Click below to verify that you have performed the activity correctly, then

click

In **Complete Compare - Export Changes** dialog, click the **Finish** button.

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save**  
**as .ER1 file** then click **OK**.

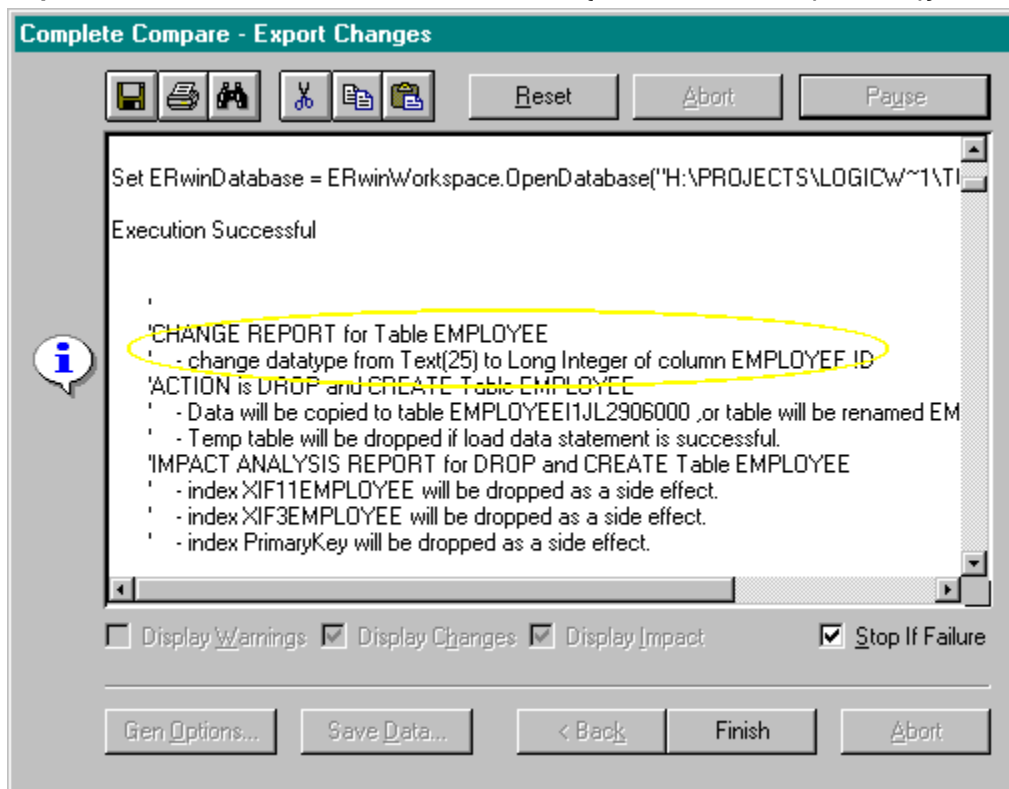
To close without saving, select **Close**  
**without saving** then click **OK**.

Click below to close this Try It card.

**- The End -**

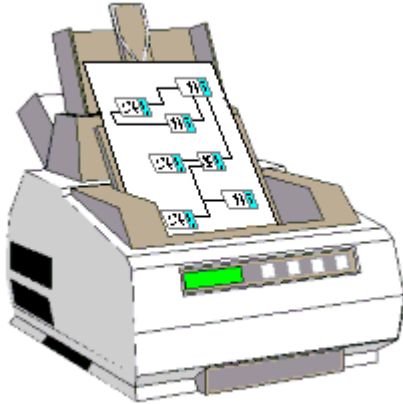


**Export a Column and Delete a Table Check** {button Close,CW("check")}





## Activity 16: Printing a Data Model {ewc HLP25632, HLP256\_Tile, banner.bmp}

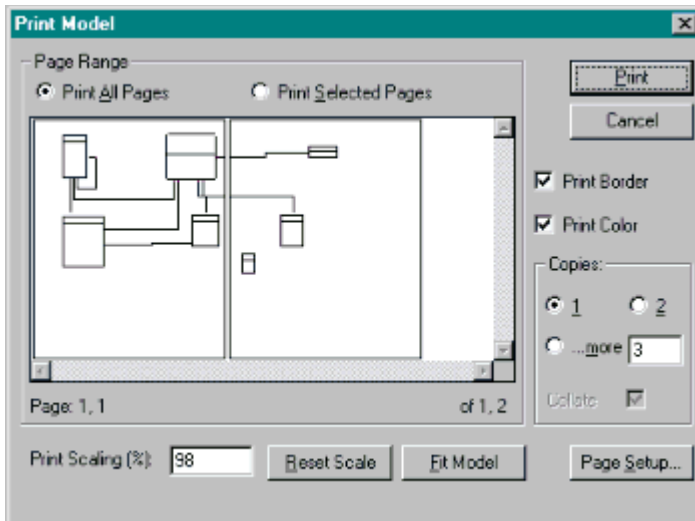


### Printing Your Model

Once you have completed your data model, you may want to print it for presentation purposes.



### Activity 16: Printing a Data Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Print Model Dialog

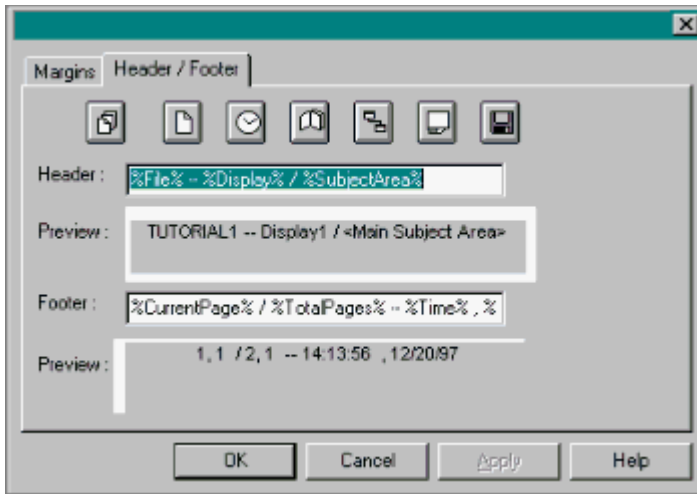
When you select the Print option on the File menu, ERwin displays the Print Model dialog. The Print Model dialog includes an embedded diagram window that lets you view the data model by page.

You can change:

- Print scaling.
- Number of pages you want to print.
- Format options, such as borders and color printing.



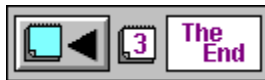
## Activity 16: Printing a Data Model, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Customizing Headers and Footer

When you click the Page Setup Button, you can add a customized header and/or footer to all the pages of the printed diagram. For example, you might want to print the model name, subject area, and stored display at the top of every page and the page number, date, and time at the bottom of every page.

If you would like to print a data model, click the **Try It** button below.





## Printing a Data Model Try It

In this exercise, you will practice customizing the header and footer of a printed diagram before printing.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\16-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...

Click **Next Step >** to continue.





## STEP 1: Open the Print Model dialog

On the **File** menu, choose **Print...**

*ERwin opens the Print Model dialog.*

Click **Next Step >**




## STEP 2: Create the header for the diagram

Click the **Page Setup...** button at the bottom right corner of the screen.

*ERwin opens the Page Setup dialog.*

Click the **Header/Footer** tab.

Delete the default text in the **Header** text box.

Click the Header text box, then click the **Date** button .

*In the Preview box, ERwin displays the date in the format that will appear in the header.*


Click **Next Step >**





### STEP 3: Create the footer for the diagram

Delete the default text in the **Footer** text box.

Click the **Footer** text box, then click the **Filename** button .

*In the Preview box, ERwin displays the filename in the format that will*



*appear in the footer.*

Click the **Apply** button.

Click **OK**.

Click **Next Step >**





#### STEP 4: Print the data model

Verify that the following are selected:

- **Print All Pages**
- **1** in the Copies group box

*For more information about this dialog, press F1.*

Click the **Print** button.

*ERwin starts to print immediately.  
ERwin temporarily changes each  
page color to yellow in the Print  
Model dialog to let you know the  
page is printing.*

Click **Next Step** >



## STEP 5: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.



To close without saving, select **Close without saving** then click **OK**.

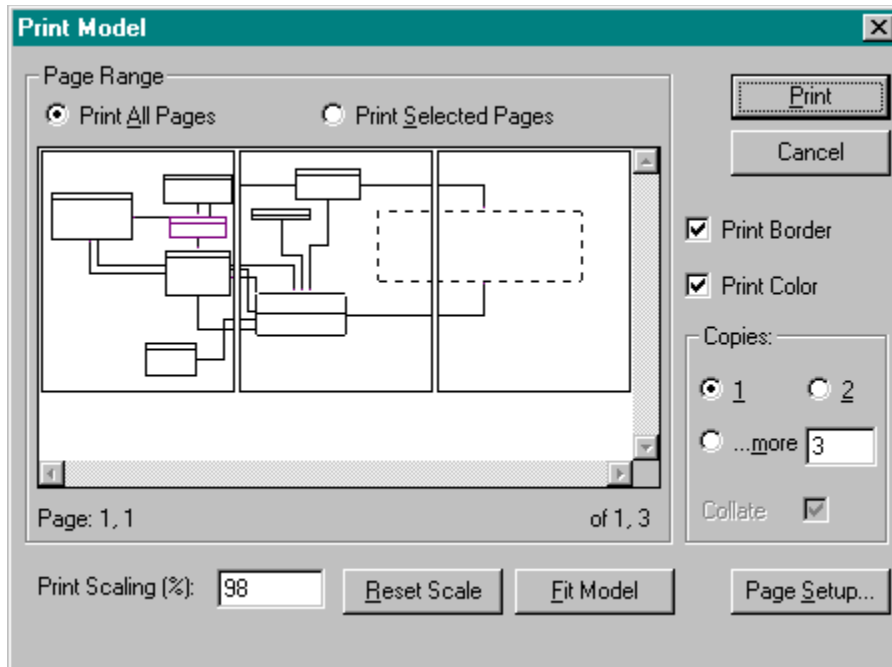
Click below to close this Try It card.

**- The End -**



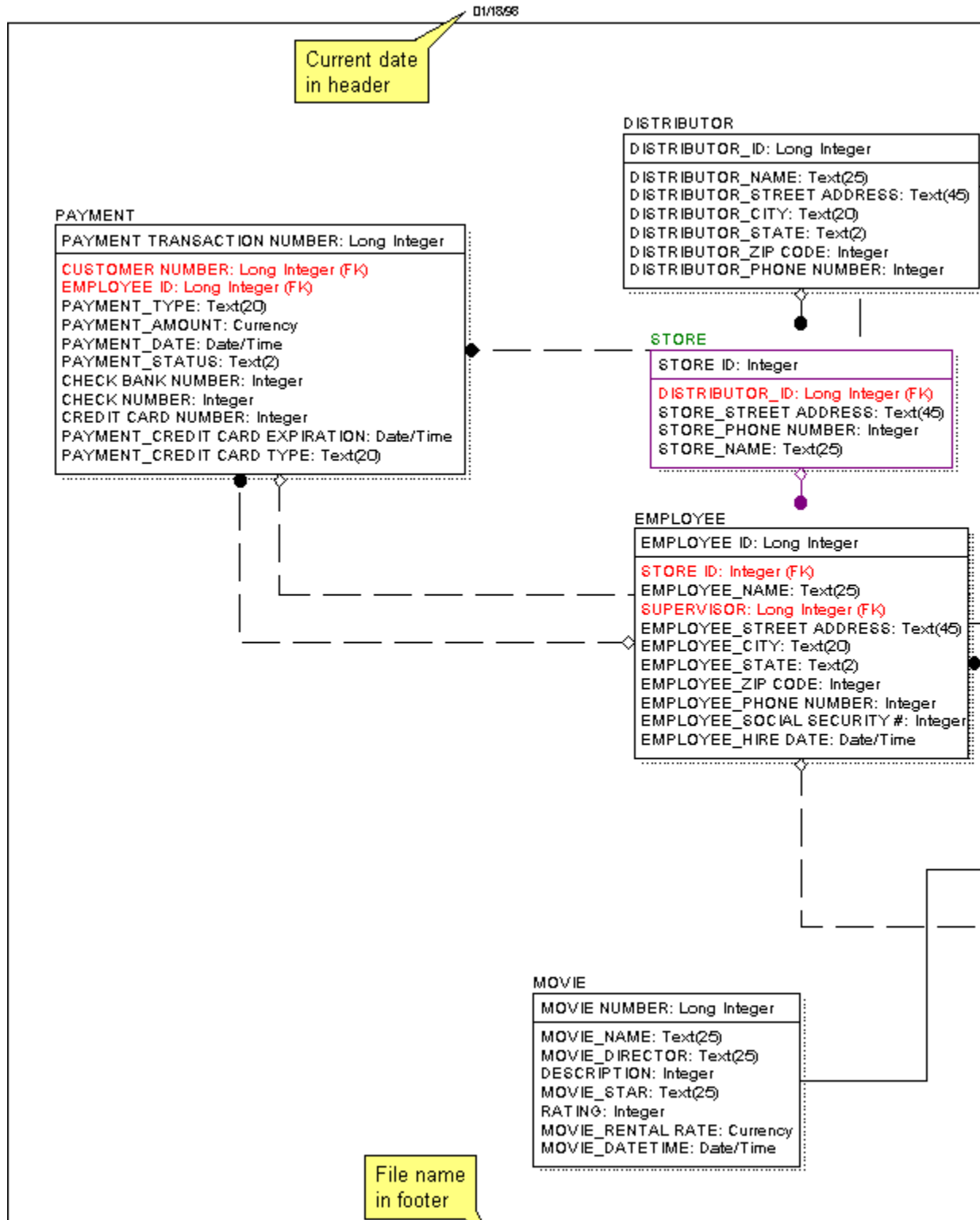


Print Data Model Check {button > Page 2,JI('ERTUT35.HLP','PRINTDM\_TRYIT\_1\_Check\_2')}  
{button Close,CW("check")}

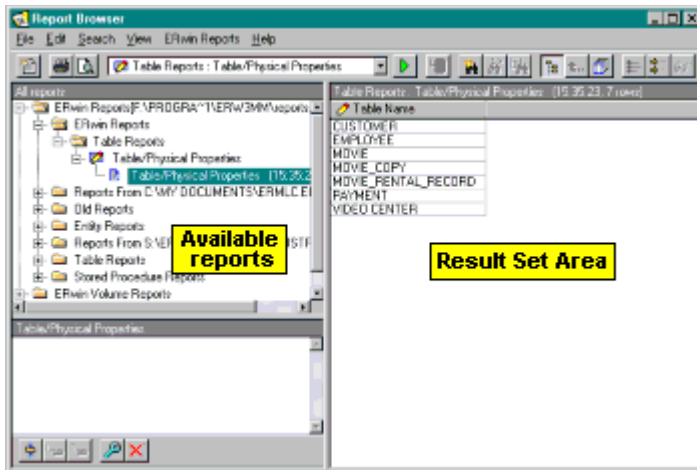


Print Data Model Check  
{button Close,CW("check")}

{button < Page 1,JI('ERTUT35.HLP','PRINTDM\_TRYIT\_1\_Check')}



## Activity 17 : Defining and Running Reports {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### ERwin Standard Reports

Once you have completed your data model, you may want to run reports against it. The Report Browser includes a set of standard reports that you can use to report on the active ERwin diagram.

To run one of the standard reports in the ERwin Reports folder, double-click on the report name. ERwin generates a report based on the active diagram, and returns the result set to the Result Set Area on the right side of the Browser.



## ERwin Report Browser

Report Browser

File Edit Search View ERwin Reports Help

Table Reports : Table/Physical Properties

All reports

- ERwin Reports[F:\PROGRA~1\ERW3MM\reports]
  - ERwin Reports
    - Table Reports
      - Table/Physical Properties (15:35:23)
  - Reports From C:\MY DOCUMENTS\ERMLC.EI
  - Old Reports
  - Entity Reports
  - Reports From S:\ERWIN3\DIAGRAMS\JUSTF
  - Table Reports
  - Stored Procedure Reports
  - ERwin Volume Reports

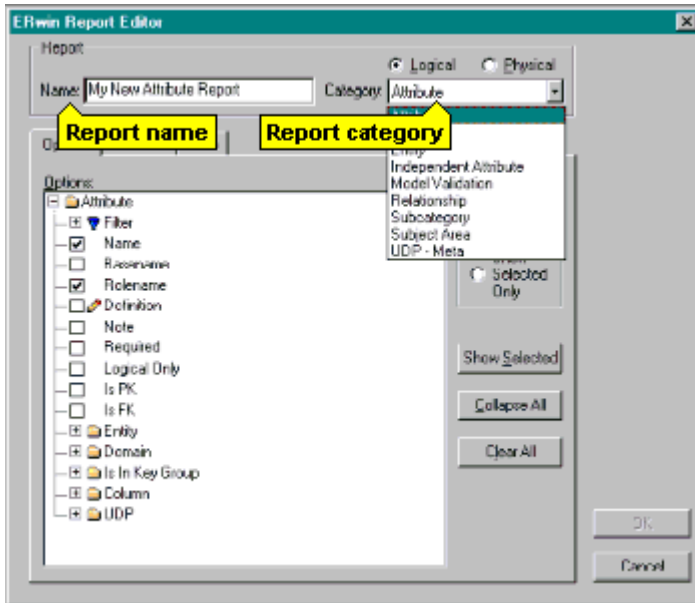
Table Reports : Table/Physical Properties (15:35:23, 7 rows)

Table Name
CUSTOMER
EMPLOYEE
MOVIE
MOVIE_COPY
MOVIE_RENTAL_RECORD
PAYMENT
VIDEO CENTER

Table/Physical Properties



### Activity 17 : Defining and Running Report, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

### Creating Customized Report - Report Categories

You can also create a customized report using the ERwin Report Editor. In this dialog, you specify the name and the category of the report. Some examples of report categories are:

- Attribute
- Column
- Diagram
- Entity
- Relationship
- Subject Area
- Table
- Validation
- View

Choosing a report category is like choosing a report template. You can filter the display of report categories for the logical and physical model.



## ERwin Report Editor

**ERwin Report Editor** [X]

Report

☒ Logical ☐ Physical

Name:  Category:

Options | Definition | Note

Options:

- ☒ Attribute
  - ☐ Filter
  - ☒ Name
  - ☐ Basename
  - ☒ Rolename
  - ☒ Definition
  - ☐ Note
  - ☐ Required
  - ☐ Logical Only
  - ☐ Is PK
  - ☐ Is FK
- ☐ Entity
- ☐ Domain
- ☐ Is In Key Group
- ☐ Column
- ☐ UDP

Options

☒ Edit

☐ Show Selected Only

Show Selected

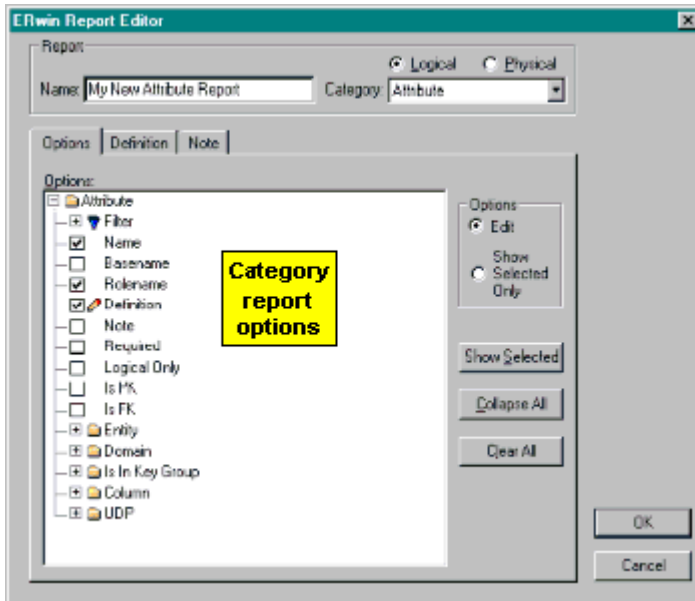
Collapse All

Clear All

OK

Cancel

## Activity 17 : Defining and Running Report, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

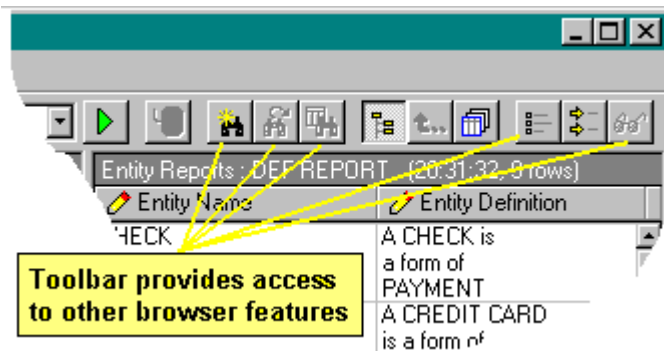
### Report Options

Each report category has a predefined set of options that you can use to include or exclude items in your report.



## ERwin Report Editor

**Activity 17 : Defining and Running Report, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



### Other Report Browser Features

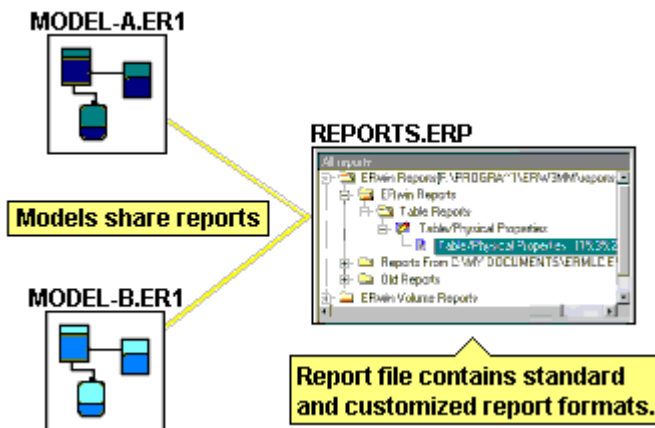
Once you generate a result set, you can use the Report Browser's extensive range of features to:

- Print the result set.
- Customize the appearance of the reports.
- Change the order of the columns.
- Sort the results set on one or more columns.
- Find a change in value.
- Find text in the results set.
- Hide result sets rows that do not match a specific criteria.





**Activity 17 : Defining and Running Report, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**

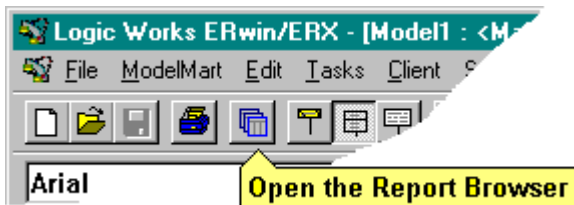


### **Report Formats as Separate Files**

ERwin saves standard and customized formats as a separate file, rather than as part of the ERwin diagram. As a result, the same set of standard reports and any customized reports that you have created are available for any diagram open in ERwin.



**Activity 17 : Defining and Running Report, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}**



### **Accessing the Report Browser**

In a few minutes you will run a report, but first let's review how to access the Report Browser dialog.

To open the Report Browser, click the button on the ERwin toolbar. A model must be open before running a report.

If you would like to define a report and run it against the sample model, click the **Try It** button below.





## Running Reports Try It

In this two part exercise, you will practice:

- 1) Running a standard report
- 2) Creating a customized report

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\17-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.


You may need to rearrange the objects in the workplace...

You must have a data model open  
before running a report.  
Click **Next Step >** to continue.





## STEP 1: Open the Report Browser

Click the **Report** button  on the ERwin toolbar.

Click **Next Step >**




## STEP 2: Run a Standard Report

Double-click on **ERwin Reports / Table Reports**.

You must click on the **+** sign to open a folder and locate a report file.

Double-click on **Table/Physical Properties**


*ERwin generates a report based on the sample diagram that you have open, and returns the results to the Result Sets Area on the right of the browser. The result sets icon  is also displayed under the report in the tree control.*

Click **Next Step >**





### STEP 3: Add Column Option and Sort

Click on **Format Options**  in upper right corner.

On the Column Options tab, check **Table Name**, **Table Column Name**, and **Table Column Datatype**.



On the Sort Tab, check **Table Name**.

Click **Next Step >**



#### STEP 4: Save this Sort View

Click **Save...** button.

Type MyView.

Click **OK**.

Click **OK** to close the Report Format dialog.

*Report is displayed alphabetically  
by Table Name.*


Click **Next Step >**





## **STEP 5: Print the Result Set**

Click  below to verify that you have

performed the activity correctly, then click  .

ERwin lets you print your result set or save it as a file.

Select the Result Set called **MyView**.



Click the **Print Result Set** button



On the ERwin Reports Menu, select  
**Save Report File As...**

Save it as `MyReport.erp`.

**- Finished with Part 1 -**

Click `below` to continue.





## Run a Standard Report Check

{button Close,CW("check")}

**Report Browser**

File Edit Search View ERwin Reports Help

Table Reports : Table/Physical Properties

All reports

- ERwin Reports[C:\ERW\3MM\MyReport.erp]
  - ERwin Reports
    - Table Reports
      - Table/Physical Properties (tutorial17.er1, 0
        - MyView
          - MyView (tutorial17.er1, 08:39:07, 8 row
  - Reports From C:\MY DOCUMENTS\ERMLC.ER1
  - Old Reports
  - Entity Reports
  - Reports From S:\ERWIN3\DIAGRAMS\JUSTFO2.ER1
  - Diagram Reports
  - Table Reports

MyView

Table Reports : Table/Physical Properties (tutorial17.er1, 08:44:59, 63 rows)

Table Name	Table Column Name	Table Column Data Type
CUSTOMER	CUSTOMER_NUMBER	Long Integer
	CUSTOMER_NAME	Text(255)
	CUSTOMER_STREET	Text(40)
	ADDRESS	
	CUSTOMER_CITY	Text(255)
	CUSTOMER_STATE	Text(255)
	CUSTOMER_ZIP	Integer
	CODE	
	CUSTOMER_PHONE	Integer
	NUMBER	
DISTRIBUTOR	DISTRIBUTOR_ID	Long Integer
	DISTRIBUTOR_NAME	Text(255)
	DISTRIBUTOR_STREET	Text(40)
	ADDRESS	
	DISTRIBUTOR_CITY	Text(255)
	DISTRIBUTOR_STATE	Text(255)
	DISTRIBUTOR_ZIP	Integer
	CODE	
	DISTRIBUTOR_PHONE	Integer
	NUMBER	
EMPLOYEE	EMPLOYEE ID	Long Integer
	STORE ID	Integer
	EMPLOYEE_NAME	Text(255)
	SUPERVISOR	Long Integer
	EMPLOYEE_STREET	Text(40)
	ADDRESS	
	EMPLOYEE_CITY	Text(255)
	EMPLOYEE_STATE	Text(255)
	EMPLOYEE_ZIP	Integer
	CODE	

## Running Reports Try It

In this part of the Try It you will create a customized report.

Click **Next Step >** to continue.



## STEP 1: Name the Report

On the Report Browser File menu,  
choose **New ERwin Report...**

On the **ERwin Report Editor**  
dialog, type the Name `MyReport`.

[Click Next Step >](#)







## STEP 2: Choose the Report Category

Choose the Report Category **Entity**.

In the Options List, check the **Name** and **Type** options for Entity.

Open the Attribute folder.

Check **Name** and **Definition**.

Click **OK** to return to the Report  
Browser.

Click **Next Step >**



### STEP 3: Run your Customized Report

Open the folder for Entity Reports.

Double-click on the report

#### **MyReport.**

*ERwin generates a report based on the sample diagram that you have open, and returns the results to the Result Sets Area on the right of the*

*browser.*

Click **Next Step** >





#### **STEP 4: Close the Model**

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

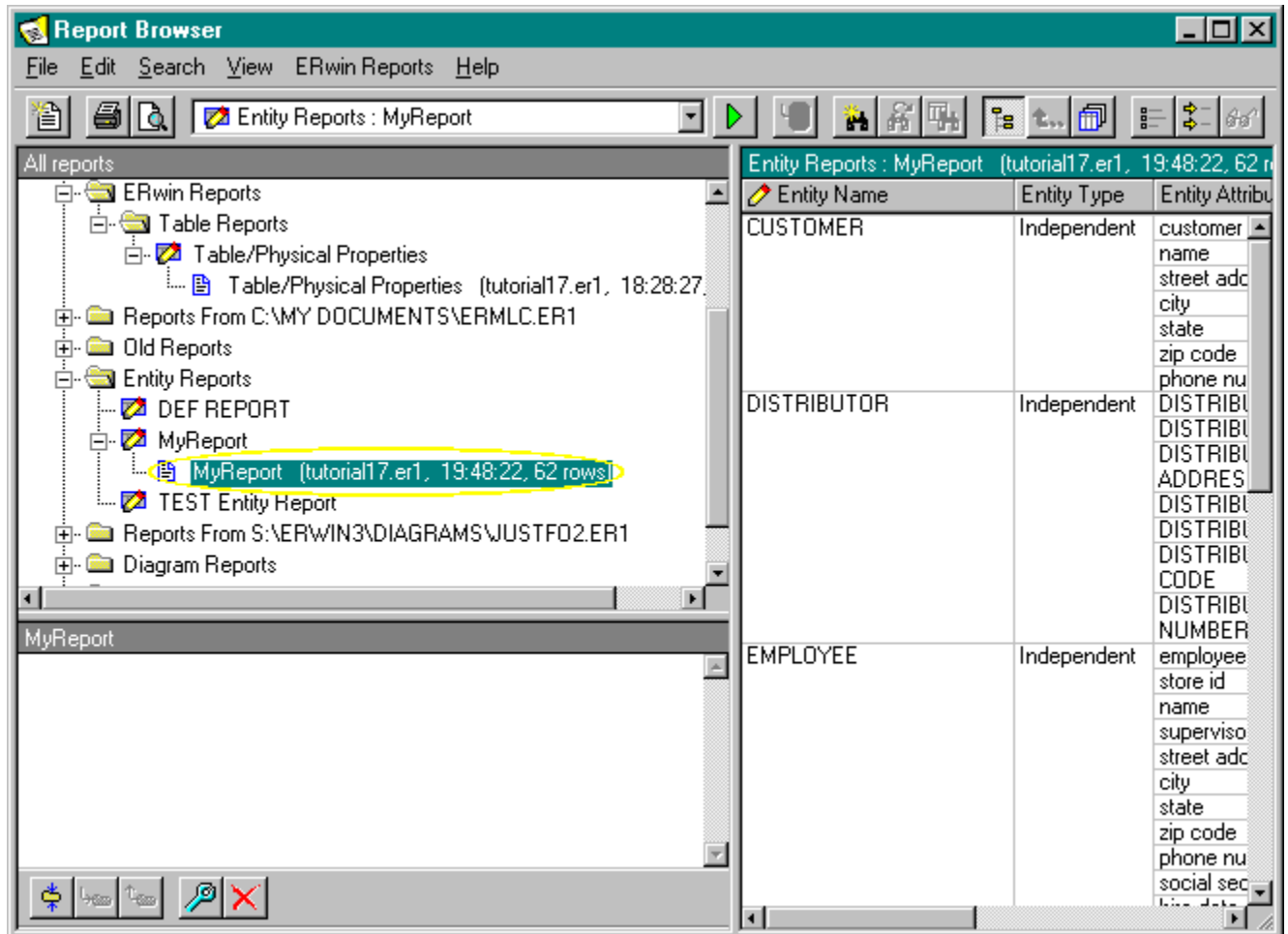
Click below to close this Try It card.

**- The End -**





Create a Customized Report Check {button Close,CW("check")}



**Activity 18: Tutorial Summary: Congratulations {ewc HLP25632, HLP256\_Tile, banner.bmp}**



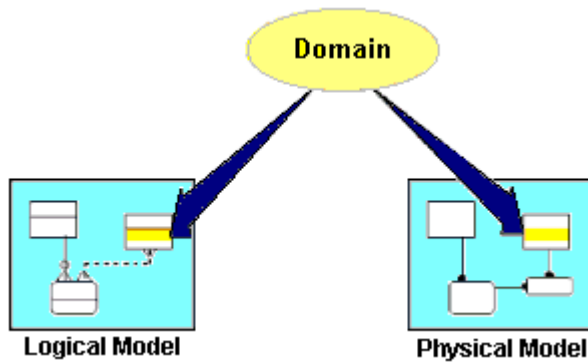
## **Congratulations!**

You have completed the ERwin Online Tutorial. This tutorial serves as an overview of ERwin's powerful features. Logic Works provides a complete curriculum of classroom training to assist you in becoming a proficient ERwin user.

Check our web site  
[www.logicworks.com](http://www.logicworks.com)  
(Education/Consulting section) for  
an up-to-date workshop schedule.



## Activity 5: Defining Domains {ewc HLP25632, HLP256\_Tile, banner.bmp}



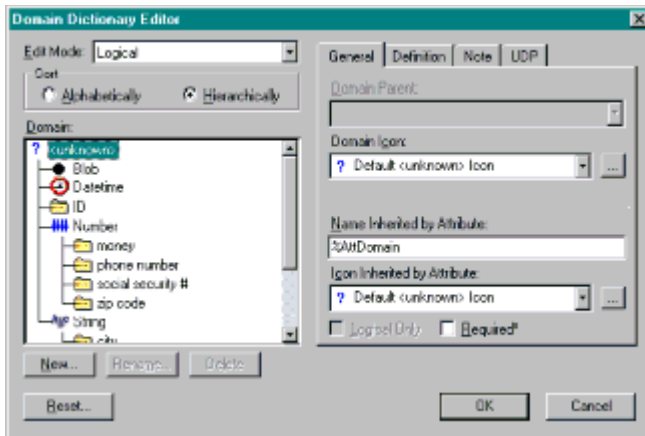
### What is a Domain?

In ERwin, a **domain** is an independent model object that you can use to quickly assign properties to an attribute or column. After you define a domain, you can reuse it in both the logical and physical model to:

- promote consistency
- reduce the time spent on development and maintenance (i.e. if you change the domain, all columns associated with the domain are also changed)
- support user-defined datatypes
- define independent attributes and columns for reuse throughout a data model



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



Click on the figure for a closer view.

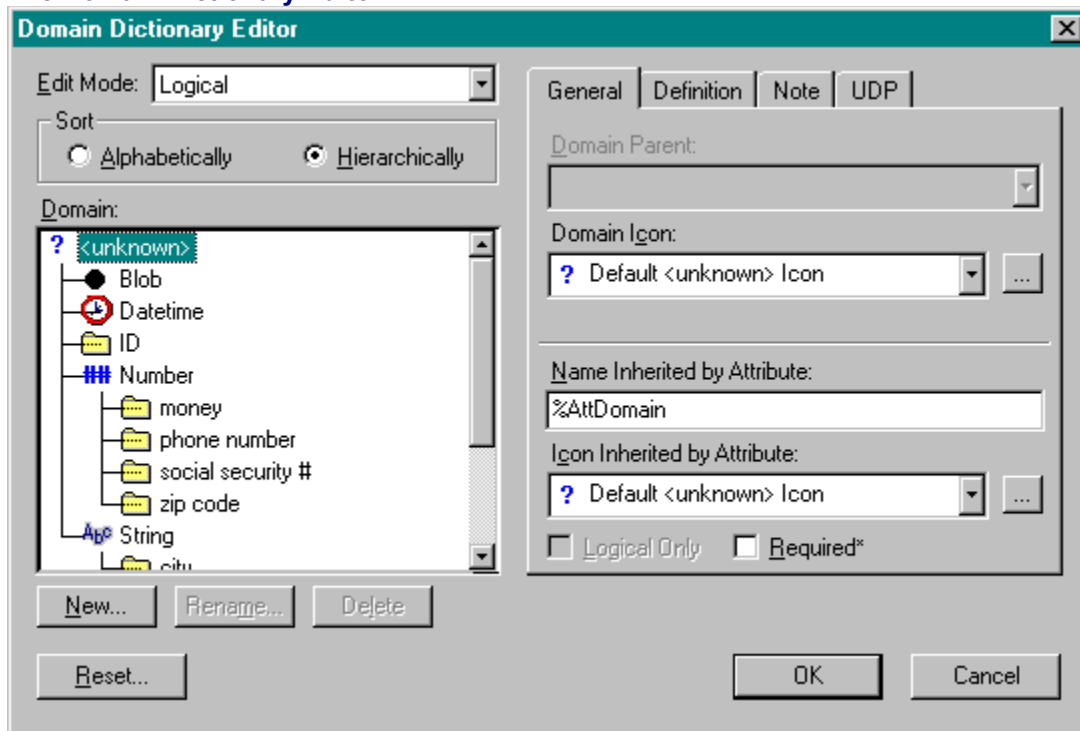
### Domain Dictionary Editor

You can create and modify both the physical and logical properties of a domain using the **Domain Dictionary Editor**.

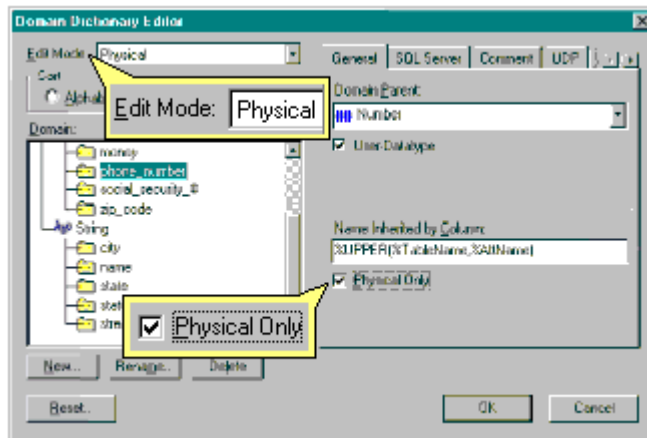
When you create a domain, it inherits all defined properties of the parent domain. The parent domain can be any of the standard ERwin domains or any domain you define.



## The Domain Dictionary Editor



### Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



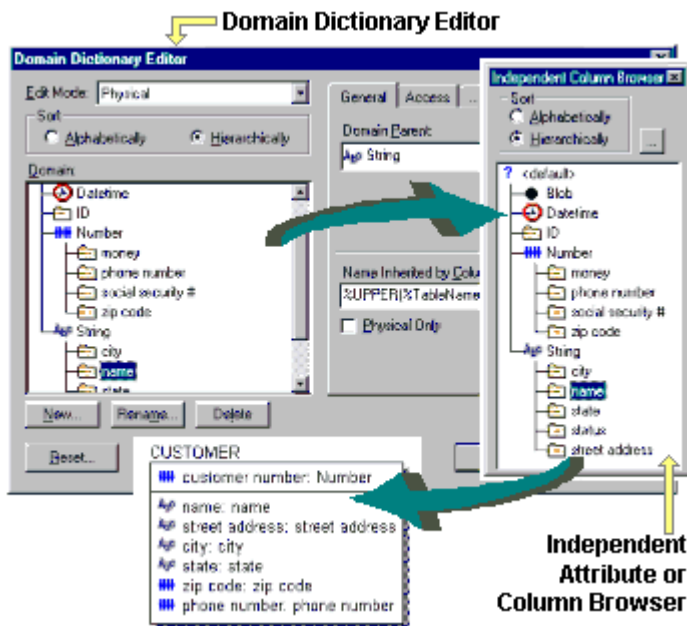
#### Domain Types

Domains can be defined as **Logical Only**, **Physical Only**, or both **Logical and Physical**.

- Logical Only - Domains can be assigned, updated, and deleted in the logical model only.
- Physical Only- Domains can be assigned, updated, and deleted in the physical model only.
- Both Logical and Physical- Domains can be assigned, updated, and deleted in both the physical and logical models. This type provides the greatest benefit when the physical model is developed from the logical model created in ERwin. Domains can be assigned in the logical model that provide the correct set of properties for the corresponding column in the physical model.



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Independent Attributes and Independent Columns

After you create a domain, ERwin adds it to a pool of **independent attributes** or **independent columns**.

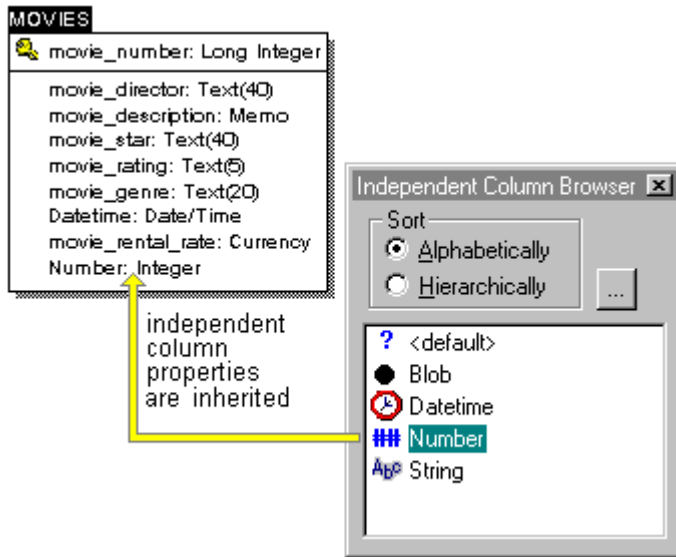
You can quickly add attributes to entities in the logical model using the Independent Attribute Browser. The Browser lists all domains defined for the logical model.

You can quickly add columns to the tables in the physical model using the Independent Column Browser. This Browser lists all domains defined for the physical model.





## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



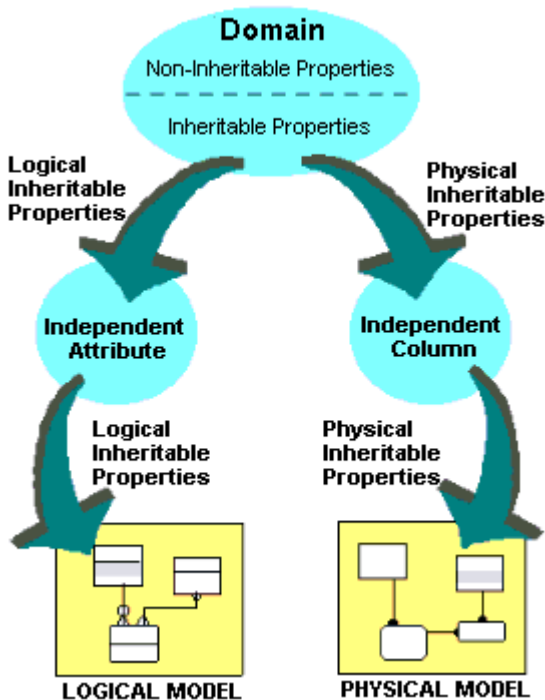
### Using the Browsers

Using a drag-and-drop operation from the Independent Attribute and Independent Column Browsers, you can add an attribute selecting from the Browser and dragging it into the new entity. You can use the same method to create a new column in a table. The new attribute or column inherits all the properties you defined in the Domain Dictionary Editor.

In the example to your left, the column *Number* is added to the MOVIE table using the Independent Column Browser.



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



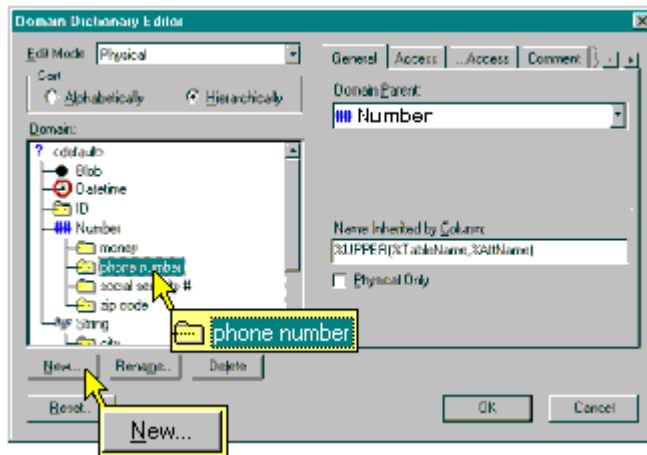
### Domain Inheritance and Overrides

In ERwin, domains have **non-inheritable** and **inheritable** properties. Non-inheritable properties do not migrate to the independent attributes or columns, while inheritable properties do migrate to child domains and to the attributes and columns associated with the domain.

When creating a new domain in the Domain Dictionary Editor, by default all the parent's inheritable properties migrate to its child domain. You can override any of the properties inherited by the child domain in the Domain Dictionary Editor.



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### User-Defined Domains

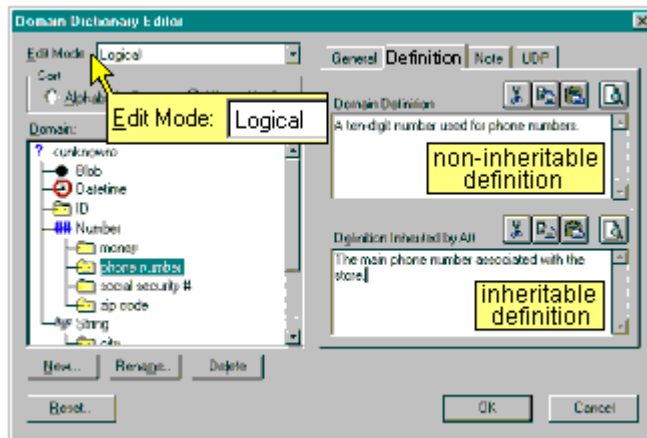
In addition to the five domains that are available in every ERwin model (blob, datetime, number, string, and unknown), you can also create your own **user-defined domains**. By default, they are represented by a folder but you can choose a different icon to represent the domains you create.

To create a domain, simply click on the New button in the Domain Dictionary Editor. Each new domain must be a child of an existing domain, which means it will automatically inherit its parents properties. The child of <default> or <unknown> inherits no properties.

In the example to the left, a new domain was added called “phone number”, which is a child of the parent domain, Number.



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Definition Tab

You use the **tabs** on the right side of the dialog to define domain properties. ERwin displays different tabs based on the edit mode you select.

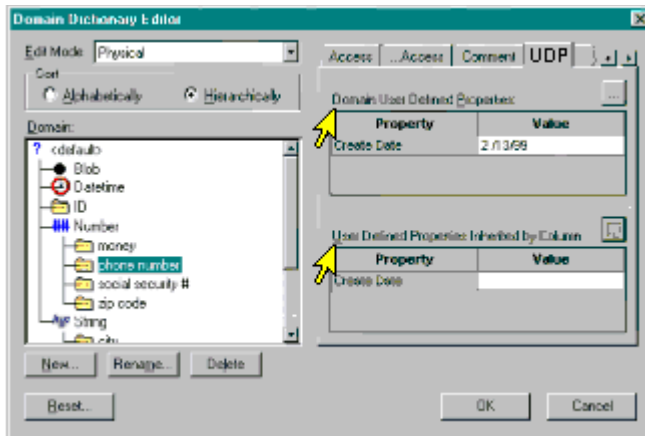
For example, in the logical edit mode you can use the Definition tab to enter or modify the inheritable and non-inheritable definition for the domain. A definition included in the logical model is automatically copied to the definition in the corresponding physical model.

The top half of this tab specifies a non-inheritable definition for the highlighted domain. Whereas, the bottom half specifies the inheritable definition.





## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### UDP Tab

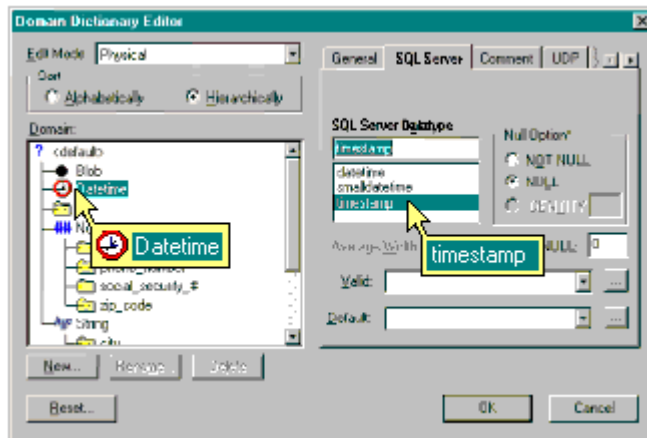
In the logical and physical edit modes, you can use the UDP tab to define the inheritable and non-inheritable user-defined properties.

The top half of this tab specifies non-inheritable UDPs for the highlighted domain. Whereas, the bottom half of this tab specifies UDPs inherited by:

- Attribute when in logical edit mode
- Column when in physical edit mode



## Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



### Modifying Domain Properties

For your particular model, you may need to define or modify some domain properties. You can use the Domain Dictionary Editor to modify:

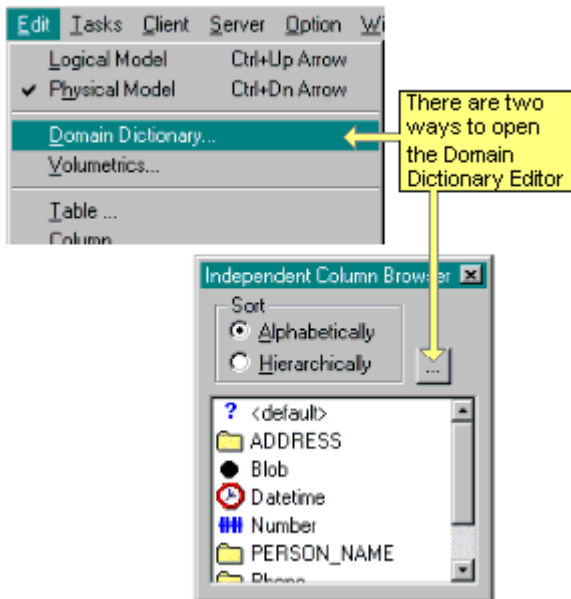
- domains provided by ERwin
- user-defined domains

In the example to your left, the Datetime property is being changed from datetime to timestamp.

Now, when you drag and drop Datetime using the Independent Column Browser, the newly added column will inherit the updated properties you assigned in the Domain Dictionary Editor. It is that easy!



### Activity 5: Defining Domains, cont. {ewc HLP25632, HLP256\_Tile, banner.bmp}



#### Edit Menu- Domain Dictionary

To define a new independent attribute or independent column, or modify properties of an existing one, you will choose **Domain Dictionary** on the Edit Menu. You can also open the Domain Dictionary Editor from the browser as shown in the figure.

If you would like to create a user-defined independent attribute, click on the **Try It** button below.







## Defining Domains Try It

In this exercise, you will practice adding a new independent attribute.

To begin, you will open a model which has been provided for this tutorial activity.

Select the model  
[path\05-tutor.er1](#)  
from the **File|Open...** menu.

Click **OK** on the “read-only” message box.

You may need to rearrange the objects in the workplace...  
Click **Next Step >** to continue.



## STEP 1: Open the Domain Dictionary Editor

On the **Edit** menu, choose **Domain Dictionary**.

Locate the **Edit Mode** field in the top left corner of the Domain Dictionary dialog, and check that you are in the **Logical** Edit Mode.

Check that **Hierarchically** is  
selected in the Sort box.

Click **Next Step >**



**STEP 2: Name the new independent attribute**

Click **New...**

Select the parent domain - **String**.

Type `email` in the Logical and Physical textboxes.

Click **OK**.

*A yellow folder labeled email is added to the Domain list.*


Click **Next Step >**







### STEP 3: Choose a domain icon

In the General Tab, click in the Domain Icon box and select the  **Default String Icon**.

Click **Next Step >**



#### STEP 4: Add attribute definition

Click the **Definition** tab.

Type email address for each employee in the **Definition Inherited by Att** text box.


[Click Next Step >](#)






## STEP 5: Add non-inheritable user-defined properties

Click the **UDP** tab


Locate the top text box Domain User  
Defined Properties and click the ellipsis  
button .



Click on the plus sign  to add a UDP.

Type `Create Date` under the **Name** column.

Click in the **Type** column, select **Date** then click **OK**.

Click in the **Value** column and use the  button to select today's date from the calendar supplied.

Click **OK**.

Click **Next Step >**



## STEP 6: Close the Model

Click below to verify that you have performed the activity correctly, then

click .

On the ERwin **File** menu, choose **Close**.

To save your work, select **Save as .ER1 file** then click **OK**.

To close without saving, select **Close without saving** then click **OK**.

Click below to close this Try It card.

**- The End -**







**Defining Domains Check**      {button Close,CW("check")}

The Independent Attribute Browser will contain the newly added attribute *email* as shown here.

If the browser is not displayed, press Ctrl+B.

