

Database Development with Visio 2010 Professional Edition

Learning Objectives

This chapter provides instructions about using the Visio 2010 Professional Edition to create entity relationship diagrams (ERDs) and convert from a table design to an ERD. After this chapter, the student should have acquired the knowledge and skills to

- Understand the role of shapes, glue, stencils, and templates in creating diagrams.
- Create entity types, attributes, and relationships using the Entity Relationship stencil.
- Use the Entity Relationship stencil to create self-referencing relationships.
- Use the Entity Relationship stencil to create generalization hierarchies.
- Map relational databases into ERDs using the Reverse Engineering Wizard.

Overview

This lab chapter presents the data modeling features of Visio 2010 Professional Edition, a powerful and easy-to-use computer-aided software engineering (CASE) tool. The practice in this lab chapter complements the data modeling concepts in textbook Chapter 5 and the Visio documentation.

This chapter provides guided practice with creating entity relationship diagrams (ERDs) using the Entity Relationship stencil of Visio 2010 Professional Edition. You will use the stencil options that support the crow's foot notation, one of the most widely used notations for ERDs. To provide a context for the guided practice, sections 1 and 2 provide background about Visio and present important drawing concepts. Section 3 describes the auto repair database that is used through most of the guided practice exercises. The guided practice begins in Section 4, in which

you will create ERDs of increasing complexity. The practice continues in Section 5 with more specialized ERDs for self-referencing relationships and generalization hierarchies. Section 6 finishes the chapter with a presentation of the Reverse Engineering Wizard to convert from a table design to an ERD.

1 Visio Professional Background and Drawing Concepts

This section presents background about Visio 2010 Professional Edition as a CASE tool and defines important concepts for creating Visio diagrams. You will apply the concepts in later sections when you create ERDs.

1.1 Visio Professional as a CASE Tool

Visio 2010 Professional Edition is a multidiscipline CASE tool. It is used in various disciplines such as software development, network design, website design, and office layout. Visio Professional provides tools for engineering projects that require flowcharts or diagrams of system designs and processes.

Similar to other CASE tools, Visio 2010 Professional Edition has important features beyond its diagramming capabilities, as discussed in textbook Chapter 5. For database development, Visio Professional provides analysis functions for conversion and error checking. CASE tools also have their own database known as a data dictionary that stores properties of an ERD as well as text descriptions, assumptions, and notes. Visio Professional provides a small data dictionary to store the details of ERDs. Thus, Visio Professional provides the tools to assist with the conceptual data modeling phase and part of the logical database design phase described in textbook Chapter 5.

Visio Professional is an entry-level CASE tool. For large information systems projects, Microsoft provides the Visio Premium Edition, a more powerful product with additional drawing templates, intelligent validation of diagrams, Microsoft Sharepoint integration, and workflow support for team development. This chapter describes Visio Professional because it is an easy-to-use and powerful tool for introductory database courses. Additional chapters are necessary to describe the database development features of Visio Premium.

1.2 Drawing Concepts

Visio diagrams have a number of unique concepts summarized in Table 1. A Visio diagram contains shapes linked with connectors. In Figure 1, the drawing area contains shapes and connecting lines used to depict an organization chart. Shapes can have behaviors that perform tasks based on a user's actions. In Figure 1, shapes representing executives have a behavior to convert an executive into a new department. To save time when editing a diagram, the glue feature keeps shapes connected when repositioned. For example, if an executive shape is moved, connecting lines remain attached to other drawing objects. To position and align shapes precisely, the snap feature pulls shapes into position when moved and sized.

Table 1: Summary of Basic Drawing Concepts

Visio Concept	Meaning
Shape	Drawing object with a 2D representation and behaviors
Connector	Line or other symbol that links shapes
Glue	Feature that keeps shapes connected when moved
Snap	Feature that supports precise alignment of shapes
Template	A collection of stencils, preset styles, and a blank drawing page
Master	A shape appearing on a stencil; a shape on a diagram is an instance of a master
Stencil	A collection of master shapes (or masters) used in a specific kind of diagram

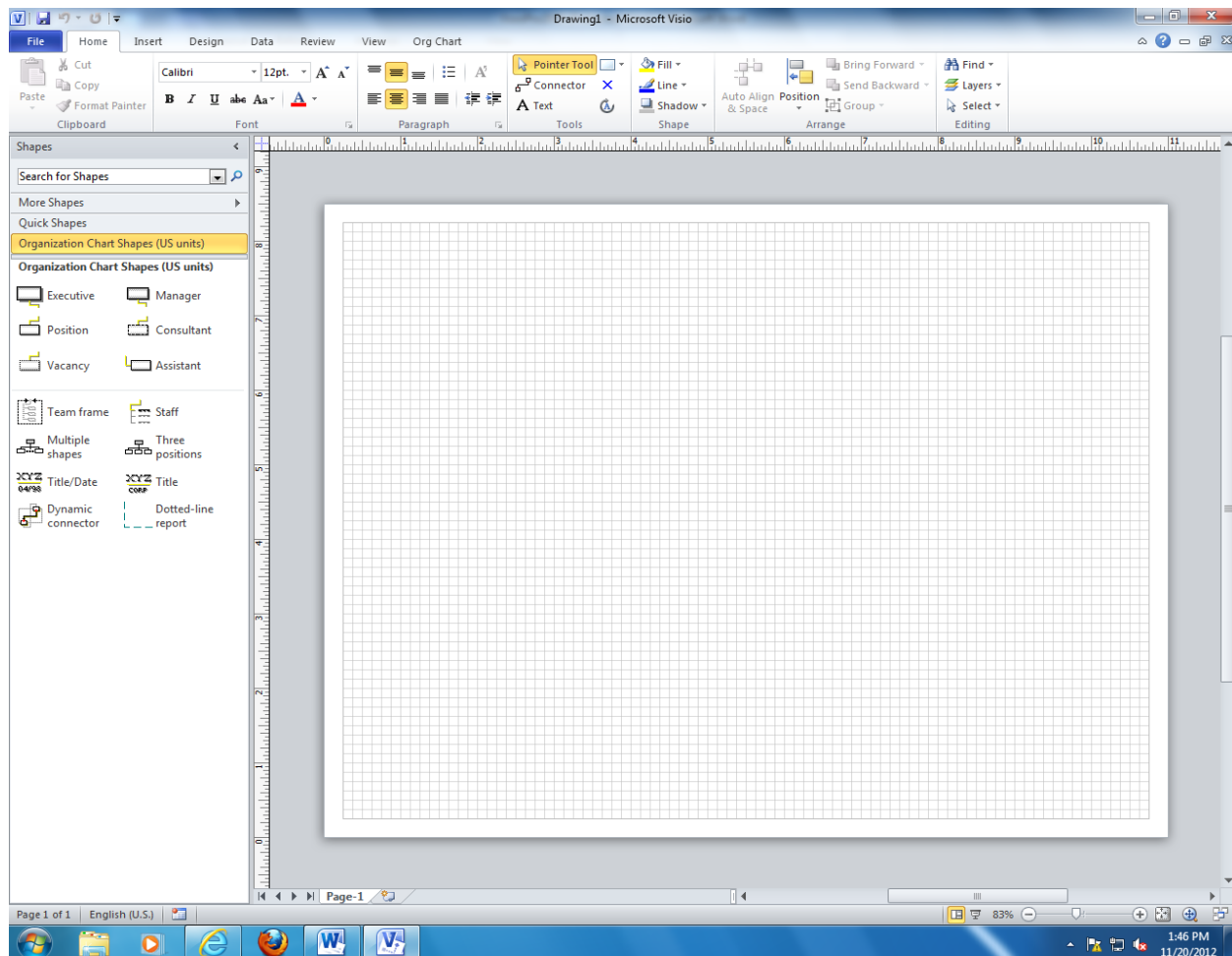


Figure 1: Visio Professional with a Template File Open

To facilitate creation of different kinds of diagrams, Visio Professional provides template files. A template file contains a collection of one or more stencils (collections of shapes), a blank drawing page, and preset styles for text, lines, and shapes. Figure 1 shows the template file for

organization charts. The stencil appears to the left of the drawing page. Similar to the toolbox in other Windows applications, shapes may be dragged from a tool area (the stencil) to the Visio drawing area. Stencils can be anchored to the left, right, top, or bottom of the drawing area or float. The stencil in Figure 1 is anchored to the left.

Since Visio Professional is used in many disciplines, you will see different template files when you open the software. You will see template files for organization charts, maps, web development, office layouts, project timelines, and many others. Although this chapter emphasizes template files for database diagrams, the range of other template files demonstrates the breadth of Visio Professional.

In addition to the unique features, Visio Professional provides the typical tools of most Windows applications. For example, the **File**, **Home**, **View**, **Insert**, and **Review** menus have many items similar to Microsoft Office programs. You also will find that the Visio drawing tools are similar to other drawing programs such as Microsoft PowerPoint for creating rectangles, circles, and lines. Like other drawing programs, it also has the ability to create “layers” of objects drawn on top of each other and then group these layers together as a unit.

2 Visio Professional Features for Database Development

Visio Professional supports database development through stencils and automation features. For data modeling, Visio Professional provides stencils for Entity Relationship diagrams and Object Relational diagrams. Figure 2 shows the list of database stencils using the More Shapes choice in the Shapes pane. Figure 3 shows the shape masters in each stencil. Each of these notations has a significant following among database professionals. In the Software folder, Visio Professional provides stencils for the Unified Modeling Language (UML), a widely

used notation for object-oriented modeling. In addition to the master shapes inside the stencils,

Visio Professional also has other drawing tools such as squares, circles, and lines.

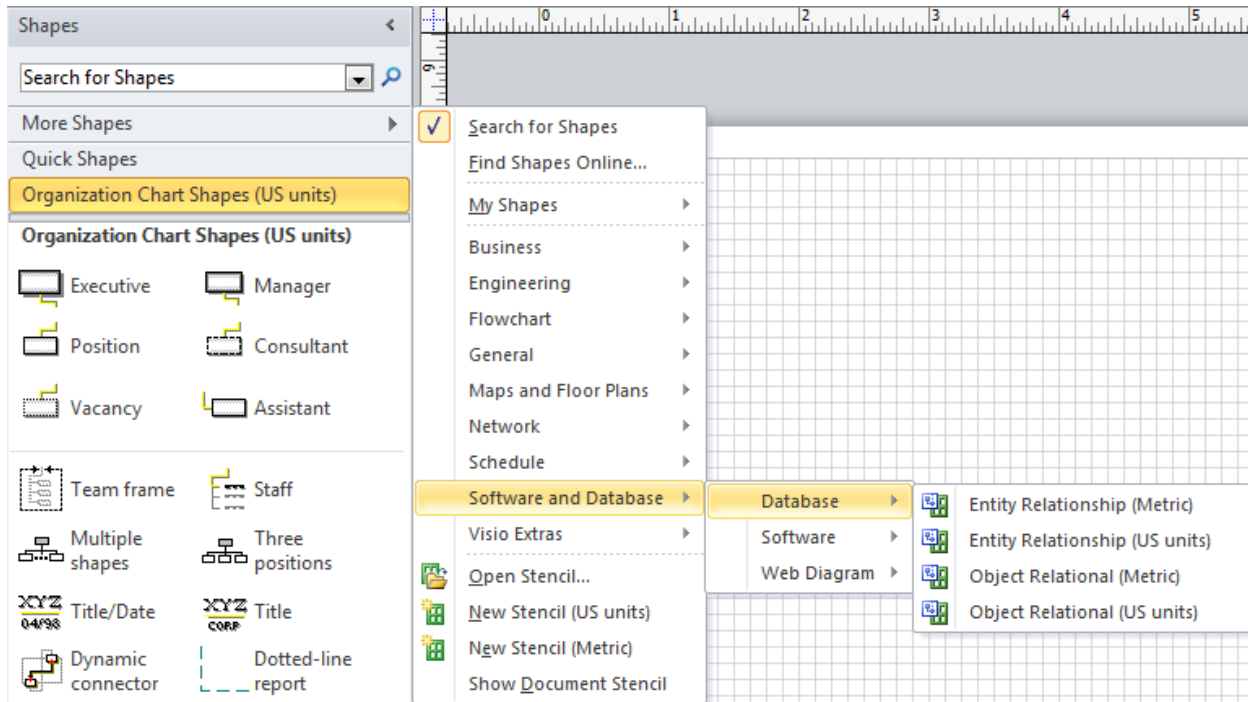
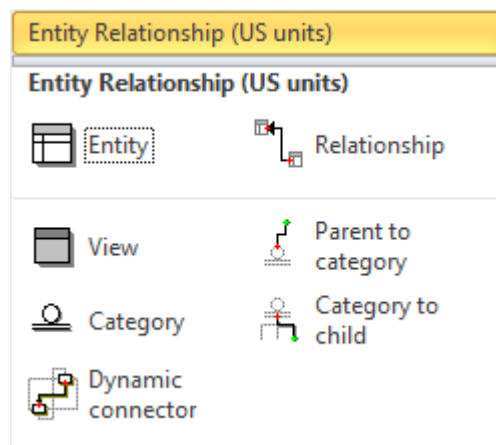


Figure 2: List of Data Modeling Stencils in the Database Folder



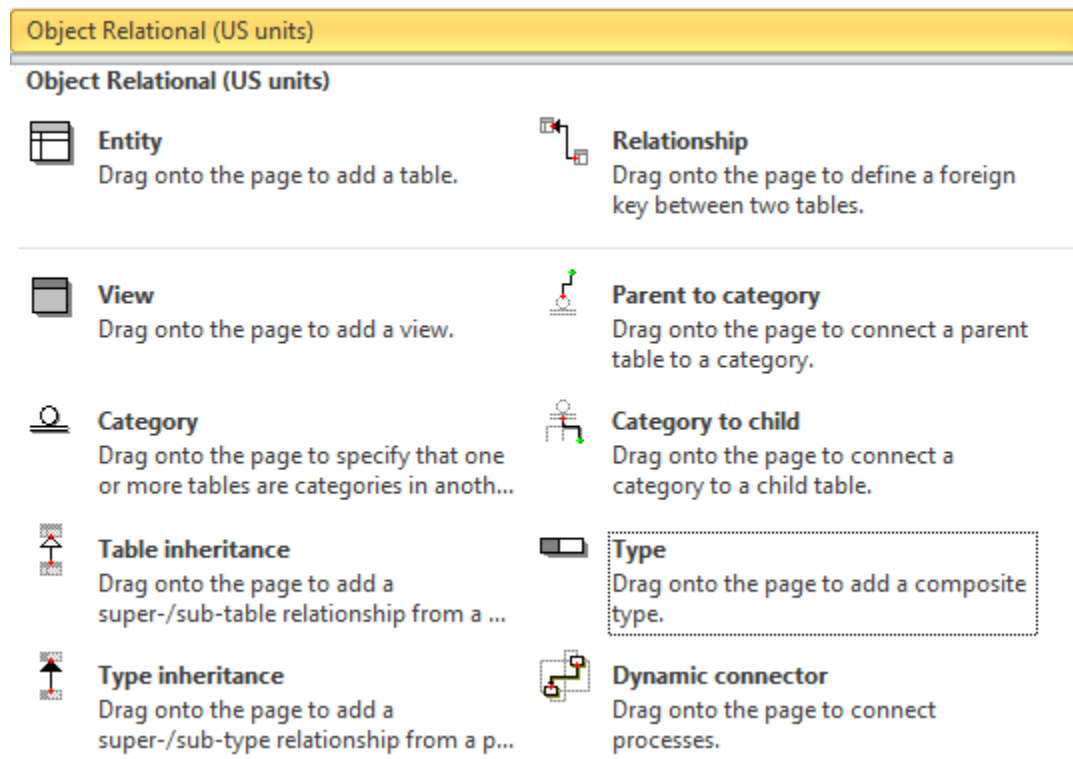


Figure 3: Master Shapes in Data Modeling Stencils

Visio Professional provides a data dictionary, definition windows, and automation features for the Entity Relationship and the Object Relational stencils in addition to the shapes. For the other data modeling stencils, Visio only provides the shapes. Because of these additional features, this chapter only presents the Entity Relationship stencil. The Object Relational stencil provides a few additional shapes and some additional flexibility in column data types than the Entity Relationship stencil. Otherwise, the two stencils are identical.

Visio Professional provides a number of automation features for the Entity Relationship and the Object Relational stencils, as summarized in Table 2. Most of these automation features are wizards that provide a sequence of windows to guide a user. Later sections will provide guided instruction about most of these automation features.

Table 2: Summary of Automation Features for Database Development

Feature	Location	Description
Reverse Engineering Wizard	Database → Reverse Engineer...	Creates an ERD from an existing relational database.
Refresh Model Wizard	Database → Refresh	Compares an ERD to a relational database. Can update the ERD to resolve conflicts.
Import Commands	Database → Import...	Convert an ERWin model or Visio Modeler into Visio 2010 format.
Lay Out Shapes	Home → Auto Align & Space	Automatically positions shapes in a diagram.

In addition to supporting database development, Visio Professional supports dynamic, data driven diagrams. Data from an external data source such as an Excel worksheet can be imported and linked to shapes in a diagram. Table 3 summarizes the tools to support dynamic, data driven diagrams.

Table 3: Summary of Tools for Dynamic, Data Driven Diagrams

Wizard	Location	Description
Automatic data linking	Data → Link Data to Shapes	Imports data from an external data source
Automatic data refresh	Data → Automatically Link	Links imported data to shapes in a diagram
Automatic data refresh	Data → Refresh All	Refreshes shapes bound to a data source according to a time interval
Data legend	Data → Insert Data Legend	Explains meaning of data-linked graphics

3 Overview of the Auto Repair Database

The auto repair database is used through most of this lab chapter. To help you become familiar with this database, this section describes the entity types and the relationships as well as the functions supported by the database.

3.1 Functions Supported by the Auto Repair Database

The auto repair database supports the operations of a small repair shop. Customers bring vehicles to repair. The repair shop completes a service order with a list of parts required to

perform the necessary work. As each part of a service order is completed, the number of parts used is recorded. When a service order is finished, the completion time and the total charge are noted. To support marketing efforts, a list of customers and vehicles is maintained. If a service order is for a new customer or vehicle, it is added to the database. To facilitate inventory control, a list of parts is maintained. Parts are usually added to the database if many service orders require parts not stocked by the repair shop.

3.2 Auto Repair Entities

The auto repair database contains five entity types. The entity types are described below with the entity type names italicized:

1. *Customer*: This entity type contains the customer's name, a unique customer number, the address, and the phone number.
2. *Vehicle*: This entity type contains the serial number of the car, its year and model, the license number, the state, the number of cylinders, and the customer number of its owner.
3. *Part*: This entity type contains the part description, the part number, the cost, the size, and the quantity in stock.
4. *RepairOrder*: This entity type contains data taken directly off a repair order form including the repair order number, the date, the car odometer reading, the car serial number, the time brought into the shop, and the time out of the shop.
5. *PartsUsed*: This associative entity type represents an M-N relationship between the *Part* and the *RepairOrder* entities. The entity type contains the repair order number, the part number, and the quantity used.

3.3 Auto Repair Relationships

The relationships support common queries about the entities, as shown in Figure 4. The *Owns* relationship shows the vehicles owned by a customer. The *Requires* relationship provides the repair orders performed for a vehicle. The identifying *Uses* and *UsedIn* relationships indicate the part requirements for a repair order.

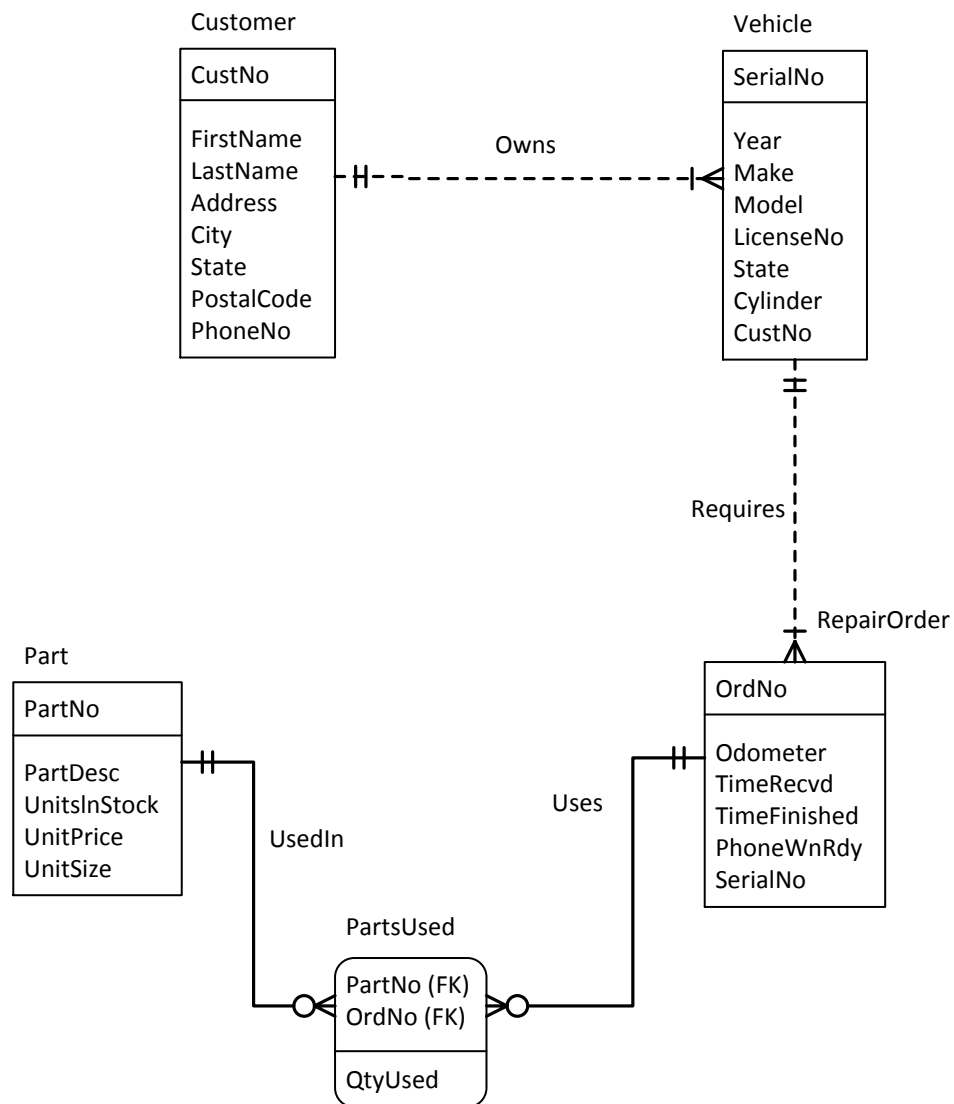


Figure 4: ERD for the Auto Repair Database

The crow's foot notation in Figure 4 differs slightly from the notation used in textbook Chapters 5 and 6. The textbook was written using the Crow's Foot stencil in a previous Visio Professional version. The current version, Visio 2010 Professional Edition, supports the IDEF1X notation with the option of using the crow's foot symbols for the relationship cardinalities. The IDEF1X notation does not support M-N relationships, attributes for relationships (typically useful only for M-N relationships), and uses different symbols for weak entity types and generalization hierarchies. Otherwise, the data modeling notation used in textbook Chapters 5 and 6 are identical to the notation supported by Visio 2010 Professional Edition.

4 Developing an ERD for the Auto Repair Database

This section provides guided instruction about creating an ERD for the auto repair database. You will create the entity types and relationships shown in Figure 4. It is assumed that you are familiar with the crow's foot notation from textbook Chapter 5. Note that the textbook refers to an entity as an "entity type". Because Visio Professional uses the term "entity," entity is used in this lab chapter.

Before you begin creating the ERD, you will open the Entity Relationship template. Then you will define the *Customer* and the *Vehicle* entities, add attributes, and define the relationship between the *Customer* and the *Vehicle* entities with cardinalities. Next, you will define the *RepairOrder* entity, add attributes, and define the relationship between the *RepairOrder* entity and the *Vehicle* entity. Finally, you will repeat this process for the *Part* and the *PartsUsed* entities and define relationships.

4.1 Starting Visio Professional

When you open Visio Professional, a window appears giving you the option of creating a new drawing or opening an existing file (Figure 5). If this is the first time using Visio Professional, you will not see any existing drawings. As shown in Figure 5, select “Database Model Diagram” and click the **OK** button to open the workspace. The Visio Professional workspace displays menus and toolbars at the top, stencils containing the drawing shapes on the left side, and a new, empty work area on the right side (Figure 6). Visio Professional opens with the Entity Relationship stencil displayed and the Object Relational stencil hidden. To become familiar with the Entity Relationship stencil, place the cursor over each master shape and read the explanation that appears in yellow. You also can practice dragging master shapes to the work area to see their appearance.

The default toolbars are the Standard toolbar (directly under the menus) and the Format toolbar (under the Standard toolbar). Later you can customize the desktop and the toolbars to your own preferences. For a summary about the Visio-specific toolbars, see Appendix A. You will use the **Connect Shapes** button from the Action toolbar in later practice. Next, point the cursor at the icons on the toolbars for a few seconds and a label appears telling you what the icon is for. You also may want to familiarize yourself with the menus at this time. Later, you will use some of the choices found in the **Database** tab (Figure 7).

You can refer to the help documentation under the **Help** menu or use the **Help** button



at the right end of the Standard toolbar. Just select it and point at anything about which you are unsure and Visio Professional will provide a definition or explanation.

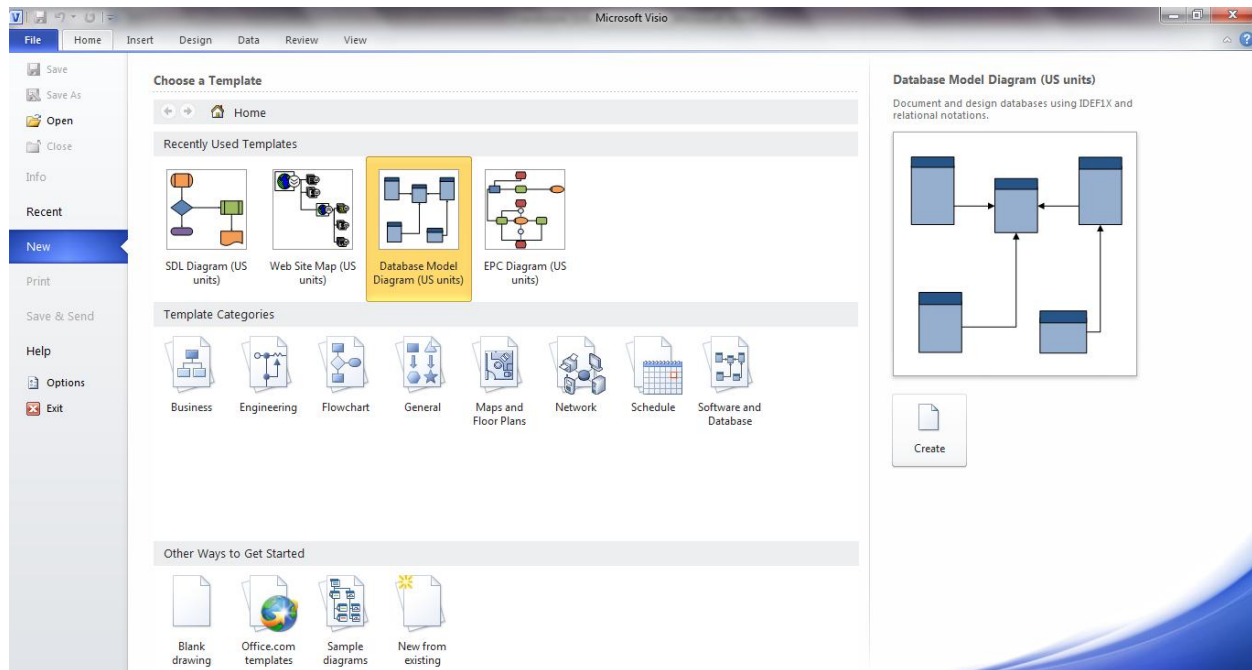


Figure 5: Initial Window when Visio Professional Opens

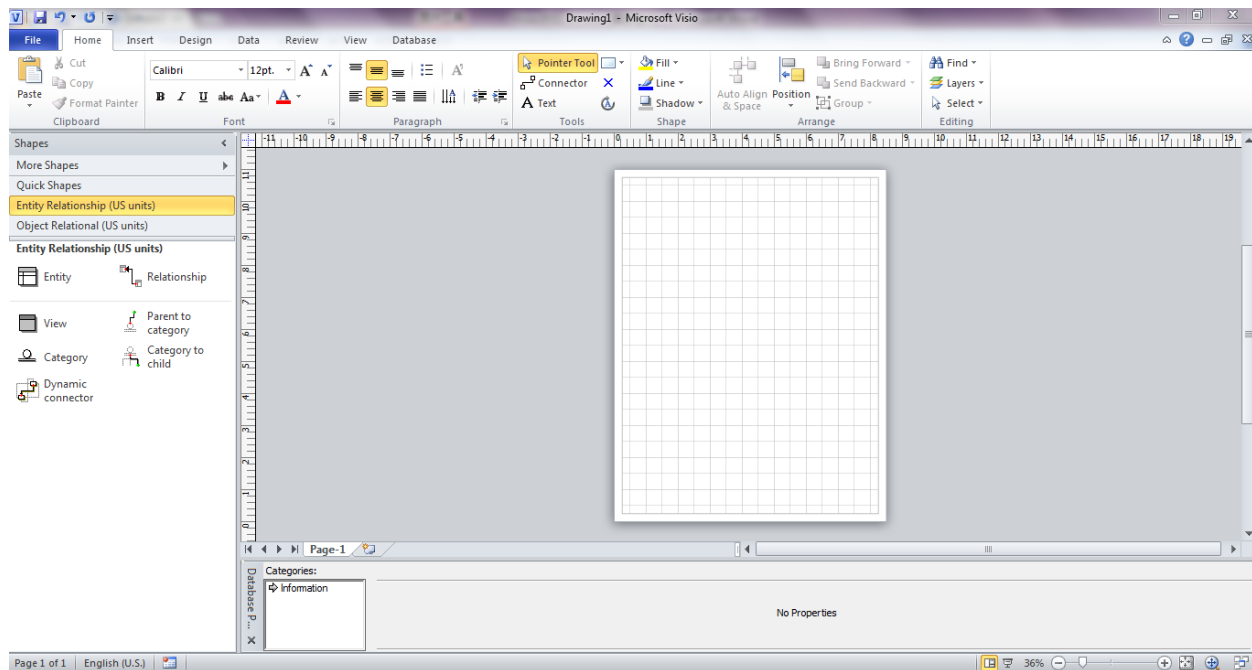


Figure 6: Initial Visio Workspace for an Entity Relationship Model Diagram

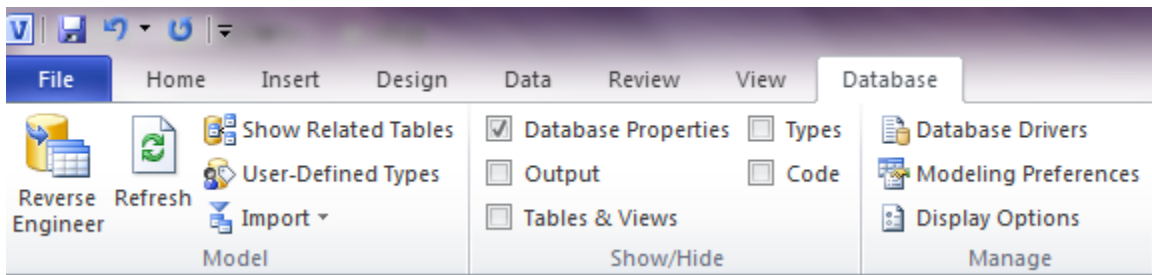


Figure 7: Database Menu Choices

4.2 Creating the Customer and the Vehicle Entities

You will first create the *Customer* entity. Drag the **Entity** shape from the stencil and drop it into the empty drawing area on the right. You can move the **Entity** shape around the work area by just clicking anywhere on the entity and dragging it while holding the mouse button down. When finished, your work area should have an entity as shown in Figure 8. The top area of the entity contains the entity name (*Entity1* in Figure 8). The bottom area contains the attributes (empty in Figure 8).

Technical Note: If the entity does not appear as shown in Figure 8, change the symbol set option in the Database Document Options window (**Database** → **Display Options** command). Select "IDEF1X" as the symbol set option in the General tab of the window.

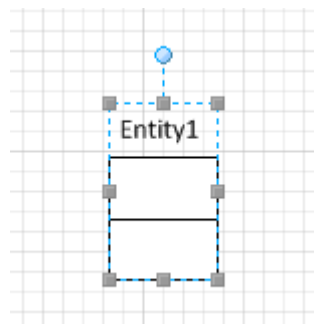


Figure 8: New Entity

Technical Note: To increase or decrease the view of the work area, go to **View** → **Zoom** and choose a size (50%–400%) as shown in Figure 9. Alternatively, you can select **View** → **Fit to Window**, **View** → **Page Width**, or adjust the zoom level in the zoom bar at the bottom of the page.

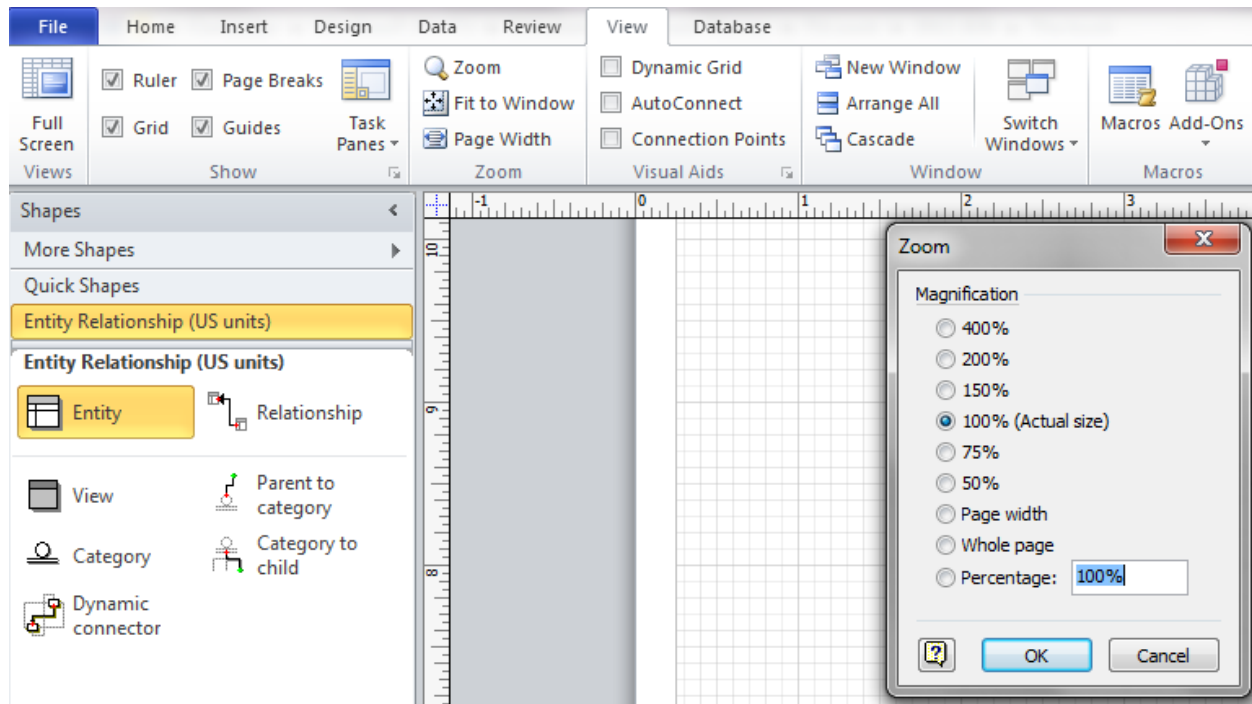


Figure 9: Options in the **View** → **Zoom** Menu

Define *Customer* Entity

Visio Professional provides the Database Properties window for each master shape in the Entity Relationship stencil. For entities, the Database Properties window allows you to define the name, columns, primary key, indexes, triggers, integrity constraints,¹ drivers for commercial database management systems, and comments. Because Visio Professional uses the terminology “column” rather than “attribute”, this chapter also uses column. Open the Database Properties window by double-clicking the entity. Alternatively, you can right-click the entity and select

¹ Textbook chapter 8 provides background about indexes, Chapter 11 about triggers, and Chapter 14 about check constraints.. This lab chapter will not cover these properties.

Database Properties... or select the entity and then choose **Shape** → **Actions** → **Database Properties....**

When the Database Properties window opens (Figure 10), use the **Definition** tab to enter the name of the entity (“Customer”). Type this name in the “Physical name” textbox and the value in the “Conceptual name” textbox changes automatically because the check box “Sync names when typing” is checked. Visio Professional provides two names (physical and conceptual) to allow differences between conceptual names on an ERD and the physical names on the implemented database.

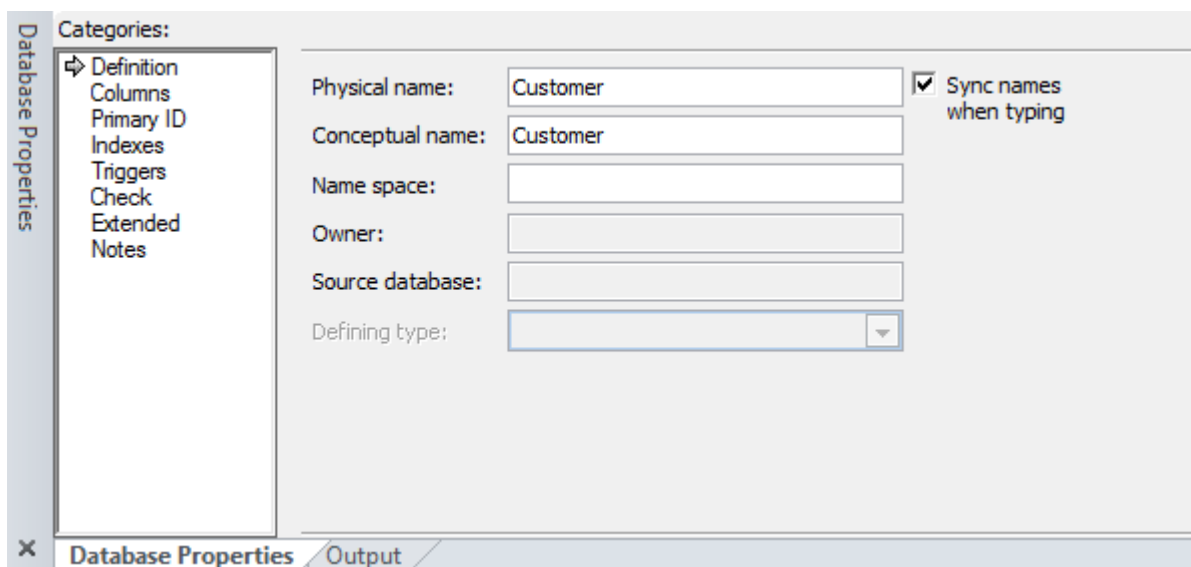


Figure 10: **Definition** Area of the Database Properties Window

Next, select the **Columns** tab and the window appears as shown in Figure 11. To name the first column, click the **Add** button. The **Columns** area of the Database Properties window changes as shown in Figure 12. You can type the column name (“CustNo”) in the designated area, but it is easier to edit all of the column properties together. Click the **Edit** button to open the Column Properties window. Type “CustNo” and deselect “Allow NULL values” and the

window appears as in Figure 13. Make choices in the **Data Type** area as shown in Figure 14.

You also can enter comments in the **Notes** area as shown in Figure 15. When you are finished, click the **OK** button to return to the Database Properties window.

Technical Note: Portable data types are independent of a DBMS. Physical data types are provided by individual DBMSs. If you are not sure of the target DBMS for a database, you should use the portable data types.

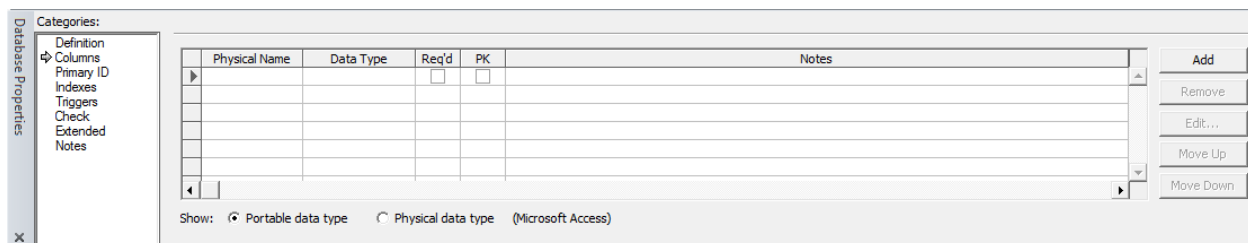


Figure 11: Empty **Columns** Area of the Database Properties Window

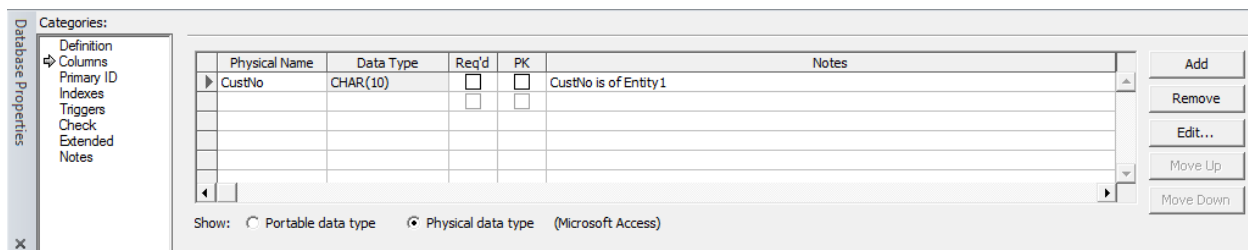


Figure 12: **Columns** Area after Clicking the **Add** Button

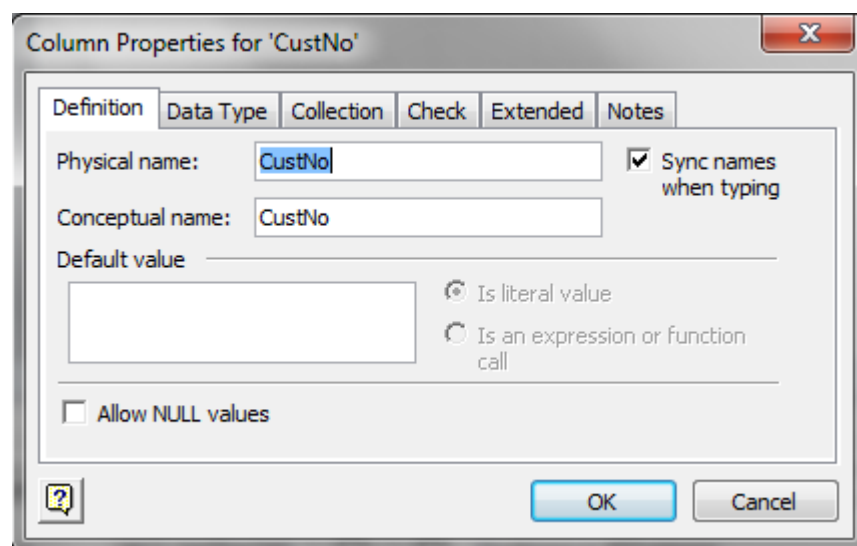


Figure 13: Column Properties Window for the First Column of the *Customer* Entity

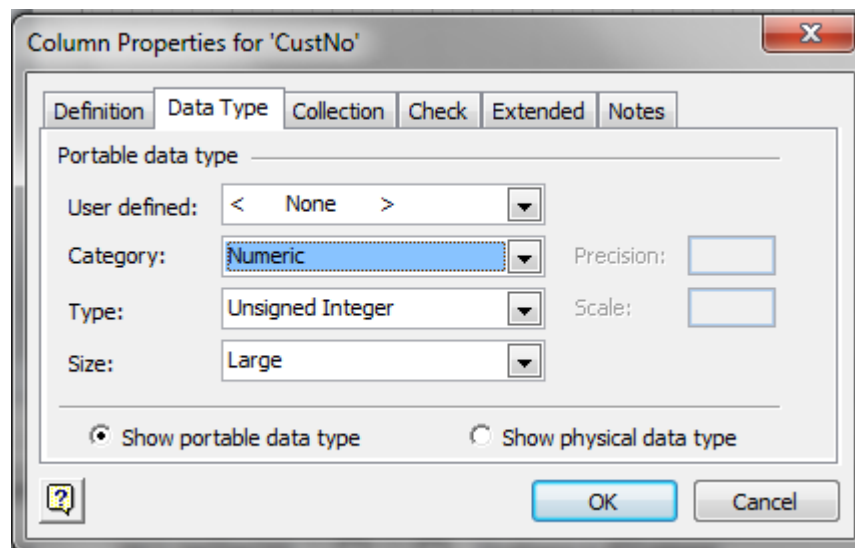


Figure 14: **Data Type** Area of the Column Properties Window

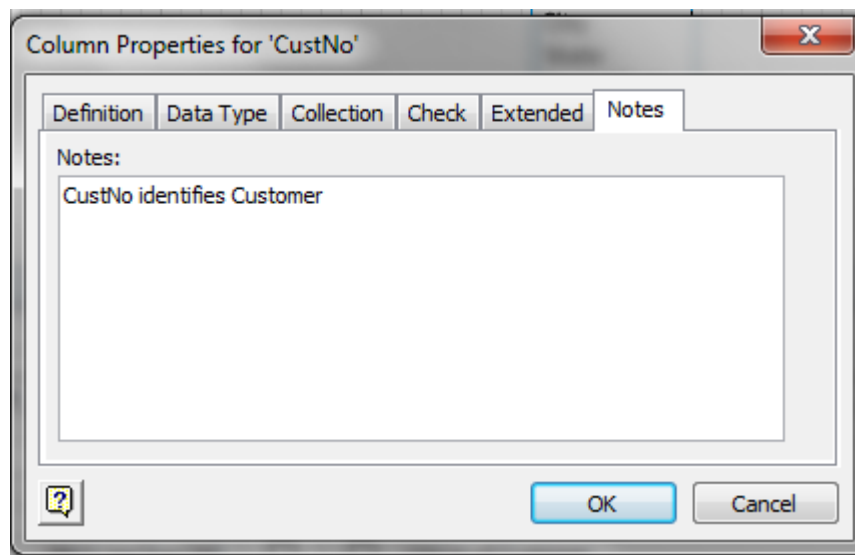
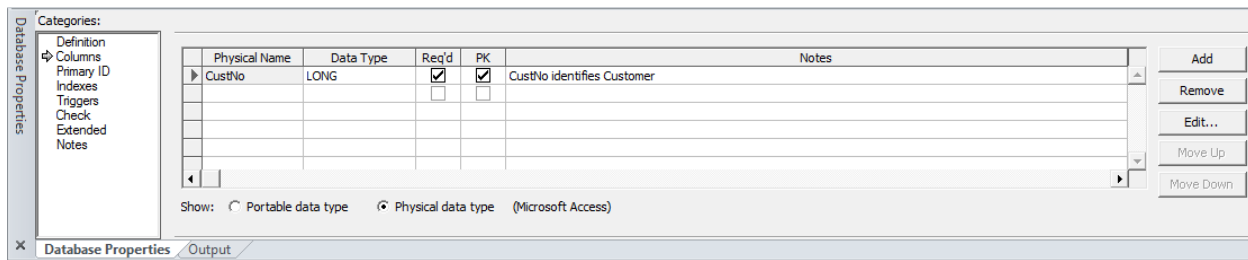
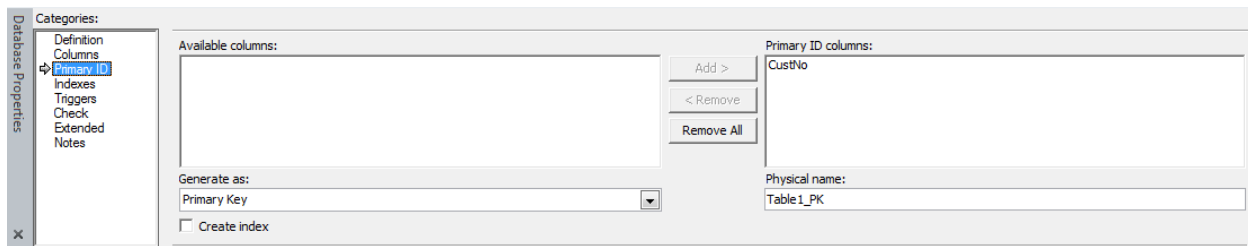
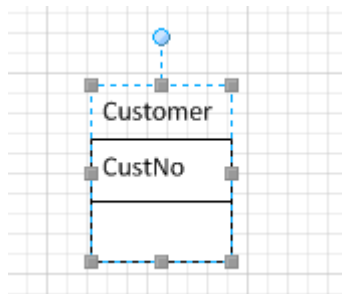


Figure 15: **Notes** Area of the Column Properties Window

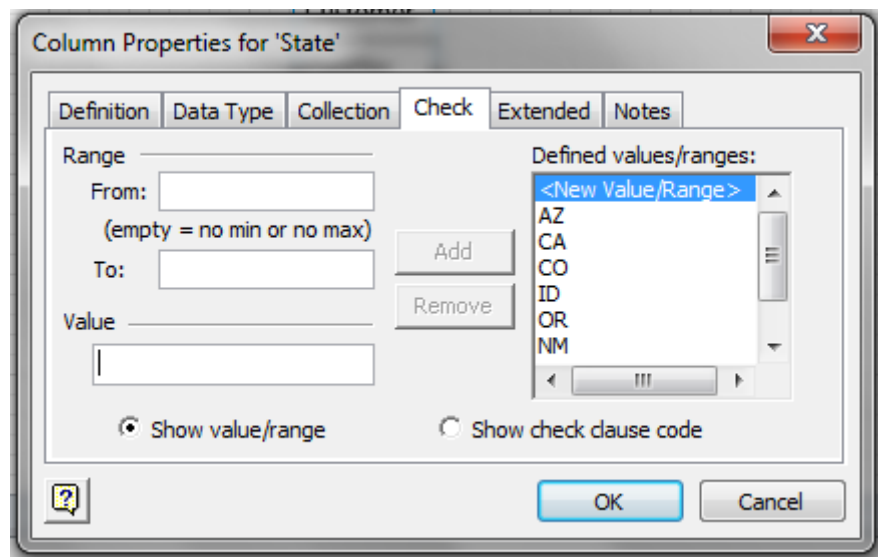
After returning to the Database Properties window, check the *PK* column and the window appears as in Figure 16. Because you checked the *PK* property, the **Primary ID** area appears as shown in Figure 17. The *Customer* entity with the underlined primary key appears as shown in Figure 18.

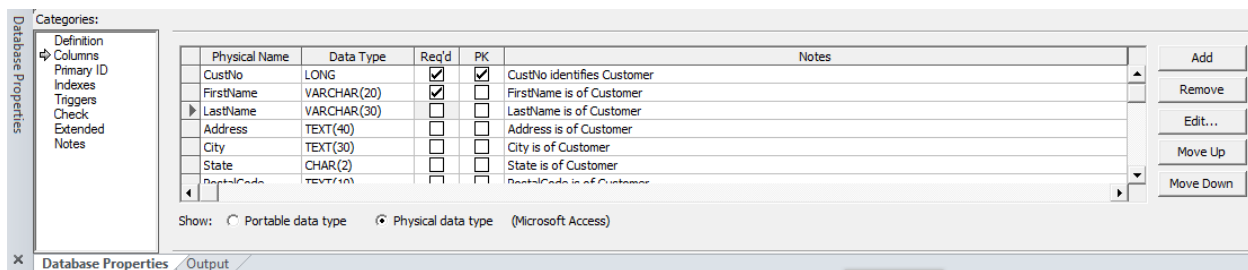
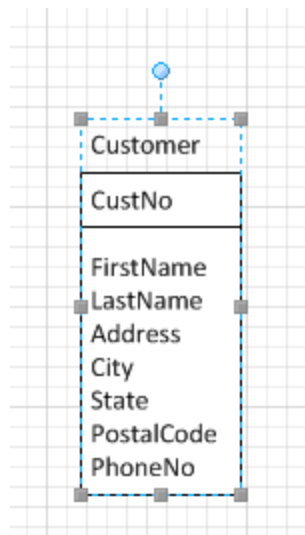
Figure 16: Database Properties Window with *PK* Property SelectedFigure 17: **Primary ID** Area when *PK* Property Is SelectedFigure 18: *Customer* Entity with *CustNo* Primary Key

Complete the remaining columns using Table 4. To define the allowable values, you need to use the **Check** area of the Column Properties window (Figure 19). To define a list of allowable values, enter a value in the Value text box and then click the **Add** button. When you have completed the columns, the Database Properties window should appear as shown in Figure 20. The completed entity should appear as shown in Figure 21.

Table 4: *Customer* Entity Column Details

Column Name	Data Type Category, Type, Size	Other Properties
<i>CustNo</i>	Long	<i>PK</i>
<i>FirstName</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "20"</i> <i>Required: "Yes"</i>
<i>LastName</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "30"</i> <i>Required: "Yes"</i>
<i>Address</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "40"</i> <i>Required: "No"</i>
<i>City</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "30"</i> <i>Required: "No"</i>
<i>State</i>	Text, Fixed Length, Single Byte Character Set	<i>Length: "2"</i> <i>Required: "No"</i> <i>Allowable values: "AZ", "CA", "CO", "ID", "OR", "NM", "OR", "WA"</i>
<i>PostalCode</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "10"</i> <i>Required: "No"</i>
<i>PhoneNumber</i>	Text, Variable Length, Single Byte Character Set	<i>Length: "10"</i> <i>Required: "No"</i>

Figure 19: **Check** Area of the Column Properties Window for the *State* Column

Figure 20: Completed Database Properties Window for the *Customer* EntityFigure 21: Completed *Customer* Entity

Visio Professional supports several different views of an entity. The appropriate view depends on the desired level of detail. For large diagrams, you typically want to suppress details. You use the **Database → Display Options...** command (Figure 22) to specify the appropriate level of detail. After opening the Database Document Options window, select the **Table** tab to see the choices in Figure 23. To see the data types, select “Show Physical” in the Data types option buttons area. Figure 24 shows the *Customer* entity with the data type properties appearing. When you are finished, remove the data type display option so that the *Customer* entity displays as shown in Figure 21.

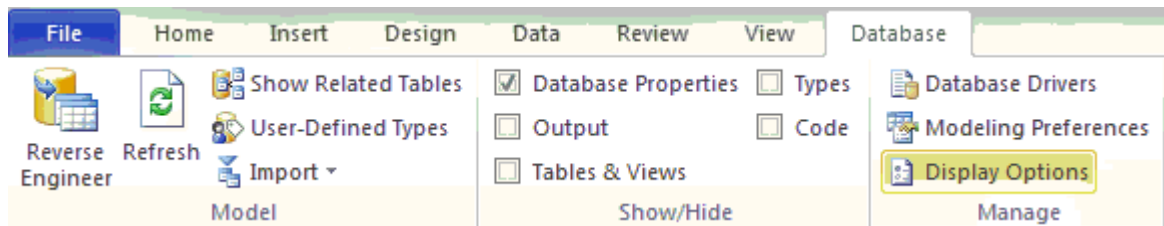


Figure 22: **Display Options** Command in the **Database** Menu

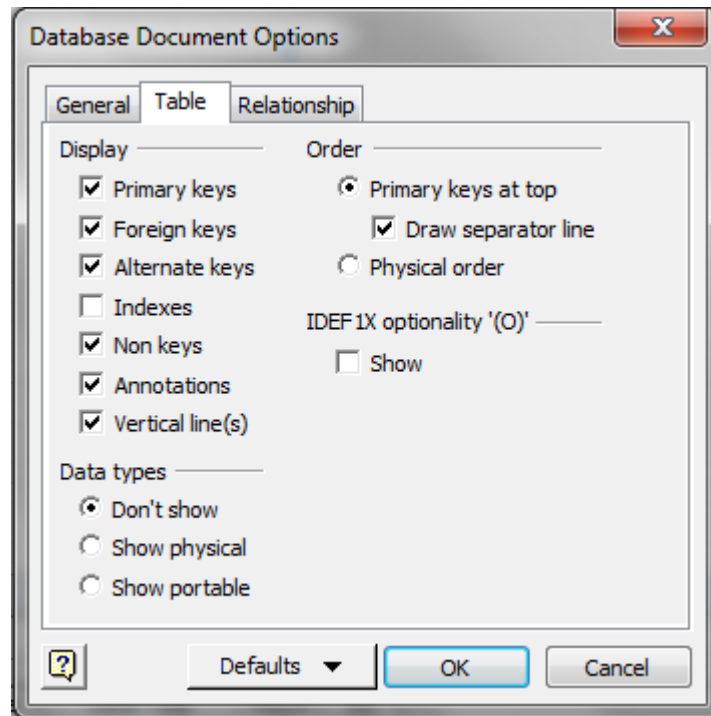


Figure 23: **Table** Area of the Database Document Options Window

Technical Note: To make permanent the settings in the Database Document Options window, use the **Defaults** button. Choose the “Set As” choice that displays after selecting the **Defaults** button.

Customer	
CusNo	LONG
FirstName	VARCHAR(20)
LastName	VARCHAR(30)
Address	TEXT(40)
City	TEXT(30)
State	CHAR(2)
PostalCode	TEXT(10)
PhoneNo	TEXT(15)

Figure 24: *Customer* Entity with Physical Data Type Properties Shown

Define *Vehicle* Entity

Next, you will create and define the properties of the *Vehicle* entity. Apply the previous instructions and use Table 5 to enter the column properties. You will need to use the **Check** area of the Column Properties window to enter the minimum value and the allowable values. When you are finished, the entities (with data types displayed) should appear as shown in Figure 25.

Save the ERD as “AutoRep.vsd” using the **File** → **Save** command.

Table 5: *Vehicle* Entity Column Information

Column Name	Data Type	Column Properties
<i>SerialNo</i>	CHAR	Length: 15 PK
<i>Year</i>	INTEGER	Required: “No” Min Value: 1910
<i>Make</i>	VARCHAR	Length: 12 Required: “No”
<i>Model</i>	VARCHAR	Length: 15 Required: “No”
<i>LicenseNo</i>	VARCHAR	Length: 10 Required: “No”
<i>State</i>	CHAR	Length: 2 Required: “No”
<i>Cylinders</i>	INTEGER	Required: “No” Allowable Values: 4, 6, 8

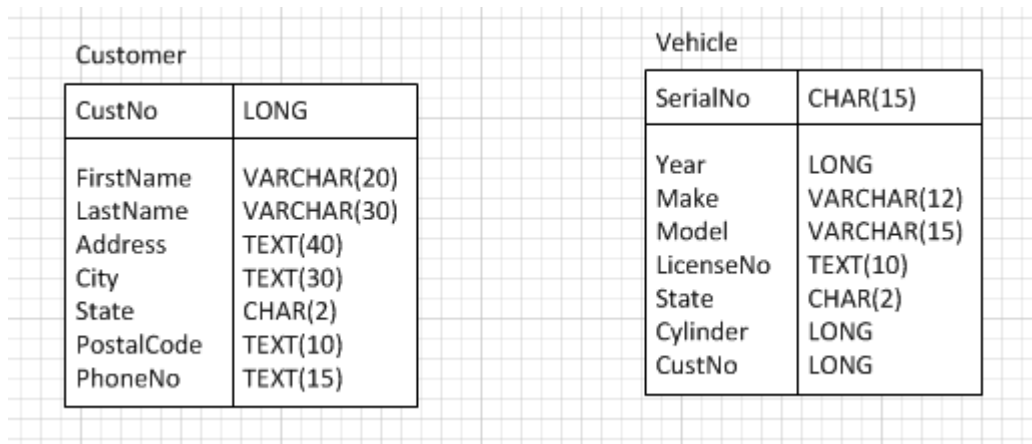


Figure 25: *Customer* and *Vehicle* Entities with Physical Data Types Displayed

4.3 Adding a Relationship

Now that these entities are defined, it is time to add a relationship between them. If you look at the ERD in Figure 4, you will see that there is a 1-M relationship between the *Customer* and the *Vehicle* entities. Textbook Chapter 5 (section 5.1.2) discusses the crow's foot notation for relationship cardinalities and specifically the placement of the symbols for minimum and maximum cardinalities on an ERD. An important point to remember is that the symbol for the minimum cardinality is on the inside facing the relationship name and the symbol for the maximum cardinality is on the outside facing the entity (Figure 26). This concept will become clearer as you create the relationships described in this subsection.

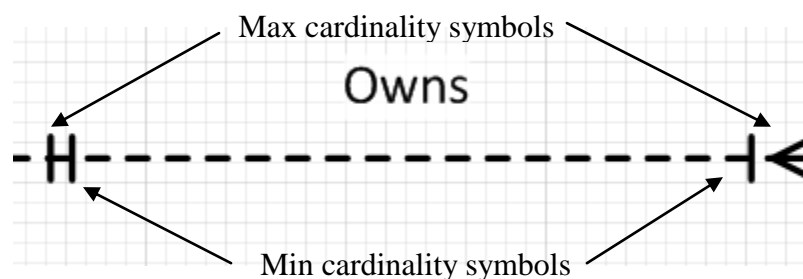


Figure 26: Minimum and Maximum Cardinalities on Each Side of a Relationship

With the above comments in mind, proceed with the instructions below to define the relationship connecting the *Customer* and the *Vehicle* entities. Visio Professional should be opened with the auto repair ERD diagram begun earlier in the drawing area.

1. Set the Cardinality Display Options: To enable the crow's feet symbols, you need to select them in the **Relationship** area of the Database Document Options window (**Database** → **Display Options...**). Set the relationship options as shown in Figure 27. Make these settings permanent by choosing the “Set As” option that appears after you click the **Defaults** button.

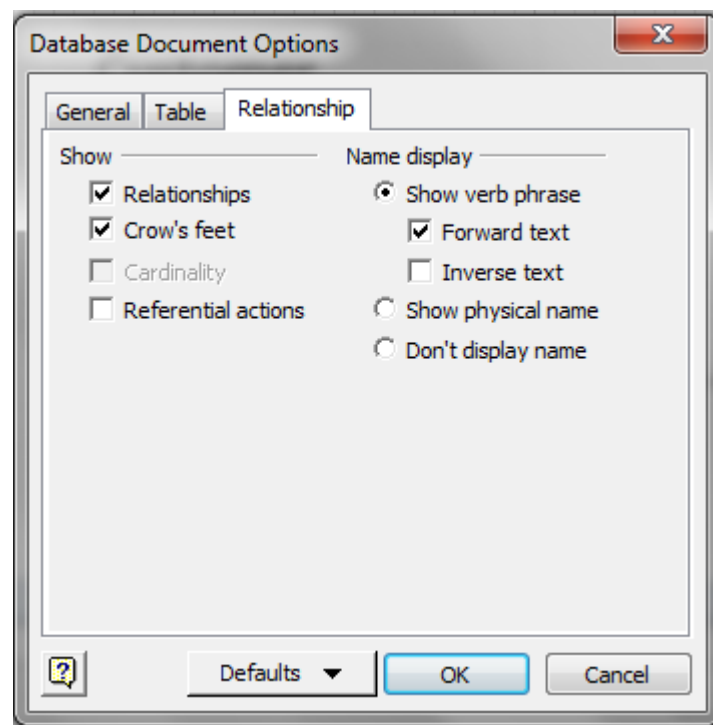


Figure 27: **Relationship** Area of the Database Document Options Window

2. Select the **Relationship** Master Shape: In the Entity Relationship stencil, select the **Relationship** master shape.
3. Connect the Entities: To create a relationship between the entity types, drag the relationship shape from the template onto the drawing area. Move the relationship symbol inside the

Customer entity type (parent entity type on the “1” side of the relationship). When positioned correctly, the border of the *Vehicle* entity type becomes red click indicating that it has been selected. After the *Customer* entity type has been selected, drag the end point (square red outline) of the relationship inside the *Vehicle* entity type (child entity type on the “M” side of the relationship). If you drag another part of the relationship (not the end point), the entire relationship shape will move so that you need to reconnect to the *Customer* entity type again. After connecting both entity types, the ERD appears as shown in Figure 28.

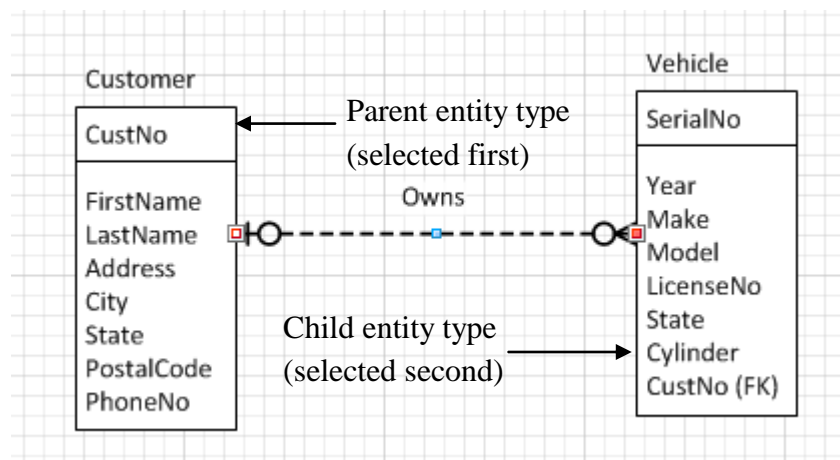


Figure 28: Owns Relationship Connecting the *Customer* and *Vehicle* Entity Types

4. Align the Entity Types: Select both entity types. Select the **Home** → **Auto Align & Space** command to automatically align the entity types. For custom positioning, you can use the **Home** → **Position** item to precisely align or position the entity types as shown in Figure 29. After alignment, your entity types and relationship appear centered, as shown in Figure 30.

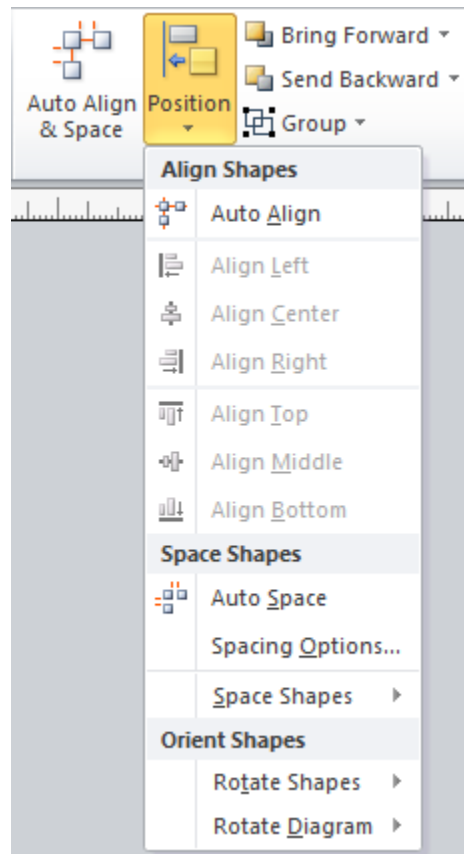
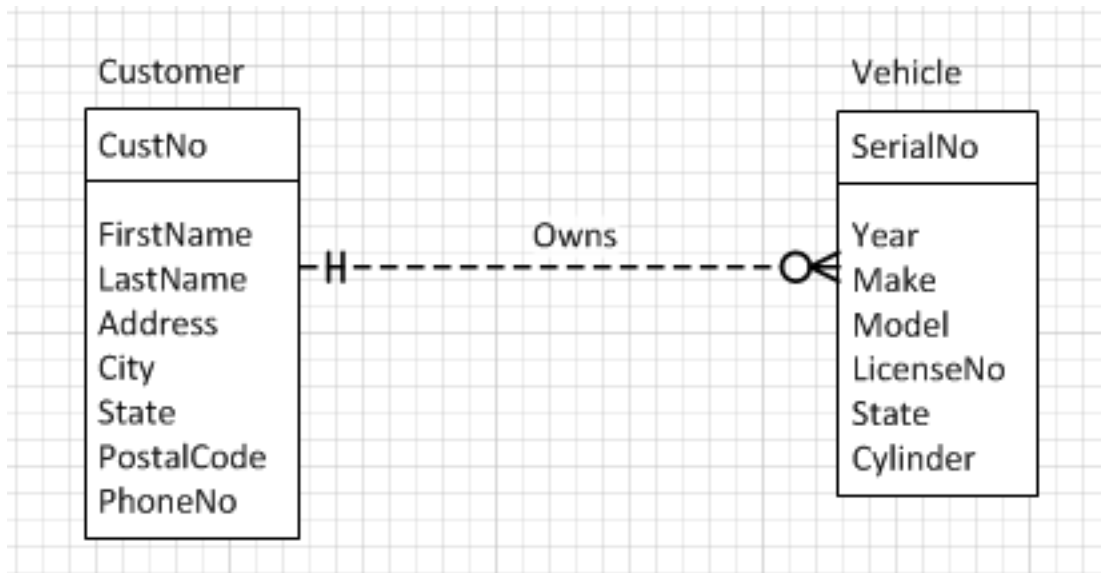
Figure 29: Options in the **Home** → **Position** item

Figure 30: Entities and Relationship after Alignment

5. Revise the Relationship Name: Double-click the relationship line to open the Database

Properties window for the Relationship. In the **Name** area, type “Owns” as the *Verb phrase* and “OwnedBy” as the *Inverse phrase*, as shown in Figure 31.

- Visio Professional supports names for both directions of a relationship. The *Verb phrase* corresponds to the name from the entity on the “one” side to the entity on the “many” side. The *Inverse phrase* corresponds to the other relationship direction. As explained in textbook Chapter 5, you should use an active verb (or verb phrase) whenever practical for the relationship name that appears on the diagram.
- In the Display Options window (**Database → Display Options**), you can display one or both relationship names on the diagram. Displaying both names can make an ERD too cluttered. It is customary to display the verb phrase rather than the inverse phrase. As shown in Figure 27, the “Forward text” in the **Relationship** area corresponds to the *Verb phrase* in the Database Properties window.

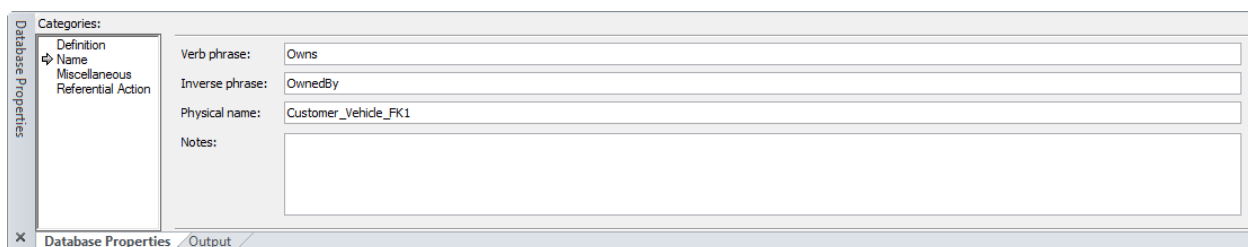


Figure 31: **Name** Area of the Database Properties Window for a Relationship

6. Revise the Relationship Cardinalities: In the **Miscellaneous** area of the Database Properties window, you can change the cardinalities for the entity appearing in the right side of the **Definition** area (Figure 32). This entity is the second one that you selected before creating the relationship. Set the cardinality in the **Miscellaneous** area as shown in Figure 33. Figure 34 shows the ERD after setting the cardinalities. Table 6 explains each cardinality choice in the

Miscellaneous area. See textbook Chapter 5 (section 5.1.2) for more details about cardinalities.

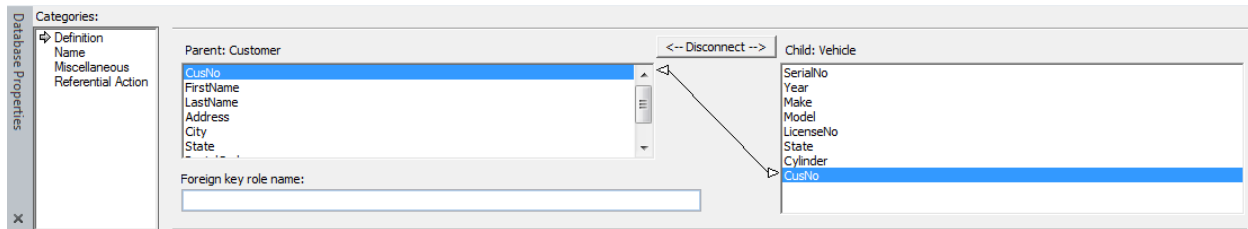


Figure 32: Entity Types in the **Definition** Area

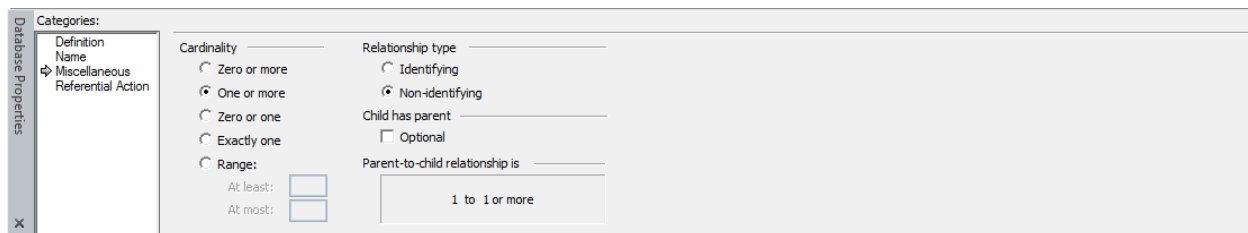


Figure 33: *Cardinality* Settings in the **Miscellaneous** Area

Table 6: Minimum and Maximum Cardinality Choices

Choice	Meaning
Zero or more	Minimum cardinality = 0 Maximum cardinality = “many”
One or more	Minimum cardinality = 1 Maximum cardinality = “many”
Zero or one	Minimum cardinality = 0 Maximum cardinality = 1
Exactly one	Minimum cardinality = 1 Maximum cardinality = 1
Range	Set exact cardinalities in the <i>At least</i> and the <i>At most</i> textboxes
Optional check box	When checked, it sets the minimum cardinality to 0 for the parent entity (the entity type on the left side of the Definition area)

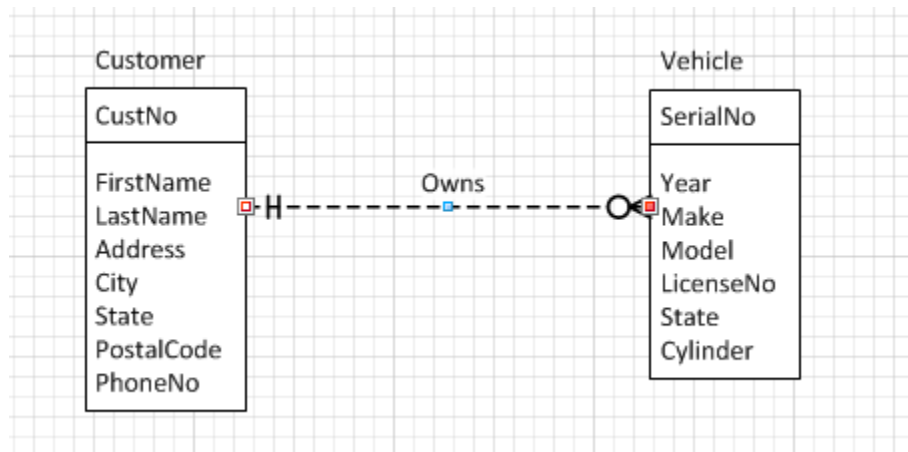


Figure 34: *Owns* Relationship after Setting the *Cardinality* Properties

7. Examine Foreign Keys: Visio Professional allows you to view foreign keys in addition to seeing the relationships. Normally foreign keys should be suppressed in an ERD because relationships convey the same information. To understand how a relationship translates to foreign keys, you can use the **Table** area (Figure 35) of the Database Document Options window (**Database** → **Display Options...**). Select the “Foreign keys” check box and then click the **OK** button to see the changes in the *Vehicle* table (Figure 36). Notice that the *CustNo* foreign key appears at the bottom of the *Vehicle* entity. After you are finished viewing the *Vehicle* entity, unselect the “Foreign keys” option in the Database Document Options window. The resulting ERD should appear again as shown in Figure 34.

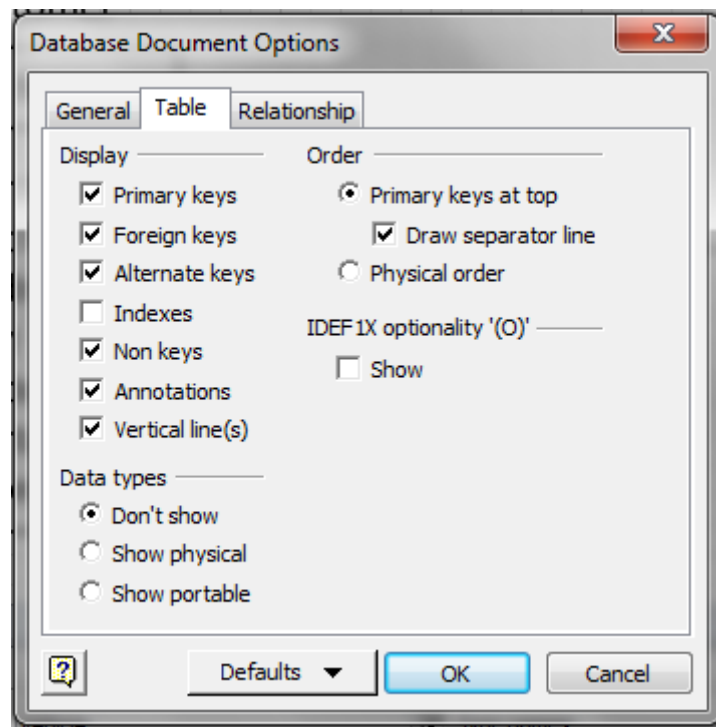


Figure 35: Display Foreign Keys Option in the Database Document Options Window

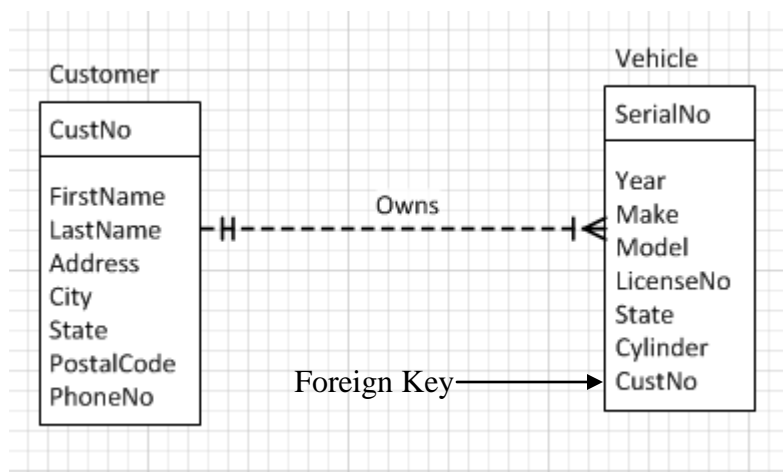



Figure 36: Owns Relationship with the Foreign Key Displayed

8. Move the Words: When you define a relationship, the relationship name is positioned inside the line. Sometimes this default positioning obscures the relationship name. You can move the relationship name away from the relationship line by using the **Text Block** button () on the **Home** tab, in the **Tools** group. The moved relationship name is shown in Figure 36.

9. Remove Text Background: To set solid text background as the default for relationship names, you need to modify the connector style. Right click on the relationship symbol and then select the **Format** → **Text...** command. Click the Text Block tab button in the Text window. You then make the change to the text background as shown in Figure 37.
10. Examine the Final Relationship: When you are finished, your relationship should appear as shown in Figure 38. Save your ERD again using the **File** → **Save** menu item.

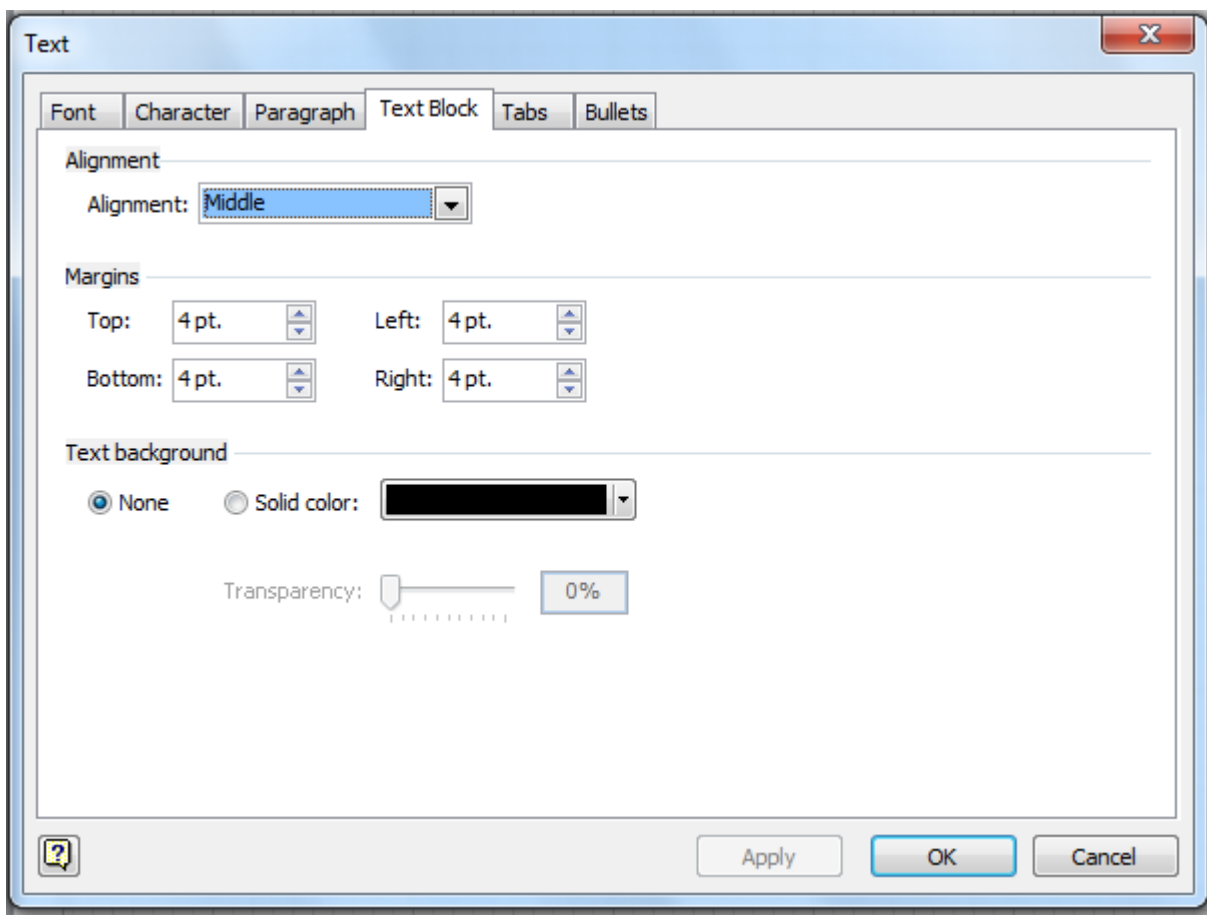
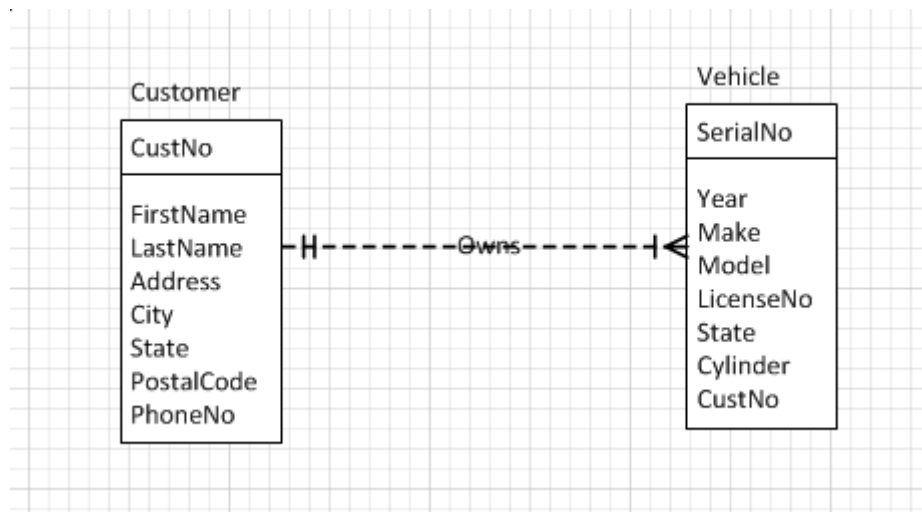


Figure 37: No Color for the Text Background Setting in the Text Window

Figure 38: *Owns* Relationship with No Text Background

4.4 Creating the Remaining Entities

To finish the ERD, you need to create the *RepairOrder*, the *Part*, and the *PartsUsed* entities. After these entities are created, you will be instructed to place the necessary relationships between them. Follow the instructions in the previous sections and use Tables 7, 8, and 9 for the column specifications. Note that the *PartsUsed* entity is a weak entity with identifying relationships (see textbook Chapter 5, section 5.2.1). When you are finished, your entities should appear as shown in Figure 39.

Table 7: Column Specifications for the *RepairOrder* Entity

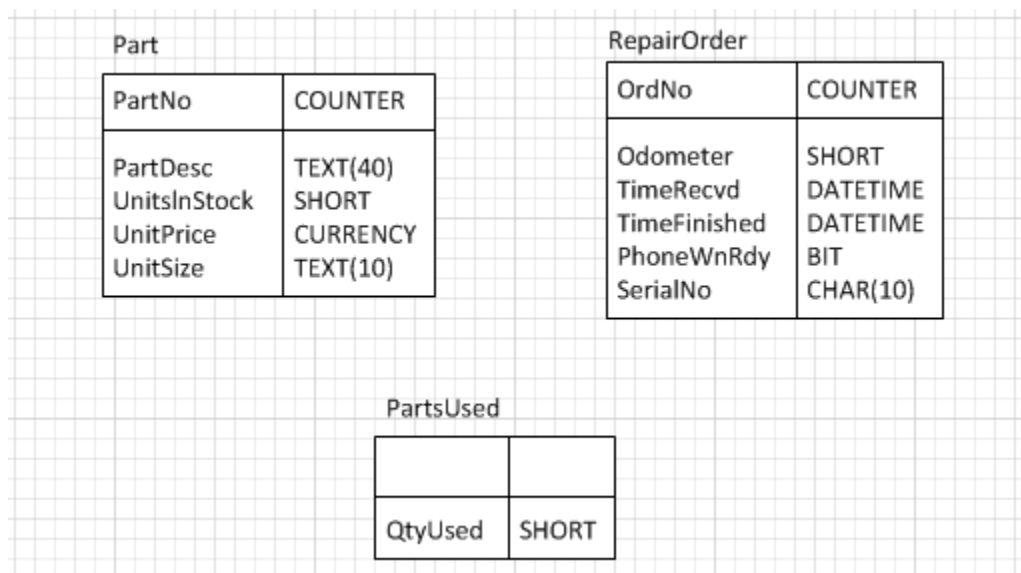
Column Name	Data Type	Column Properties
<i>OrdNo</i>	Counter	<i>Primary Key</i> <i>Precision: 10</i>
<i>Odometer</i>	Short	<i>Required: "No"</i>
<i>TimeRecvd</i>	DateTime	<i>Required: "No"</i>
<i>TimeFinish</i>	DateTime	<i>Required: "No"</i>
<i>PhoneWnRdy</i>	Bit	<i>Required: "No"</i>

Table 8: Field Specifications for the *Part* Entity

Column Name	Data Type	Column Properties
<i>PartNo</i>	Counter	<i>Primary Key</i> <i>Precision: 10</i>
<i>PartDesc</i>	Text	<i>Length: 40</i> <i>Required: "No"</i>
<i>UnitsInStock</i>	Short	<i>Required: "No"</i>
<i>UnitPrice</i>	Currency	<i>Precision: 10</i> <i>Scale: 2</i>
<i>UnitSize</i>	Text	<i>Length: 10</i> <i>Required: "No"</i>

Table 9: Column Specifications for the *PartsUsed* Entity

Column Name	Data Type	Column Properties
<i>QtyUsed</i>	Short	<i>Required: "No"</i>

Figure 39: *RepairOrder*, *Part*, and *PartsUsed* Entities with Physical Data Types

4.5 Creating the Remaining Relationships

The remaining three relationships involve the *Vehicle* and the *RepairOrder* entities, the *RepairOrder* and the *PartsUsed* entities, and the *PartsUsed* and the *Part* entities. For each

relationship, follow the instructions in Section 4.3 for creating a relationship. Table 10 provides specifications for each relationship.

1. Align Entities: To begin, align the entities as shown in Figure 40 with the *RepairOrder* entity directly below the *Vehicle* entity and the *PartsUsed* entity centered below the *RepairOrder* and the *Part* entities.
2. Select Entities in the Specified Order: Before creating each relationship, please select the entities in the order specified in Table 10 because Visio Professional assumes that the first entity selected is on the one side of the relationship.
3. Make Identifying Relationships: The *Uses* and *UsedIn* relationships are identifying relationships. To make a relationship identifying, you need to set the *Identifying* option in the **Miscellaneous** area of the Database Properties window as shown in Figure 41.
4. View the ERD: When you are finished, your ERD should appear as shown in Figure 42. Notice the solid lines for each identifying relationship. In addition, the *PartsUsed* entity displays foreign keys because they are part of the primary key and the “Primary keys” display option was selected in the Document Database Options window.

<p>Technical Note: If a weak entity has a column that is part of the primary key, check the <i>PK</i> option for the column in the Database Properties window. When you make a relationship identifying, the foreign key from the related entity will become part of the primary key of the weak entity. The resulting primary key consists of the borrowed foreign key plus the column in the entity.</p>

5. Change to the IDEF1X Symbols: When using the Relational Model symbols, Visio Professional does not use any symbol for weak entities.² If you want to see a special symbol for weak entities, you need to use the IDEF1X symbols. Select “IDEF1X” as the symbol set option in the Database Document Options window, as shown in Figure 43. After you click

² A previous version of Visio Professional used the weak entity symbol that appears in textbook Chapter 5.

the **OK** button, the ERD appears as shown in Figure 44. After you are finished viewing the

ERD, change the symbol set back to the Relational symbol set.

Technical Note: After changing the symbol set, Visio Professional may change some relationships from “Non-Identifying” to “Identifying”. The author believes this to be a software bug. If the relationships are changed, you will need to manually change them back or undo the symbol set change.

6. Save the ERD: Save your ERD again before finishing this section.

Table 10: Details about the Remaining Relationships

Entities	Relationship Name	Relationship Type	Cardinality	Comments
<i>Vehicle RepairOrder</i>	<i>Requires</i>	Non-Identifying	One or more	Each repair order requires a max of one vehicle and a min of one vehicle. Each vehicle requires a max of many repair orders and a min of one repair order.
<i>RepairOrder PartsUsed</i>	<i>Uses</i>	Identifying	Zero or more	Each part used has a max of one repair order and a min of one repair order. Each repair order has a max of many parts used and a min of zero parts used since there may not always be a part used in a repair order.
<i>Part PartsUsed</i>	<i>UsedIn</i>	Identifying	Zero or more	Each part uses a max of many parts used and a min of zero parts used. Each parts used has a max of one part and a min of one part.

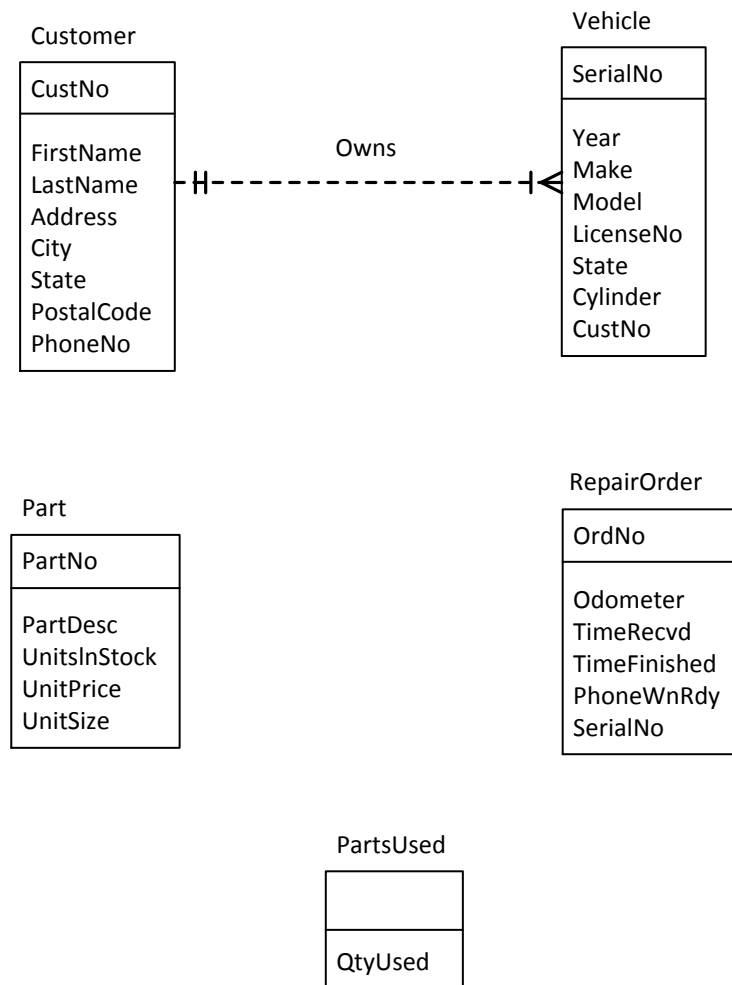
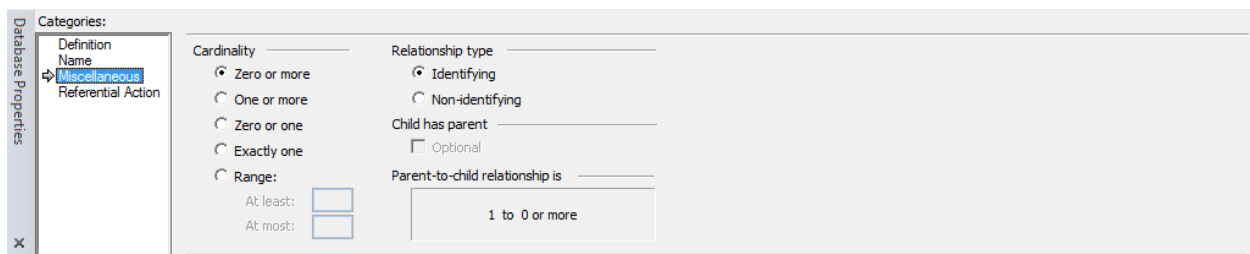


Figure 40: Entities Aligned Prior to Creating Relationships

Figure 41: **Miscellaneous** Area in the *Uses* Relationship Properties Window

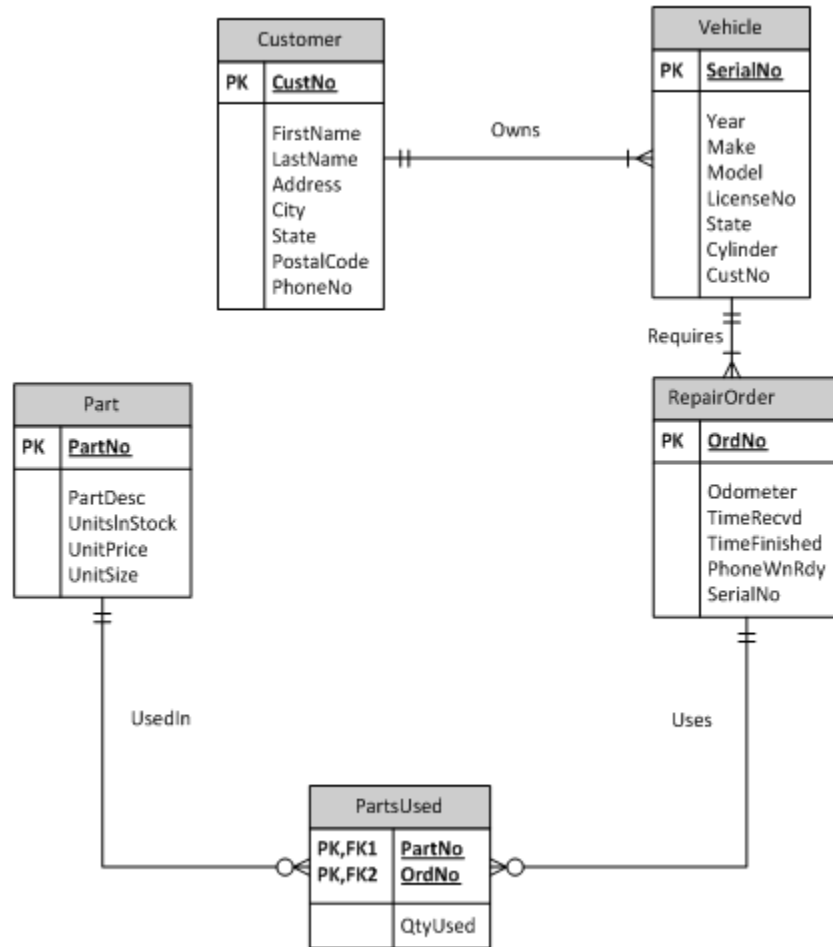


Figure 42: Completed Auto Repair ERD

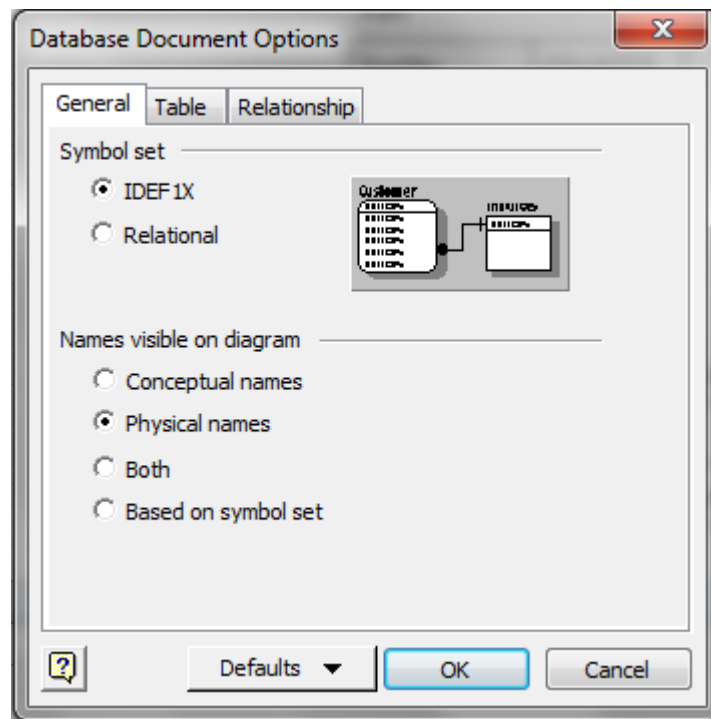


Figure 43: Selecting the IDEF1X Option in the Database Document Options Window

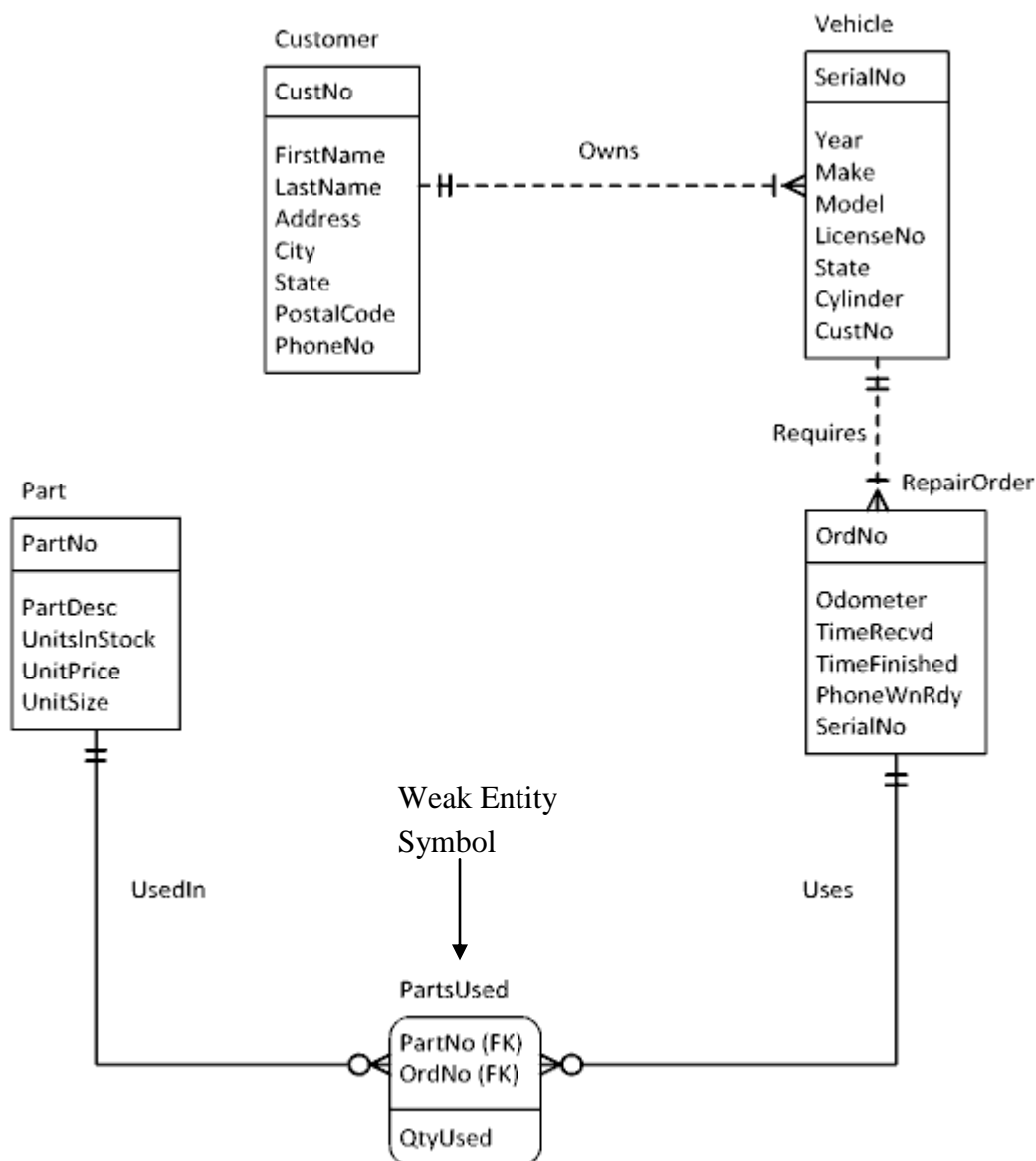


Figure 44: Auto Repair ERD using the IDEF1X Symbol Set

4.6 Automating the Layout of an ERD

Visio Professional provides an automatic layout feature to help you improve the readability of your ERDs. This feature is especially useful for large diagrams, but it also can be helpful on smaller diagrams such as the auto repair ERD. For large diagrams, it is often essential

to use the automatic layout feature as a first step to achieve a readable layout. Use the following instructions to apply the automatic layout feature to the auto repair ERD.

1. Choose Layout Property Settings: After you select the **Design** → **Page Setup** → Layout and Routing... command, the Layout and Routing window appears (Figure 45). Use the settings as shown in Figure 45. Table 11 summarizes the properties in the Layout Routing window.
2. View the Result: After clicking the **OK** (or **Apply**) button, you can view the new layout (Figure 46). You might want to change the zooming level to 50% or 75% to see the entire ERD. If you are not satisfied, you can undo the changes using the **Edit** menu. Save the diagram when you are finished viewing it.

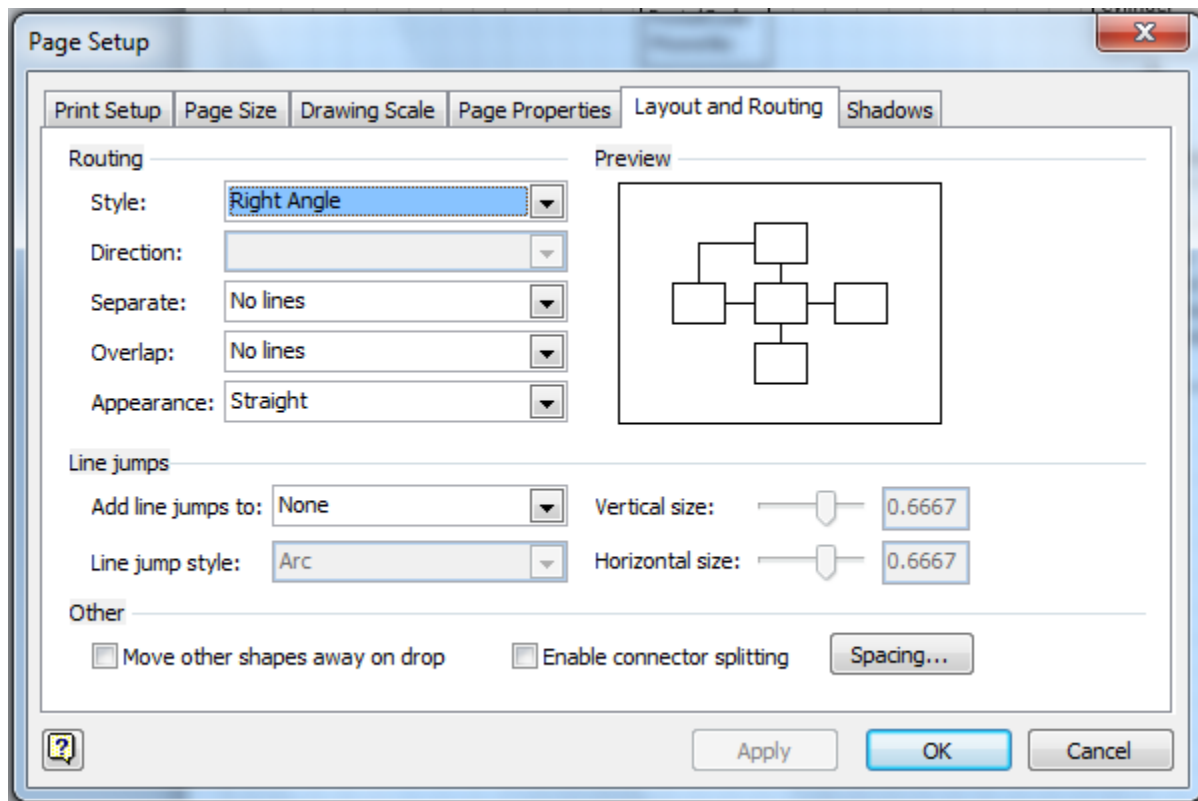


Figure 45: Lay Out Shapes Window with Default Settings

Table 11: Layout Property Settings

Layout Property	Explanation and Values
<i>Placement Style</i>	Use “Radial” or “Circular” for undirected diagrams such as ERDs. Use the “Flowchart” setting for directed diagrams such as flowcharts.
<i>Placement Direction</i>	Determines the direction in which shapes will be arranged. Cannot be specified for the radial or circular styles. For the other styles, the values include “Top to Bottom”, “Bottom to Top”, “Left to Right”, and “Right to Left”.
<i>Placement Depth</i>	Determines the height of the layout. A shallow depth means a wide but short layout. A deep layout means a more narrow but tall layout.
<i>Connectors Style</i>	Determines the angle at which to connect shapes. Values include “Right Angle”, “Straight”, and “Center to Center”.
<i>Connectors Direction</i>	Determines the direction in which shapes will be connected. Cannot be specified for the radial or circular styles. For most diagrams, the <i>Connectors Direction</i> should match the <i>Placement Direction</i> .
“Apply routing style to connectors”	Indicates that the routing style will be applied to all connections on a page.
“Align shapes to grid when possible”	Check to use grid settings specified through the Spacing button.
“Enlarge page ...”	Check to automatically enlarge the page if needed.

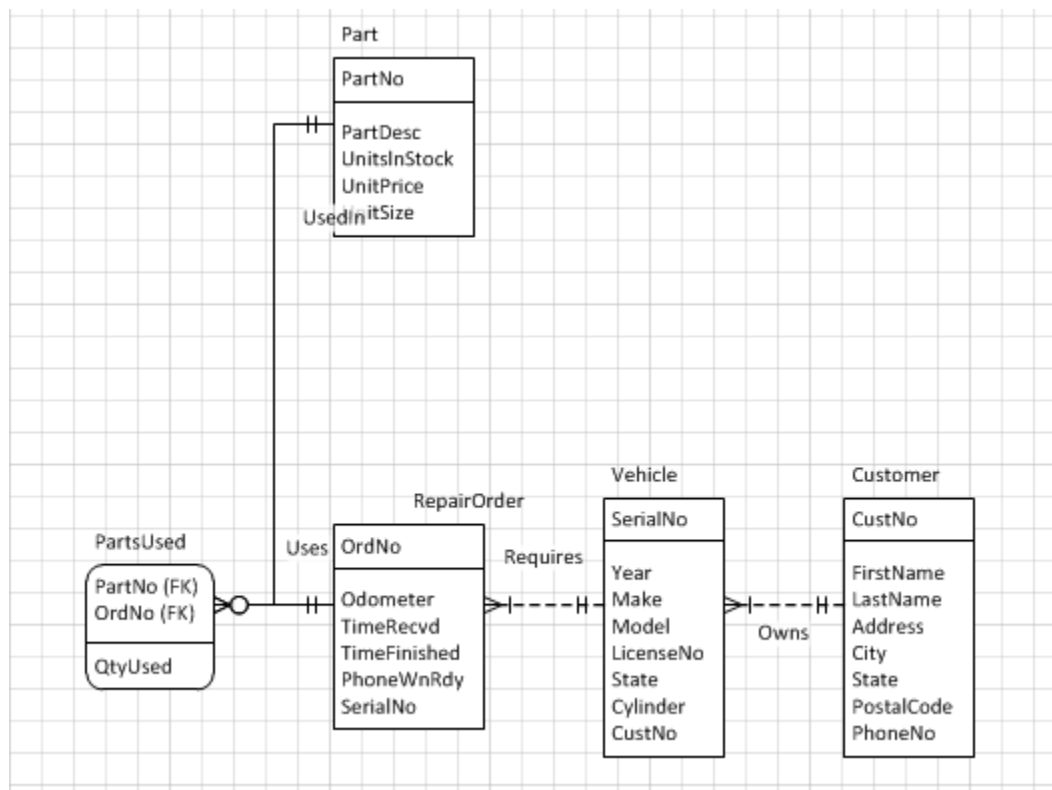



Figure 46: Automated Layout for the Auto Repair ERD

5 Using Visio Professional for ERD Variations

In Section 4, you used Visio Professional for a typical ERD. The auto repair ERD demonstrates 1-M relationships and a weak entity with identifying relationships. This section demonstrates how to use Visio Professional for two important ERD variations: self-referencing relationships and generalization hierarchies. Since the examples to depict these variations do not use the auto repair database, you will create a new diagram with one page for each example ERD.

5.1 Creating Self-Referencing Relationships

In this subsection, you will create a new diagram to practice self-referencing (unary) relationships. You will create one self-referencing relationship involving the *Faculty* entity example from textbook Chapter 5. Because the IDEF1X notation does not support M-N relationships, you will create a second example to indirectly represent an M-N self-referencing relationship for the *Course* entity example in textbook Chapter 5. In the second example, you will use a weak entity and identifying relationships. If you have forgotten about self-referencing relationships, review textbook Chapter 5 before proceeding. Follow the instructions below to complete these two examples.

1. Open a New Diagram: Select the **New** → **Database** → **Database Model Diagram** command to create an empty ERD. You should see a blank drawing page with the Entity Relationship stencil on the left. If the stencil does not appear, you should open it using the Open Stencil button  button.
2. Create Entities and Add Attributes: Create the *Faculty* and the *Course* entities. Define columns as follows:

- Columns for the *Faculty* Entity: *FacNo* (primary key, Text, Fixed Length, Single Byte Character Set, 11 characters), *FacName* (Text, Variable Length, Single Byte Character Set, 50 characters), and *SuprFacNo* (Text, Fixed Length, Single Byte Character Set, 11 characters). The last column is necessary to establish the self-referencing relationship.
 - Columns for the *Course* Entity: *CourseNo* (primary key, Text, Fixed Length, Single Byte Character Set, 10 characters) and *CrsDesc* (Text, Variable Length, Single Byte Character Set, 50 characters).
3. Create a Self-Referencing Relationship: You cannot use the **Connector** button to create a self-referencing relationship. Instead you will use the **Relationship** shape in the Entity Relationship stencil.
- Add a Relationship Shape: Drag and drop the **Relationship** master shape
-
- from the stencil to the drawing area.
- Select the Parent Entity: Select the relationship that you just added to the drawing area. Drag and drop the “one” side of the relationship to the *Faculty* entity. Release the mouse when the entity turns red.
 - Select the Child Entity: Select the relationship again. Then drag and drop the “many” side of the relationship to the *Faculty* entity. Release the mouse when the entity turns red.
 - Make the Relationship Association: In the **Definition** area of the relationship’s Database Properties window, select “FacNo” on the left and “SuprFacNo” on the right. Then click the **Associate** button and the Database Properties window appears as shown in Figure 47.

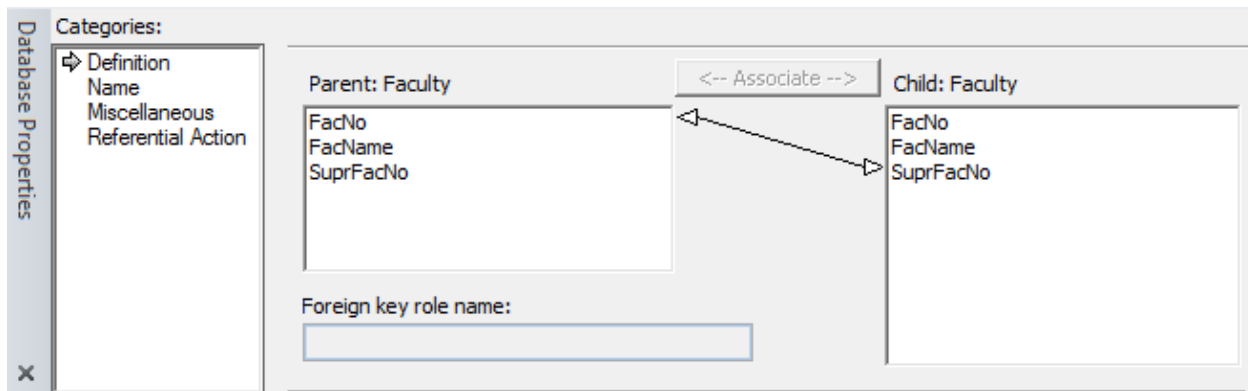


Figure 47: Relationship Properties after Associating the Primary and the Foreign Key

- Define the Other Relationship Properties: *Verb phrase:* “Supervises”, *Inverse phrase:* “SupervisedBy”, *Cardinality:* “Zero or more”, “Optional”. Each faculty can be supervised by a maximum of one faculty and a minimum of zero faculty. Each faculty can supervise a maximum of many faculty and a minimum of zero faculty.
- View the Relationship: Close the Database Properties window to view the self-referencing relationship. To ensure that the you have properly created the relationship, you should view the foreign key. In the **Table** area of the Database Document Options window (**Database** → **Display Options...**), check the “Foreign keys” display option.

Figure 48 shows the resulting ERD.

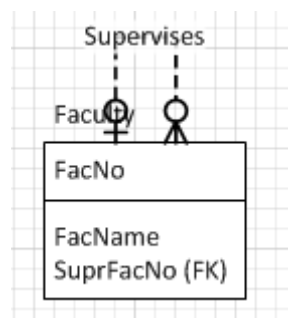


Figure 48: *Supervises* Relationship with the Foreign Key Displayed

- Refine the Appearance of the Relationship: To improve the appearance of the relationship, you should select it and then move the midpoints so that the entity type name is no longer obscured. Figure 49 shows the *Supervises* relationship after the midpoints were moved. Note that the entity type name is no longer obscured. It would seem a good idea to reroute the relationship so that it connects from the top to the side. However, the author's attempts to reroute the relationship were not successful.

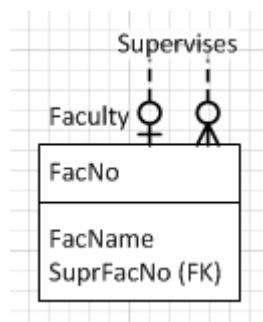


Figure 49: *Supervises* Relationship after Moving Midpoints

4. Create an Indirect Representation of an M-N Self-Referencing Relationship: In textbook Chapter 5, the *Prereq_To* relationship, an M-N self-referencing relationship involving the *Course* entity, provides the prerequisite structure of courses. A course can have many prerequisites and a course can be a prerequisite to many courses. Because the IDEF1X notation does not support M-N relationships, you need to create an indirect representation with a weak entity and two identifying relationships. Follow these steps to create an indirect representation of the *Prereq_To* relationship.
- Create *Prerequisite* Entity: Create a new entity named “Prerequisite”. Create one column as follows: *PrereqCourseNo* (Text, Fixed Length, Single Byte Character Set, 10 characters). This column is used to establish one of the identifying relationships.

- Create the *PrerequisiteOf* Relationship: Create a relationship with the *Course* entity as the parent and the *Prerequisite* entity as the child. The relationship connects *Course.CourseNo* to *Prerequisite.CourseNo*. Set the relationship properties as follows: *Verb phrase*: “Prerequisite”, *Inverse phrase*: “CourseOf”, *Cardinality*: “Zero or more”, and *Relationship type*: “Identifying”. This relationship provides the list of prerequisite courses for a given course.
- Create the *PrerequisiteTo* relationship: Create a second relationship with the *Course* entity as the parent and the *Prerequisite* entity as the child. The relationship connects *Course.CourseNo* to *Prerequisite.PrereqCourseNo* as shown in Figure 50. Set the relationship properties as follows: *Verb phrase*: “PrerequisiteTo”, *Inverse phrase*: “CourseOf”, *Cardinality*: “Zero or more”, and *Relationship type*: “Identifying”. This relationship provides the list of courses that immediately follow a given course.

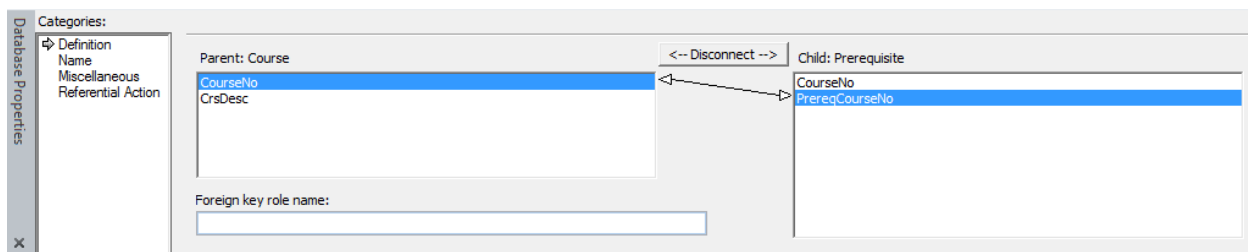


Figure 50: Relationship Properties after Associating the Primary and the Foreign Key

- View the Relationships: Close the Database Properties window to view the relationships. To ensure that you have properly created the relationships, you should view the foreign key. In the **Table** area of the Database Document Options window (**Database** → **Display Options...**), check the “Foreign keys” display option. Figure 51 shows the resulting ERD.

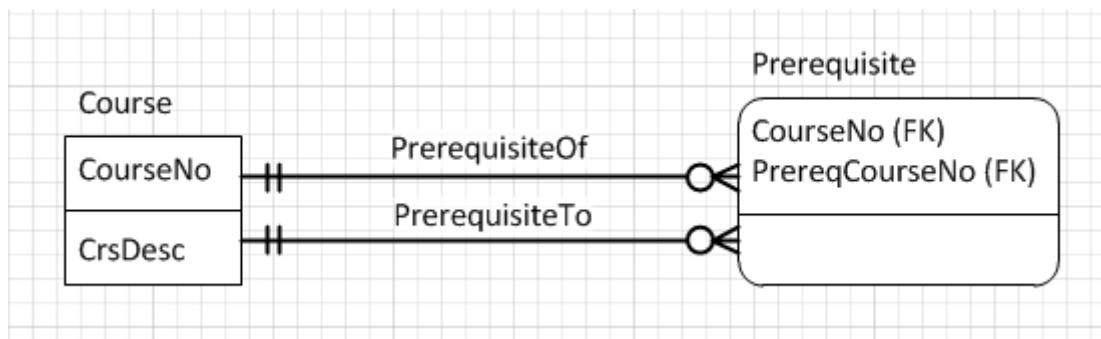


Figure 51: Indirect Representation of an M-N Relationship

5. Save the Diagram: The final diagram with both examples appears as shown in Figure 52.

When finished viewing it, save it as “Section5Examples.vsd”.

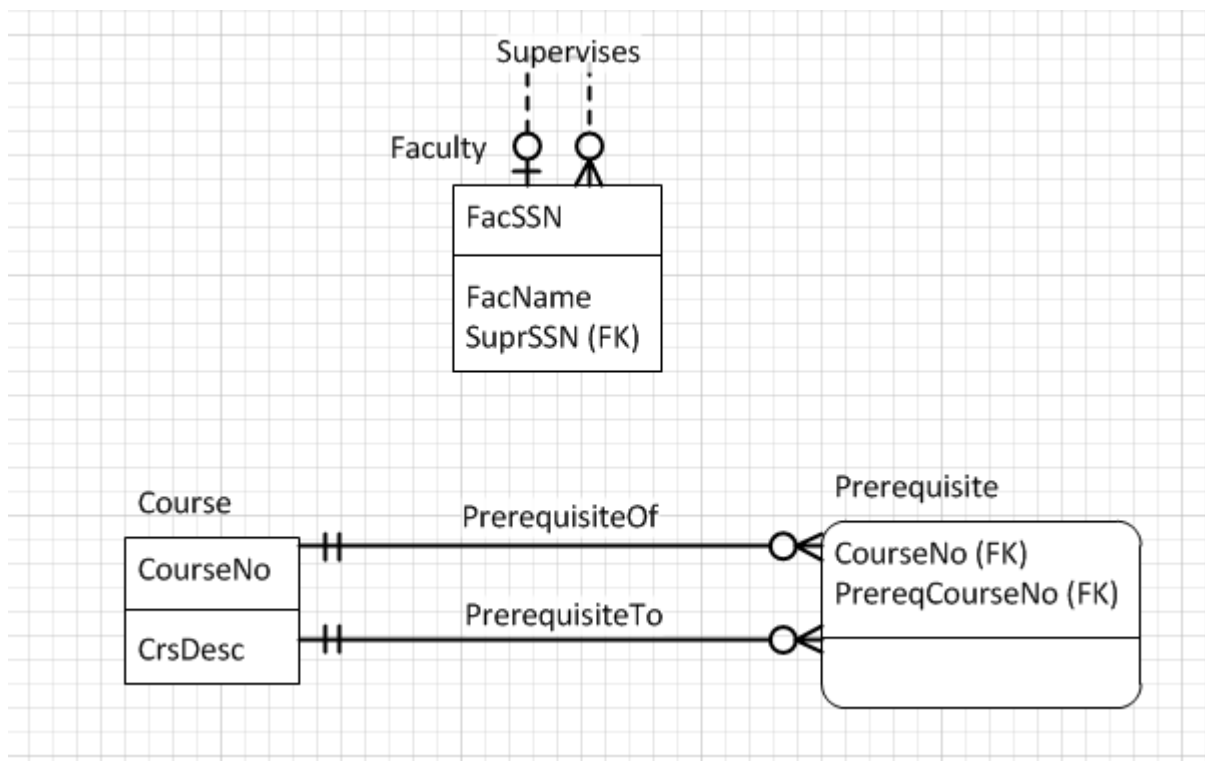


Figure 52: Drawing Area Showing Both Self-Referencing Examples

5.2 Creating a Category

In the IDEF1X notation of Visio Professional, a generalization hierarchy is known as a category. A category relates a collection of subtype entities. Before proceeding with the practice

of creating a category, you should understand the differences between IDEF1X categories and generalization hierarchies in textbook Chapter 5.

- IDEF1X uses slightly different symbols for categories than shown in Chapter 5 for generalization hierarchies.
- IDEF1X allows a parent entity to have multiple categories. In contrast, the notation in textbook Chapter 5 allows only a single generalization hierarchy for a parent entity. Since this extension (multiple categories) is seldom used, the practice in this section ignores it.
- The IDEF1X notation allows a parent entity to contain a discriminating column. A discriminating column contains one value for each subtype in the category. The discriminating column can be checked in a condition to determine class membership of a parent entity occurrence. In textbook Chapter 5, there is no mention of discriminating attributes although such an attribute can be defined. Defining a discriminating attribute is an alternative to a disjointness constraint as long as the attribute does not allow null values.
- The IDEF1X notation supports completeness constraints, but the symbol used is different than shown in textbook Chapter 5.

Before you create a category, you will create a second page for your diagram and add a new diagram on the second page. Follow the steps below to create a second drawing page.

- Use the **Insert** → **Pages** → **Blank Page ...** command to create a second page in the diagram from Section 5.1. Accept the default settings in the Setup Page window and click **OK**.

- You will see a blank drawing area with tabs for each page as shown in Figure 53. The current page is **Page-2**. If you click the tab for **Page-1**, you will see the diagram that you created in Section 5.1.

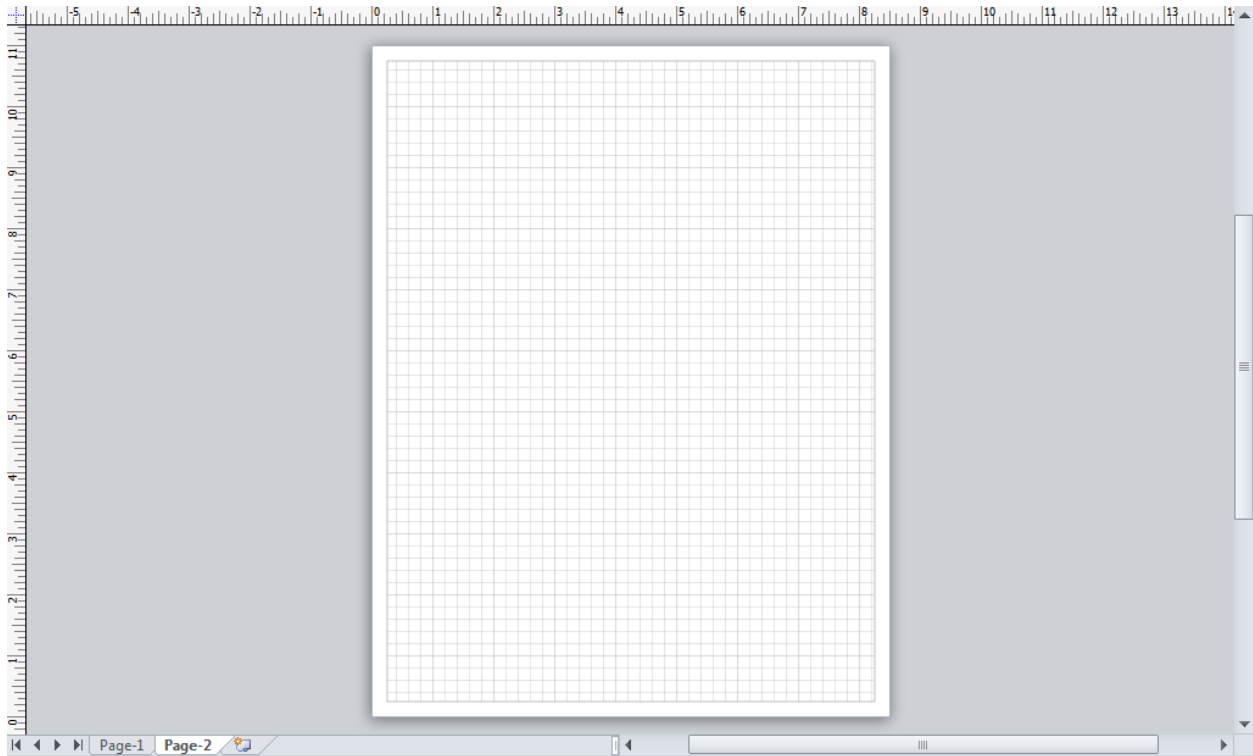


Figure 53: Blank Drawing Area Showing Two Pages

After creating a new page, you are ready to begin creating a category. Before beginning this practice, you may want to review textbook Chapter 5. Section 5.3.1 discusses generalization hierarchies, while section 5.3.2 provides an explanation about disjointness and completeness constraints. To provide continuity with the textbook, you will create the security generalization hierarchy example from section 5.3.2 of the textbook. Follow the instructions below to complete this exercise.

1. Create Entities and Add Columns: Create the *Security*, the *Stock*, and the *Bond* entities. Note that the last two entities do not have primary keys since they will be inherited from the *Security* entity. When you are finished, the entities should appear as shown in Figure 54.
 - Columns for the *Security* Entity: Add three columns named *Symbol* (primary key, Text, Fixed Length, 10 characters), *SecName* (Text, Variable Length, 100 characters), and *LastClose* (Numeric, Money, Large).
 - Columns for the *Stock* Entity: Add two columns named *OutShares* (Numeric, Unsigned Integer, Large) and *IssuedShares* (Numeric, Unsigned Integer, Large).
 - Columns for the *Bond* Entity: Add two columns named *Rate* (Numeric, Floating Point, Large) and *FaceValue* (Numeric, Money, Large).

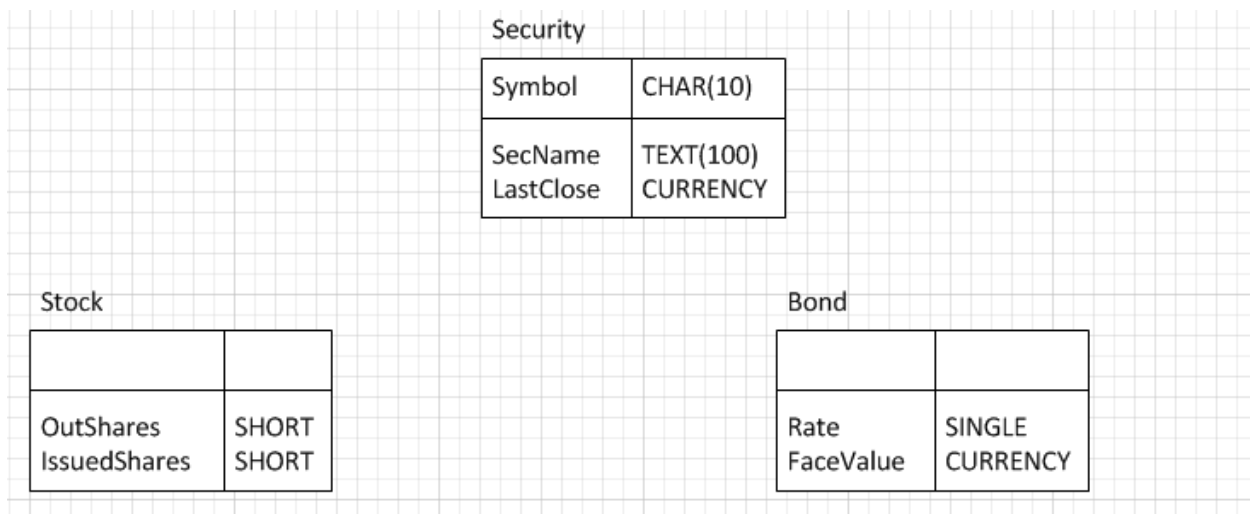




Figure 54: *Security*, *Stock*, and *Bond* Entities with Data Types Displayed

2. Align Entities: Use the the **Auto Align**  **Auto Align** and **Auto Space**  buttons On the **Home** tab, in the **Arrange** group, click **Position** to lay out the entities. Alternatively, you can drag the entities so that the *Security* entity is centered

above the *Stock* entity on the lower left and the *Bond* entity on the lower right, as shown in Figure 55.

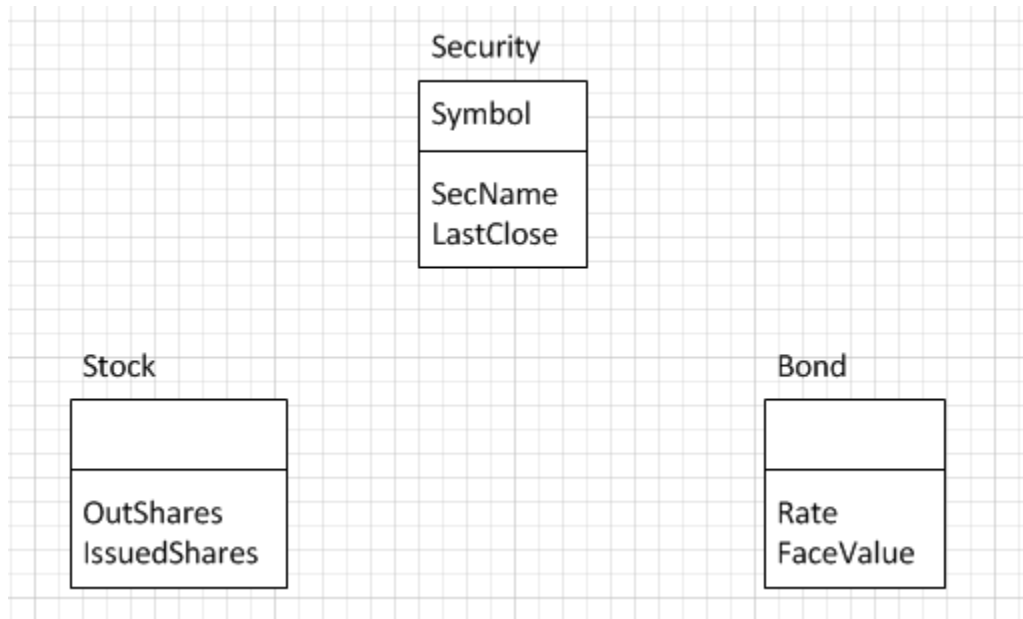


Figure 55: Improved Layout for the *Security*, the *Stock*, and the *Bond* Entities


3. Create Category: Drag the **Category** shape  **Category** from the stencil to the drawing area. If the parent entity has a discriminating attribute, you can double-click on the **Category** shape to set the *Discriminator* property as shown in Figure 56. After you close the Database Properties window, align the category so that it appears as shown in Figure 57. The double lines below the circle mean that the category is complete. Incomplete categories only have a single line below the circle.



Figure 56: Database Properties Window for the Category

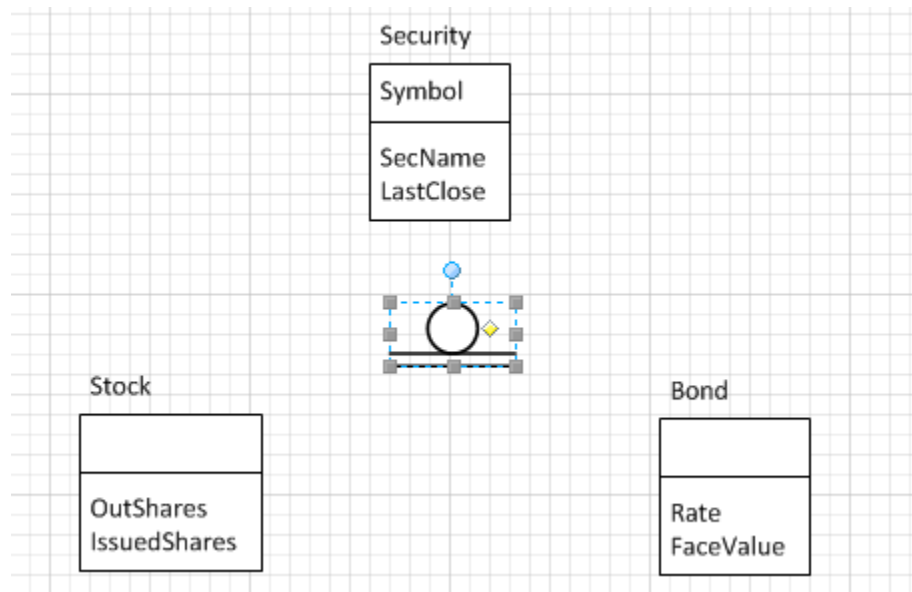



Figure 57: ERD with an Unconnected Complete Category

4. Connect Category to Parent: Drag the **Parent to Category** shape  **Parent to category** from the stencil and drop it in the drawing area. After selecting the **Parent to Category** shape in the drawing area, drag one end to the *Security* entity. Release the mouse when the entity turns red. Drag the other end of the **Parent to Category** shape to the **Category** shape. Release the mouse when the **Category** shape turns red. When you are finished, the ERD appears as shown in Figure 58.

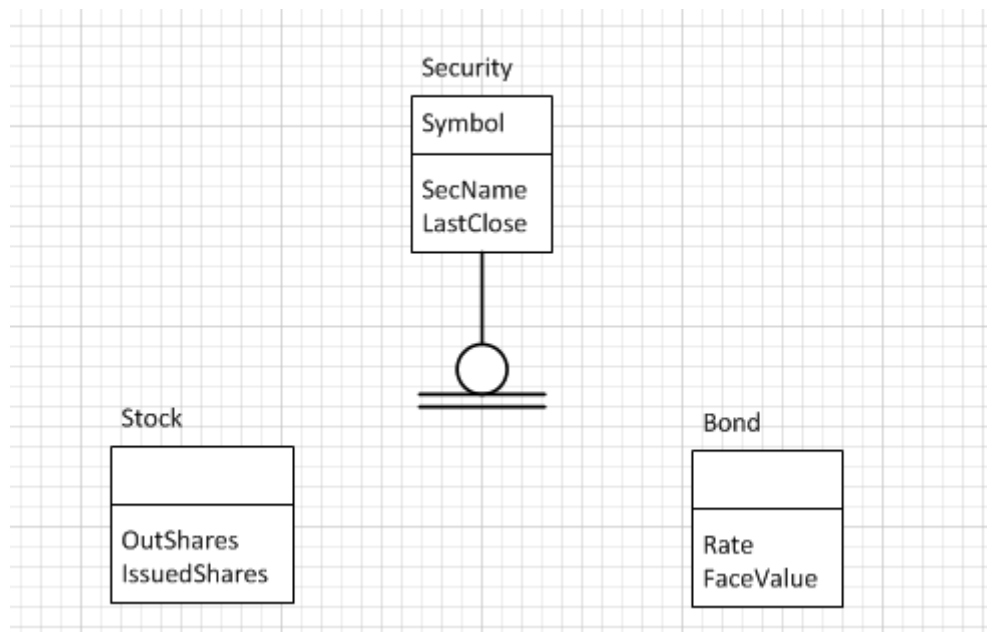


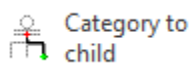
Figure 58: ERD with Category Connected to the Parent Entity Type

5. Examine the Properties of the Parent Connection: Double-click on the line connecting the category to the parent entity. The Database Properties window for the connection opens as shown in Figure 59. In this window, you can specify the discriminating column in the parent entity. Since discriminating columns are optional, one was not defined in the *Security* entity type. Close the Database Properties window to continue with the exercise.



Figure 59: Database Properties Window for the Category-Parent Connection

6. Connect the Category to the *Stock* Entity: Drag the **Category to Child** shape



from the stencil and drop it in the drawing area. After selecting the

Category to Child shape in the drawing area, drag one end to the **Category** shape. Release

the mouse when the **Category** shape turns red. Drag the other end of the **Category to Child** shape to the *Stock* entity. Release the mouse when the entity shape turns red. When you are finished, the ERD appears as shown in Figure 60. Notice that the primary key of the *Security* entity is now the primary key of the *Stock* entity.

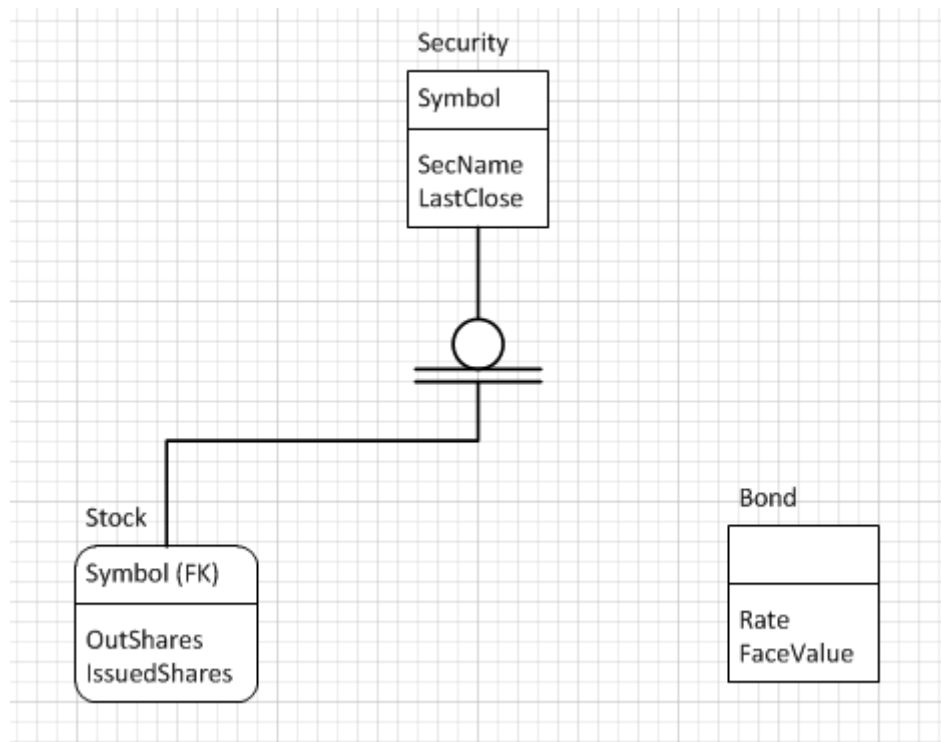


Figure 60: ERD with Category Connected to the *Stock* Entity Type

7. Set the Properties for the Category-to-Stock-Entity Connection: Double-click on the line connecting the category to the *Stock* entity. The Database Properties window for the connection opens. In the window, specify the *Verb phrase*: “ISA”, the *Inverse phrase*: “HasCategory”, and the *Cardinality*: “Zero or one”. The *Relationship type* already should be set to “Identifying”.
8. Connect the Category to the Bond Entity and Set Properties: Perform the same actions for the *Bond* entity as for the *Stock* entity. You need to connect the **Category** shape to the *Bond*

entity using the directions in step 6. Then set the database properties as described in step 7.

When you are finished, the ERD appears as shown in Figure 61.

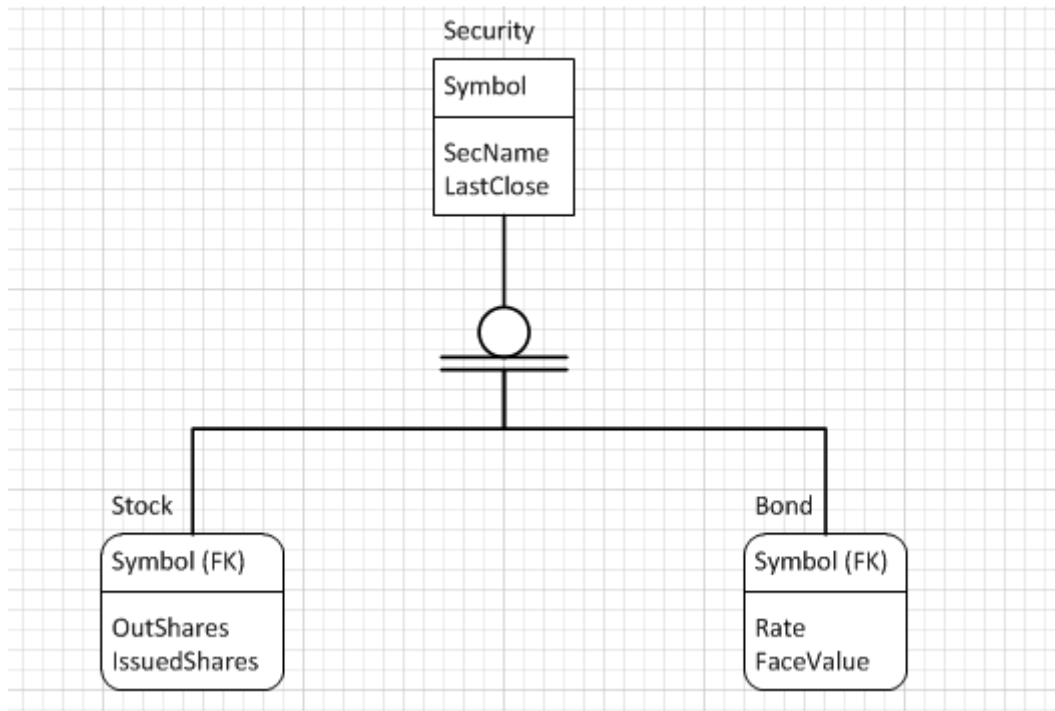


Figure 61: Completed ERD with Category Connected to All Entity Types

9. Save the Diagram: When finished, save the diagram.

6 Reverse Engineering Relational Databases

Visio Professional supports conversion from a relational database to an ERD in the IDEF1X notation. This process is useful to document and reengineer an existing relational database. Many relational database management systems do not have good drawing and documentation tools. Reverse engineering allows you to take advantage of the features of Visio Professional without recreating the database diagram. The automated layout feature can be especially useful for large database diagrams.

A previous version of Visio Professional (Visio Professional 5) provided the Create Database Wizard to convert from a Visio ERD to a relational database. Unfortunately, the Create Database Wizard is now packaged with Visio 2000 Enterprise Edition, a much more expensive product than Visio 2000 Professional Edition. Instead Visio 2000 Professional Edition has replaced the Create Database Wizard with a considerably less useful feature known as the Refresh Wizard. The Refresh Wizard compares an ERD to an existing relational database. For each difference, the Refresh Wizard can change the ERD to be consistent with the relational database. Because refreshing an ERD is useful only in limited situations, the Refresh Wizard is not presented in this section.

6.1 Converting a Relational Database to a Visio ERD

The reverse conversion process uses the Reverse Engineer Wizard. The Reverse Engineer Wizard can convert ODBC-compliant databases into Visio ERDs formatted according to either the IDEF1X or the Object Relational stencil. With any reverse conversion, the resulting ERD may not be complete because of the differences between the Relational Model and the Entity Relationship Model. After the conversion, the ERD lacks relationship names and some minimum cardinalities. You will need to add these missing elements to complete your ERD. In addition, SQL2-compliant relational databases lack generalization hierarchies that are found in the ERD notations. These omissions are not a drawback of the Reverse Engineer Wizard but of the differences between the Relational Model and the Entity Relationship Model. Note that generalization hierarchy differences may disappear as DBMSs adopt SQL3, the new relational database standard. Textbook Chapter 19 describes the generalization hierarchy features of SQL:2011.

To begin you should open a blank database drawing page with the Entity Relationship stencil on the left. Select the **Database → Reverse Engineer...** command and then use the following instructions to reverse engineer the auto repair database, an Access 2007 database. You should download this database from the web page containing this lab chapter before you begin the instructions. Please remember the local directory location of the database after downloading it.

1. First Wizard Window: The first window (Figure 62) explains the reverse engineering process and allows you to select the database driver. Choose the settings as shown in Figure 62.
2. Connect Data Source Window: After clicking the **Next** button, the Connect Data Source window appears. You can enter the user name and password of the database administrator. Since no administrator has been defined for this database, you can leave the settings blank as shown in Figure 63.
3. Locate the Database: After clicking the **OK** button, you can locate the database in the Select Database window (Figure 64). Navigate in this window until you find the location of the auto repair database that you downloaded before beginning these instructions.
4. Second Wizard Window: After you successfully locate the database, the second wizard window (Figure 65) appears. This window allows you to indicate what parts of the database to convert. Choose the selections shown in Figure 65 and click the **Next** button.
5. Third Wizard Window: The third window (Figure 66) asks you to choose the database tables to convert. Click the **Select All** button to select all of the tables in the list. Click the **Next** button to continue.

6. Final Wizard Window: The final window presents the choices that you previously made. The final wizard window should appear as shown in Figure 67 except for the location of the database. If it does not, click the **Back** button to make corrections; otherwise, click **Finish**.
7. View the Output and the Tables Windows: After the Reverse Engineer Wizard finishes, the Output and the Tables windows appear in the drawing area (Figure 68). The Output window lists the actions taken by the Reverse Engineer Wizard. The Tables window shows the tables converted from the relational database to entities. Close the Output window when finished reading it.
8. Lay out and View the New ERD: Select all entities in the Tables window. Drag and drop them to the drawing area. Visio Professional automatically lays out the ERD. Close the Tables window and enlarge the drawing area to see the new ERD. The new ERD created from the auto repair database appears as shown in Figure 69. The relationship names are default names generated by the Reverse Engineer Wizard. To complete the ERD, you should change the relationship names and examine the minimum cardinalities on the “many” sides of relationships. You also may want to use the automatic layout feature to achieve a more readable layout and consider using categories to capture implicit generalization hierarchies in the relational database.
9. Save the Diagram: When finished, save this diagram as “AutoRep2.vsd”.

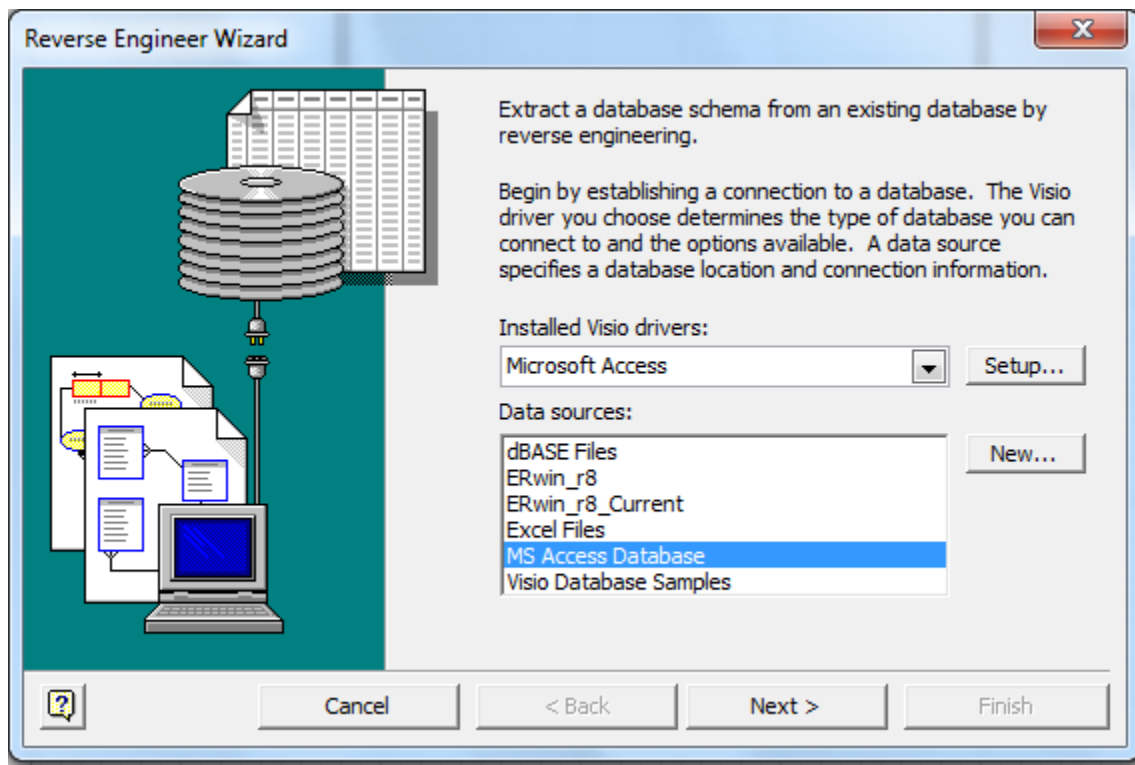


Figure 62: First Window of the Reverse Engineer Wizard

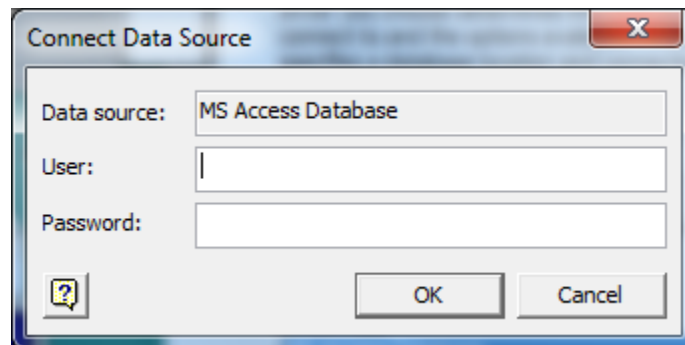


Figure 63: Connect Data Source Window

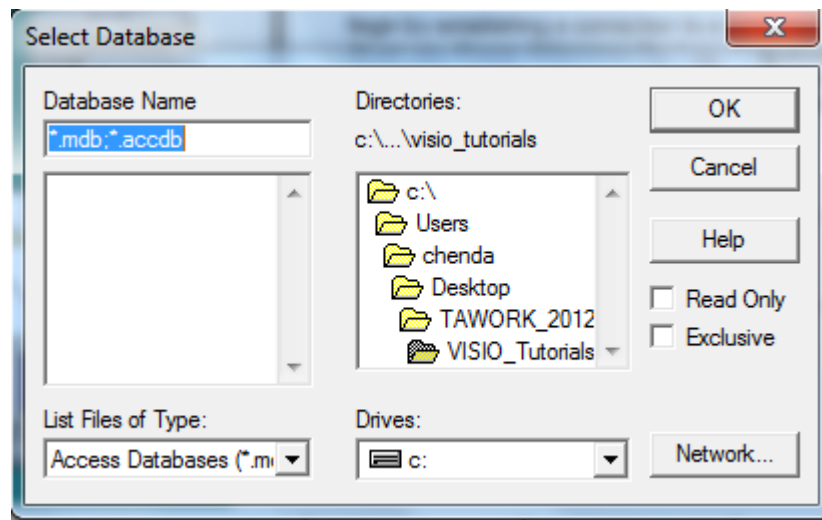


Figure 64: Select Database Window

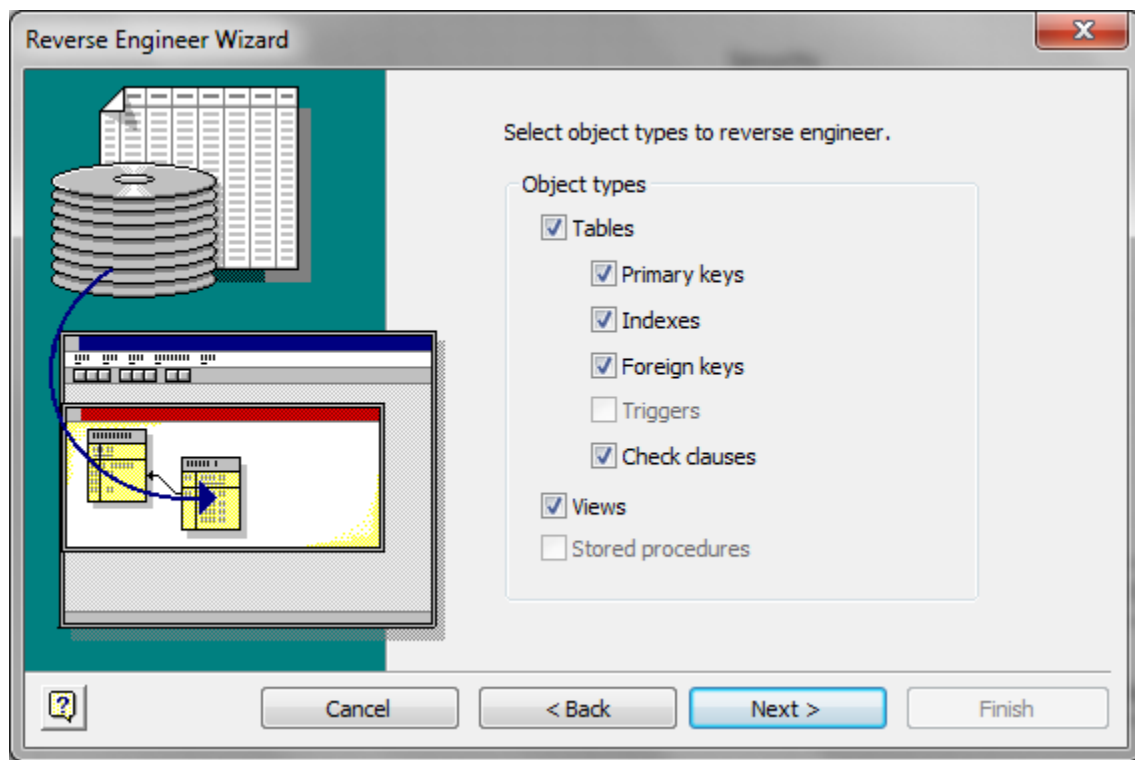


Figure 65: Second Window of the Reverse Engineer Wizard

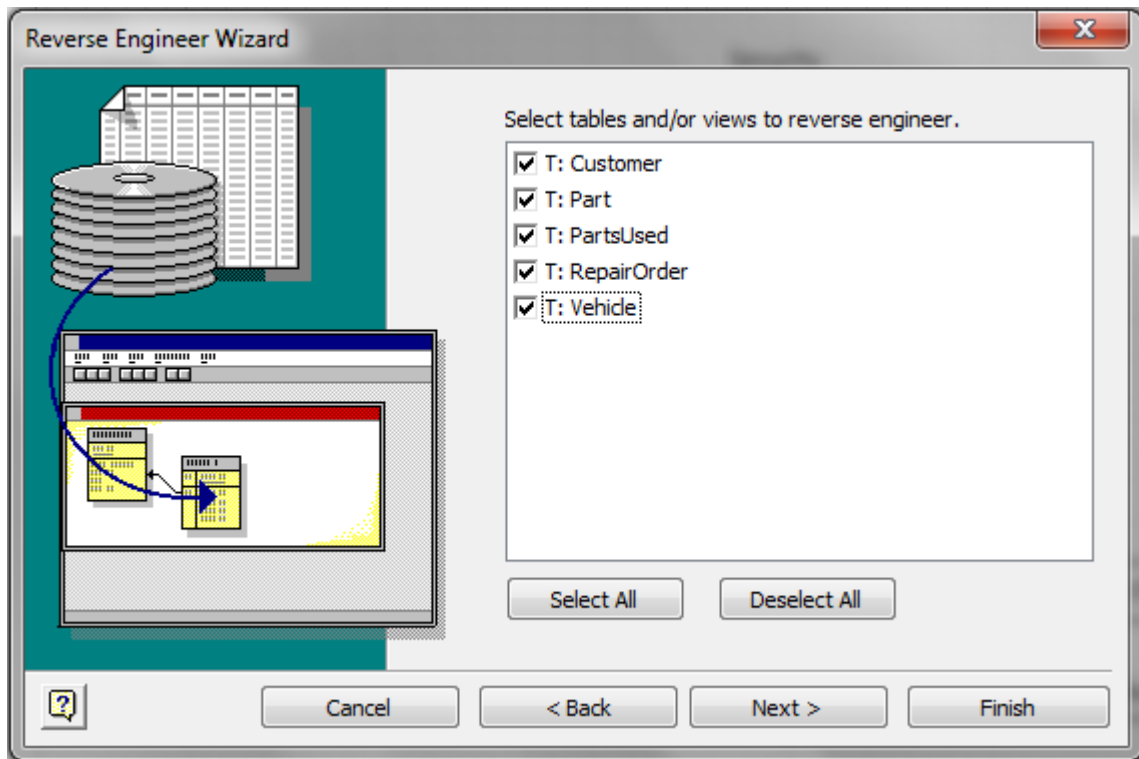


Figure 66: Third Window of the Reverse Engineer Wizard

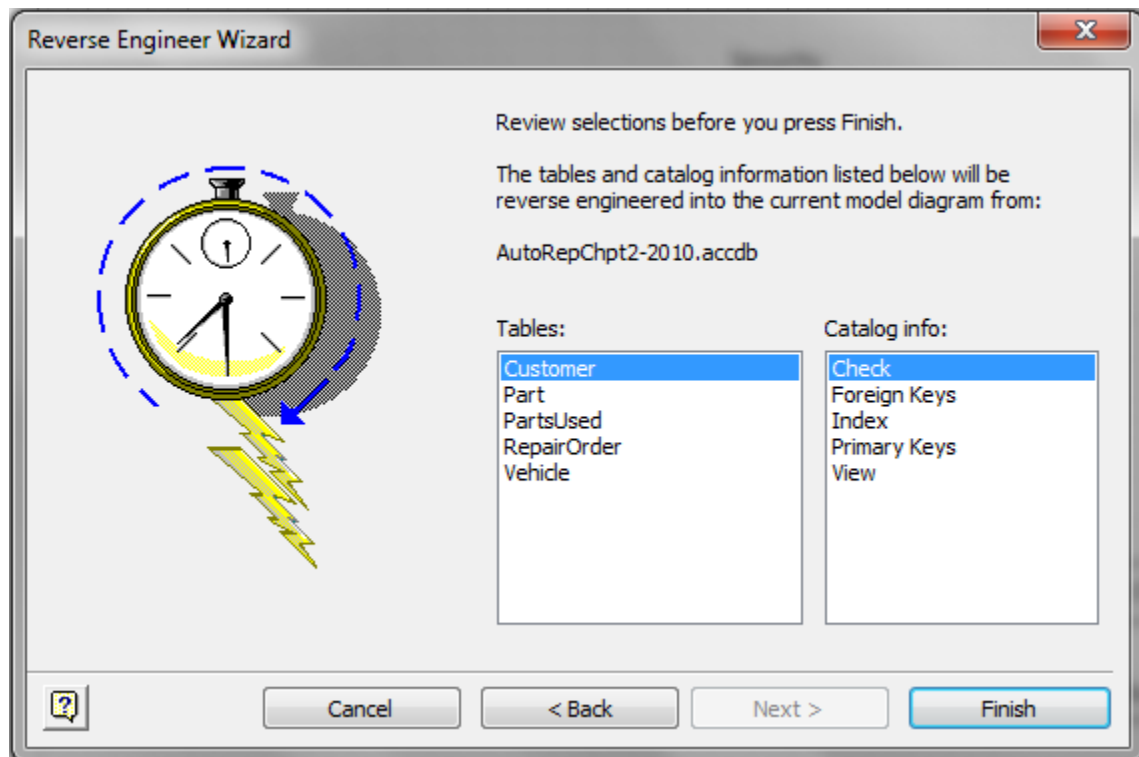


Figure 67: Final Window of the Reverse Engineer Wizard

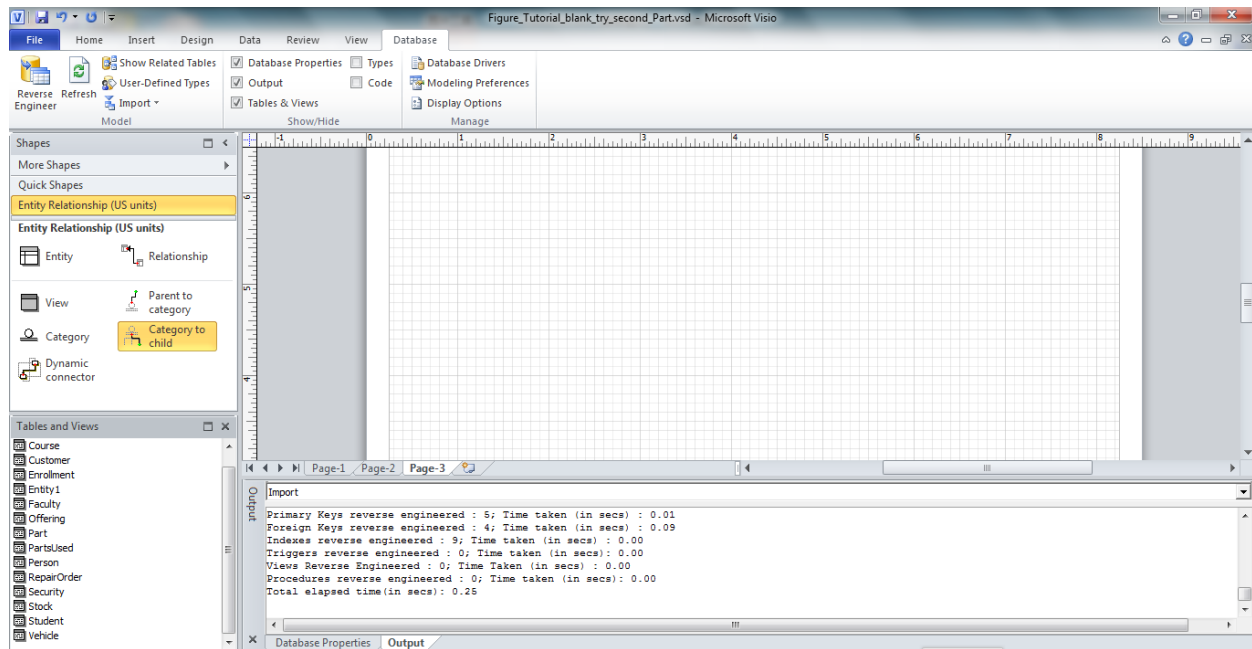


Figure 68: Output and Tables Windows in the Drawing Area

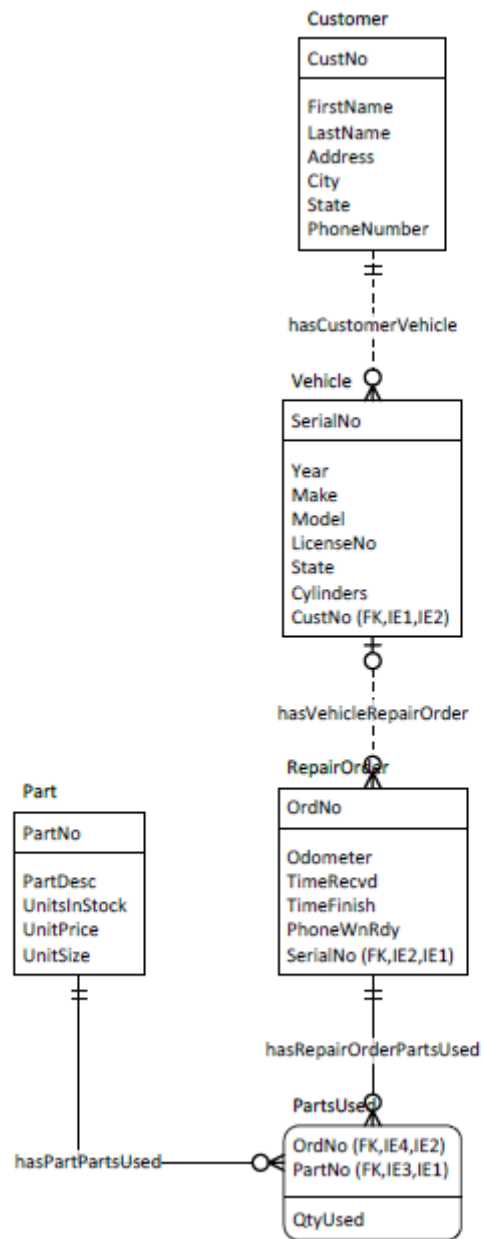


Figure 69: ERD for the Auto Repair Database after Conversion

Closing Thoughts

This chapter has provided guided instruction for creating ERDs using Visio 2010 Professional Edition. Visio Professional provides a number of convenient tools to support the conceptual data modeling and logical design phases of database development. To support

conceptual data modeling, Visio Professional provides stencils (collection of shapes and definition windows) for popular ERD notations such as the IDEF1X and the Object Relational notations. You used the IDEF1X notation with the crow's foot cardinality symbols to create ERDs with entities, 1-M relationships, weak entities and associated identifying relationships, self-referencing relationships, and generalization hierarchies. The predefined shapes and definition windows of the stencil and the automated layout feature are much easier to use than manually creating shapes with general-purpose drawing tools. To support the logical database design phase of database development, Visio Professional provides a wizard to reverse engineer an existing relational database. You used the wizard to convert the Auto Repair database to an ERD.

Chapter Reference

The chapter reference summarizes procedures that you practiced. For wizards discussed in the chapter, the procedures highlight important parts of the wizards but do not list all of the steps.

Procedure 1: Create a New Database Drawing (Section 4.1)

1. When you open Visio Professional, a window appears giving you the option of creating a new drawing or opening an existing file. Select "Database Model Diagram" and click the **Create** button. A new drawing opens with the Entity Relationship and the Object Relational stencils positioned on the left of the workspace.
2. If Visio Professional is already opened on an existing diagram, choose **File** → **New**. Choose **Database Model Diagram** from the list of recently used templates.. A new drawing opens with the Entity Relationship and the Object Relational stencils positioned on the left of the workspace.

Procedure 2: Create and Define an Entity (Section 4.2)

1. Drag the **Entity** shape from the stencil to the work area on the right.
2. Use the Database Properties window to name the entity and add the columns. You can open the window by double-clicking the entity or by right-clicking the entity and

selecting **Database Properties...** from the pop-up menu.

3. When the Database Properties window first opens, it will default to the **Definition** area to enter the name of the entity.
4. To define columns, click on the **Columns** tab in the Database Properties window.
5. Use the **Add** button to add each column. Use the **Edit** button to specify column properties such as the data type category, type, size, length, precision, and constraints.

Procedure 3: Create and Define a Relationship (Section 4.3)

1. To create a relationship, you can use the **Relationship** shape in the stencil. With either method, the entities must be created before the relationship is created.
2. To use the **Relationship** master shape, drag and drop the **Relationship** shape from the stencil to the drawing area. Then select the relationship line and drag the “one” side of the relationship line to the parent entity. Release the mouse when the parent entity turns red. Repeat this procedure for the child entity, this time dragging the “many” side of the relationship line.
3. To define the relationship properties, double-click on the relationship to open the Database Properties window. You can define the relationship name and cardinalities in the window.

Procedure 4: Set the Display Options (Section 4.3)

1. Choose **Database** → **Display Options** to open the Database Document Options window.
2. In the **General** area, you can define the symbol set (“IDEFIX” or “Relational”) and the names visible on the diagram (conceptual or physical).
3. In the **Table** area, you can define the display options for a table. The typical options are to display the primary key and nonkey columns. You also can display foreign keys, alternate keys, annotations, and data types.
4. In the **Relationship** area, you can indicate the cardinality symbols (crow’s foot or IDEFIX endings) and relationship names (forward text or reverse text) displayed. Normally the forward text is displayed because it corresponds to the active verb phrase.

Procedure 5: Make a Solid Color for the Text Background (Section 4.3)

1. If the relationship line obscures the relationship name, you should use a solid color for the text background.
2. After right-clicking the relationship, choose the **Format** → **Text...** command. In the **Text Block** area of the Text window, use a solid color for the text background.

Procedure 6: Create a Self-Referencing Relationship (Section 5.1)

1. You need to use the **Relationship** shape in the Entity Relationship stencil instead of the **Connector** button.
2. You need to add a column in the entity to represent the self-referencing relationship.
3. Drag and drop the **Relationship** shape from the stencil to the drawing area. Drag and drop the “one” side of the relationship to the entity. Drag and drop the “many” side of the relationship to the same entity. You can adjust the appearance of the relationship by stretching it horizontally and vertically.
4. Define the relationship in the usual manner (name and cardinalities) using the Database Properties window.
5. In the **Definition** area, connect the primary key of the left entity to the foreign key of the right entity added in step 2 above. You use the **Associate** button in the **Definition** area of the Database Properties window to make the connection.

Procedure 7: Create Multiple Relationships between the Same Entities (Section 5.1)

1. There are a number of reasons for having multiple relationships between the same entities such as the indirect representation of an M-N self-referencing relationship.
2. Add a column in the entity on the “many” side of the relationship. The column will be used to link the entities in one of the relationships.
3. The first relationship links automatically from the parent entity to the child entity. The linking column in the child entity is assumed to be the same name as the primary key in the parent entity.
4. The second relationship links from the primary key of the parent entity to the new column in the child entity. To link the entities in the second relationship, you use the **Associate** button in the **Definition** area of the Database Properties window for the relationship.

Procedure 8: Create a Category (Section 5.2)

1. In the IDEF1X notation, a generalization hierarchy is known as a category.
2. Create all the entity types in the category before defining the category. In the parent entity, you can define a discriminating column if desired.
3. Drag and drop the **Category** shape from the Entity Relationship stencil to the drawing area.
4. Drag and drop the **Parent to Category** shape from the stencil to the drawing area. After selecting the **Parent to Category** shape in the drawing area, drag one end to the parent entity. Release the mouse when the **Entity** shape turns red. Drag the other end of the **Parent to Category** shape to the **Category** shape. Release the mouse when the **Category** shape turns red.
5. If the parent entity contains a discriminating column, associate the discriminating column with the category in the Database Properties window for the category.
6. Drag and drop the **Category to Child** shape from the stencil to the drawing area. For each child entity, after selecting the **Category to Child** shape in the drawing area, drag one end to the **Category** shape. Release the mouse when the **Category** shape turns red. Drag the other end of the **Category to Child** shape to the child entity. Release the mouse when the **Entity** shape turns red.

Additional Practice

The following problems provide additional practice with the extended auto repair database as well as the textbook databases.

Part 1: Database Development for this Lab Chapter's ERDs

1. Add entities and relationships to the AutoRep ERD:

- Create a new drawing page in the AutoRep ERD. Copy the original AutoRep ERD from page 1 to page 2.
- Add an entity called "Labor" with attributes labor code (*LabCode*), labor description (*LabDesc*), hourly rate (*HourRate*), and standard time (*StdTime*), representing the standard number of hours to complete the labor.
- Add an entity called "LaborUsed" with the attribute hours worked (*Hours*).

- Add an identifying 1-M relationship from the *RepairOrder* entity to the *LaborUsed* entity.
 - Add an identifying 1-M relationship from the *Labor* entity to the *LaborUsed* entity.
 - Use the automated layout feature to improve the layout.
2. Add another level of generalization hierarchy to the security example in Section 5.2. Add the *Common* and the *Preferred* entities as subtypes of the *Stock* entity as depicted in textbook Chapter 5. The *Common* entity has columns *PERatio* and *Dividends*. The *Preferred* entity has columns *CallPrice* and *Arrears*. The category is complete. You can add a discriminating column in the *Stock* entity if you wish.
 3. Use the Reverse Engineer Wizard to convert the extended auto repair database to a crow's foot ERD. The extended auto repair database will be provided on the textbook's website. After converting the database, revise the relationship names, the cardinalities, and the layout.

Part 2: Database Development for the University Database in Textbook Chapter 5

1. Create an ERD for the university database used in textbook Chapter 5. Use the automated layout feature to adjust the layout.
2. Use the Reverse Engineer Wizard to convert the university database from textbook Chapter 5 to an ERD. After converting the database, revise the relationship names, cardinalities, and diagram layout.

Part 3: Database Development for the Order Entry Database in Textbook Chapter 5

1. Create an ERD for the order entry database used in textbook Chapter 5. Use the automated layout feature to adjust the layout.

2. Use the Reverse Engineer Wizard to convert the order entry database from textbook Chapter 5 to a crow's foot ERD. After converting the database, revise the relationship names, cardinalities, and diagram layout.

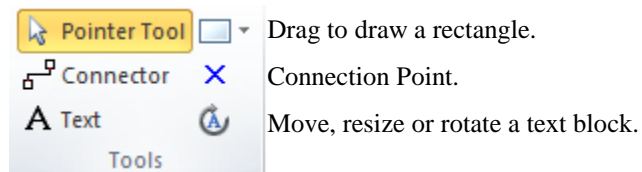
Part 4: Data Modeling and Conversion Problems in Textbook Chapter 5

Use the Entity Relationship stencil to work the examples and problems in textbook Chapter 5. Compare the representation in textbook Chapter 5 notation with the Entity Relationship stencil in Visio 2010.

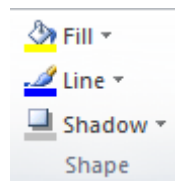
Appendix A: Summary of Visio-Specific Toolbars

This appendix shows the toolbars that are specific to Visio Professional. Toolbars common to most Windows applications are not shown.

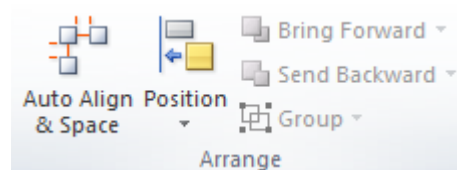
Tools Toolbar



Shape Toolbar



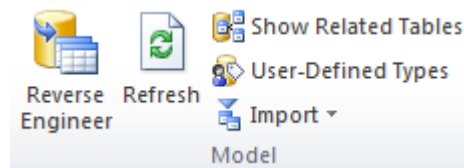
Arrange Toolbar



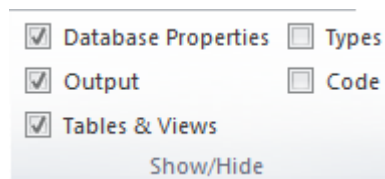
Editing Toolbar



Model Toolbar



Show/Hide Toolbar



Manage Toolbar

