

# Appendix A1

## Designing Databases with Visio Professional 2010: A Tutorial

**Microsoft Visio Professional is a powerful database design and modeling tool. The Visio software has so many features that it is impossible to demonstrate all of them in this short tutorial. However, you will learn how to:**

- Start Visio Professional.
- Select the Crow's Foot entity relationship diagram (ERD) option.
- Create the entities and define their components.
- Create the relationships between the entities and define the nature of those relationships.
- Edit the Crow's Foot ERDs.
- Insert text into the design grid and format the text.

### Preview

Once you have learned how to create a Visio Crow's Foot ERD, you will be sufficiently familiar with the basic Visio Professional software features to experiment on your own with other modeling and diagramming options. You will also learn how to insert text into the Visio diagram to document features you consider especially important or to simply provide an explanation of some segment of the ERD.

Note: The screens and instructions in this tutorial are for Microsoft Visio Professional 2010. If you are using an earlier version of Visio, your screen will vary slightly.

### Data Files and Available Formats

[MS Access](#) [Oracle](#) [MS SQL](#) [My SQL](#)

[MS Access](#) [Oracle](#) [MS SQL](#) [My SQL](#)

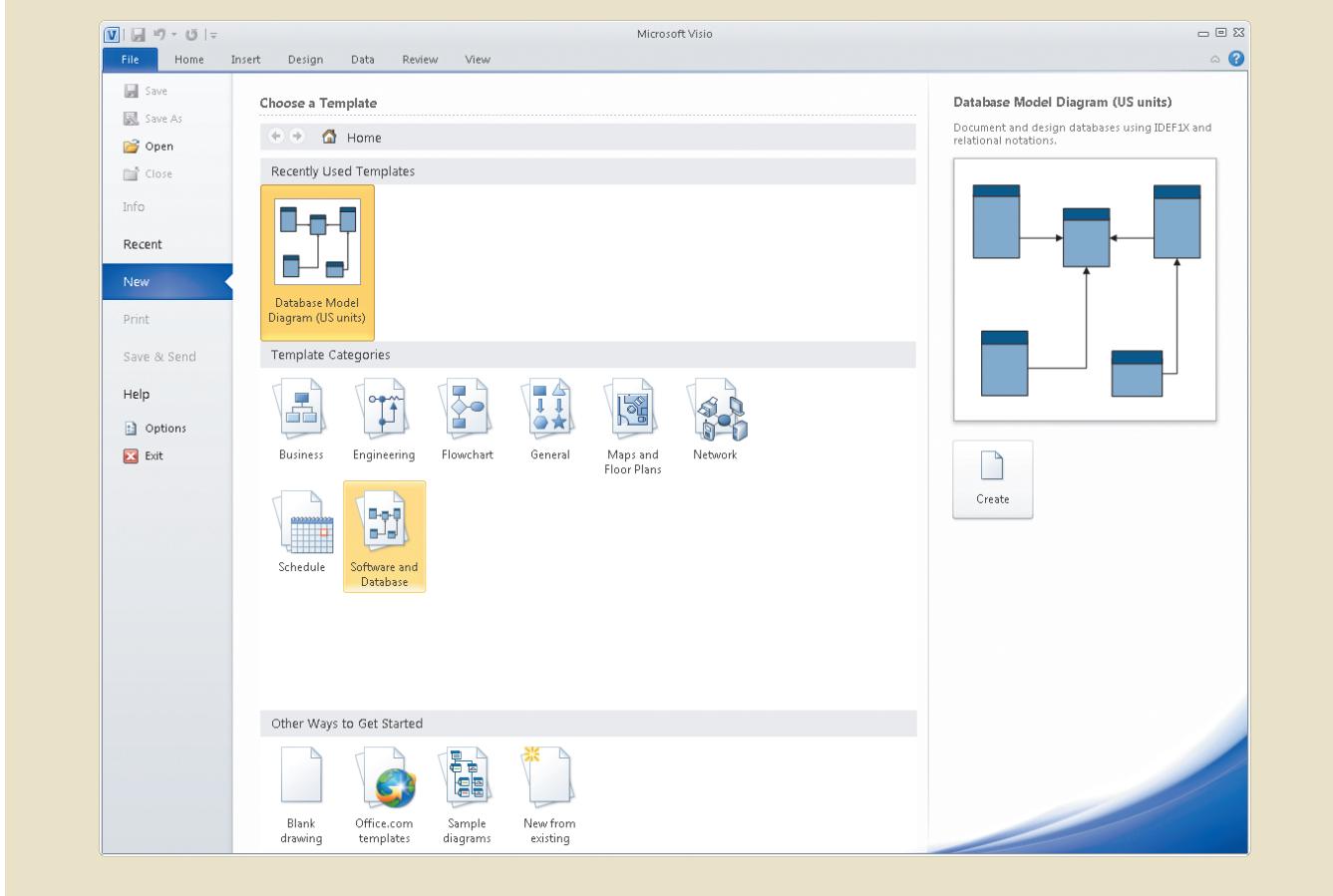
There are no data files for this appendix.

*Data Files Available on [cengagebrain.com](http://cengagebrain.com)*

## A1-1 Starting Visio Professional

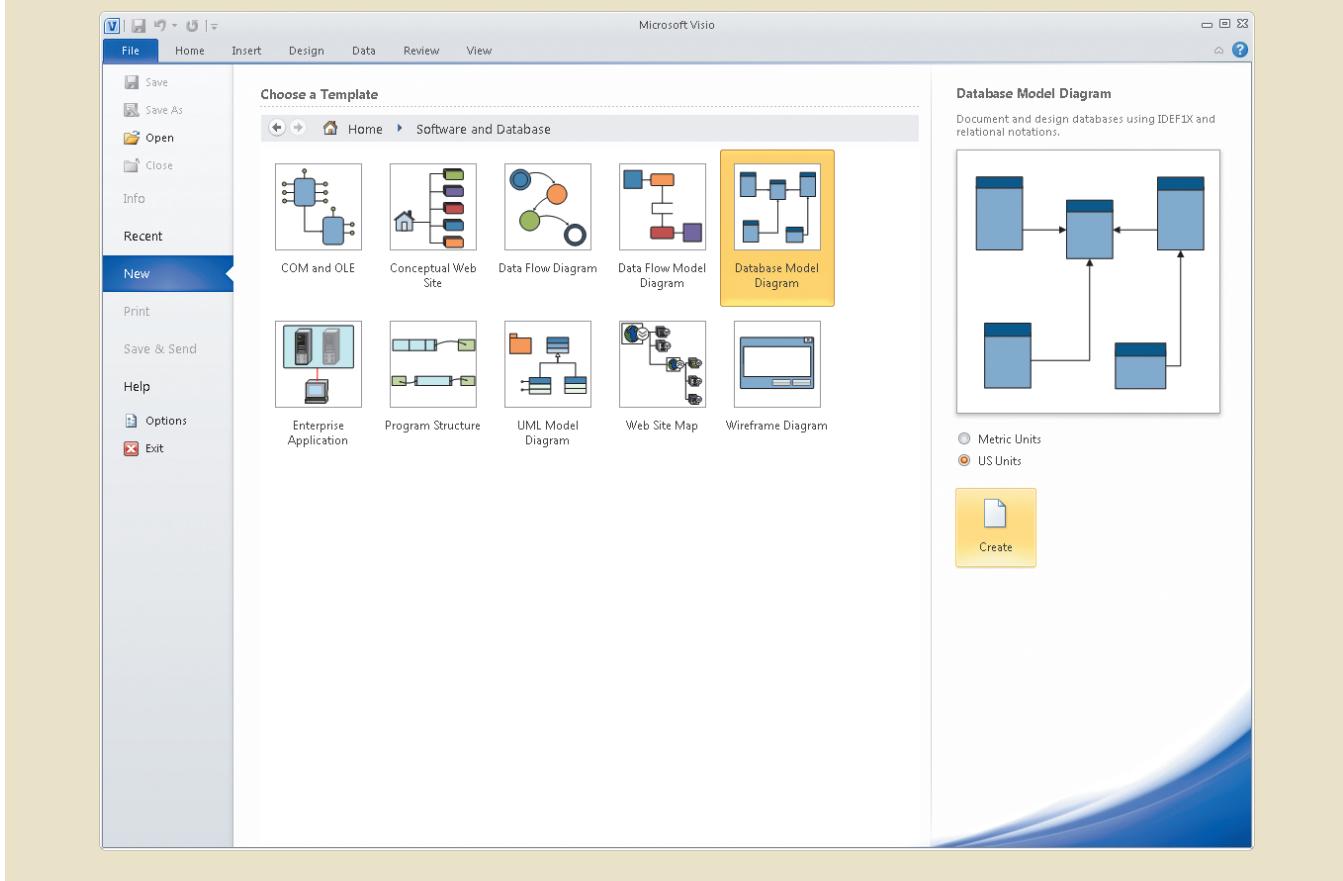
The typical Visio Professional software installation lets you select Visio through the **Start**, **(All) Programs**, **Microsoft Visio** sequence. After the Visio software has been activated, click the **Software and Database** option to match the screen shown in Figure A1.1. (Previously created Visio files show up in the **Recent Documents** header on the right side of the screen.)

FIGURE A1.1 THE VISIO PROFESSIONAL OPENING SCREEN



With the **Software and Database** selection shown in Figure A1.1, select the **Database Model Diagram** object. Note that your selection results in a highlight around the object. As shown in Figure A1.2, you will see the Database Modeling Template description in the right side of your screen.

**FIGURE A1.2 THE DATABASE MODEL OBJECT SELECTION**



Click the **Create** button on the right side of the screen as shown in Figure A1.2 to begin a new diagram, producing the screen shown in Figure A1.3. Because the preference here is for a larger grid than the one shown in Figure A1.3, start by selecting the **View** ribbon at the top of the screen. Click the **Zoom** option to generate the list of size options. Figure A1.4 shows that the **100%** option has been selected. When you click the **100%** selection and click **OK**, the grid expands to fill the screen.

#### A1-4 Appendix A1

FIGURE A1.3 THE DRAWING BOARD

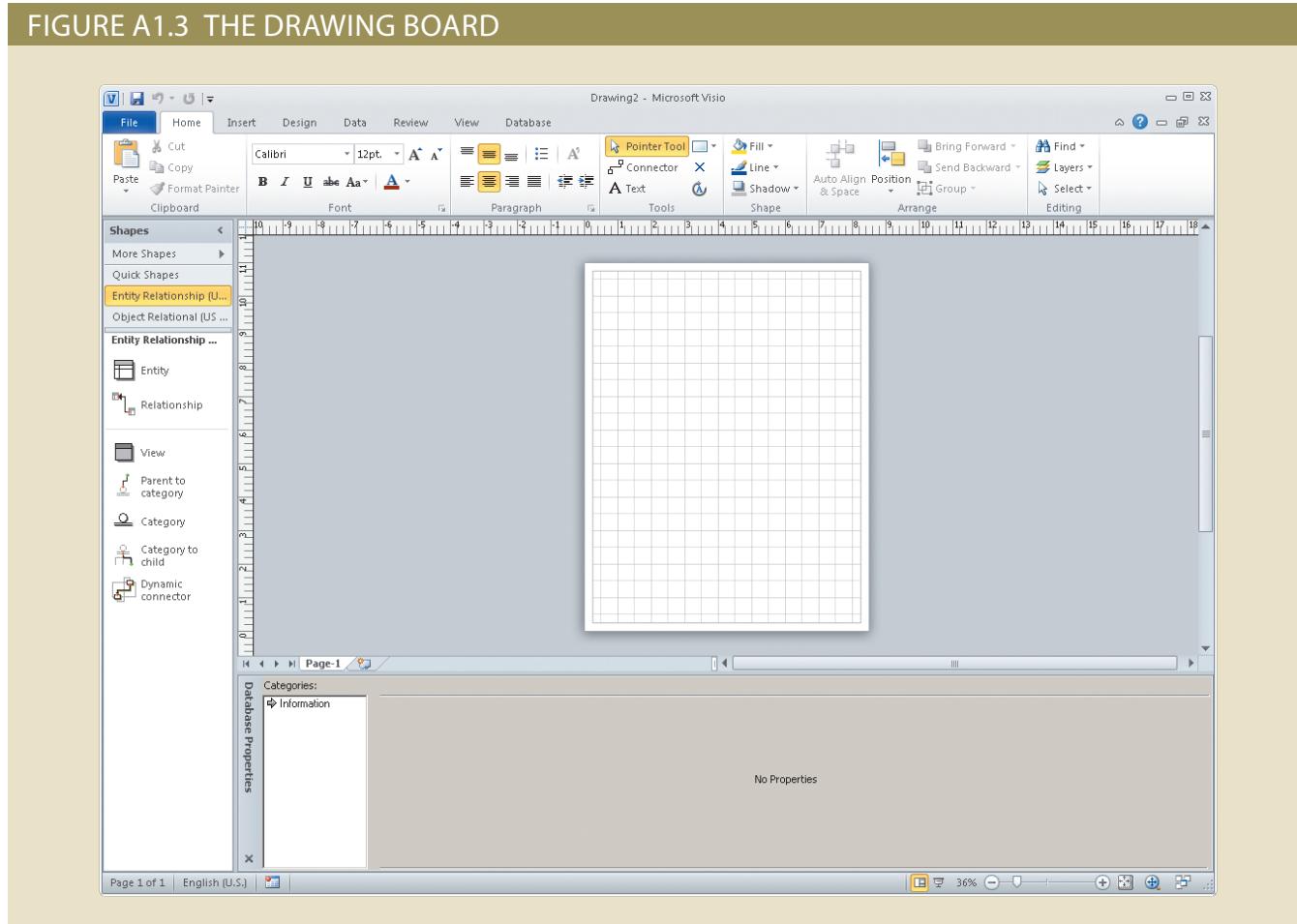
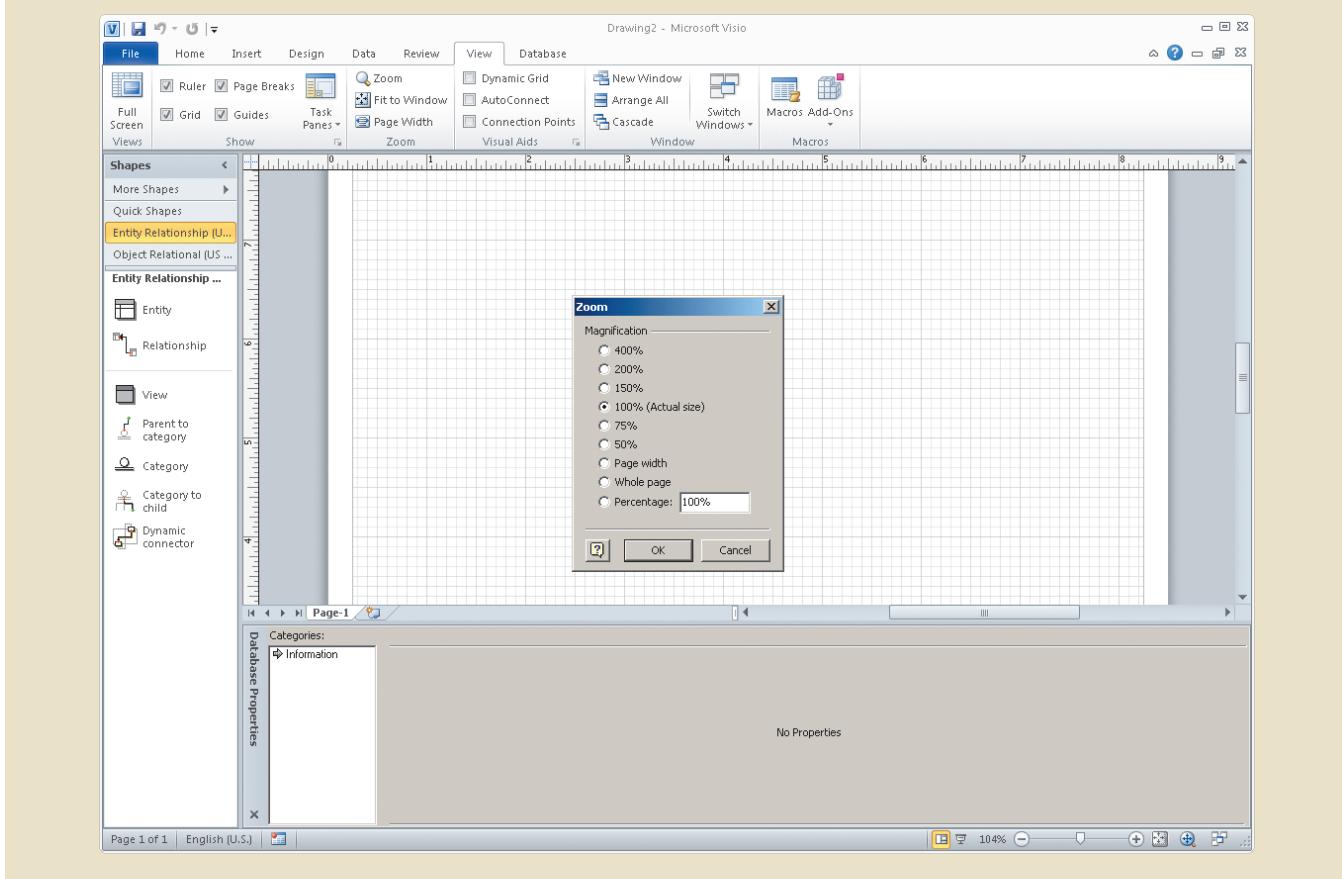


FIGURE A1.4 THE DRAWING BOARD SIZE OPTION



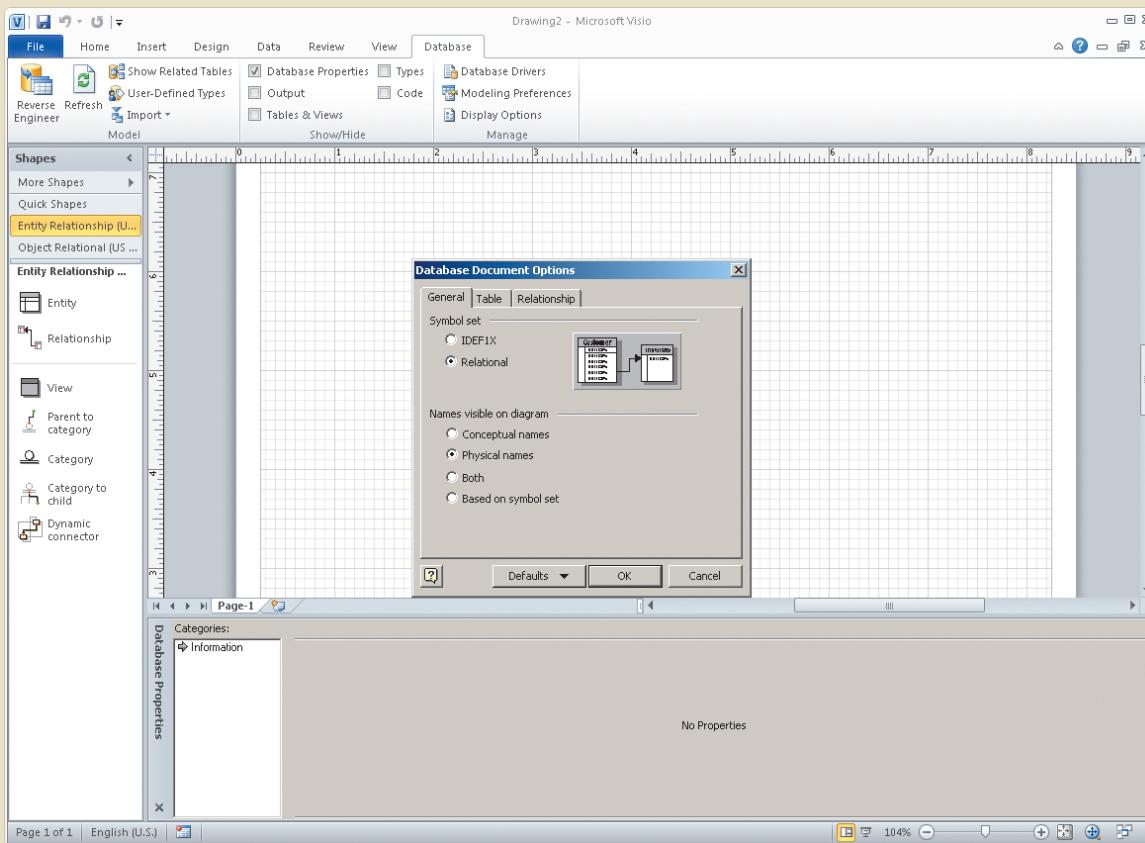
By selecting the Visio Professional database option and its drawing board, you have completed the preliminary work required to create ERDs. You are now ready to draw the ERDs on the drawing board. You will use the Crow's Foot option, the same one used to create all of the ERDs in this text.

## A1-2 Setting The Stage For Creating a Crow's Foot ERD

To select the Crow's Foot option, select the **Database** ribbon, and then click **Display Options** to display the Database Document Options window shown in Figure A1.5. The default selection is the *General* tab. Note that the default selections in the *General* tab are *Relational* and *Physical names*. Ensure that you have these default options selected.

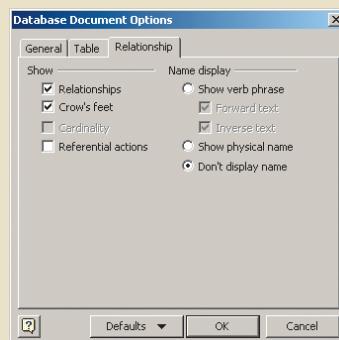
## A1-6 Appendix A1

FIGURE A1.5 THE DATABASE DOCUMENT OPTIONS WINDOW



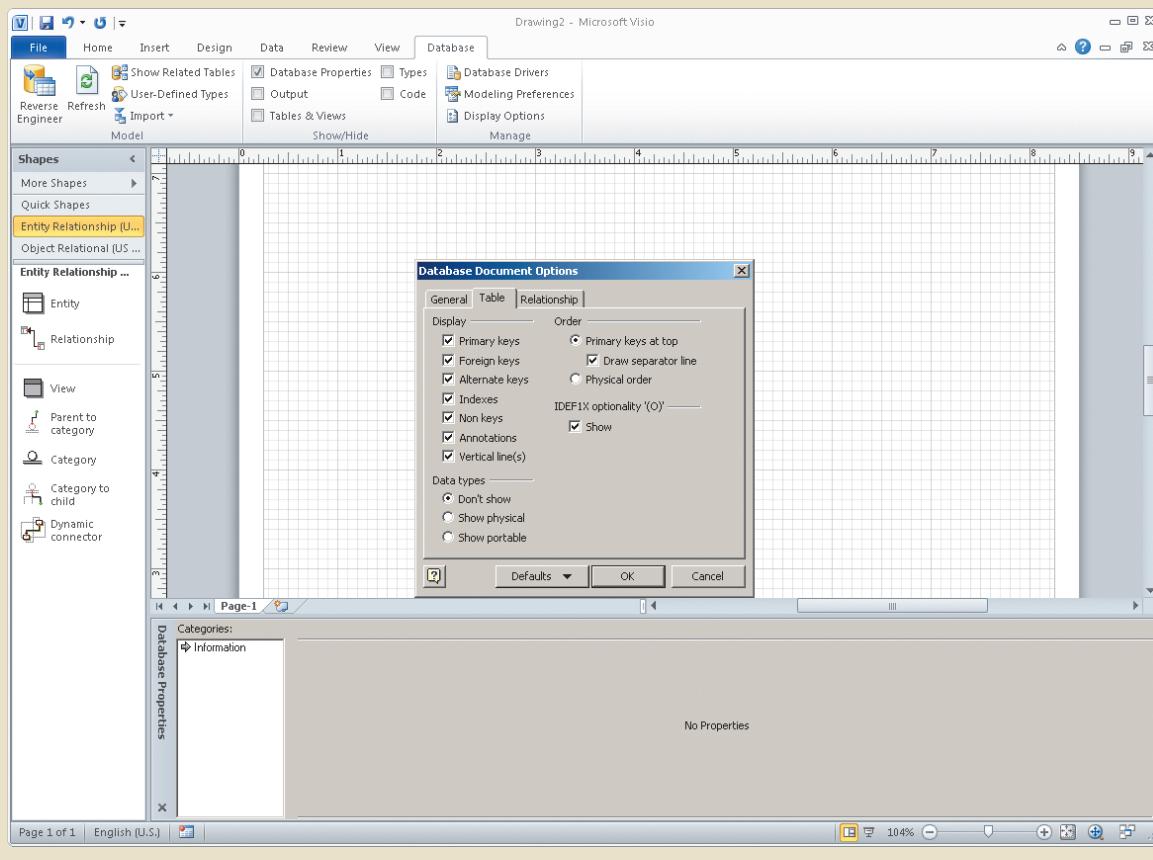
Select the **Relationship** tab and click to select the **Crow's feet** option as shown in Figure A1.6. As you examine the tab, keep in mind that we have not created and named any relationships yet. Therefore, we have not determined *how* the names will be displayed. You will return to this dialog box later to see the effect of name display options and to demonstrate that you can edit the displays when you are working on the relationships.

FIGURE A1.6 THE DATABASE DOCUMENT OPTIONS, RELATIONSHIP TAB



Next, select the **Table** tab in the Database Document Options dialog box, as shown in Figure A1.7. Make sure that the check boxes are marked as shown here, and then click the **OK** button to begin creating Crow's Foot ERDs.

FIGURE A1.7 THE DATABASE DOCUMENT OPTIONS, TABLE TAB



## A1-2a The Business Rules

To illustrate the development of the Visio Professional's Crow's ERD, you will create a simple design based on the following business rules:

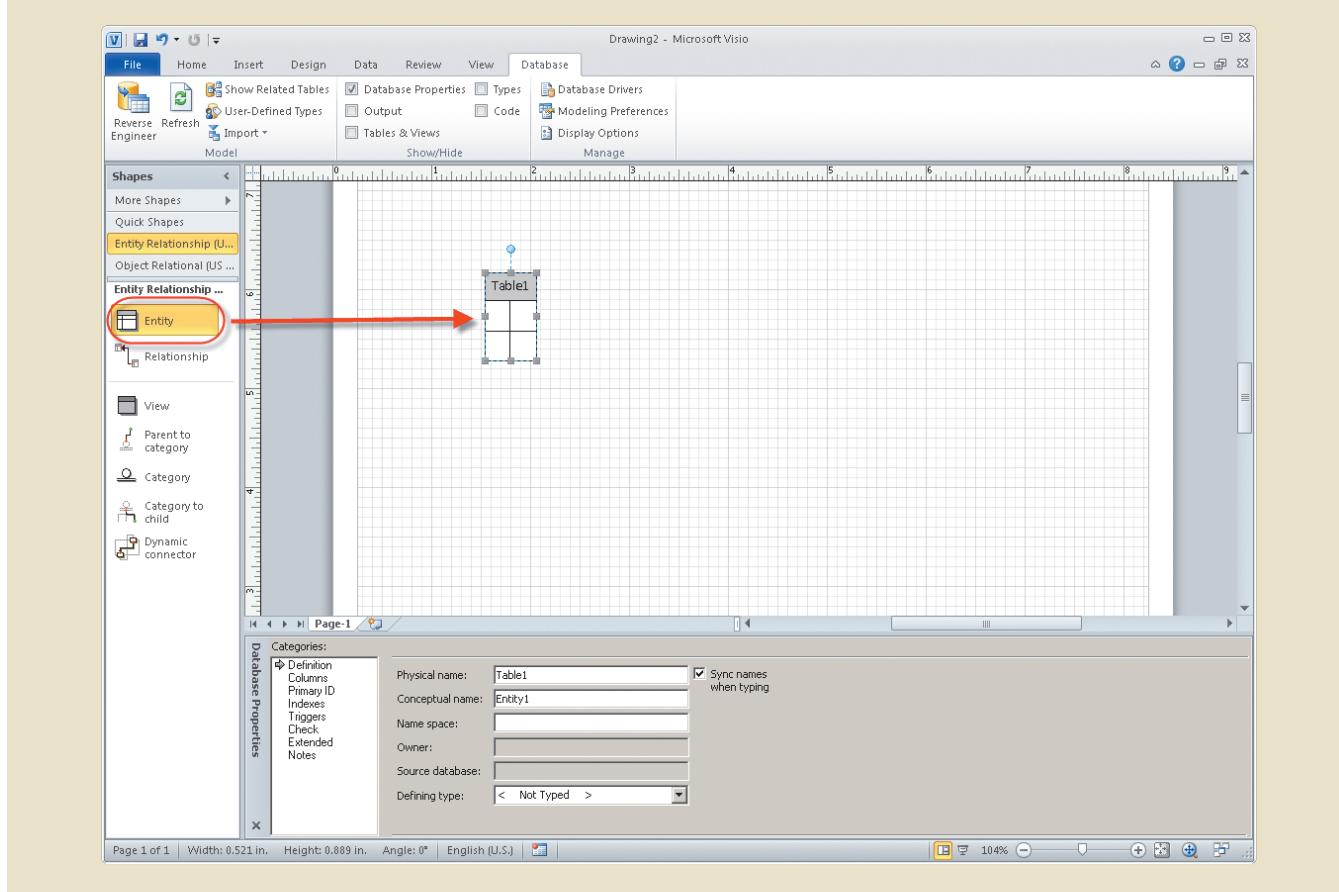
1. A course can generate many classes.
2. Each class is generated by a course.
3. A course may or may not generate a class.

Note that a class has been defined as a section of a course. That definition reflects the real world's use of the labels *class* and *course*. Students have a *class* schedule rather than a section schedule. The catalog that lists all of the courses offered by a department is called a *course catalog*. Some courses are not taught each semester, so they may not generate a class during any given semester.

## A1-3 Creating an Entity

Now that you have some idea of the proposed design components, let's create the first entity for the design. Click the **Entity** object shown in Figure A1.8. (It is circled in the figure.) Drag the **Entity** object to the grid and then drop it. That action will produce the **Table1** object shown in the grid in Figure A1.8. (The **1** in the **Table1** label indicates that this is the first entity object to be placed on the grid.) Note that the entity object is shown as a table.

**FIGURE A1.8 PLACING THE ENTITY OBJECT IN THE GRID**



As you examine Figure A1.8, note that the small “locks” around the **Table1** object perimeter indicate that the **Table1** object has been selected. You can deselect the object by clicking an empty portion of the grid. If the **Table1** object has not been selected, click it to select it.

## A1-3a The Database Properties Window

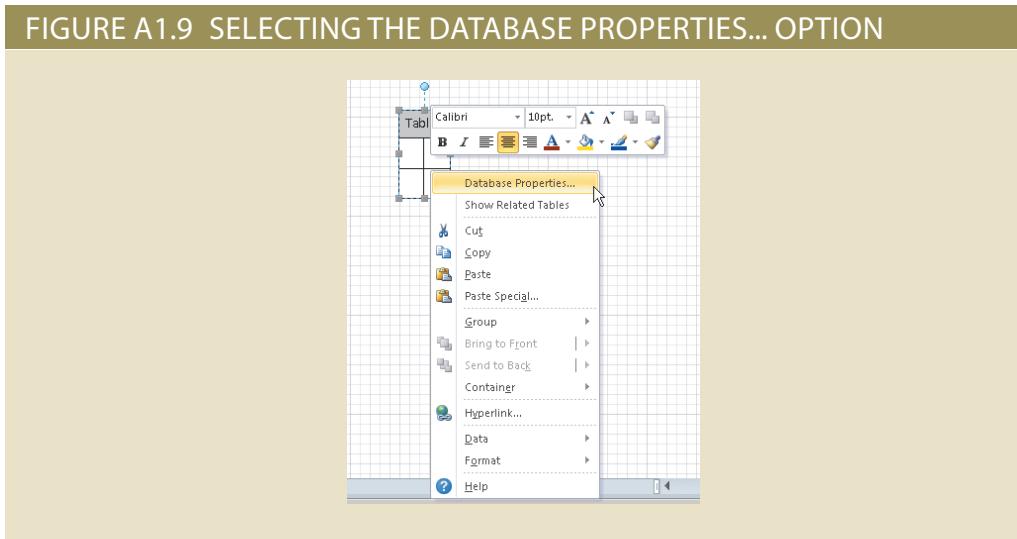
If the **Table1** object is selected, you will see the default **Database Properties** window at the bottom of the screen. (You will see later in this section that the window’s location and format may be changed to become the new default. However, you will start by using the standard default window shown in Figure A1.8.)

As you examine the **Database Properties** window in Figure A1.8, note the selection of the **Definition** option in the **Categories:** listing. (To select any option in the list, click it. The selection is indicated by the arrow to the left of the option. In this case, the arrow appears next to the **Definition** option.) At this point, the default **Table1** label shows up in the **Physical name:** slot.

## A1-3b Creating the Default Database Properties Window

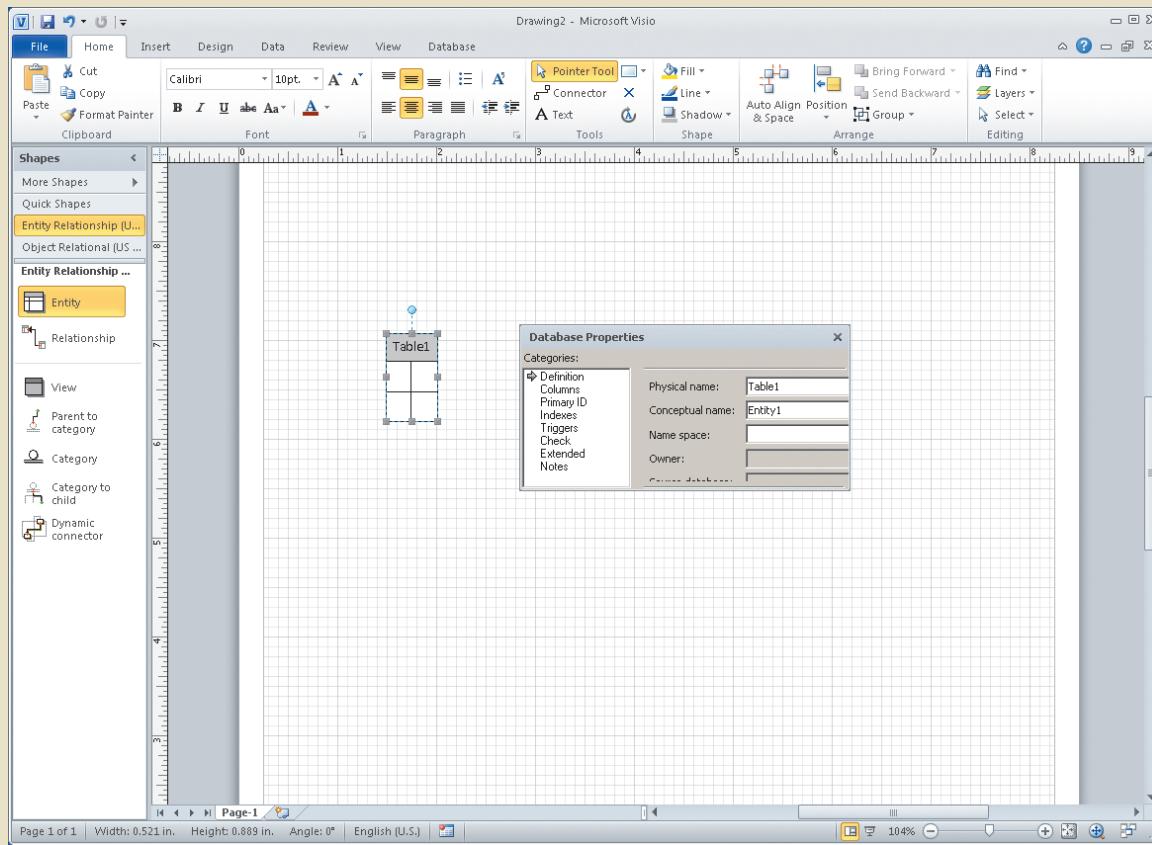
Depending on how you configured the Visio Professional software and/or on what operating system you use, you may not see the Database Properties window. If your screen does not show a default Database Properties window, right-click the **Table1** object in the grid to generate the **Database Properties...** option shown in Figure A1.9.

FIGURE A1.9 SELECTING THE DATABASE PROPERTIES... OPTION



Click the **Database Properties...** option shown in Figure A1.9 to display the **Database Properties** window. Figure A1.10 shows you a typical placement of the window. In that example, the **Database Properties** window is located on the grid, next to the **Table1** object. You will learn how to change the window’s location and format.

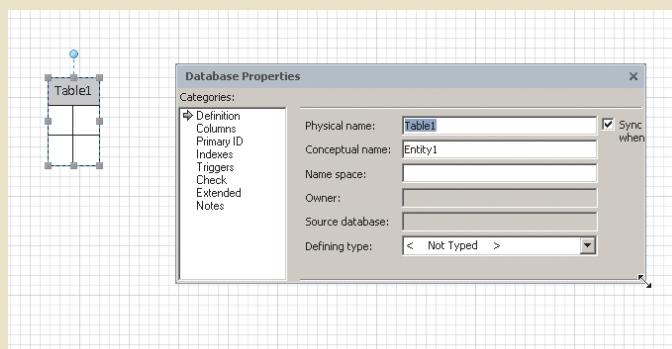
FIGURE A1.10 THE DATABASE PROPERTIES WINDOW



### A1-3c Sizing the Database Properties Window

You can size the **Database Properties** window as you would size any Windows object. For example, note that placing the cursor in the lower right margin (see Figure A1.11) changes the cursor shape to a double-sided arrow in preparation for resizing the window by dragging its lower and right limits.

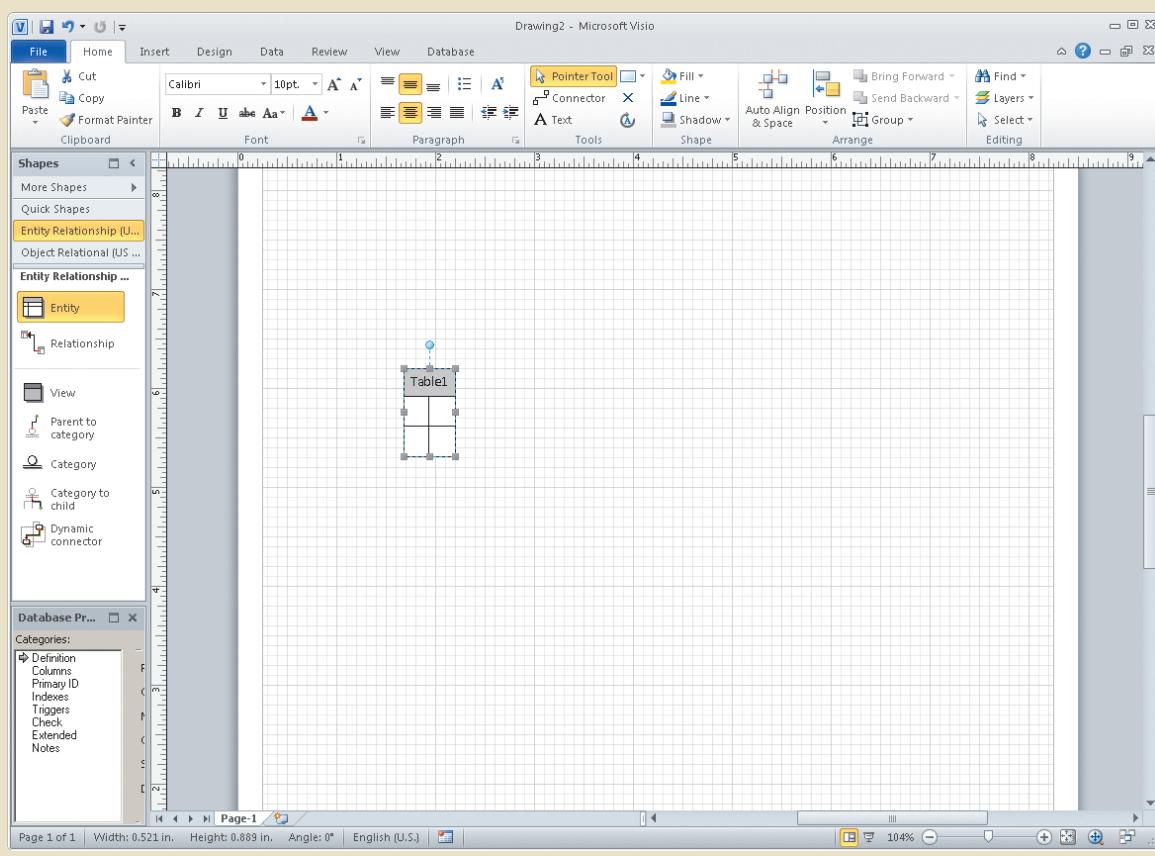
FIGURE A1.11 SIZING THE DATABASE PROPERTIES WINDOW



### A1-3d Moving the Database Properties Window

You can also drag and drop the entire **Database Properties** window to the screen's lower-left corner. (See Figure A1.12.)

FIGURE A1.12 THE DATABASE PROPERTIES WINDOW IN THE LOWER-LEFT CORNER

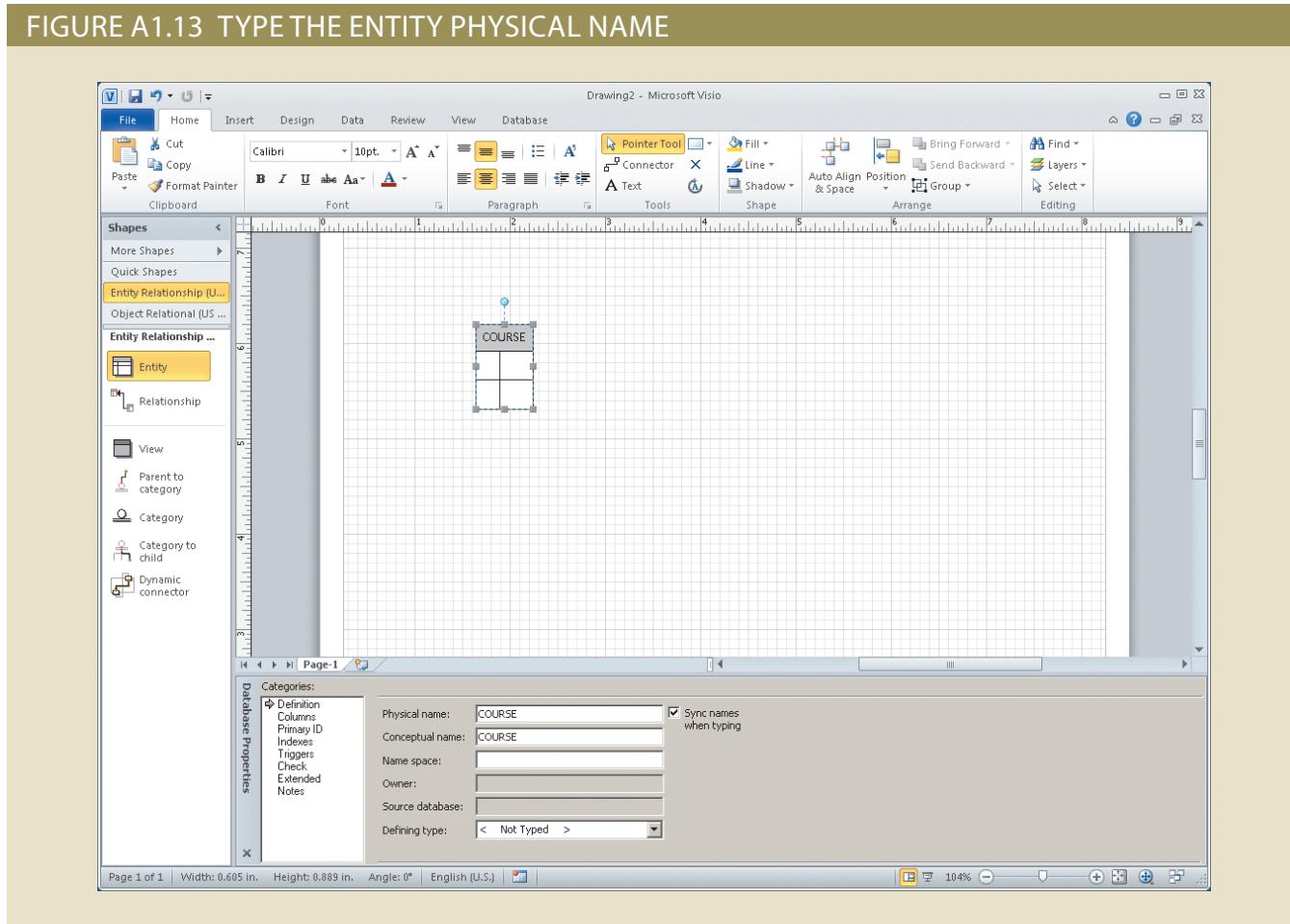


Naturally, you can also drag and drop the **Database Properties** window back to its original position depicted in Figure A1.8. (Just drag and drop to the screen's bottom margin.) Because that location allows you to see more of the database properties without blocking part of the entities you draw on the screen, that's the position you'll use.

### A1-3e Creating the Entity Name

First, create a COURSE entity by placing the cursor in the **Physical name:** slot and typing COURSE, as shown in Figure A1.13. Because the **Sync names when typing** (default) option was selected in Figure A1.13, the **Physical name:** and **Conceptual name:** entries are the same.

**FIGURE A1.13 TYPE THE ENTITY PHYSICAL NAME**



When you have finished typing the COURSE label in the **Physical name:** slot as shown in Figure A1.13, note that the conceptual table in the grid automatically inherits the COURSE label. You are now ready to start defining the table columns.

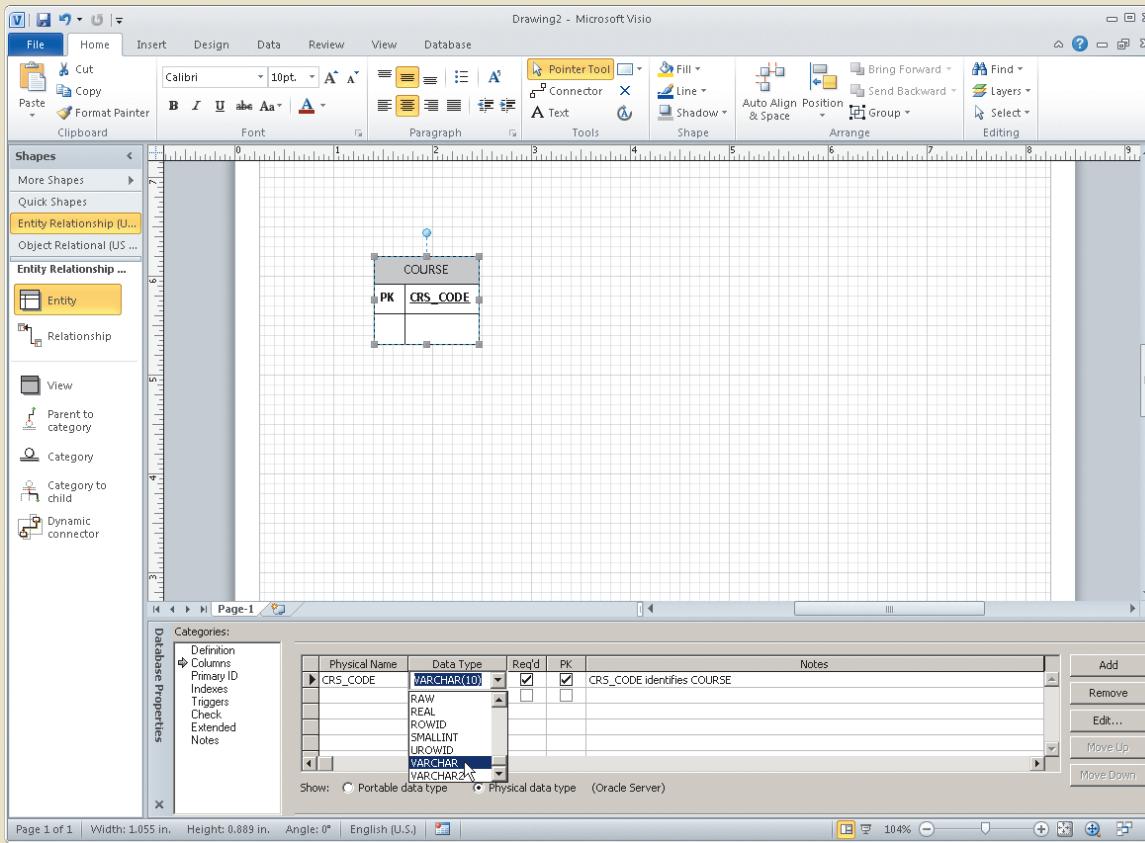
## A1-3f Defining the Entity Attributes (Columns)

Each table column represents one of the characteristics (attributes or fields) of the entity. For example, if the COURSE entity, represented by the COURSE table, is described by the course code, the course description, and the course credits, you can expect to define three columns in the COURSE table. Table A1.1 provides a preview of the expected COURSE table structure. (A few sample records are entered to give you an idea of the COURSE table contents.)

<b>TABLE A1.1</b> <b>SOME SAMPLE COURSE RECORDS</b>			
<b>CRS_CODE</b>	<b>CRS_TITLE</b>	<b>CRS_DESCRIPTION</b>	<b>CRS_CREDITS</b>
ACCT-345	Managerial Accounting	Accounting as a management tool. Prerequisites: Junior standing and ACCT-234 and 245.	3
CIS-456	Database Systems Design	Creation of conceptual models, logical models, and design implementation. Includes basic database applications development and the role of the database administrator. Prerequisites: Senior standing and at least 12 credit hours in computer information systems, including CIS-234 and CIS-345.	4
ECON-101	Introduction to Economics	An introduction to economic history and basic economic principles. Not available for credit to economics and finance majors.	3

To define the columns of the COURSE table, you must assign column names and characteristics. The first column in the COURSE table will be the CRS\_CODE, which serves as the table's primary key (PK). Because typical course code entries might be values such as CIS-456 or ACCT-234, each data entry involves a *character string*. Note that not all entries will have the same number of characters, so the course code is a variable length character string. In structured query language (SQL) terms, the CRS\_CODE data are best defined as VARCHAR() data. Figure A1.14 shows you how the CRS\_CODE name and data characteristics were specified.

FIGURE A1.14 THE COLUMN PK SELECTION

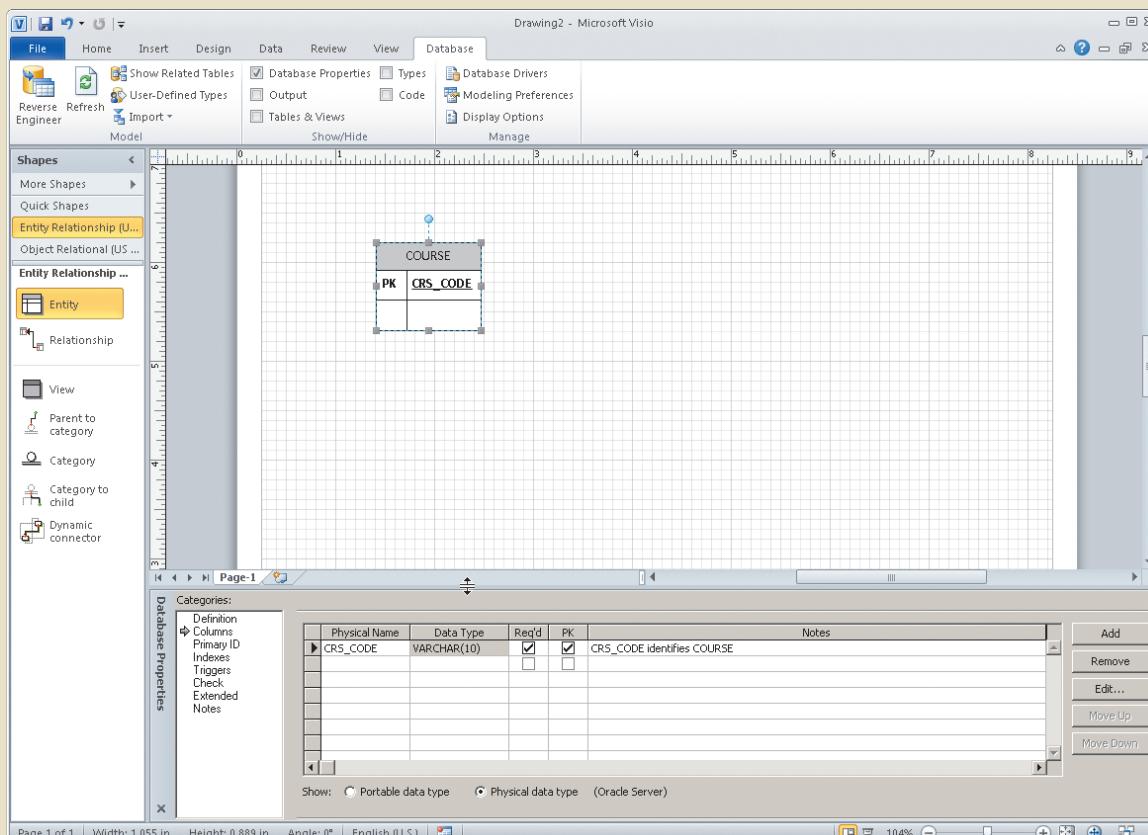


To generate the appropriate input for the column characteristics shown in Figure A1.14, follow these steps:

1. Make sure that the COURSE table object—shown in the grid—is selected. (The handles around the perimeter show that the selection was made properly.)
2. Select the **Columns** option in the **Database Properties** window at the bottom of the screen. (Note that the selection was marked with an arrow.)
3. Step 2 generates the column-specific dialog box. Type **CRS\_CODE** in the first line under the **Physical Name** header. Moving along the line for the **CRS\_CODE** entry:
  - a. Select the **VARCHAR** option from the drop-down list under the **Data Type** header. (Click the **down arrow** to generate the list.)
  - b. Because a course code is required to define the course offering, place a check mark—by clicking the check box—under the **Req'd** header.
  - c. Because the **CRS\_CODE** is the PK, place a check mark—by clicking the check box—under the **PK** header.

Before you enter the remaining attribute names and characteristics, you may need to enlarge the **Database Properties** window by dragging its upper limit (see Figure A1.15) to increase the desired space. That action lets you see sufficient space for all of the remaining attributes in the COURSE table. Now place the cursor on the second **Columns** line and get ready to enter the remaining attributes.

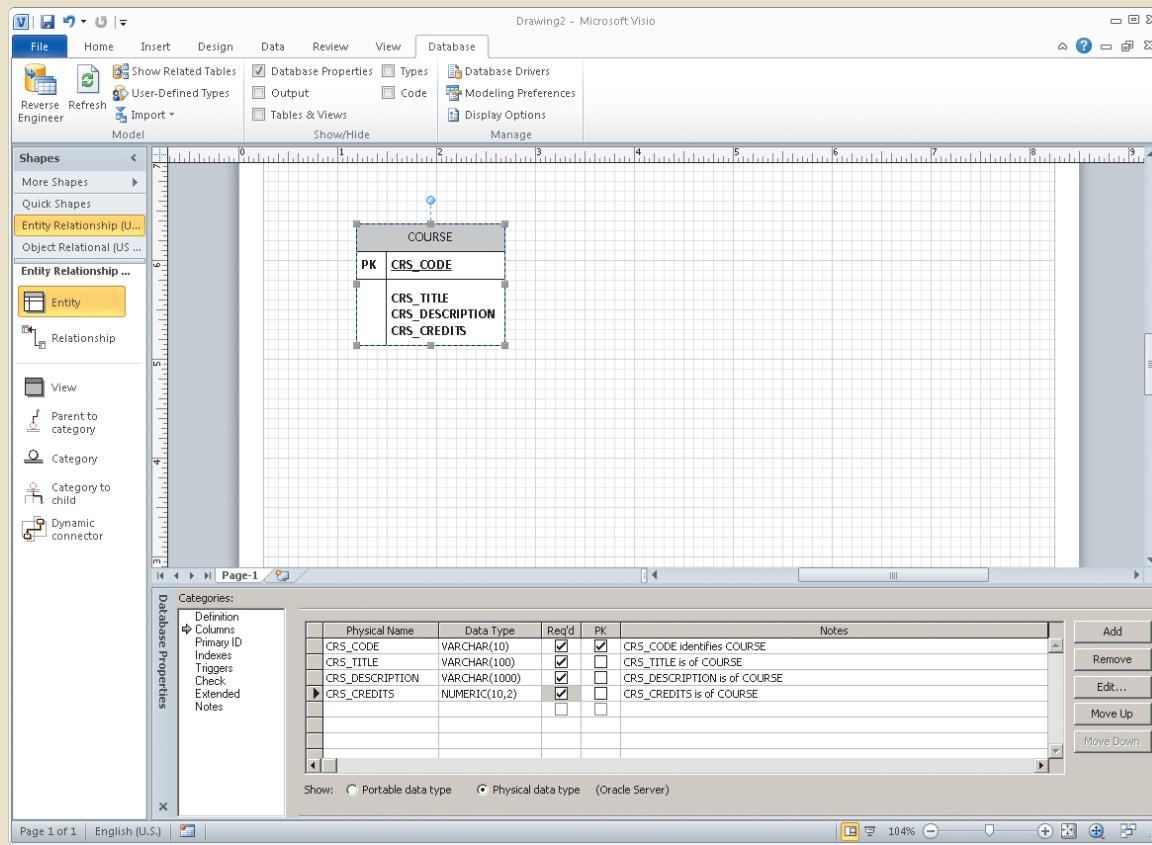
FIGURE A1.15 DRAG THE DATABASE PROPERTIES BOX LIMIT TO SHOW MORE COLUMNS



You are now ready to make the entries for the second COURSE attribute. Name this attribute **CRS\_TITLE**. Typical entries are *Database Design and Implementation* or *Intermediate Accounting*. (Check the sample entries in Table A1.1.) Therefore, the CRS\_TITLE is a character field. Double-click the selected data type to allow editing of the length of the character string allowed. Set the CRS\_TITLE to a maximum length of 100 characters. The course title is required, but it is not a PK. Similarly, enter the CRS\_DESCRIPTION entries with a maximum length of 1000 characters, and make it required. The CRS\_CREDITS entries are numeric, and they are required; they will be used at some point to help compute grade point averages for the students taking a section of this course. When the appropriate entries are made, the screen will look like Figure A1.16. Note that the attribute names become boldfaced when the **Req'd** (required) option is checked for the **Column** property. Selecting that option means that the design should be implemented such that the end user will be required to provide a value for the checked attributes during data entry activities.

## A1-16 Appendix A1

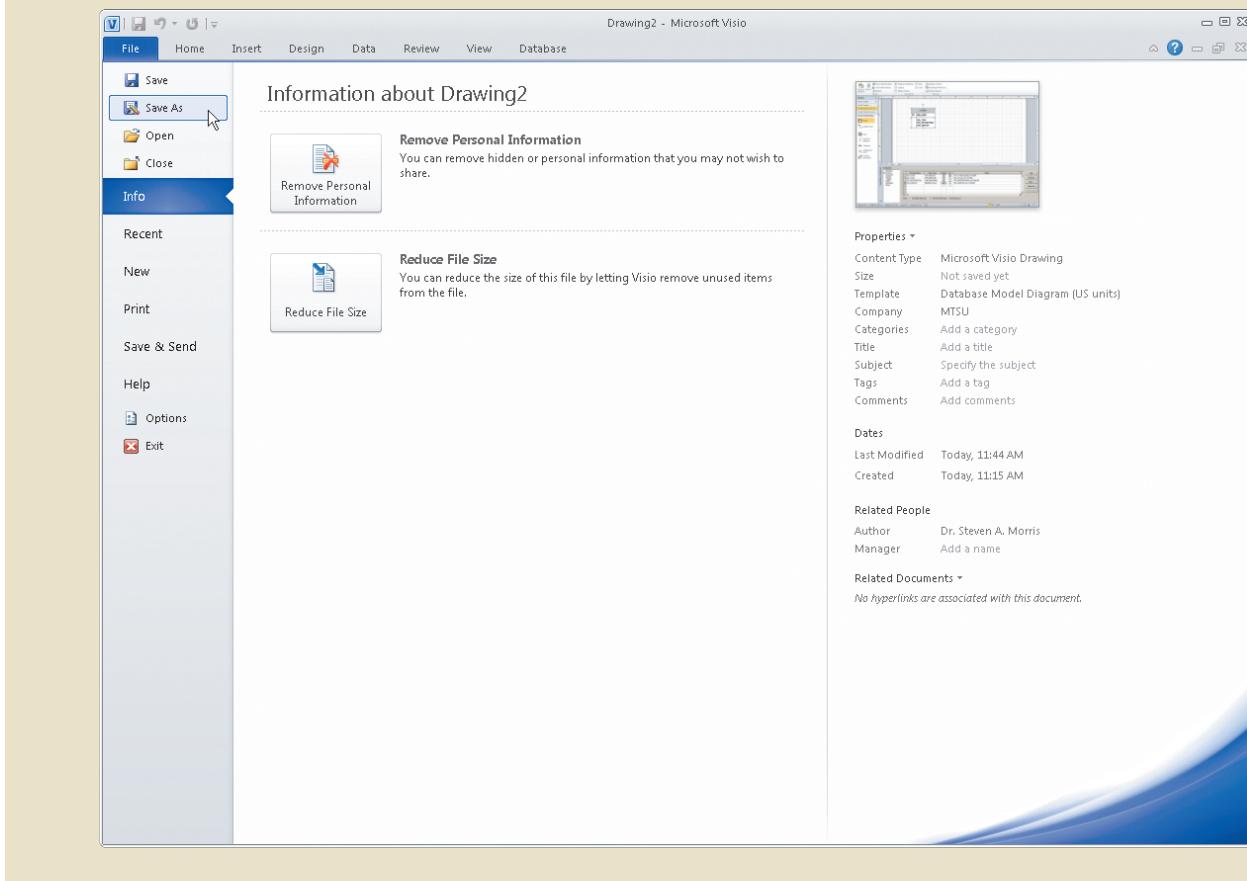
FIGURE A1.16 ENTER THE REMAINING COLUMNS



## A1-4 Saving and Opening the Visio ERD

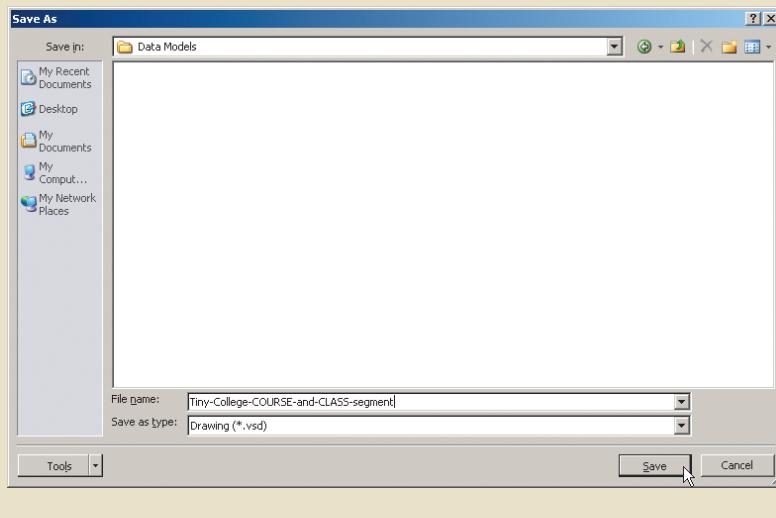
To ensure that you don't lose this first Visio Professional ERD segment, save it in an appropriate folder. Use the **File, Save As** option to select the folder location and the file-name, as shown in Figures A1.17 and A1.18.

**FIGURE A1.17 SELECT THE SAVE AS OPTION TO SAVE THE FILE**



**A1-18** Appendix A1

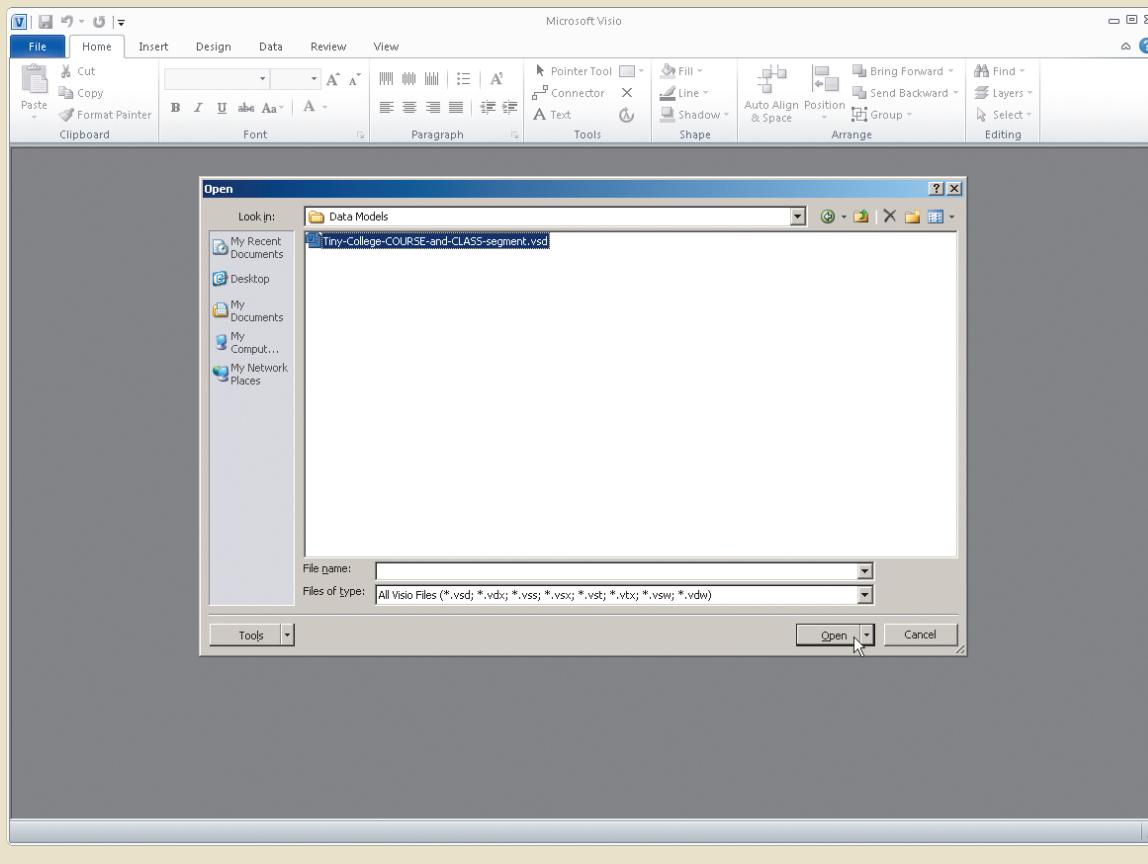
FIGURE A1.18 SELECT THE FOLDER, TYPE THE FILE NAME, AND SPECIFY THE FILE TYPE



As you examine Figure A1.18, note that the filename describes its origin and purpose. In this example, the ERD is named **Tiny-College-COURSE-and-CLASS-segment**. The naming convention serves the important purpose of self-documentation. Note also that the file is saved as a Visio Drawing (\*.vsd).

You can now go ahead and close the file—and, of course, make a backup copy! The next time you want to use the file, after you start Visio Professional, use the standard Windows **File, Open** option to retrieve and open the file, the process used to generate the screen shown in Figure A1.19.

**FIGURE A1.19 OPEN THE PREVIOUSLY SAVED FILE**

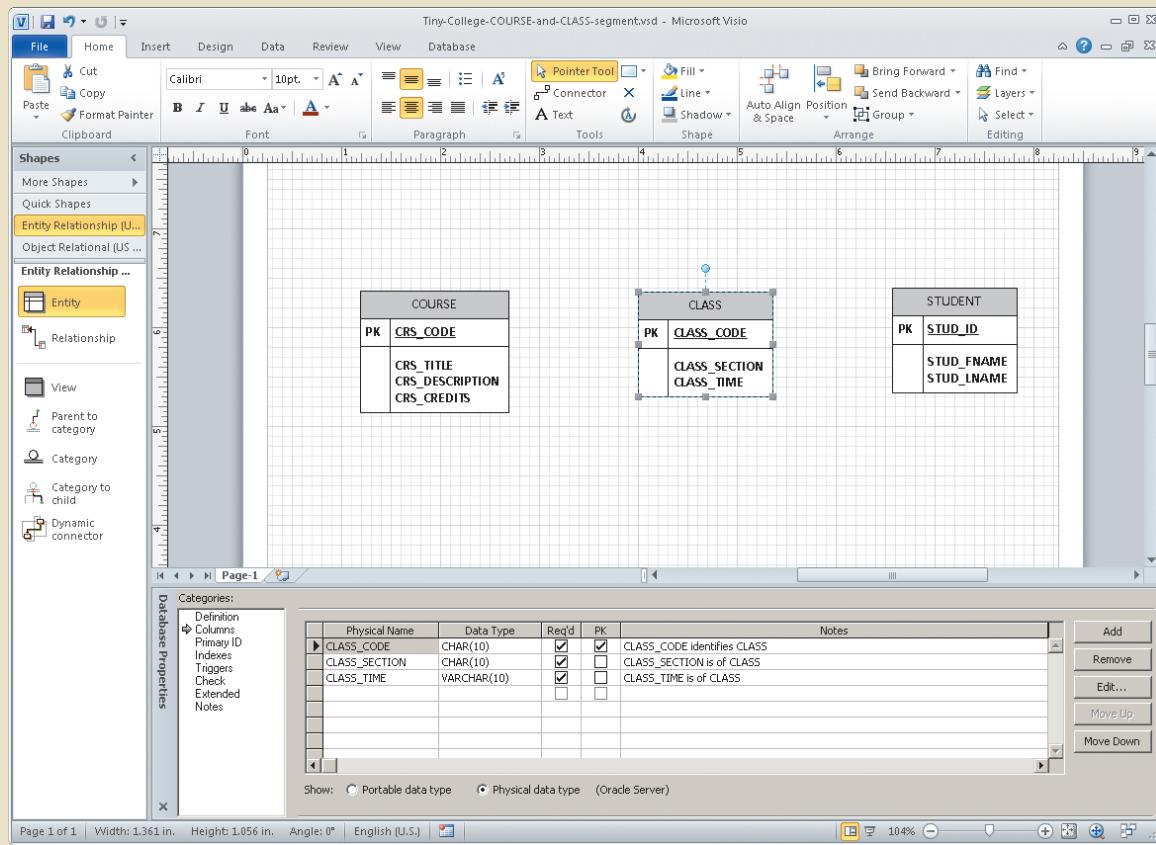


Note that the just-opened file does not show any entity properties. If you want to see this entity's properties, click the COURSE table to display its characteristics in the **Database Properties** window again.

You are now ready to define the CLASS and STUDENT entities, using the same techniques you used to create the COURSE entity. When you are done, the screen will look like Figure A1.20.

## A1-20 Appendix A1

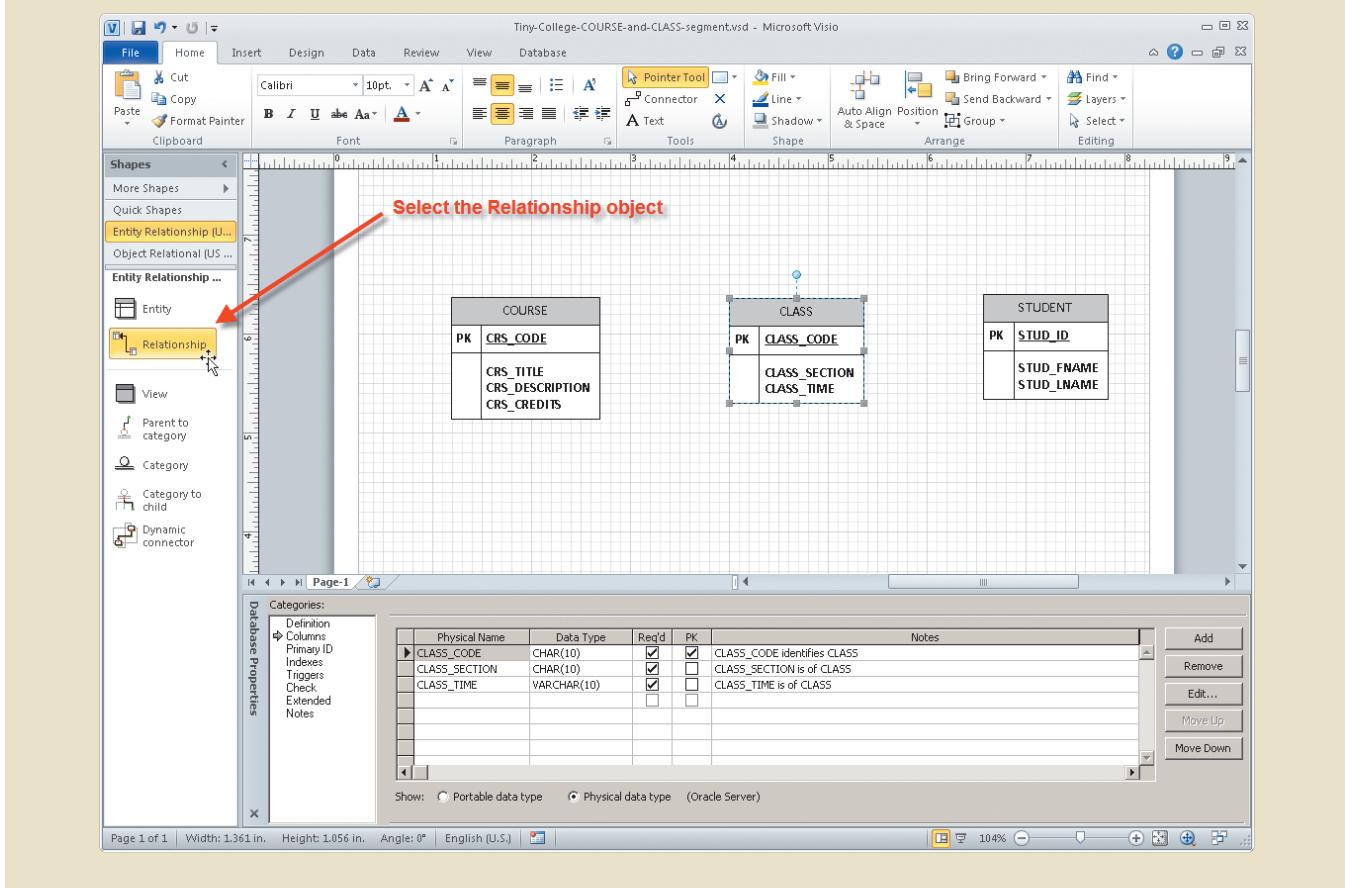
**FIGURE A1.20 ADDING THE CLASS AND STUDENT ENTITIES**



## A1-5 Defining Relationships

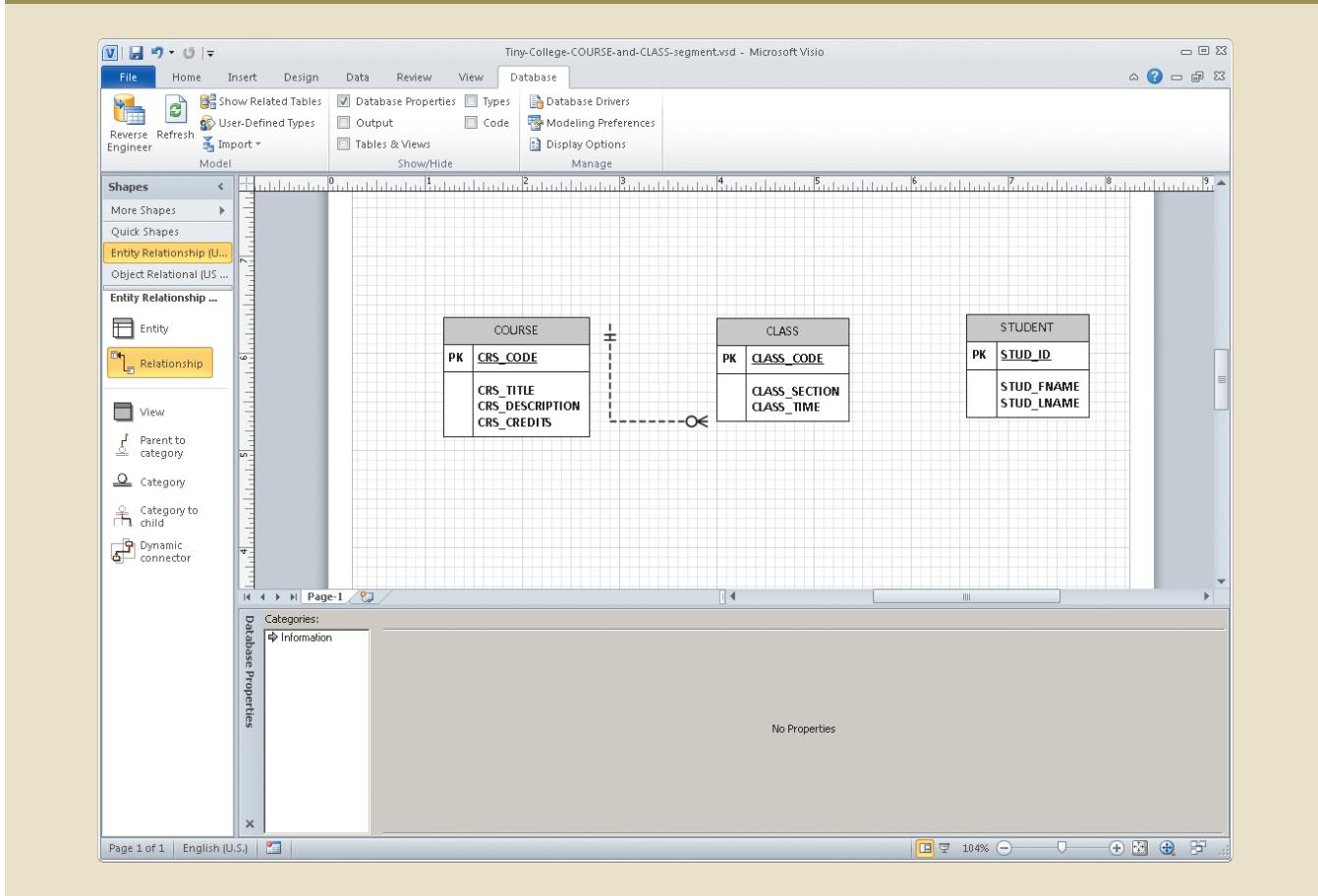
As you examine Figure A1.20, note that a foreign key (FK) has not been defined in CLASS to relate CLASS to COURSE. Instead, *Visio Professional will define the FK field when you specify the relationship between the two entities*. Do **not** enter your own FK fields! (Visio Professional tells you what the relationship option will do for you—read the relationship text in Figure A1.21).

FIGURE A1.21 SELECT THE RELATIONSHIP OBJECT



To create a relationship between the entities, click the **Relationship** object, drag it to the grid, and drop it between the COURSE and CLASS entities to produce the results shown in Figure A1.22.

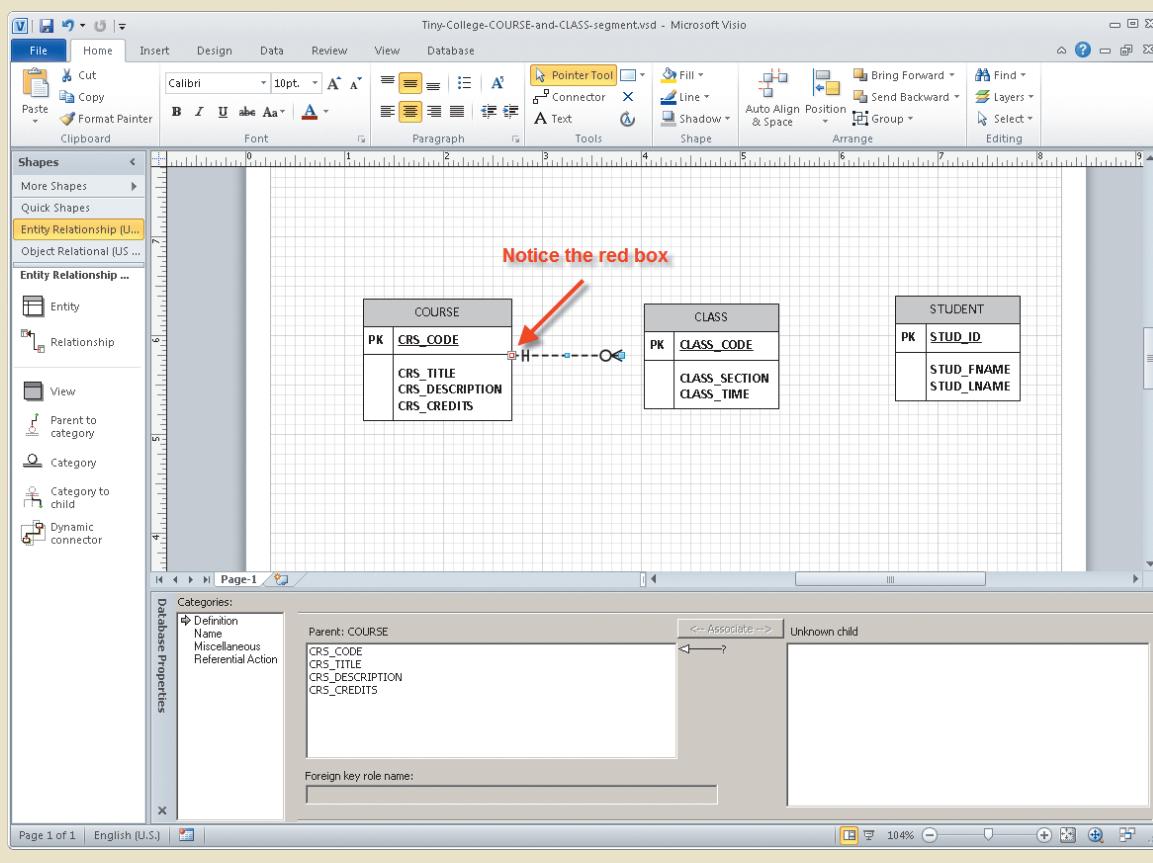
**FIGURE A1.22 DRAG AND DROP THE RELATIONSHIP OBJECT**



Dropping the **Relationship** object on the grid produces the relationship line. Further note that the symbols at the two ends of the relationship line reflect default cardinalities of (1,1) and (0,N). Finally, remember that the relationship to be established between COURSE and CLASS reflects the business rule “One COURSE may generate many CLASSES.” Therefore, the COURSE represents the “1” side of the relationship and the CLASS represents the “many” side of the relationship.

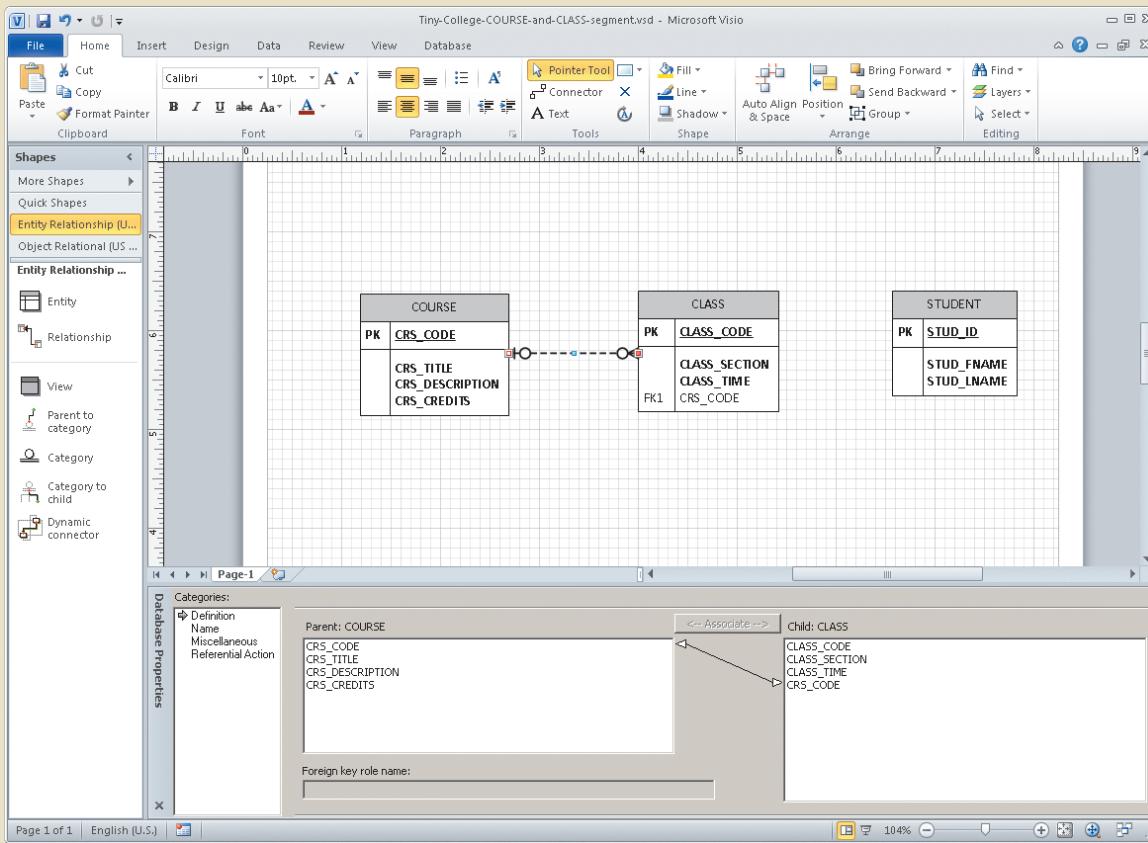
Attach the “1” side of the relationship line to the COURSE entity by dragging the “1” end of the relationship line to the COURSE entity, as shown in Figure A1.23. Note—and this is very important—the relationship is not attached until the COURSE table is outlined in red. (You may have to drag the relationship line’s end all the way to the inside of the table before the red outline shows up.) When you release the relationship line, its attachment is verified by the red square on the entity (table) perimeter.

FIGURE A1.23 ATTACH THE "1" SIDE OF THE RELATIONSHIP LINE



Using the same technique that was used to attach the “1” side of the relationship, drag the “M” side of the relationship line to the CLASS entity to produce Figure A1.24. (Make sure that you see the red square on the CLASS entity side of the relationship line when you are done.)

FIGURE A1.24 ATTACH THE "M" SIDE OF THE RELATIONSHIP LINE



As you examine Figure A1.24, note these features:

1. The two red rectangles at the margin of each table indicate that the relationship was successfully established and that it is still selected. (If the relationship line is no longer selected, the red squares disappear. *To reselect the relationship line, click it.*)
2. Visio Professional created the CRS\_CODE foreign key in the CLASS table, labeling it **FK1** to indicate that it is the first FK created for this table. Note that CRS\_CODE in the CLASS table is not in boldfaced type. This lack of boldface indicates that, at this point, you have not yet specified that a FK value is mandatory. (Of course, it should be because a CLASS cannot exist without a COURSE. You will edit this FK property later.)
3. The cardinality next to COURSE was automatically changed to indicate an optional (0,1) relationship between CLASS and COURSE. Because each class must be related to one course, a depiction of a (1,1) cardinality is appropriate. (A CLASS cannot exist without a COURSE.) Therefore, you'll have to edit this cardinality later.
4. The **Database Properties** window shows that the (default) **Definition** option is selected. (Look under the **Categories:** header.)
5. The relationship is reflected in the double-sided arrow linking the COURSE table's CRS\_CODE and the CLASS table's CRS\_CODE.

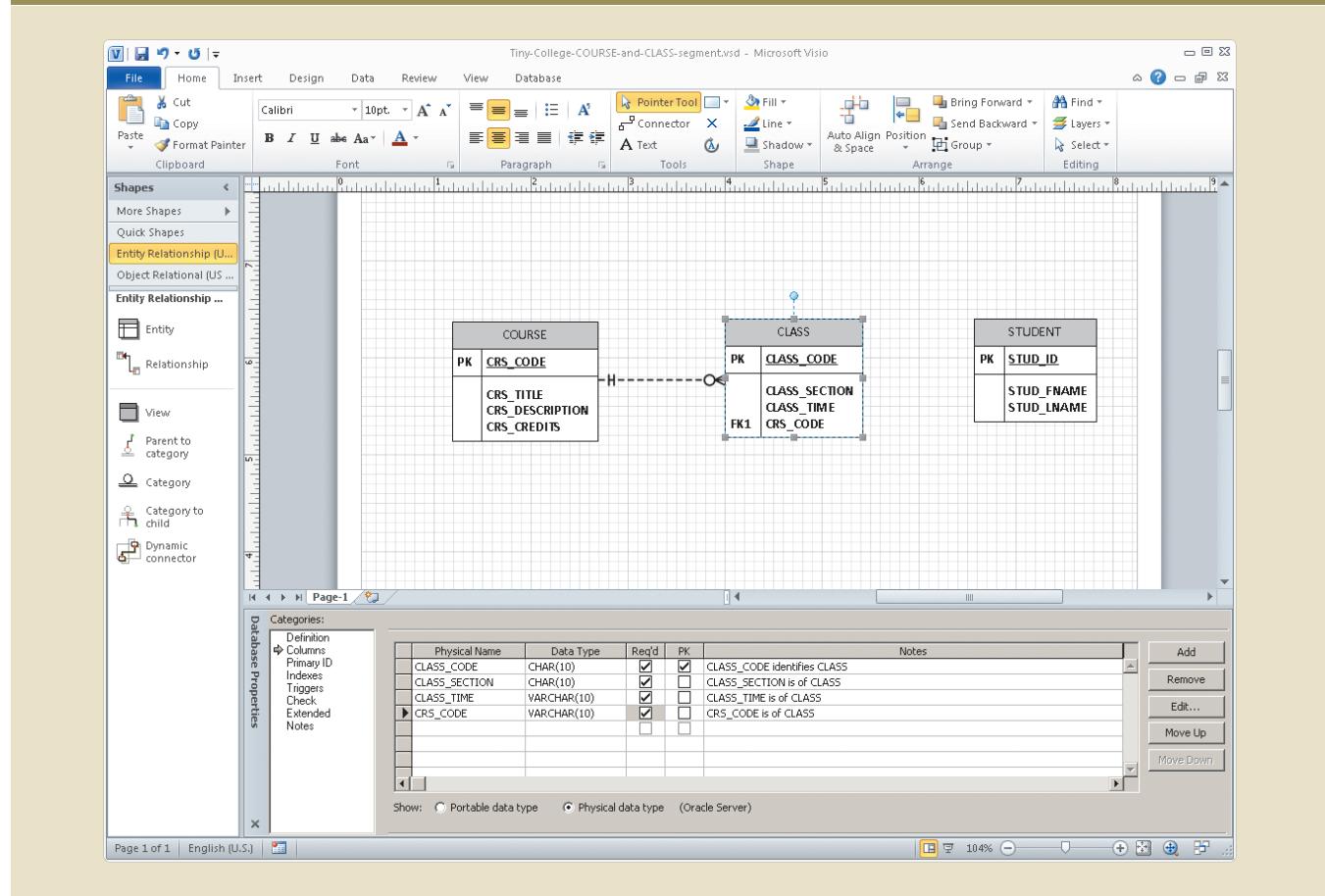
## A1-5a Editing the Cardinalities

If you examine Figure A1.24, you'll notice that the CRS\_CODE in the CLASS entity is not in boldfaced type. This lack of boldface indicates that the CRS\_CODE in CLASS may be null, thus indicating incorrectly that COURSE is optional to CLASS. To change the (0,1) cardinality to a (1,1) cardinality:

1. Select the **CLASS** entity.
2. Check the CRS\_CODE and note that its **Req'd** check box is *not* checked. (That means that a value entry is not required, thus allowing nulls—and making the relationship between CLASS and COURSE optional.)
3. Click the CLASS entity's CRS\_CODE **Req'd** check box to place a check mark in it. (That means that a value entry will be required, thus making the relationship between CLASS and COURSE mandatory.)

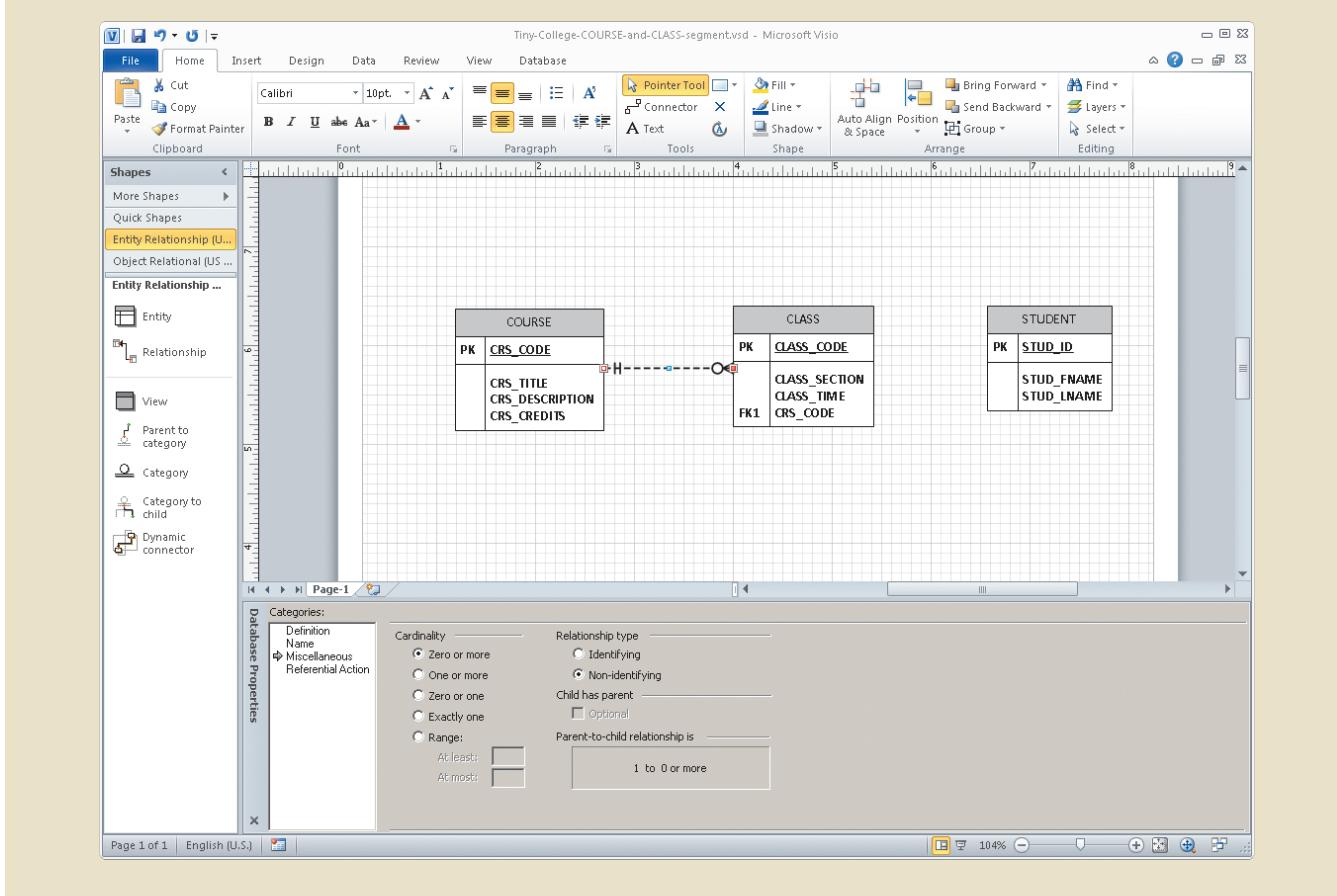
When you have completed those three steps, you will see the results in Figure A1.25. Note that the CRS\_CODE in the CLASS entity is now in boldface to indicate the mandatory relationship between CLASS and COURSE. That mandatory relationship is reflected by the change in the (0,1) cardinality to a (1,1) cardinality on the COURSE entity.

**FIGURE A1.25 FORCING A MANDATORY ENTRY FOR A FOREIGN KEY VALUE**



You can edit the “M” side of the 1:M relationship by selecting the relationship line and the **Miscellaneous** option in the **Categories:** list. Then select the **Zero or more** cardinality (if it is not already selected). Figure A1.26 shows the screen after the selections have been properly made.

**FIGURE A1.26 SELECTING THE CARDINALITY FOR THE "MANY" SIDE OF A RELATIONSHIP**



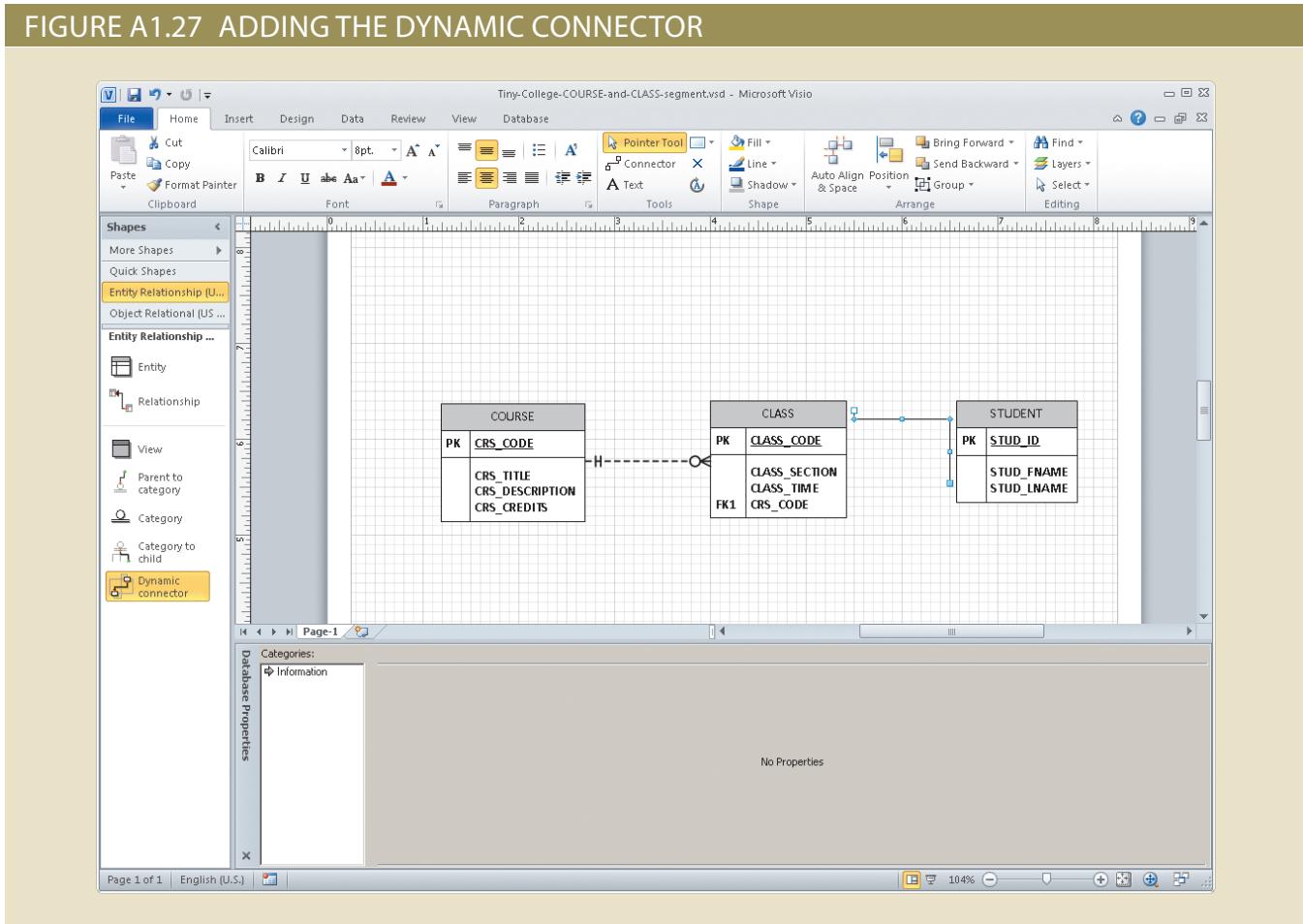
## A1-5b Creating a “many-to-many” Relationship

As discussed in Chapter 4, Entity Relationship Modeling, “many-to-many” relationships are appropriate in conceptual data models. However, Visio focuses on the creation of logical models for relational databases, hence the name “Database Model Diagram” instead of “Data Model Diagram” for the drawing template. Recall that M:M relationships cannot be directly implemented in a relational database without the use of an associative or composite entity, and part of the process of converting a conceptual model to a logical model is the decomposition of M:M relationships into 1:M relationships. The result is that as a logical modeling tool, Visio does not expect you to be drawing M:M relationships. While it is possible to create these relationships in Visio, it is slightly different from other relationships.

To create a “many-to-many” relationship between the entities, click the **Dynamic connector** object, drag it to the grid, and drop it between the **CLASS** and **STUDENT** entities as shown in Figure A1.27. A dynamic connector is not the same as a relationship object in Visio. It does not automatically place foreign keys, Visio does not associate cardinalities with it, and it cannot embed primary key/foreign key relationship. A dynamic connector is simply a line that we can manually manipulate. Note that there are

no symbols at the ends of the relationship line to indicate cardinality. Remember that the relationship to be established between **CLASS** and **STUDENT** reflects the business rule “Many STUDENTs may enroll in many CLASSES.” Therefore, we need to change the representation of the relationship to represent the “many-to-many” side of it:

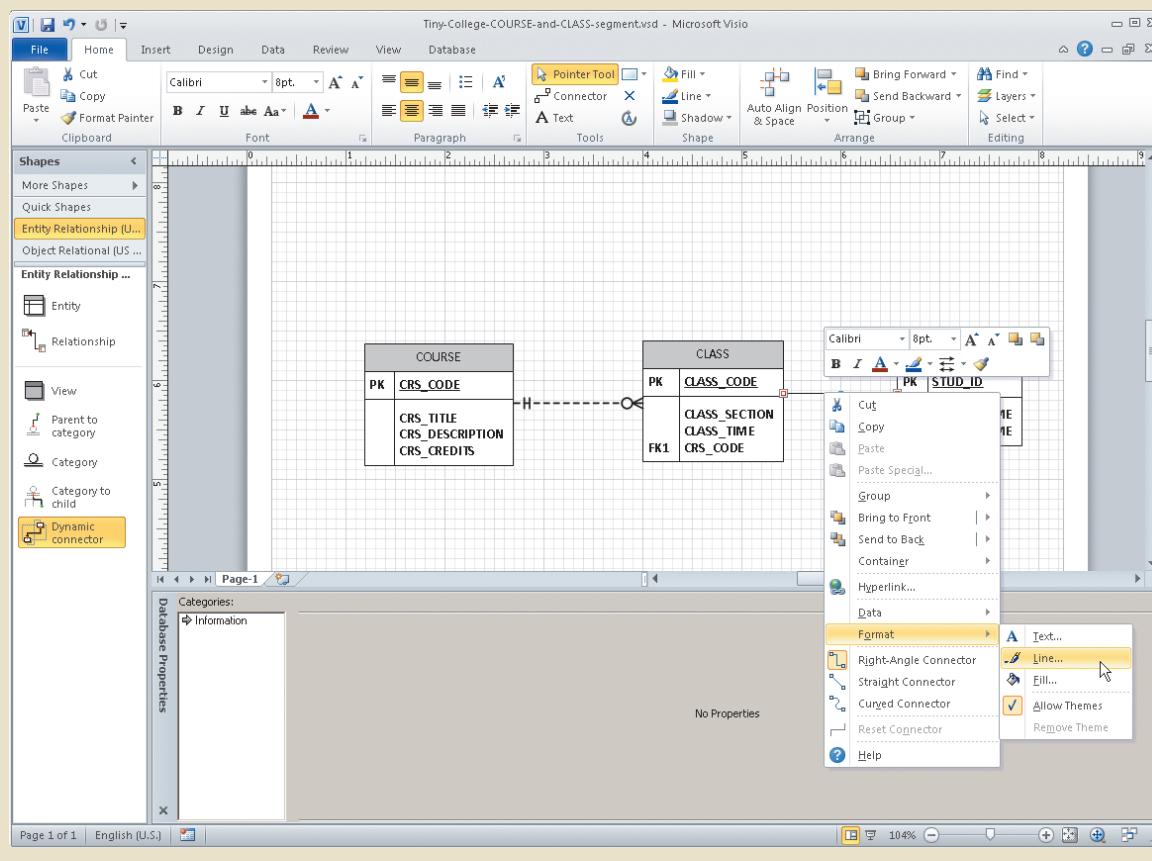
FIGURE A1.27 ADDING THE DYNAMIC CONNECTOR



Attach one side of the dynamic connector line to the **CLASS** entity and the other side of the line to the **STUDENT** entity. Right-click the line and select **Format → Line** as shown in Figure A1.28.

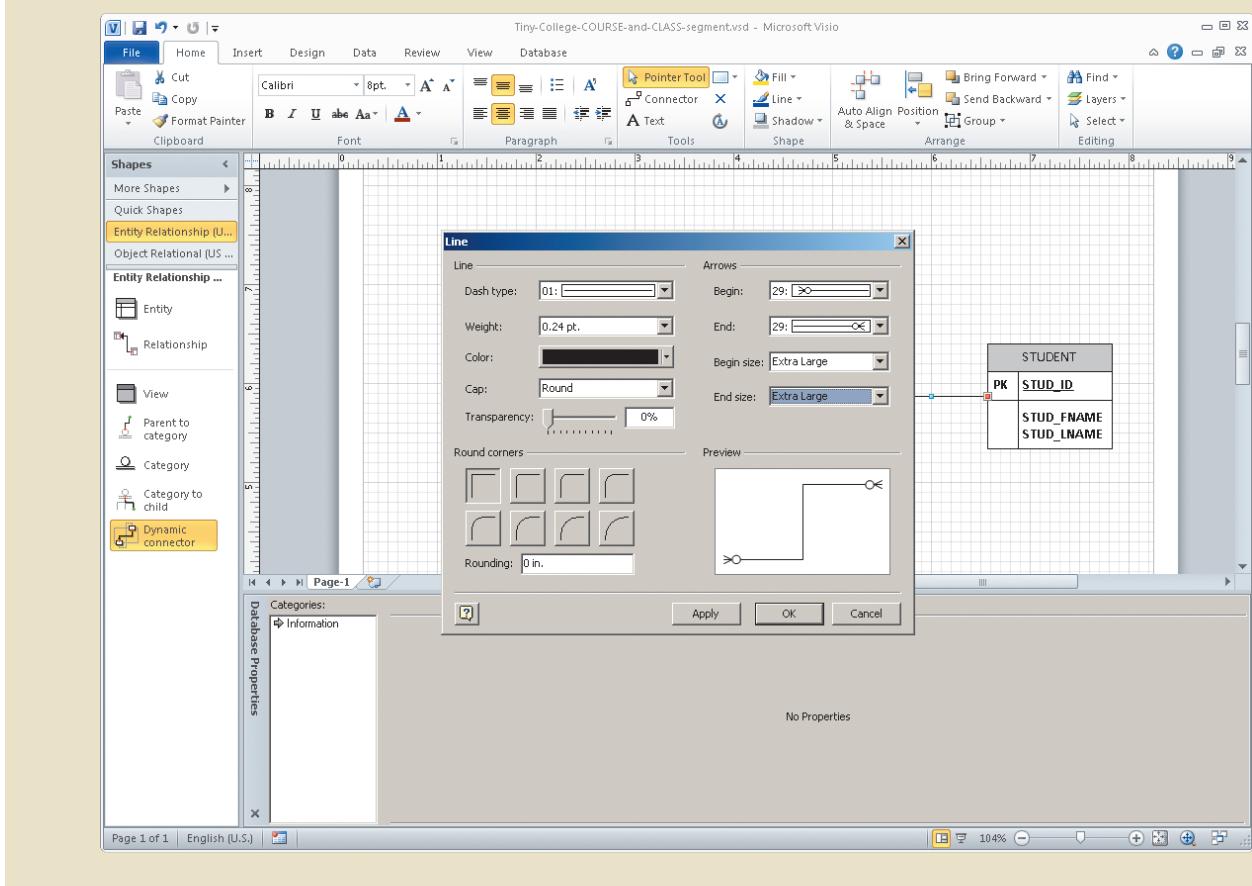
## A1-28 Appendix A1

FIGURE A1.28 FORMAT THE "MANY-TO-MANY" RELATIONSHIP



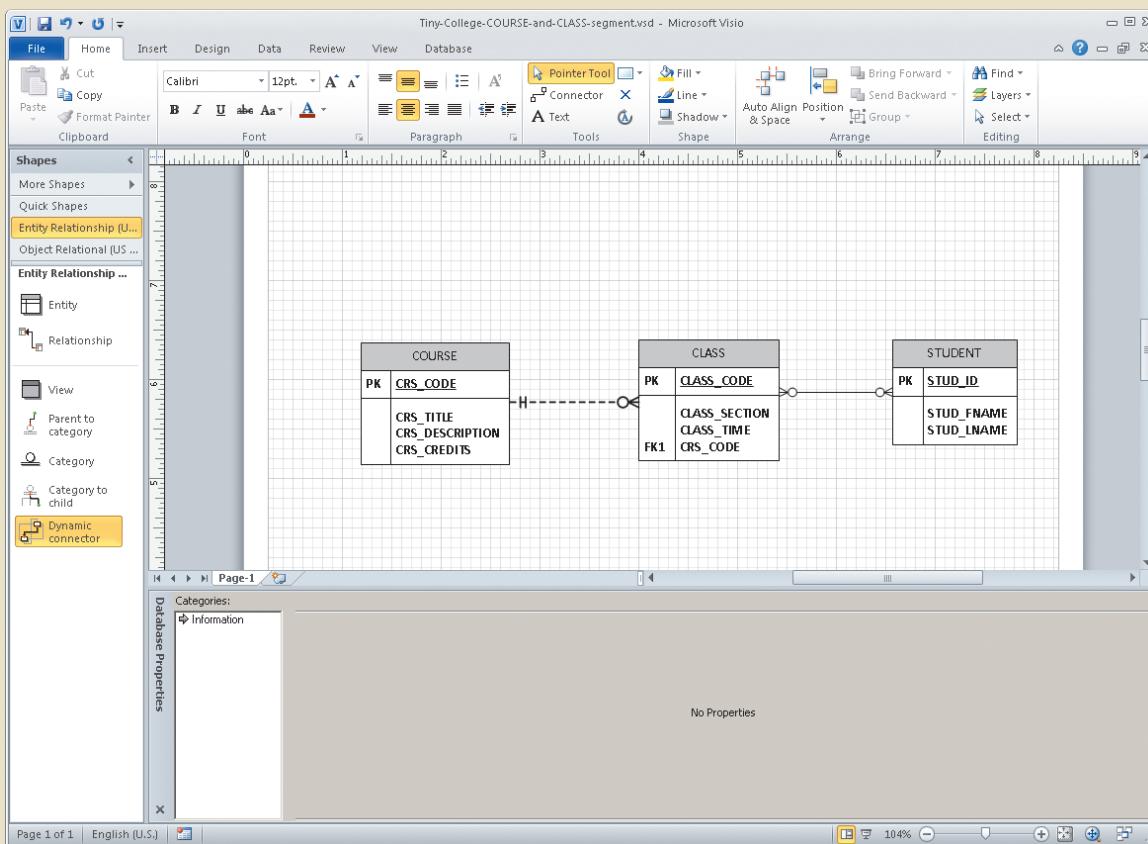
In the Line format window, we can manually select the appropriate Crow's foot notation symbols to represent the connectivity and participation that reflect our business rules. Select 29: (0, M) as the **Begin** and **End** symbols for the relationship line. Also, select to make the begin size and end size be **Extra Large**. Click **OK** to accept your selections.

FIGURE A1.29 SELECTING THE SYMBOLS OF A "MANY-TO-MANY" RELATIONSHIP



At this point, the diagram should accurately reflect the cardinalities of the relationships specified by the business rules. Figure A1.30 shows the many-to-many relationship between **CLASS** and **STUDENT** after the selections have been properly made. You will notice that the relationship line for the M:M relationship created using a dynamic connector is smaller than the 1:M relationship created using a relationship object. We will look at how to change the presentation of the relationship line, whether created by a relationship object or a dynamic connector, later in this tutorial.

FIGURE A1.30 "MANY-TO-MANY" RELATIONSHIP



### A1-5c Selecting the Relationship Type

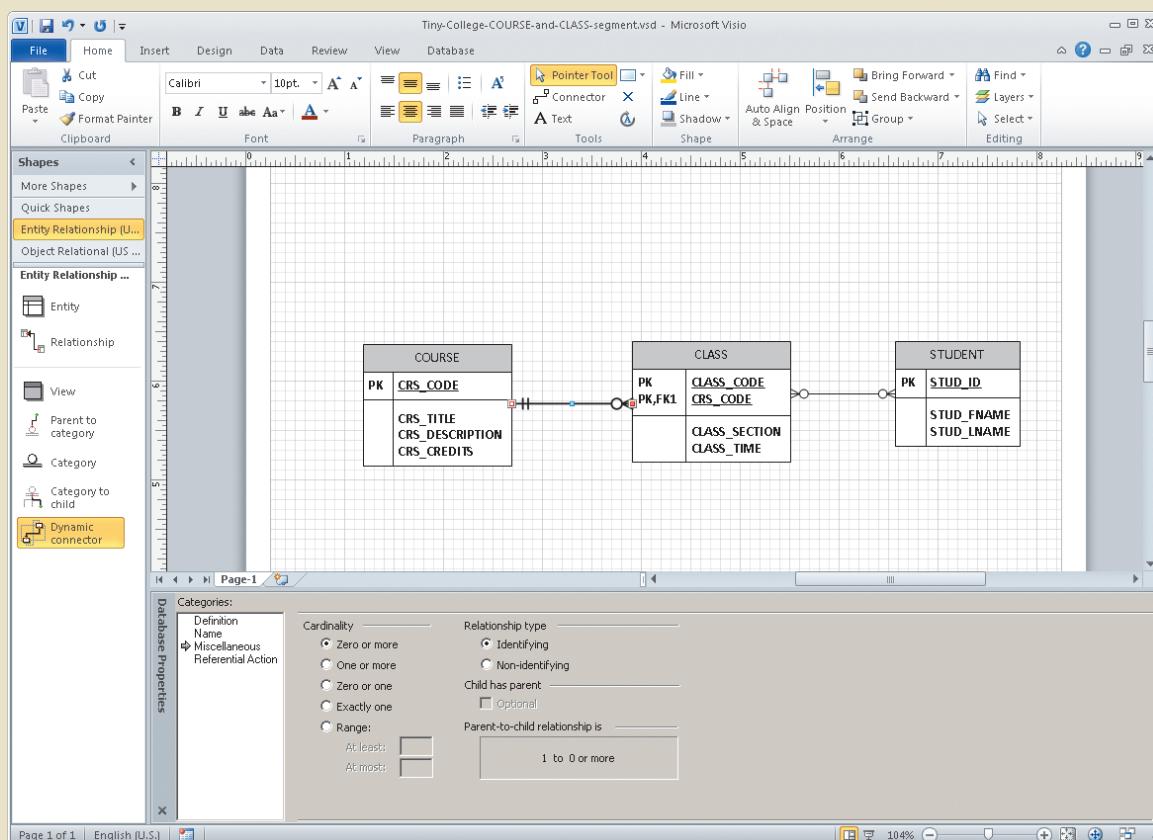
As you examine Figure A1.26, note the many options you have available. In this case, the relationship type is properly defined to be **Non-identifying** because the dependent CLASS entity did not inherit its PK from the parent COURSE entity. (When you created the CLASS entity, you defined its PK to be CLASS\_CODE, which is not found in the COURSE entity. In other words, the ERD in Figure A1.26 indicates that the CLASS entity is *not* a weak entity. A weak entity always has a strong relationship—that is, an identifying relationship—with its parent entity.)

The nature of the relationships between entities, the effect of optional and mandatory participation, and the existence of weak entities all have critical effects on the database design. If necessary, review Chapter 4 to review the nature and implementation of relationships.

Figure A1.26 shows the relationship between COURSE and CLASS as a dashed line. In Visio, a dashed relationship line between two entities always indicates a non-identifying (weak) relationship. You should recall that a weak (non-identifying) relationship always indicates the existence of a strong dependent entity. Conversely, a strong (identifying) relationship always indicates the existence of a weak dependent entity.

If you select an *identifying* relationship between COURSE and CLASS, Visio will automatically rewrite the PK of the CLASS entity for you and the relationship line will be solid. Figure A1.31 shows the effect of the relationship revision. After you have examined the effect of the identifying relationship selection, reset the relationship type to the one shown in Figure A1.26. (If you want to preserve the identifying relationship version of the ERD, save it with a different name, such as **Tiny-College-COURSE-and-CLASS-segment-Identifying-Relationship**.)

FIGURE A1.31 AN ILLUSTRATION OF AN IDENTIFYING (STRONG) RELATIONSHIP

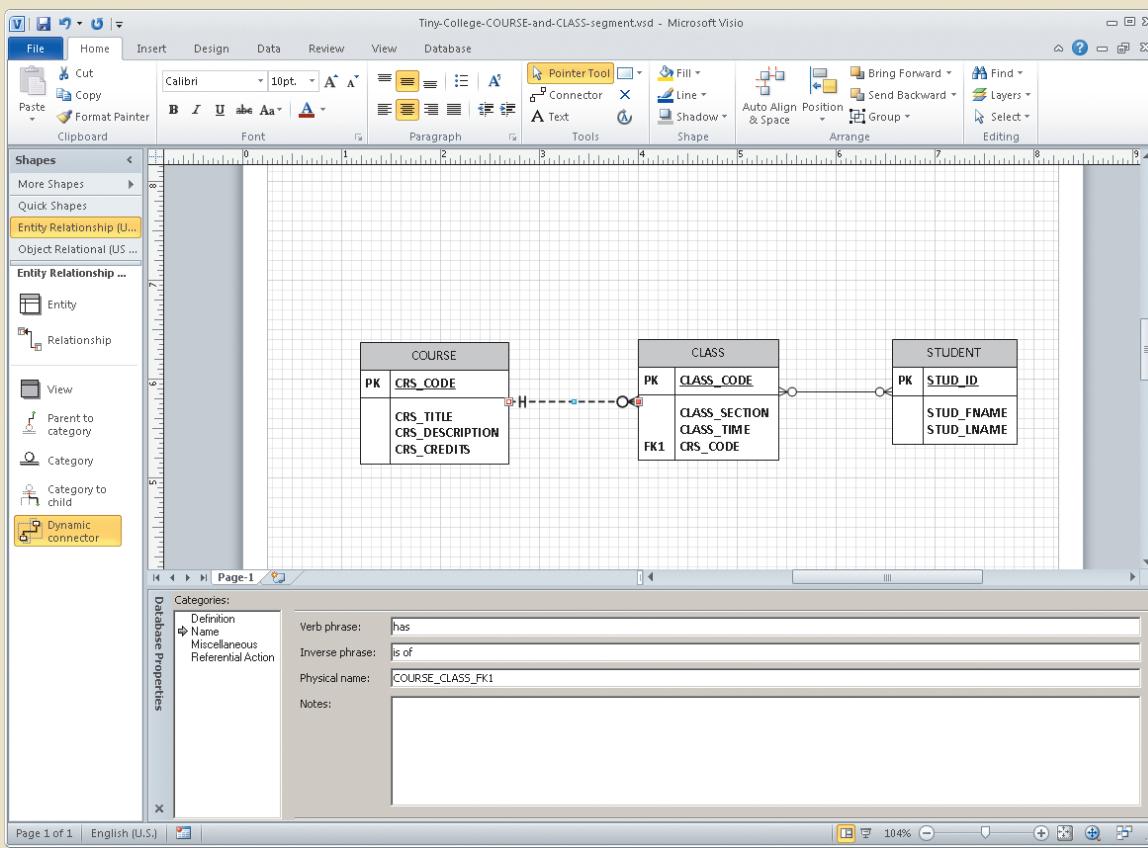


## A1-5d Naming the Relationships

Make sure that the relationship line is still selected. Then click the **Name** option in the **Database Properties** window at the bottom of the screen to produce the results displayed in Figure A1.32. (Note that the original ERD has been used to show the preferred non-identifying relationship between COURSE and CLASS.)

## A1-32 Appendix A1

**FIGURE A1.32 THE DEFAULT RELATIONSHIP NAME**



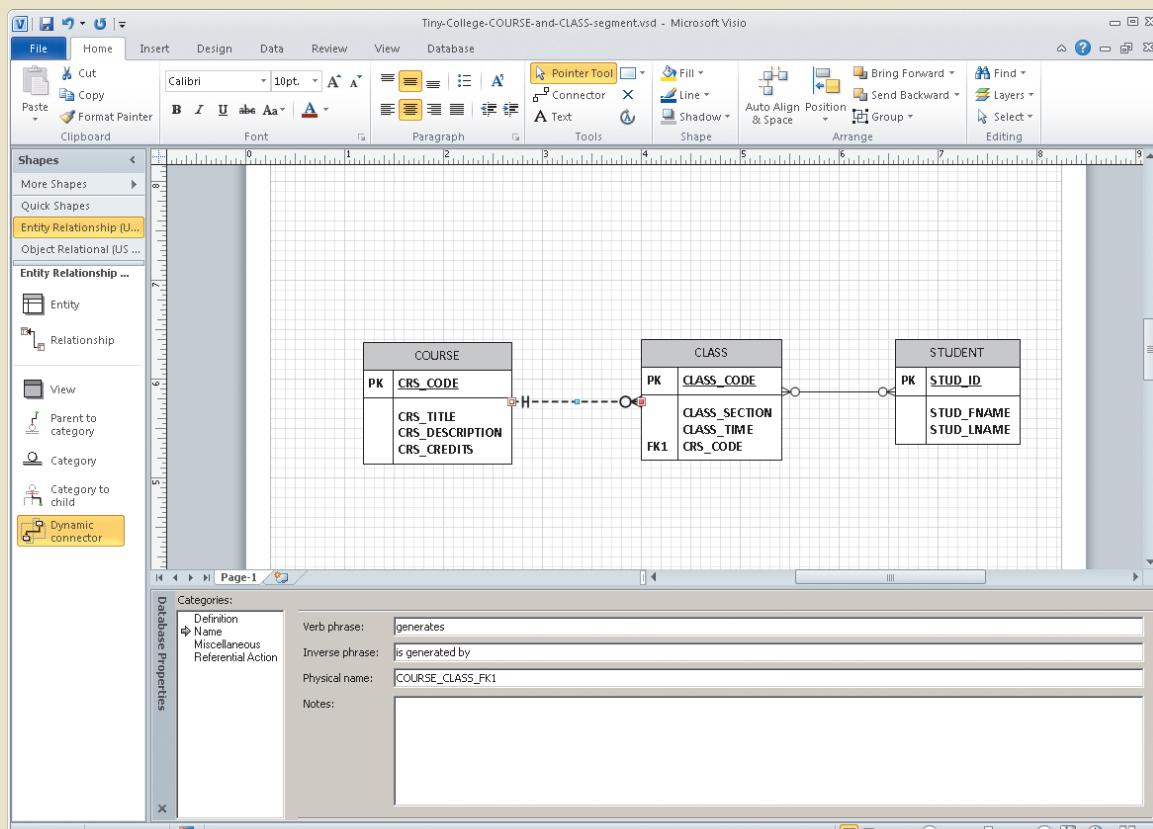
As you examine Figure A1.32, note that the default **Verb phrase** selection is **has** and that the default **Inverse phrase** selection is **is of**. It's useful to remember that:

1. All relationships are defined in both directions – from the “1” side to the “M” side, and from the “M” side to the “1” side.
2. Active verbs are typically used to label relationships from the “1” to the “M” side. Passive verbs are used to label relationships from the “M” to the “1” side.
3. Relationship names are written in lowercase.

Using the **Name** selection in Figure A1.32, type the **Verb phrase** and **Inverse phrase** entries “generates” and “is generated by” as shown in Figure A1.33. Note that active and passive verbs have been selected to describe the relationship between COURSE and CLASS in both directions, respectively:

1. COURSE **generates** CLASS.
2. CLASS **is generated by** COURSE.

FIGURE A1.33 THE NAMED RELATIONSHIP



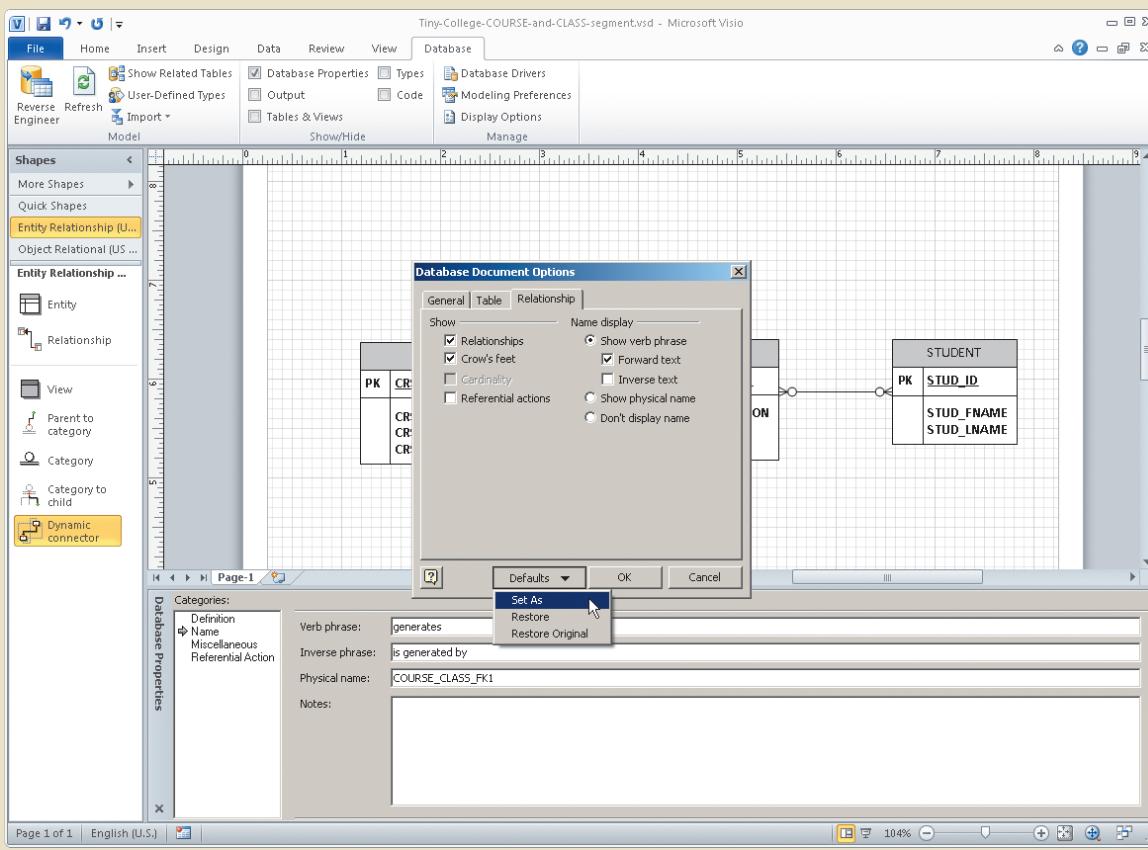
## A1-5e Showing the Relationship Names

As you examine Figure A1.33, you may note that the relationship names are not shown. If you *do* want those relationship names shown, click the **Database** ribbon shown at the top of the screen and then select **Display Options** as you first saw in Figure A1.5.

Next, select the database document options **Relationship** tab (see Figure A1.34), select the **Show verb phrase** option, and then select the **Forward text** option and unselect the **Inverse text** option. (If you select *both* the **Forward text** and the **Inverse text** options, Visio writes the two relationship names on the same line and separates them with a slash. That option takes more space, so you may have to move the tables farther apart to make the relationship names readable.) Finally, select the **Defaults** button and choose **Set As** as shown in Figure A1.34, and then click the **OK** button to set the selection as the default.

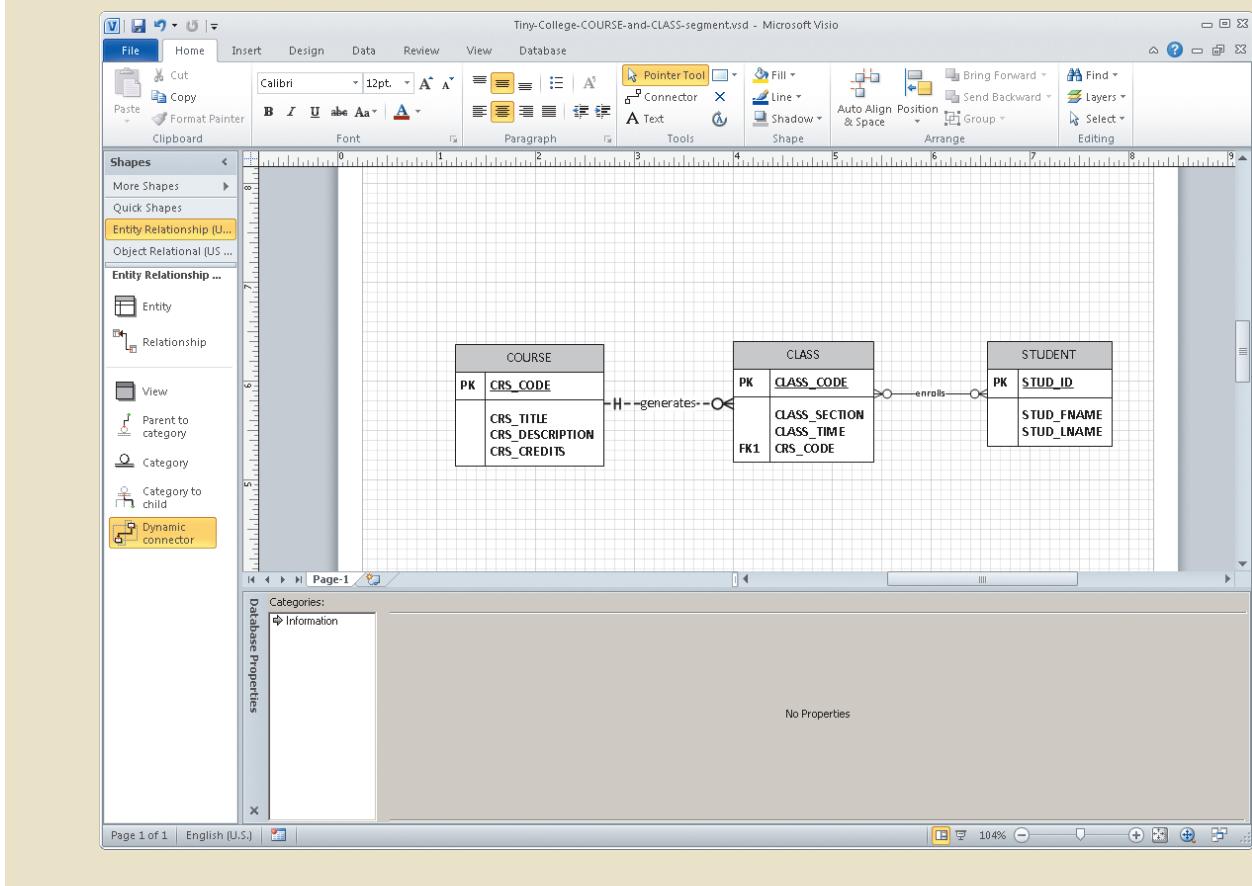
## A1-34 Appendix A1

FIGURE A1.34 SET THE RELATIONSHIP NAMES AS THE DEFAULT



To add a name to the M:M relationship that was created using a dynamic connector, click to select the connector and simply begin typing. Remember, dynamic connectors only provide a visual representation – Visio is not actually tracking or maintaining a relationship through the dynamic connector. Figure A1.35 shows the ERD with the relationship names visible.

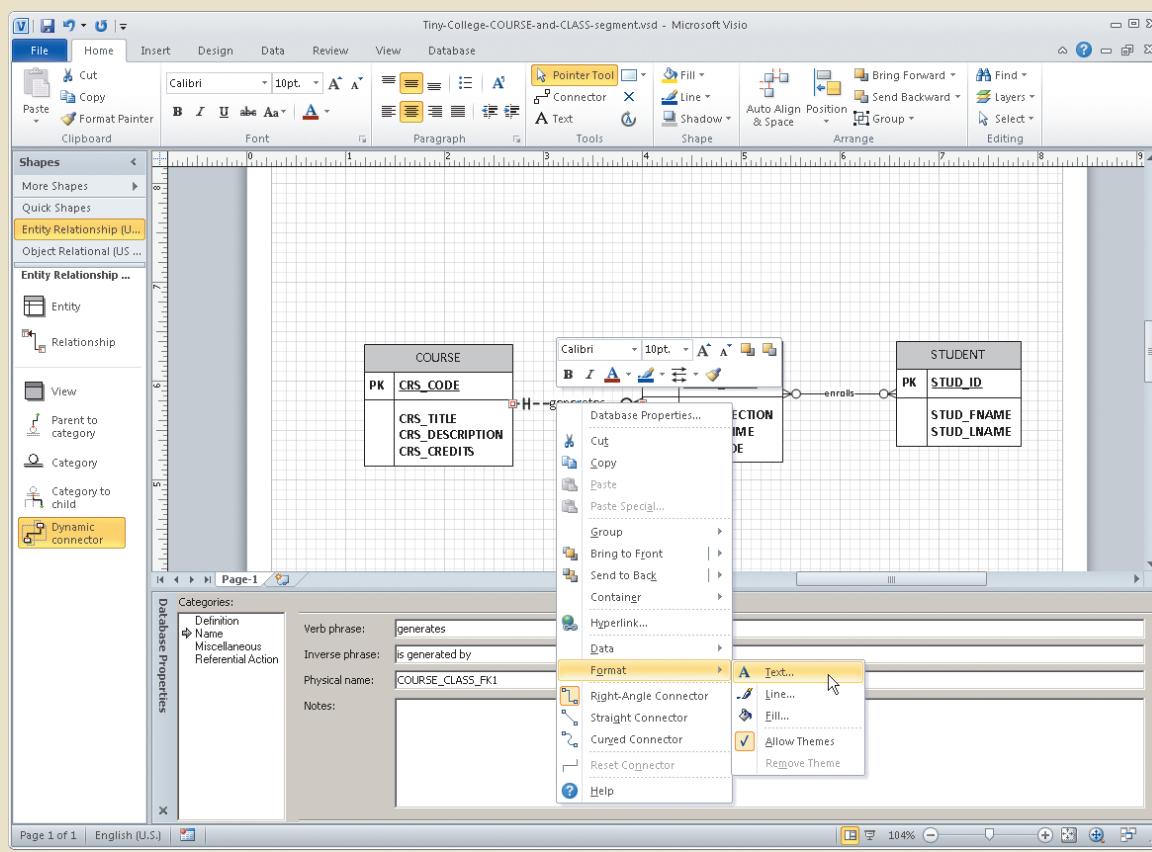
FIGURE A1.35 SHOWING THE RELATIONSHIP NAMES



As you can tell by looking at the relationship name in Figure A1.35, it is written through the relationship line, thus making it difficult to read. You can change the placement of the relationship name through font control. For example, if you want to place the relationship name above the relationship line, right-click the relationship and choose the **Format → Text...** selection shown in Figure A1.36.

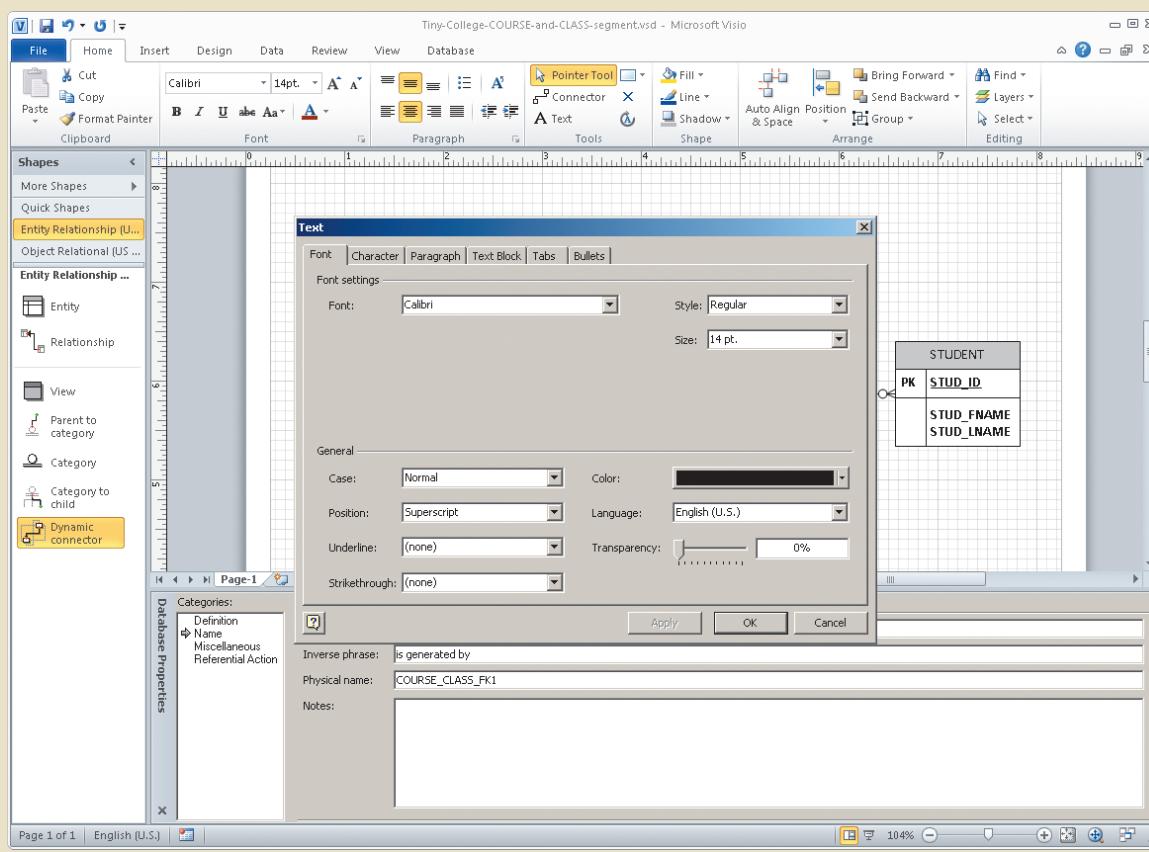
## A1-36 Appendix A1

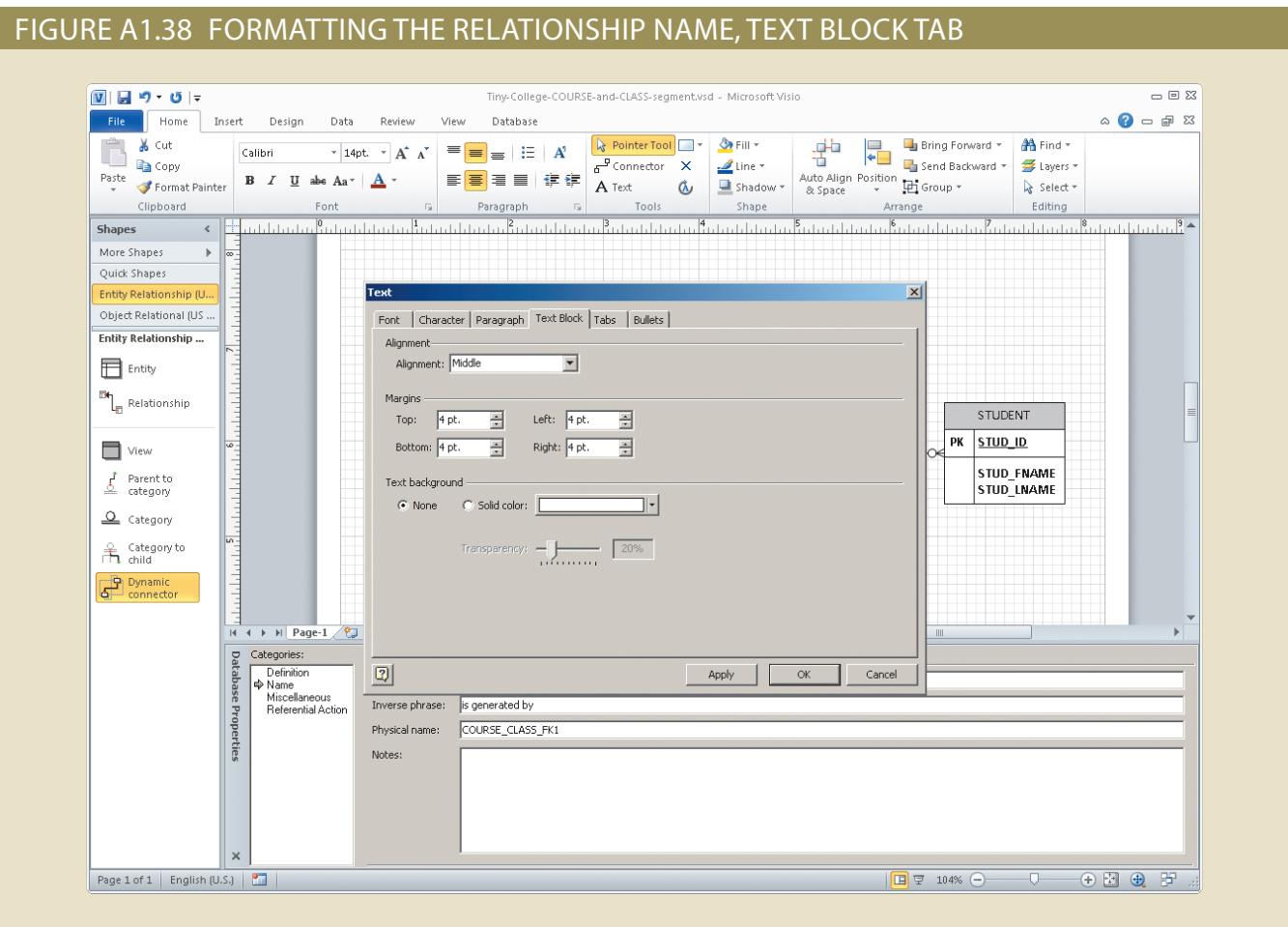
FIGURE A1.36 SELECTING THE RELATIONSHIP NAME TEXT FORMAT



When you click the **Text...** selection shown in Figure A1.36, you will see the window in Figure A1.37. On the **Font** tab, change the **Position** to be **Superscript** and the font **Size** to be **14 pt**. On the **Text Block** tab, change the **Text background** option to **None** as shown in Figure A1.38. Click **OK** to apply these changes. Then make the same changes to the text for the M:M relationship.

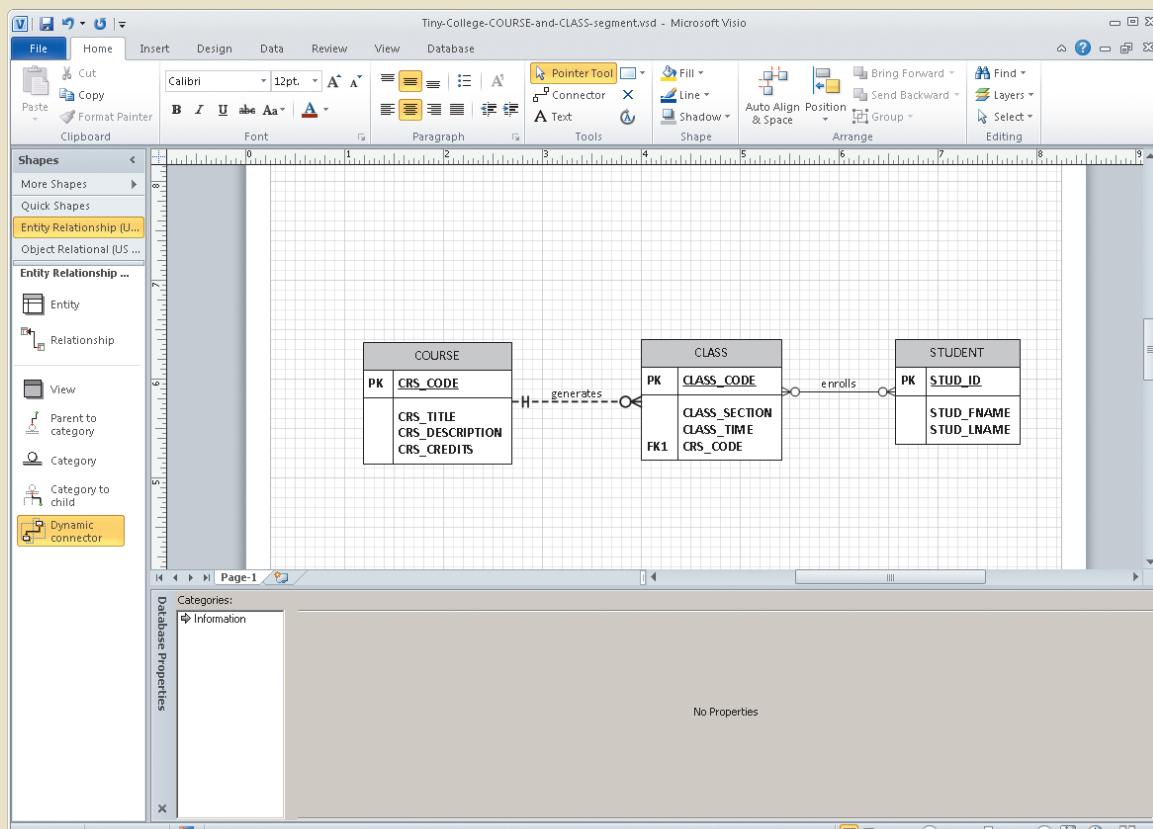
FIGURE A1.37 FORMATTING THE RELATIONSHIP NAME, FONT TAB



**A1-38 Appendix A1**

After you have made these changes to both relationships, the relationship names will appear above the relationship lines as shown in Figure A1.39. (The relationship line has been deselected by clicking an empty portion of the grid to make it easier to read the repositioned relationship name.)

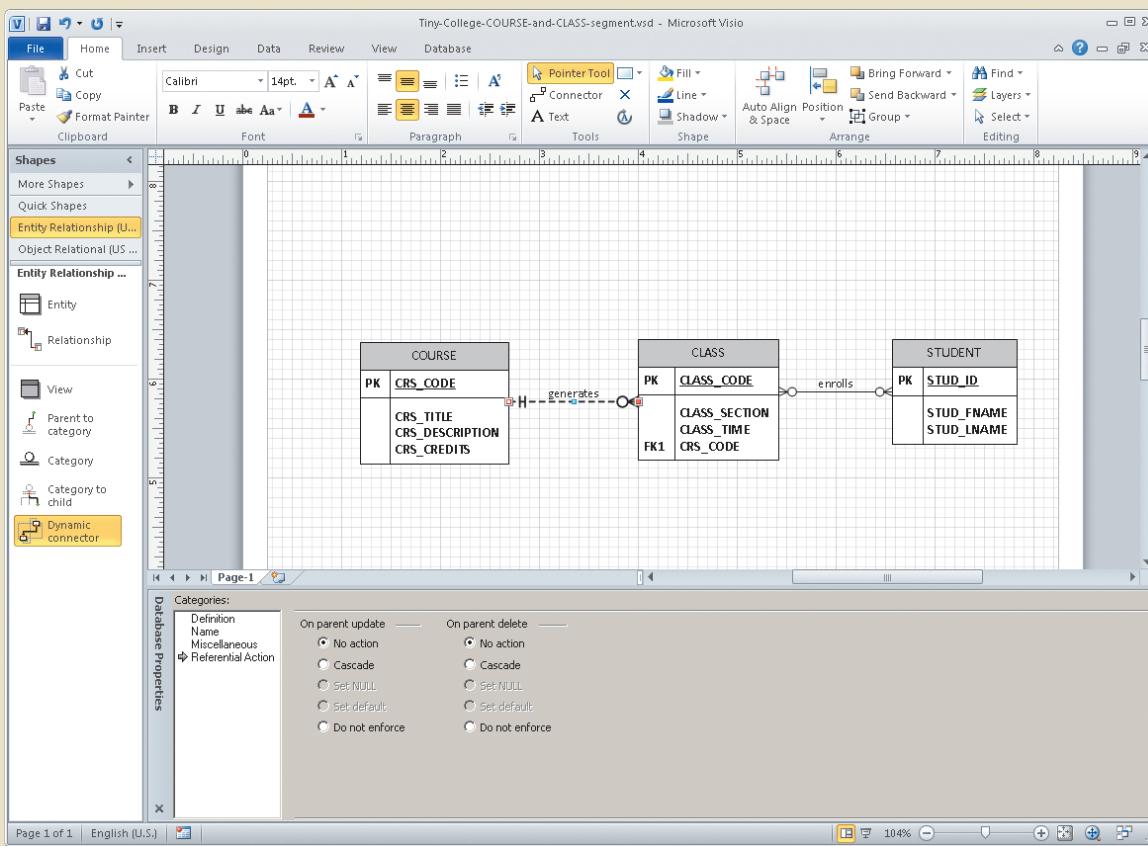
FIGURE A1.39 THE REPOSITIONED RELATIONSHIP NAMES



## A1-6 Referential Action

Select the “generates” relationship line, and then click the **Referential Action** option in the **Database Properties** window at the bottom of the screen to produce the results displayed in Figure A1.40.

FIGURE A1.40 THE DEFAULT REFERENTIAL ACTION



As you examine Figure A1.40, think of the consequences of a deletion in the parent (COURSE) table. For example, if a COURSE is deleted, do you want to delete all of the classes that are associated with that course? That is, do you want to **Cascade** the deletion? Similarly, if the CRS\_CODE for a row in the COURSE table changes, should that change be reflected in the CRS\_CODE of the related rows in the CLASS table? It is important to remember that referential integrity action describes what should be done in the child table in response to changes in the parent table. It does not impact what can be done in the parent table, but how that action is reflected in the related child rows.

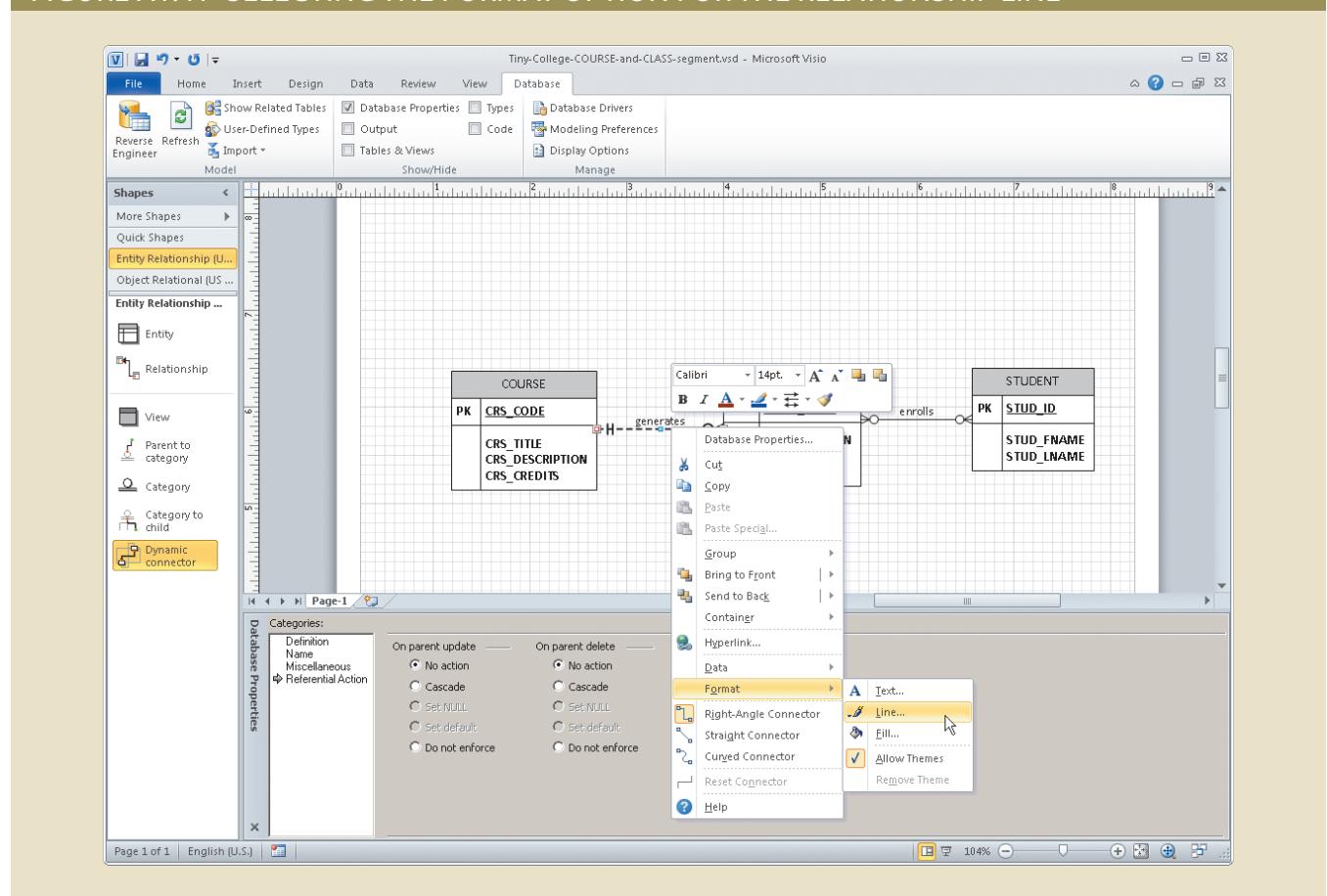
Since our relationship has mandatory participation from CLASS to COURSE (every CLASS must be associated with a COURSE), Visio indicates that we have limited options to maintain referential integrity when rows are deleted or primary keys updated in the parent table. We can either not define a referential integrity action (which is the default selection in Visio), we can cascade the change from the COURSE table into the CLASS table, or we can choose not to enforce referential integrity for that action. If the participation from CLASS to COURSE had been optional – that is, if a CLASS could exist without being associated with a COURSE – then Visio would have given the additional options of setting the value of the related FK in the CLASS table to either NULL or some default value when the value of the PK changes in the parent table or when rows are deleted.

The **Referential action** selection forces you to make sure that the database design is appropriate to the data environment and that you really do understand the ramifications of any database action. Given the many action options shown in Figure A1.40, you may want to create a small database and try each action to see its effect.

## A1-7 Controlling the ERD's Presentation Format

If you want to modify the ERD presentation format, Visio Professional provides many options. For example, if you want to color the relationship lines dark red, right-click the relationship line, and then select the **Format → Line...** option shown in Figure A1.41.

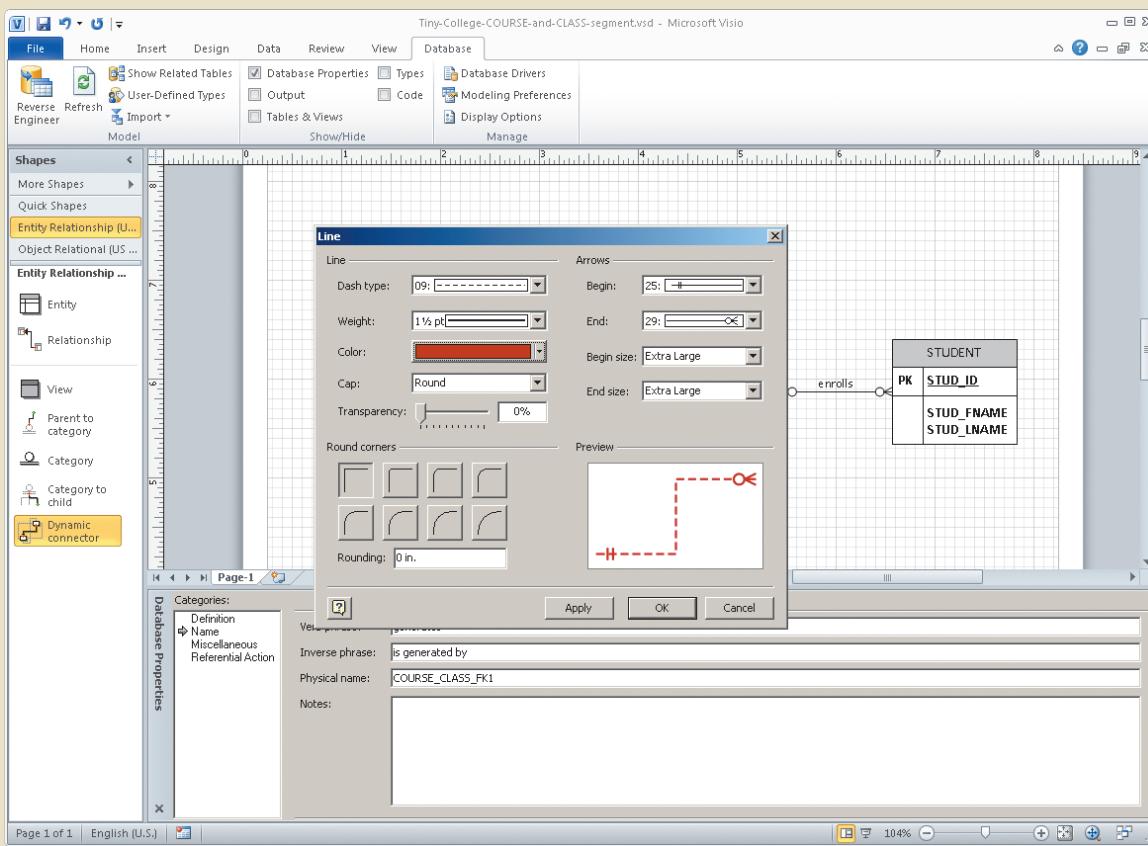
**FIGURE A1.41 SELECTING THE FORMAT OPTION FOR THE RELATIONSHIP LINE**



When you select the **Line...** option shown in Figure A1.41, you will see the options shown in Figure A1.42. Each selection option has its own drop-down list from which to make a selection. Note that the color **Dark Red** and the line weight **1½ pt.** have been selected. We have left the remaining options in their default settings. Click the **OK** button to accept the format changes shown in Figure A1.42. Make the same changes to the “enrolls” relationship line as well; however, also change the **Dash type** option to **09** so that the relationship line will be a dashed line indicating a non-identifying (weak) relationship.

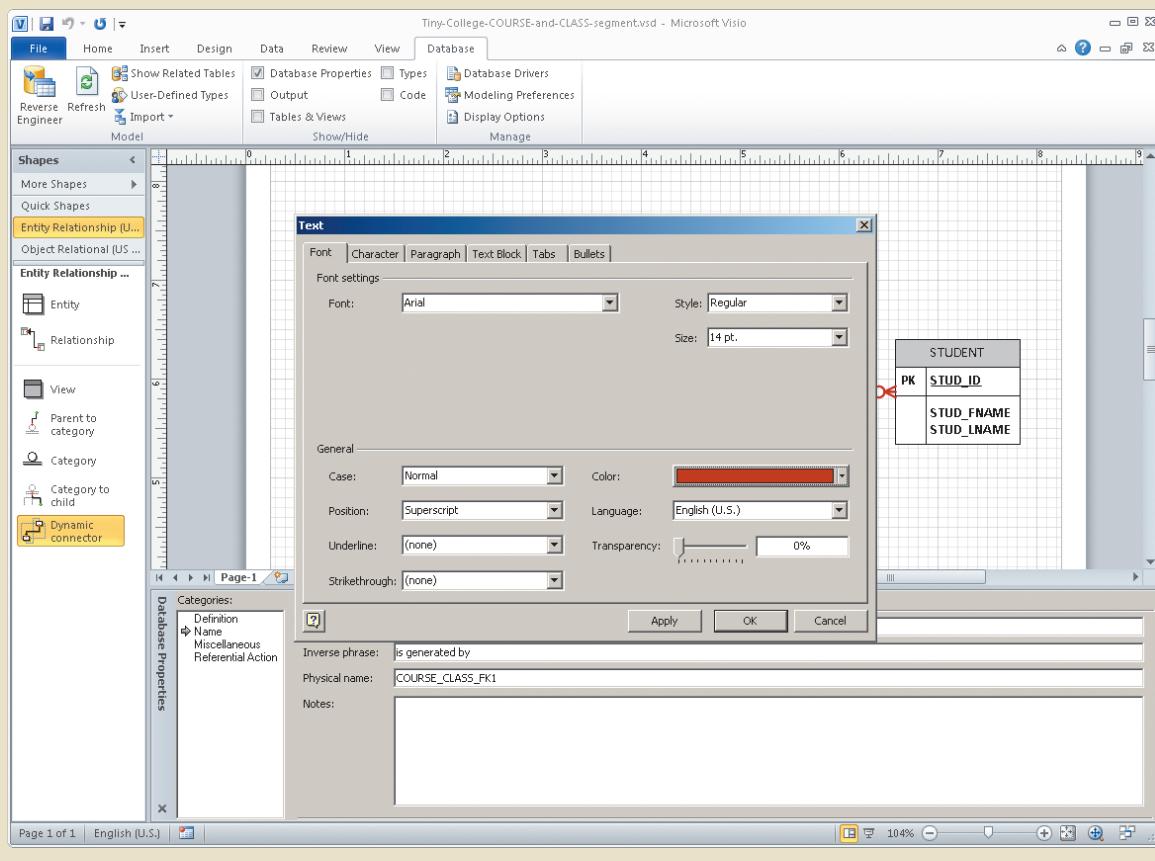
## A1-42 Appendix A1

FIGURE A1.42 FORMATTING THE RELATIONSHIP LINE



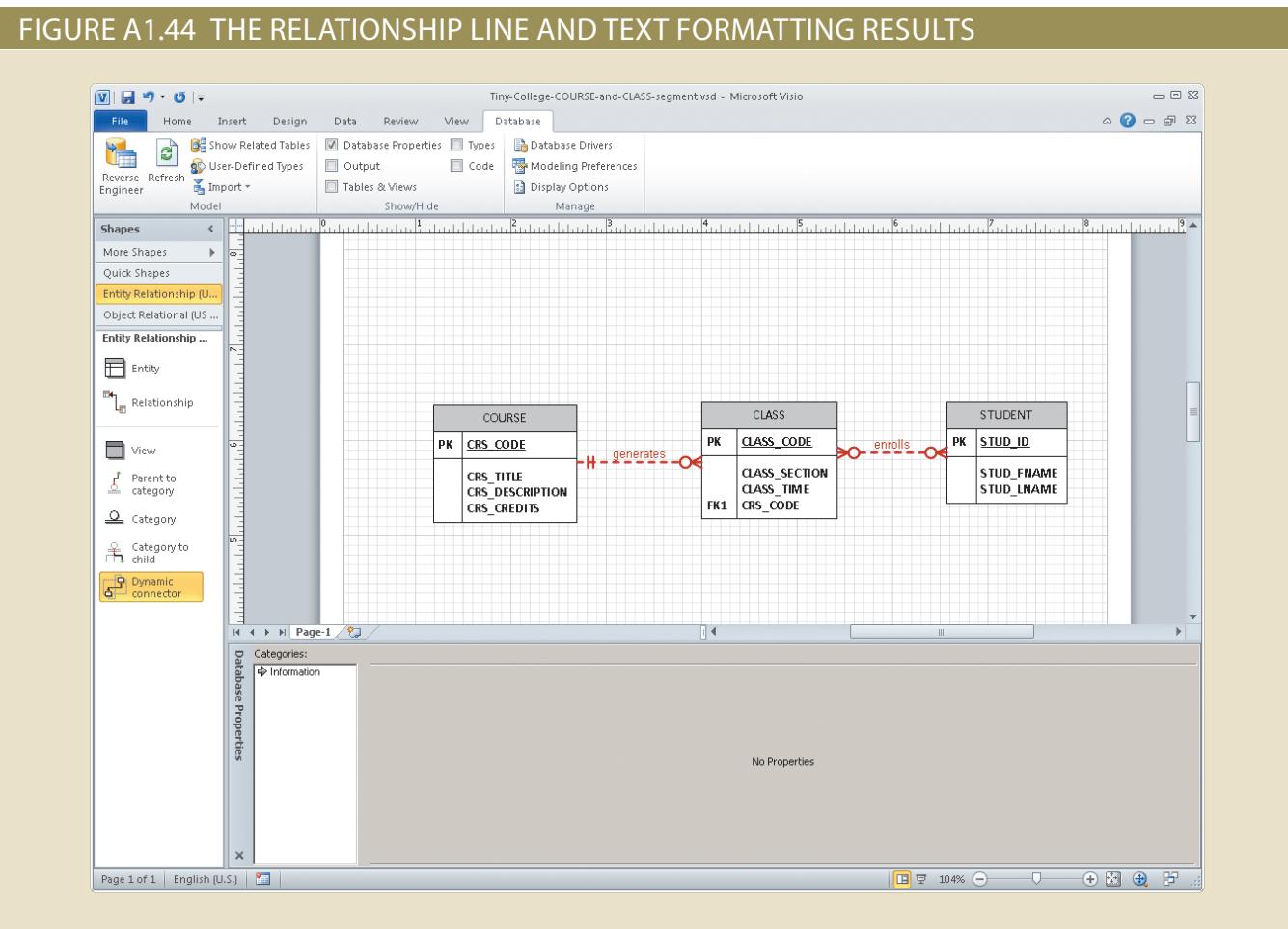
Using the same text formatting window used to reposition the relationship name, we can format the relationship name's text, too. To do that, select the **Format → Text...** option shown in Figure A1.36 to generate the window displayed in Figure A1.37. Be certain that the **Font** tab is selected. Select **Arial** as the font, and the **Dark Red** text color to match the color of the relationship line. The font **Size: (14 pt.)** and **Position: (Superscript)** reflect the choices made earlier. Make the same changes to the “enrolls” relationship line.

FIGURE A1.43 FORMATTING THE RELATIONSHIP TEXT



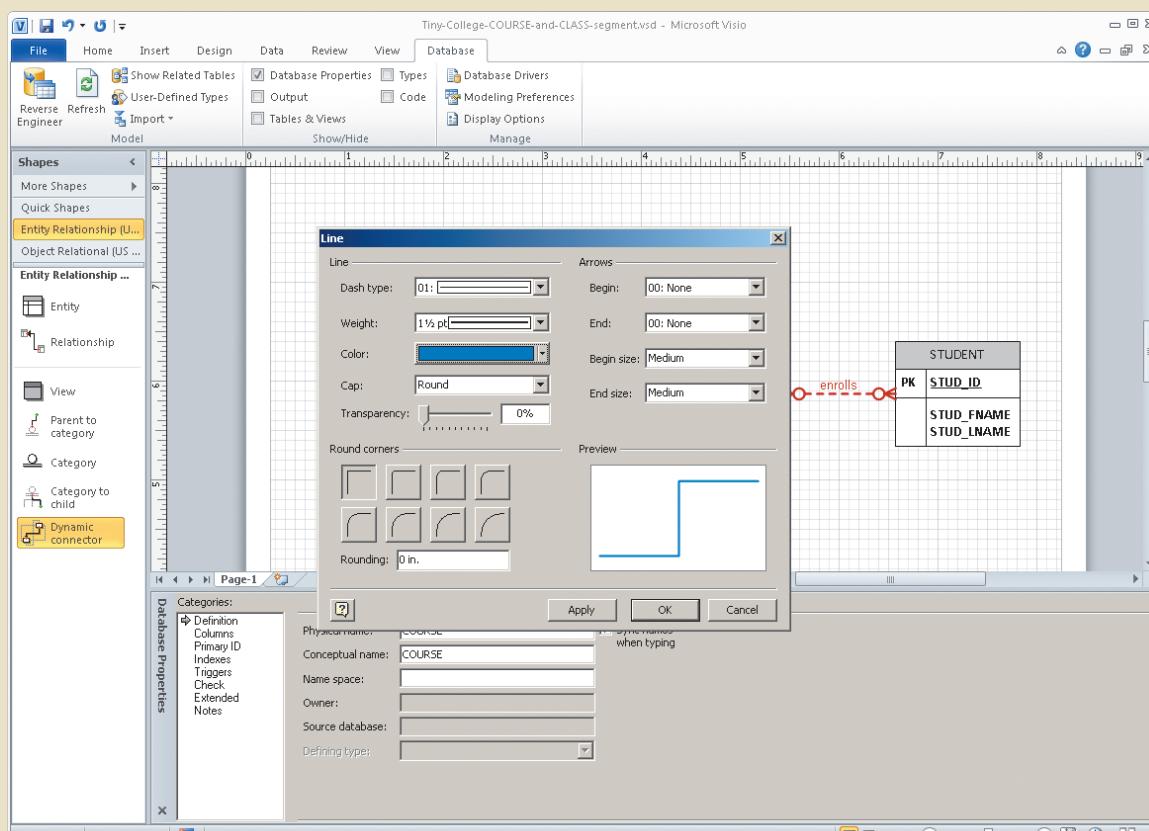
## A1-44 Appendix A1

The results of the relationship line and text formatting are shown in Figure A1.44.

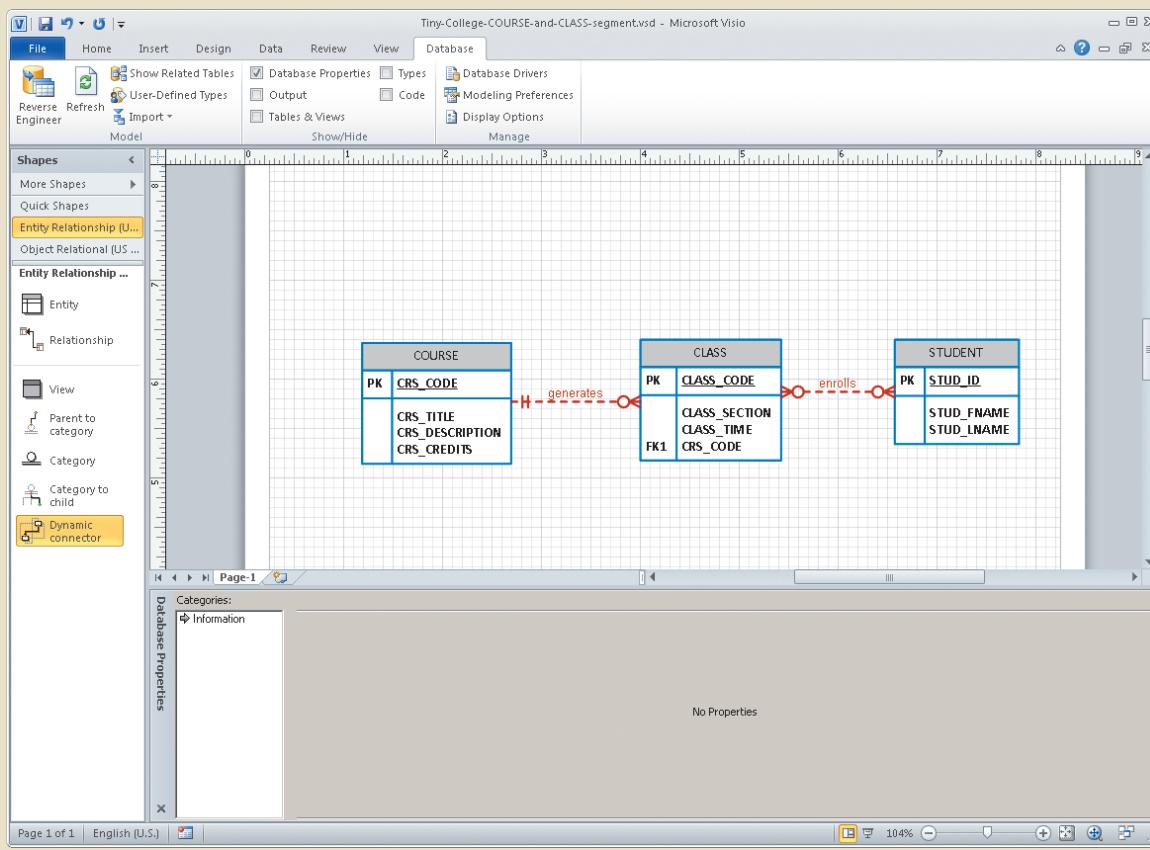


Naturally, you can also control the table's presentation format. To illustrate that process, let's make the table borders blue. To do that, right-click the table you want to format. Then select the format option (**Format → Line...**) to generate the line options shown in Figure A1.45.

FIGURE A1.45 FORMATTING THE TABLE LINE



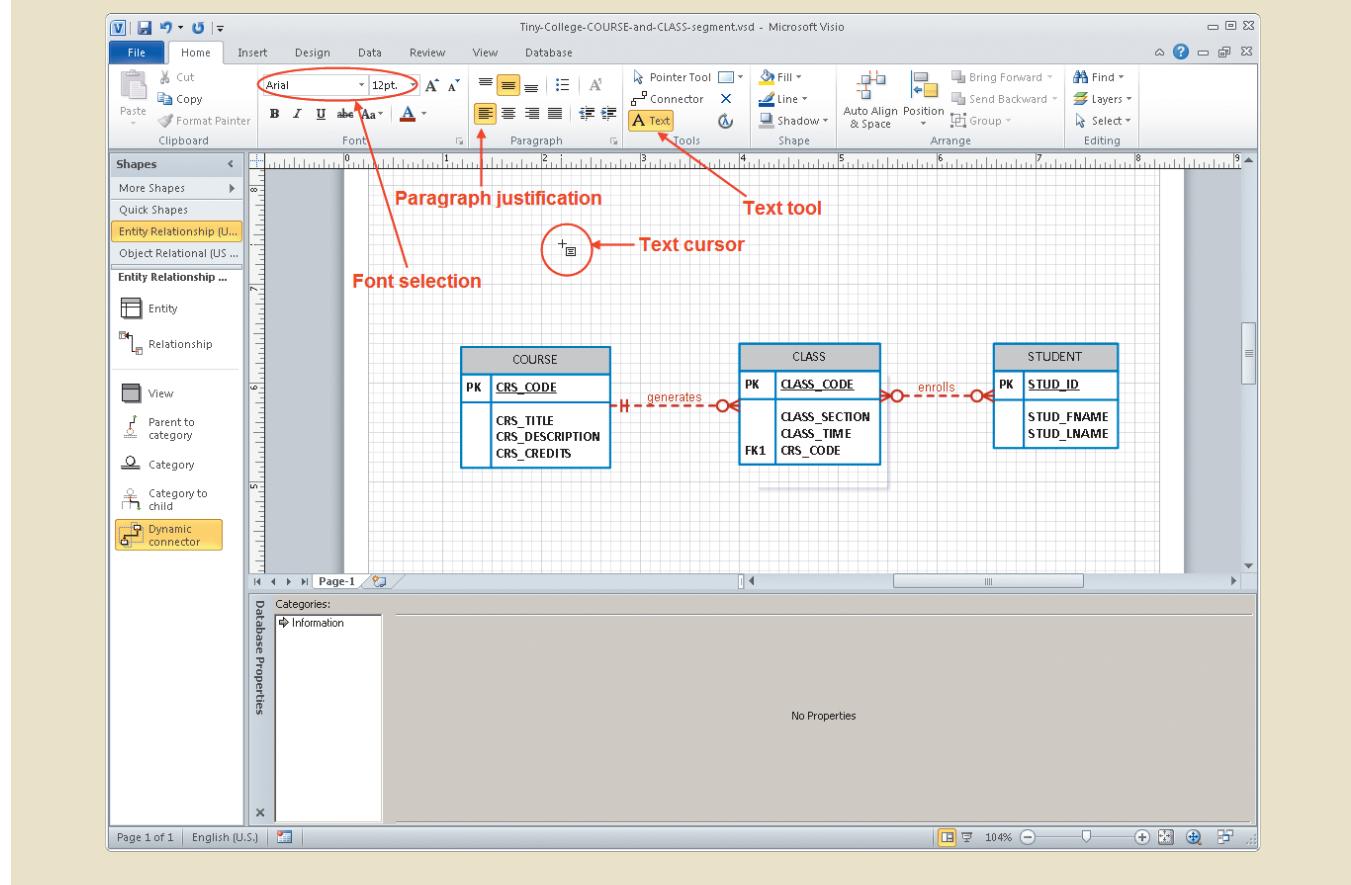
As you examine Figure A1.45, note that the line **Color:** was selected to be **Blue**. The selected line **Weight:** is **1½ pt**. Remember to click the **OK** button to save the changes. Now repeat the process for the CLASS and STUDENT tables to produce the results shown in Figure A1.46.

**A1-46 Appendix A1****FIGURE A1.46 THE REFORMATTED TABLE LINES**

## A1-8 Placing Text on the Grid

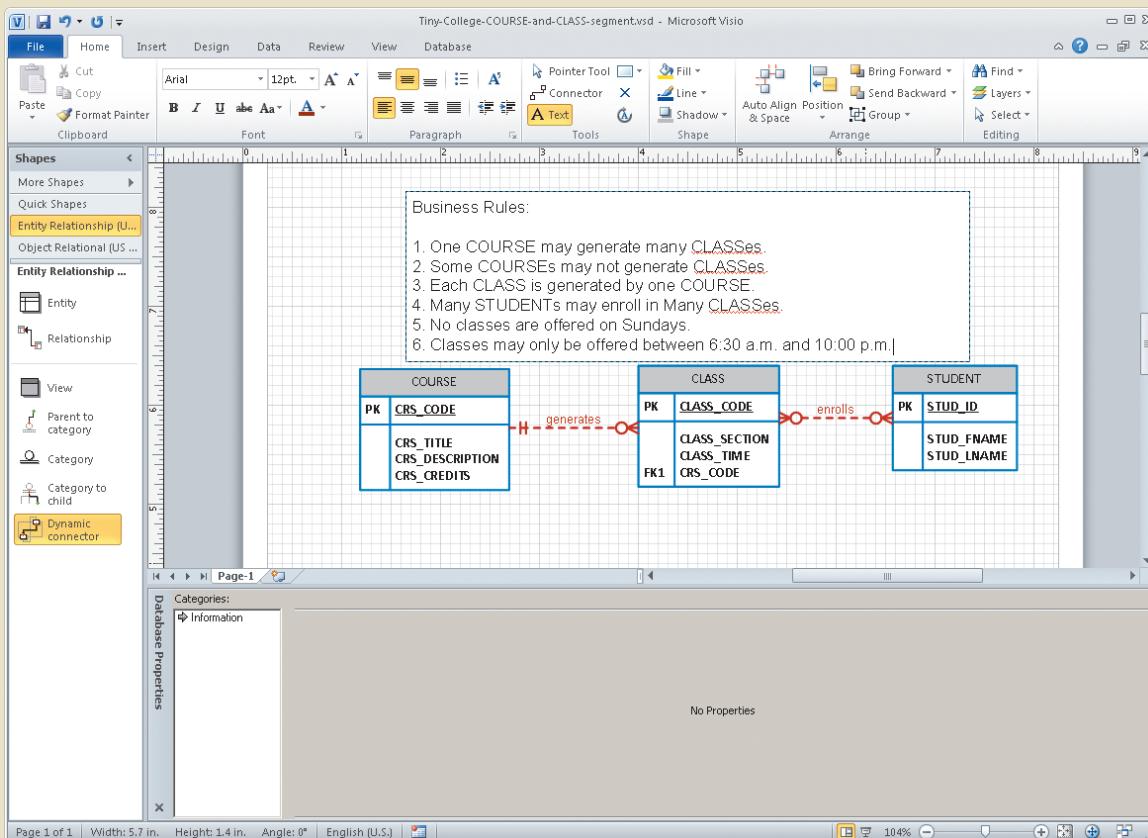
To help document the ERD, it may be helpful to place explanatory notes on the grid. Make sure that you have not selected any object by clicking a blank area of the screen. Select the **Text tool** (marked A) shown at the top of the screen on the **Home** ribbon. You will see the effect of your selection when you note the cursor's new look. Select the text format to suit your needs—left justification and a font face and size of Arial 12pt have been selected in Figure A1.47.

**FIGURE A1.47 SELECTING THE TEXT TOOL**



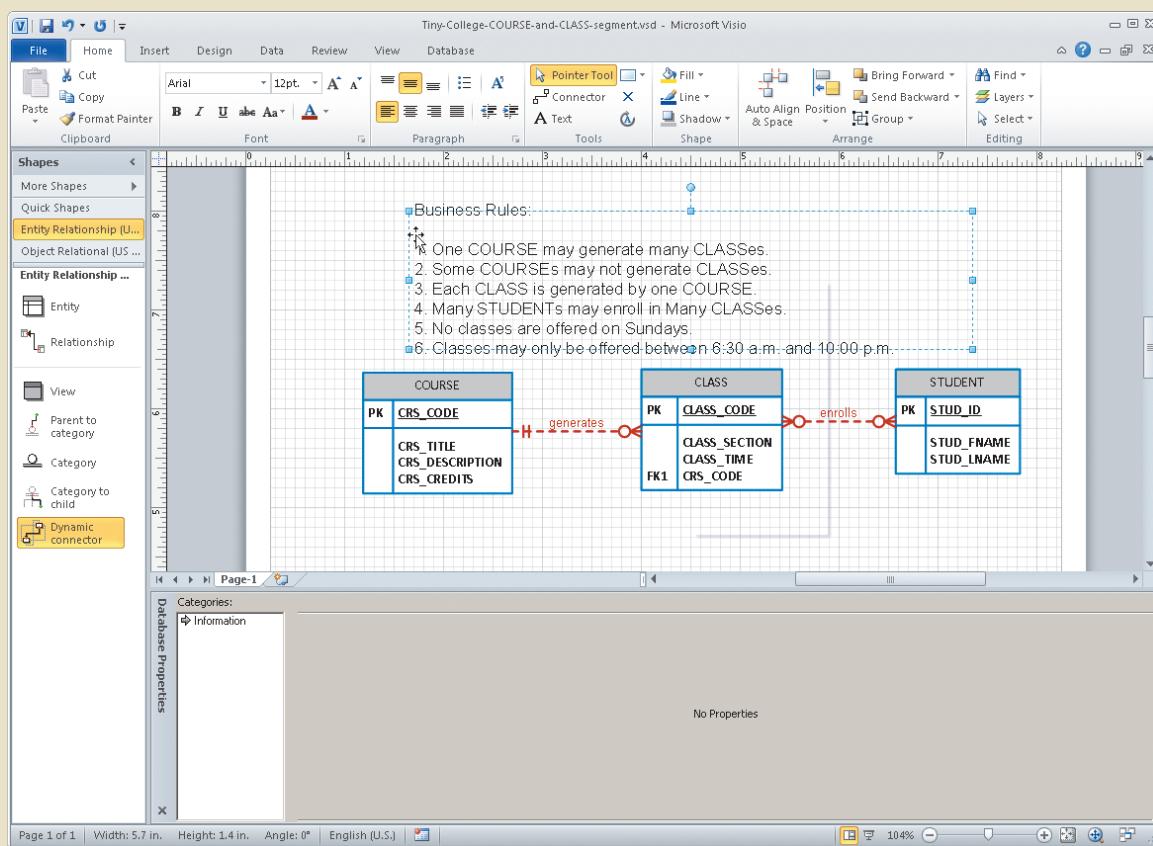
After making the selections shown, enter the text as shown in Figure A1.48. (You can modify any text format such as the font, size, color, and justification later.)

**FIGURE A1.48 THE INITIAL TEXT**



To move the text box, you must first make sure that the text tool has been deselected. If the text tool is still active, click the pointer tool. (You will know that the text tool is active when the cursor looks like the one shown in Figure A1.47.) If the text tool is not active, clicking the text box produces a set of small squares (handles) shown on the text box perimeter. You can see the handles around the text box in Figure A1.49. Also note that the four-sided arrow by the cursor indicates that the text box may be moved by dragging and dropping. (If you don't see the four-sided arrow on your screen, move the cursor until the four-sided arrow appears.)

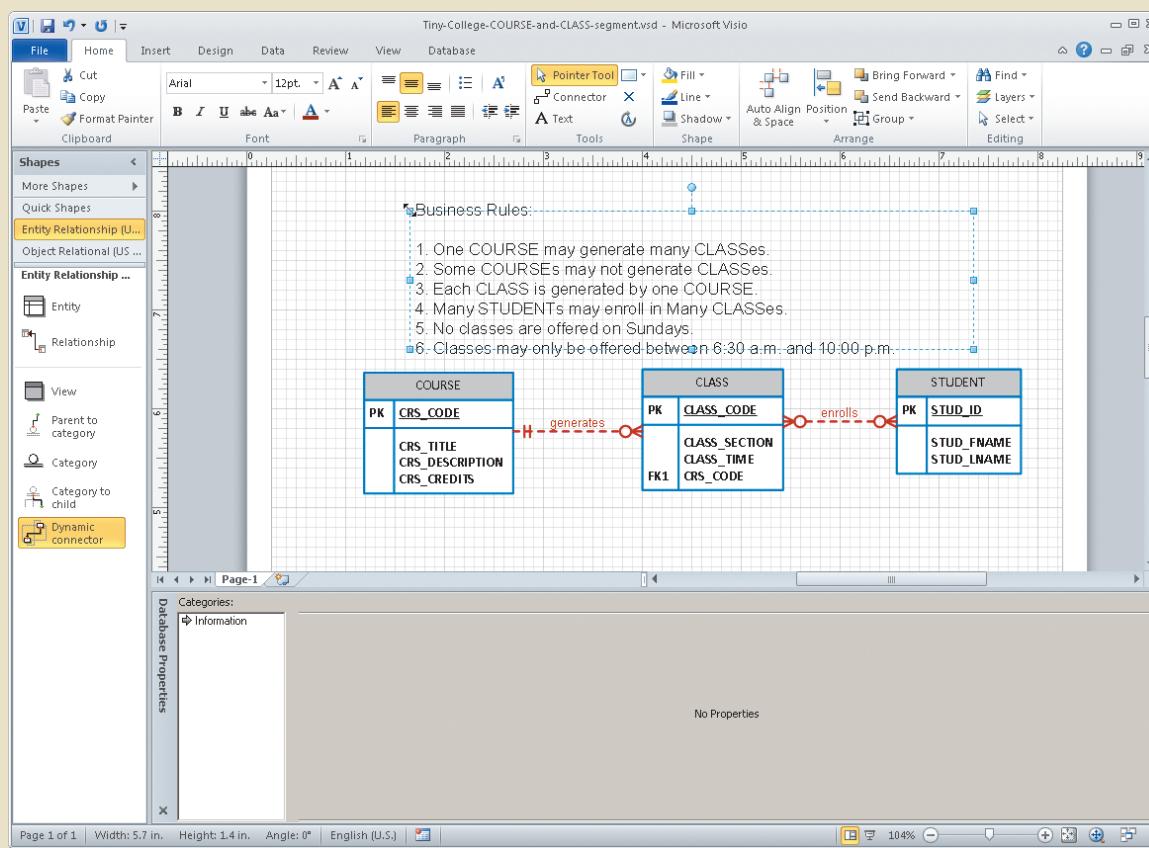
FIGURE A1.49 SELECTING THE TEXT BOX TO MOVE IT



After the text box has been selected as shown in Figure A1.49, you can drag and drop it as you would any other object on the screen. In fact, the text box behaves like any other Windows object. For example, you can change the size of the text box by dragging its perimeter to move it in or out. Just place the cursor anywhere along the text box perimeter to produce the two-sided arrow shown in Figure A1.50.

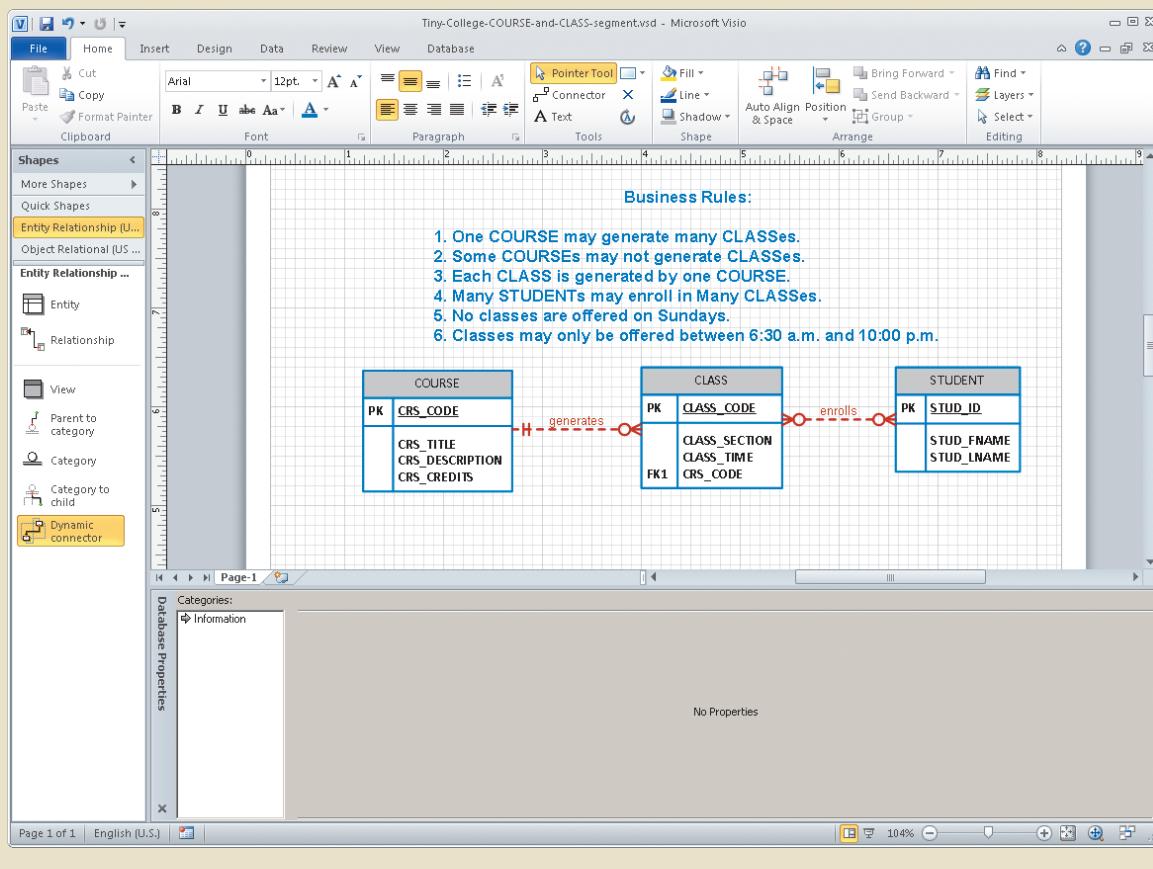
## A1-50 Appendix A1

FIGURE A1.50 SELECTING THE TEXT BOX TO SIZE IT



You can now finish typing the text, formatting it to suit your needs. The final text box is shown in Figure A1.51. Note that a blue text color and boldface has been selected.

FIGURE A1.51 THE COMPLETED TEXT BOX



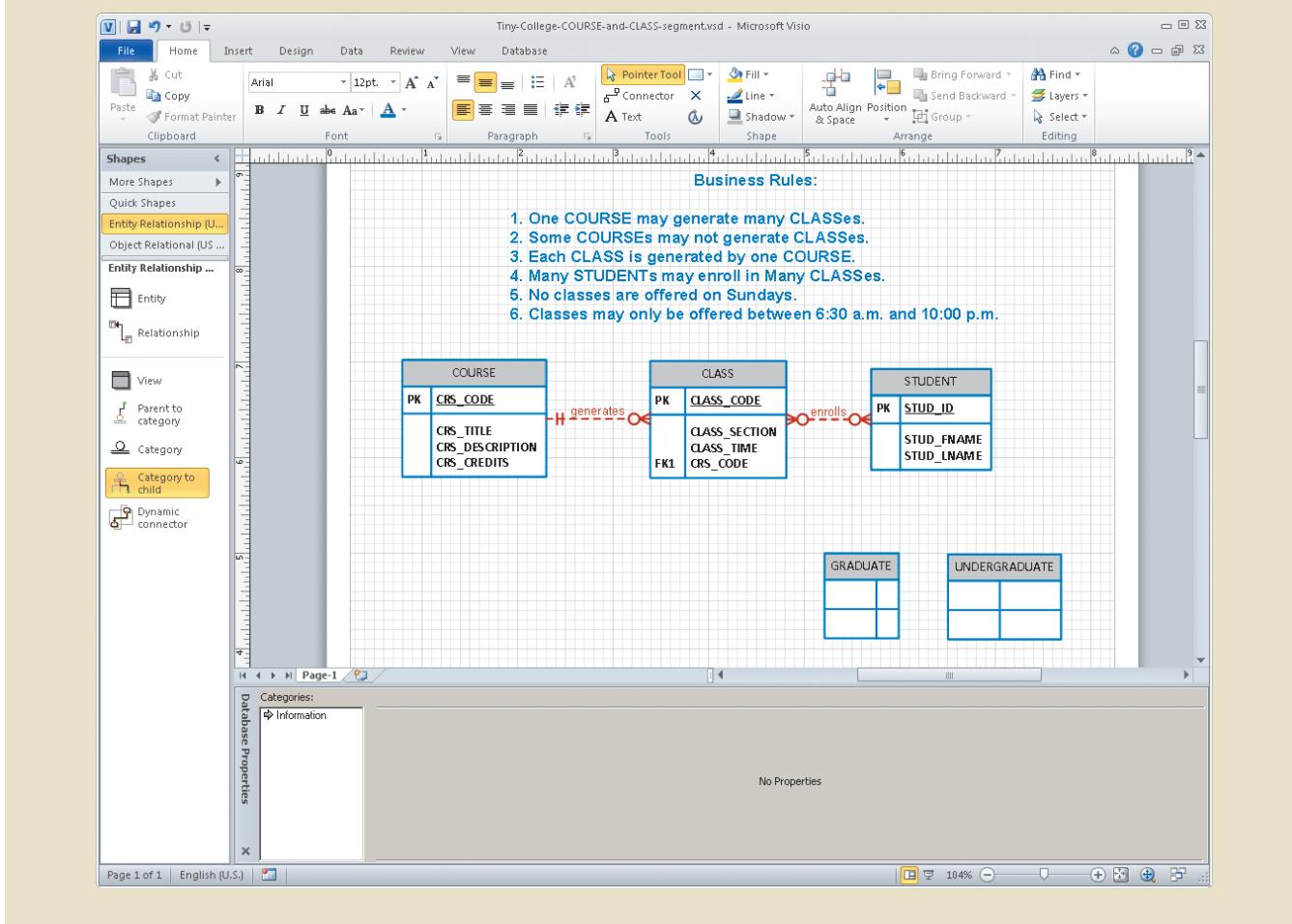
Don't forget to save your Visio file before you exit. As with all Windows applications, you will be reminded to save the file if you try to close it without first saving it.

## A1-9 Using MS Visio 2010 To Create a Specialization Hierarchy

Specialization hierarchies are used to depict a special type of relationship as discussed in Chapter 5, Advanced Data Modeling. These relationships rely on supertype and subtype entities. In Visio, these entities are created and manipulated in exactly the same way as the entities created previously in this tutorial. The only thing different about the specialization hierarchy is the modeling of the relationship.

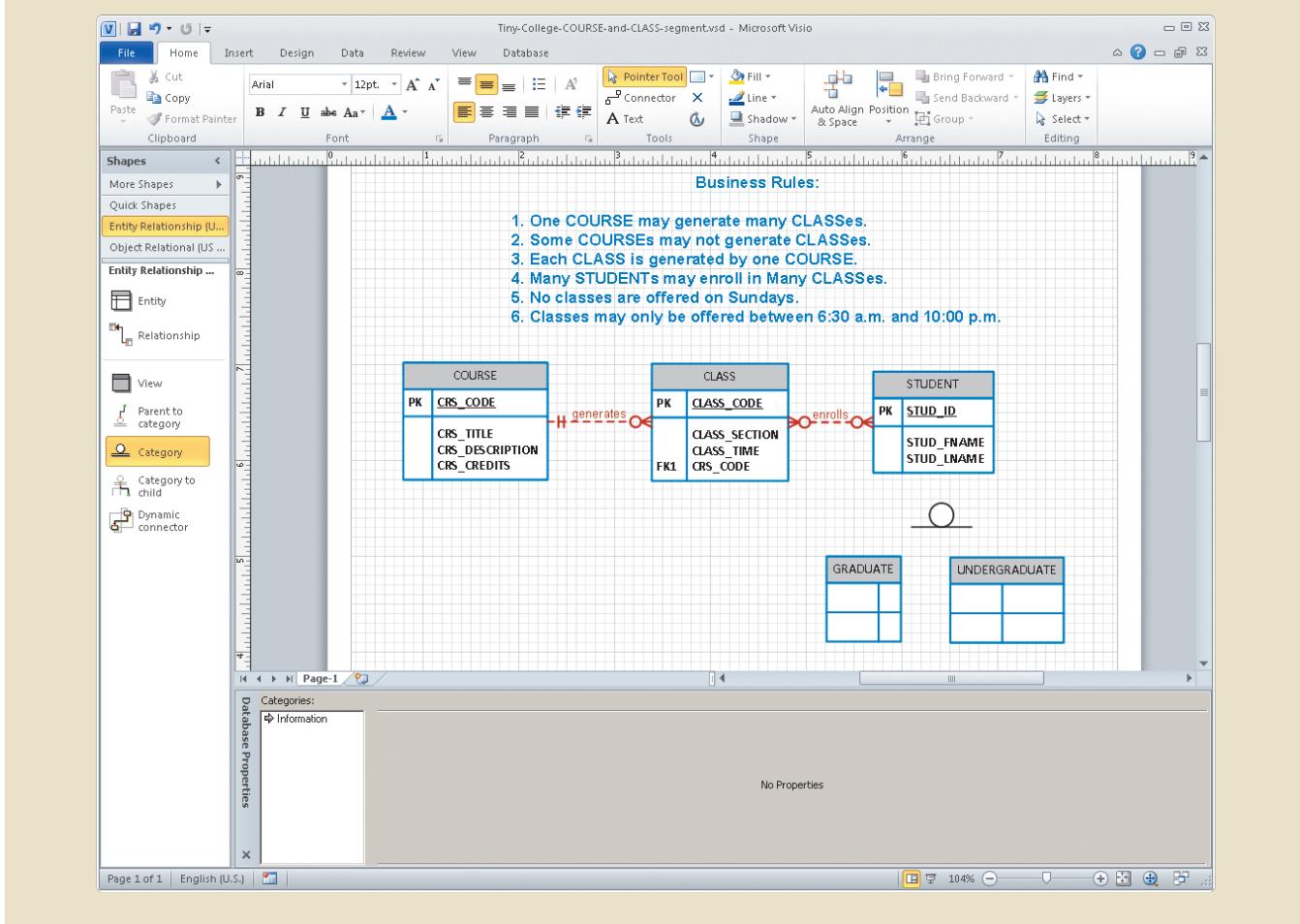
To begin, add two new entities named GRADUATE and UNDERGRADUATE to the existing data model as shown in Figure A1.52. These entities will be subtypes of STUDENT.

**FIGURE A1.52 ADDING GRADUATE AND UNDERGRADUATE SUBTYPES**



Visio uses Category notation to indicate a specialization hierarchy. The Category symbol is composed of a circle on top of one or two lines and is added to the diagram by dragging the **Category** object from the left side of the screen and dropping it on the drawing grid. Place a Category symbol between the STUDENT supertype entity and the subtype entities as shown in Figure A1.53.

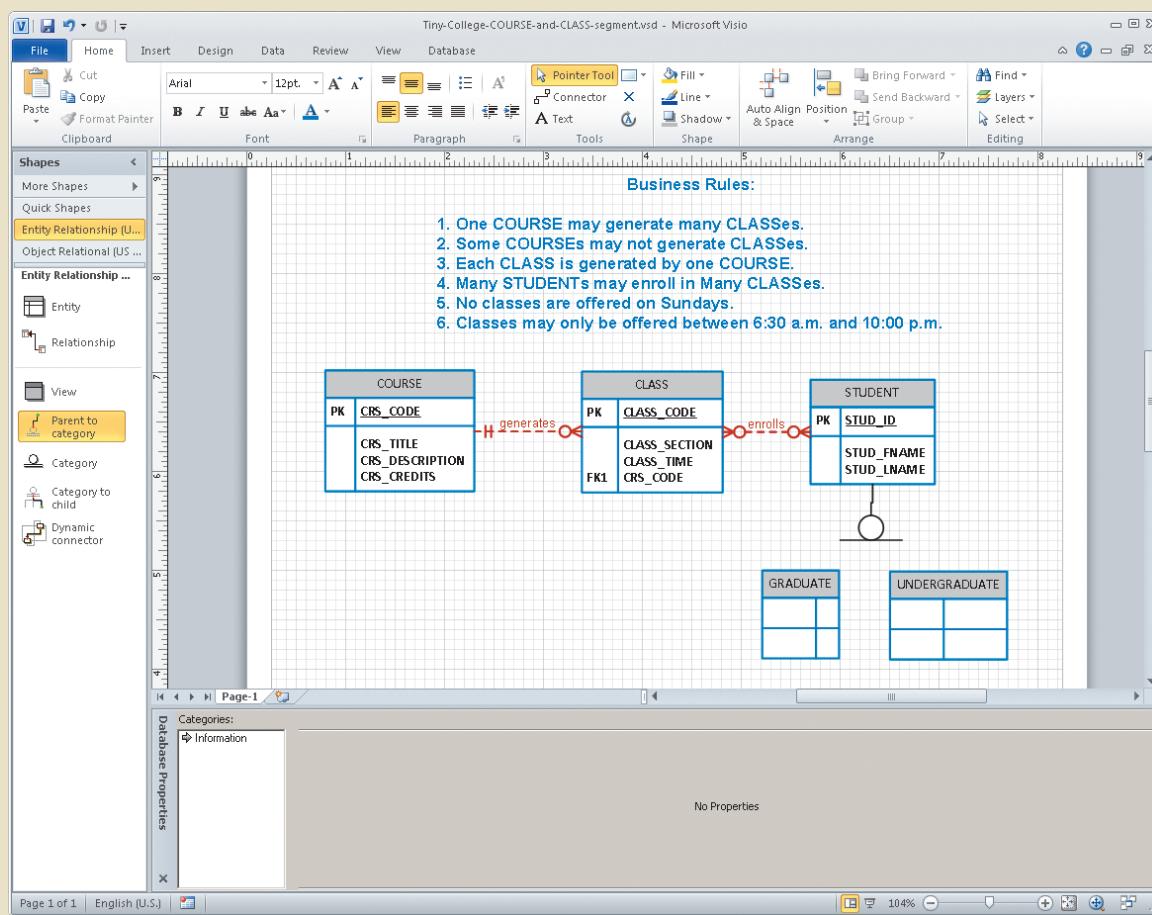
FIGURE A1.53 PLACING THE CATEGORY SYMBOL



The specialization hierarchy is modeled by relating the supertype entity, also called the parent entity, and the subtype entities, also called the child entities, to the Category object. Depicting these relationships on the diagram is the same as depicting other relationships except that instead of using the **Relationship** object or the **Dynamic connector** object, the **Parent to category** object is used to relate the supertype to the category and the **Category to child** object is used to relate the subtypes to the category. Use the **Parent to category** object to relate STUDENT to the category symbol as shown in Figure A1.54.

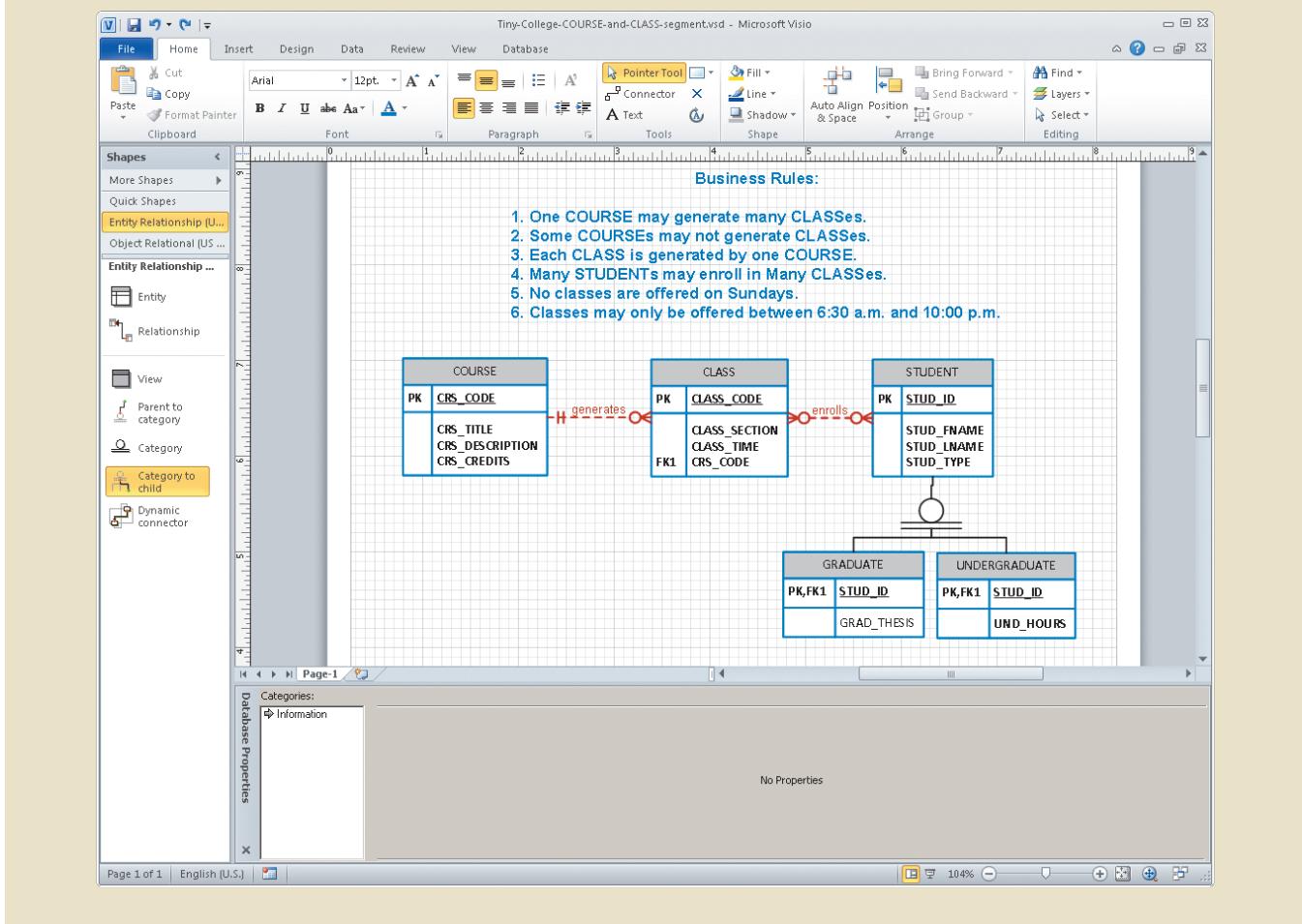
## A1-54 Appendix A1

FIGURE A1.54 RELATING THE SUPERTYPE



Use the **Category to child** object to relate the category to GRADUATE. Then, use **Category to child** again to relate the category to UNDERGRADUATE as shown in Figure A1.55. Note that when you relate the subtypes to the category, Visio automatically (and correctly) places the PK of the supertype as both the primary key and a foreign key in the subtypes.

FIGURE A1.55 RELATING THE SUBTYPES

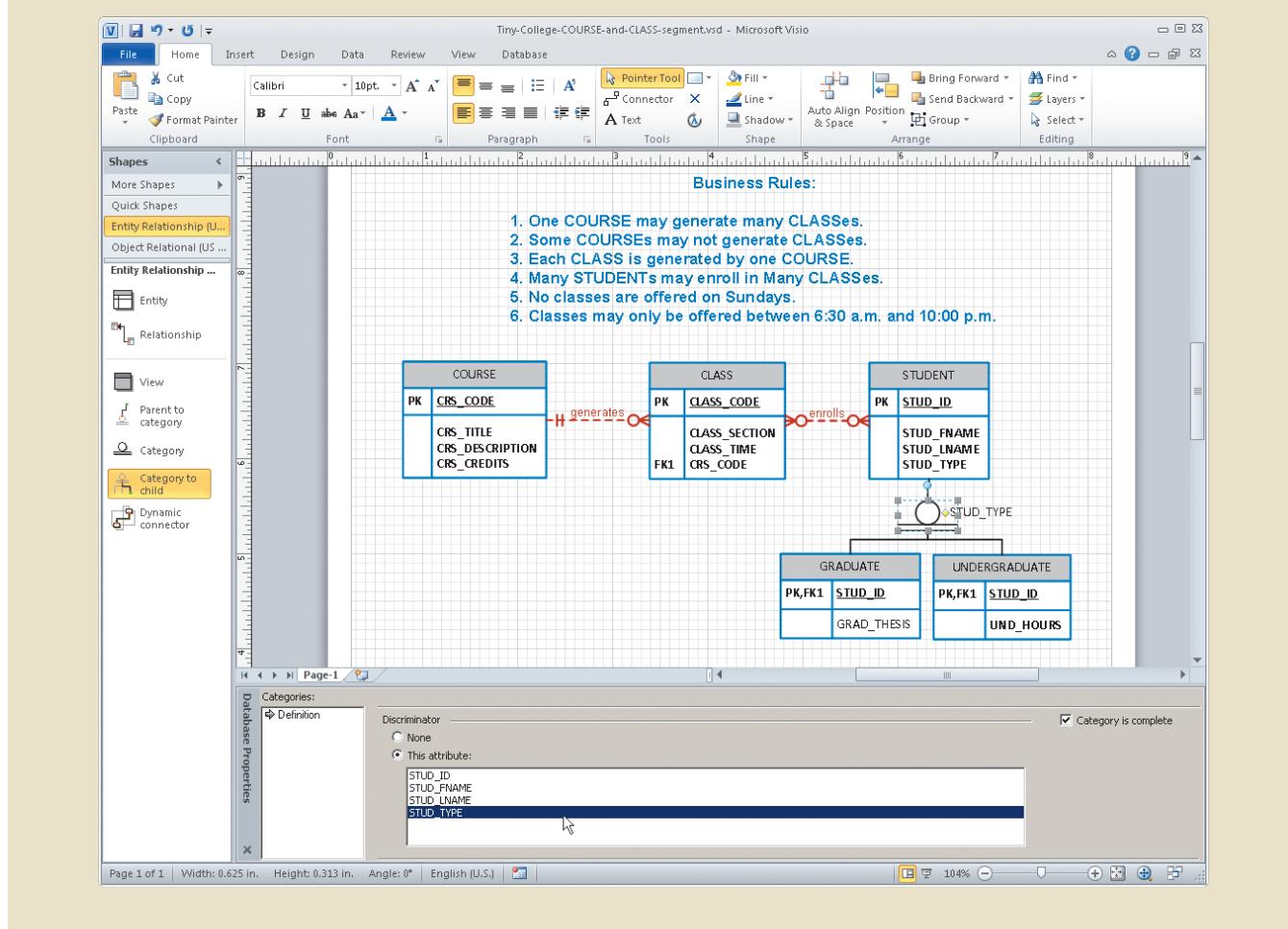


Both the supertype and subtypes can be edited to add additional attributes as necessary. This is done using exactly the same technique as manipulating the attributes of any entity. Add a required STUD\_TYPE attribute to the STUDENT entity to serve as a subtype discriminatory. Also, add GRAD\_THESIS to the GRADUATE entity, and required UND\_HOURS attribute to the UNDERGRADUATE entity.

The specialization hierarchy can be modeled as requiring total completeness or partial completeness. Recall from Chapter 5 that a value of “total” for the completeness constraint means that every entity instance in the supertype must be a member of at least one subtype, while a value of “partial” for the complete constraint means that it is optional for an instance of the supertype to be a member of any subtype. By default, the category symbol indicates a partial completeness constraint – it has a single line at the bottom of the symbol. Click the category symbol to select it, then in the **Database Properties** window at the bottom of the screen, click to mark that the **Category is complete**. Also in the **Database Properties** window, note that you can choose the subtype discriminator for the specialization hierarchy. Select **This attribute:** and choose STUD\_TYPE for the list of supertype entity attributes as shown in Figure A1.56.

## A1-56 Appendix A1

**FIGURE A1.56 CATEGORY OPTIONS**



Visio does not support adding text to category symbols. Therefore, to place the indicator for whether the subtypes are disjoint or overlapping, use the Text tool to place a “d” or “o” on the category symbol. The category symbol and the relationship lines that relate the supertype and subtypes to it can all be formatted using the same techniques previously described to format the other entities and relationships. The completed diagram is shown in Figure A1.57.

FIGURE A1.57 THE COMPLETED ERD

