DATAARCHITECT

# Getting Started

POWERDESIGNER 6.1

December 1997

# Contents

# About This Book

**Subject**

This book contains step-by-step tutorials for the PowerDesigner DataArchitect data modeling environment. It shows you how to do the following:

- Build a Conceptual Data Model (CDM) and a Physical Data Model (PDM)
- Use business rules and other model objects
- Generate triggers, procedures, extended attributes, scripts, and databases

**Audience**

This book is for anyone who will be building data models with PowerDesigner DataArchitect. Some familiarity with relational databases, SQL, and design methodology is helpful, but not required.

PART ONE

# CDM Tutorial

This part is an introduction to the DataArchitect conceptual data modeling environment.

CHAPTER 1

# About the CDM Tutorial

This tutorial is a series of eight lessons in which you learn how to use DataArchitect to build a Conceptual Data Model (CDM).

In this tutorial, you complete the CDM delivered in the tutorial file. This model is based on a real-life example of an information system. It provides a context for the exercises. As you build onto this model, you learn how to create all the basic elements of a CDM. You can then apply your knowledge to creating a CDM to suit your needs.

## What is a CDM?

A CDM represents the overall structure of an information system. It describes the conceptual relationships of different types of information rather than their physical structures. A CDM is independent of a particular database management system (DBMS).

## Why build a CDM?

In a CDM, you can put design issues first because you do not have to worry about the details of physical implementation. You deal principally with entities and their relationships. These are easy to understand and to manipulate.

Through a simple generation procedure, you can transfer the solid design framework of the CDM to the Physical Data Model (PDM). The PDM adapts your design to the specifics of a DBMS and puts you well on the way to complete physical implementation.

# What you will do

Chapter 2

You begin by preparing a business description and stating the business problems to solve. You will refer to this statement in order to decide what business rules, data items, and entities you need to create.

Chapter 3

You will start DataArchitect and open the tutorial CDM. This CDM presents the conceptual structure of a publishing enterprise.

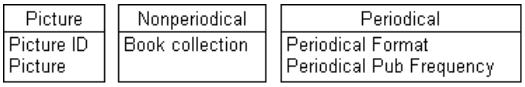
You will specify model preferences and properties, then save the model under a new name.

Chapter 4

You will create business rules, domains, and data items to complete the CDM. Next, you will apply check parameters to a domain.

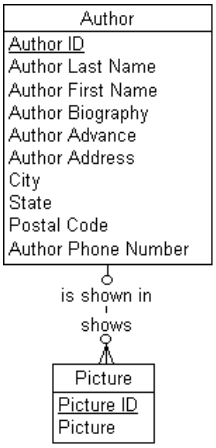
Chapter 5

You will add entities to the tutorial model. You will define entity attributes and designate an identifier. You will also attach a business rule to an entity.



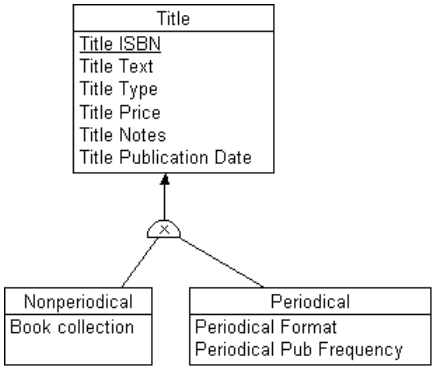

Chapter 6

You will create relationships between entities. You will create relationships with different cardinalities, as well as mandatory and dependent relationships.

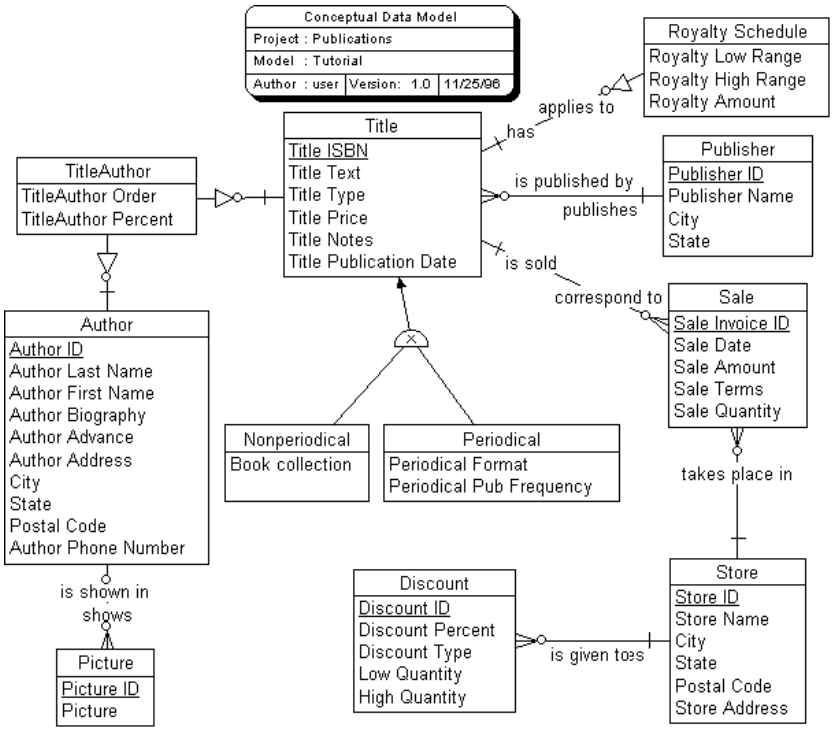


Chapter 7 You will create a submodel to which you will add entities.

Chapter 8 You will create an inheritance link using the entities in the submodel.




Chapter 9 You will use the display options and tools to organize the model. You will insert a title box and change its color. You will use the alignment tools to align symbols and straighten relationship lines.



| | |
|---|---|
| Chapter 10 | You will generate a Physical Data Model (PDM) from the CDM. You will define the generation parameters, generate the PDM, then save and close the PDM. |

## How long it will take

You can do the CDM tutorial in one sitting in about 1½ hours.

You can do the entire tutorial in about 2½ hours. You can stop after any chapter and continue at another time.

## What you will learn

You will learn basic DataArchitect techniques for creating a CDM, including:

- How to create the basic elements of a CDM: domains, data items, business rules, and entities
- How to create different types of relationships between entities
- How to create submodels and add objects to a submodel
- How to create an inheritance
- How to use the display functions to arrange symbols and to center the model on the page
- How to create submodels and add objects to a submodel
- How to generate a Physical Data Model (PDM) from a CDM

# Setting up

Before you begin, make sure that the files you need for the exercises are on your hard disk. When you install DataArchitect, these files are installed in subdirectories of your PowerDesigner directory. When you have finished with this tutorial you can delete them if you want.

The CDM tutorial uses the following files:

| File | Subdirectory | Description |
|---|---|---|
| CDMBEFOR.CDM | PD6\EXAMPLES | Starting tutorial CDM |
| CDMAFTER.CDM | PD6\EXAMPLES | Finished tutorial CDM |

CHAPTER 2

# Stating the Business Problem

The first step to creating a Conceptual Data Model (CDM) is to formulate a statement of the business problems that the CDM will address. To do this, you draw up a description of business activities. This helps you decide:

- What information you need to store
- What entities are involved in the business
- How the business operates

When the way the business operates is clear, you can begin to develop the CDM.

## Business description

You will start this tutorial with a CDM that is the conceptual model of the information system for a publishing company.

A business description indicates that the entities involved in the business include titles, authors, book stores, and discounts. It indicates the details related to each of these entities: from international standard book numbers (ISBN) for titles to store addresses.

## Business problem

In the course of the CDM tutorial, you will solve three business problems:

- **Royalties for multiple authors** Keep track of different royalty percentages for authors of the same book
- **Author pictures and biographies** Identify and store author pictures and biographies for use on book covers
- **Title categories** Classify each title as a periodical or a non-periodical and save related information

On the basis of these business problems, you will extend the tutorial model to manage additional data. For example, you will add an entity for pictures and you will create a data item to store information about percentages of royalties that a particular author receives.

CHAPTER 3

# How to Begin the CDM Tutorial

You will begin the tutorial by running DataArchitect. You will learn how to use the tool palette.

You will open the tutorial file and assign it preferences, options, and properties. Next, you will save it under a different name so that you can go back and use the original model again if you want to repeat the exercises.

In this chapter you will:

- Start DataArchitect
- Use the tools in the tool palette
- Open the tutorial CDM
- Define CDM preferences and options
- Define CDM properties
- Save the tutorial CDM

**How long will it take?**
About 10 minutes.

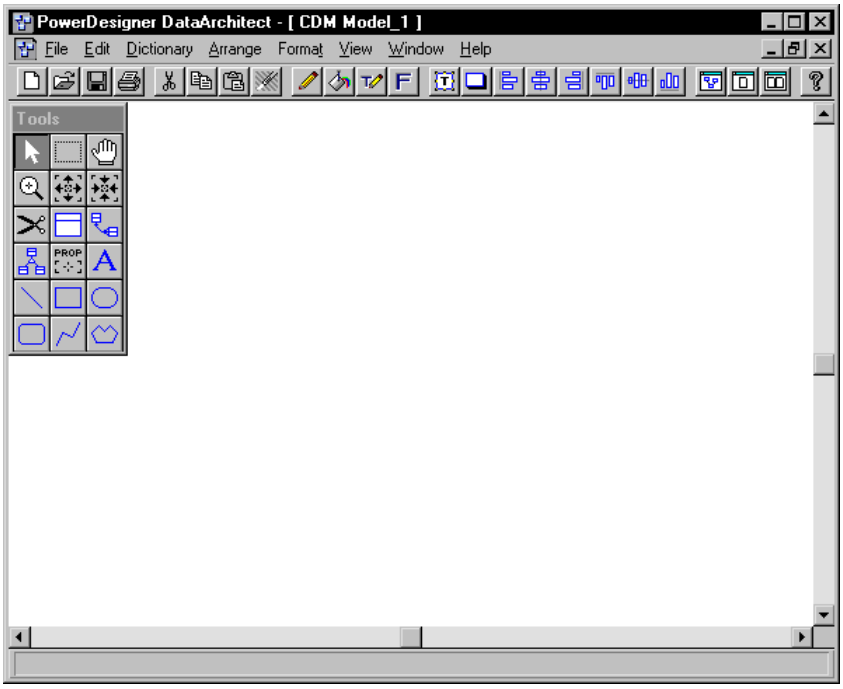
# Start DataArchitect

**Where you are**
Chapter 3 How to Begin
➡ Start DataArchitect
Use the tools in the tool palette
Open the tutorial CDM
Define CDM preferences and options
Define CDM properties
Save the tutorial CDM

**1 Click the *DataArchitect* program icon.**

The DataArchitect window appears. It contains an empty CDM workspace, and the CDM tool palette.

The DataArchitect window has a menu bar across the top. The menu bar gives you access to all the management functions.



**Your screen looks different**
All the screen captures in this book were taken in a Windows 95 environment, with an 800x600 screen resolution. The appearance and proportions of the images which appear on your screen may be slightly different.

# Use the tools in the tool palette

**Where you are**
Chapter 3 How to Begin
Start DataArchitect
➡ Use the tools in the tool palette
Open the tutorial CDM
Define CDM preferences and options
Define CDM properties
Save the tutorial CDM

The buttons in the CDM tool palette enact all major functions needed to build and modify a CDM.


**PDM tool palette**
The Physical Data Model (PDM) tool palette is almost identical. In the PDM tool palette, tools for creating PDM objects replace the tools for creating CDM objects.

The following table indicates the names of all tools in the tool palette.

| Tool | Name | Action |
|---|---|---|
|  | Pointer | Select symbol |
|  | Lasso | Select symbols in an area |
|  | Grabber | Select and move all symbols |
|  | Zoom | Select a zoom area |
|  | Zoom in | Zoom in (increase view scale) |
|  | Zoom out | Zoom out (decrease view scale) |
|  | Scissors | Delete symbol |

| Tool | Name | Action |
|---|---|---|
| | Entity | Insert entity symbol |
| | Relationship | Insert relationship symbol |
| | Inheritance | Insert inheritance symbol |
| | Property | Display object property sheet |
| | Text | Insert text |
| | Line | Draw a line |
| | Rectangle | Draw a rectangle |
| | Oval | Draw an oval |
| | Rounded rectangle | Draw a rounded rectangle |
| | Polyline | Draw a jagged line |
| | Polygon | Draw a polygon |

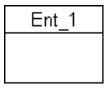
So that you know how to use these tools before you begin creating the CDM, you will create a few objects using the tool palette.

**1 Click the *Entity tool* in the tool palette.**

The cursor takes the form of an entity.

**2 Click anywhere in the CDM workspace.**

An entity symbol appears at the click position. The entity has the name Ent_*n*, where n is a number assigned to the entity in the order of creation of objects.



3 **The Entity tool is still active, so click again in the CDM workspace to create another entity.**

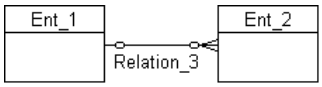
There are now two entities in the CDM workspace.


4 **Click the *Relationship tool* in the tool palette.**

The Entity tool is now released and the Relationship tool is active.

5 **Click inside the *first entity*.**
**Drag the *cursor* to the second entity.**
**Release the *mouse button* inside the second entity.**

This creates a relationship.

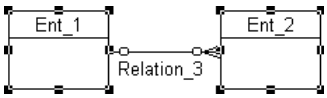



6 **Click the *right mouse button* to release the Relationship tool.**

> **To release a tool**
> A tool remains active until you release it. You can release a tool, by selecting another tool or by clicking the right mouse button. By default, the Pointer tool is selected.

7 **Double-click the *Entity tool* in the tool palette.**

This selects all the entities. Handles appear on the entities to show that they are selected.




> **Selecting all objects of a particular type**
> You can select all objects of a particular type by double-clicking the corresponding tool in the tool palette. You can select all objects of all types, by double-clicking the Pointer tool.

**8 Drag the entities to a new position.**

The relationship moves with the entities.

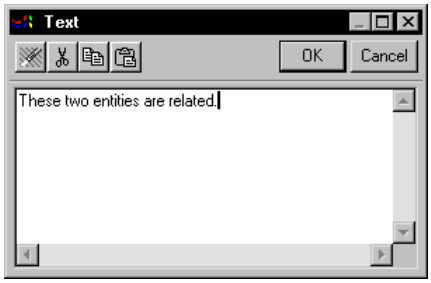
**9 Click the *Text tool* in the tool palette.**

The cursor takes the form of an I beam.

**10 Click the CDM workspace.**

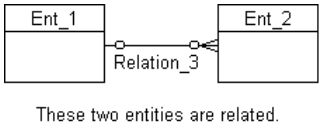
A text box appears.

**11 Type a short text into the text box.**




**12 Click *OK*.**

The text appears in the CDM workspace.




**13 Click the *Pointer tool* in the tool palette.**

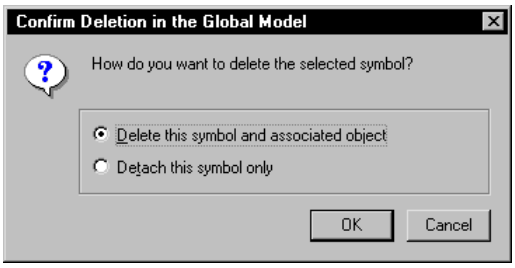
You will use this tool to select and delete one of the objects.

**14 Click one of the *entity* symbols.**

This selects the object you want to delete.

**15 Press the DEL key.**

The Confirm Deletion in the Global Model message box appears asking you how you want to delete the selection.




**Deleting objects**
If you select Delete Symbols and Associated Objects, you erase the graphic symbol and delete the definition of the object from the dictionary. If you select Detach Symbols Only you erase the graphic symbol, but keep the definition of the object in the dictionary.

**16 Click *OK*.**

The entity and associated relationship are removed from the CDM. Their definitions are deleted from the dictionary.

**17 Double-click the *Pointer* tool in the tool palette.**

This selects the remaining entity and the text.

**18 Press the DEL key and click *OK* when the deletion message appears.**

The remaining entity and the text are erased.

### What you learned

In this section, you learned how to use some of the tools in the tool palette. You now know how to:

- Select a tool
- Release the active tool either by selecting another tool or by clicking the right mouse button
- Select all objects of a particular type
- Move graphic objects
- Create text to document the CDM
- Delete objects

# Open the tutorial CDM

**Where you are**
Chapter 3 How to Begin
Start DataArchitect
Use the tools in the tool palette
➡ Open the tutorial CDM
Define CDM preferences and options
Define CDM properties
Save the tutorial CDM

To perform the rest of the tutorial, you must open the tutorial file.

**1 Select** *File*➤*Open***.**

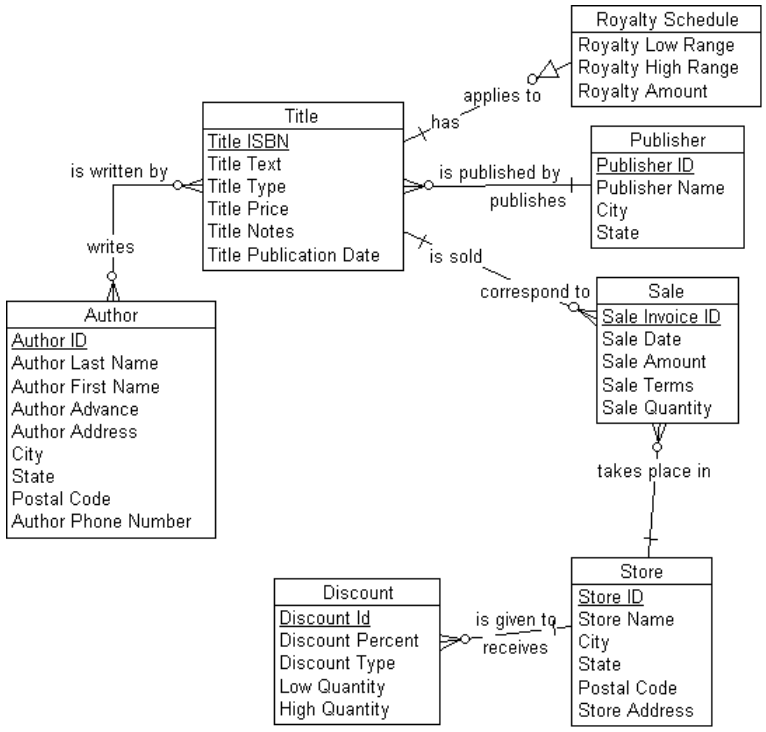
A file selection window appears.

**2 Go to the** *EXAMPLES* **directory.**
**Select the** *CDMBEFOR.CDM* **file.**
**Click** *OK***.**

DataArchitect displays the model in the CDM workspace.

**3 If you can only see part of the model, click the *Zoom All* tool in the toolbar that runs across the top of the window.**

This displays the whole model in the CDM workspace. The model may not look exactly like the one shown below.




**Adjust display scale**
You can choose the display scale most comfortable for your eyes, by selecting View➤Scale and choosing a scale.

# Define CDM preferences and options

**Where you are**
Chapter 3 How to Begin
Start DataArchitect
Use the tools in the tool palette
Open the tutorial CDM
➡ Define CDM preferences and options
Define CDM properties
Save the tutorial CDM

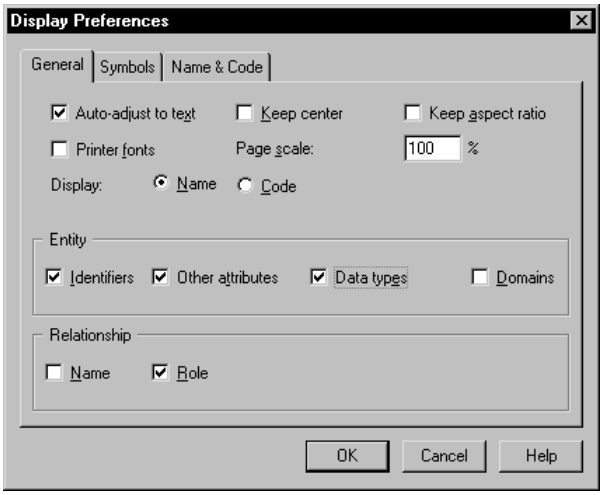
Before you begin working, you will define certain display preferences and model options for the CDM.

For a complete description of all CDM preferences and options, see *PowerDesigner DataArchitect User's Guide*.

**1 Select *File➤Display Preferences* from the menu bar.**

The Display Preferences dialog box appears.

**2 Select or clear checkboxes and radio buttons so that the dialog box looks like the one shown below.**



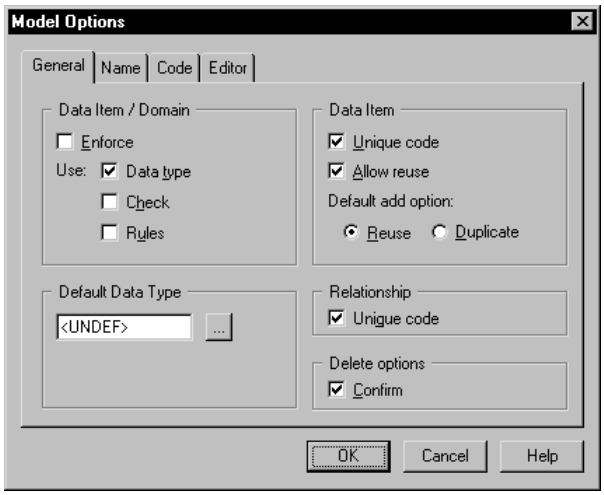

These preferences display object names, entity identifiers, entity attributes, and relationship roles. They adjust entities to fit their text.

**3 Click** *OK*.

**4 Select** *File➢Model Options* **from the menu bar.**

The Model Options dialog box appears.

**5 Select or clear checkboxes and radio buttons so that the dialog box looks like the one shown below.**



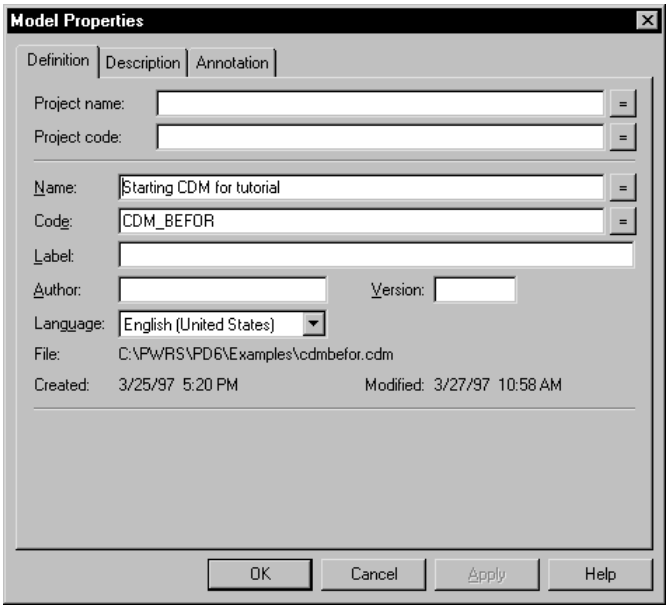

**6 Click** *OK*.

# Define CDM properties

**Where you are**
Chapter 3 How to Begin
Start DataArchitect
Use the tools in the tool palette
Open the tutorial CDM
Define CDM preferences and options
➡ Define CDM properties
Save the tutorial CDM

**1 Select *Dictionary➢Model Properties* from the menu bar.**

The Model property sheet appears.




**2 Type *Publications* in the Project Name box.**

This is the name of the project to which the CDM belongs.

**3 Click the = button at the end of the Project Code box.**

This sets the code equal to the name.

4 **Type *Tutorial* in the Name box.**

This is the name of the CDM.

5 **Click the = button at the end of the Code box.**

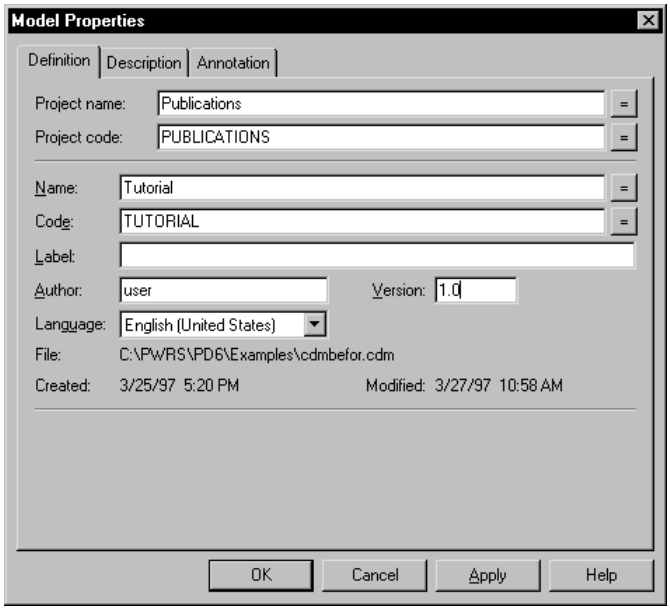
This sets the code equal to the name.

6 **Type your *name* in the Author box.**

This indicates that you are the owner of the CDM.

7 **Type *1.0* in the Version box.**

This indicates that this is the first version of the CDM.




8 **Click *OK*.**

# Save the tutorial CDM

**Where you are**
Chapter 3 How to Begin
Start DataArchitect
Use the tools in the tool palette
Open the tutorial CDM
Define CDM preferences and options
Define CDM properties
➡ Save the tutorial CDM

You will save the tutorial CDM in another file. This leaves the original tutorial CDM intact so you can use it again if you want to redo the exercises.

**1 Select** *File➤Save As*.

The File Save As dialog box appears.

**2 Type** *TUTORIAL.CDM* **in the File Name box.**

This is the name of the file in which you will work and save your modifications.

**3 Click** *OK*.

This saves your model in the TUTORIAL.CDM file.

**Save your work**
Save your work periodically while doing these exercises by selecting File➤Save.

CHAPTER 4

# Defining Business Rules, Domains, and Data Items

## About business rules

A **business rule** is a written expression of the way a business operates. When you create a business rule, you refer to the ideas set out in the business description.

There are four types of business rules: facts, definitions, formulas and validations:

| Type of business rule | Example |
|---|---|
| Fact | A publisher may publish one or more titles |
| Definition | An author is identified by a name and an address |
| Formula | The amount of royalties is a percentage of the sales which increases according to the amount of sales |
| Validation | The percentage of royalties paid to all authors of a book must add up to 100% of the royalties |

When you use business rules to document the CDM, they are not translated into executable code. You will see the ways in which you can implement business rules later, in the PDM part of the tutorial.

## About data items

A **data item** is an elementary piece of information. For example, this CDM contains data items for author last name and for book titles.

## About domains

A **domain** defines a standard data structure that you can apply to multiple data items. When you modify a domain you globally update the data items associated with the domain. This makes it easier to standardize data characteristics and modify your model consistently when you need to make changes.

## About check parameters

**Check parameters** specify more precisely what values to allow for a domain or a data item. In a CDM, check parameters define standard parameters, such as minimum, maximum, or accepted values.

What you will do

In this chapter you will:

- Create a new business rule
- Create a new domain
- Create a new data item
- Attach a data item to a domain
- Attach a check parameter to a domain

**How long will it take?**
About 20 minutes.

# Create a new business rule

**Where you are**
Chapter 4 Defining Business Rules, Data Items, and Domains
➧ Create a new business rule
Create a new domain
Create a new data item
Attach a data item to a domain
Attach check parameters to a domain

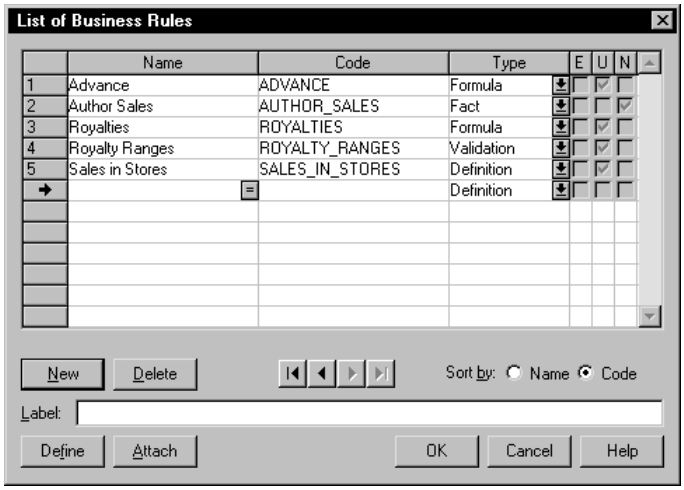
You will create a business rule that states how to attribute royalties to authors.

**1 Select** *Dictionary➢List of Business Rules*.

The List of Business Rules dialog box displays existing business rules.

**2 Click** *New*.

An arrow appears at the beginning of the first blank line.




**3 Type** *Author Percent* **in the Name column.**

This is the name of the business rule.

**4 Click the *Code* column.**
**Click the = button that appears in the Code column.**

This sets the code equal to the name, with lowercase characters replaced by uppercase, and the space replaced by an underscore.

> **Using the keyboard**
> You can also move to the next column in a list dialog box by pressing TAB. If the cursor is in the Name or Code column, you can press F4 to select the = button.

**5 Select *Validation* from the dropdown list in the Type column.**

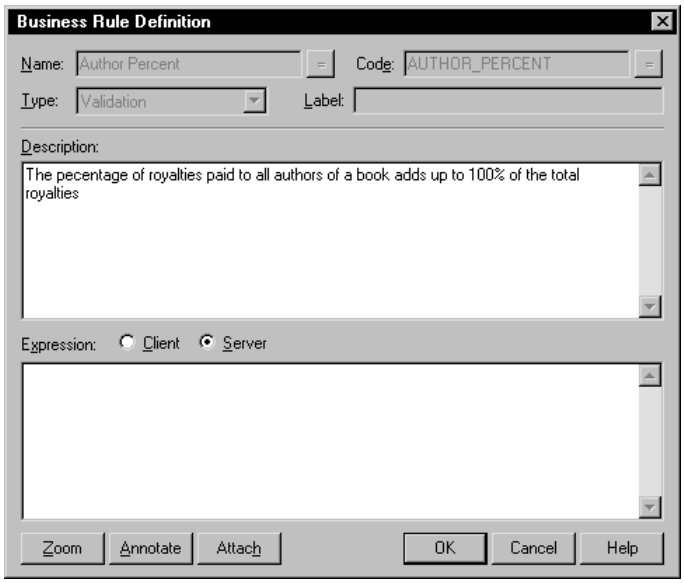

This defines the business rule as a validation rule.

**6 Click the *Define* button in the bottom part of the dialog box.**

The Business Rule Definition dialog box appears.

7 **In the Description box, type** *The percentage of royalties paid to all authors of a book adds up to 100% of the total royalties*.

This text explains the meaning of the business rule.




8 **Click** *OK* **in each of the dialog boxes.**

This validates creation of the business rule.

> **Business rules sorted alphabetically**
> When you close the list of business rules, the business rules are sorted alphabetically. The order of appearance of the business rules will therefore be different next time you open the list.

## What you learned

In this section, you learned how to:

- Use the statement of the business problem to create the description of the business rule
- Create a business rule
- Identify a business rule by a name, a code, and a type

# Create a new domain

**Where you are**
Chapter 4 Defining Business Rules, Data Items, and Domains
Create a new business rule
➡ Create a new domain
Create a new data item
Attach a data item to a domain
Attach check parameters to a domain

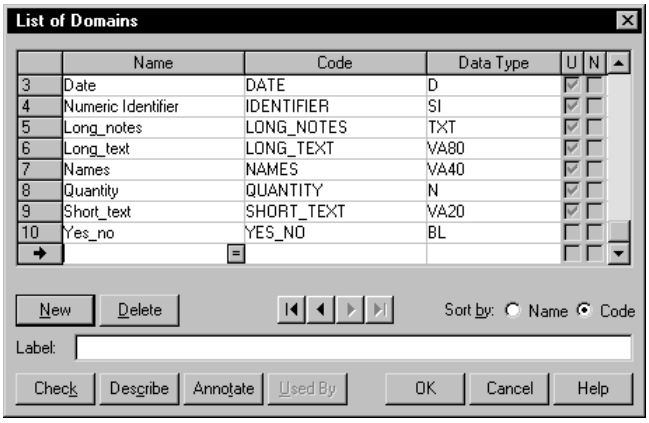
You will create two domains that will define a standardized data type for money amounts and for percentages in the model.

**1 Select *Dictionary➢List of Domains* from the menu bar.**

The List of Domains dialog box displays the existing domains.

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

An arrow appears at the beginning of the first blank line.




**3 Type *Amount* in the Name column.**

This is the name of the domain.

**4 Click the *Code* column.**
**Click the = button that appears in the Code column.**

This sets the code equal to the name, with lowercase characters replaced by uppercase.

**5 Click the *Data Type* column.**
**Click the ... button that appears in the Data Type column.**

The Standard Data Types dialog box appears. You use this dialog box to specify the form of the data affected by the domain.

> **Using the keyboard**
> If the cursor is in the Data Type column, you can open the Standard Data Types dialog box by pressing F4.

**6 Click the *Money* radio button.**

The domain now has a money data type. A money data type stores numbers with a fixed decimal point. Later, when you apply this domain to the data items that are used to store amounts of money, you will see that they inherit this data type.

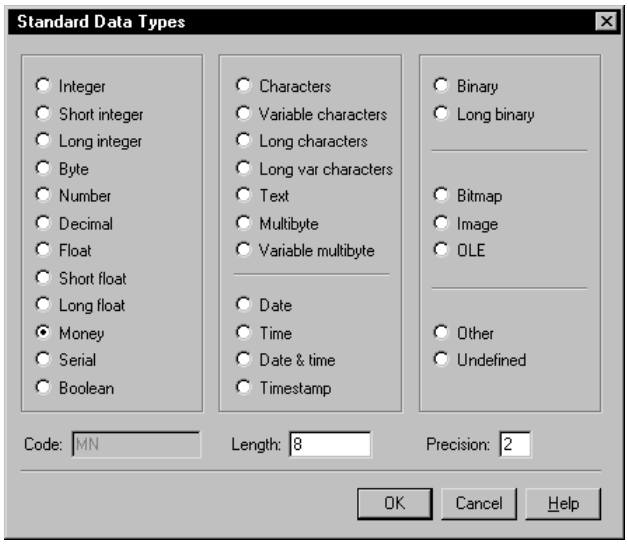
> **Data type codes**
> All data types have a one, two, or three letter code. When you select the Money radio button, MN appears in the Code box. This is the code for a money data type.

**7 Type *8* in the Length box.**

The maximum number of figures in a data item attached to this domain will be 8.

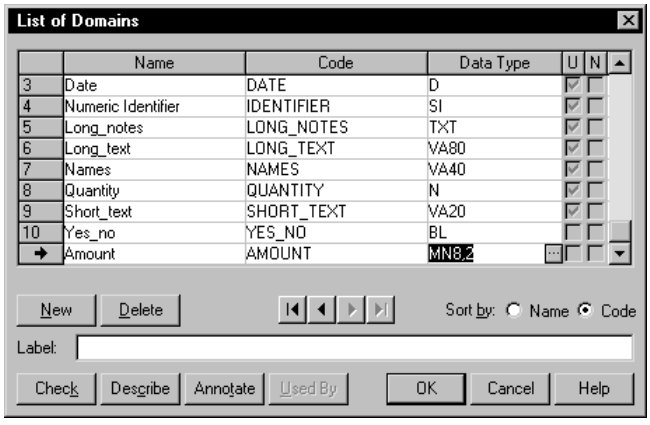
**8 Type *2* in the Precision box.**

A data item attached to this domain can take two positions after the decimal point.




**9 Click *OK*.**

You return to the list of domains. The value MN8,2 appears in the Data Type column. MN is the code for a money data type. Eight indicates that an amount of money can have 8 figures. Two indicates that the amount has a decimal precision of 2.



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

An arrow appears at the beginning of a blank line.

**11 Type *Percent* in the *Name* column.**

This is the name of the domain.

**12 Click the *Code* column.**
**Click the = button that appears in the Code column.**

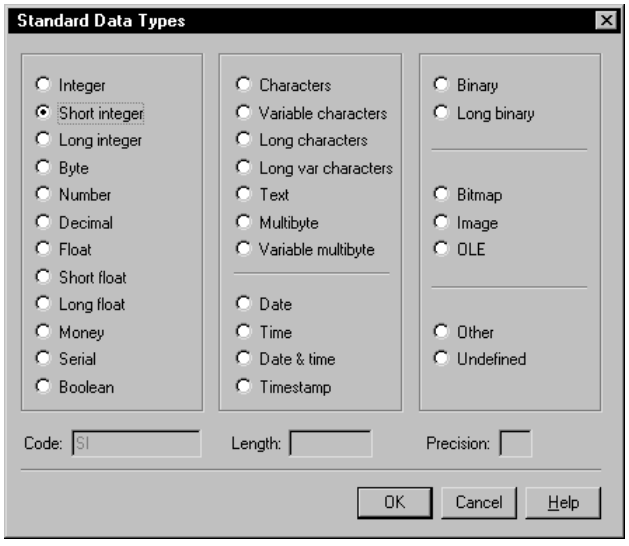
This sets the code equal to the name, with lowercase characters replaced by uppercase, and restricted characters (such as spaces) replaced by underscores.

**13 Click the *Data Type* column.**
**Click the ... button that appears in the Data Type column.**

The Standard Data Types dialog box appears.

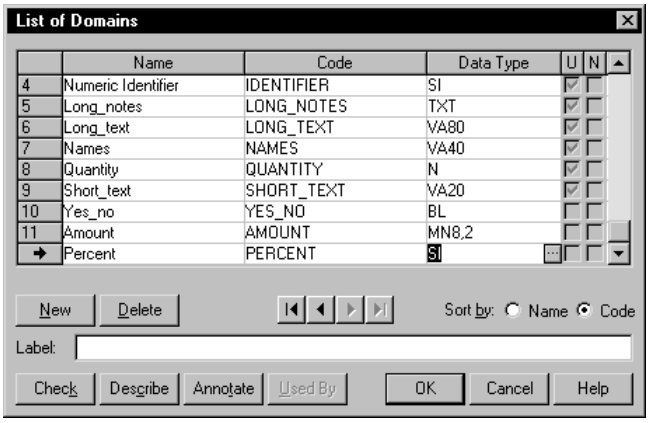
**14 Click the *Short Integer* radio button.**

The code SI indicates that the Percent domain has a short integer data type. The Length and Precision boxes are not available because you do not need to specify a length and precision for the short integer data type.



**15 Click** *OK*.

You return to the list of domains.




**Default data type**
When you do not define a data type for a domain, it receives the default data type. You can designate a default data type by selecting File➤Model Options.

**16 Click** *OK*.

This saves the definition and returns you to the model window.

## What you learned

In this section, you learned how to:

- Look for data items storing similar types of data, and create a domain to standardize the data characteristics of these data items
- Identify a domain by a name and a code
- Specify the data type, length, and precision of the domain

# Create a new data item

**Where you are**
Chapter 4 Defining Business Rules, Data Items, and Domains
Create a new business rule
Create a new domain
➧ Create a new data item
Attach a data item to a domain
Attach check parameters to a domain

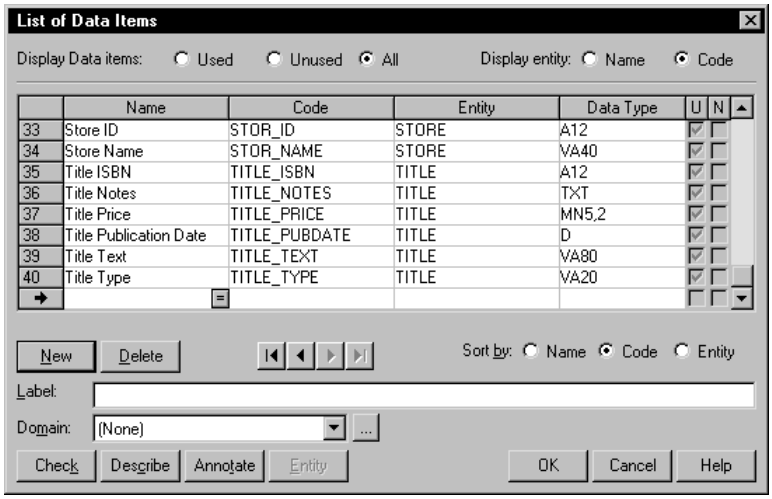
To manage books with multiple authors, you will create data items for the percentage of royalties for each author and for the order of an author's name in the list of authors.

**1 Select *Dictionary➢List of Data Items* from the menu bar.**

The List of Data Items displays existing data items.

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

An arrow appears at the beginning of the first blank line.



**3 Type *TitleAuthor Percent* in the Name column.**

This is the name of the data item.

**4 Type *TA_PERCENT* in the Code column.**

> **The Entity column**
> The Entity column indicates the entity to which each data item belongs. This column is empty because the new data items are not yet associated with entities.

**5 Select *Percent* from the Domain dropdown listbox in the bottom part of the dialog box.**

This applies the data type of the Percent domain to the new data item. In the Data Type column, SI indicates a short integer.

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

An arrow appears at the beginning of a blank line.

**7 Type *TitleAuthor Order* in the Name column.**

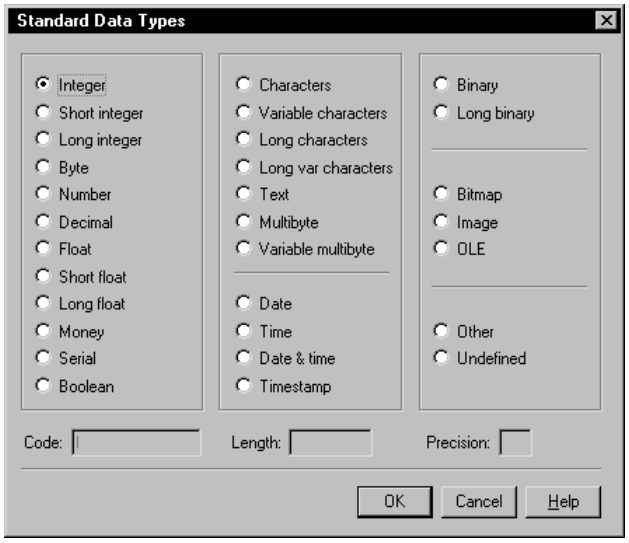
**8 Type *TA_ORDER* in the Code column.**

**9 Click the *Data Type* column.**
**Click the = button that appears in the column.**

The Standard Data Types dialog box appears.

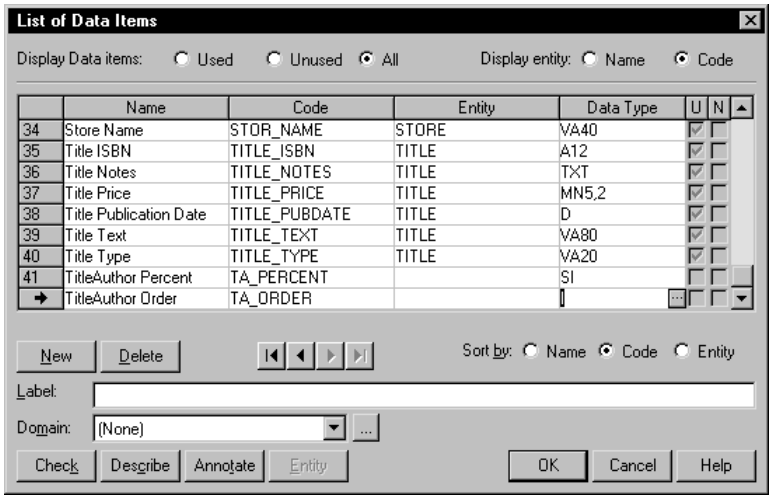
**10 Click the *Integer* radio button.**

This gives the TitleAuthor Order data item an Integer data type.




**11 Click *OK*.**

You return to the List of Data Items.



**12 Click** *OK*.

This saves the data item definitions and returns you to the model window.

## What you learned

In this section, you learned how to:

- Refer to the statement of the business problem to decide what additional information you need to be able to manage
- Create data items to reflect these needs
- Identify a data item by a name, a code, and a data type
- Select a data type by attaching a data item to a domain
- Select a data type for a specific data item

# Attach a data item to a domain

**Where you are**
Chapter 4 Defining Business Rules, Data Items, and Domains
Create a new business rule
Create a new domain
Create a new data item
➡ Attach a data item to a domain
Attach check parameters to a domain

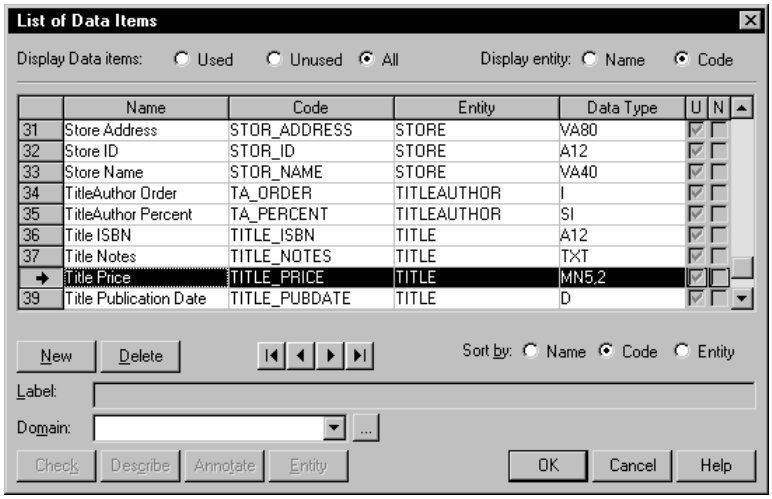
You will attach the AMOUNT domain to all data items that store amounts of money and the PERCENT domain to all data items that store percentages.

**1 Select** *Dictionary➢List of Data Items* **from the menu bar.**

The List of Data Items dialog box appears.

**2 Click the number of the data item** *Author Advance*.
**Press** CTRL **while you click the number of the data item** *Royalty Amount*.
**Press** CTRL **while you click the number of the data item** *Sale Amount*.
**Press** CTRL **while you click the number of the data item** *Title Price*.

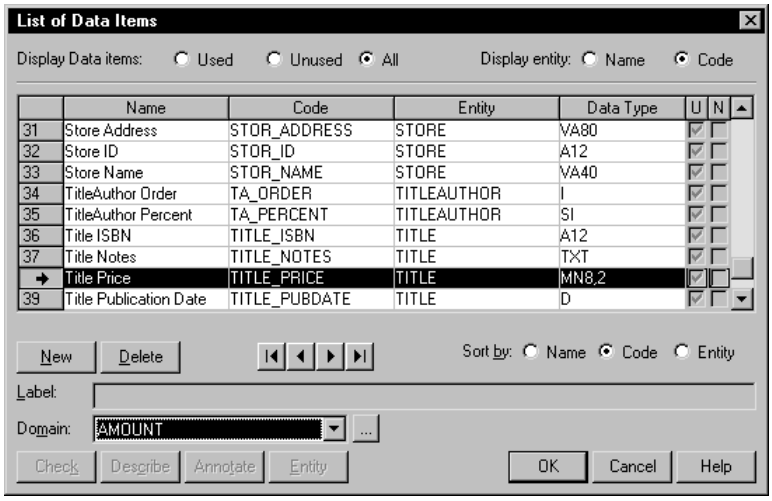
These data items are selected.



**3 Select *AMOUNT* from the Domain dropdown listbox in the bottom part of the dialog box.**

The data items inherit the data type of the domain. MN8,2 appears in the Data Type column. This code indicates a money data type with a length of 8 and a precision of 2.

The data items Author Advance, Royalty Amount, Sales Amount, and Title Price which before had different data types, now inherit a standardized data type from the Amount domain.




**4 Click the number of the data item *Discount Percent* and select *PERCENT* from the Domain dropdown listbox.**

**5 Click *OK*.**

This saves your changes and attaches the data items to the domains.

## What you learned

It is good practice to attach data items to domains. By doing so, when you modify a domain you can globally update the characteristics of the attached data items.

In this section, you learned how to:

- Ensure data consistency by attaching data items to domains

# Attach check parameters to a domain

**Where you are**
Chapter 4 Defining Business Rules, Data Items, and Domains
Create a new business rule
Create a new domain
Attach a data item to a domain
Create a new data item
➡ Attach check parameters to a domain

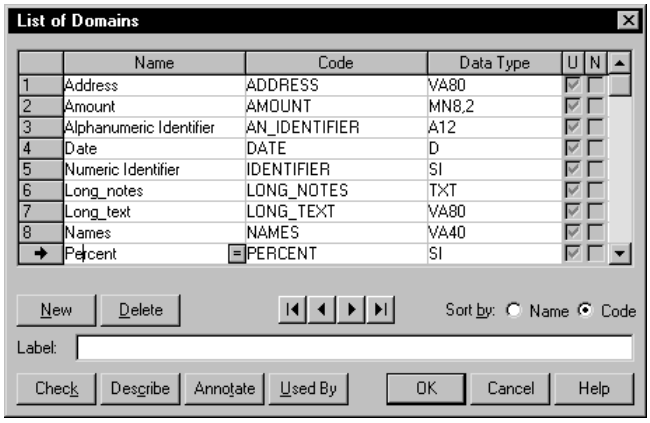
By attaching check parameters to the PERCENT domain, you will further specify the format of percentages for all data items attached to this domain.

**1 Select *Dictionary➢List of Domains* from the menu bar.**

The List of Domains dialog box appears.

**2 Click the *Percent* domain**

An arrow appears at the beginning of the line.




**3 Click the *Check* button.**

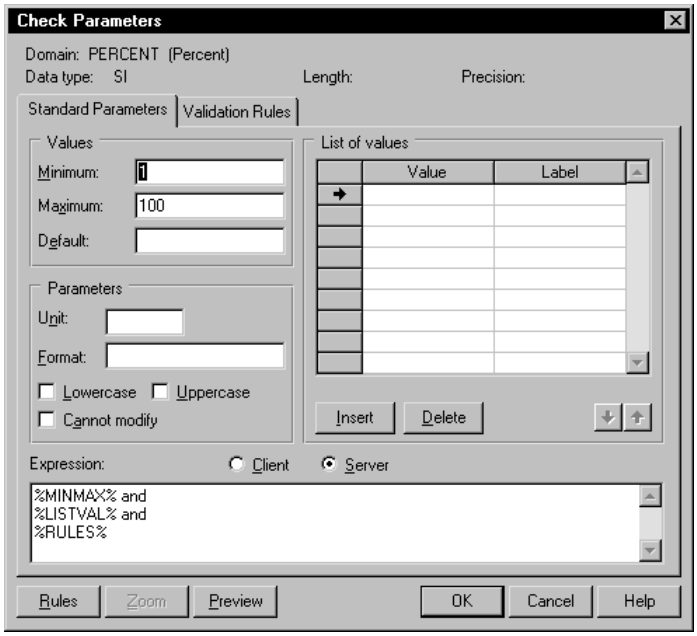
The Check Parameters dialog box appears.

**4 Type *1* in the Minimum box.**

This means that the minimum percentage of royalties is 1.

**5 Type *100* in the Maximum box.**

This means that the maximum percentage of royalties is 100.

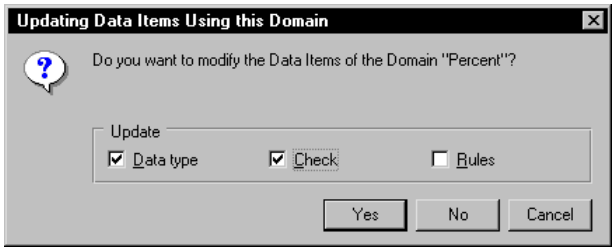



**6 Click *OK* in each of the dialog boxes.**

A message asks you if you want to update the data items attached to the domain.

**7 Select the *Check* checkbox.**

This choice indicates to update check parameters for data items attached to the domain.



**Domain enforcement**
You receive this confirmation box if you did not choose Enforce Domains as a model option. You define domain enforcement by selecting File➤Model Options.

8 **Click** *Yes*.

This updates the data type of the data items TitleAuthor Percent and Discount Percent.

You return to the CDM workspace.

## What you learned

In this section, you learned how to:

- Standardize the data structure of data items more precisely by using check parameters to define acceptable values
- Globally specify acceptable values for a number of data items by attaching check parameters to a domain
- Define a range of acceptable values by specifying a minimum and maximum acceptable value

CHAPTER 5

# Defining Entities

About entities

An **entity** is an object about which you want to store information. For example, in the tutorial model the AUTHOR entity groups information like author name and address.

Why create an entity

The business problem indicates which entities you need to create. For example, to identify and store pictures of authors, you will create a PICTURE entity that contains all information related to pictures.

Why create an associative entity

To respond to another business problem, you need to keep track of the percentage of royalties received by each author of each title.

One title may be written by many authors, and one author may have many titles to his or her credit. This is called a many-to-many relationship.

Because each author must be unique in the Author entity, and each Title must be unique in the Title entity, you will create an associative entity that has a unique occurrence for each title-author combination. You will then be able to attach a percentage to each unique case.

About entity attributes

An **entity attribute** is an elementary piece of information (data item) which you attach to an entity. For example, Last Name is an attribute of the AUTHOR entity because it provides information about an author. You will add a biography attribute to the AUTHOR entity.

About entity identifiers

An **entity identifier** is made up of one or more attributes unique to the entity, such that each value of the identifier corresponds to one, and only one, occurrence of the entity. For example, ISBN is the identifier of the TITLE entity because an ISBN uniquely identifies a title. You will assign a picture identification number as the identifier of the PICTURE entity.

## Attaching business rules to entities

You attach a business rule to an entity as a reminder of the conditions attached to the entity. You will attach the business rule you created in the preceding lesson to one of the entities.

In this chapter you will:

- Create an entity
- Create an associative entity
- Define entity attributes
- Designate an identifier
- Attach a business rule to an entity

**How long will it take?**
About 15 minutes

# Create a new entity

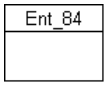
**Where you are**
Chapter 4 Defining Entities
➡ Create a new entity
Create an associative entity
Define entity attributes
Designate an identifier
Attach a business rule to an entity

You will create an entity that contains information related to pictures, an entity that associates titles to authors, and two entities that differentiate title categories: periodicals and non-periodicals.

**1 Click the *Entity* tool in the tool palette.**

**2 Click in the model workspace.**

An entity symbol appears at the click position.




At creation, an entity is named Ent_*n* where *n* is a number assigned to the entity in the order of creation of objects.

**3 Click the *Pointer* tool in the tool palette.**

**4 Double-click the symbol of the *entity* you just created.**

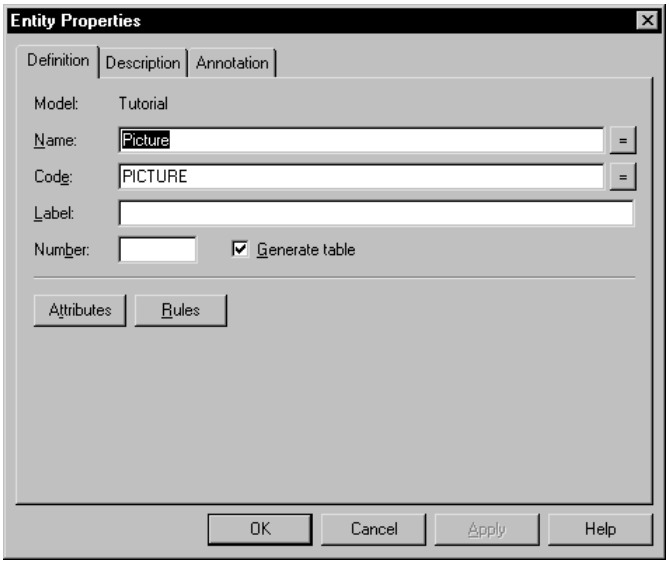
The Entity property sheet appears.

**5 Type *Picture* in the Name box.**

This is the name of the entity.

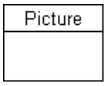
**6 Click the = button at the end of the Code box.**

This sets the code equal to the name.




**7 Click *OK*.**

The new entity displays the name Picture.




You created this entity by first creating its symbol, then identifying it from a property sheet. You can also create entities from the list of entities.

**8 Select *Dictionary➤List of Entities*.**

The List of Entities dialog box displays existing entities.

**9 Click *New*.**

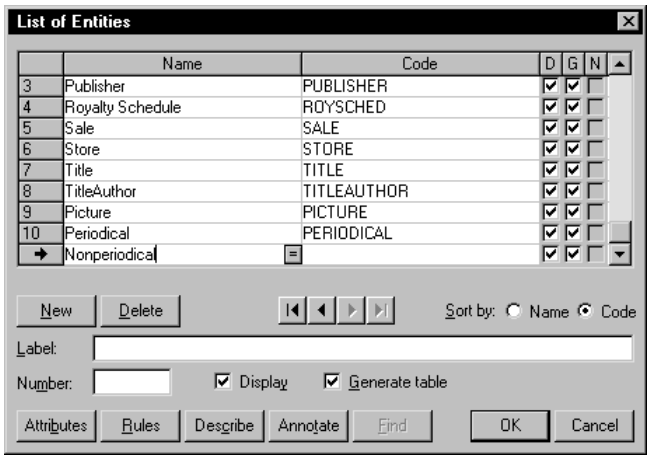
An arrow appears at the beginning of a blank line.

**10 Type *Periodical* in the Name column.**
**Click *New*.**

The code is automatically set equal to the name. An arrow appears at the beginning of a blank line.

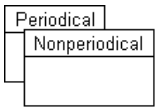
**11 Type *Nonperiodical* in the Name column.**

The new entities appear on the list. You do not need to type a code for this entity. By default, the code will be set equal to the name.




**12 Click *OK*.**

The new entities appear in the CDM.




**Moving entity symbols**
When you create entities from the list, they will not be arranged as shown above. You do not control the insert position of the symbols. You can move an entity symbol by selecting it and dragging it to a new position.

## What you learned

In this section, you learned how to:

- Use the statement of the business problem to decide what entities you need to create
- Create an entity by inserting an entity symbol and assigning a name and code from its property sheet
- Create an entity on the list of entities

# Create an associative entity

**Where you are**
Chapter 4 Defining Entities
Create a new entity
➡ Create an associative entity
Define entity attributes
Designate an identifier
Attach a business rule to an entity

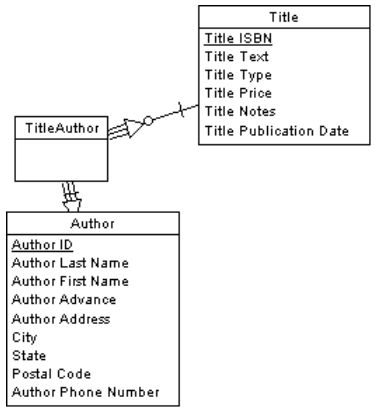
You will replace the existing relationship between AUTHOR and TITLE with an associative entity. This associative entity will have a unique occurrence for each title-author combination so that you can attach a royalty percentage to each unique case.

**1 Right-click the *relationship* that links Author and Title entities.**

A relationship context menu appears.

**2 Select *Change to Entity* from the context menu.**

A new entity is inserted between Author and Title entities.



# Define entity attributes

**Where you are**
Chapter 4 Defining Entities
Create a new entity
Create an associative entity
➡ Define entity attributes
Designate an identifier
Attach a business rule to an entity

You will define entity attributes for the entities TITLEAUTHOR, PICTURE, PERIODICALS, and NONPERIODICALS by attaching one or more data items to each entity.

You will create entity attributes by:

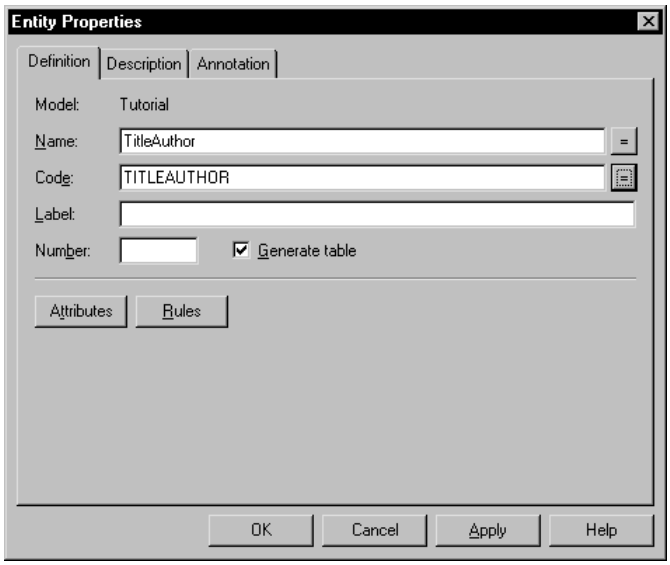
- Adding data items to an entity
- Creating a new entity attribute

## Add data items to an entity

You will attach existing data items to the entities TITLEAUTHOR, PICTURE, PERIODICAL, and NONPERIODICAL.

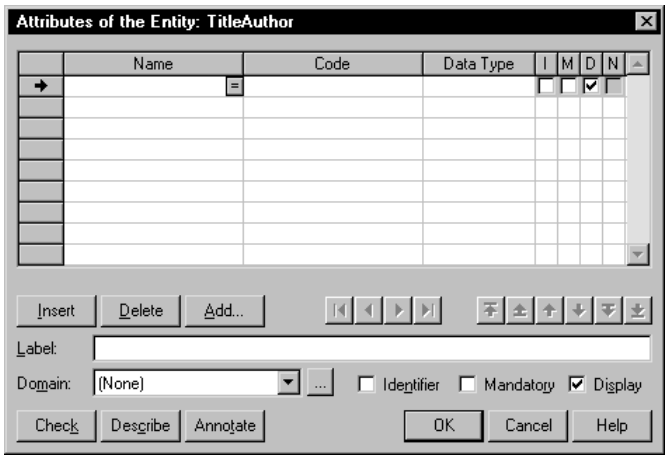
**1 Double-click the *TitleAuthor* entity.**

The Entity property sheet appears.



**2 Click the *Attributes* button.**

The Attributes of the Entity dialog box appears. The list is empty because the entity does not have any associated attributes.




**Go directly to the list of entity attributes**
You can go directly to the list of entity attributes by pressing CTRL while you double-click an entity.

**3 Click the *Add* button.**

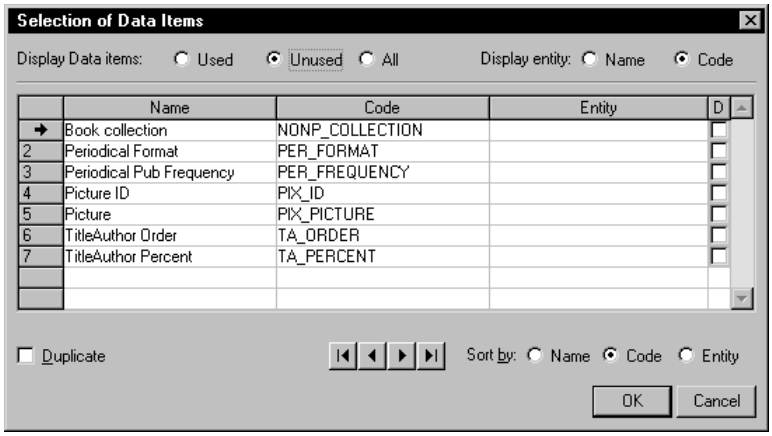
The Selection of Data Items dialog box appears. It displays the list of data items.

**4 Click the *Unused* radio button at the top of the dialog box.**

This displays only the data items that are not attached to an entity.

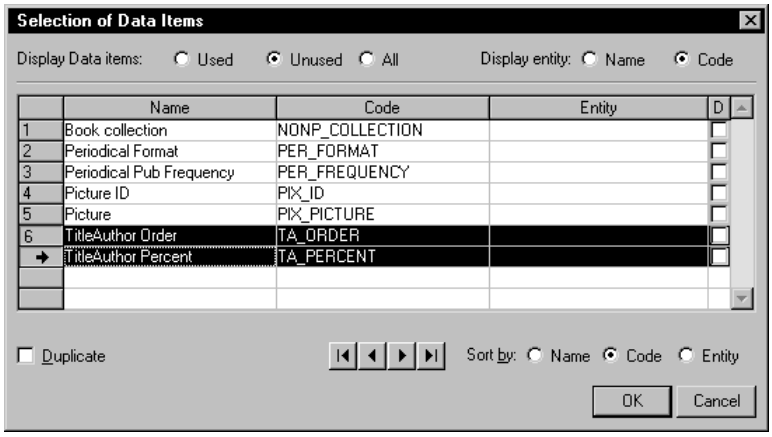
**5 Click the *Code* radio button at the bottom of the dialog box.**

This sorts the list of data items by code.



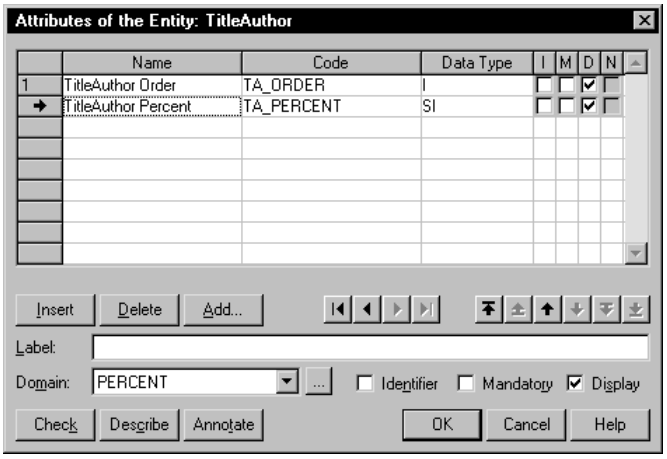

**6 Click the number at the beginning of the *TitleAuthor Order* line. Press CTRL while you click the number at the beginning of the *TitleAuthor Percent* line.**

The two data items are selected.



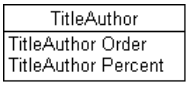
**7 Click *OK*.**

The data items appear in the list of attributes for the TitleAuthor entity.




**8 Click *OK* in each of the dialog boxes.**

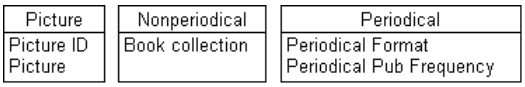
In the CDM workspace, the TITLEAUTHOR entity displays its attributes.




**9 Repeat steps 1-7 for the following entities:**

| For the entity | Select the data items |
|---|---|
| PICTURE | Picture ID<br>Picture |
| PERIODICAL | Periodical Format<br>Periodical Pub Frequency |
| NONPERIODICAL | Book Collection |

The CDM displays these entities with their attributes.

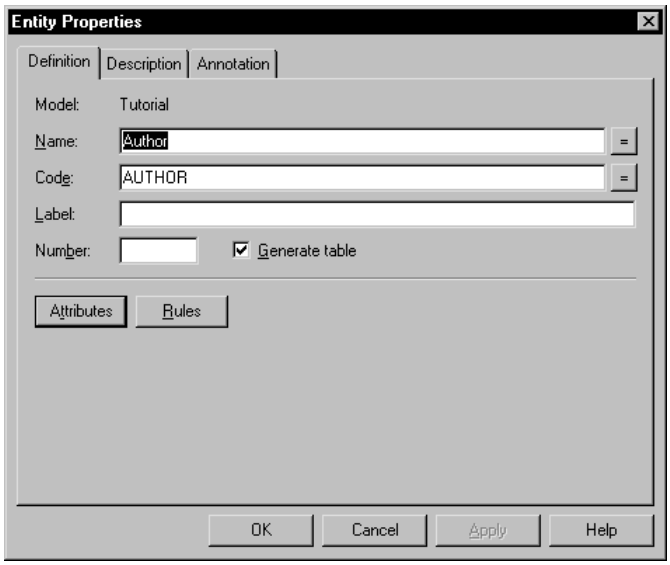


## Create a new entity attribute

You will create a new Biography attribute for the text of the author's biography.

**1 Double-click the *AUTHOR* entity symbol.**

The Entity property sheet appears.



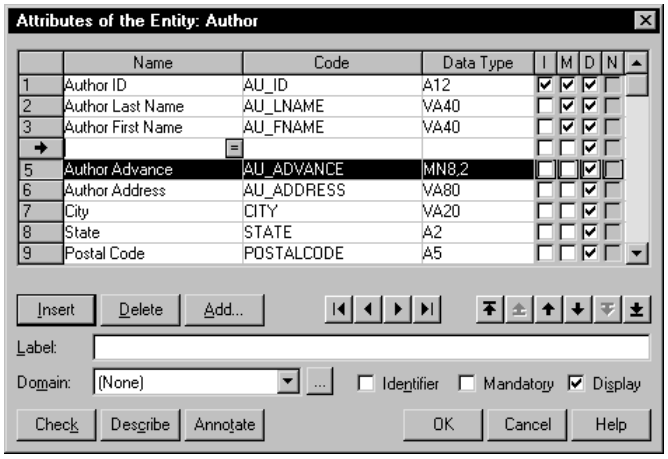

**2 Click the *Attributes* button.**

The Attributes of the Entity dialog box displays the attributes belonging to the Author entity.

**3 Select the *Author Advance* attribute.**

**4 Click the *Insert* button.**

A blank line is inserted above the Author Advance line.



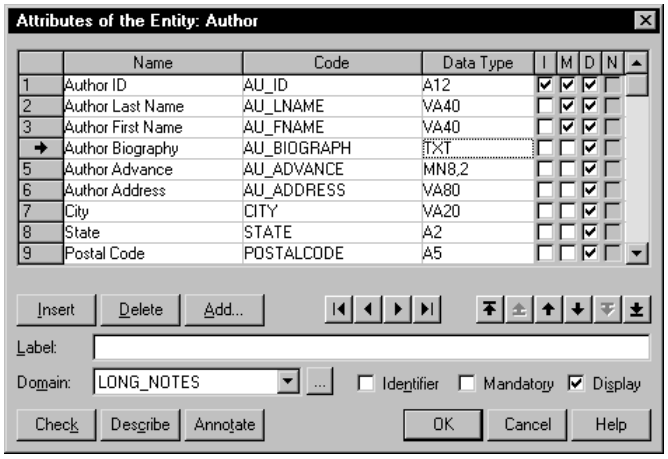

**5 Type *Author Biography* in the Name column.**

This is the name of the attribute.

**6 Type *AU_BIOGRAPH* in the Code column.**

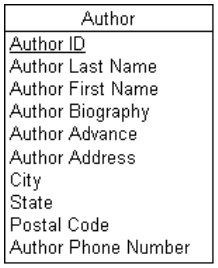
**7 Select *LONG_NOTES* from the Domain dropdown listbox in the bottom part of the dialog box.**

The text data type (TXT) appears in the Data Type column.



**8 Click *OK* in each of the dialog boxes.**

The Author entity displays its new attribute.




## What you learned

In this section, you learned how to:

- Add existing data items to an entity to create entity attributes
- Create a new entity attribute
- Identify an entity attribute by a name, a code, and a data type

# Designate an identifier

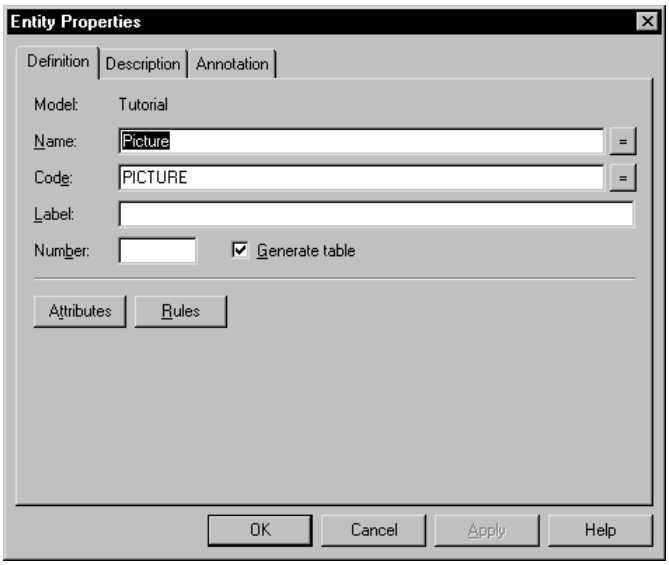
**Where you are**
Chapter 4 Defining Entities
Create a new entity
Create an associative entity
Define entity attributes
➡ Designate an identifier
Attach a business rule to an entity

An identifier is an entity attribute that uniquely identifies each occurrence of the entity.

You will designate Picture ID as the identifier of the PICTURE entity.

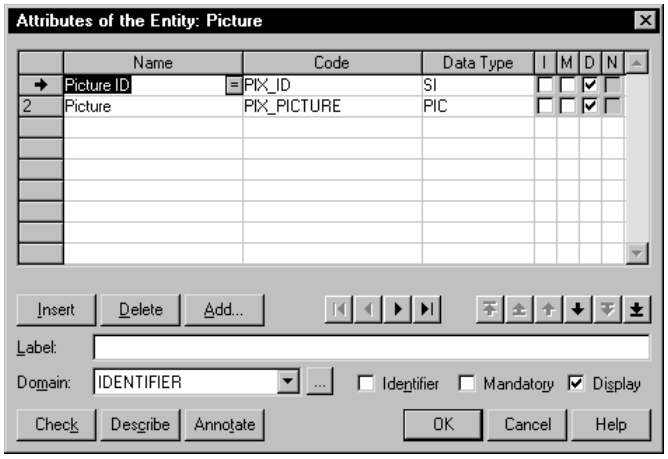
**1 Double-click the *PICTURE* entity symbol.**

The Entity property sheet appears.



**2 Click the *Attributes* button.**

The Attributes of the Entity dialog box appears.



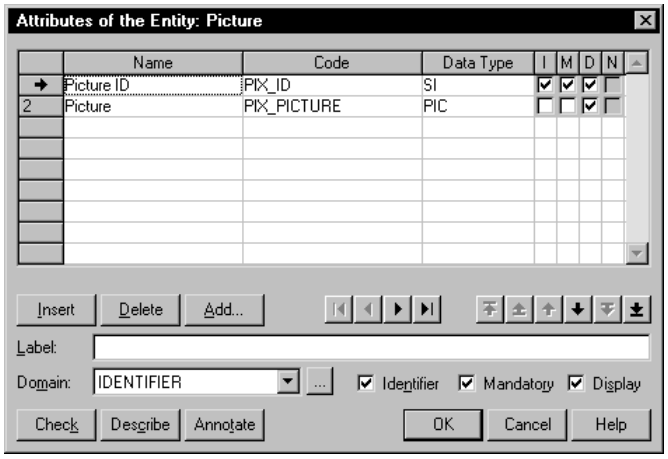

**3 Click the *Picture ID* attribute.**

An arrow appears at the beginning of the line.

**4 Select the *Identifier* checkbox in the bottom part of the dialog box.**

In the Picture ID line, checks in the I column and the M column indicate that this attribute is an identifier and a mandatory attribute, respectively.

> **Checkbox columns on the list of attributes**
> The list of attributes has four checkbox columns: I for identifier, M for mandatory, D for display, and N for notes. The N column contains a check if the attribute is annotated.

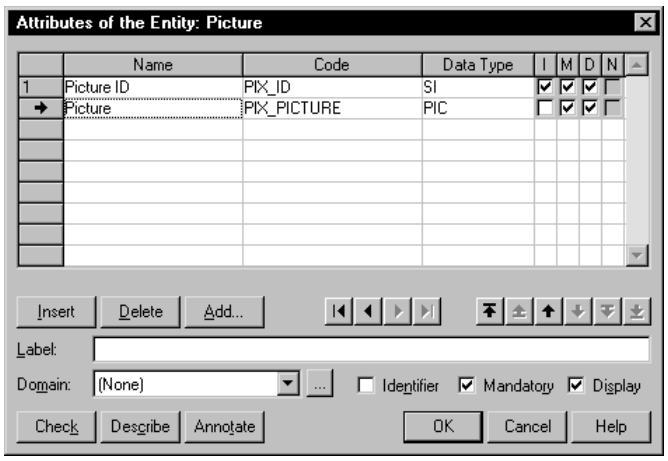



**5 Click the *Picture* attribute.**

An arrow appears at the beginning of the line.

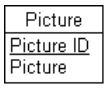
**6 Select the *Mandatory* checkbox in the bottom part of the dialog box.**

A check appears in the M column of the Picture attribute. This means the attribute is mandatory. In other words, each occurrence of the Picture entity must include a picture.




**7 Click *OK* in each of the dialog boxes.**

The Picture ID attribute is underlined in the PICTURE entity symbol to indicate that it is the identifier.




## What you learned

In this section, you learned how to:

- Designate an entity attribute as an identifier
- Make an entity attribute mandatory

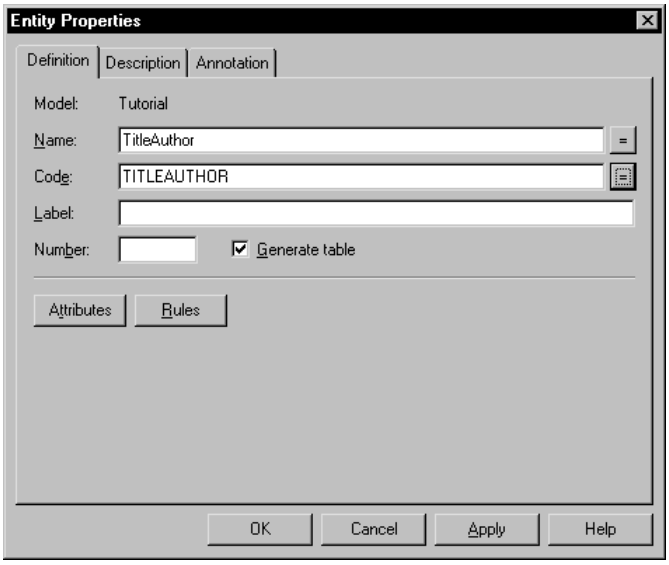
# Attach a business rule to an entity

**Where you are**
Chapter 4 Defining Entities
Create a new entity
Create an associative entity
Define entity attributes
Designate an identifier
➡ Attach a business rule to an entity

You will attach the Author Percent business rule to the TITLEAUTHOR entity to remind you of the role of this entity in the calculation of author royalties.

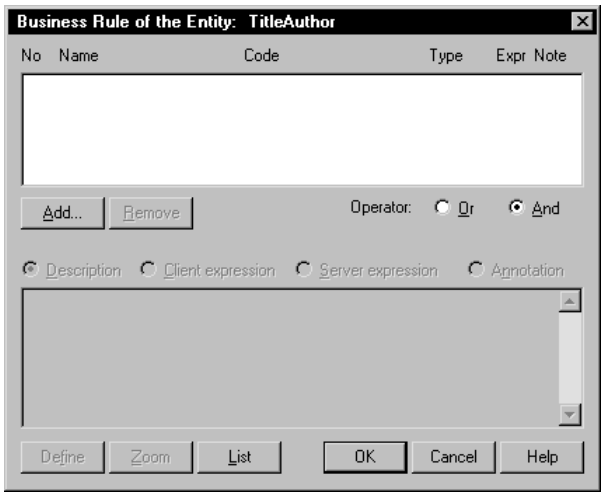
**1 Click the *Properties* tool in the tool palette.**
**Click the *TitleAuthor* entity symbol.**

The Entity property sheet appears.



**2 Click the *Rules* button.**

The Business Rule of the Entity dialog box lists the business rules associated with the entity. This list is empty.

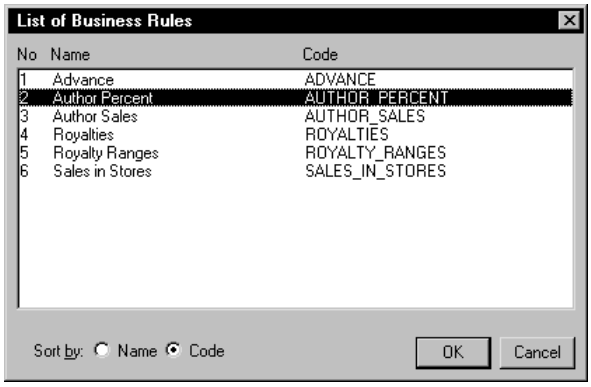



**3 Click the *Add* button.**

The List of Business Rules appears. This lists all the available business rules.

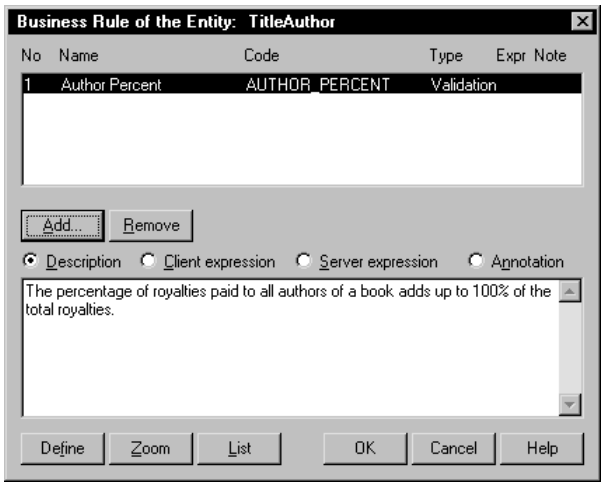
**4 Click *Author Percent*.**

This selects the Author Percent business rule.



**5 Click *OK*.**

You return to the Business Rule of the Entity dialog box. Author Percent appears in the list.




**6 Click *OK* in each of the dialog boxes.**

The business rule is attached to the entity. You return to the model window.

## What you learned

In this section, you learned how to:

- Attach a business rule to a particular entity in the CDM
- Use a business rule as a reminder of the role of an object in solving a business problem

CHAPTER 6

# Defining Relationships

## About relationships

A **relationship** is a named association between entities. It expresses the fact that two entities are somehow related. For example, in the tutorial model, a relationship links the entities PUBLISHER and TITLE because publishers publish books. An entity can have a relationship with itself, called a reflexive relationship.

## About cardinality

**Cardinality** indicates the maximum number of instances (one or many) of one entity in relation to another. To determine the cardinality of a relationship, ask the following question about each entity in the relationship, "Can more than one occurrence of this entity exist for one occurrence of the other entity?"

For example, in the direction author-to-picture, the cardinality is *many* because one author can be shown in several pictures. In the direction picture-to-author, the cardinality is *one* because a picture can only show one author. This type of relationship is called a one-to-many relationship.

## About mandatory relationships

You can define a relationship as mandatory from the point of view of one or both of its entities. To determine if a relationship is mandatory, ask the following question about each entity in the relationship, "Does an occurrence of this entity require an occurrence of the other entity?"

For example, the relationship for DISCOUNT to STORE is mandatory because all discounts must be associated with a specific store. On the other hand, the relationship from AUTHOR to PICTURE will be optional because an author may not have any pictures on file.

## About dependent relationships

In a dependent relationship one entity depends on another to uniquely identify it. For example, there are dependent relationships from TITLEAUTHOR to TITLE and to AUTHOR because a unique occurrence of TITLEAUTHOR is formed by the combination of one author and one title.

What you will do

In this chapter you will:

- Create a one-to-many relationship
- Define roles in an optional relationship

**How long will it take?**
About 10 minutes.

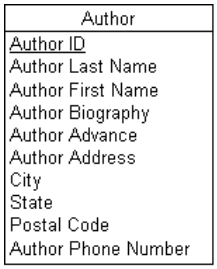
# Create a relationship

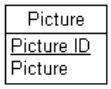
**Where you are**
Chapter 5 Defining Relationships
➡ Create a relationship
Define roles in an optional relationship

You will create a relationship between AUTHOR and PICTURE entities.

**1 Click the *Pointer tool* in the tool palette.**

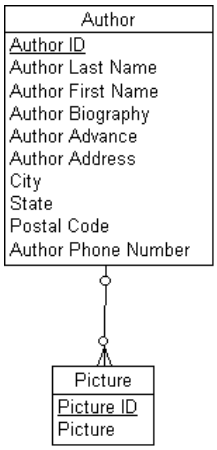
**2 Drag the *Picture* entity symbol under the AUTHOR entity symbol.**




**3 Click the *Relationship tool* in the tool palette.**

**4 Click and drag a *relationship* line from AUTHOR to PICTURE.**




The contact points of the relationship indicate that the cardinality of the relationship from AUTHOR to PICTURE is one-to-many, as follows:

- A single contact point on AUTHOR indicates that there is only one author for each picture
- Three contact points (a crow's foot) on PICTURE indicates that the same author can have more than one picture

The circles before both termination points indicate that both sides of the relationship are optional.

> **Relationship properties**
> Relationships that you create using the relationship tool are one-to-many and optional. You can change these and other properties from the relationship property sheet.

## What you learned

In this section, you learned how to:

- Translate the links between objects in the information system into relationships between entities
- Ask appropriate questions to help you decide the cardinality of a relationship

# Define roles in an optional relationship

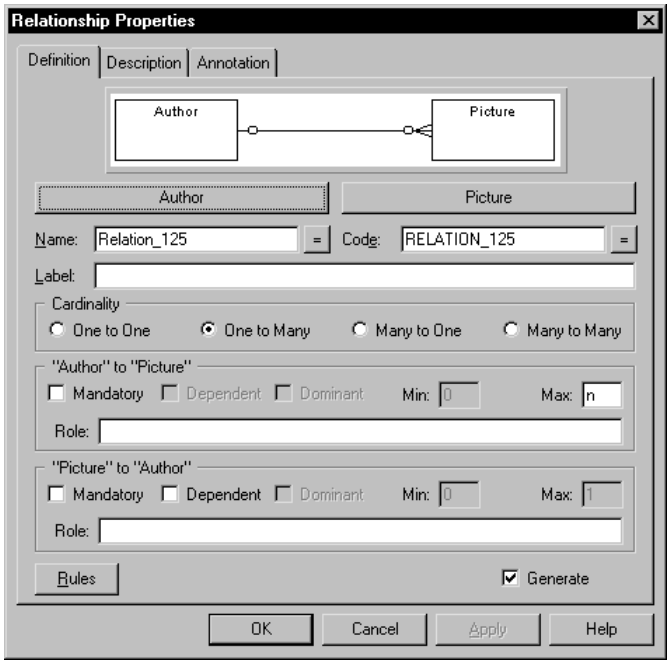
**Where you are**
Chapter 5 Defining Relationships
Create a relationship
➡ Define roles in an optional relationship

You will define an optional relationship between AUTHOR to PICTURE. An author does not have to have a picture. A picture does not have to be of an author.

**1 Click the *Pointer* tool in the tool palette.**

**2 Double-click the *relationship* line between AUTHOR and PICTURE.**

The Relationship property sheet appears:




**3 Type *Author Picture* in the Name box.**

This is the name of the relationship.

**4 Click the = button at the end of the Code box.**

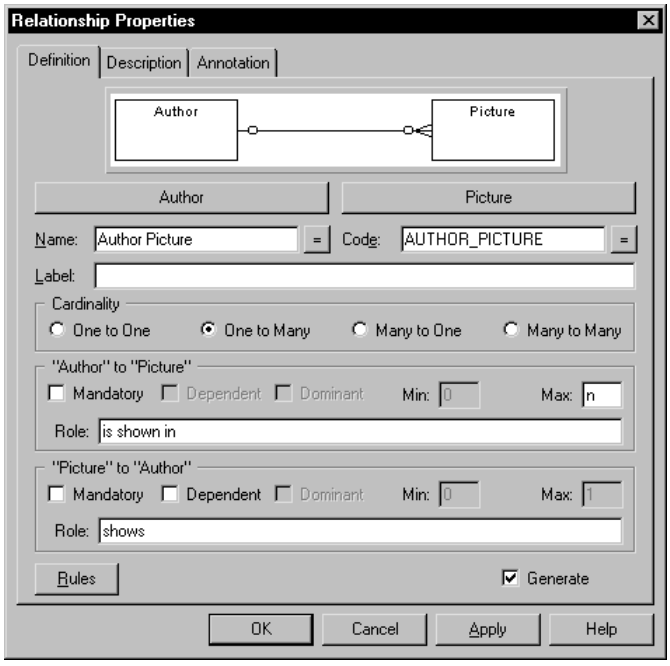
This sets the code equal to the name.

**5 Type *is shown in* in the Role box in the "Author" to "Picture" groupbox.**

This label indicates that an author *is shown in* a picture.

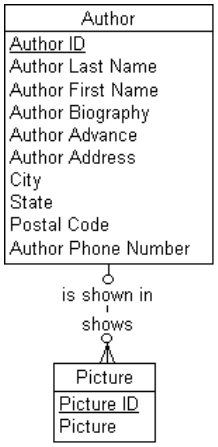
**6 Type *shows* in the Role box in the "Picture" to "Author" groupbox.**

This label indicates that a picture *shows* an author.



**7 Click** *OK*.

The relationship appears in the model.




**Display relationship roles**
You can display roles by selecting File➤Display Preferences and selecting the Role checkbox in the Relationship groupbox.

CHAPTER 7

# Creating a Submodel

Submodels help you manage large models. The global model contains all objects. The submodel contains a selection of objects from the global model.

You will create a submodel containing the entities TITLE, PERIODICAL, and NONPERIODICAL.

In this chapter you will:

- Create a submodel
- Add objects to the submodel

**How long will it take?**
About 5 minutes.

# Create a submodel

**Where you are**
Chapter 7 Creating a submodel
➡ Create a submodel
Add objects to the submodel

You will start building the submodel with the TITLE entity.

**1 Click the *TITLE* entity symbol to select it.**

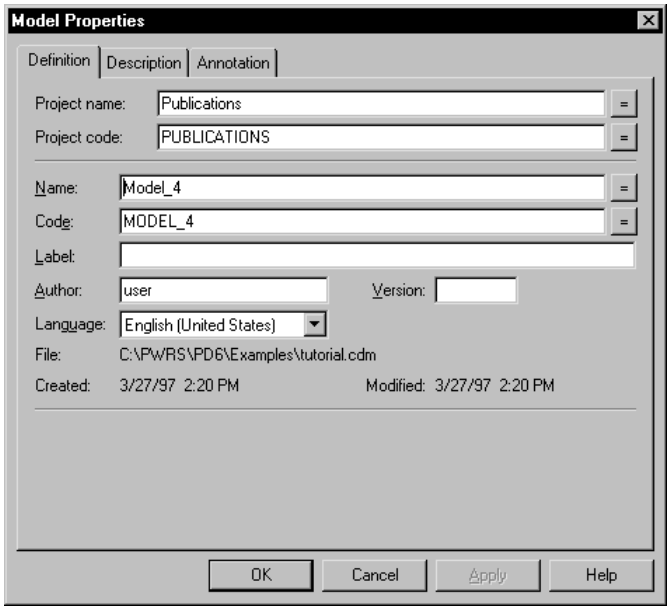
**2 Select *Dictionary*➢*Submodel*➢*New* from the menu bar.**

A new model window appears containing the TITLE entity.

**Create an empty submodel**
You can create an empty submodel by selecting Submodel➢New without selecting an object.

**3 Select** *Dictionary➤Model Properties* **from the menu bar.**

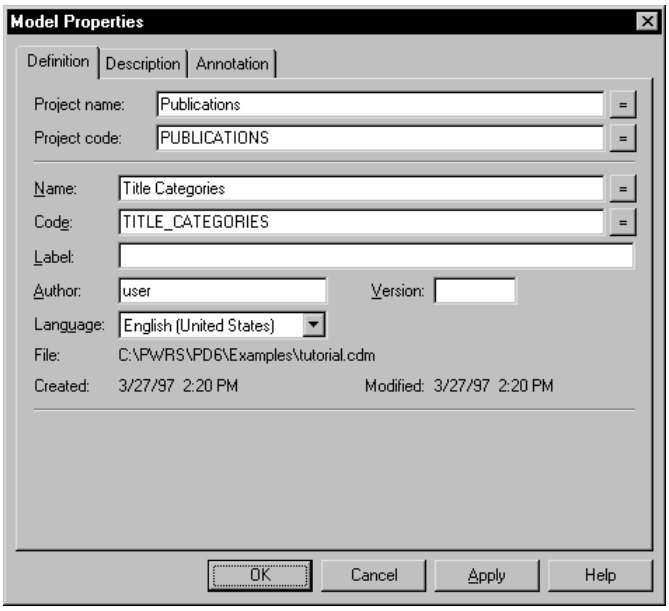
The submodel property sheet appears. It has the same project name and code as the global model.




**4 Type** *Title Categories* **in the Name box.**

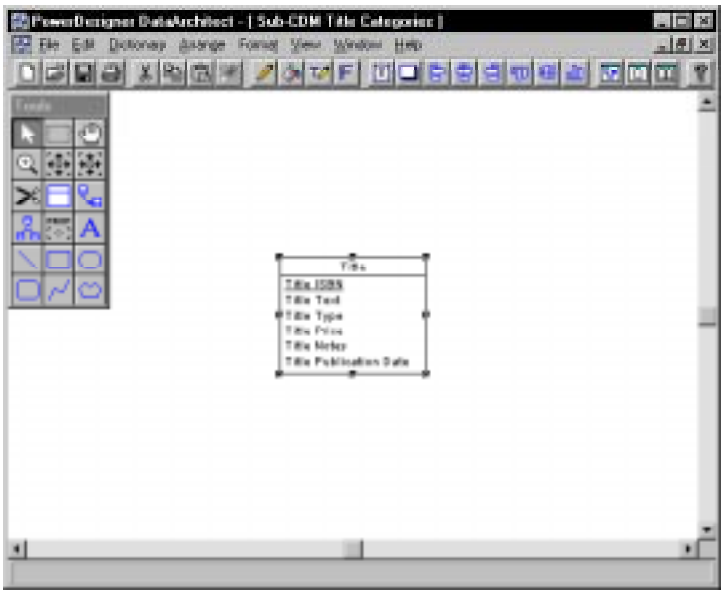
You overwrite the name Model_*n* name assigned by default.

**5 Click the = button at the end of the Code box.**




**6 Click *OK*.**

You return to the submodel CDM workspace.



What you learned

In this section, you learned how to:

- Create a submodel starting with one of the objects in the global model
- Identify a submodel by a name and a code

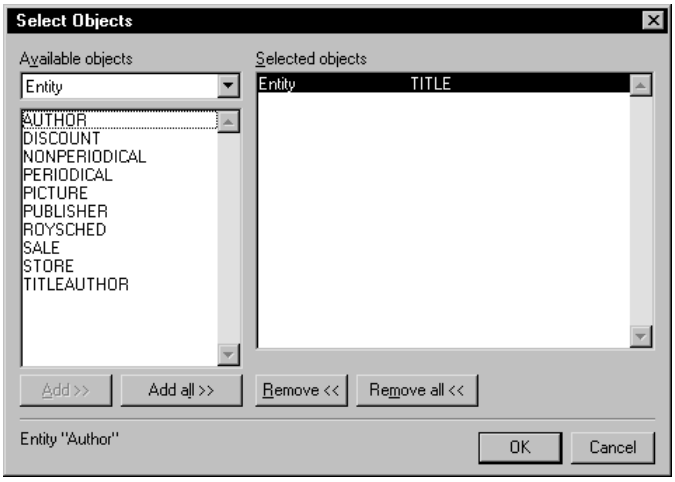
# Add objects to the submodel

**Where you are**
Chapter 7 Creating a submodel
Create a submodel
➡ Add objects to the submodel

You will add the PERIODICAL and NONPERIODICAL entities to the submodel.

**1 Select** *Dictionary➢Submodel➢Add/Remove Objects***.**

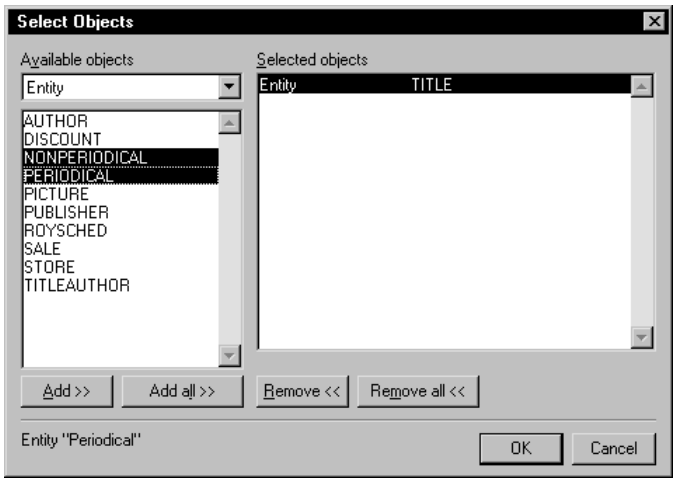
The Select Objects dialog box appears.




The dialog box displays all the entities in the global model. The TITLE entity appears in the list of selected objects because it is already in the submodel.

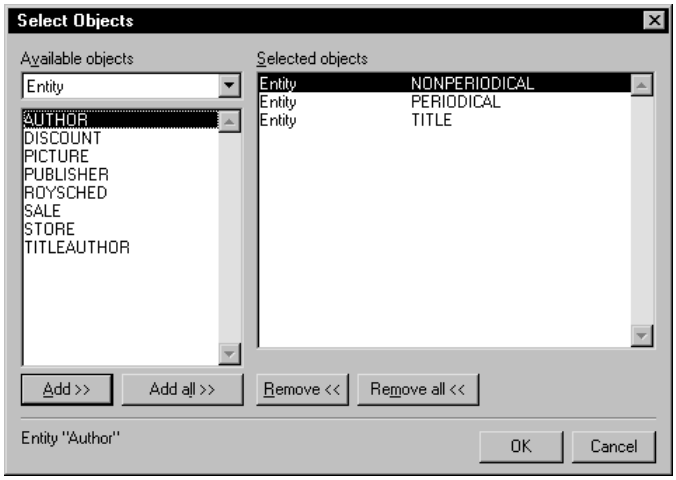
**2 Select the** *NONPERIODICAL* **entity.**

**3 Press the SHIFT key while you select the *PERIODICAL* entity.**




**4 Click the *Add* button.**

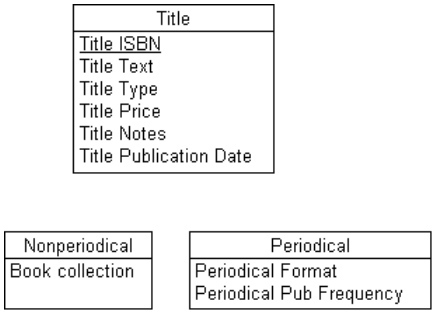
PERIODICAL and NONPERIODICAL appear in the Selected objects list.



**5 Click *OK*.**

Three entities appear in the submodel window.

**6 Drag and drop the *PERIODICAL* and *NONPERIODICAL* entity symbols to position them under TITLE.**




## What you learned

In this section, you learned how to:

- Select the objects you want to include in a submodel
- Add the objects to the submodel

CHAPTER 8

# Defining Inheritance

Inheritance allows you to define an entity as a special case of a more general entity. For example, you will define periodicals and non-periodicals as special cases of titles.

### About parent entities

The general entity is known as the **parent** (or supertype) entity, and contains all of the common characteristics. In this tutorial, you will use TITLE as the parent entity.

### About child entities

The special case entity is known as the **child** (or subtype) entity, and contains all of the particular characteristics. In this tutorial, you will create two child entities, PERIODICAL and NONPERIODICAL, because they are special cases of titles with specific attributes.

### Making children mutually exclusive

You can make an inheritance mutually exclusive. When an inheritance is mutually exclusive it means that there cannot be an occurrence of both child entities at the same time.

For example, a title can be a periodical or a non-periodical. Because it cannot be both, it is a mutually exclusive inheritance.

### What you will do

In this chapter you will:

- Create an inheritance link
- Define inheritance properties
- Add the submodel graphics to the global model

**How long will it take?**
About 10 minutes.

# Create an inheritance link

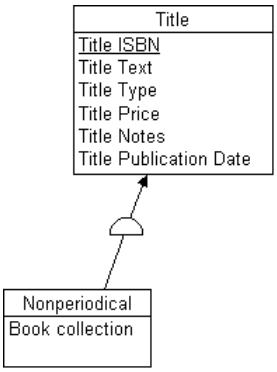
**Where you are**
Chapter 7 Defining Inheritance
➡ Create an inheritance link
Define inheritance properties
Add the submodel graphics to the global model

You will define an inheritance from the TITLE entity to the PERIODICAL and NONPERIODICAL entities.

**1 Select the *Inheritance* tool in the tool palette.**

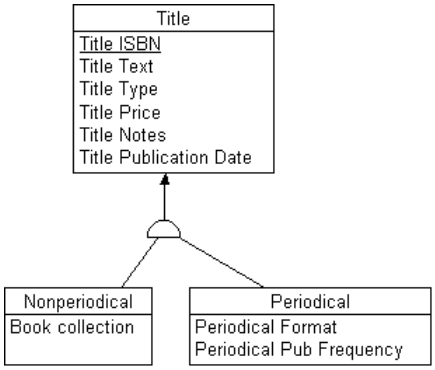
**2 Drag an *inheritance* link from TITLE to NONPERIODICAL.**

An inheritance link appears between these entities. The link has a semicircle in the middle, and an arrow that points to TITLE, which is the parent entity. NONPERIODICAL is the child entity.



**3 Click the *semicircle* and drag an *inheritance* link to the PERIODICAL entity.**

The inheritance symbol changes accordingly.




**Position inheritance symbol**
To arrange the position of the inheritance symbol, select the semicircle then select Arrange➤Disposition➤Arrange Symbols. The inheritance symbol moves to the center of gravity of the symbols it connects.

## What you learned

In this section, you learned how to:

- Identify entities which are special cases of other entities
- Create an inheritance link between entities

# Define inheritance properties

**Where you are**
Chapter 7 Defining Inheritance
Create an inheritance link
➡ Define inheritance properties
Add the submodel graphics to the global model

You will assign a name to the inheritance, make it mutually exclusive, and define its generation mode.

**1 Click the *Pointer* tool in the tool palette.**

**2 Double-click the *semicircle* in the middle of the inheritance link.**

The Inheritance property sheet appears.

**3 Type *Title inherit* in the Name box.**

This is the name of the inheritance.

**4 Click the  button at the end of the Code box.**

This sets the code equal to the name.

**5 Select the *Mutually exclusive* checkbox.**

This indicates that the inheritance is mutually exclusive because a title is either a periodical or a non-periodical, never both.

**6 Clear the *Generate children* checkbox in the Generation Mode groupbox.**

The choices you make in the Generation Mode groupbox indicate how the inheritance will be generated in the Physical Data Model (PDM). Selecting only Generate Parent means only one table will be generated for all titles. In this case, you will need a specifying attribute to differentiate occurences of each child.

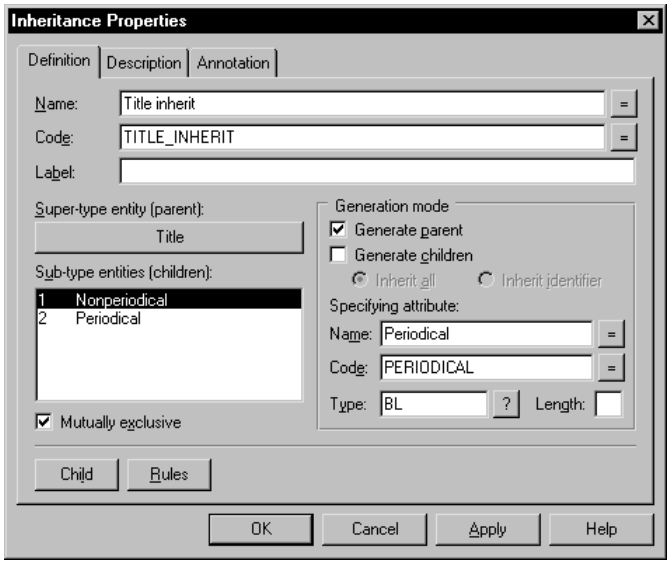
**7 Type *Periodical* in the Name box in the Generation Mode groupbox.**

Periodical is the name of the specifying attribute. This specifying attribute will generate a column named Periodical in the TITLE table that will result from generation. This column will indicate whether a title is a periodical or not.

**8 Click the = button at the end of the Code box in the Generation Mode groupbox.**

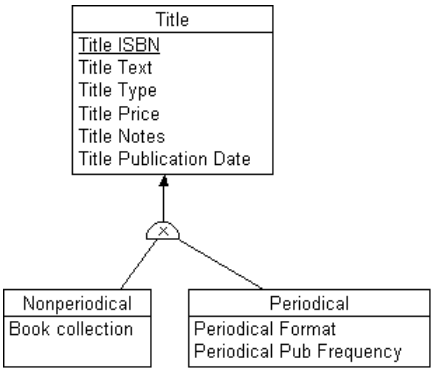
**9 Type *BL* in the Type box in the Generation Mode groupbox.**

You define a Boolean (BL) data type for the specifying attribute because there are only two possible choices: periodical or not.



**10 Click** *OK*.

A cross appears in the semicircle to indicate that the inheritance is mutually exclusive.




## What you learned

In this section, you learned how to:

- Identify an inheritance by a name
- Express the fact that only one child entity can exist at a time by making the inheritance mutually exclusive
- Select a generation mode to indicate what tables to generate for the inheritance
- Define a specifying attribute to identify subtypes in the same table

# Close the submodel and update graphics in the global model

**Where you are**
Chapter 7 Defining Inheritance
Create an inheritance link
Define inheritance properties
➡ Close the submodel and update graphics in the global model

When you create an object in a submodel, the definition of the object is automatically included in the global model. On the other hand, the graphic symbol of the object does not automatically appear in the global model. You have to explicitly request to update the global model graphics.

You created an inheritance in the submodel Title Categories. In order to display the inheritance link in the model, you will update graphics in the global model.

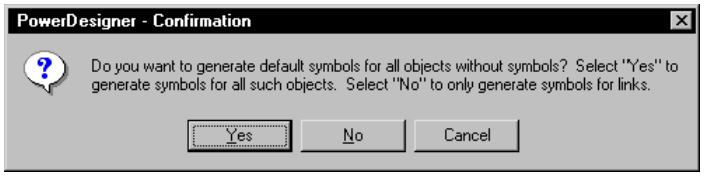
**If you created the inheritance in the global model**
If you created the inheritance in the global model window, you can skip the procedure below. You create an inheritance in exactly the same manner from the global model window as you do from the submodel window.

1 **From the submodel window, select** *File ➢Close*.

This closes the submodel and returns you to the window containing the global model.

2 **Select** *Dictionary ➢Submodel ➢Update Graphics* **from the menu bar in the global model window.**

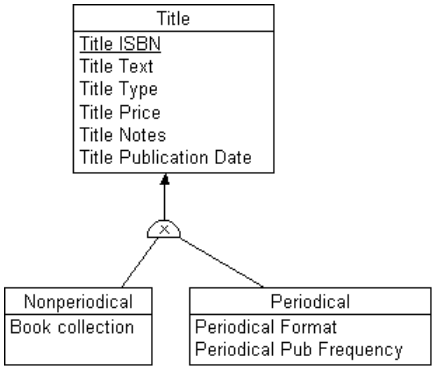
A Confirmation message box appears.



**3 Click** *Yes***.**

The inheritance link appears in the global model.

**4 Drag and drop the inheritance link and entity** *symbols* **to position them under the TITLE entity symbol.**




## What you learned

In this section, you learned how to:

- Update the graphic display so that the global model displays objects created in a submodel

CHAPTER 9

# Organizing the Display

When you finish creating the CDM, you can organize the model using the graphic display options.

In this chapter you will:

- Add a title box
- Change the color of the title box
- Change the color of the window
- Arrange the symbols
- Straighten relationship lines
- Center the model on the page
- Print the model

**How long will it take?**
About 15 minutes.

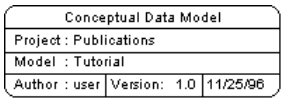
# Add a title box

**Where you are**
Chapter 8 Organizing the Display
➡ Add a title box
Change the color of the title box
Change the color of the window
Arrange the symbols
Straighten relationship lines
Center the model on the page
Print the model

You will add a title box to the model. The title box contains the essential information about the model.

**1 Select *Edit➢Add Title* from the menu bar.**

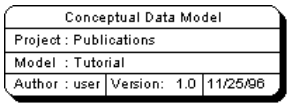
A title box appears in the middle of the model.




**2 Drag the *title box* to the top of the model.**

**3 Select *Format➢Shadow*.**

A shadow appears behind the title box. You can apply shadow to any symbol in the model.




## What you learned

In this section, you learned how to:

- Display the model properties in a title box
- Apply shadow to a title box

# Change the color of the title box

**Where you are**
Chapter 8 Organizing the Display
Add a title box
➡ Change the color of the title box
Change the color of the window
Arrange the symbols
Straighten relationship lines
Center the model on the page
Print the model

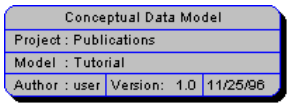
**1** **Select the** *title box*.

**2** **Select** *Format*➢*Background Color*.

**3** **Select a** *color* **in the palette.**

**4** **Click** *OK*.

The title box background changes to the selected color.




**Set default colors**
You can set default colors for symbols in the model, by selecting File➢Display Preferences and indicating color options on the Symbols page.

# Change the color of the window

**Where you are**
Chapter 8 Organizing the Display
Add a title box
Change the color of the title box
➡ Change the color of the window
Arrange the symbols
Straighten relationship lines
Center the model on the page
Print the model

By changing the window color, you can change the background color of the model.

**1 Select** *Window*➤*Window Color*.

**2 Select a** *color* **in the palette.**

**3 Click** *OK*.

The model background changes to the selected color.

**Window color applies to all models**
Window color is saved in the Windows registry or PD6.INI file. After you choose a color, it applies to all models that you open.

# Arrange the symbols

**Where you are**
Chapter 8 Organizing the Display
Add a title box
Change the color of the title box
Change the color of the window
➡ Arrange the symbols
Straighten relationship lines
Center the model on the page
Print the model

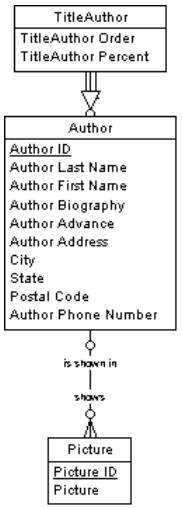
You will align the entity symbols on the left of the model.

**1 Using the *Pointer tool*, draw a rectangle encompassing the TITLEAUTHOR, AUTHOR and PICTURE entity symbols.**

**Selecting several symbols at once**
You can also select more than one symbol at a time by pressing the SHIFT key whil you click each symbol in turn.

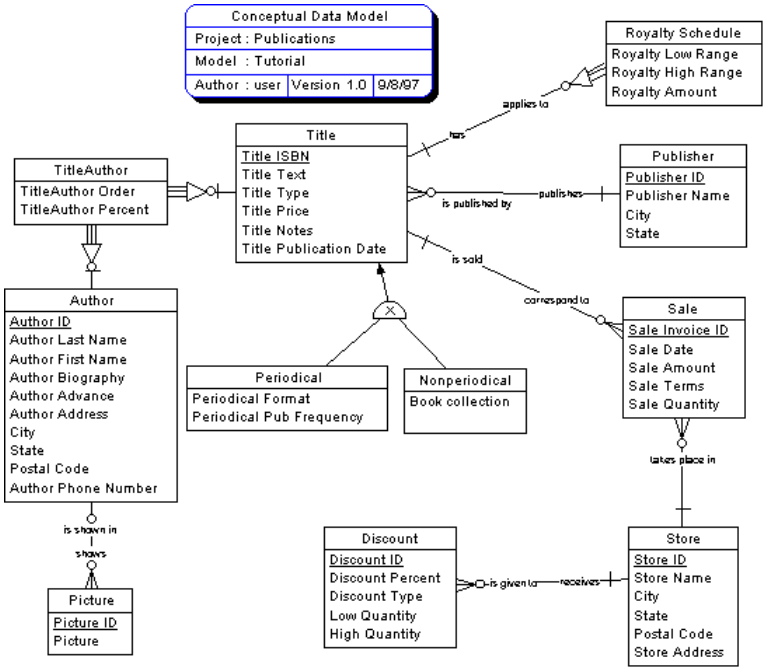
**2 Click the *center align* tool in the toolbar.**

This aligns the selected symbols along a central vertical axis.



**3 Use the *Alignment tools* to align the other symbols.**

You can align your model as follows:

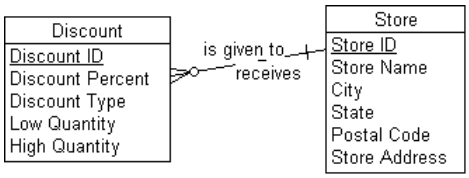


# Straighten relationship lines

**Where you are**
Chapter 8 Organizing the Display
Add a title box
Change the color of the title box
Change the color of the window
Arrange the symbols
➡ Straighten relationship lines
Center the model on the page
Print the model

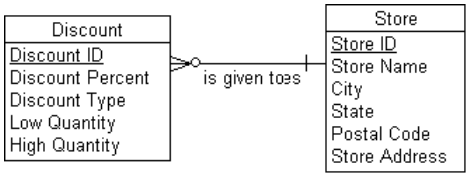
You will straighten the relationship line between the DISCOUNTS and STORES entities. If this line is already straight, use another relationship line.

**1 Click the *relationship* line.**




**2 Select *Arrange➤Disposition➤Horizontal*.**

The relationship line becomes horizontal.



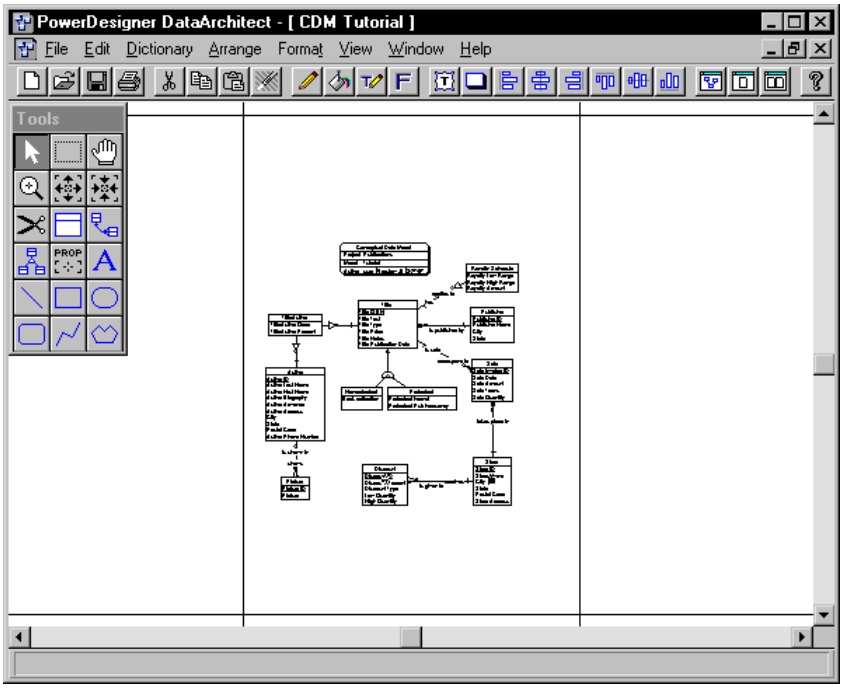

**Select all relationships at once**
You can select all the relationships at once by double-clicking the Relationship tool in the tool palette.

# Center the model on the page

**Where you are**
Chapter 8 Organizing the Display
Add a title box
Change the color of the title box
Change the color of the window
Arrange the symbols
Straighten relationship lines
➡ Center the model on the page
Print the model

**1 Click the *Zoom to Current Page* tool in the toolbar.**

This displays the entire current page in the work area. Your model may be spread over two pages, or may be off-center.



**2 Click the *Grabber tool* in the tool palette.**

**3 Click the *model*.**

This selects the whole model.

**4 Drag the model to the center of the page.**

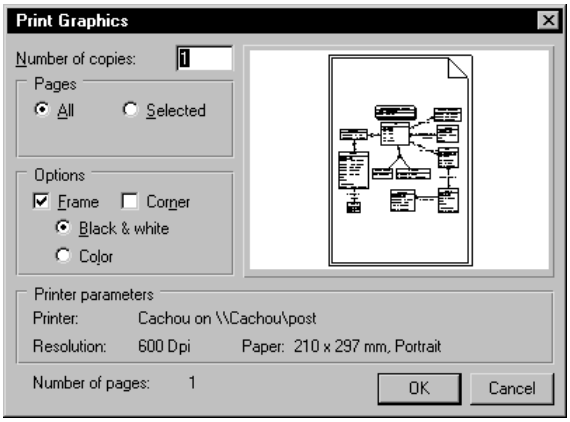
# Print the model

**Where you are**
Chapter 8 Organizing the Display
Add a title box
Change the color of the title box
Change the color of the window
Arrange the symbols
Straighten relationship lines
Center the model on the page
➡ Print the model

You can now print the finished model.

**1 Select** *File➢Print Graphics*.

The Print Graphics dialog box appears. One page is selected for printing.




**2 Click** *OK*.

**Select pages to print**
When a model is spread over several pages you can select and deselect pages for printing by clicking them. Selected pages display a turned down corner.

CHAPTER 10

# Generating the PDM from the CDM

In this lesson you will generate a Physical Data Model (PDM) from a Conceptual Data Model (CDM).

## What happens when you generate a PDM

You generate a PDM for a particular Database Management System (DBMS). Before you generate the PDM, you must select the DBMS referred to as the target database. DataArchitect translates the data types specified in the CDM into the physical data types which the target database supports.

The correspondence between conceptual and physical data types is defined in a DEF file. There is a DEF file for each type of target database.

When you generate a PDM, DataArchitect also translates the following conceptual objects into the following physical objects:

| Conceptual object | Physical object |
|---|---|
| Entity | Table |
| Entity attribute | Table column |
| Identifier | Primary key |
| Relationship | Reference and foreign key |

## What you will do

In this lesson you will:

- Generate the PDM
- Save and close the PDM
- Save CDM and exit DataArchitect

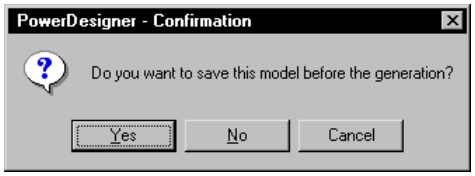
**How long will it take?**
About 5 minutes.

# Generate the PDM

**Where you are**
Chapter 10 Generating the PDM from the CDM
➡ Generate the PDM
Save and close the PDM
Exit DataArchitect

**1 Select** *Dictionary➢Generate Physical Model***.**

A message box asks you if you want to save changes to TUTORIAL.CDM.




**If you have installed Metaworks**
If you have installed Powerdesigner Metaworks you receive the Generate a Physical Data Model message box instead of the confirmation box shown above. In this case, select the Save Model Before Generation radio button.

**2 Click the** *Yes* **button**

The Generate a Physical Data Model dialog box appears.

**3 Select** *Sybase SQL Anywhere 5.5* **from the Database Name dropdown listbox.**

To follow the exercises in the PDM tutorial, you must select Sybase SQL Anywhere 5.5. However, you do not have to install the DBMS itself.

The dialog box proposes TUTORIAL.PDM as the default name for the PDM file. It is the same as the CDM file name but with the extension PDM.

**4 Clear the** *Preserve Modifications* **checkbox.**

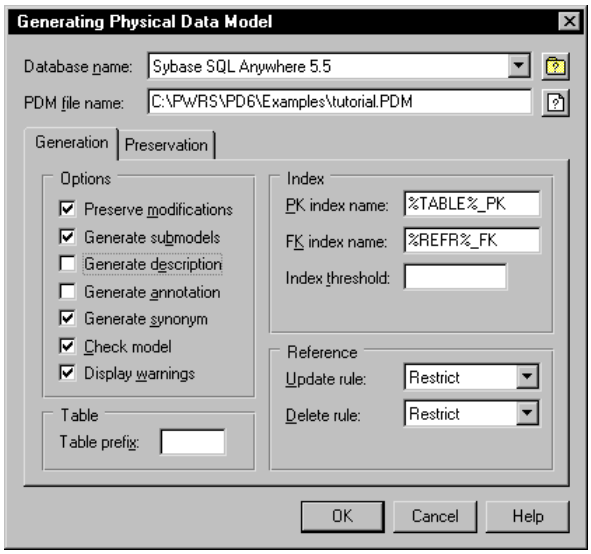
With this option, you will replace an existing PDM with the same name.

**5 Clear the *Generate Submodels* checkbox.**

This means that the submodel that contains the inheritance of TITLE, PERIODICAL and NONPERIODICAL will not generate a PDM submodel.

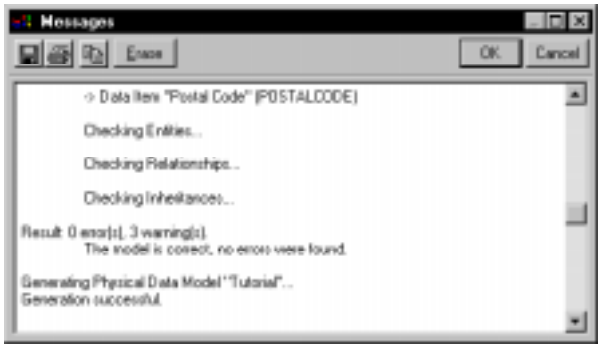
**6 Select the *Display Warnings* checkbox.**

This means that warning messages will be displayed in the Messages window when you generate the PDM.




**7 Click *OK*.**

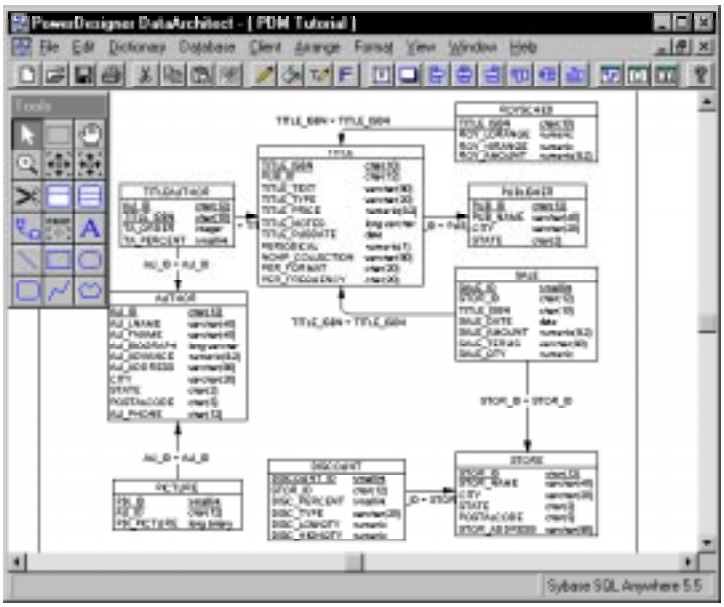
DataArchitect checks the model before it generates the PDM. The Messages window displays the result of the check.



Warning messages indicate that the CDM reuses certain data items for more than one entity. These warnings are informational and do not prevent DataArchitect from generating the PDM.

**8 Click *OK* to close the message window.**

The PDM appears in the model window.




**Crowded PDM**
If your PDM is difficult to read because tables display too much information, you can reduce the amount of information displayed by selecting File➤Display Preferences and clearing checkboxes.

**Three inheritance entities resulted in one table**
In the CDM, you defined the inheritance Title Inherit to generate the parent only. In the PDM, the resulting table, Title, includes a column, Periodical, that indicates if a title is a periodical or not. It also includes columns related to periodicals or nonperiodicals.

## What you learned

In this section, you learned how to:

- Define generation parameters according to how you want to generate objects in the PDM
- Generate a PDM from a CDM

# Save and close the PDM

**Where you are**
Chapter 10 Generating the PDM from the CDM
Generate the PDM
➡ Save and close the PDM
Exit DataArchitect

You will save and close the generated PDM.

**1 Select *File*➤*Save* from the menu bar.**

This saves the PDM.

**2 Select *File*➤*Close* from the menu bar.**

This closes the PDM window and returns you to the CDM window.

# Exit DataArchitect

**Where you are**
Chapter 10 Generating the PDM from the CDM
Generate the PDM
Save and close the PDM
➡ Exit DataArchitect

You will close the CDM and exit DataArchitect.

- **Select *File*➤*Exit* from the menu bar.**

  You exit the DataArchitect application.

You have now completed the CDM tutorial.

PART TWO

# PDM Tutorial

This part is an introduction to the DataArchitect physical data modeling environment.

CHAPTER 11

# About the PDM Tutorial

This tutorial is a series of nine lessons in which you learn how to use DataArchitect to create a Physical Data Model (PDM).

In this tutorial, you complete the PDM that you generated in the CDM tutorial. If you did not do the CDM tutorial, you can start with the PDM tutorial.

You learn how to denormalize the generated PDM so that you can archive data, speed up access to information in the database, and ensure database integrity.

## What is a PDM?

A PDM represents the physical structure of the data as it will be implemented by a Database Management Systems (DBMS). The PDM takes into account the features and physical restrictions of the DBMS of your choice.

## Why build a PDM?

In a PDM, you can optimize database characteristics by customizing tables, columns, indexes, referential integrity, views, physical storage, triggers, and stored procedures. Database generation and modification procedures implement these characteristics in the way best adapted to the DBMS of your choice.

You build the PDM based on the solid design framework of the Conceptual Data Model (CDM). You can update the CDM and preserve modifications to the PDM, thereby updating your design without losing any details of physical implementation.

# What you will do

Chapter 12

You will outline your approach to physical implementation by preparing a list of physical considerations to be taken into account in the PDM.

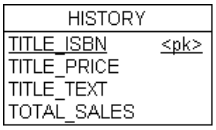
Chapter 13

You will start DataArchitect and open the tutorial PDM. This PDM presents the physical structure of a publishing enterprise.

You will specify model preferences,options, and properties. You will save the model.

Chapter 14

You will denormalize your PDM by adding a table, assigning it columns including a primary key column.



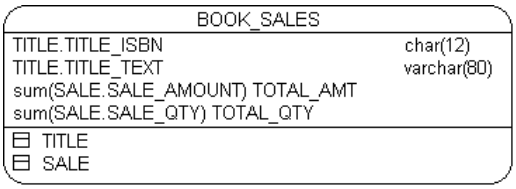

Next, you will create a primary key index for the new table. You will also denormalize the PDM for performance enhancement by creating additional indexes.

Chapter 15

You will define a reference that indicates how this new table relates to the rest of the database. You will define referential integrity for another reference.

Chapter 16

You will create a view to allow users to view subsets of tables without giving them full access to the tables themselves. You will create calculated columns in the view.




Chapter 17

You will create extended attributes and save them in a file.

Chapter 18

You will define a cascade-type trigger and you will preview the trigger script. You will create a stored procedure which adds data to the calculated column in the new table. You will generate the triggers and procedure in a script file.

Chapter 19

You will define a server expression for a business rule and you see how this validation rule applies to a table. You will generate the script which you can use to generate a database.

Chapter 20

You will create data profiles and define test data generation sources for each profile. You will assign the data profiles to appropriate columns in a table, then generate a script which you can use to generate test data in a database. You will then save and close the PDM.

## How long it will take

You can do this part of the tutorial in one sitting in about one hour and a quarter. You can also stop after any lesson, save your model, and continue at another time.

## What you will learn

You will learn basic DataArchitect techniques for modifying a PDM, including:

- How to add tables and columns to a PDM, and how to designate primary keys
- How to create references and define referential integrity
- How to create indexes
- How to create and customize a view
- How to create extended attributes
- How to create, preview, and generate a trigger script
- How to generate a database creation script
- How to generate a test data creation script

# Setting up

Before you begin, make sure that the files you need for the exercises are on your hard disk. When you install DataArchitect, these files are installed in subdirectories of your PowerDesigner directory. When you have finished with this tutorial you can delete them if you want.

The PDM tutorial uses the following files:

| File | Subdirectory | Description |
|---|---|---|
| PDMBEFOR.PDM | PD6\EXAMPLES | Starting tutorial PDM |
| PDMAFTER.PDM | PD6\EXAMPLES | Finished tutorial PDM |

CHAPTER 12

# Defining Physical Implementation

You build a Physical Data Model (PDM) to address the details of physical implementation. To do this, you need to decide:

- What sort of information you need to store
- What sort of information you need to retrieve, how often, and in which form
- How will the database be accessed

When the particular needs of implementation are clear, you can begin to develop the PDM.

## Physical description

You will start this tutorial with a PDM that is the model of the information system for a publishing company. The database that implements this information system will be used by publishers and their staff, as well as by book stores and distributors.

The database must provide fast access to information about titles and sales.

## Implementation issues

In the course of the PDM tutorial, you will address the following implementation issues:

- Store and update historical information on sales for each title
- Provide consultation-only access to a limited amount of information
- Control updates and deletion

CHAPTER 13

# How to Begin the PDM Tutorial

You will begin the tutorial by running DataArchitect and opening the tutorial Physical Data Model (PDM).

In this chapter you will:

- Open the tutorial PDM
- Define PDM preferences and options
- Define PDM properties
- Save the tutorial PDM

**How long will it take?**
About 5 minutes.

# Open the tutorial PDM

**Where you are**
Chapter 3 How to Begin
➡ Open the tutorial PDM
Define PDM preferences and options
Define PDM properties
Save the tutorial PDM

**1 Click the *DataArchitect* program icon.**

The DataArchitect window appears. It contains an empty CDM workspace. You have to open the PDM workspace.

**2 Select *File*➤*Physical Model*➤*Open*.**

A file selection window appears.

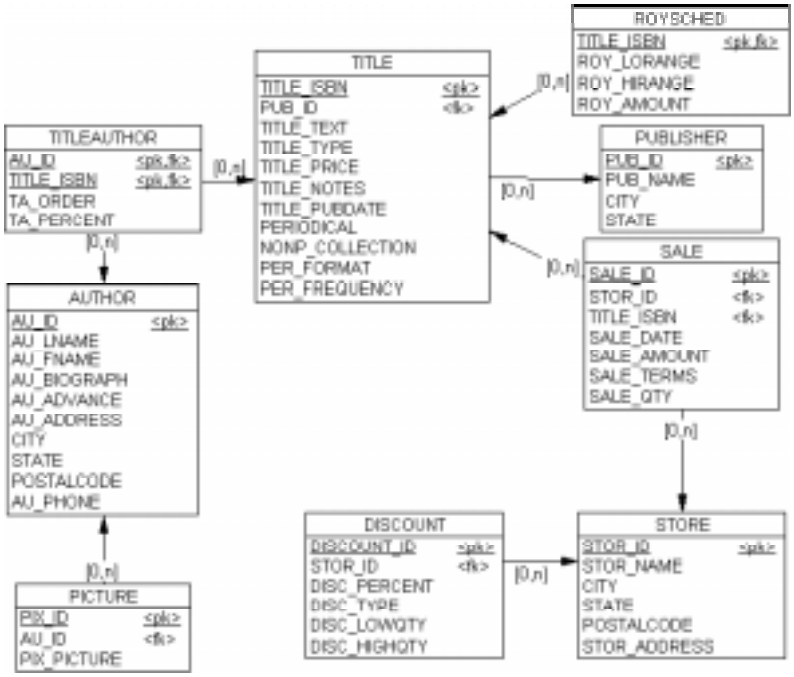
**3 Select *PDMBEFOR.PDM* from the EXAMPLES directory.**

**4 Click *OK*.**

DataArchitect displays the model in the PDM workspace.

**5 Click the *Zoom All* tool in the toolbar.**

The whole model fits in the PDM workspace.




**Your screen looks different**
All the screen captures in this book were taken in a Windows 95 environment, with an 800x600 screen resolution. The appearance and proportions of the images which appear on your screen may be slightly different.

# Define PDM preferences and options

**Where you are**
Chapter 3 How to Begin
Open the tutorial PDM
➡ Define PDM preferences and options
Define PDM properties
Save the tutorial PDM

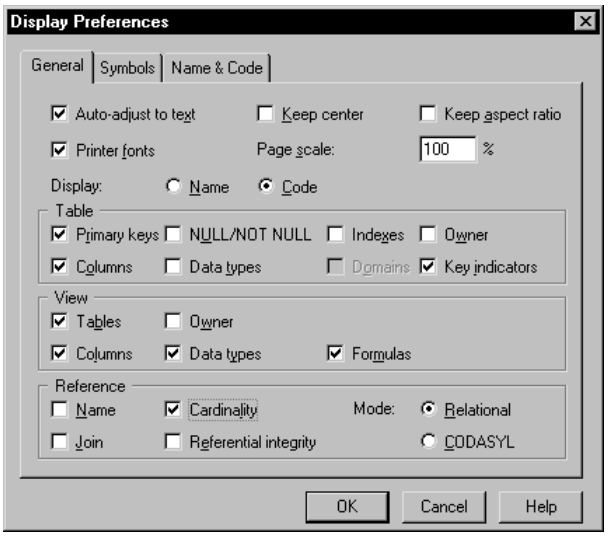
Before you begin working, you will define certain display preferences and model options for the PDM.

For a complete description of all PDM preferences and options, see *PowerDesigner DataArchitect User's Guide*.

**1 Select *File➤Display Preferences* from the menu bar.**

The Display Preferences dialog box appears.

**2 Select or clear checkboxes and radio buttons so that the dialog box looks like the one shown below.**



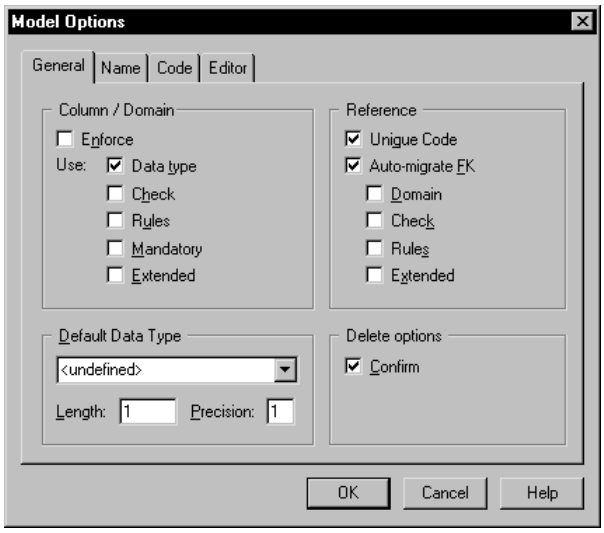

These display preferences display object codes and identify all keys. They adjust tables and views to fit their text.

**3 Click** *OK*.

**4 Select** *File*➢*Model Options* **from the menu bar.**

The Model Options dialog box appears.

**5 Select or clear checkboxes and radio buttons so that the dialog box looks like the one shown below.**



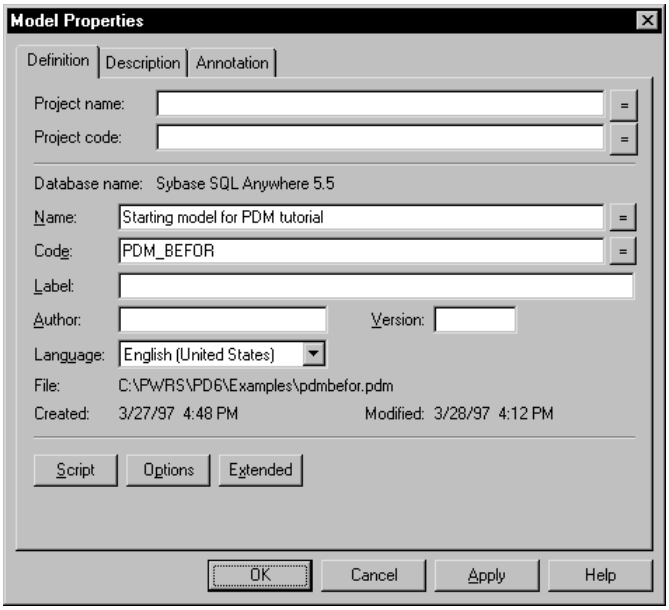

**6 Click** *OK*.

# Define PDM properties

**Where you are**
Chapter 3 How to Begin
Open the tutorial PDM
Define PDM preferences and options
➡ Define PDM properties
Save the tutorial PDM

**1 Select *Dictionary➢Model Properties* from the menu bar.**

The Model property sheet appears.




**2 Type *Publications* in the Project Name box.**

This is the name of the project to which the PDM belongs.

**3 Click the = button at the end of the Project Code box.**

This sets the code equal to the name.

4 **Type *Tutorial* in the Name box.**

This is the name of the PDM.

5 **Click the = button at the end of the Code box.**

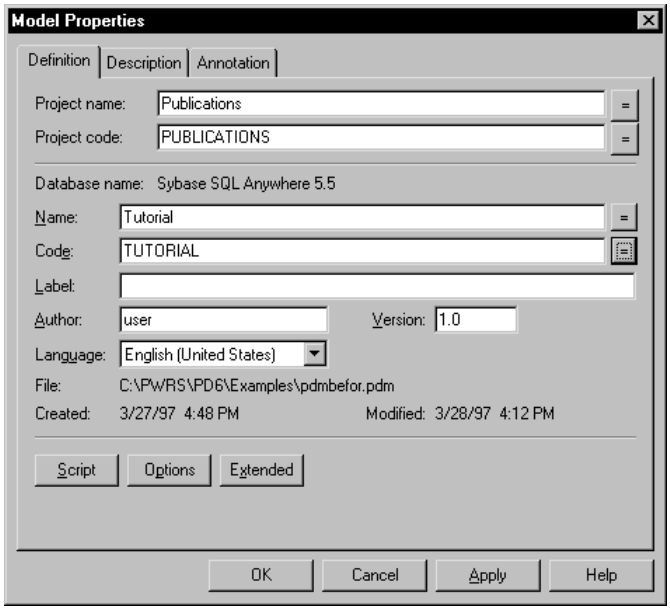
This sets the code equal to the name.

6 **Type your *name* in the Author box.**

This indicates that you are the owner of the PDM.

7 **Type *1.0* in the Version box.**

This indicates that this is the first version of the PDM.




8 **Click *OK*.**

# Save the tutorial PDM

**Where you are**
Chapter 3 How to Begin
Open the tutorial PDM
Define PDM preferences and options
Define PDM properties
➡ Save the tutorial PDM

You will save the tutorial PDM in another file. This leaves the original tutorial PDM intact so you can use it again if you want to redo the exercises.

**1 Select** *File➤Save As*.

The File Save As dialog box appears.

**2 Type** *TUTORIAL.PDM* **in the File Name box.**

This is the name of the file in which you will work and save your modifications.

**3 Click** *OK*.

This saves your model in the TUTORIAL.PDM file.

**Save your work**
Save your work periodically while doing these exercises by selecting File➤Save.

CHAPTER 14

# Creating a Table in the PDM

You will create a table in the PDM. This table will be named HISTORY and will contain a record of all sales.

You will create this table directly in the PDM because you only use it to store data. It has no role in the functional structure of the database and, therefore, does not form part of the conceptual model.

You will also add columns to the table and designate a primary key.

Adding objects to a PDM in this way is called denormalizing the physical model.

## About indexes

Indexes speed up access to data when you are searching for information in tables. You can create indexes for particular columns of a table in the PDM. When you perform a search in a table, for example, by executing a SQL SELECT query, the search process can find the rows matching the search criteria more quickly when the model columns are indexed. It is much more efficient to scan an index than to scan through the data contained in each row.

## Primary and foreign key indexes

DataArchitect automatically creates indexes for primary and foreign key columns, but you can also create your own indexes for other columns, depending on the type of information you are likely to want to access in the database.

For example, if you want to search for an author in the database you are likely to perform a search on the author's last name. In this case you would index the columns which contain this information.

In this lesson you will:

- Add a table
- Add columns to the table
- Designate a primary key
- Create a primary key index
- Create an alternate key index

**How long will it take?**
About 15 minutes.

# Add a table

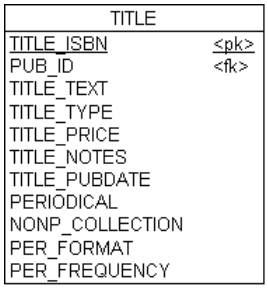


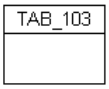
You will add the HISTORY table to the PDM.

**1 Click the *Table* tool in the tool palette.**

**2 Click the *PDM workspace* underneath the TITLE table symbol.**

A table symbol appears at the click position.




The table has the code TAB_*n*, where *n* is a number assigned in the order of creation of objects.

**3 Click the *Pointer* tool in the tool palette.**

**4 Double-click the *table* symbol.**

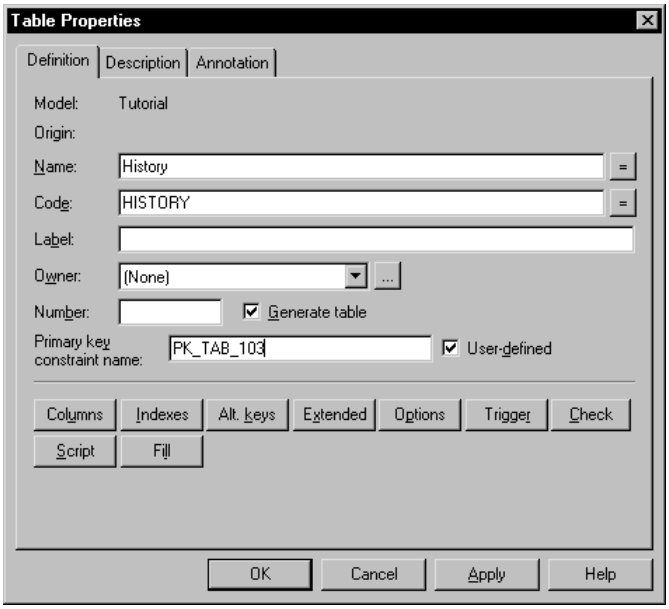
The Table Properties dialog box appears.

**5 Type *History* in the Name box.**

This is the name of the table.

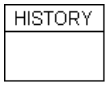
**6 Click the = button at the end of the Code box.**

This sets the code equal to the name.




**7 Click *OK*.**

The table symbol appears in the PDM workspace.




## What you learned

In this section, you learned how to:

- Customize a PDM by adding a table to store data

# Add columns

**Where you are**
Chapter 14 Creating a Table in the PDM
Add a table
➡ Add columns
Create columns
Create a primary key index
Create an alternate key index

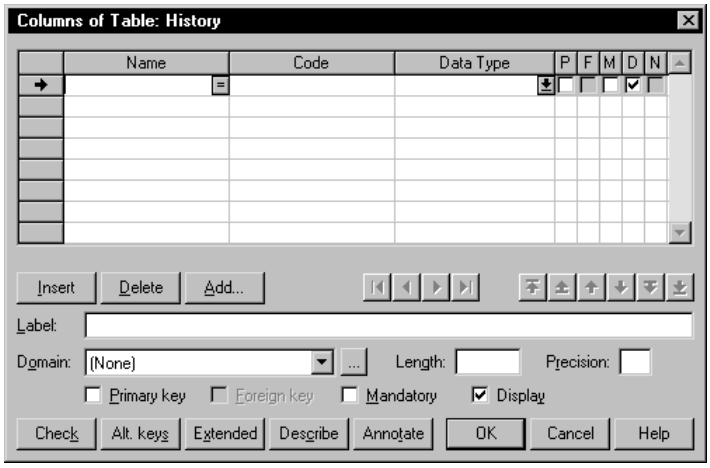
You will add the columns TITLE_ISBN number and TITLE_TEXT to the HISTORY table.

**1 Double-click the *HISTORY* table symbol.**

The Table Properties dialog box appears.

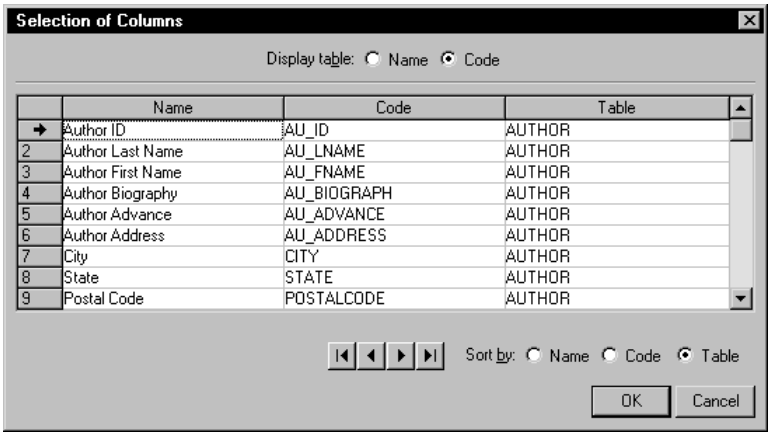
**2 Click the *Columns* button.**

The Columns of the Table dialog box appears. The table does not yet have any columns so the list is empty.



**3 Click the *Add* button.**

The Selection of Columns dialog box appears. It shows the name of each column, its code, and the tables to which it belongs.

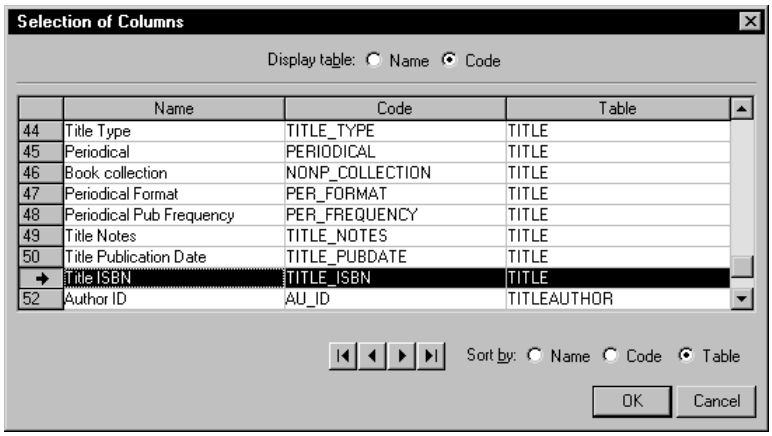



**4 Click the *Table* radio button.**

This sorts the columns alphabetically by table.

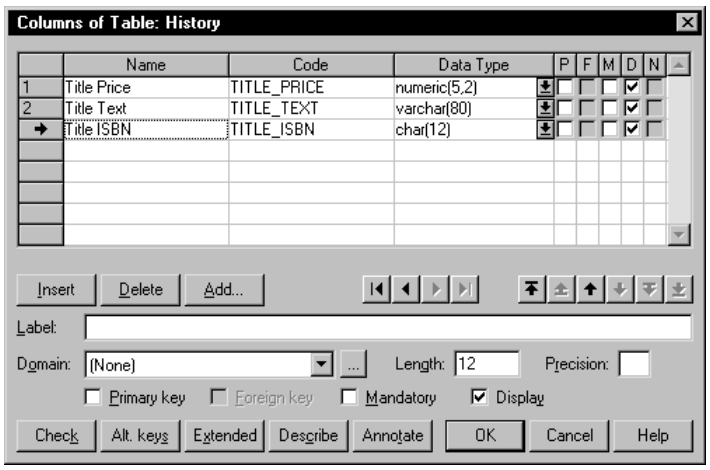
**5 Scroll down the list to the columns in the *TITLE* table.**

6 **Click the number at the beginning of the *Title ISBN* line.**
**Press** CTRL **while you click the number at the beginning of the *Title Price* line.**
**Press** CTRL **while you click the number at the beginning of the *Title Text* line.**




7 **Click *OK*.**

The selected columns appears in the list of columns.




8 **Click the number at the beginning of the *Title ISBN* line.**

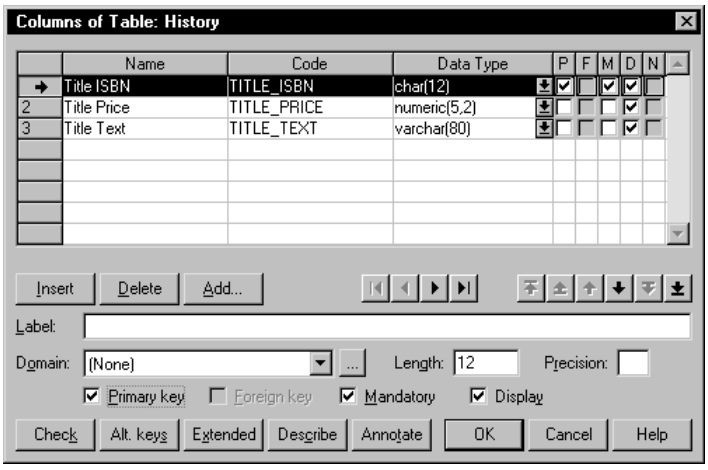
An arrow appears at the beginning of the line and the whole line is highlighted.

**9 Click the button.**

Title ISBN move to the top of the list. This means it will generate the first column in the database table.

**10 Select the *Primary key* checkbox in the bottom part of the dialog box.**

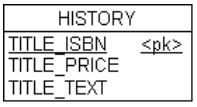
A check appears in the P and M columns. The check in the P column indicates that Title ISBN is the primary key.




**Mandatory values**
Mandatory values translate to a Not null field in most SQL databases.

**11 Click *OK* in each of the dialog boxes.**

This saves the column definition. The HISTORY table symbol appears in the PDM workspace. The fact that TITLE_ISBN is underlined and has the symbol <pk> next to it indicates that it is a primary key column.




## What you learned

In this section, you learned how to:

- Add an existing column to a table in the PDM
- Change the order of columns in a table

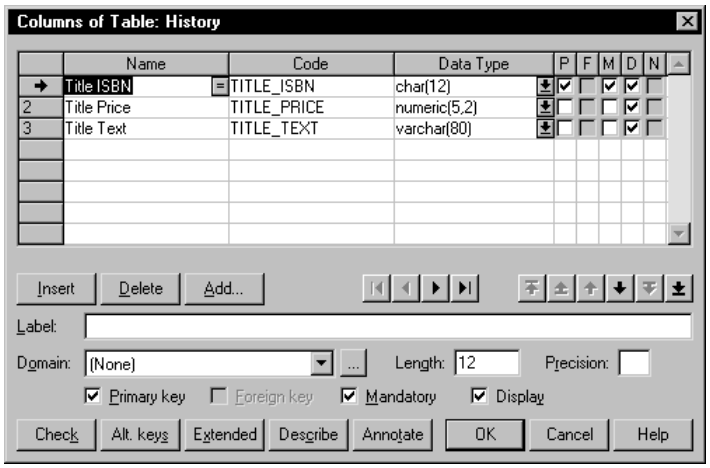
# Create columns

**Where you are**
Chapter 14 Creating a Table in the PDM
Add a table
Add columns
➡ Create columns
Create a primary key index
Create an alternate key index

You will create a new column in the HISTORY table.

**1 Press** CTRL **while you double-click the** *HISTORY* **table symbol.**

The list of columns in the HISTORY table appears.



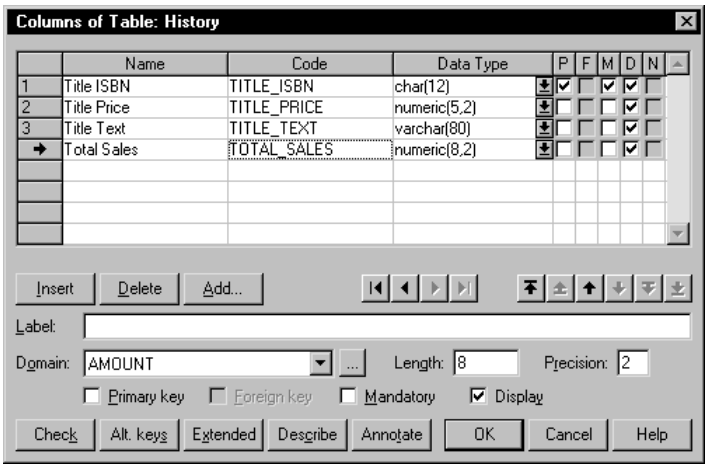

**2 Type** *Total Sales* **in the Name column of the first blank line.**

**3 Click the** *Code* **column.**
**Click the = button that appears in the Code column.**

This sets the code equal to the name, with lowercase characters replaced by uppercase and the space replaced by an underscore.

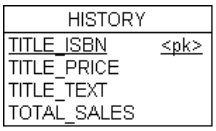
**4 Select *AMOUNT* from the Domain dropdown listbox in the bottom part of the dialog box.**

The data type column displays numeric(8,2) which is a data type available for the target database (in this case, Sybase SQL Anywhere).




**5 Click *OK* in each of the dialog boxes.**

In the PDM workspace, the table symbol contains the new column.




## What you learned

In this section, you learned how to:

- Create a new column
- Identify a column by a name and a code
- Attach a column to a domain

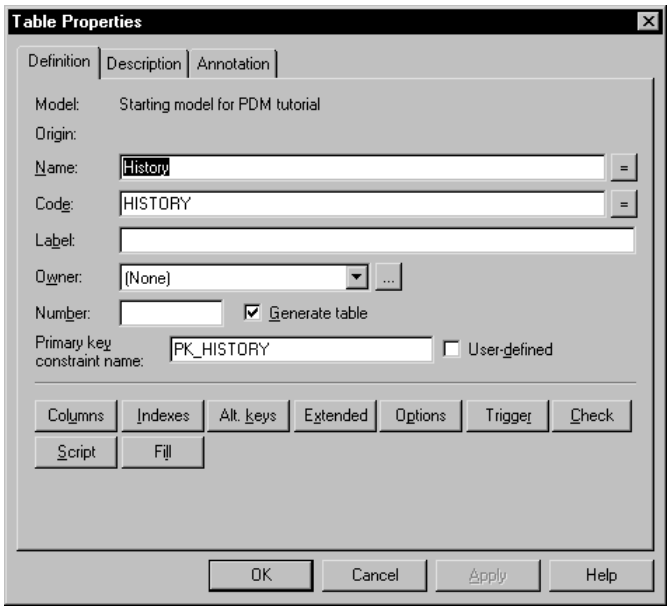
# Create a primary key index

**Where you are**
Chapter 14 Creating a Table in the PDM
Add a table
Add columns
Create columns
➡ Create a primary key index
Create an alternate key index

You will define a primary key index for the HISTORY table.

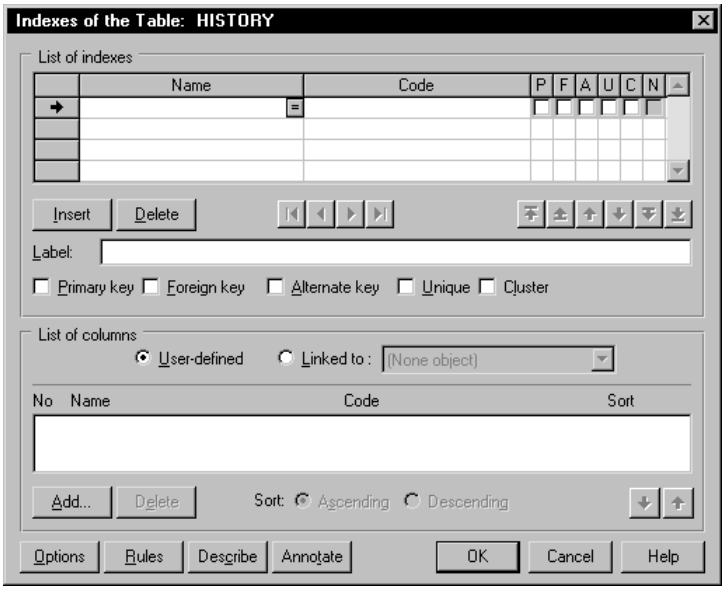
**1 Double-click the *HISTORY* table.**

The Table Properties dialog box appears.



**2 Click the *Indexes* button.**

The Indexes of the Table dialog box shows that this table has no indexes.



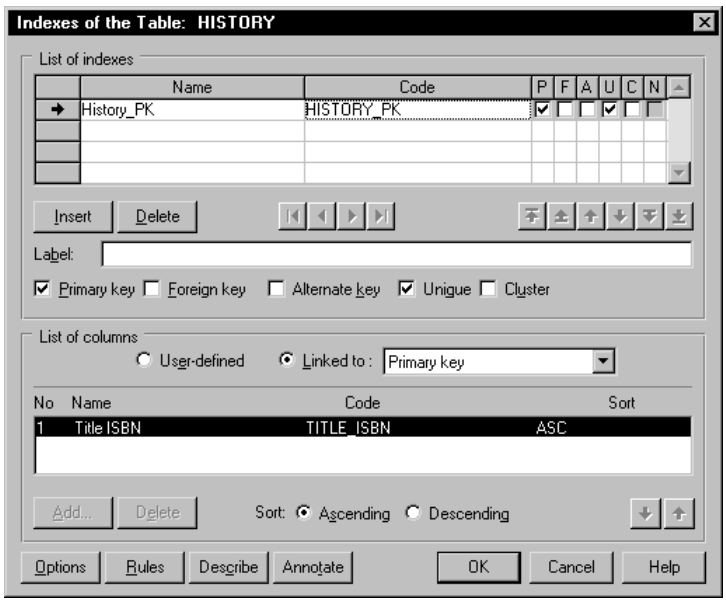

**3 Click the first *blank line* in the list.**

An arrow appears at the beginning of the line.

**4 Type *HISTORY_PK* in the Name column.**
**Click the *Code* column.**
**Click the = button that appears in the column.**

**5 Select the *Primary Key* checkbox.**

The Unique checkbox is selected automatically. The index History_PK is automatically linked to the primary key TITLE_ISBN. TITLE_ISBN appears in the List of Columns.




**6 Click *OK* in each of the dialog boxes.**

You return to the PDM workspace.

# Create an alternate key index

**Where you are**
Chapter 14 Creating a Table in the PDM
Add a table
Add columns
Create columns
Create a primary key index
➡ Create an alternate key index

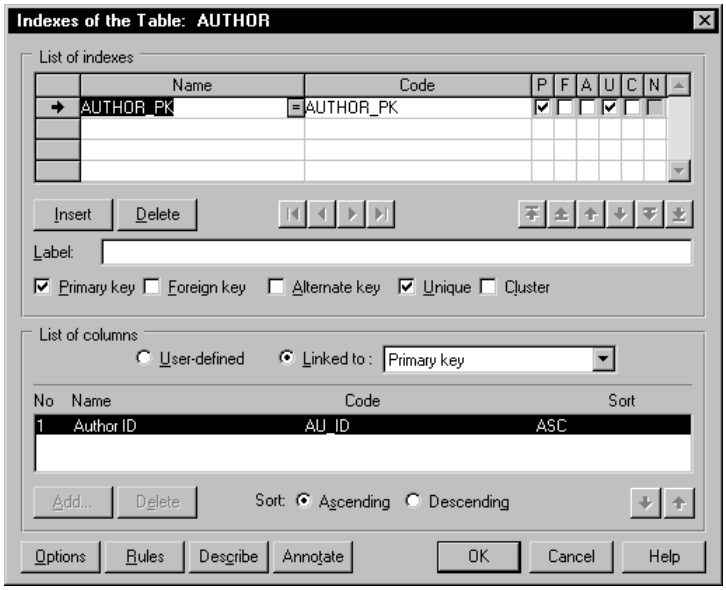
The column that contains author last names is not automatically indexed because it is not a primary or a foreign key. To be able to search for an author more quickly when you only know the author's last name, you will create an index for this column.

**1 Double-click the *AUTHOR* table.**

The Table Properties dialog box appears.

**2 Click the *Indexes* button.**

The Indexes of the Table dialog box displays the indexes belonging to the table. This list shows that the primary key is indexed.

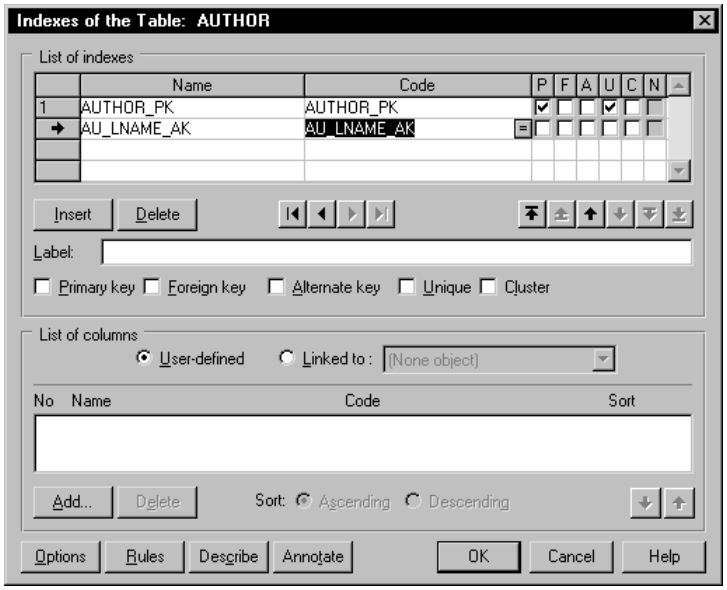


**3 Click the first *blank line* in the list.**

An arrow appears at the beginning of the line.

**4 Type *AU_LNAME_AK* in the Name column.**
**Click the *Code* column.**
**Click the = button that appears in the column.**

This is the name of the index for the last names of the authors.



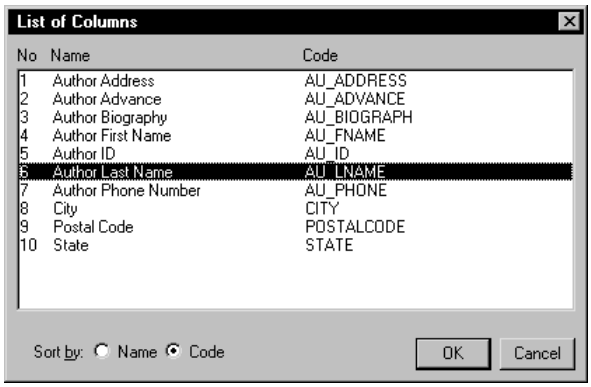

**5 Select the *Alternate Key* checkbox.**
The Unique checkbox and the User-Defined radio button are selected automatically.

**6 Click the *Add* button.**

The List of Columns dialog box appears. It lists all the columns in the table.

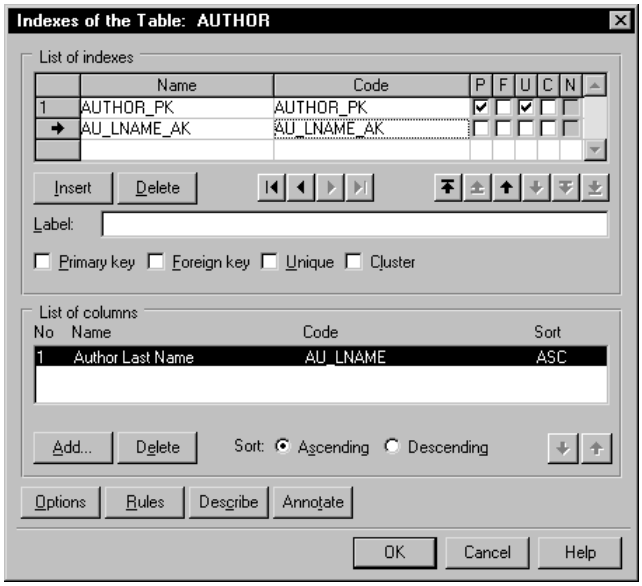
**7 Select** *Author Last Name*.

This is the column to which you want to attach the AU_LNAME_AK index.




**8 Click** *OK*.

You return to the list of indexes of the table. Author Last Name appears in the List of columns. ASC in the Sort column means that the index sorts the last names of authors in ascending alphabetic order from A to Z.



**9 Click *OK* in each of the dialog boxes.**

You return to the PDM workspace.

## What you learned

In this section, you learned how to:

- Speed access to information in the database by creating indexes for the columns storing relevant data
- Identify an index by a name and a code
- Select the columns you want to index
- Define how values inside the indexed columns are sorted

CHAPTER 15

# Defining a Reference and Referential Integrity

## About references

A **reference** symbolizes the link between two tables.

You can create a reference between two tables in the PDM. You do this when you want one column to refer to another. When you create a reference, the primary key in the parent table migrates to a foreign key in the child table.

## About referential integrity

**Referential integrity** dictates what happens to a foreign key column in a child table when you update or delete the value of the corresponding primary key column in the parent table.

Store ID is the primary key column in the STORE table. It contains the unique identification code of a store. A reference links the value of the Store ID in the STORE table to the Store ID column in the DISCOUNT table. Using the referential integrity options, you can specify that if you delete a store from the STORE table, you also delete all of its corresponding records in the DISCOUNT table.

In this lesson you will:

- Create a reference
- Define referential integrity

**How long will it take?**
About 5 minutes.

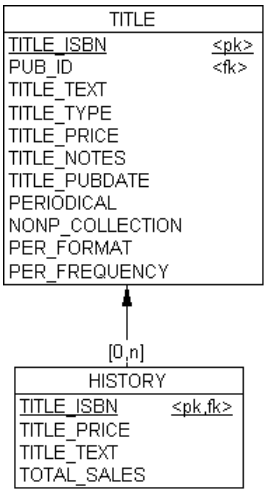
# Create a reference

**Where you are**
Chapter 15 Defining a Reference and Referential Integrity
➡ Create a reference
Define reference properties
Define referential integrity

You will create a reference between the HISTORY and TITLE tables. This reference represents the sales history for each title.

**1 Click the *Reference* tool in the tool palette.**

**2 Drag a *reference* link from HISTORY to TITLE.**




HISTORY is the child table and TITLE is the parent table.
TITLE_ISBN is the primary key of both tables. It is also the foreign key in th HISTORY table as indicated by the symbol <pk,fk>.

## What you learned

In this section, you learned how to:

- Create a reference between two tables
- Identify the parent and child tables in a reference

# Define reference properties

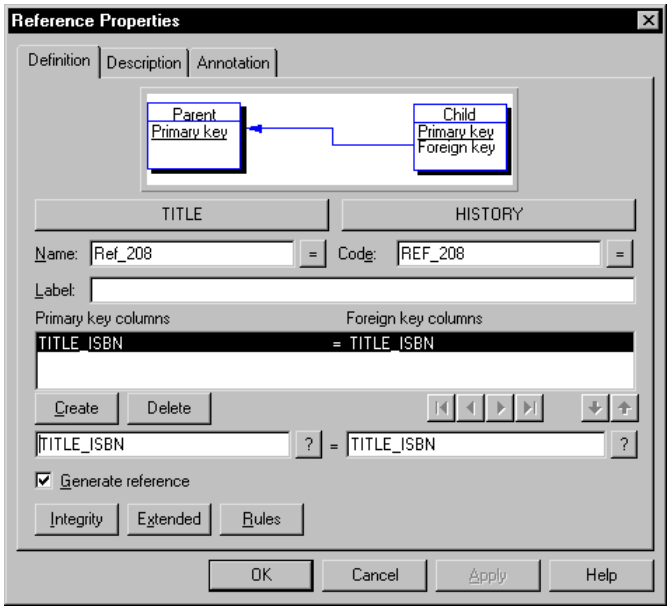
**Where you are**
Chapter 15 Defining a Reference and Referential Integrity
Create a reference
➡ Define reference properties
Define referential integrity

You can define a name for a reference. This makes it easier to identify.

**1 Click the *Pointer* tool in the tool palette.**

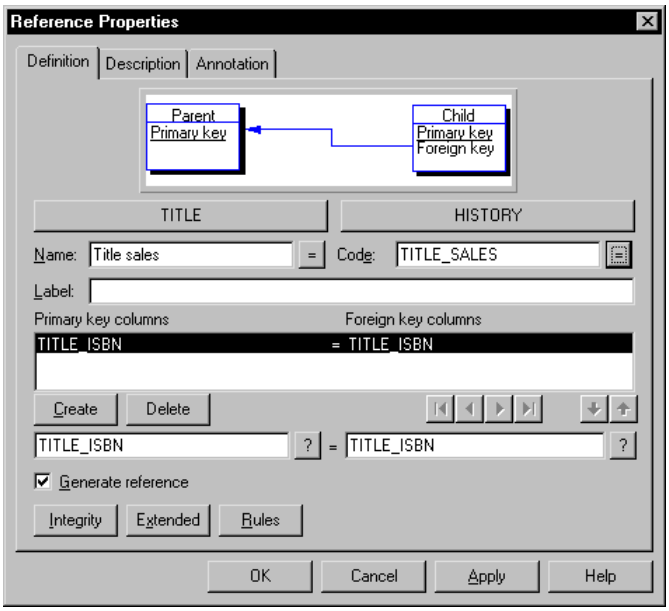
**2 Double-click the *reference* link between HISTORY and TITLE.**

The Reference Properties dialog box appears.



3 **Type** *Title Sales* **in the Name box.**

4 **Click the** = **button at the end of the Code box to set the code equal to the name.**




5 **Click** *OK*.

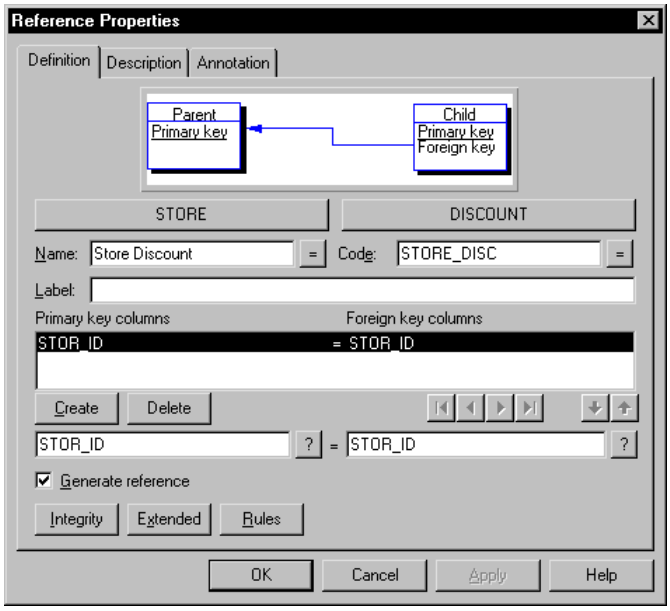
# Define referential integrity

**Where you are**
Chapter 15 Defining a Reference and Referential Integrity
Create a reference
Define reference properties
➡ Define referential integrity

You will use the referential integrity options to implement cascade update when a store is deleted. Since discounts are specific to store, if you delete a store you delete its associated discounts.

**1 Double-click the *reference* link between DISCOUNT and STORE.**

The Reference Properties dialog box appears.



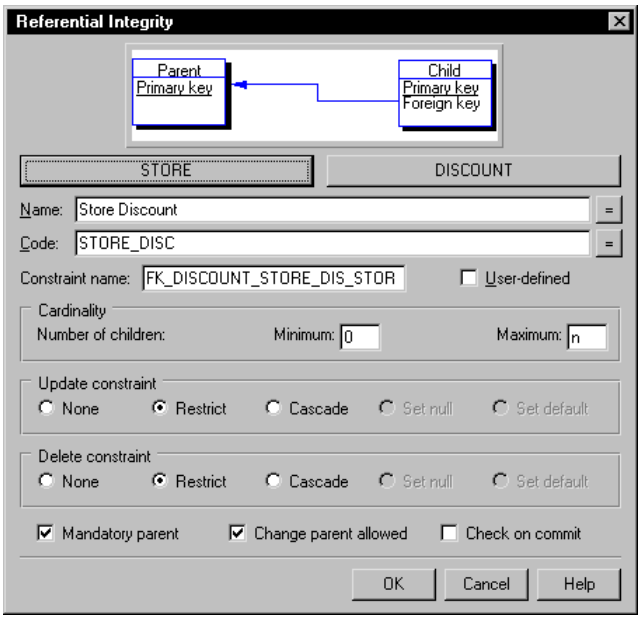

**2 Click the *Integrity* button.**

The Referential Integrity dialog box appears.

The Restrict radio buttons indicate to use Restrict referential integrity by default.

**3 Click the *Cascade* radio button in the Delete Constraint groupbox.**

This means that if you delete the Store ID in the STORE table you delete the corresponding records in the DISCOUNT table.




**4 Click *OK* in each of the dialog boxes.**

You return to the PDM workspace.

## What you learned

In this section, you learned how to:

- Define what happens to related values in child tables when you delete or modify a value in the parent table

CHAPTER 16

# Creating a View

## About views

A **view** is an alternative way of looking at the data in one or more tables. A view contains a subset of columns from one or more tables. You can create a view to allow users to see subsets of tables without giving them full access to the tables themselves.

## What happens when you create a view

Creating a view is equivalent to defining a SQL SELECT query to select objects in the database. When you create a view you select the tables and columns you want to include in the view. In this lesson, you will create a view on the TITLE and SALES tables. This view generates a SQL query which automatically selects all the columns in these tables and displays a graphic symbol representing the view.

In this lesson you will:

- Compose the view
- Customize the view
- Create calculated columns
- Preview the SQL query

**How long will it take?**
About 10 minutes.

# Compose the view

**Where you are**
Chapter 16 Creating a View
➡ Compose the view
Customize the view
Create calculated columns
Preview the SQL query

You will compose a view of the TITLE and SALES tables.

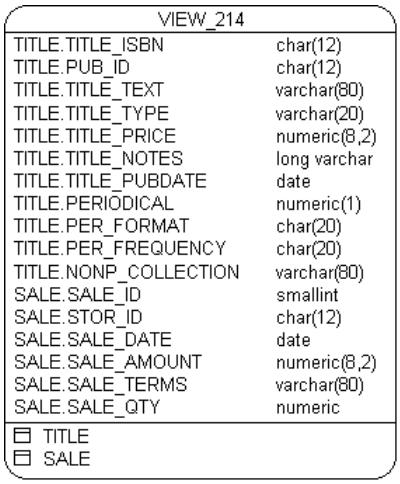
**1 Select the *TITLE* table symbol.**

**2 Press** SHIFT **while you click the *SALE* symbol.**

Both table symbols are selected.

**3 Select *Dictionary*➢*Views*➢*New* from the menu bar.**

A view appears in the PDM workspace. The view lists all the columns belonging to the selected tables. At the bottom of the view, it lists the tables.




The view has the code VIEW_*n*, where *n* is a number assigned in the order of creation of objects.

# Customize the view

**Where you are**
Chapter 16 Creating a View
Compose the view
➡ Customize the view
Create calculated columns
Preview the SQL query

You will customize the view so that it contains only certain columns.

**1 Double-click the *view* symbol.**

The View Properties dialog box appears.

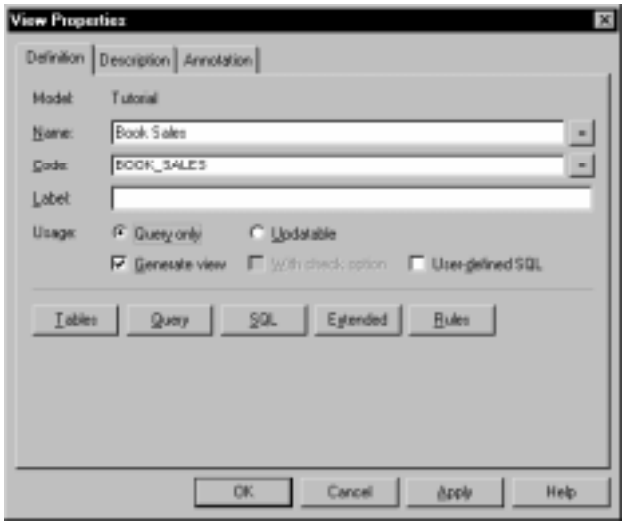
**2 Type *Book Sales* in the Name box.**

This is the name of the view.

**3 Click the = button at the end of the Code box to set the code equal to the name.**

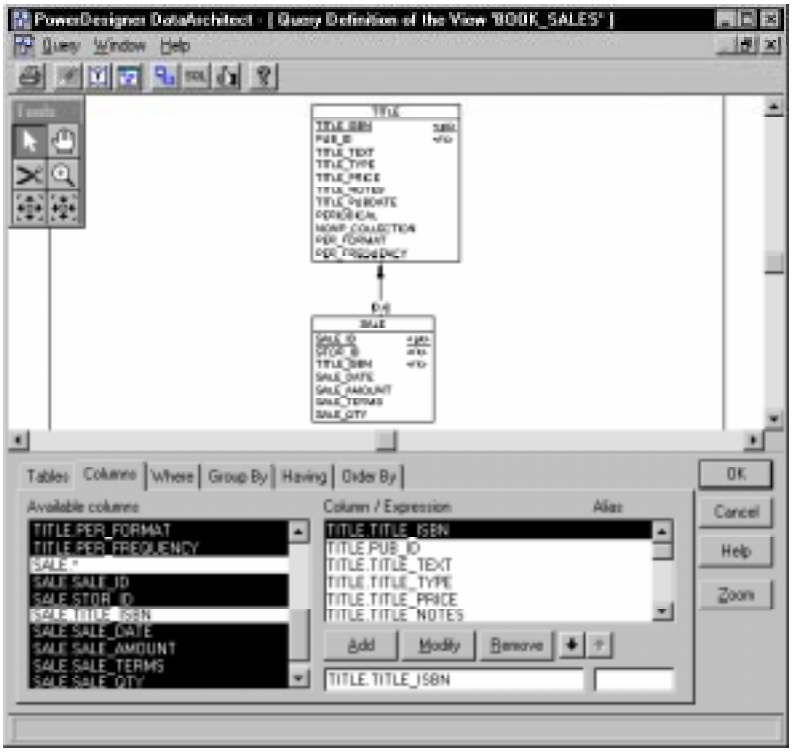
**4 Select the *Query Only* radio button.**

This limits access to consultation only.



**5 Click the *Query* button.**

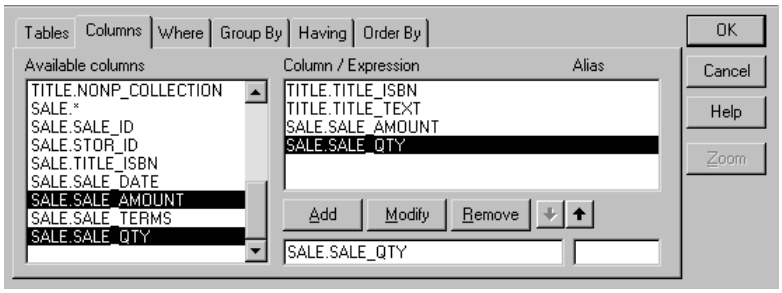
A window appears, showing the definition of the view.




The top part of the window shows the graphic symbols representing the TITLE and SALE tables. The bottom part of the window lists all the columns included in the view and the tables to which they belong.

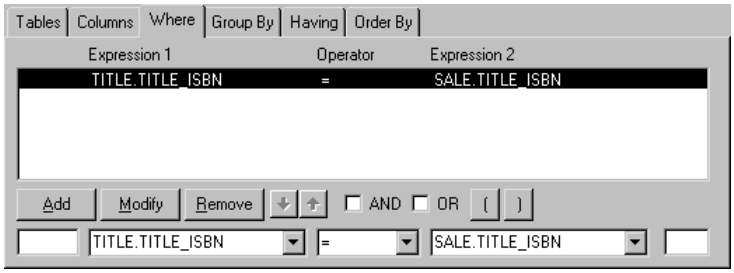
**6 In the Available Columns list, deselect all columns except for the following: *TITLE.TITLE_ISBN*, *TITLE.TITLE_TEXT*, *SALE.SALE_AMOUNT*, and *SALE.SALE_QTY*.**

The Column/Expression list displays the selected columns.




**7 Click the *Where* tab.**

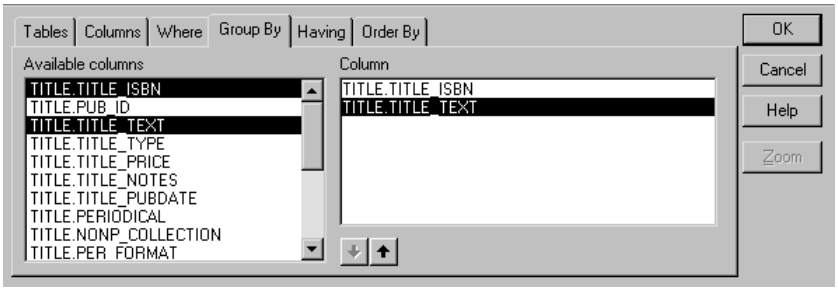
The Where page shows the link between SALE and TITLE tables by TITLE_ISBN.




**8 Click the *Group By* tab.**

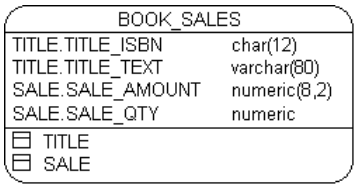
**9 Select *TITLE.TITLE_ISBN* and *TITLE.TITLE_TEXT* in the List of Available columns.**

This choice groups titles by ISBN number and title.




**10 Click *OK*.**

You return to the PDM workspace. The view now only lists the selected columns.




## What you learned

In this section, you learned how to:

- Restrict access to information further by including only certain columns in a view
- Optimize the view using a GROUP BY statement

# Create calculated columns

**Where you are**
Chapter 16 Creating a View
Compose the view
Customize the view
➡ Create calculated columns
Preview the SQL query

You will add two calculated columns to the view. One column will contain the total number of books sold for a title. The other will contain the total dollar amount of sales for a title.

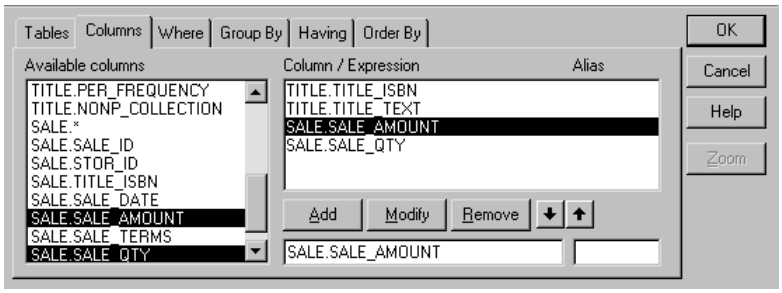
**1 Double-click the *view* symbol.**

The View Properties dialog box appears.

**2 Click the *Query* button.**

The query window opens to the Columns page.

**3 Select *SALE.SALE_AMOUNT* in the Column/Expression list.**



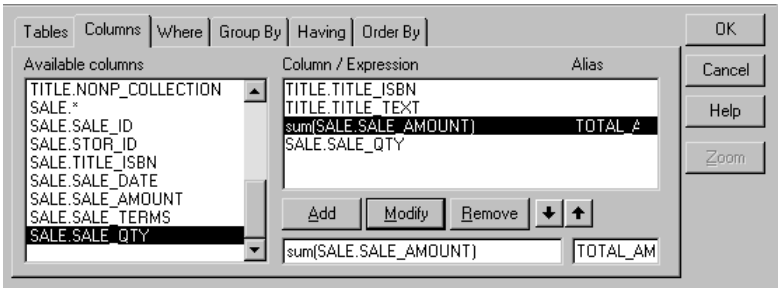

**4 Type *sum(SALE.SALE_AMOUNT)* below the Add button.**

**5 Type *TOTAL_AMT* in the box below the ⬆ button.**

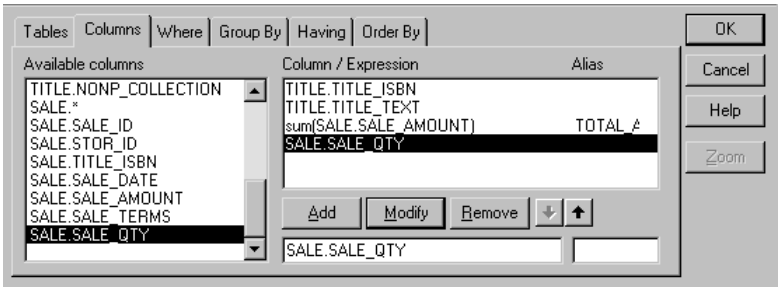
TOTAL_AMT is the alias for the calculated field.

**6 Click the *Modify* button.**

The Column/Expression list displays the calculated column.




**7 Select *SALE.SALE_QTY* in the Column/Expression list.**



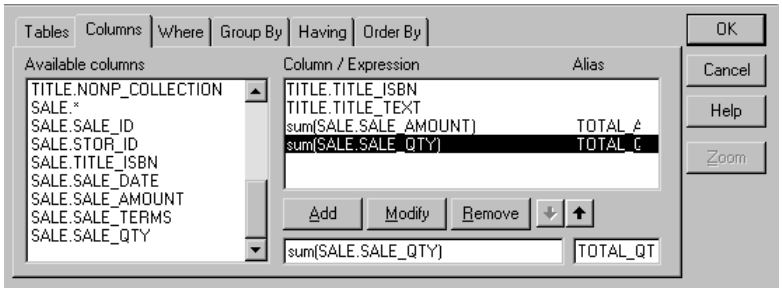

**8 Type *sum(SALE.SALE_QTY)* below the Add button.**

**9 Type *TOTAL_QTY* in the box below the ↑ button.**

TOTAL_QTY is the alias for the calculated field.

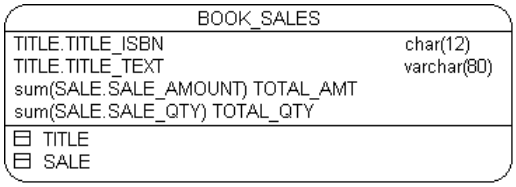
**10 Click the *Modify* button.**

The Column/Expression list displays the calculated column.




**11 Click *OK*.**

You return to the PDM workspace where the view symbol includes the calculated column.




## What you learned

In this section, you learned how to:

- Define a column that contains the sum of values in another column
- Define an alias for a calculated column

# Preview the SQL query

**Where you are**
Chapter 16 Creating a View
Compose the view
Customize the view
Create calculated columns
➡ Preview the SQL query

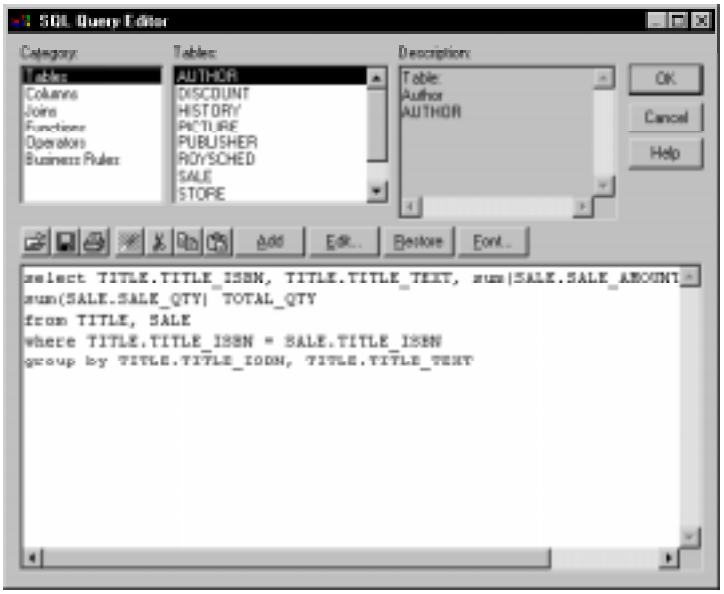
You will preview the SQL query that the view will generate.

**1 Double-click the *view*.**

The View Properties dialog box appears.

**2 Click the *SQL* button.**

The SQL Query Editor window appears. The window shows the SQL query.




**3 Click *OK* in each of the dialog boxes.**

You return to the PDM workspace.

CHAPTER 17

# Defining Extended Attributes

When you generate an application, extended attributes provide you with additional information about objects in the PDM. You can create your own extended attributes.

You will create extended attributes that define an alternate label for a column and a font for a table header. The alternate label can store a coded internal label or a label in a different language.

To save your own extended attribute definitions, you can export the attributes to an EXA file.

In this lesson, you will:

- Create extended attributes
- Modify the value of an extended attribute
- Export extended attributes

**How long will it take?**
About 5 minutes.

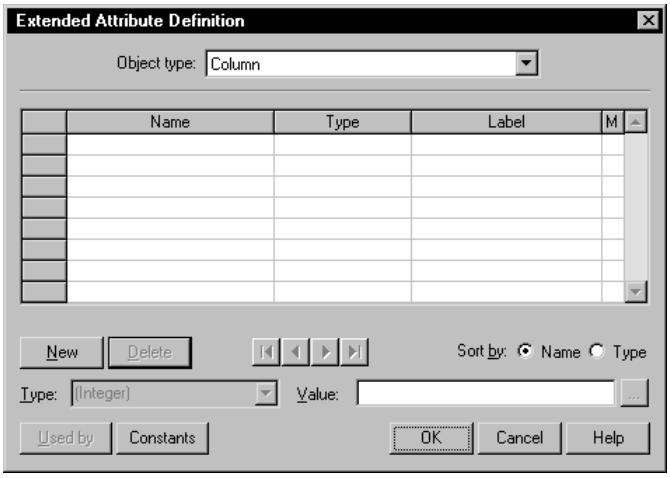
# Create extended attributes

**Where you are**
Chapter 17 Defining Extended Attributes
➡ Create extended attributes
Modify the value of an extended attribute
Export extended attributes

You will create extended attributes that define an alternate label for a column and a font for a table header. The alternate label can store a coded internal label or a label in a different language.

**1 Select** *Dictionary➤Extended Attributes➤List of Attributes*.

The Extended Attribute Definition dialog box appears. The Object Type dropdown listbox indicates that this is the list of attributes for columns.




**2 Type** *Alternate Label* **in the Name column on the first blank line.**

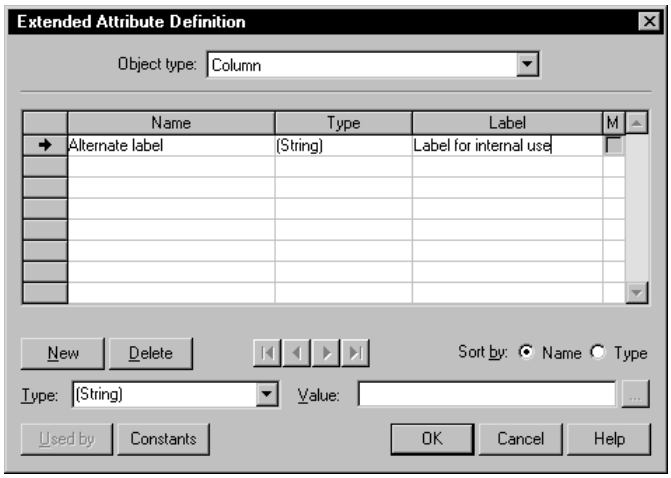
This is the name of the extended attribute.

**3 Select** *String* **from the Type dropdown listbox in the lower part of the dialog box.**

The data type appears in the Data Type column for the extended attribute.

**4 Type *Label for internal use* in the Label column.**

This is a description of the extended attribute. You will leave the Value box blank because you do not want to impose a default label.




**5 Select *Table* from the Object Type dropdown listbox at the top of the dialog box.**

This validates the extended attribute that you defined for columns and displays a blank list of extended attributes for tables.

**6 Type *Header Font* in the Name column of the first blank line.**

This is the name of the extended attribute.

**7 Select *Font* from the Type dropdown listbox.**

The data type appear in the Data Type column for the extended attribute.

**8 Type *Font for table header* in the Label column.**

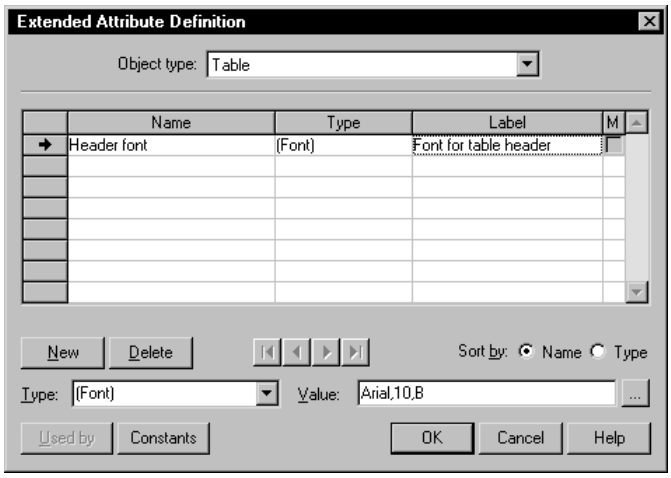
**9 Click the ... at the end of the Value box.**

A standard font selection dialog box appears.

**10** **Select** *Arial,10,Bold*.

**11** **Click** *OK*.

In the Value box, Arial,10,B indicates that the default value from this extended attribute is now Arial 10 point in bold.




**12** **Click** *OK*.

# Modify the value of an extended attribute

**Where you are**
Chapter 17 Defining Extended Attributes
Create extended attributes
➡ Modify the value of an extended attribute
Export extended attributes

For a particular table, you will assign a value to the extended attribute you created. In this case, you will associate a different font size for Title table header.

**1 Double-click the *TITLE* table symbol.**

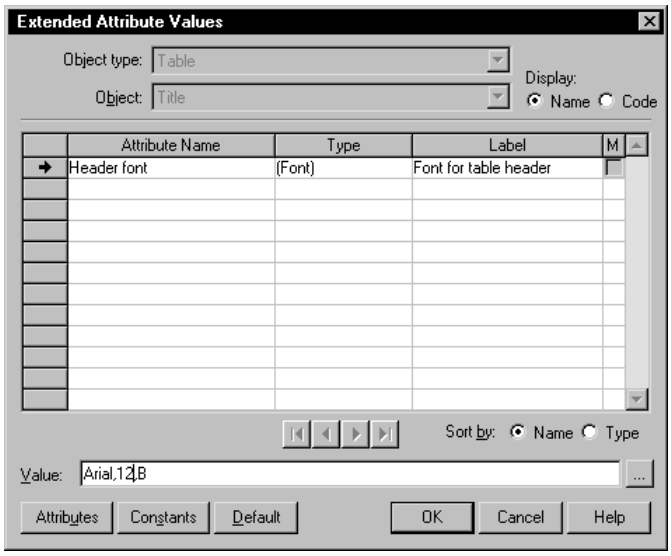
The table property sheet appears.

**2 Click the *Extended* button.**

The list of extended attributes for the table shows the extended attribute Header Font which you just created. The Value box shows the default value.

**3 Type *12* in the place of 10 in the Value box.**

In the Value box, Arial,12,B indicates that the value of this extended attribute for the Title table is now Arial 12 point in bold.




**4 Click *OK* in each of the dialog boxes.**

# Export extended attributes

**Where you are**
Chapter 17 Defining Extended Attributes
Create extended attributes
Modify the value of an extended attribute
➡ Export extended attributes

To save the extended attributes you export them to an EXA file. You will export the new list of extended attributes to a new EXA file.

**1 Select** *Dictionary➤Extended Attributes➤Export Attributes* **from the menu bar.**

A Save As dialog box appears.

**2 Type** *TUTATTRB.EXA***.**

**3 Click** *OK***.**

This saves the extended attributes you defined in a file name TUTATTRB.EXA.

**Importing extended attributes**
You can import the extended attributes into a PDM by selecting Dictionary➤Extended Attributes➤Import Attributes and selecting an EXA file.

CHAPTER 18

# Using Triggers and Procedures

## About triggers

A **trigger** is a procedure that goes into effect when you insert, delete, or update a specified table or column. Triggers enforce referential integrity, for example, by displaying an error message if you try to update a primary key column which has an update restriction.

A PDM can automatically generate certain referential integrity triggers depending on the type of the target database. You can also define additional user-defined triggers.

You can generate a SQL script containing the triggers, or you can generate the triggers directly in the database.

## About trigger templates

A trigger script is based on a template. There is a different template for each type of trigger: insert, delete, and update. You can create your own trigger templates or customize an existing template, for example by changing the type of error message displayed.

A trigger template contains variables for tables, columns, and so on. Before you generate the script, you can preview the trigger to see the script with the variables replaced by actual names.

In this lesson you will:

- Preview the trigger
- Create a stored procedure
- Generate a script for triggers and procedures

**How long will it take?**
About 20 minutes.

# Preview a trigger

**Where you are**
Chapter 18 Using Triggers and Procedures
➡ Preview a trigger
Create a stored procedure
Generate a script for triggers and procedures

In a previous lesson you defined cascade referential integrity for deletion in the Sales table. You will now see how this referential integrity can be interpreted as a trigger.

**1 Double-click the *STORE* table symbol.**

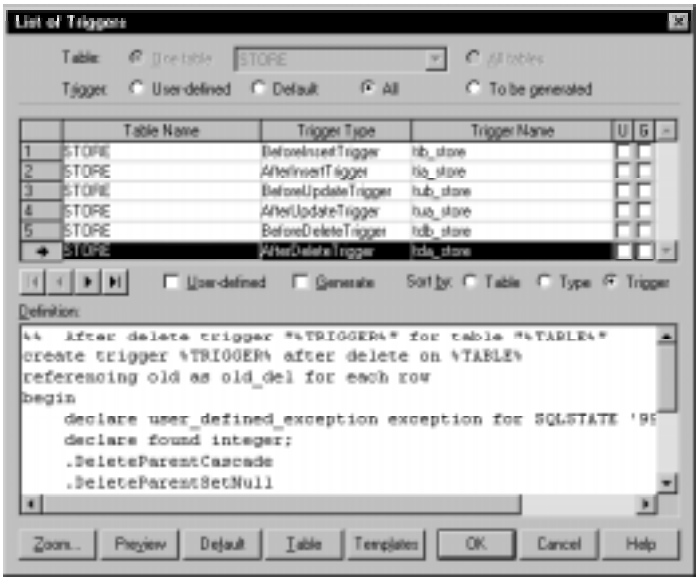
The table property sheet appears.

**2 Click the *Trigger* button.**

The List of Triggers dialog box displays triggers defined for the table.

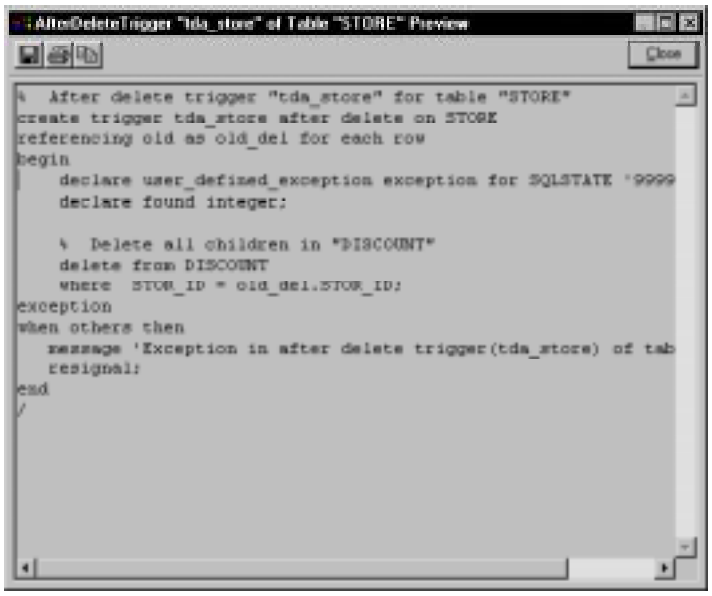
**3 Click the *AfterDeleteTrigger*.**

An arrow appears at the beginning of the line and the trigger script appears in the bottom of the dialog box. It is not instanciated for this table.



**4 Click the *Preview* button.**

The trigger that will be generated appears. It indicates that a deletion in the STORE table cascades to deletion in the DISCOUNT table.




**5 Click *Close*.**

**6 Click *OK* in each of the dialog boxes.**

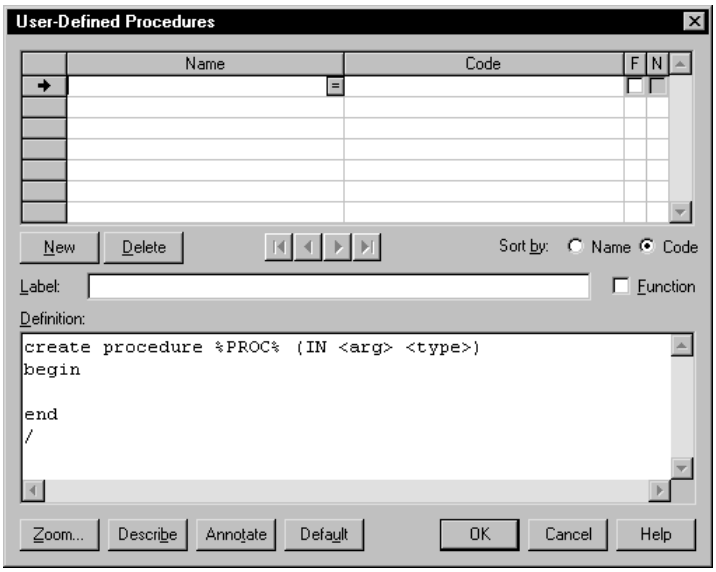
# Create a stored procedure

**Where you are**
Chapter 18 Using Triggers and Procedures
Preview a trigger
➡ Create a stored procedure
Generate a script for triggers and procedures

You will create a stored procedure that updates the sales statistics in the HISTORY table.

**1 Select** *Dictionary➢Triggers and Procedures➢List of Procedures*.

The List of Procedures dialog box contains an empty list.



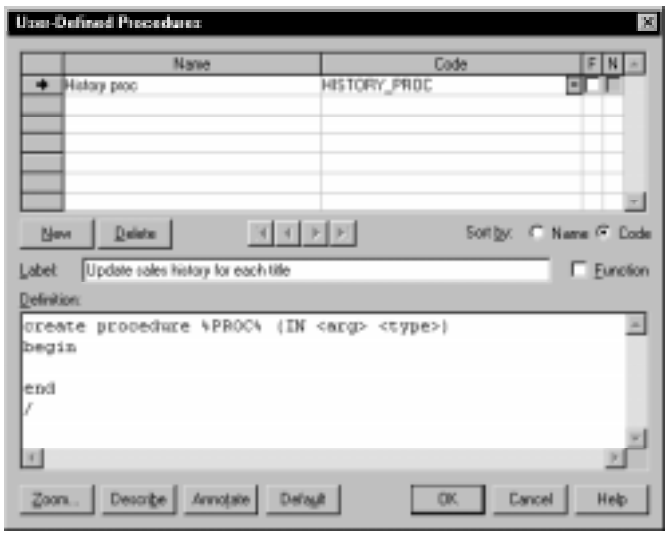

**2 Type** *History Proc* **in the Name column of the first blank line.**

This is the name of the procedure.

**3 Click the** *Code* **column.**
**Click the = button that appears in the Code column.**

This sets the code equal to the name, with lowercase characters replaced by uppercase, and the space replaced by an underscore.

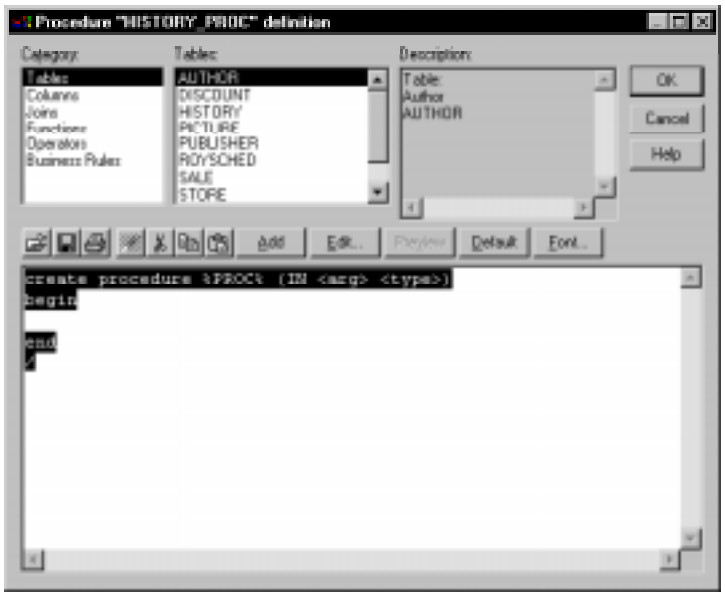
**4 Type** *Update sales history for each title* **in the Label box.**




**5 Click the** *Zoom* **button.**

The Procedure Definition window appears.

**6 Select the** *text* **in the Definition textbox.**

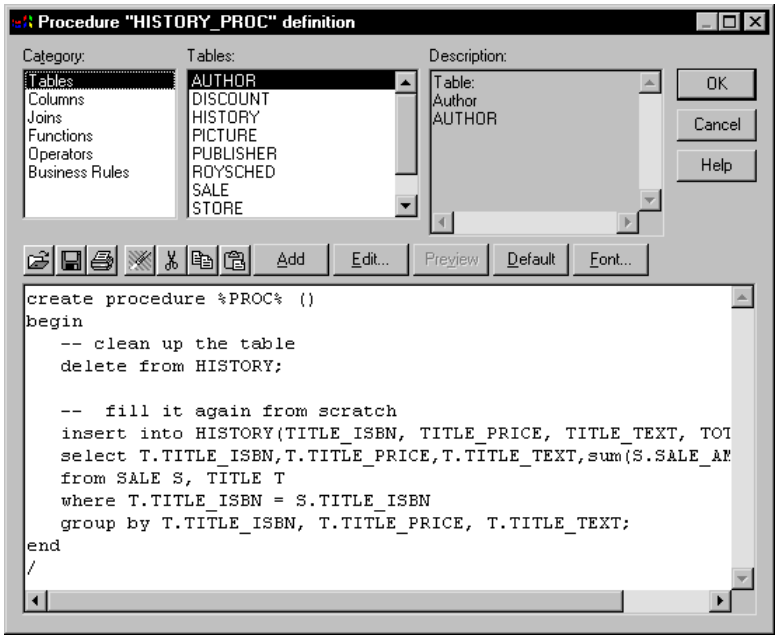


**7 Click the *File* tool.**

A File Open dialog box appears.

**8 Select *TUT_PROC.TXT* from the EXAMPLES directory. Click *OK*.**

The text of TUT_PROC.TXT fills the Definition textbox.




**9 Click *OK* in each of the dialog boxes.**

## What you learned

In this section, you learned how to:

- Create a stored procedure to update a table
- Use the contents of a file for a stored procedure

# Generate a script for triggers and procedures

**Where you are**
Chapter 18 Using Triggers and Procedures
Preview a trigger
Create a stored procedure
➡ Generate a script for triggers and procedures

You will define parameters to generate the script for the triggers and procedures you defined.

**1 Select** *Database➢Generate Triggers and Procedures* **from the menu bar.**

A dialog box containing the generation parameters appears. It proposes a default name and directory for the trigger script.

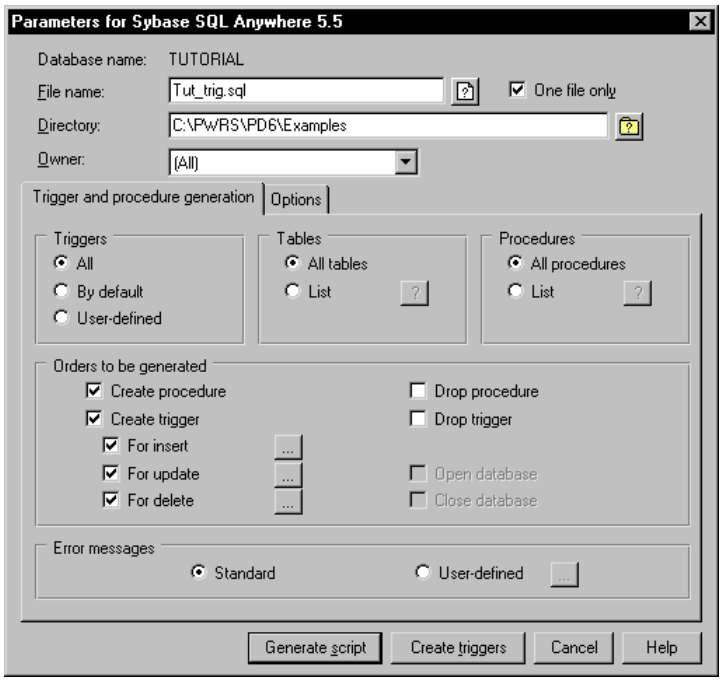
**2 Type** *TUT_TRIG.SQL* **in the File Name box.**

**3 Type** *Examples* **in the Directory box.**

4 **Select the** *generation parameters* **shown below.**

These parameters generate all triggers and procedures.

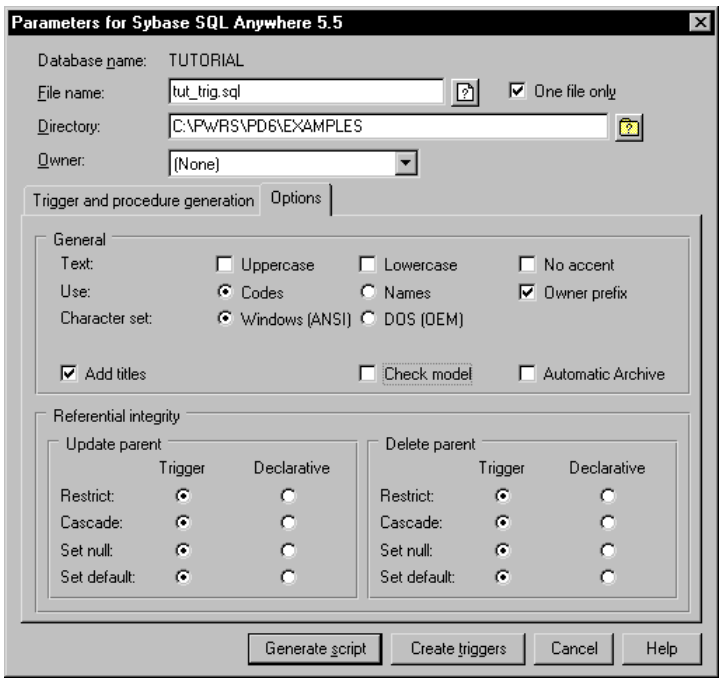
For a complete description of generation parameters, see *PowerDesigner DataArchitect User's Guide*.




5 **Click the** *Options* **tab.**

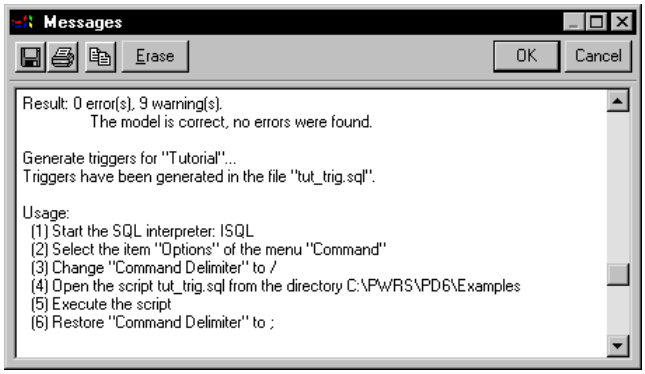
**6 Select the *options* shown below.**

In the Referential Integrity groupbox, you select the Trigger radio buttons. This indicates that you want to generate referential integrity triggers, as opposed to declarative referential integrity.

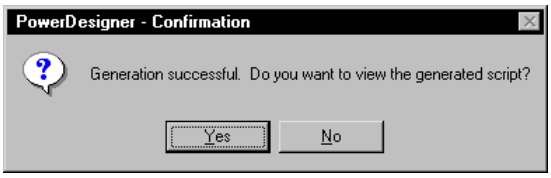



**7 Click the *Generate Script* button.**

The Messages window displays the generation messages.

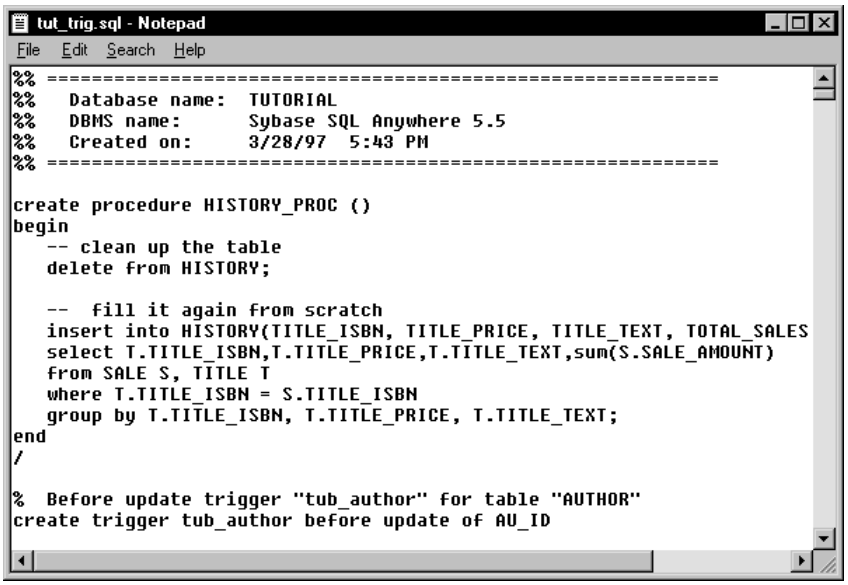


A Confirmation box asks you if you want to view the script.




**8 Click** *Yes*.

A window displays the generated script. The script starts with the HISTORY_PROC stored procedure that you defined.




**9 Select** *File*➤*Exit* **to close the script window.**

**10 Click** *OK* **to close the Messages window.**

## What you learned

In this section, you learned how to:

- Select options that will generate triggers
- Generate a script for triggers and procedures

CHAPTER 19

# Generating a Database Script

You can generate a database directly from a PDM, or you can generate a database script which you can run in your DBMS environment. You create a script for a particular target database.

The generation parameters which are available depend on the target database you select. By default, the target database is the one you select when you open the PDM, but you can select another before generating the script.

In this chapter you will:

- Attach an expression to a business rule
- View a validation rule
- Generate a database creation script

**How long will it take?**
About 5 minutes.

# Attach an expression to a business rule

**Where you are**
Chapter 19 Generating a Database Script
➡ Attach an expression to a business rule
View a validation rule
Generate a database creation script

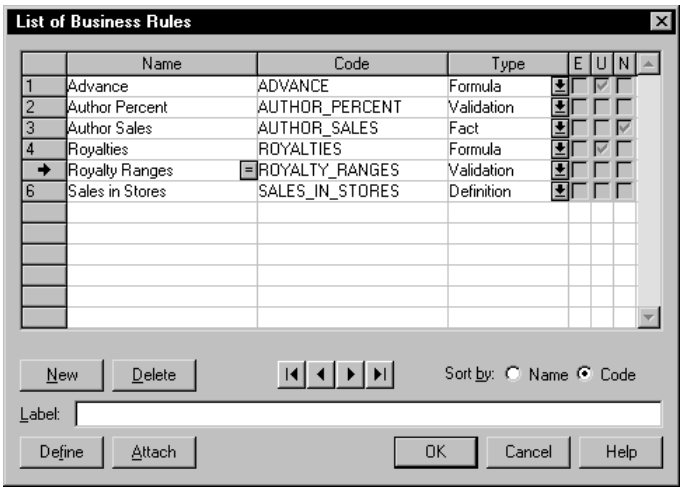
You will create a validation rule that controls values of columns in the ROYALTY table. You will do this by attaching a server expression to a business rule that is already attached to a table.

**1 Select** *Dictionary➢List of Business Rules*.

The List of Business Rules appears.

**2 Click the** *Royalty Ranges* **line in the list.**

An arrow appears at the beginning of the line.

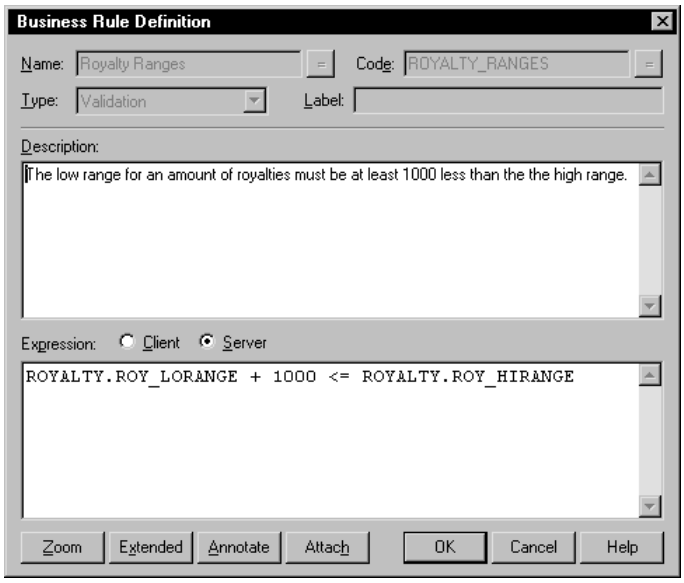


**3 Click the *Define* button.**

The definition of the business rule appears. The description appears in the upper part of the dialog box. The Server radio button is selected, but the expression pane is empty.

**4 Type the following SQL command in the Expression pane.**

```
ROYALTY.ROY_LORANGE + 1000 <= ROYALTY.ROY_HIRANGE
```




**5 Click *OK* in each of the dialog boxes.**

# View a validation rule

**Where you are**
Chapter 19 Generating a Database Script
Attach an expression to a business rule
➡ View a validation rule
Generate a database creation script

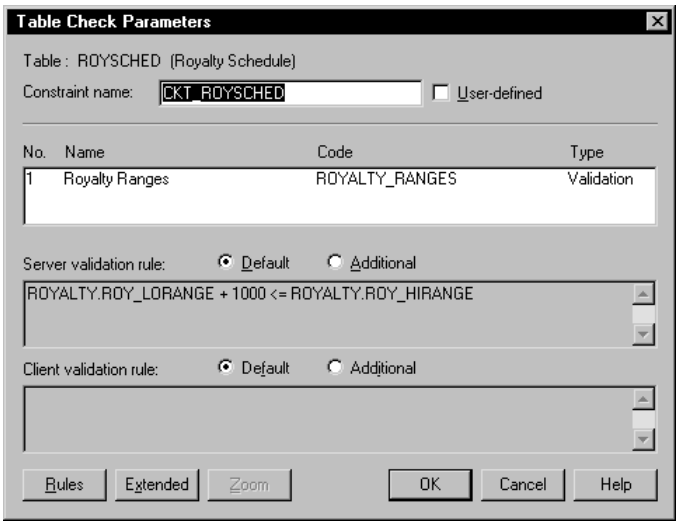
You will view the validation rule that you just created as it relates to the table ROYSCHED.

**1 Double-click the *ROYSCHED* table symbol.**

The table property sheet appears.

**2 Click the *Check* button.**

The table check parameters dialog box appears. The validation rule that you defined appears in the server validation rule pane.




**3 Click *OK* in each of the dialog boxes.**

# Generate a database creation script

**Where you are**
Chapter 19 Generating a Database Script
Attach an expression to a business rule
View a validation rule
➡ Generate a database creation script

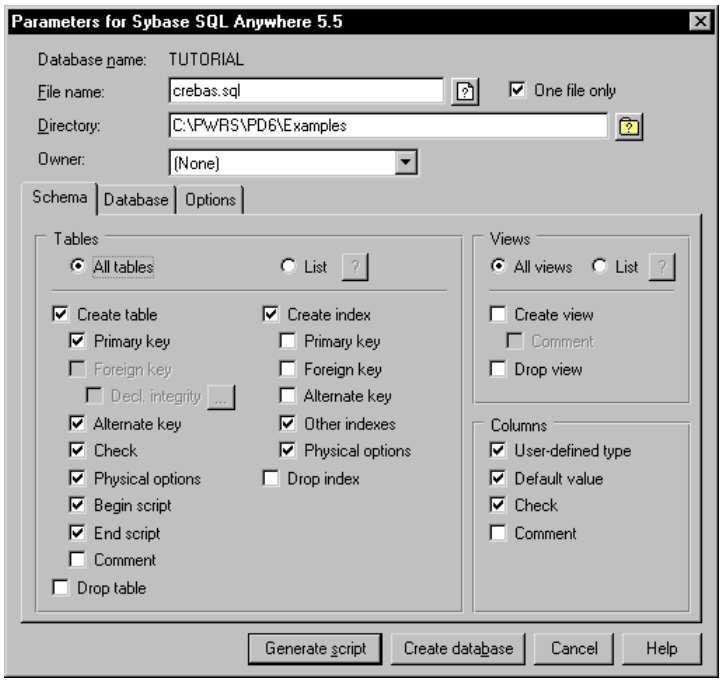
**1 Select** *Database*➤*Generate Database*.

The Parameters for Sybase SQL Anywhere dialog box appears. It displays the generation parameters. Certain parameters are already selected.

**2 Type** *TUTORIAL.SQL* **in the File Name box.**

**3 Type** *EXAMPLES* **in the Directory box.**

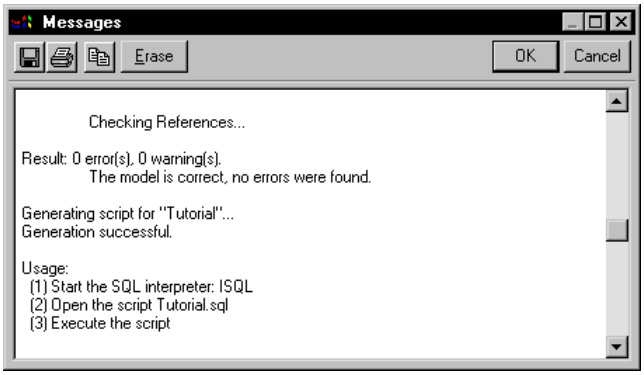
**4 Select the** *generation parameters* **shown below.**



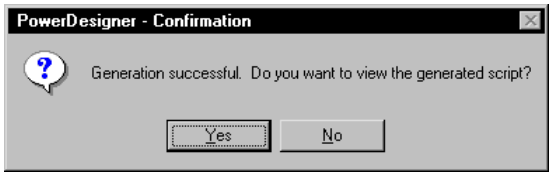
For a complete description of database generation parameters, see *PowerDesigner DataArchitect User's Guide*.

**5 Click the *Generate Script* button.**

The Messages window displays the generation messages.

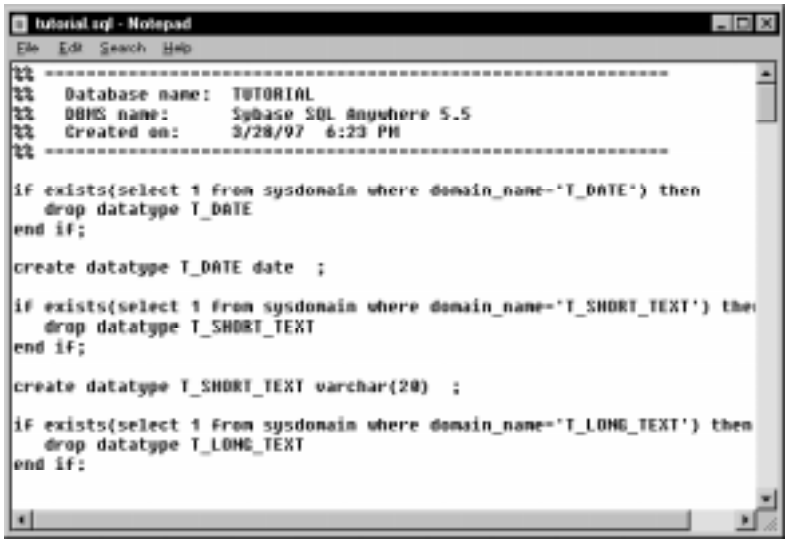



A Confirmation box confirms generation of the script and asks if you want to view the script.



6 **Click** *Yes*.

A window displays the generated script.




7 **Select** *File*➤*Exit* **to close the script window.**

8 **Click** *OK* **to close the Messages window.**

CHAPTER 20

# Generating a Test Data Script

## About test data

**Test data** is sample data that you can define and generate for one or more tables in a PDM. When you generate test data, it automatically generates rows of data in database tables.

Test data is normally used as a development tool in designing data bases.

You can use test data to verify the performance of the database when it is filled with large amounts of data, or when it is accessed by different applications or users.

You can generate test data directly from a PDM, or you can generate a test data script which you can run in your database environment.

## About data profiles

You generate test data for a table based on data profiles. A **data profile** is a named class of data types that has a defined data generation source.

You assign a data profile to one or more columns. Test data is generated for the columns using the data source defined for each data profile.

In this chapter you will:

- Create data profiles
- Define an automatic test data generation source for selected data profiles
- Define a file containing data as a test data generation source for a data profile
- Assign data profiles to relevant columns in a selected table
- Generate a test data script for the table
- Exit DataArchitect

**How long will it take?**
About 15 minutes.

# Create data profiles

**Where you are**
Chapter 20 Generating a Test Data Script
➡ Create data profiles
Define an automatic test data generation source
Define a file as a test data generation source
Assign data profiles for a selected table
Generate a test data creation script
Exit DataArchitect

You will create two data profiles called Sales and Price, which both have the class Number. These data profiles will represent columns that have number data types such as numeric or integer.

You will then create another data profile called Long_Text, which will have the class Character. This data profile can represent any column that has a character data type.

**1 Select** *Dictionary➤Profiles➤List of Profiles* **from the menu bar.**

The List of profiles appears. By default an arrow appears at the beginning of the first line.

**2 Type** *Sales* **in the first line of the Name column.**

This is the name of the data profile.

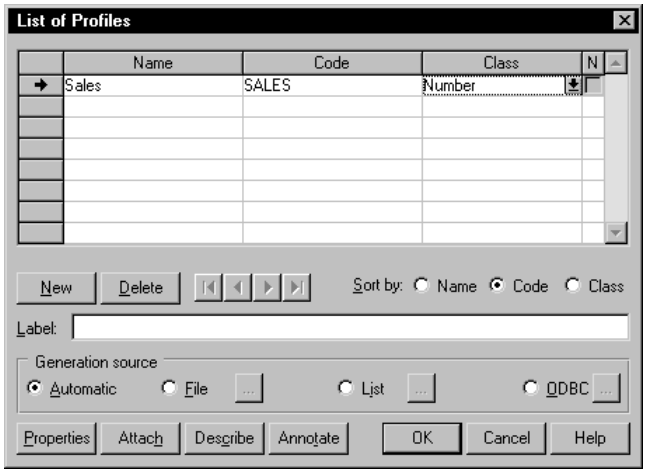
**3 Click the *Code* column.**
**Click the = button that appears in the Code column.**

This sets the code equal to the name.

**4 Click the *Class* column.**
**Click the ... button that appears in the Class column.**

The Class dropdown list appears.

**5 Select *Number* from the dropdown list in the Class column.**




This defines the Number class for the data profile Sales.

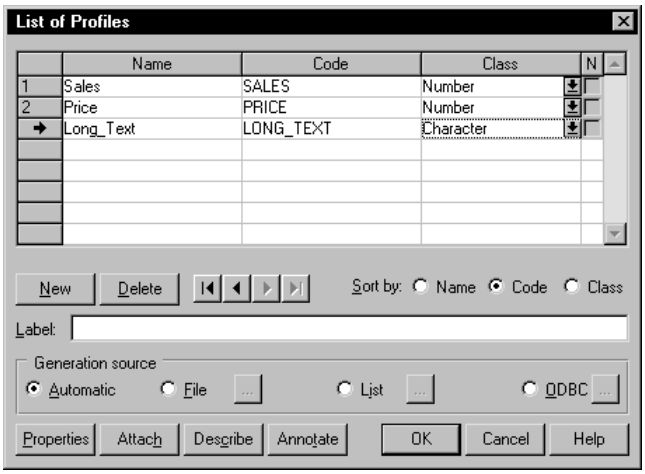
**6 Click the *New* button.**

An arrow appears at the beginning of the first blank line in the list.

**7 Repeat steps 2 to 6 to create the following data profiles with their corresponding classes:**

| Data profile | Class |
|---|---|
| Price | Number |
| Long_Text | Character |

The List of Profiles should appear as shown below:




**8** **Click** *OK*.

# Define an automatic test data generation source

**Where you are**
Chapter 20 Generating a Test Data Script
Create data profiles
➡ Define an automatic test data generation source
Define a file as a test data generation source
Assign data profiles for a selected table
Generate a test data creation script
Exit DataArchitect

To generate test data you need to define a data source for each data profile. PowerDesigner can generate test data automatically, using generation parameters that are defined for each data profile.

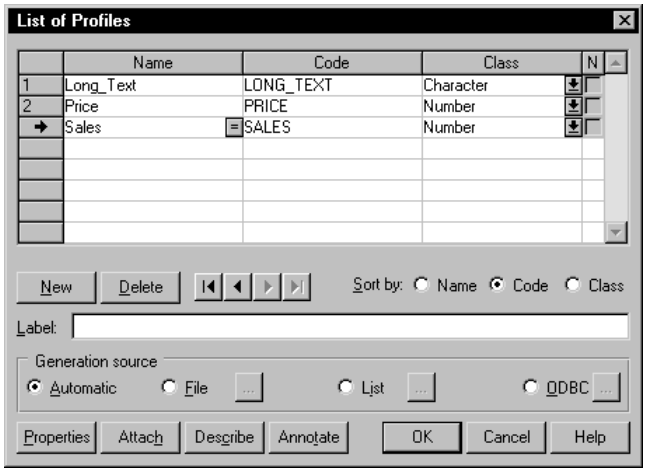
You will define automatic test data generation sources for the data profiles Sales and Price.

**1 Select *Dictionary➤Profiles➤List of Profiles* from the menu bar.**

The List of profiles appears.

**2 Click the *Sales* line in the list.**

An arrow appears at the beginning of the line.



**3 Select the *Automatic* radio button from the Generation Source group box.**

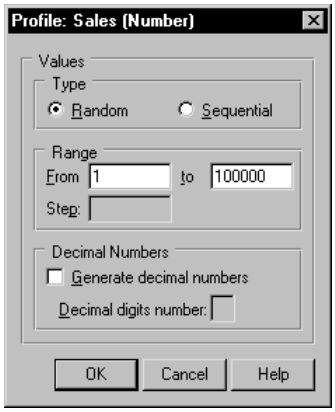
**4 Click the *Properties* button.**

The profile Number dialog box appears.

**5 Select the *Random* radio button.**

**6 Type *1* in the From textbox and *100000* in the To textbox.**

You define a range of between 1 and 100,000 for the random test numbers that will be generated for the data profile Sales.




**7 Click OK.**

You return to the List of Profiles.

**8 Click the *Price* line.**

An arrow appears at the beginning of the line.

**9 Select the *Automatic* radio button from the Generation Source group box.**

**10 Click the *Properties* button.**

The profile Number dialog box appears.

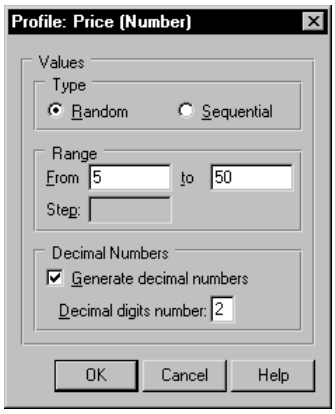
**11 Select the *Random* radio button.**

**12 Type *5* in the From textbox and *50* in the To textbox.**

**13 Select the *Generate decimal Numbers* checkbox.**
**Click the Decimal Digits Number textbox.**
**Type the number *2* in the Decimal Digits Number textbox.**

You define a range of between 5.00 and 50.00 for the random test numbers that will be generated for the data profile Price.

The Profile Number dialog box should appear as shown below.




**14 Click OK.**

You return to the List of profiles.

**15 Click OK.**

# Define a file as a test data generation source

**Where you are**
Chapter 20 Generating a Test Data Script
Create data profiles
Define an automatic test data generation source
➡ Define a file as a test data generation source
Assign data profiles for a selected table
Generate a test data creation script
Exit DataArchitect

You can specify a CSV format file which contains data, and use the data as a test data generation source for a data profile.

You will define a CSV file as a test data source for the data profile Long_Text. The directory TESTDATA in the PowerDesigner path contains a number of CSV files that can be used as test data generation source files.

**1 Select *Dictionary➢Profiles➢List of Profiles* from the menu bar.**

The List of profiles appears.

**2 Click the *Long_Text* line in the list.**

An arrow appears at the beginning of the line.

**3 Select the *File* radio button, from the generation Source group box.**

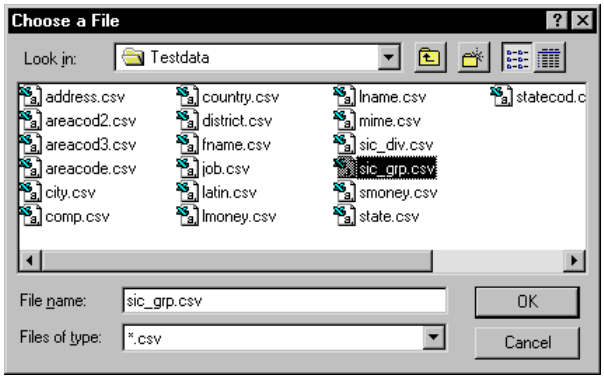
**4 Click the ... button that is next to the *File* radio button.**

A standard file selection box appears.

**5 Select the directory *TESTDATA* in the PowerDesigner path.**

**6 Select the file** *SIC_GRP.CSV*

This file contains data which classifies a number of businesses and industries into categories based on their type of commercial activity. This type of data, which is both general and structured, is a useful data source for columns that require a character string of meaningful data with a limited length.




**7 Click OK.**

You return to the List of Profiles.

**8 Click OK.**

# Assign data profiles for a selected table

**Where you are**
Chapter 20 Generating a Test Data Script
Create data profiles
Define an automatic test data generation source
Define a file containing data as a test data generation source
➡ Assign data profiles for a selected table
Generate a test data creation script
Exit DataArchitect

You will assign a data profile to each column for the table History in the PDM model. For this tutorial you will assign data profiles from the Table Properties sheet. However, you can also assign data profiles from the List of domains and List of columns.

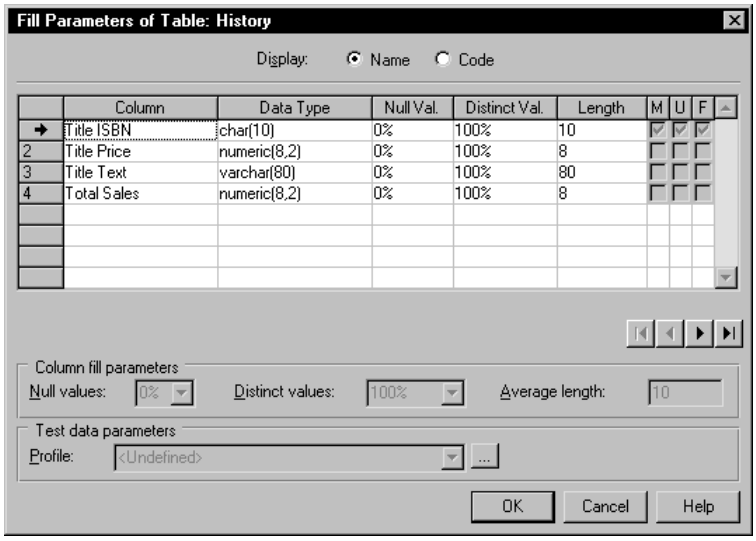
For information on selecting a data profile for a domain, see the *PowerDesigner DataArchitect User's Guide*.

**1 Double-click the *History* table symbol.**

The Table Properties dialog box appears.

**2 Click the *Fill* button.**

The Fill Parameters dialog box appears. Title ISBN is the first line in the list. It is selected by default. The Profile dropdown list is greyed because Title ISBN is a foreign key. You can not assign a data profile to a foreign key column. It automatically takes the data profile of the primary key in the parent table.

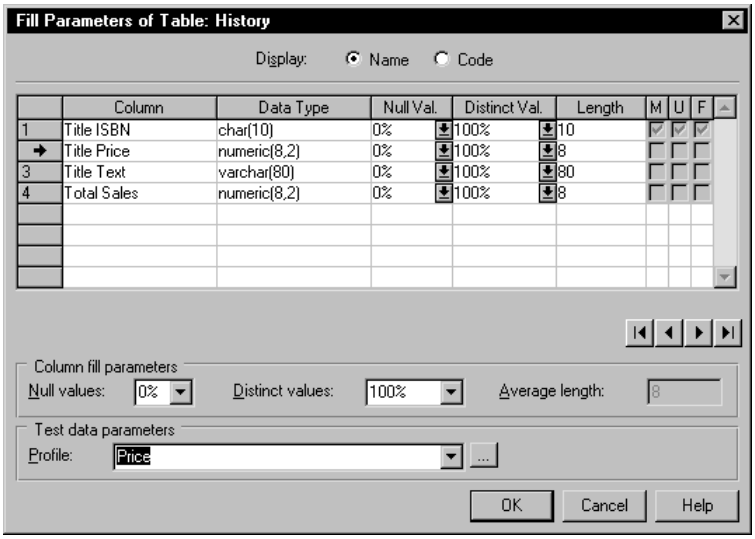



**3 Click the *Title Price* line in the list.**

An arrow appears at the beginning of the line. Title Price has the data type NUMERIC(8,2). Based on this data type, and the sort of data that you want to generate for the column, you will assign the data profile Price to this column.

**4 Select *Price* from the profile dropdown list.**

The data profile Price is assigned to the column Title Price.




**5 Click the *Title Text* line in the list.**

An arrow appears at the beginning of the line. Title Text has the data type VARCHAR(80). Based on this data type, and the sort of data that you want to generate for the column, you will assign the data profile Long_Text to this column.

**6 Select *Long_Text* from the *Profile* dropdown list.**

The data profile Long_Text is assigned to the column Title Text.

7 **Select the column** *Total Sales*.

An arrow appears at the beginning of the line. The column Total Sales has the data type NUMERIC(8,2). Based on this data type and the sort of data that you want to generate for this column, you will assign the data profile Sales to this column.

8 **Select** *Sales* **from the profile dropdown list.**

9 **Click** *OK*.

You return to the Table Properties dialog box.

10 **Click** *OK.*

# Generate a test data creation script

**Where you are**
Chapter 20 Generating a Test Data Script
Create data profiles
Define an automatic test data generation source
Define a file as a test data generation source
Assign data profiles for a selected table
➡ Generate a test data creation script
Exit DataArchitect

You will generate a test data script for the table History. You can execute a test data script in a database that contains the same tables that you use to generate the script.

**1 Select** *Database➢Generate Test Data***.**

The Test Data Generation dialog box opens to the Parameters page. It displays the test data generation parameters. Certain parameters are already selected.

**2 Type** *INSERT.SQL* **in the File Name box.**

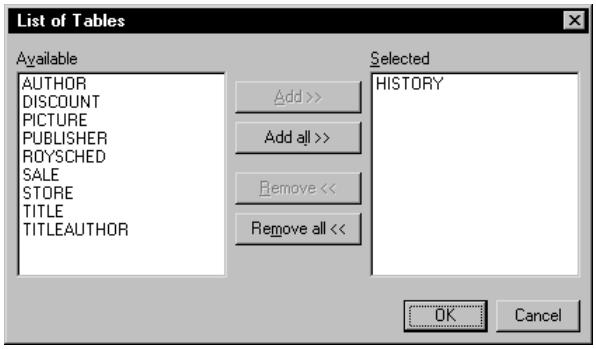
**3 In the** *Directory* **box, specify the** *EXAMPLES* **directory in the PowerDesigner path.**

**4 Select the** *List* **radio button and click the**  **button.**

A List of Tables appears.

**5 From the Available pane, select *History* and click the *Add* button.**

History appears in the Selected pane.




**6 Click *OK*.**

You return to the Parameters page of the Test Data Generation dialog box.

**7 Select the *Delete Old Data* Checkbox.**

All the existing data in the table you are using to generate test data will be deleted and replaced by the test data.

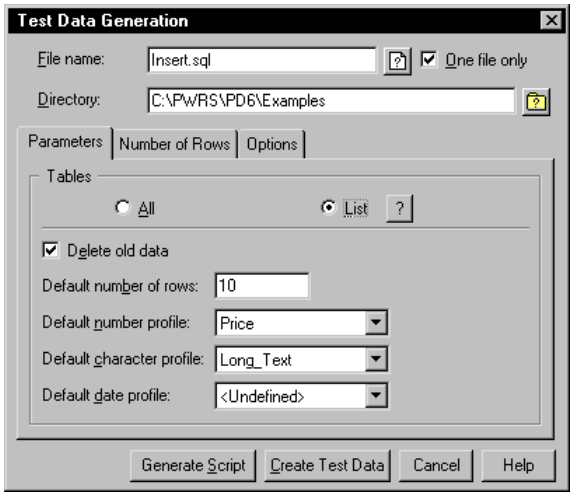
> **Default Number of Rows**
> The default number of rows is the number of rows of test data that will be generated for tables, that do not have a value defined in the Number of Rows page of the Test Data Generation dialog box. You will use the default value 10, which means you will generate ten rows of test data for the table History.

**8 Select *Price* from the *Default Number Profile* dropdown list.**
**Select *Long_Text* from the *Default Character Profile* dropdown list.**

The default data profiles are the profiles that will be automatically assigned to any column with the appropriate data type, that does not have a data profile already assigned to it.

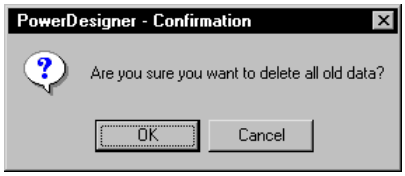
You do not need to change the <Undefined> option for the Date profile, because the table History does not have a column that takes a Date data profile.

The Parameters page for the Test Data Generation dialog box, should appear as shown below.




**9 Click the *Generate Script* button.**

A Confirmation box asks you if you want to delete all old data.



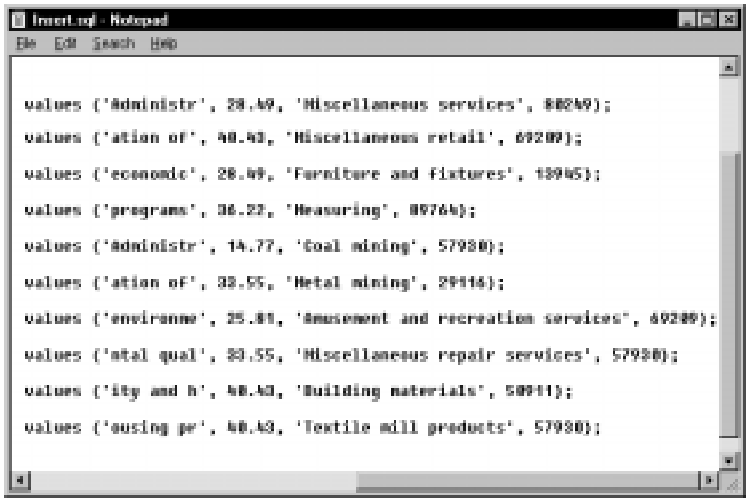
**10 Click** *OK*.

A confirmation box confirms generation of the script and asks if you want to view the script.

**11 Click** *Yes*.

A window displays the generated script. The title of each column in the table History appears after the insert command.

**12 Scroll to the right to view the generated test data.**




**13 Select** *File*➢*Exit* **to close the script window.**

**14 Click** *OK* **to close the Messages window.**

**15 Click** *Cancel* **to close the Test Data Generation dialog box.**

# Exit DataArchitect

**Where you are**
Chapter 20 Generating a Test Data Script
Create data profiles
Define an automatic test data generation source
Define a file as a test data generation source
Assign data profiles for a selected table
Generate a test data creation script
➡ Exit DataArchitect

You will close the PDM and exit DataArchitect.

**1 Select *File*➤*Save* from the menu bar.**

This saves the PDM.

**2 Select *File*➤*Close* from the menu bar.**

This closes the PDM window.

**3 Select *File*➤*Exit* from the menu bar.**

You exit the DataArchitect application.

# Glossary

| | |
|---|---|
| **alternate key** | Column or columns whose values uniquely identify a row in the table and are not primary key columns |
| **business rule** | Written statement specifying what the information system must do or how it must be structured to support business needs |
| **clustered index** | Index in which the physical order and the logical (indexed) order is the same |
| **column** | Data structure that contains an individual data item within a row (record), model equivalent of a database field |
| **Conceptual Data Model (CDM)** | Entity-relationship diagram that models the information system without considering the details of physical implementation |
| **constraint** | Named check that enforces data requirements, default values, or referential integrity on a table or a column |
| **data item** | Elementary piece of information |
| **data source** | Identification of the data to access, its operating system, DBMS, and network platform |
| **data profile** | Named class of data types that has a defined data generation source, and can be assigned to one or more columns. |
| **domain** | Set of values for which a data item is valid |
| **entity** | Person, place, thing, or concept that has characteristics of interest to the enterprise and about which you want to store information |

| | |
|---|---|
| **entity attribute** | Elementary piece of information attached to an entity |
| **extended attribute** | Additional information that completes the definition of an object for documentary purposes or for use by an external application such as a fourth-generation language (4GL) |
| **foreign key** | Column or columns whose values depend on and migrate from a primary key in another table |
| **4GL** | External application that uses a fourth-generation language, usually to generate a client/server application |
| **index** | Data structure that is based on a key and that speeds access to data and controls unique values |
| **Inheritance** | Special relationship that defines an entity as a special case of a more general entity |
| **ODBC** | Open Database Connectivity (ODBC) interface which gives PowerDesigner access to data in database management systems (DBMS) |
| **ODBC driver** | Part of the Open Database Connectivity (ODBC) interface that processes ODBC functions calls, submits SQL requests to a specific data source, and returns results to the application |
| **Physical Data Model (PDM)** | Table-reference diagram that models the information system including the details of physical implementation |
| **primary key** | Column or columns whose values uniquely identify a row in a table |
| **property sheet** | Window that displays the properties of an object |
| **reference** | Link between the primary key and the foreign key of different tables |
| **referential integrity** | Rules governing data consistency, specifically the relationships among primary keys and foreign keys of different tables |

| | |
|---|---|
| **relationship** | Named connection or association between entities |
| **storage** | Named partition that stores tables and indexes on a storage device |
| **table** | Collection of rows (records) that have associated columns (fields) |
| **tablespace** | Named partition that stores tables and indexes in a database |
| **trigger** | Special form of stored procedure that goes into effect when you insert, delete, or update a specified table or column |
| **unique index** | Index in which no two rows can have the same index value, including NULL |
| **view** | Alternate way of looking at the data in one or more tables. Usually created as a subset of columns from one or more tables |

# Index

## D



## E

## F



## G



## I



## K



## L



## M



## N



## O

## P



## Q



## R



## S

## T



## V



## W