

Môn học:

Công cụ Thiết Kế Hệ Thống Thông Tin

Mã MH: **ITEC3407**

Số TC: 3 (**2LT, 1TH**)

Khoa CNTT, ĐH Mở tp. HCM

CHƯƠNG 4: Mô hình VẬT LÝ DỮ LIỆU

- ◆ 4.1. Cơ bản về mô hình VẬT LÝ DỮ LIỆU
- ◆ 4.2. Sử dụng các quy tắc trong mô hình VẬT LÝ DL
- ◆ 4.3. Xây dựng biểu đồ VẬT LÝ DỮ LIỆU
- ◆ 4.4. Xây dựng biểu đồ VẬT LÝ DỮ LIỆU đa chiều
- ◆ 4.5. Làm việc với mô hình VẬT LÝ DỮ LIỆU
- ◆ 4.6. Các mô hình được sinh ra từ MH VẬT LÝ DL
- ◆ 4.7. Tổng kết chương & Bài tập

4.1. Cơ bản về mô hình VẬT LÝ DỮ LIỆU

4.1.1: Khái niệm về MH quan niệm dữ liệu (PDM)

4.1.2. Môi trường của PDM

4.1.3. Định nghĩa PDM

4.1.4. Định nghĩa các gói (package) trong PDM

4.1.5. Sử dụng công cụ SQL

4.1.1: Khái niệm về MH vật lý dữ liệu (1)

What is a PDM?

The PDM is a database design tool for defining the implementation of physical structures and data queries.

Depending on the type of database you want to design, you will use different types of diagrams in the PDM.

Database	Diagram
Operational	Physical diagram to define the physical implementation of the database
Date warehouse or Data mart	Physical diagram to store business data
OLAP	Multidimensional diagram to define the possible queries to perform on the operational data

4.1.1: Khái niệm về MH vật lý dữ liệu (2)

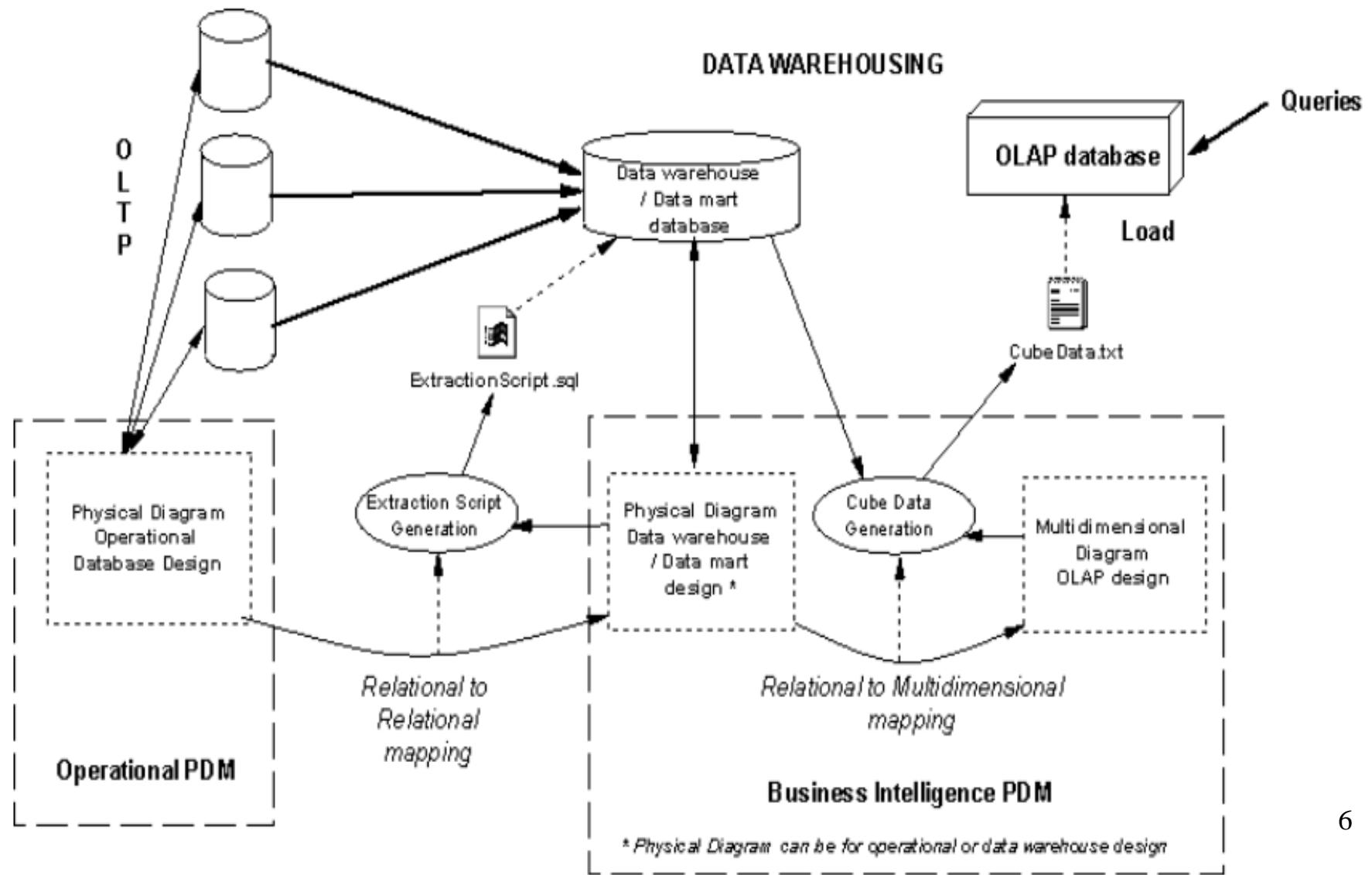
What is a PDM?

The PDM is a database design tool for defining the implementation of physical structures and data queries.

Depending on the type of database you want to design, you will use different types of diagrams in the PDM.

Database	Diagram
Operational	Physical diagram to define the physical implementation of the database
Date warehouse or Data mart	Physical diagram to store business data
OLAP	Multidimensional diagram to define the possible queries to perform on the operational data

4.1.1: Khái niệm về MH vật lý dữ liệu (3)



4.1.1: Khái niệm về MH vật lý dữ liệu (4)

Operational PDM

The physical diagrams of the PDM are used to design the structure of a database handling large amounts of operational data. Usually, the physical analysis follows the conceptual analysis in data modeling. At the physical stage, the designer considers the details of actual physical implementation of data in a database.

Business Intelligence PDM

You can use a PDM to design the structure of a data warehouse handling business data. A data warehouse environment is made of:

- ◆ A data warehouse or data mart database
- ◆ An OLAP database and tools for sending queries to the database

4.1.1: Khái niệm về MH vật lý dữ liệu (5)

Data warehouse
and data mart
database

Data warehouse and data mart databases store historical business data from heterogeneous systems. The data warehouse database gathers all the data manipulated in a company for example, whereas the data mart focuses on smaller entities in the company.

OLAP database

An **OLAP database** shows a multidimensional view of business data. Information in an OLAP database is organized to facilitate the queries performed by different tools. Business analysts use OLAP databases to send queries and retrieve business information from the different dimensions existing in the database.

4.1.1: Khái niệm về MH vật lý dữ liệu (6)

Logical model

The logical model allows you to design the database structure and perform some database denormalization actions. This model is independent from any DBMS physical requirement.

In PowerDesigner, you can design a logical model using a PDM with the <Logical Model> DBMS. This PDM is a physical model with standard objects, and without DBMS specific physical options and generation capabilities.

4.1.1: Khái niệm về MH vật lý dữ liệu (7)

Objects in a PDM

The PDM includes different types of diagrams, with their objects:

Physical objects	Object	Tool	Description
	Table		Collection of rows (records) that have associated columns (fields)
	Column	—	Data structure that contains an individual data item within a row (record), model equivalent of a database field.
	Key	—	Column or columns whose values uniquely identify each row in a table. A key can be designated as a primary or alternate key
	Primary key	—	Column or columns whose values uniquely identify each row in a table, and are designated as the primary identifier of each row in the table
	Alternate key	—	Column or columns whose values uniquely identify each row in a table, and which is not a primary key
	Foreign key	—	Column or columns whose values depend on and migrate from a primary or alternate key in another table.
	Index	—	Data structure associated with one or more columns in a table, in which the column values are ordered in such a way as to speed up access to data

4.1.1: Khái niệm về MH vật lý dữ liệu (8)

Reference		Link between a primary or an alternate key in a parent table, and a foreign key of a child table. Depending on its selected properties, a reference can also link columns that are independent of primary or alternate key columns
View		Data structure that results from a SQL query and that is built from data in one or more tables
Procedure		Precompiled collection of SQL statements stored under a name in the database and processed as a unit
Web service	—	Collection of SQL statements stored in a database to retrieve relational data in HTML, XML, WSDL or plain text format, through HTTP or SOAP requests
Web operation	—	Sub-object of a Web service containing a SQL statement and displaying Web parameters and result columns

4.1.1: Khái niệm về MH vật lý dữ liệu (9)

Multidimensional objects

Object	Tool	Description
Cube		Collection of measures related to aspects of the business and used to carry out a decision support investigation
Fact	—	Group of measures used among cubes
Dimension		Axis of investigation of a cube (time, product, geography)
Association		Association that relates a cube to a dimension
Attribute	—	Used to qualify a dimension. For example, attribute Year qualifies the Date dimension
Fact Measure	—	Variable linked to a fact, used as the focus of a decision support investigation
Hierarchy	—	Organizational structure that describes a traversal pattern through a dimension

4.1.2. Môi trường của PDM (1)

Defining the PDM environment

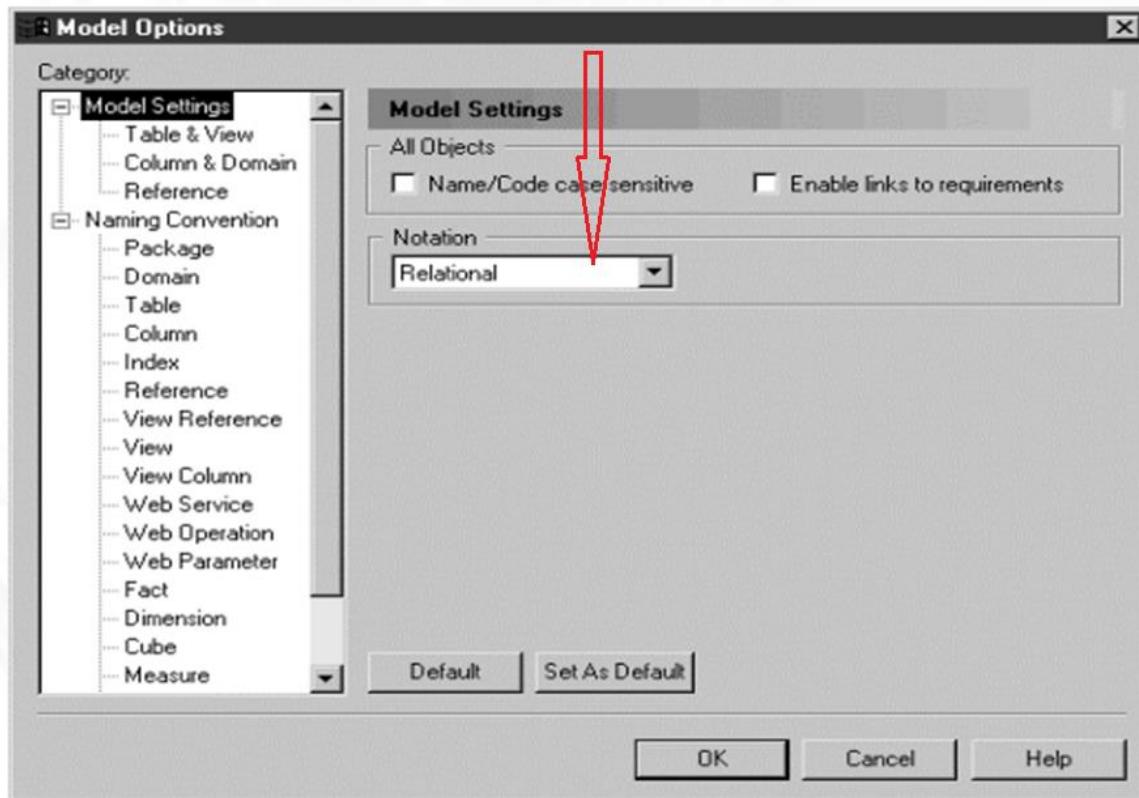
The PDM environment includes a set of parameters and configuration options that define various aspects of the model content and behavior. You can set these parameters:

- ◆ At model creation
- ◆ After creating a model with default options and parameters
- ◆ When creating a model template

PDM DBMS

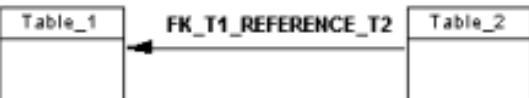
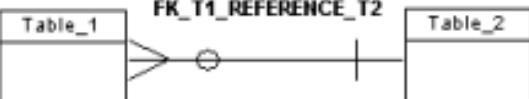
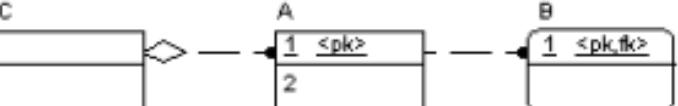
PowerDesigner can be used with many different DBMS. For each of these DBMS, a standard definition file is included and provides an interface between PowerDesigner and the DBMS so as to establish the best relationships between them.

3.1.2. Môi trường của PDM (2)



4.1.3. Định nghĩa PDM (1)

From the Notation groupbox, you can choose to use one of the following notation types in the current model:

Mode	Displays	Symbols
Relational*	Arrow pointing to primary key	
CODASYL	Arrow pointing to foreign key	
Conceptual	Cardinality displayed in IE format (crow's feet)	
IDEF1X	Cardinality and mandatory status displayed on reference, primary columns in separate containers and dependent tables with rounded rectangles. See below	

* This notation is the default notation used in this manual.

4.1.3. Định nghĩa PDM (2)

PDM database properties

You can define a database for a PDM.

☞ For more information on creating a database, see the section [Creating a database](#).

A database definition includes the following properties:

Property	Description
Name	Name for the database
Code	Code for the database. This code is generated in database scripts
Comment	Descriptive label for the database
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
DBMS	DBMS for the database
Options	Physical options available in the DBMS
Script	Begin and end scripts that are inserted at the start and end of a database creation script
Rules	Business rules for the database

4.1.3. Định nghĩa PDM (3)

Defining a PDM

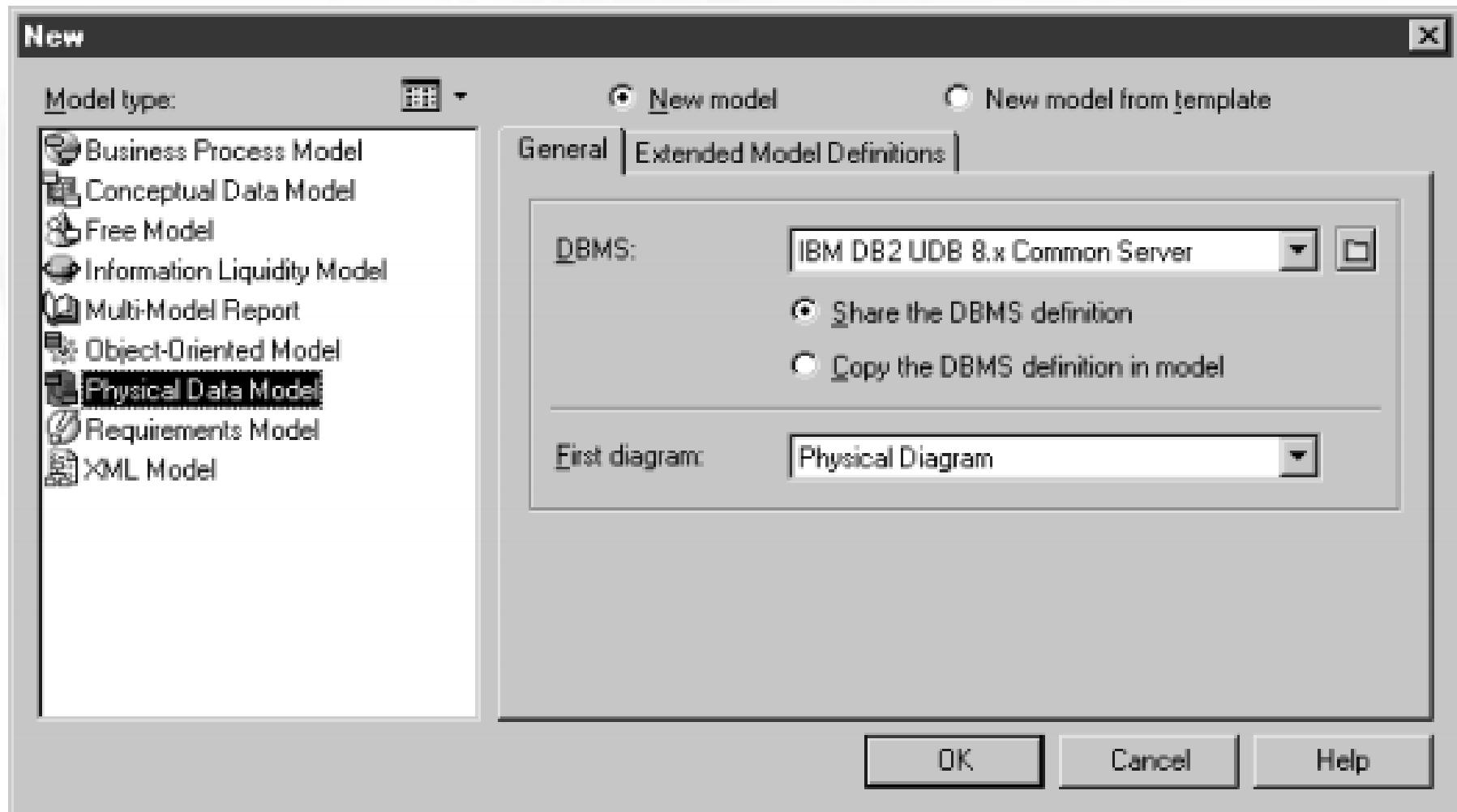
You can create a new PDM, or open an existing PDM. You can also set options to define how your model is created or modified, and the type of information that it displays.

Creating a PDM

There are several ways to create a PDM:

- ◆ Create a new PDM
- ◆ Create a new PDM using a template
- ◆ Create an PDM using existing elements (reverse engineering a database, importing an ERwin model (.erx), generating from a CDM or OOM, open a version 6 warehouse architect model, etc.)

4.1.3. Định nghĩa PDM (4)

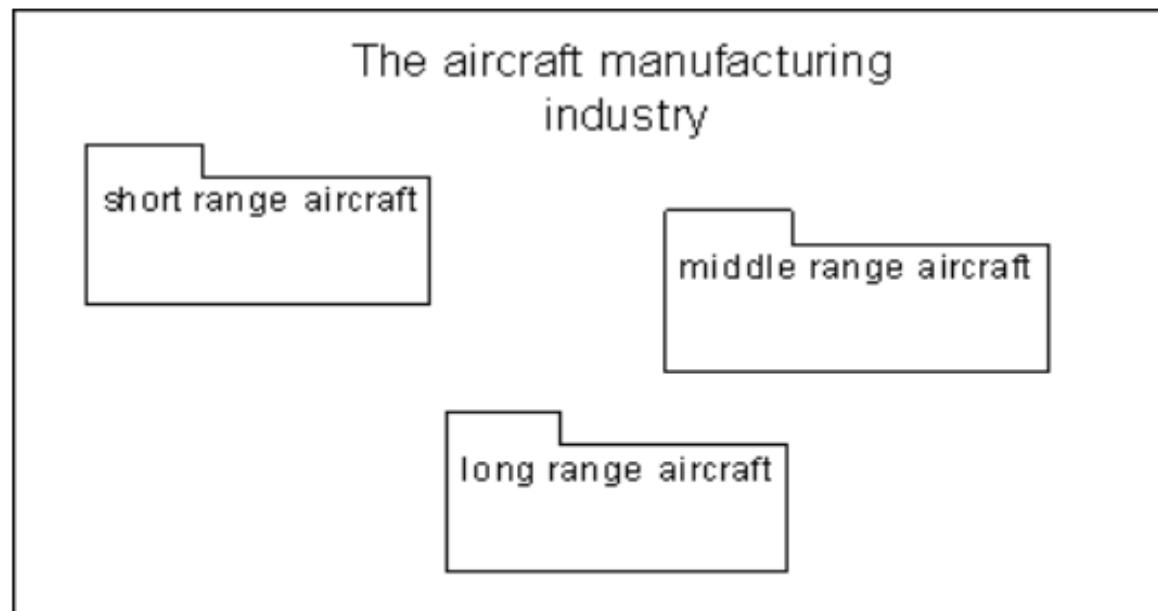


4.1.4. Định nghĩa các gói (package) trong PDM (1)

Defining Packages in a PDM

A package is a piece of a model.

When working with a large model, you can split the model into smaller subdivisions to avoid manipulating the entire set of model objects. Packages can be useful to assign portions of a model, representing different tasks and subject areas, to different development teams.



4.1.4. Định nghĩa các gói (package) trong PDM (2)

PDM package properties

Packages have properties displayed on property sheets. All packages share the following common properties:

Property	Description
Name	Name that clearly identifies the package
Code	Codes are references for packages
Comment	Optional label that describes a package and provides additional information
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Default diagram	Diagram displayed by default when you open the package
Use parent namespace	Defines the package as being the area in which the name of an object must be unique in order to be used

4.1.5. Sử dụng công cụ SQL (1)

Using SQL tools

Generating a PDM produces SQL code for your target DBMS.

PowerDesigner provides tools to change and preview SQL code during PDM development. These tools are:

- ◆ The SQL editor
- ◆ The SQL preview

The SQL editor allows you to define queries and the SQL preview allows you to see a SQL query script before it is generated.

4.1.5. Sử dụng công cụ SQL (2)

Defining queries with the SQL Editor

You can use the editing features of the SQL Editor to define a query. The SQL Editor dialog box is divided into specific panes containing the information shown below:

Information	Pane location
Objects types	Upper left part of the dialog box
Available objects	Upper right part of the dialog box
Query script textbox	Lower part of the dialog box

The list of available objects depends on the selected object type. You can select individual objects from the list of available objects for insertion in the query script textbox.

4.1.5. Sử dụng công cụ SQL (3)

You can also define expressions by entering basic arithmetic operators such as add, subtract, multiply, divide (+, -, *, /) and with the syntax tools shown below:

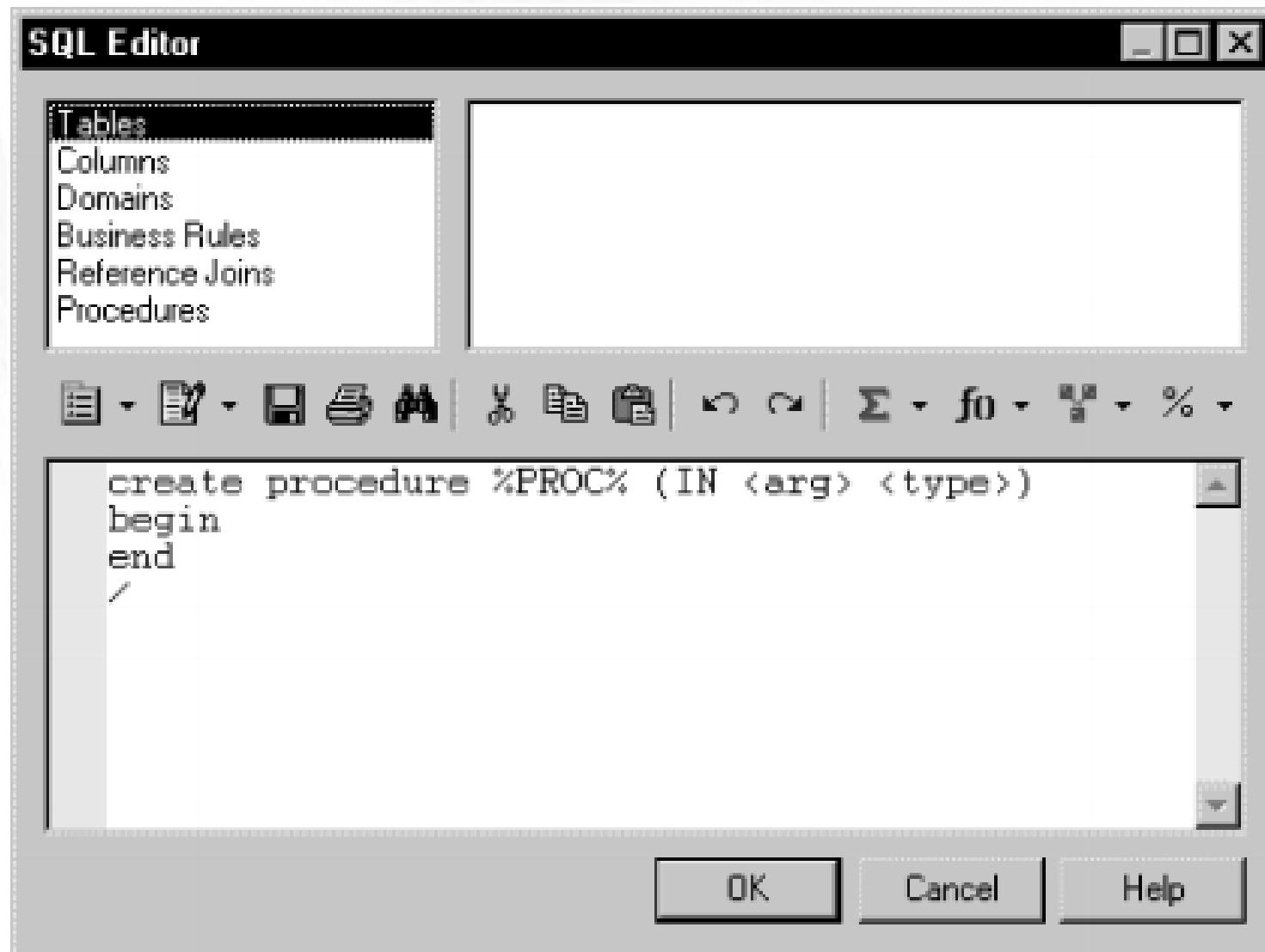
Syntax tool	Content
Functions	Provides group, number, string, date, conversion and other functions
Operators	Provides logical operators
Variables	Provides variables for use with operators and functions
Macros	Provides macros to accelerate the creation of a template item definition

These syntax tools allow greater flexibility in defining complex expressions.

The SQL Editor can be used for the following tasks:

- ◆ Defining a query for a view, a procedure, a trigger
- ◆ Defining a computed column

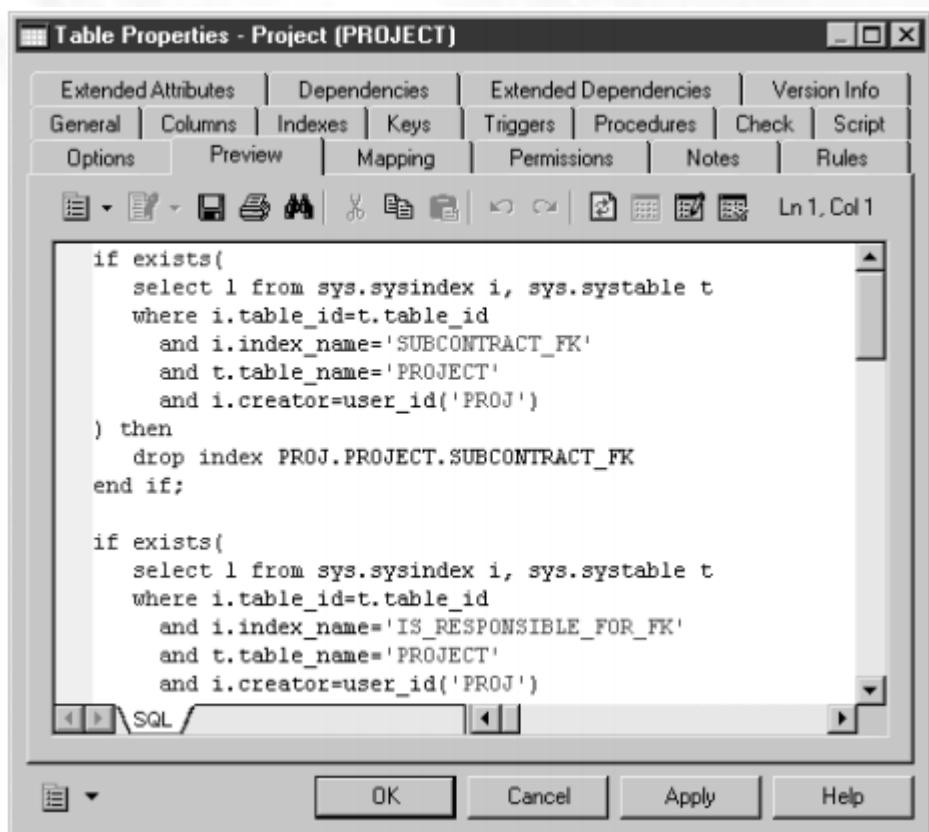
4.1.5. Sử dụng công cụ SQL (4)



4.1.5. Sử dụng công cụ SQL (5)

Previewing SQL statements

The Preview page shows an SQL query script before it is generated. The displayed script reflects the options selected in the Database Generation dialog box. This script cannot be modified.



The screenshot shows the 'Table Properties - Project (PROJECT)' dialog box. The 'Preview' tab is selected. The main area displays an SQL script:

```
if exists(
    select 1 from sys.sysindex i, sys.systable t
    where i.table_id=t.table_id
        and i.index_name='SUBCONTRACT_FK'
        and t.table_name='PROJECT'
        and i.creator=user_id('PROJ')
) then
    drop index PROJ.PROJECT.SUBCONTRACT_FK
end if;

if exists(
    select 1 from sys.sysindex i, sys.systable t
    where i.table_id=t.table_id
        and i.index_name='IS_RESPONSIBLE_FOR_FK'
        and t.table_name='PROJECT'
        and i.creator=user_id('PROJ')
)
```

The bottom of the dialog box has buttons for OK, Cancel, Apply, and Help.

4.2. Sử dụng các quy tắc trong mô hình VẬT LÝ DL

4.2.1. Quy tắc nghiệp vụ BR (Business Rules) trong PDM là gì?

4.2.2. Định nghĩa về BR

4.2.3. Sử dụng các BR trong PMD

4.2.1. Quy tắc nghiệp vụ BR (Business Rules) trong PDM là gì?

What is a business rule in a PDM?

A business rule is a rule that your business follows. This can be a government-imposed law, a customer requirement, or an internal guideline.

4.2.2. Định nghĩa về BR (1)

Defining business rules in a PDM

You can define a business rule which can be attached to an object in the PDM.

Business rule properties in a PDM

A business rule definition includes the following properties:

Property	Description
Name	Name for the rule
Code	Reference name for the rule
Comment	Descriptive label for the rule
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Type	Indicates if the rule is a definition, a fact, a formula, a requirement or a validation

A business rule definition also includes the following properties, each with their respective page:

28

Property	Description
Expression	Presence of associated expression
Notes	Presence of associated notes

28/16

4.2.2. Định nghĩa về BR (2)

Types of business rules in a PDM

The different business rule types that you can define in a PDM are described below:

Rule type	Describes	Example
Definition	Characteristics or properties of an object in the information system	A customer is a person identified by a name and an address
Fact	Certainty or existence in the information system	A client places one or more orders
Formula	Calculation used by the information system	The total order is the sum of all the order line costs
Requirement	Functional specification in the information system	The model is designed so that total losses do not exceed 10% of total sales
Validation	Constraint on a value in the information system. Validation business rules are generated in the database	The sum of all order totals for a client must not be greater than the allowance for the client
Constraint	Additional check constraint on a value. You can assign multiple constraint business rules to a table or a column. Constraint business rules are generated in the database	The start date should be inferior to the end date of a project

4.2.2. Định nghĩa về BR (3)

Creating a business rule in a PDM

Before you create business rules, formulate your rules by asking yourself the following questions:

- ◆ What business problems do I want to address?
 - ◆ Are there any mandatory procedures for my system?
 - ◆ Do any specifications set the scope of my project?
 - ◆ Do any constraints limit my options?
 - ◆ How do I describe each of these procedures, specifications, and constraints?
 - ◆ How do I classify these descriptions: as definitions, facts, formulas, or validation rules?

4.2.2. Định nghĩa về BR (4)

Attaching an expression to a business rule in a PDM

A business rule typically starts out as a description. As you develop your model and analyze your business problem, you can complete a rule by adding a technical expression. The syntax of expressions depends on the target database.

Example Let's consider the table Project:

Project
Project number
Customer number
Employee number
Start date (act)
End date (act)
Project name
Project label

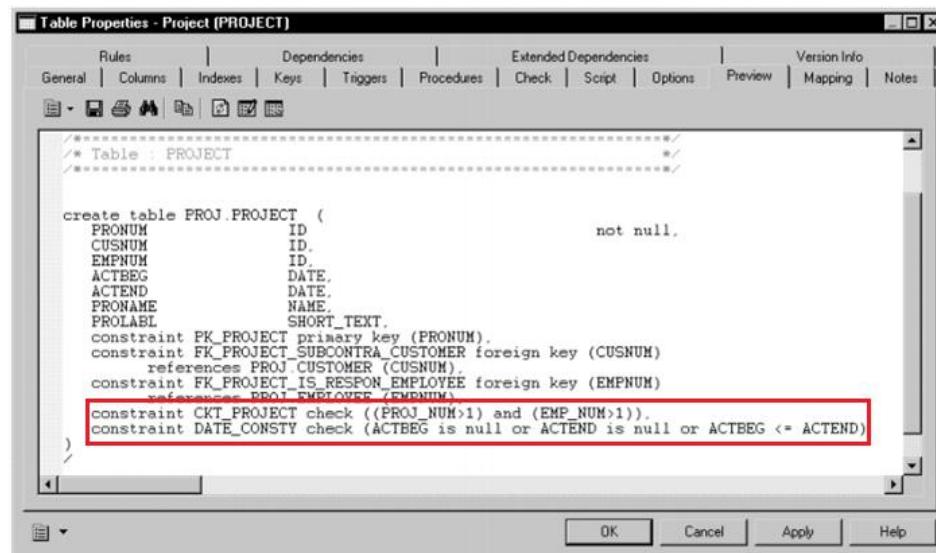
The following checks are defined for this table:

Check	Description
Check parameter (in the Check page of the table)	This check verifies that the customer number is different from the employee number
Validation business rule	PROJ_NUM to check that the column project number is not null EMP_NUM to check that the employee number is not null
Constraint business rule	DATE_CONSTY to check that the start date of the project is inferior to the end date of the project

4.2.2. Định nghĩa về BR (5)

Using constraint business rules in a PDM

When you display the table code preview, you can verify that the check parameters and validation business rules are concatenated into a single constraint, whereas the constraint business rule appears as a different constraint in the script.

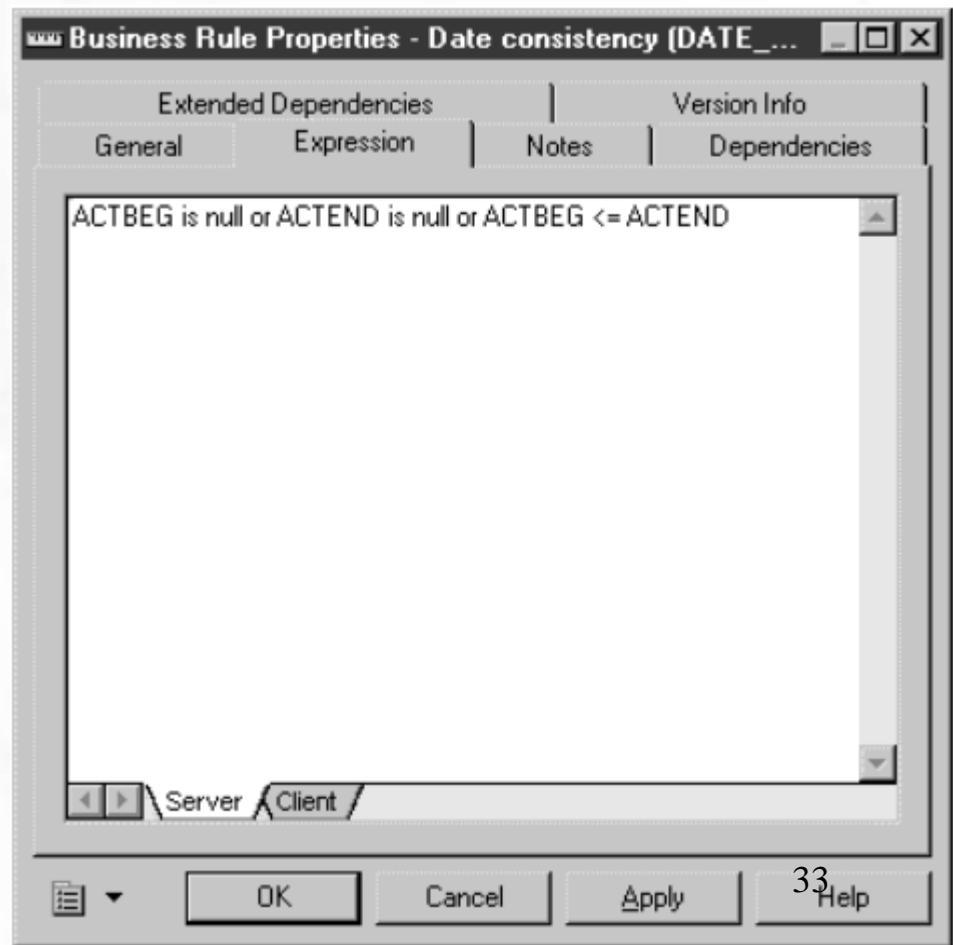
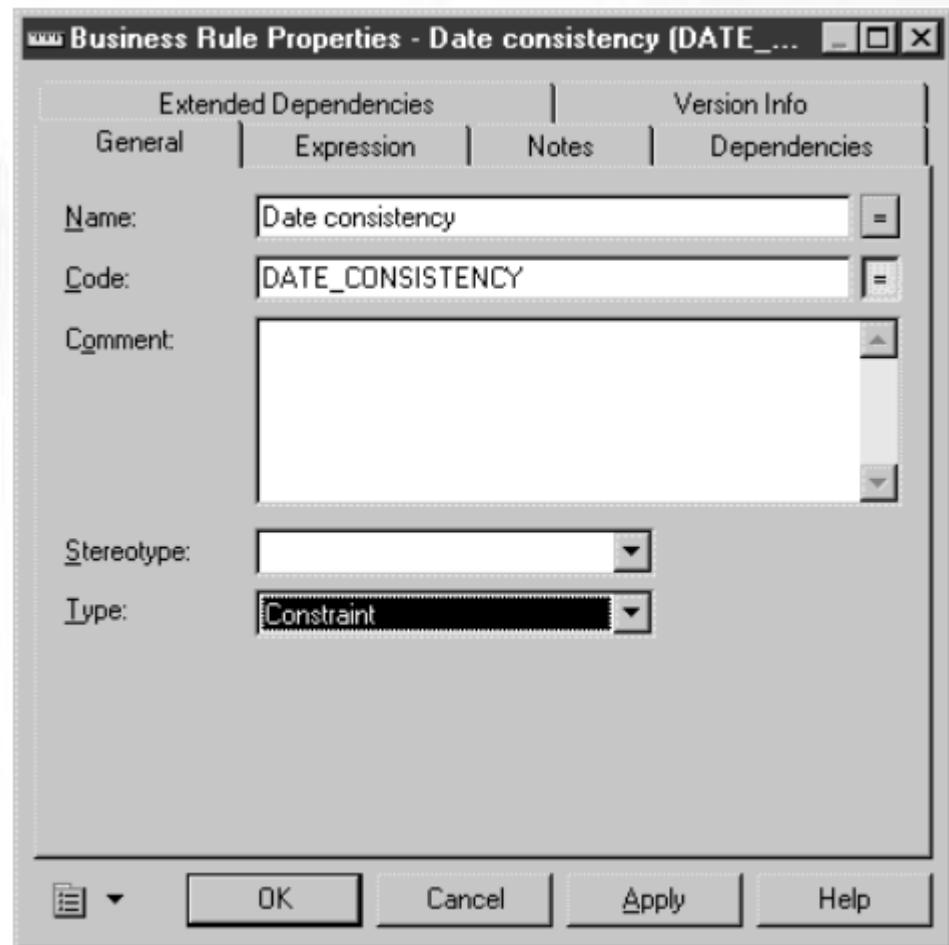


The screenshot shows the 'Table Properties - Project (PROJECT)' dialog box. The 'Script' tab is selected. The SQL code for the 'PROJECT' table is displayed:

```
/* Table : PROJECT */
create table PROJ.PROJECT (
    PRONUM          ID,
    CUSNUM          ID,
    EMPNUM          ID,
    ACTBEG          DATE,
    ACTEND          DATE,
    PRONAME         NAME,
    PROLABEL        SHORT_TEXT,
    constraint PK_PROJECT primary key (PRONUM),
    constraint FK_PROJECT_SUBCONTRA_CUSTOMER foreign key (CUSNUM)
        references PROJ.CUSTOMER (CUSNUM),
    constraint FK_PROJECT_IS_RESPON_EMPLOYEE foreign key (EMPNUM)
        references PROJ.EMPLOYEE (EMPNUM),
    constraint CKT_PROJECT check ((PRONUM>1) and (EMP_NUM>1)),
    constraint DATE_CONSTY check (ACTBEG is null or ACTEND is null or ACTBEG <= ACTEND)
)
```

The constraint 'constraint CKT_PROJECT check ((PRONUM>1) and (EMP_NUM>1))' and the constraint 'constraint DATE_CONSTY check (ACTBEG is null or ACTEND is null or ACTBEG <= ACTEND)' are highlighted with a red rectangle.

4.2.2. Định nghĩa về BR (6)



4.2.3. Sử dụng các BR trong PMD (1)

There are three ways to use business rules in a PDM:

- ◆ You can apply a business rule to an object in the PDM
- ◆ You can create a server expression for a business rule which can be generated to a database
- ◆ A business rule expression can also be inserted in a trigger or stored procedure

Applying a constraint business rule to a table or a column

You can attach a constraint business rule to a table or a column, however you cannot reuse a constraint business rule between different objects, so you must create as many constraint business rules as needed for your model objects.

4.2.3. Sử dụng các BR trong PMD (2)

The screenshot shows two overlapping windows from Oracle SQL Developer. The top window is titled "Table Properties - Project (PROJECT)". It has tabs for General, Columns, Indexes, Keys, Triggers, Procedures, Check, Script, Dependencies, Extended Dependencies, Version Info, Options, Preview, Mapping, Notes, and Rules. The "Rules" tab is selected. A toolbar below the tabs includes icons for Save, Print, Undo, Redo, Copy, Paste, Find, Replace, and others. A table grid lists three rules:

	Name	Code	Rule Type
1	Date consistency	DATE_CONSTY	Constraint
2	Project number	PROJ_NUM	
3	Employee number	EMP_NUM	

The bottom window is also titled "Table Properties - Project (PROJECT)" and displays the generated SQL code for the table:

```
/*
 * Table : PROJECT
 */
create table PROJ.PROJECT (
    PRONUM          ID,
    CUSNUM          ID,
    EMPNUM          ID,
    ACTBEG          DATE,
    ACTEND          DATE,
    PRONAME         NAME,
    PROLABL        SHORT_TEXT,
    constraint PK_PROJECT primary key (PRONUM),
    constraint FK_PROJECT_SUBCONTRA_CUSTOMER foreign key (CUSNUM)
        references PROJ.CUSTOMER (CUSNUM),
    constraint FK_PROJECT_IS_RESPON_EMPLOYEE foreign key (EMPNUM)
        references PROJ.EMPLOYEE (EMPNUM),
    constraint CKT_PROJECT check ((PROJ_NUM>1) and (EMP_NUM>1)),
    constraint DATE_CONSTY check (ACTBEG is null or ACTEND is null or ACTBEG <= ACTEND)
)
```

Both windows have standard "OK" and "Cancel" buttons at the bottom.

4.3. Xây dựng biểu đồ VẬT LÝ DỮ LIỆU

4.3.1. **Tổng quan** về PDM

4.3.2. Định nghĩa các **bảng** (table) trong PDM

4.3.3. Định nghĩa các **miền** (domain) trong PDM

4.3.4. Các **thiết lập mặc định** trong PDM

4.3.5. Dùng **dữ liệu abstract** trong PDM

4.3.6. Định nghĩa các **cột** (column) trong PDM

4.3.7. Định nghĩa các **khóa** (key) trong PDM

4.3.8. Định nghĩa các **tham chiếu** (reference) trong PDM

4.3.9. Định nghĩa các **chỉ mục** (index) trong PDM

4.3.10. Định nghĩa các **view** trong PDM

4.3.1. Tổng quan về PDM

Defining a physical diagram

A physical diagram allows you to define the database structure, from the physical implementation point of view. It takes into account the physical resources: DBMS, data storage structures and software, to describe the structure of the database.

Why build a physical diagram?

You build a physical diagram at the end of the data analysis process, before you start the software programming. The physical diagram allows you to define how data from conceptual model are implemented in the database.

4.3.2. Định nghĩa các bảng trong PDM (1)

Defining tables in a physical diagram

A table represents a collection of data arranged in columns and rows. It is the model equivalent of a database table.

From the property sheet of a table, you can access the following information about the table:

- ◆ Columns are named properties of a table that describe its characteristics.
- ◆ Indexes are data structures associated with a table that are logically ordered by key values.
- ◆ Keys are columns, or combinations of columns, that uniquely identify rows in a table.
- ◆ Triggers are segments of SQL code associated with tables, and stored in a database. They are invoked automatically whenever there is an attempt to modify data in associated tables.
- ◆ Options are values used in some databases to specify physical parameters for tables, indexes, tablespaces, storages, primary keys, and alternate keys.
- ◆ Preview allows you to see the SQL code of a table that will be used for its generation.

4.3.2. Định nghĩa các bảng trong PDM (2)

Table properties

You can double-click any table symbol in a diagram to display its properties:

Property	Description
Name	Name for the table. This name improves the readability of the model
Code	Reference code for the table. This code is generated in database scripts
Comment	Descriptive label for the table
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of table owner. You choose an owner from a list of users. A table can only have one owner at a time. This is normally the table creator
Number	Estimated number of occurrences in the physical database for the table (the number of records). This number is used to estimate database size. The Number box is automatically filled with the actual number of records if you select the check box Retrieve Number of Rows in Tables in the Options page of the Database Reverse Engineering dialog box
Generate	Indicates to generate the table in the database
Dimensional type	Indicates the multidimensional type of the table, that is Dimension or Fact

4.3.2. Định nghĩa các bảng trong PDM (3)

A table definition also includes the following properties:

Property	Description
Check	Table constraint that enforces data requirements of check parameters.
Script	Begin and end scripts for table creation
Preview	SQL code for table
Options	Physical options for the table. These are DBMS specific
Version Info	Table owner, version number, modification and creation details
Notes	Description associated with table

A table definition also includes the following properties, each with their own respective property sheet that can be accessed independently of the table.

- ◆ Columns
- ◆ Indexes
- ◆ Keys
- ◆ Triggers
- ◆ Business rules

4.3.2. Định nghĩa các bảng trong PDM (4)

Creating a table

There are several ways to create a table:

- ◆ From a diagram
- ◆ From the Browser
- ◆ From the list of tables

At creation, a table has a default name including a number, this number is assigned in the order of creation of objects.

4.3.2. Định nghĩa các bảng trong PDM (5)

Modifying the table display preferences

You can modify the following display preferences for a table using the Tools→Display Preferences command:

Table display	Text display preference	When selected, what it displays
	Columns	All columns, primary key columns, or a defined number (limit) of columns depending on the selections listed below*
	*All columns	All table columns
	*PK columns	Primary key columns only
	*Key columns	All key columns
	*Limit	(x) Number of columns depending on defined value of x
	Indexes	All indexes defined for table columns
	Keys	All keys defined on table
	Triggers	All triggers defined for table
	Owner	Name of table owner
	Stereotype	Stereotype of the table. You can also display the stereotype of columns, keys, indexes, and triggers

4.3.2. Định nghĩa các bảng trong PDM (6)

Display domain
and data type

You can display the domain of an attribute in the symbol of a table. There are four display options available:

Selected check box	Result	Symbol			
Data types	Displays only the data type, if it exists	<table border="1"><tr><td>SALE</td></tr><tr><td>SALE_ID <Undefined></td></tr><tr><td>STORE_ID char(12)</td></tr></table>	SALE	SALE_ID <Undefined>	STORE_ID char(12)
SALE					
SALE_ID <Undefined>					
STORE_ID char(12)					
Domains	Displays only the domain, if it exists	<table border="1"><tr><td>SALE</td></tr><tr><td>SALE_ID SALE_IDENTIFIER</td></tr><tr><td>STORE_ID <None></td></tr></table>	SALE	SALE_ID SALE_IDENTIFIER	STORE_ID <None>
SALE					
SALE_ID SALE_IDENTIFIER					
STORE_ID <None>					
Data types Domains	Displays both data type and domain, if they exist	<table border="1"><tr><td>SALE</td></tr><tr><td>SALE_ID <Undefined> SALE_IDENTIFIER</td></tr><tr><td>STORE_ID char(12) <None></td></tr></table>	SALE	SALE_ID <Undefined> SALE_IDENTIFIER	STORE_ID char(12) <None>
SALE					
SALE_ID <Undefined> SALE_IDENTIFIER					
STORE_ID char(12) <None>					
Data types Replace by domains	If domain exists and data type does not exist, then displays domain. If domain does not exist and data type exists, then displays data type.	<table border="1"><tr><td>SALE</td></tr><tr><td>SALE_ID SALE_IDENTIFIER</td></tr><tr><td>STORE_ID char(12)</td></tr></table>	SALE	SALE_ID SALE_IDENTIFIER	STORE_ID char(12)
SALE					
SALE_ID SALE_IDENTIFIER					
STORE_ID char(12)					

4.3.3. Định nghĩa các **miền** trong PDM (1)

Defining domains in a PDM

Domains help you identify the types of information in your model. They define the set of values for which a column is valid. Applying domains to columns makes it easier to standardize data characteristics for columns in different tables.

In a Physical diagram, you can associate the following information with a domain:

- ◆ Data type, length, and precision
- ◆ Check parameters
- ◆ Business rules
- ◆ Mandatory property

4.3.3. Định nghĩa các **miền** trong PDM (2)

Domain properties in a PDM

Each domain definition includes the following properties:

Property	Description
Name	Name for the domain
Code	Reference name for the domain
Comment	Descriptive label for the domain
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of domain owner. You choose an owner from a list of users. A domain can only have one owner at a time. This is normally the domain creator
Data type	Form of the data corresponding to the domain, such as numeric, alphanumeric, Boolean, or others
Length	Maximum number of characters
Precision	Number of places after the decimal point, for data values that can take a decimal point
Mandatory	Domain values are mandatory for all columns using that domain
Identity	(For Adaptive Server Enterprise, MS SQL Server and those DBMS that support it). When selected, indicates that the data is auto-incremented for columns using that domain

4.3.3. Định nghĩa các **miền** trong PDM (3)

Property	Description
With default	(For those DBMS that support it). When selected, indicates if a default value is assigned to a column using the domain, when a Null value is inserted
Profile	Test Data profile assigned to the domain

A domain definition can also include the following properties, which have associated values or information used by columns attached to the domain:

Property	Description
Standard checks	Check parameters defined for the domain
Additional checks	Domain constraints or validation rules not defined by standard check parameters
Rules	Business rules attached to the domain

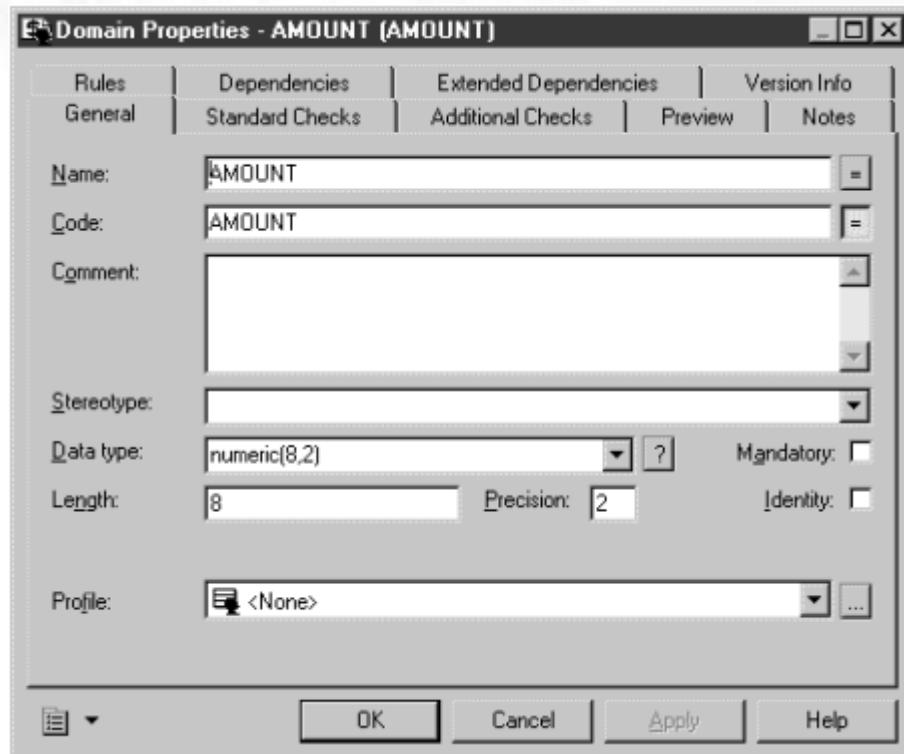
4.3.3. Định nghĩa các **miền** trong PDM (4)

Creating a domain in a PDM

You create a domain from the list of domains.

Accessing the List of Domains

You can access the List of Domains from the current model, or by right clicking the appropriate model node in the Browser, and selecting New→Domain from the contextual menu.



4.3.3. Định nghĩa các **miền** trong PDM (5)

Indicating data type, length, and precision

The data types that you can select in a Physical diagram depend on your current DBMS.

Length and precision

The properties length and precision do not apply to all data types. Depending on data type, length may indicate a maximum or a fixed number of characters.

4.3.3. Định nghĩa các **miền** trong PDM (6)

Numeric data types

Conceptual data type	Content	Length	Mandatory Precision
Integer	32-bit integer	—	—
Short Integer	16-bit integer	—	—
Long Integer	32-bit integer	—	—
Byte	256 values	—	—
Number	Numbers with a fixed decimal point	Fixed	✓
Decimal	Numbers with a fixed decimal point	Fixed	✓
Float	32-bit floating point numbers	Fixed	—
Short Float	Less than 32-bit point decimal number	—	—
Long Float	64-bit floating point numbers	—	—
Money	Numbers with a fixed decimal point	Fixed	✓
Serial	Automatically incremented numbers	Fixed	—
Boolean	Two opposing values (true/false; yes/no; 1/0)	—	—

4.3.3. Định nghĩa các **miền** trong PDM (7)

Character data types

Conceptual data type	Content	Length
Characters	Character strings	Fixed
Variable Characters	Character strings	Maximum
Long Characters	Character strings	Maximum
Long Var Characters	Character strings	Maximum
Text	Character strings	Maximum
Multibyte	Multibyte character strings	Fixed
Variable Multibyte	Multibyte character strings	Maximum

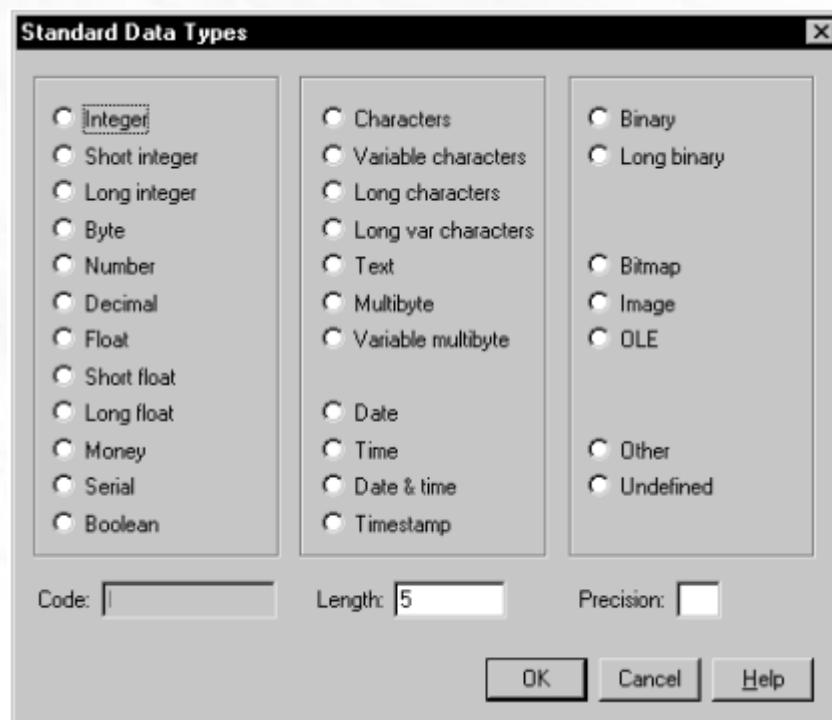
Time data types

Conceptual data type	Content
Date	Day, month, year
Time	Hour, minute, and second
Date & Time	Date and time
Timestamp	System date and time

4.3.3. Định nghĩa các **miền** trong PDM (8)

Other data types

Conceptual data type	Content	Length
Binary	Binary strings	Maximum
Long Binary	Binary strings	Maximum
Bitmap	Images in bitmap format (BMP)	Maximum
Image	Images	Maximum
OLE	OLE links	Maximum
Other	User-defined data type	—
Undefined	Not yet defined data type	—

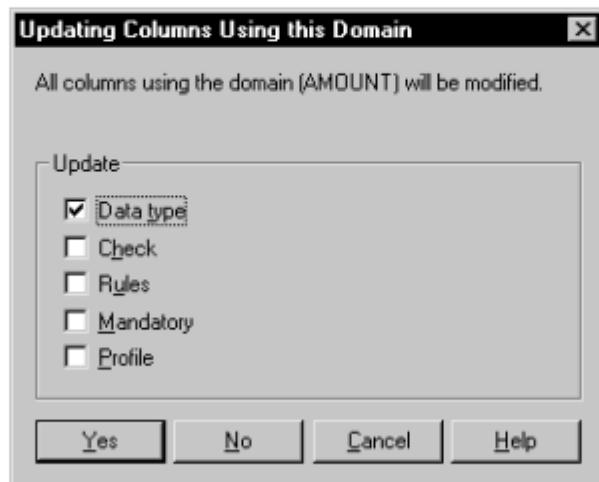


4.3.3. Định nghĩa các **miền** trong PDM (9)

Modifying domain properties in a PDM

You can modify domain properties from its property sheet.

When you modify data types associated with a domain, an update confirmation box appears asking if you want to modify the columns currently using the domain.



You can now click one of the buttons with the following effects:

Button	Effect
Yes	The columns currently using the domain are modified according to the update
No	The columns currently using the domain are not modified according to the update but the current modification is accepted if domain divergence is allowed in the model options
Cancel	The update is cancelled and nothing is changed

4.3.4. Các thiết lập mặc định trong PDM (1)

Defining defaults in a PDM

A default is a value that can be assigned to a column or a domain in the DBMS of the Sybase Adaptive Server Enterprise and Microsoft SQL Server families.

You select a default from the Default dropdown listbox in the Check Parameters page of a column or domain property sheet.

Example

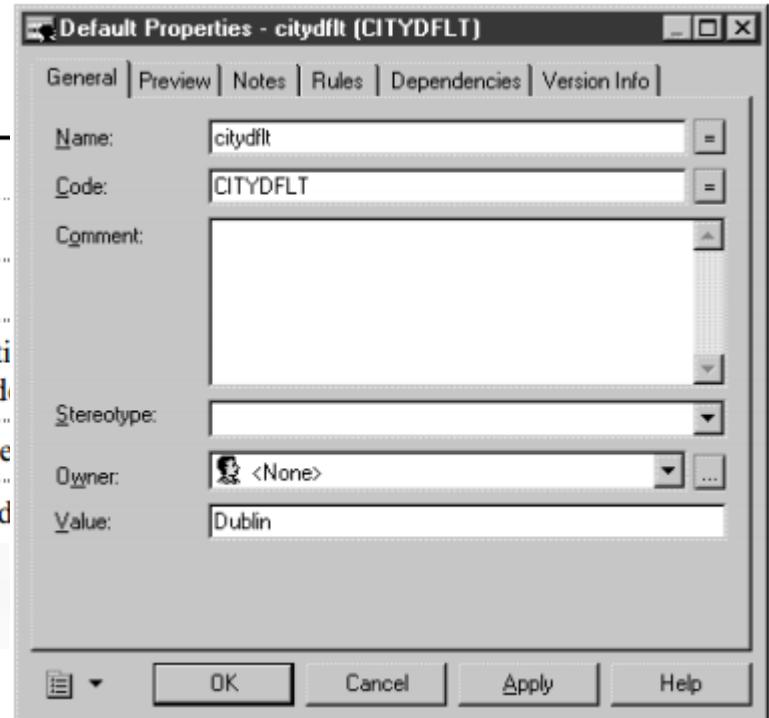
The default object **citydf1t** is used to assign the same default value to all columns of type city.

4.3.4. Các thiết lập mặc định trong PDM (2)

Default properties in a PDM

A default includes the following properties:

Property	Description
Name	Name for default
Code	Reference code for default
Comment	Descriptive label
Stereotype	Sub-classification used to extend the semantics without changing its structure; it can be predicted by the system
Owner	Name of default owner. You choose an owner from the list of users
Value	Value of default object that will be generated



Creating a default in a PDM

You create a default from the list of defaults.

When you create defaults, make sure you do not create several defaults with the same value. This will raise a warning message during check model.

4.3.5. Dùng dữ liệu abstract trong PDM (1)

Using abstract data types in a physical diagram

An **abstract data type (ADT)** is a user-defined data type which can encapsulate a range of data values and functions. The functions can be both defined on, and operate on the set of values.

Example

An abstract data type for the Gregorian calendar which has functions defined to do the following:

- ◆ Read and write roman numerals
- ◆ Convert dates from the Julian calendar to the Gregorian calendar
- ◆ Convert dates from the Gregorian calendar to the Julian calendar

4.3.5. Dùng dữ liệu abstract trong PDM (2)

Types of abstract data types used in PowerDesigner

Depending on the current DBMS, the following groups of abstract data types can be represented in PowerDesigner:

Type	Description	Example
Array	Fixed length collection of elements	VARRAY (Oracle 8 and higher)
List	Unfixed length collection of objects	TABLE (Oracle 8 and higher)
Java	Java class	JAVA (Adaptive Server Anywhere, and Adaptive Server Enterprise)
Object	Contains a list of attributes and a list of procedures	OBJECT (Oracle 8 and higher)
SQLJ Object	Contains a list of attributes and a list of procedures	SQLJ OBJECT (Oracle 9i and higher)
Structured	Contains a list of attributes	NAMED ROW TYPE (Informix 9.x, and IBM DB2 5.2)

4.3.5. Dùng dữ liệu abstract trong PDM (3)

How are abstract data types used in PowerDesigner?

Abstract data types can be used in the following ways in a Physical diagram:

Abstract data type is	Description
Created	You can create an abstract data type for each category supported by the current DBMS. If you create an abstract data type of type JAVA, you can link it to a Java class in an OOM to access the Java class properties
Reverse engineered	Declaration of an abstract data type is reverse engineered into a PDM

4.3.5. Dùng dữ liệu abstract trong PDM (4)

Linking to Java classes in the object-oriented model

Automatic linking of reversed Java classes

Linking Java classes

Java classes cannot be created in the PDM. You need to specify an abstract data type as type JAVA, and then link it to an existing Java class in the OOM. Once a Java class is linked, you can access its properties.

You cannot create a Java class in a PDM. You need to specify the name and code of a Java class in the List of Abstract Data Types, then link it to a Java class in a PowerDesigner Object-Oriented Model. The OOM must be open in the current Workspace. Once an abstract data type is linked to a Java class, you can access the Java class property sheet.

You can reverse engineer a database that contains Java classes into a PDM. To have access to the Java class properties, you must also reverse engineer the Java classes into an OOM. The columns or domains that use the Java classes as data types in the PDM are then automatically linked to the corresponding Java classes in the OOM.

4.3.5. Dùng dữ liệu abstract trong PDM (5)

Abstract data type properties

An abstract data type includes the following properties:

Property	Description
Name	Name for abstract data type
Code	Reference code for abstract data type
Comment	Descriptive label
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of abstract data type owner. You choose an owner from a list of users
Type	Defining group that includes the abstract data type

4.3.5. Dùng dữ liệu abstract trong PDM (6)

Depending on its type, an abstract data type definition can also include the following properties:

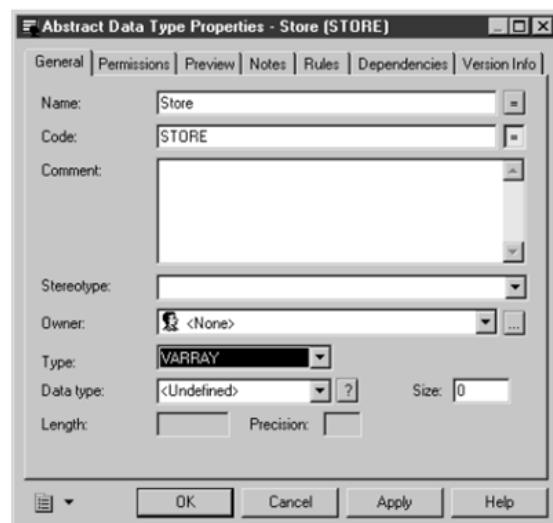
- ◆ Data type, Length, and Precision
- ◆ Size (for arrays)
- ◆ Linked class name (for Java types)
- ◆ File name and path, which contains the declaration of the class
- ◆ Authorization (for objects): Invoker Right attribute used for DDL generation
- ◆ Supertype (for objects): Parent abstract data type from which the current abstract data type can inherit the procedures
- ◆ Final and Abstract (for objects): When Final is checked, the current abstract data type cannot be used as supertype by another abstract data type. When Abstract is checked, the current abstract data type cannot be instantiated. Final and Abstract are mutually exclusive
- ◆ Java class (for SQLJ objects): Name of an external Java class to which the SQLJ object points. Beside the Java class box, there is a dropdown listbox to select a mapping interface (CustomDatum, OraData or SQLData)

4.3.5. Dùng dữ liệu abstract trong PDM (7)

Creating an abstract data type

You can create an abstract data type from the List of Abstract Data Types. If the current DBMS does not support abstract data types, the menu item does not appear.

When you create an abstract data type it then becomes available to be used as a data type for columns and domains in the physical diagram.



4.3.6. Định nghĩa các **cột** trong PDM (1)

Defining columns in a physical diagram

A column contains an individual data item within a row. It is the model equivalent of a database column. A column is always defined for a table. When you create a column, it must be assigned a name and code. You can also select a data type for the column. This can be done directly from a list of available data types, or by attaching the column to a domain.

4.3.6. Định nghĩa các **cột** trong PDM (2)

Column properties

You can access column properties from a table property sheet:

Property	Description
Name	Name for the column. This name improves the readability of the model
Code	Reference code for the column
Comment	Descriptive label for the column
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Table	Table which contains column
Data type	Form of the data corresponding to the column, such as numeric, alphanumeric, boolean, or others
Displayed	When selected, allows the display of selected column in list of columns, otherwise the column is grayed
Length	Maximum length of the data type
Precision	Maximum number of places after the decimal point
Identity	When selected, indicates that the data is auto-incremented (not available for all DBMS)
Domain	Name of the associated domain

4.3.6. Định nghĩa các **cột** trong PDM (3)

Property	Description
Computed	When selected, designates that the column is computed from an expression using values from other columns in the table (not available for all DBMS)
Primary key	When selected, designates a column whose values uniquely identify a row in the table
Foreign key	When selected, designates a column that depends on and migrates from a primary key column in another table
Mandatory	When selected, indicates a column that must be assigned a not null value
With default	When selected, indicates if a default value is assigned to the column when a Null value is inserted (not available for all DBMS)

4.3.6. Định nghĩa các **cột** trong PDM (4)

A column also includes the following properties:

Property	Description
Detail	Column fill parameters. These determine the percentage of column entries for which a null value is permitted, and the percentage of column rows that contain unique values. Test data parameters. You can select a test data profile from the list available using the dropdown listbox Computed Expression. You can enter a computed expression for the column
Standard checks	Standard check parameters
Additional checks	SQL expression defining a check parameter
Options	Physical options for columns. These are DBMS-specific

4.3.6. Định nghĩa các **cột** trong PDM (5)

Column details

Column details indicate common data constraints. The following table lists column detail parameters:

Parameter	Description
Null Values	Percentage of column entries for which a null value is permitted
Distinct Values	Percentage of column entries for which a distinct value is permitted
Average Length	Average length of a value
Test Data Parameters	Test data profile selected from the dropdown listbox. Profiles can use characters, numbers or date/time data types
Computed Expression	Computed expression typed directly in the Computed Expression pane or defined with the SQL Editor (accessed with the Edit tool) which helps you define more complex expressions

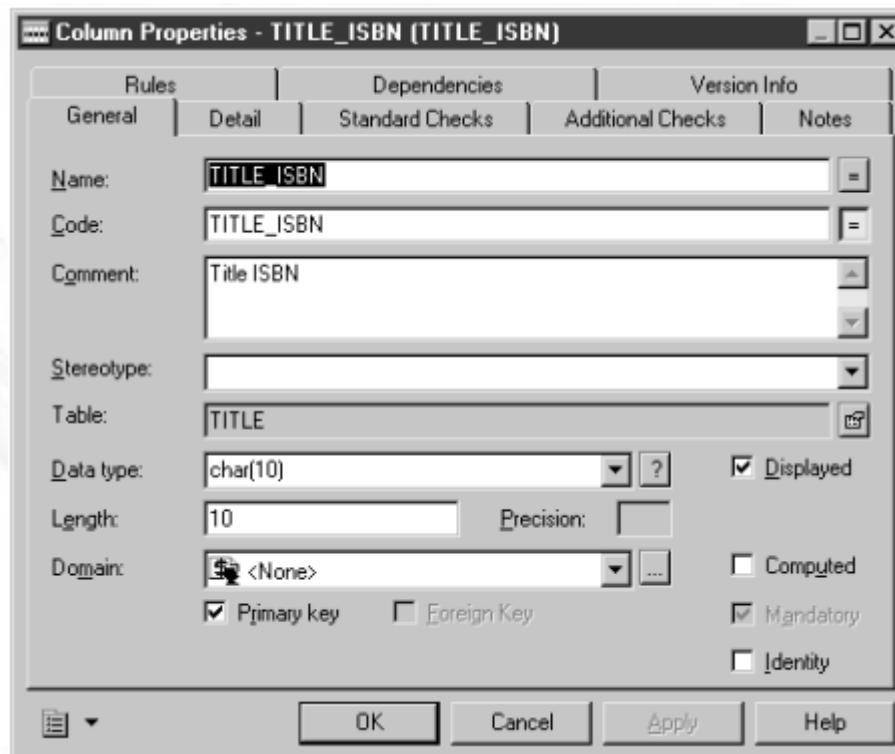
4.3.6. Định nghĩa các **cột** trong PDM (6)

Creating columns

You create a column from the table property sheet. To display the table property sheet, you can:

- ◆ Double-click the table symbol
- ◆ Right-click the table node in the Browser and select Properties

At creation, a column has a default name including a number, this number is assigned in the order of creation of objects.



4.3.6. Định nghĩa các **cột** trong PDM (7)

Creating a computed c

A comp
expressi
column

Simple
Express
complex
found o

Comput

Example

Assume
widgets
number

Column
Number
Widget p
Widget s

While o
complex



4.3.6. Định nghĩa các **cột** trong PDM (8)

Creating a sequence for a column

When your DBMS supports the feature, you can create a sequence for a column.

To understand sequences, you must first understand auto-incremented columns. An auto-incremented column is one whose values automatically increment by 1 thereby producing a list of consecutive numbers. Sequences are special auto-incremented columns with features that allow you to define a more complex list of numbers. For example, sequences allow you to define a list of numbers ranging between two values with an increment by any number (integer) you want.

Once you define a sequence, you can apply and enable it to a column. The data type for the column receiving the sequence must be a numeric data type. Such auto-incremented columns can be used in a key for a PDM table.

4.3.6. Định nghĩa các **cột** trong PDM (9)

Effect of DBMS change on sequences and auto-incremented columns

If you create a sequence attached to a column in a DBMS supporting sequences, such as Oracle 8 and higher, Interbase or PostgreSQL, or create an auto-incremented column in a DBMS supporting this feature, and then decide to change the target DBMS, the following effects occur:

DBMS change	Defined in original DBMS	Effect on sequence objects and auto-incremented columns
DBMS supporting sequences to a DBMS supporting auto-incremented columns	Sequence attached to a column	The sequence disappears and the column to which it was attached becomes an auto-incremented column in the DBMS
DBMS supporting auto-incremented columns to a DBMS supporting sequences	Auto-incremented column	The auto-incremented column is deleted and replaced by a sequence object called S_TABLENAME which is attached to the original column

4.3.6. Định nghĩa các **cột** trong PDM (10)

Sequences and intermodel generation

When a CDM or an OOM is generated from a PDM, the data type of a table column attached to a sequence is translated to a numeric data type in the new model:

PDM generated to	Sequence is converted to
CDM	A serial data type for an entity property. The data type has the format NO%<i>n</i> where <i>n</i> is a number indicating the length of the data type
OOM	A serial data type for a class attribute. The data type has the format NO%<i>n</i> , where <i>n</i> is a number indicating the length of the data type

Example

Assume that you want to create a column listing the months of the year when quarterly reports are published: March, June, September, and December. The first report is published on the third month, the second on the sixth, the third on the ninth and the last on the twelfth.

You can define the proper sequence by typing the following values for sequence option parameters:

Parameter name	Description	Value
Start with	March is the third month of the year	3
Increment by	Look three months ahead to identify the next month in the list	3
Maxvalue	Stop when you have reached the last month of the year	12

71

The sequence created with these parameter settings allows you to automatically create the list of months in a year when quarterly reports are published.

71/16

4.3.6. Định nghĩa các **cột** trong PDM (11)

Sequence properties

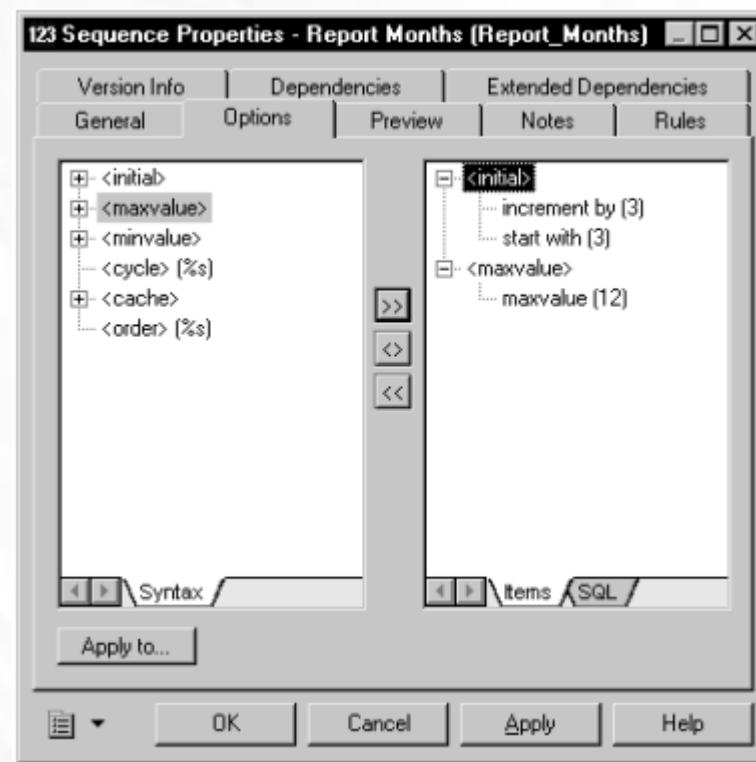
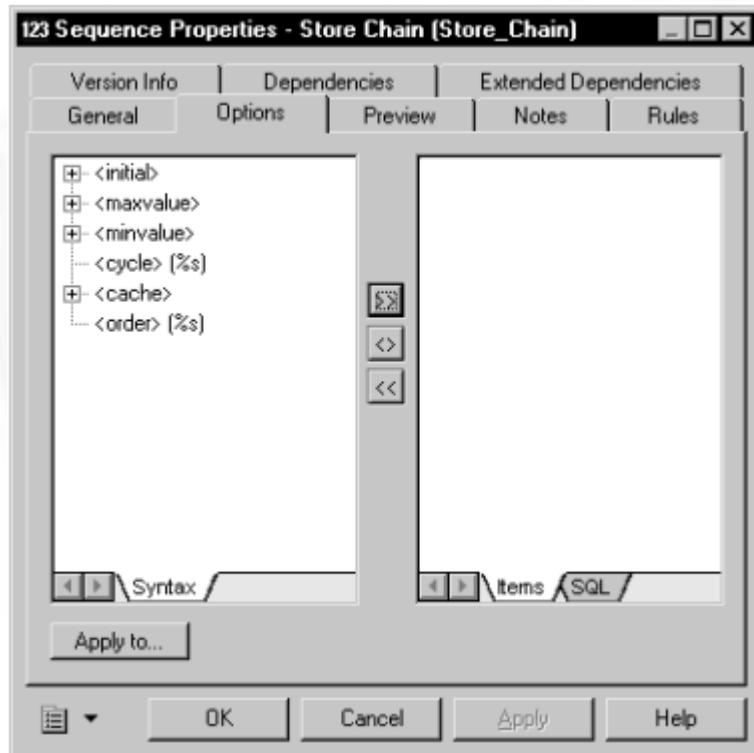
A sequence includes the following properties:

Property	Description
Name	Name for the sequence. This name improves the readability of the model
Code	Reference code for the sequence
Comment	Descriptive label for the sequence
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of sequence owner. You choose an owner from a list of users. A column can only have one owner at a time. This is normally the column creator

A sequence also includes the following property, with its own page:

Property	Description
Options	Physical options for sequences. These are DBMS-specific

4.3.6. Định nghĩa các **cột** trong PDM (12)



4.3.6. Định nghĩa các **cột** trong PDM (13)

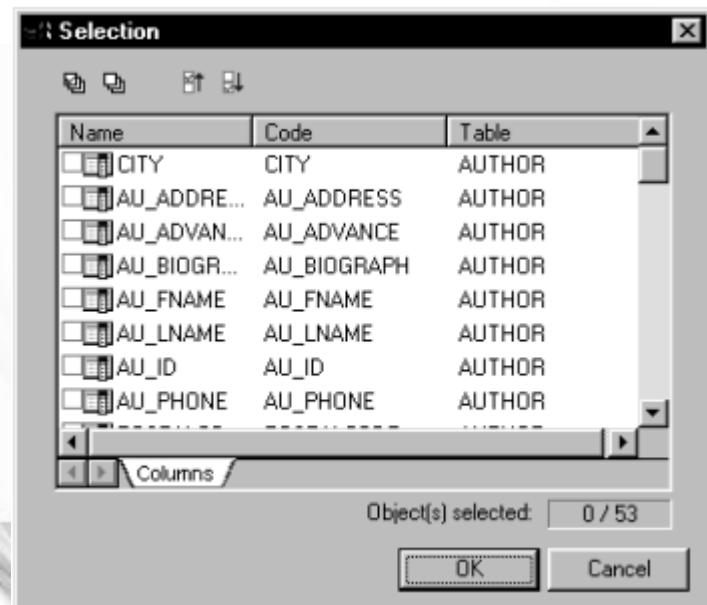
Selecting a data type for a column

There are two ways to select a data type for a column:

- ◆ **Attach the column to a domain** The domain dictates a data type, a length, and a level of precision, as well as optional check parameters
- ◆ **Manually select a data type** You select a data type along with a length, a level of precision, and optional check parameters

Copying a column to another table

You can copy a column from one table and add it to another table. If the table already contains a column with the same name or code as the copied column, the copied column is renamed. For example, the column PUB_ID is renamed PUB_ID2 when it is copied to a table which already contains a column PUB_ID.



4.3.7. Định nghĩa các khóa (key) trong PDM (1)

Defining keys in a physical diagram

A key is a column, or a combination of columns, that uniquely identifies a row in a table. Each key can generate a unique index or a unique constraint in a target database.

The physical diagram supports the following types of keys:

Key	Description
Primary	Column or combination of columns whose values uniquely identify every row in a table. A table can have only one primary key
Alternate	Column or combination of columns (not the same column or combination of columns as for a primary key) whose values uniquely identify every row in a table
Foreign	Column or combination of columns whose values are required to match a primary key, or alternate key, in some other table

4.3.7. Định nghĩa các khóa (key) trong PDM (2)

Example

The TITLE table shown below has a primary, alternate and foreign key:

- ◆ TITLE_ID is the primary key and consists of the column TITLE ISBN which identifies each book title in the table
- ◆ TITLE_NAME is an alternate key containing the columns TITLE NAME and TITLE TYPE. It allows each title to be identified by its name and type, The fact that it is an alternate key indicates that there is a constraint that no two titles of the same type can have the same name

The TITLE table also contains the foreign key column PUBLISHER ID. This column references the primary key column in the Publisher table.

Title	
Title ISBN	<pk>
Publisher ID	<fk>
Title Name	<ak>
Title Type	<ak>
Title Price	
Title Publication Date	
One Title_ID	<pk>
One Title_Name	<ak>

4.3.7. Định nghĩa các khóa (key) trong PDM (3)

Designating a primary key

A primary key is the primary identifier for a table. A primary key is made up of one or more columns whose values uniquely identify every row in the table.

Every table must have a primary key, composed of one or more of its columns.

Example

Employee number is the primary key for the table Employee. This means that each employee must have one unique employee number.

Employee		
Employee number	numerico(5)	<pk>
Division number	numerico(5)	<fk>
First name	char(30)	
Last name	char(30)	
Employee function	char(30)	
Employee salary	numerico(8,2)	
Primary Key <pk>		

You can define one or more columns as the primary key of a table from the list of columns.

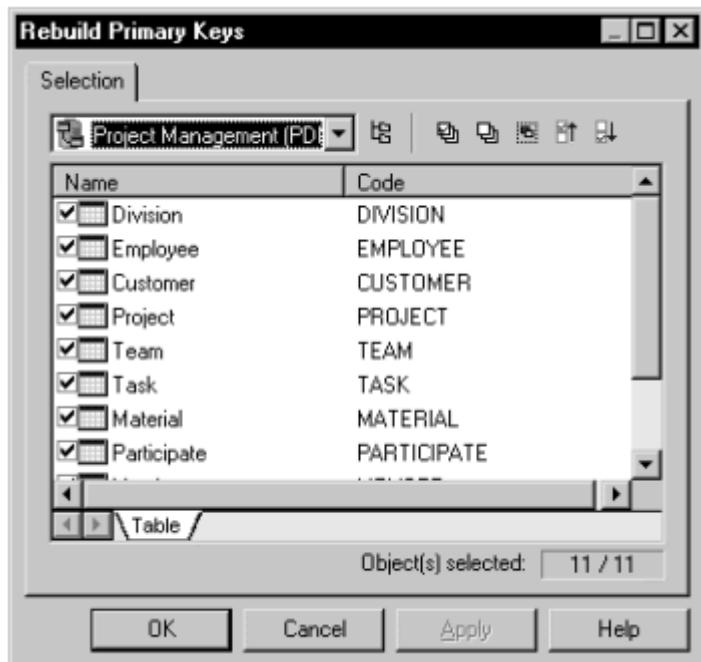
4.3.7. Định nghĩa các khóa (key) trong PDM (4)

Rebuilding primary keys

Rebuilding primary keys in a physical diagram updates primary keys for tables.

Rebuilding primary keys is useful following the reverse engineering of a database in which all of the primary keys could not be reverse engineered, or if you did not select the rebuild option for primary keys when you reverse engineered the database. The rebuild option for primary keys creates primary keys for tables that have no key and a single unique index.

You can choose to rebuild all primary keys in your model, or select the tables for which you want to rebuild the primary keys.



4.3.7. Định nghĩa các khóa (key) trong PDM (5)

Designating a foreign key

A foreign key is a primary key, or an alternate key, that migrates from another table. Depending on selected model options, a primary key can be automatically migrated to a child table as a foreign key at reference creation.

The columns that are defined in a foreign key can also be user-specified at creation and changed at any time from the Joins page of the reference property sheet.

Designating an alternate key

An alternate key is made up of one or more columns whose values uniquely identify every row in the table, and is not a primary key. An alternate key can also be a foreign key. Each alternate key can generate a unique index or a unique constraint in a target database.

4.3.7. Định nghĩa các khóa (key) trong PDM (6)

Naming key constraints

Naming key constraints helps you to identify and customize key constraints in scripts for database creation and modification. The constraint name gives you greater flexibility for modifying keys in subsequent database generations.

If you do not specify a constraint name, PowerDesigner creates a default constraint name automatically.

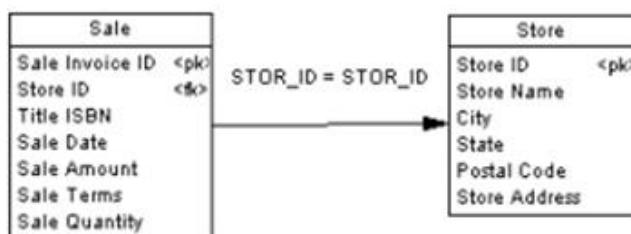
4.3.8. Định nghĩa các tham chiếu trong PDM (1)

Defining references in a physical diagram

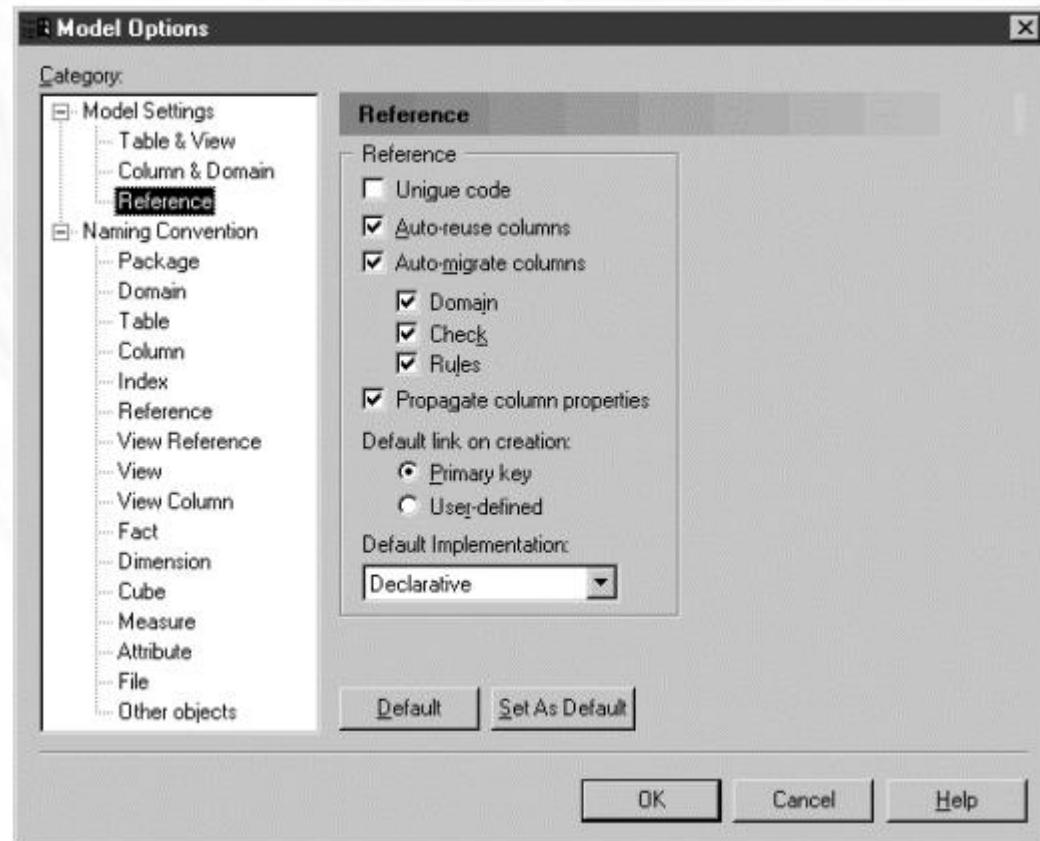
A **reference** is a link between a parent table and a child table. It defines a referential integrity constraint between column pairs for a primary key, or alternate key, and a foreign key, or between user specified columns in both tables.

Example

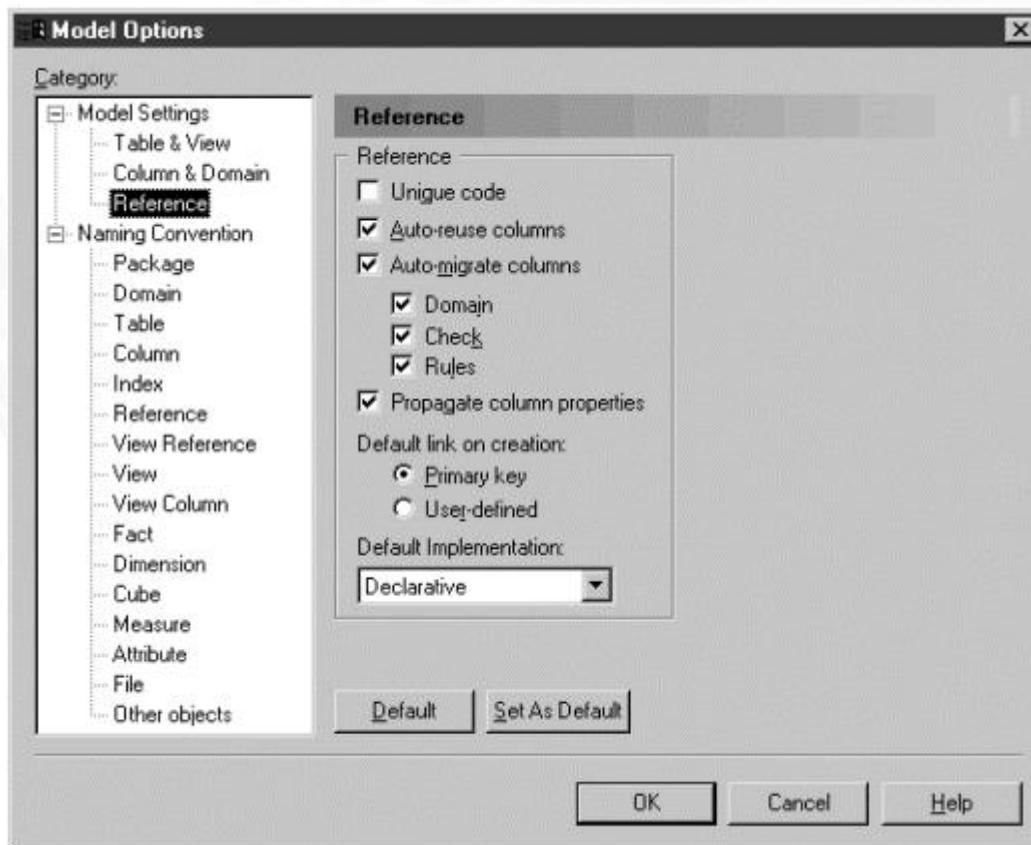
The two tables SALE and STORE are linked by a reference. STORE is the parent table and SALE is the child table. The reference contains a join which links the primary key column STORE ID (the referenced column) to the foreign key column STORE ID (the referencing column).



4.3.8. Định nghĩa các tham chiếu trong PDM (2)



4.3.8. Định nghĩa các tham chiếu trong PDM (3)



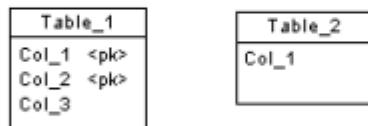
4.3.8. Định nghĩa các tham chiếu trong PDM (4)

The Child table column must have the same code as the migrating primary key column, and cannot already be a foreign key column.

PowerDesigner does not automatically reuse existing foreign key columns. This must be done manually from the Joins page of the reference property sheet.

Matching child table column exists

The following table shows the results of migrating primary key columns to a child table that contains a matching column for one of the primary key columns. The original two tables are also shown below:



Auto-reuse	Auto-migrate	Result	Description of child table								
Selected	Selected	<table border="1"><tr><td>Table_1</td><td>Table_2</td></tr><tr><td>Col_1 <pk></td><td>Col_1 = Col_1</td></tr><tr><td>Col_2 <pk></td><td>Col_2 = Col_2</td></tr><tr><td>Col_3</td><td></td></tr></table>	Table_1	Table_2	Col_1 <pk>	Col_1 = Col_1	Col_2 <pk>	Col_2 = Col_2	Col_3		Col_1 is reused and Col_2 is created
Table_1	Table_2										
Col_1 <pk>	Col_1 = Col_1										
Col_2 <pk>	Col_2 = Col_2										
Col_3											

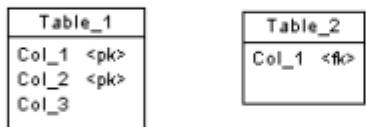
4.3.8. Định nghĩa các tham chiếu trong PDM (5)

Auto-reuse	Auto-migrate	Result	Description of child table
Not selected	Selected	<pre> graph LR T1[Table_1
Col_1 <pk>
Col_2 <pk>
Col_3] -- "Col_1 = t1_Col_1" --> T2[Table_2
H1_Col_1 <pk>
Col_2 <fk>
Col_1] T1 -- "Col_2 = Col_2" --> T2 </pre>	T1_Col_1 is created and Col_2 is created
Selected	Not selected	<pre> graph LR T1[Table_1
Col_1 <pk>
Col_2 <pk>
Col_3] -- "Col_1 = Col_1" --> T2[Table_2
Col_1 <fk>] T1 -- "Col_2 = ?" --> T2 </pre>	Col_1 is reused and Col_2 is not created
Not selected	Not selected	<pre> graph LR T1[Table_1
Col_1 <pk>
Col_2 <pk>
Col_3] -- "Col_1 = ?" --> T2[Table_2
Col_1] T1 -- "Col_2 = ?" --> T2 </pre>	No column is reused and no column is created

4.3.8. Định nghĩa các tham chiếu trong PDM (6)

Matching child table column is already a FK column

The following table shows the results of migrating primary key columns to a child table that contains a matching child table column that is already a foreign key column for another table. The original two tables are also shown below:



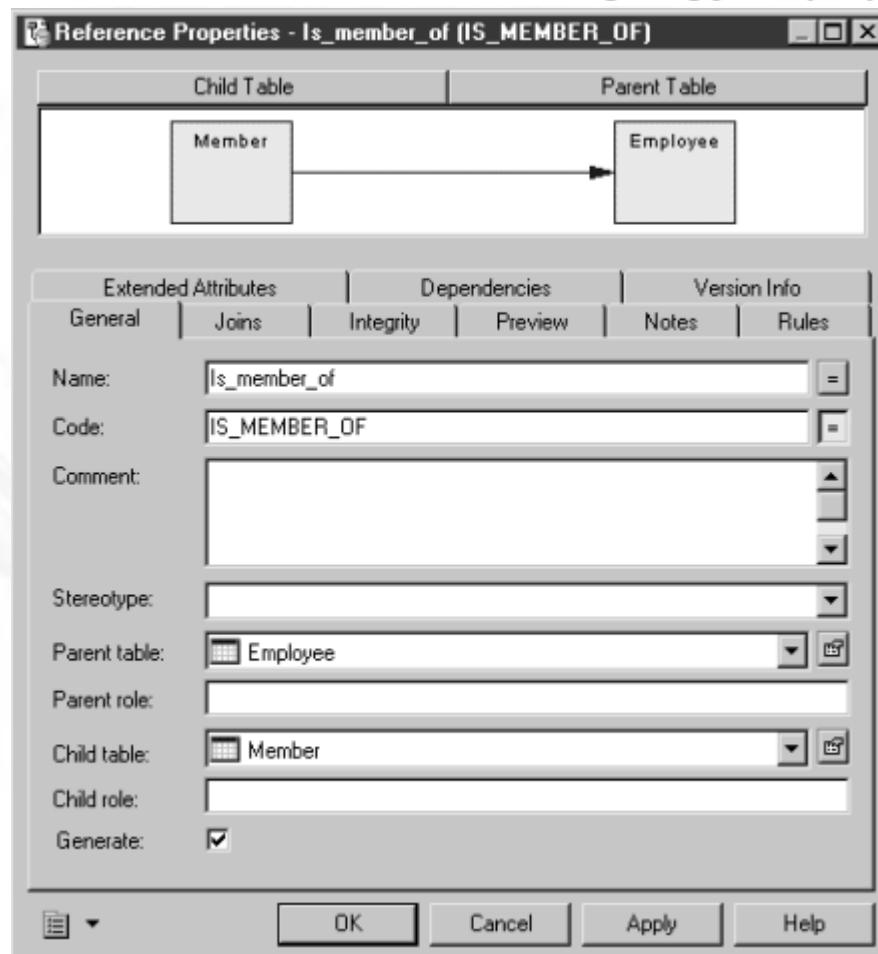
Auto-reuse	Auto-migrate	Result	Description of child table
Selected	Selected		T1_Col_1 is created and Col_2 is created
Unselected	Selected		T1_Col_1 is created and Col_2 is created
Selected	Not selected		No columns are reused or created
Unselected	Unselected		No columns are reused or created

4.3.8. Định nghĩa các tham chiếu trong PDM (7)

Creating a reference

You can create a reference that links a primary key, or alternate key, to a foreign key, or user-specified columns in both parent and child tables.

When you create a reference, its properties depend on the current model options, and the existence, or non-existence, of a foreign key column in the child table with the same code as a migrating primary key column.

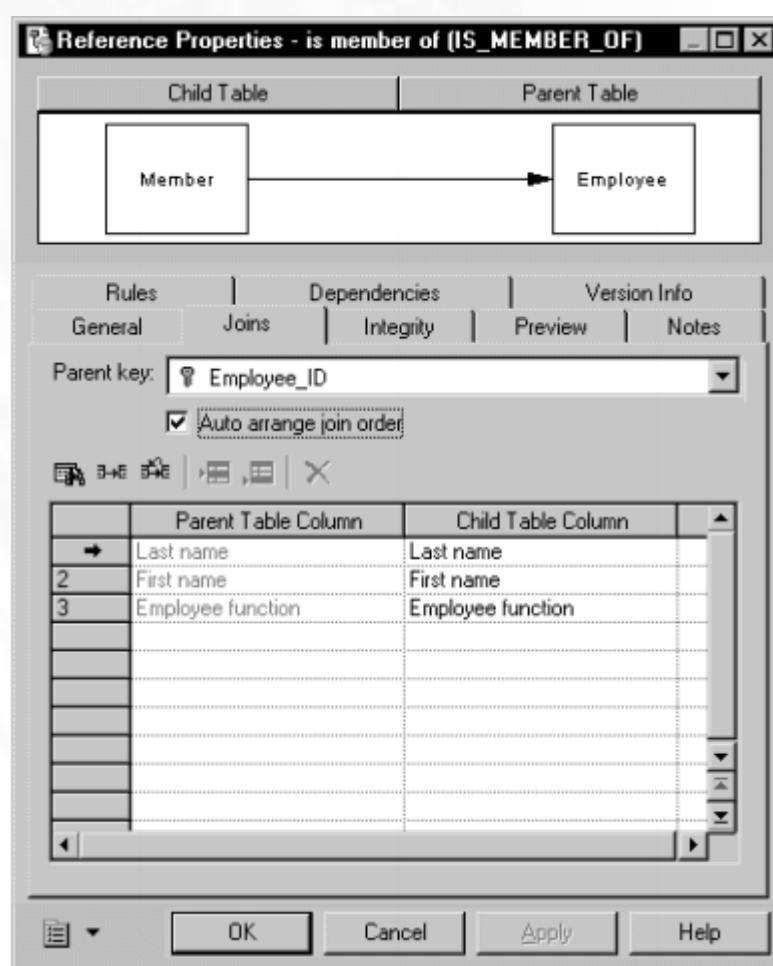


4.3.8. Định nghĩa các **tham chiếu** trong PDM (8)

Defining reference joins

A **join** is a link between a column in a parent table and a column in a child table (column pair) that is defined within a reference.

A join can link primary key, or alternate key, and foreign key columns, or user specified columns in the parent and child tables that are independent of key columns.



4.3.8. Định nghĩa các tham chiếu trong PDM (9)

Rebuilding references

Rebuilding references in a Physical diagram creates default references between PK columns in a table and columns with identical code and data type in another table. Note that rebuild is not possible between two tables with PK columns.

Rebuilding references is useful following the reverse engineering of a database in which all of the references could not be reverse engineered.

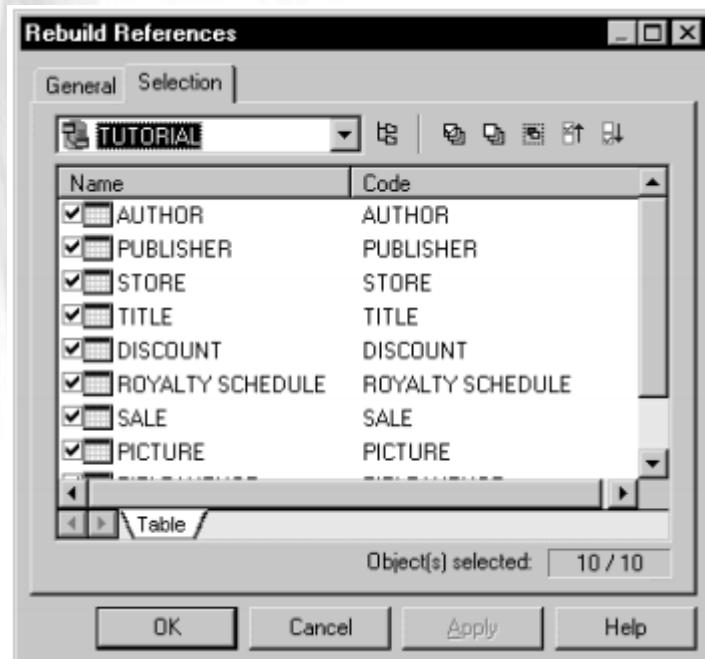
You can choose to rebuild references in one of two ways:

Rebuild option	Description
----------------	-------------

Delete and rebuild All existing references are deleted, and new references built based on matching key columns

Preserve All existing references are kept, and new references built based on new matching key columns

You can choose to rebuild all references in your model, or select the tables that are linked by the references that you want to rebuild.



4.3.8. Định nghĩa các tham chiếu trong PDM (10)

Using referential integrity

Referential integrity is a collection of rules that govern data consistency between primary keys, alternate keys and foreign keys.

Referential integrity dictates what happens when you update or delete a value in a referenced column in the parent table, and when you delete a row containing a referenced column from the parent table.

You can implement referential integrity using one of two methods:

Referential integrity	Description
Declarative	Referential integrity constraints are defined for particular references. When the reference is generated the target DBMS evaluates the reference validity and generates appropriate error messages
Using triggers	Referential integrity constraints are implemented by triggers based on the integrity constraints defined in the reference property sheet. The trigger evaluates reference validity and generates appropriate user-defined error messages

4.3.8. Định nghĩa các tham chiếu trong PDM (11)

Referential integrity properties

The definition of referential integrity includes the following properties:

Property	Description
Constraint name	Name of the referential integrity constraint. Maximum length is 254 characters
Implementation	Indicates whether implementation of referential integrity is declarative, or by the use of triggers
Cardinality	Indicates the maximum and minimum number of instances in a child table that can appear for each corresponding instance in the parent table
User-defined	Indicates a user-defined constraint name
Update constraint	How updating a key value, in the parent table affects the foreign key value in the child table
Delete constraint	How deleting a row in the parent table affects the child table
Mandatory parent	Each foreign key value in the child table must have a corresponding key value, in the parent table
Change parent allowed	A foreign key value can change to select another value in the referenced key in the parent table
Check on commit*	Verifies referential integrity only on the commit, instead of verifying it after row insertion. You can use this feature to control circular dependencies

* Only available for Sybase SQL Anywhere 5.0 and 5.5.

91

91/16

4.3.8. Định nghĩa các tham chiếu trong PDM (12)

Cardinality

Defining cardinality allows you to control the minimum and maximum number of children permitted for each parent. You can use the following values to define cardinality:

Cardinality	Minimum	Maximum
0	A parent does not have a minimum number of children. Child is optional	—
1	At least one child must exist for each parent	Only one child can exist for each parent
any integer	Minimum number is the number indicated	Maximum number is the number indicated
N or *	—	Any number of children can exist for a each parent

4.3.8. Định nghĩa các tham chiếu trong PDM (13)

Cardinality format	Cardinality can be expressed in the following formats:		
	Format	Description	Example
	x..y	Minimum and maximum interval, where x is any integer, and Y can be any of the following: Integer n (infinite) * (infinite)	(2..10) There must exist between two and ten children for each parent (0..n) Child is optional, and there can be any number of children for any one parent 10* There must exist at least ten children for each parent
	y	Maximum value which is equivalent to the form (0..y)	(10) There can be up to ten children for each parent. Child is optional
	x..y, x..y	Series of intervals, where the cardinality can be any one of the intervals	(1..2, 4..n) There can exist either one or two children for each parent, or at least four children for each parent

4.3.8. Định nghĩa các tham chiếu trong PDM (14)

Update and delete constraints

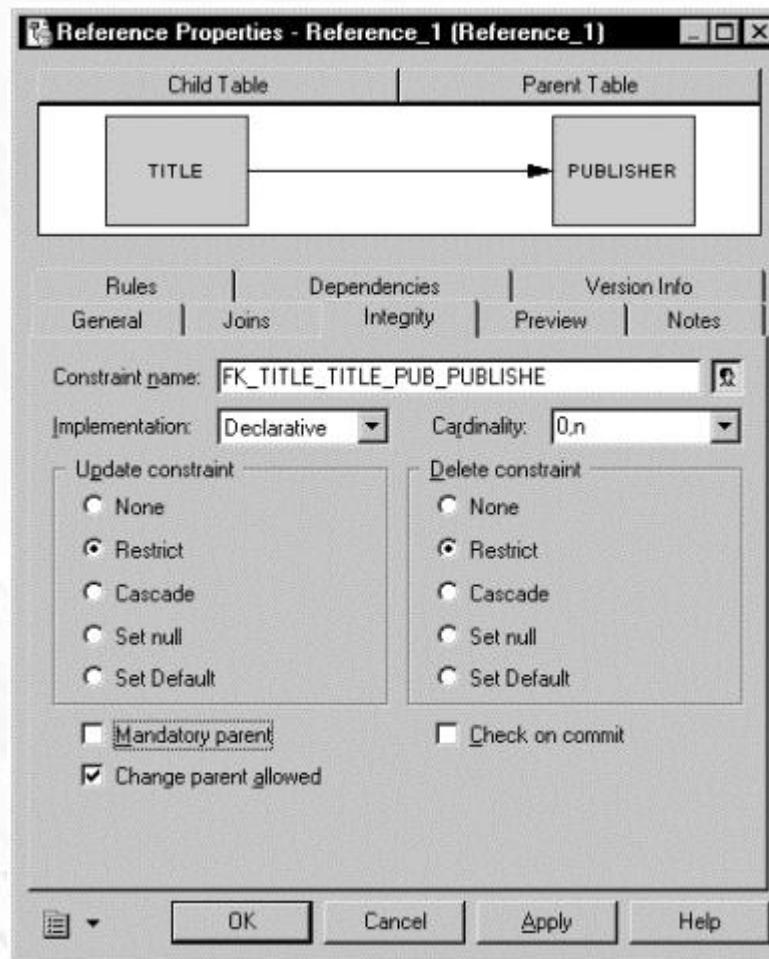
Update constraint and delete constraint take any of the following values:

Value	Change to parent table	Result in child table
None	Update or delete parent value	None
Restrict	Cannot update or delete parent value if one or more matching child values exist	None
Cascade	Update or delete parent value	Update or delete matching child values
Set null	Update or delete parent value	Set matching child values to NULL
Set default	Update or delete parent value	Set matching child values to default value

4.3.8. Định nghĩa các tham chiếu trong PDM (15)

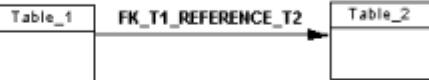
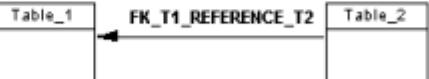
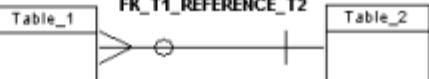
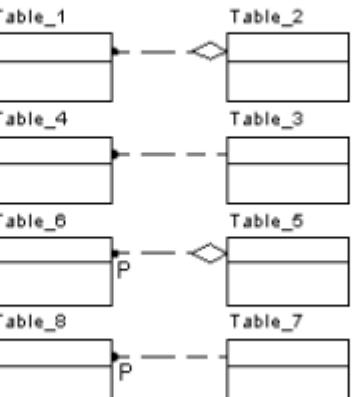
Defining referential integrity

You define referential integrity from the Integrity page of the reference property sheet.



4.3.8. Định nghĩa các tham chiếu trong PDM (16)

PowerDesigner supports the following notations:

Notation	Displays	Symbols
Relational	Arrow pointing to primary key	
CODASYL	Arrow pointing to foreign key	
Conceptual	Cardinality displayed in IE format (crow's feet)	
IDEF1X	Cardinality and mandatory status displayed on reference	

4.3.8. Định nghĩa các tham chiếu trong PDM (17)

Default Cardinality labels

The default cardinality labels for PowerDesigner indicate the minimum and maximum number of children as follows:

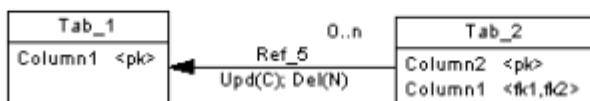
[minimum..maximum]

For example, the cardinality label [0..n] indicates that any number of children is acceptable.

Referential integrity example

The referential integrity label shown below indicates the following:

- ◆ Cascade on update
- ◆ Set null on delete
- ◆ Cardinality is 0..n (any number of children is acceptable).



Modifying a reference graphically

From the physical diagram graphic, you can modify a reference as follows:

- ◆ Bend a reference symbols
- ◆ Straighten a reference symbol
- ◆ Drag a reference to a different table
- ◆ Select reference display mode

4.3.9. Định nghĩa các **chỉ mục** trong PDM (1)

Defining indexes in a physical diagram

An index is a data structure associated with a table that is logically ordered by the values of a key. It improves database performance and access speed.

You normally create indexes for columns that you access regularly, and where response time is important. Indexes are most effective when they are used on columns that contain mostly unique values.

Example

In a table called Author, you create indexes for the primary key Author ID and the column Author name, but not for the column City. The values for city, are not likely to be unique, nor searched regularly, and do not help reduce query time.

4.3.9. Định nghĩa các **chỉ mục** trong PDM (2)

Index properties

You can access index properties from a table property sheet:

Property	Description
Name	Name of the index
Code	Reference name of the index
Comment	Descriptive label for the index
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of index owner. You choose an owner from a list of users, the index and table owners can be identical or different. An index can only have one owner at a time. This is normally the index creator
Table	Indicate table to index
Type	Proprietary index type (for Sybase IQ, and Oracle only)
Unique	Indicate whether an index is a unique index
Cluster	Indicate whether an index is a clustered index

4.3.9. Định nghĩa các chỉ mục trong PDM (3)

An index also includes the following properties:

Property	Description
Columns	Columns associated with index. This includes column definition indicating the primary key, foreign key, or alternate key to index. The index is associated with the columns of the selected key
Options	Physical options for the index. These are DBMS specific
Rules	Business rules attached to the index

Index types

The following index types exist:

One clustered index per table

A table cannot have more than one clustered index.

Type	Description
Bitmap	(Oracle) In a bitmap index, a bitmap for each key value is used instead of a list of row Ids
HG	(Sybase IQ) HighGroup indexes are used for GROUP BY, COUNT(DISTINCT) and SELECT DISTINCT statements when data has more than 1000 unique values
HNG	(Sybase IQ) HighNonGroup indexes make equality comparisons, SUM and AVG calculations very fast when data has more than 1000 unique values. Nonequality comparisons can also be done
LF	(Sybase IQ) LowFast indexes are used for columns that have a very low number of unique values. This index also facilitates join index processing. It is one of the two indexes allowed for columns used in join relationships
CMP	(Sybase IQ) Compare indexes are used for columns that store the binary comparison (<, >, or =) of any two distinct columns with identical data types, precision, and scale
WD	(Sybase IQ) Is used to index keywords by treating the contents of a CHAR or VARCHAR column as a delimited list

100

100/1

4.3.9. Định nghĩa các chỉ mục trong PDM (4)

Creating an index

You can either create a user defined index which is associated with one or more columns, or create an index that you link to a primary key, alternate key, or foreign key.

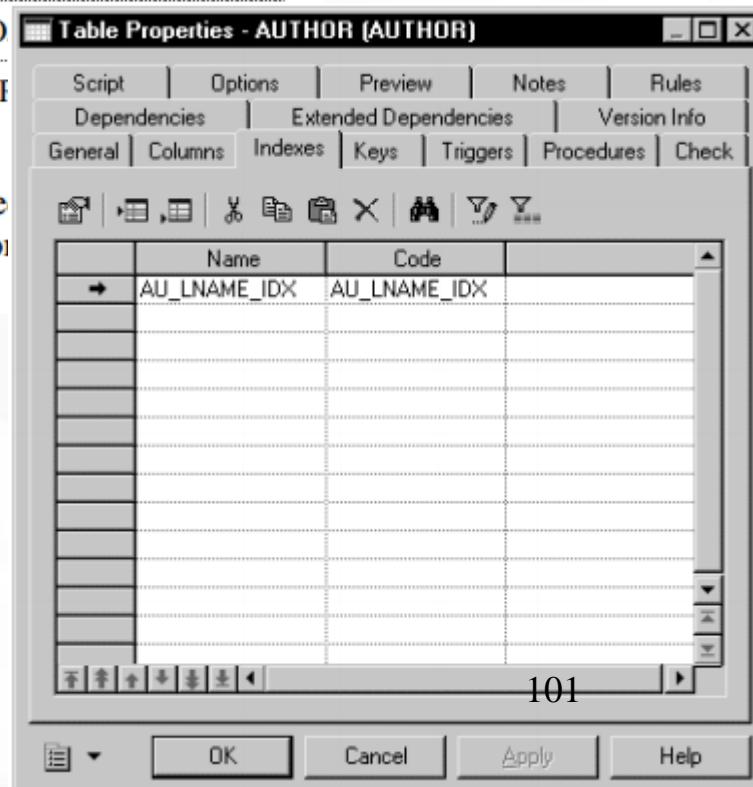
Index naming conventions

Use the following naming conventions for indexes:

Index	Naming convention
Primary key	Table code followed by PK; for example EMPLOYEE _PK
Foreign key	Table code followed by FK; for example PRO
Alternate key	Table code followed by AK; for example EMI

Index ascending or descending order

You can indicate the index order for each column attached selecting Ascending or Descending in the Sort column found on the Columns page of the Index property sheet.



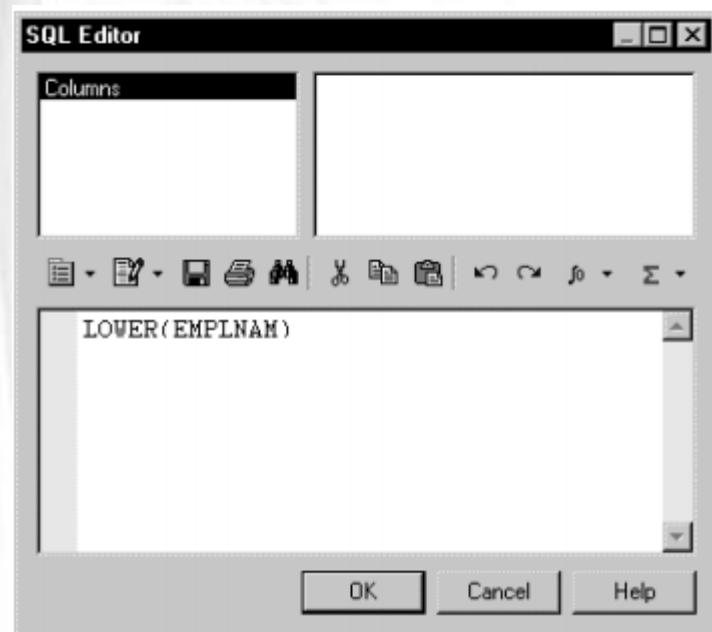
4.3.9. Định nghĩa các chỉ mục trong PDM (5)

Creating a function-based index

In some DBMS, you can create indexes on functions and expressions that involve one or more columns in the table being indexed. A function-based index precomputes the value of the function or expression and stores it in the index. The function or the expression will replace the index column in the index definition.

Function-based indexes provide an efficient mechanism for evaluating statements that contain functions in their WHERE clauses.

Example



Function-based indexes defined on UPPER(column_name) or LOWER(column_name) can facilitate case-insensitive searches.

You want to define an index that will put all names in lowercase on the table EMPLOYEE in order to ease search. You can define the following index (syntax for Oracle 8i):

```
CREATE INDEX low_name_idx ON EMPLOYEE (LOWER (EMPLNAM))
```

Then the DBMS can use it when processing queries such as:

```
SELECT * FROM EMPLOYEE WHERE LOWER (EMPLNAM) = "brown"
```

4.3.9. Định nghĩa các chỉ mục trong PDM (6)

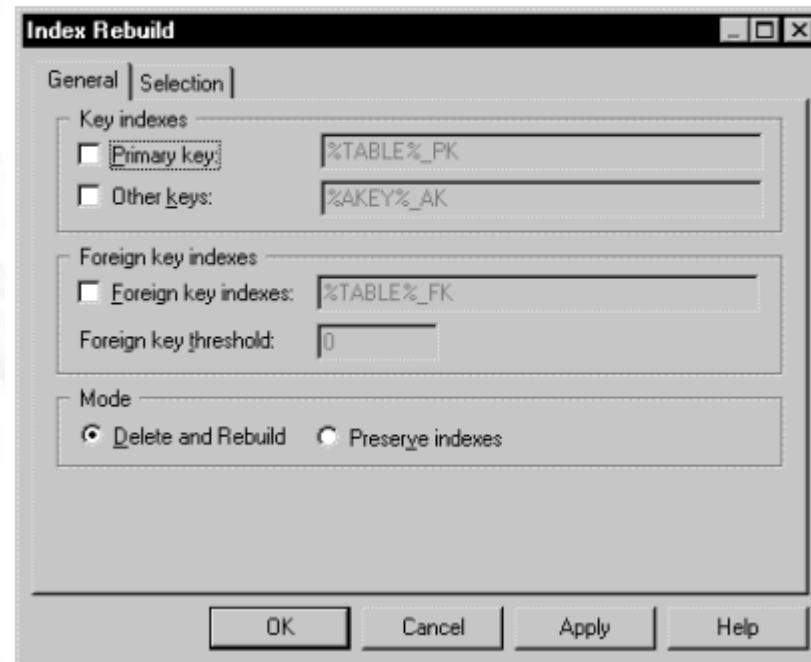
Rebuilding indexes

Rebuilding indexes in a physical diagram automatically updates any changes that you have made to primary keys, foreign keys, or alternate keys in your model.

You can rebuild indexes for one or more tables in a physical diagram. You can also choose to rebuild indexes from the model, or from a package.

Select Tools→Rebuild Objects→Rebuild Indexes.

The Rebuild Indexes dialog box appears.



4.3.10. Định nghĩa các **view** trong PDM (1)

Defining views in a physical diagram

A view is an alternative way of looking at the data in one or more tables. It is made up of a subset of columns from one or more tables.

You define a SQL query for each view.

4.3.10. Định nghĩa các view trong PDM (2)

View properties

You can double-click any view symbol in a diagram to display its properties:

Property	Description
Name	Name for the view
Code	Code for the view. This code is generated in database scripts
Comment	Descriptive label for the view
Stereotype	Sub-classification used to extend the semantics of an object without changing its structure; it can be predefined or user-defined
Owner	Name of view owner. You choose an owner from a list of users. A view can only have one owner at a time. This is normally the view creator
Usage	Defines the use of the view: Query only defines a view for consultation only, view cannot update tables; Updatable defines a view for consultation and update, view can update tables; and With Check options implements controls on view insertions
Dimensional type	Indicates the multidimensional type of the view, that is Dimension or Fact
Type	For those DBMS that support it, allows you to define the type of a view. You can select materialized query table, materialized view or summary table
Generate	Includes view generation as part of database generation script
User-defined	When selected, makes sure the view query is not parsed by PowerDesigner internal parser. This protects the view query from any update using model objects and keeps its syntax as defined by user. Otherwise, the view query is parsed and modified according to model values

105

105/1

4.3.10. Định nghĩa các view trong PDM (3)

A view also includes the following properties:

Property	Description
Columns	Columns of the view
Triggers	Triggers of the view
SQL Query	SELECT statements defining the view
Preview	SQL preview of view definition

View columns

The Columns page in a view property sheet displays the list of columns in the view. This list of columns reflects the SELECT orders from the queries of the current view. The only way to add or remove columns from this list is to modify the query of the view.

4.3.10. Định nghĩa các view trong PDM (4)

If the view was created from one or several tables or views The name, code, description and data type of the view column are those of the corresponding column in the linked table or view.

If the view is user-defined It implies the view is not linked to another object. The name and code of the view column comes from the column name in the first query in the view definition. For example, MyView is defined by the following queries:

```
select Name, Comment  
from Property  
union  
select Signature, Body  
from Method
```

Only the two columns of the first query are used to create the corresponding view columns:

MyView
Name
Comment
<input type="checkbox"/> Property
<input type="checkbox"/> Method

In this case, if you modify the view column code, the view creation script will reflect the change. In our example, if you rename Name in ClientName, the view creation script is the following:

```
create view MYVIEW (ClientName, "Comment") as  
select Name, Comment  
from Property
```

4.3.10. Định nghĩa các view trong PDM (5)

View triggers

You can define triggers on views for those DBMS that support this feature.

You create a trigger on a view when you want the view to behave like a table; this feature allows you to define a trigger to fire when one or more attributes of a table view column are modified.

The triggers you can define on a view are triggers of type "instead of".

Creating a view

There are several ways to create a view:

- ◆ Create a view for tables and views selected in the diagram
- ◆ Create an empty view then select tables and views from a list of tables and views

At creation, a view has a default name including a number, this number is assigned in the order of creation of objects.

4.3.10. Định nghĩa các view trong PDM (6)

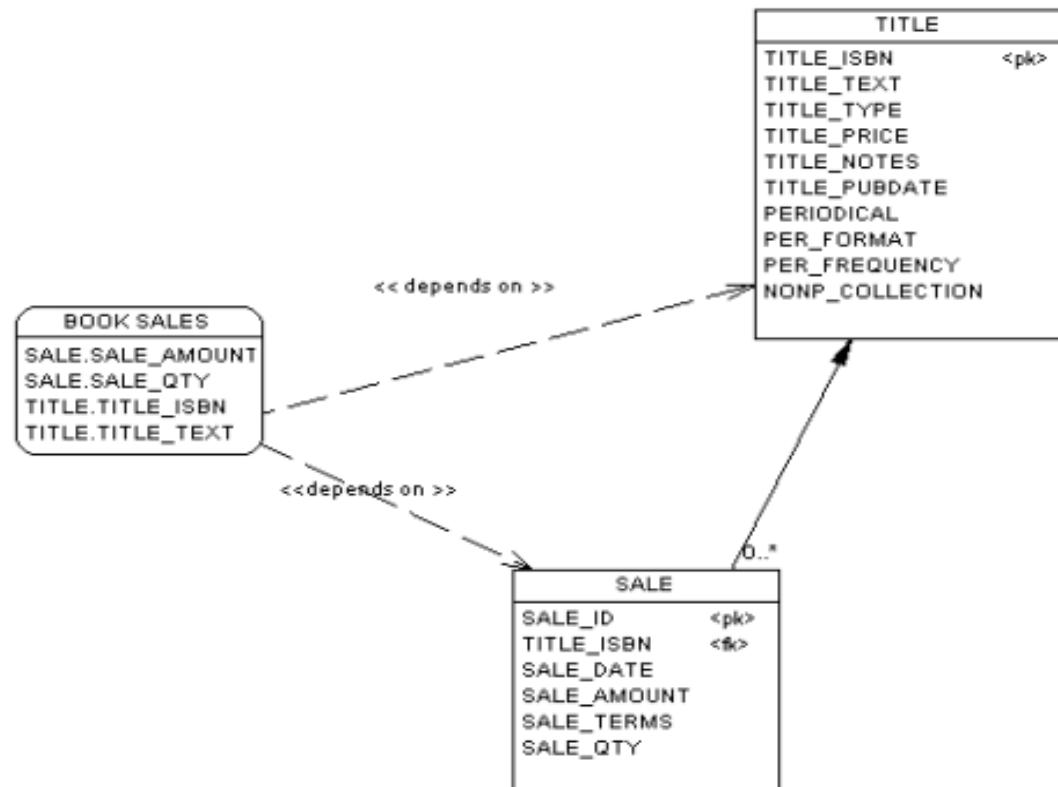
Using extended dependencies for views

Extended dependencies are links between physical diagram objects. These links help to make object relationships clearer between model objects but are not interpreted and checked by PowerDesigner as they are meant to be used for documentation purposes only.

You can complement these links by applying stereotypes.

Example

A view identified as Book Sales can have two extended dependencies indicating that the view depends on the Title and Sale tables. The diagram displays their extended dependencies and stereotypes.



4.3.10. Định nghĩa các view trong PDM (7)

Defining a query for a view

You define queries for a view from the SQL Query page of the view property sheet. Each SELECT query is shown in the Query dropdown listbox. The definition for the selected query appears in the textbox.

Creating relations between selected tables

Depending on the selected DBMS, you can construct links between tables selected from the Query dropdown listbox. These links allow you to create more complex queries.

Option	The resulting query	Example
Union	Binds two or more SELECT statements. The query result displays all the retrieved data. Repeating query results are not displayed	SELECT 1: ABC SELECT 2: BCD Result is ABCD
Union All	Binds two or more SELECT statements. The query result displays all the retrieved data. Repeating query results are displayed	SELECT 1: ABC SELECT 2: BCD Result is ABCBCD
Intersect	Binds two or more SELECT statements. The query result displays all the retrieved data that is common to the SELECT statements	SELECT 1: ABC SELECT 2: BCD Result is BC
Minus	Binds two or more SELECT statements. The query result displays the difference between all the retrieved data that is common to the consecutive SELECT statements (SELECT 1-SELECT 2-...)	SELECT 1: ABC SELECT 2: BCD Result is AD

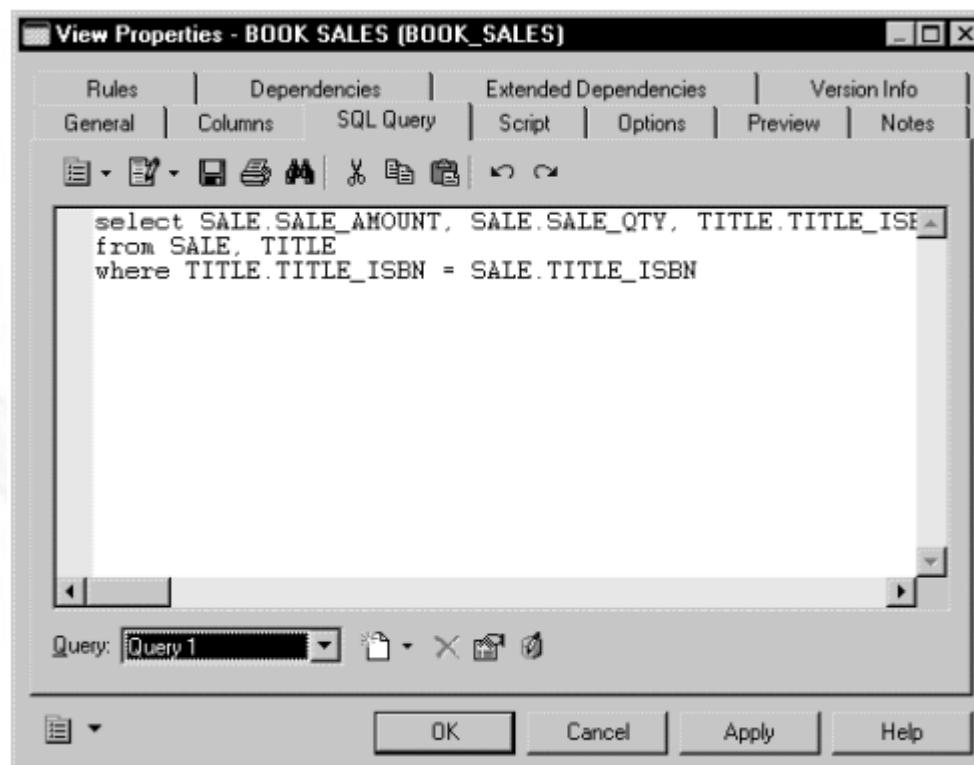
4.3.10. Định nghĩa các view trong PDM (8)

Creating a query
with the SQL editor

You can also construct a query using the SQL Editor. The SQL Editor provides a more complete query definition environment than the SQL query page.

In addition to standard SQL statements (SELECT, FROM, WHERE), the SQL Editor offers greater flexibility through syntax tools for functions and operators.

You can open the SQL Editor by clicking the Edit tool in the SQL Query toolbar.



4.4. Xây dựng biểu đồ VẬT LÝ DỮ LIỆU đa chiều

4.4.1. **Tổng quan** về PDM đa chiều

4.4.2. Định nghĩa các **chiều** (dimension) trong PDM đ.chiều

4.4.3. Định nghĩa các **thuộc tính** (attribute) trong PDM đ.chiều

4.4.4. Định nghĩa các **kế thừa** trong PDM đ.chiều

4.4.5. Định nghĩa các **khối** (cube) trong PDM đ.chiều

4.4.6. Định nghĩa các **dữ kiện** (fact) trong PDM đ.chiều

4.4.7. Định nghĩa các **measure** (key) trong PDM đ.chiều

4.4.8. Định nghĩa các **liên kết** (association) trong PDM đ.chiều

4.4.1. Tổng quan về PDM đa chiều (1)

Defining a multidimensional diagram

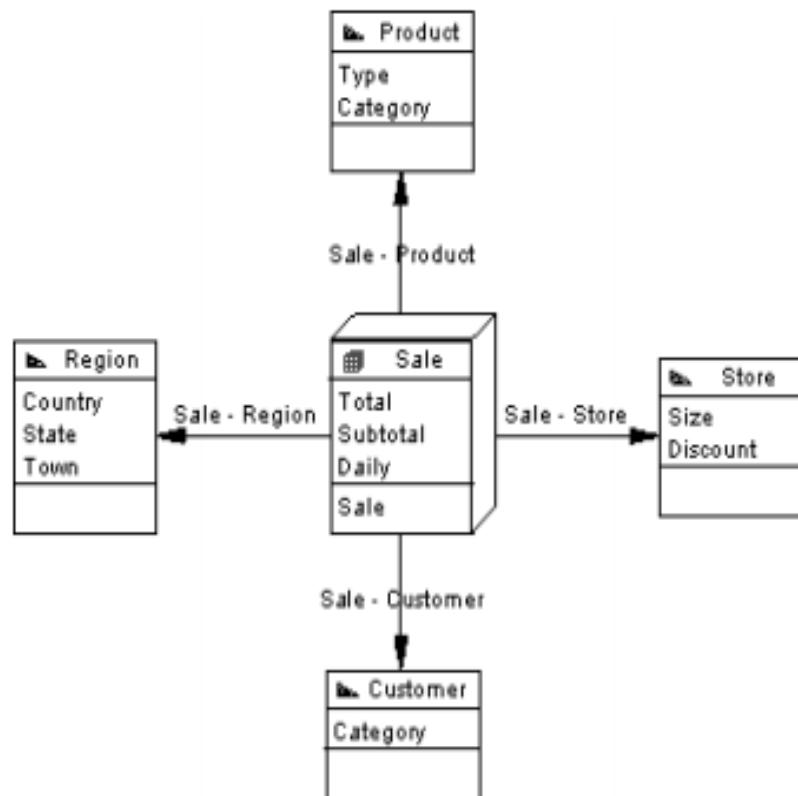
A **multidimensional** diagram is a model of business activities in terms of cubes and dimensions. It organizes business data into one or other of these categories. Numeric values or measures such as sales total, budget limits, are the facts of the business. The area covered by a business, in terms of geography, time, or products are the dimensions of the business.

The multidimensional diagram is used to design an OLAP database. Information in an OLAP database is organized to facilitate the queries performed by different tools.

4.4.1. Tổng quan về PDM đa chiều (2)

Example

Sales data can have the dimensions product, region, customer, and store. Facts, for example, the sales totals, are viewed through the user-defined dimensions. When you retrieve the sales total of a particular product for a particular region, you are viewing the sales total through the product and region dimensions. The most common dimension is time because the purpose of multidimensional analytical queries is to find trends.



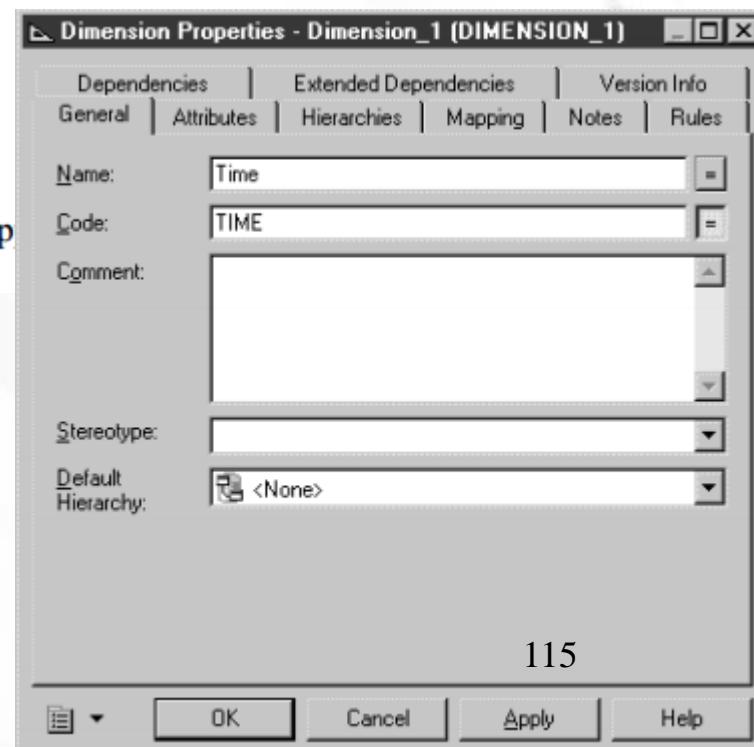
4.4.2. Định nghĩa các **chiều** trong PDM đ.chiều

Defining dimensions in a multidimensional diagram

A dimension is an axis of analysis in a multidimensional structure.

The dimension is made of an ordered list of attributes that share a common semantic meaning in the domain being modeled. Each attribute designates a unique position along the axis.

Time
Year
Quarter
Month
Week
Year_time <Default> <h>



The dimension can be mapped to tables or views: this map transfer operational data to the dimension.

4.4.3. Định nghĩa các **thuộc tính** trong PDM đ.chiều

Defining attributes in a multidimensional diagram

An **attribute** is used to qualify dimensions used in queries. For example, the Time dimension can contain attributes Year, Quarter, Month, and Week.

Time
Year
Quarter
Month
Week
Year_time <Default> <h>

Attributes can be organized in hi

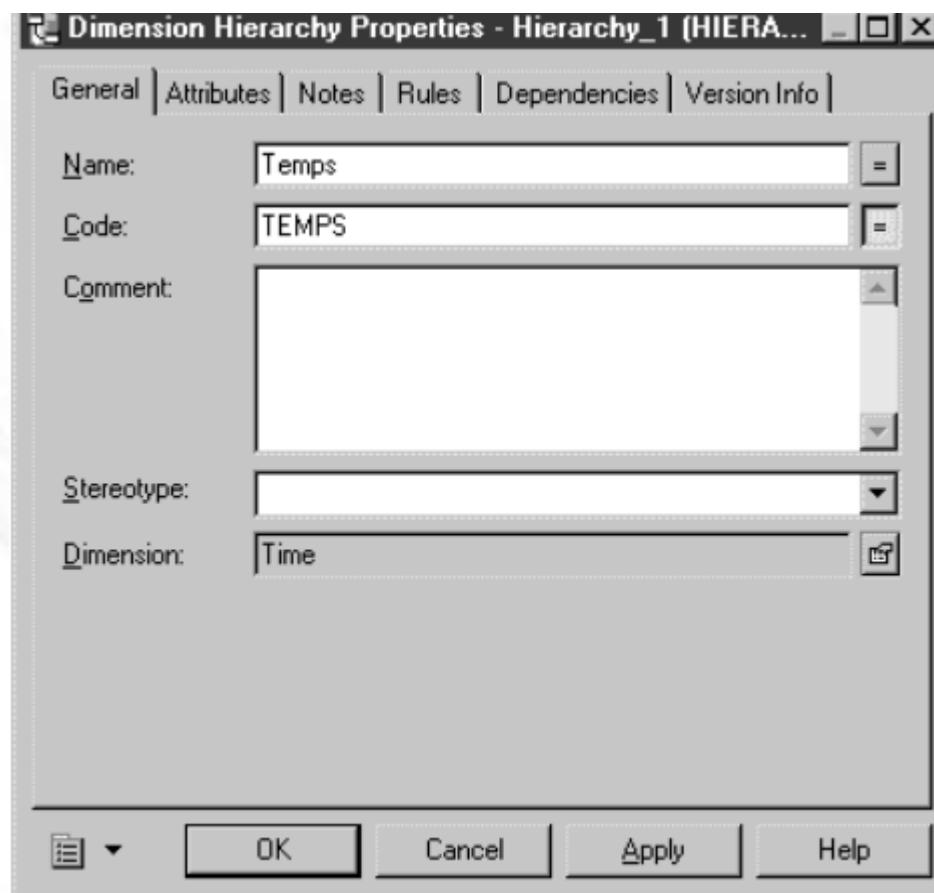


4.4.4. Định nghĩa các **kế thừa** trong PDM đ.chiều

Defining hierarchies in a multidimensional diagram

A **hierarchy** defines navigational and consolidation paths through dimensions.

As an **organizational** path, the hierarchy describes a traversal pattern through a dimension, from the most general to the most specific attribute of the dimension. It is an ordered subset of the attributes.

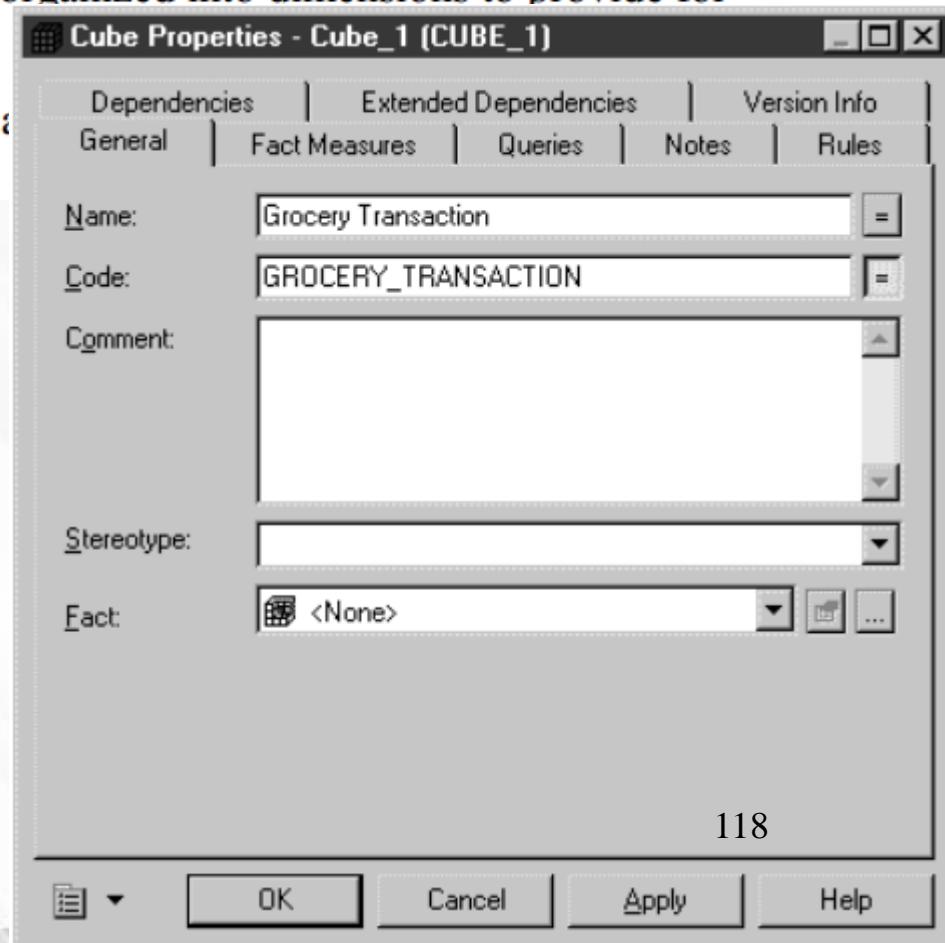


4.4.5. Định nghĩa các khối trong PDM đ.chiều

Defining cubes in a multidimensional diagram

A cube is a collection of measures corresponding to values stored into each of its data cell. The measures are organized into dimensions to provide for faster retrieval and drill-down.

Usually a cube is associated with a fact measure and other measures among cubes.



118

118/1

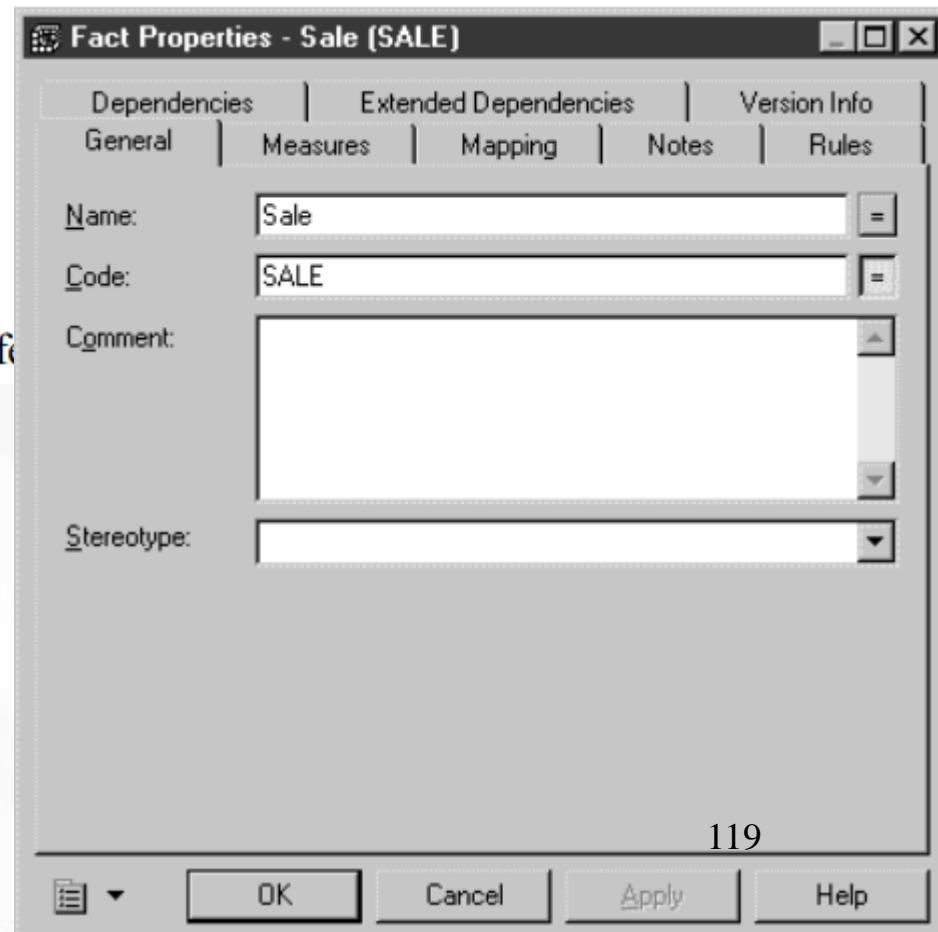
4.4.6. Định nghĩa các dữ kiện trong PDM đ.chiều

Defining facts in a multidimensional diagram

A fact corresponds to the focus of a decision support investigation. It is a set of measures manipulated by the cube. For example, Sale, Revenue, Budget could be facts.



Facts can be reused among different cubes.



119

119/1

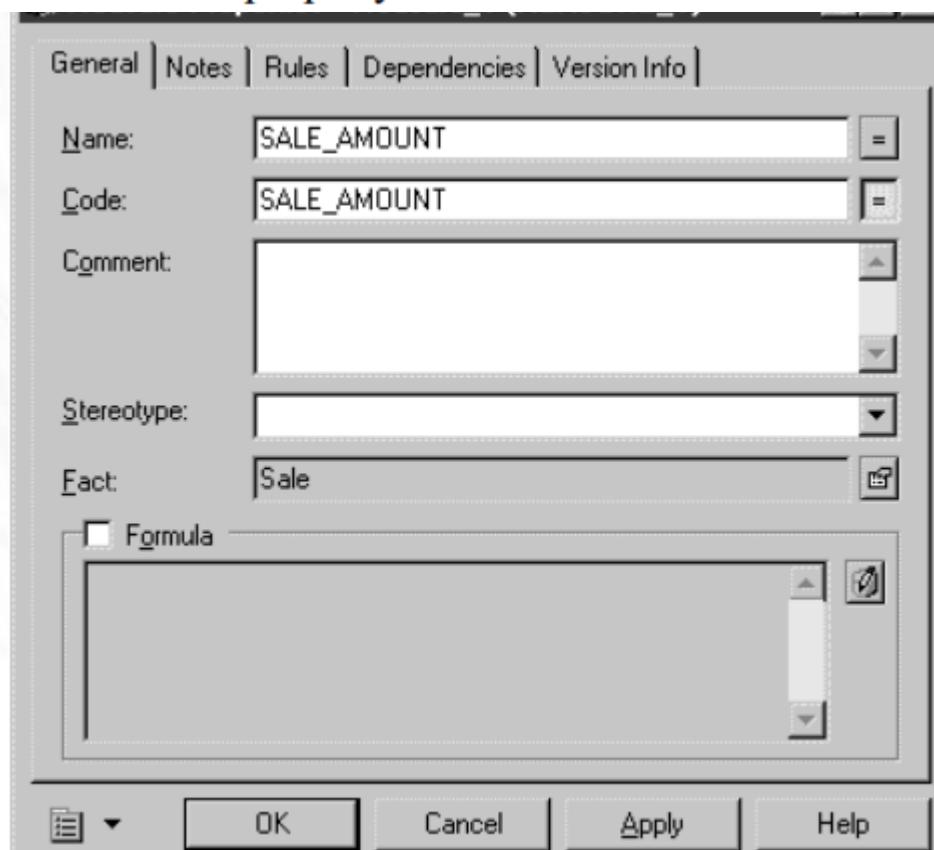
4.4.7. Định nghĩa các measure trong PDM đ.chiều

Defining measures in a multidimensional diagram

A measure is a variable that corresponds to the focus of an investigation. Measures describe the meaning of the analytical values stored in each data cell of the cube.

Measures are most of the time numeric values like for example Price or Total.

Measures can also be the result of an operation or calculation as indicated in the formula box of the measure property sheet.

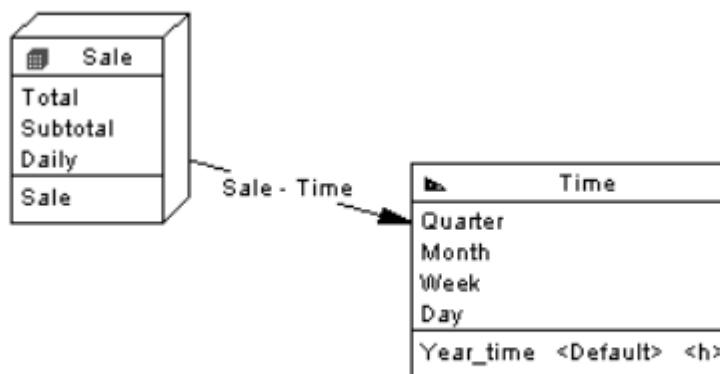


4.4.8. Định nghĩa các **liên kết** trong PDM đ.chiều (1)

Defining associations in a multidimensional diagram

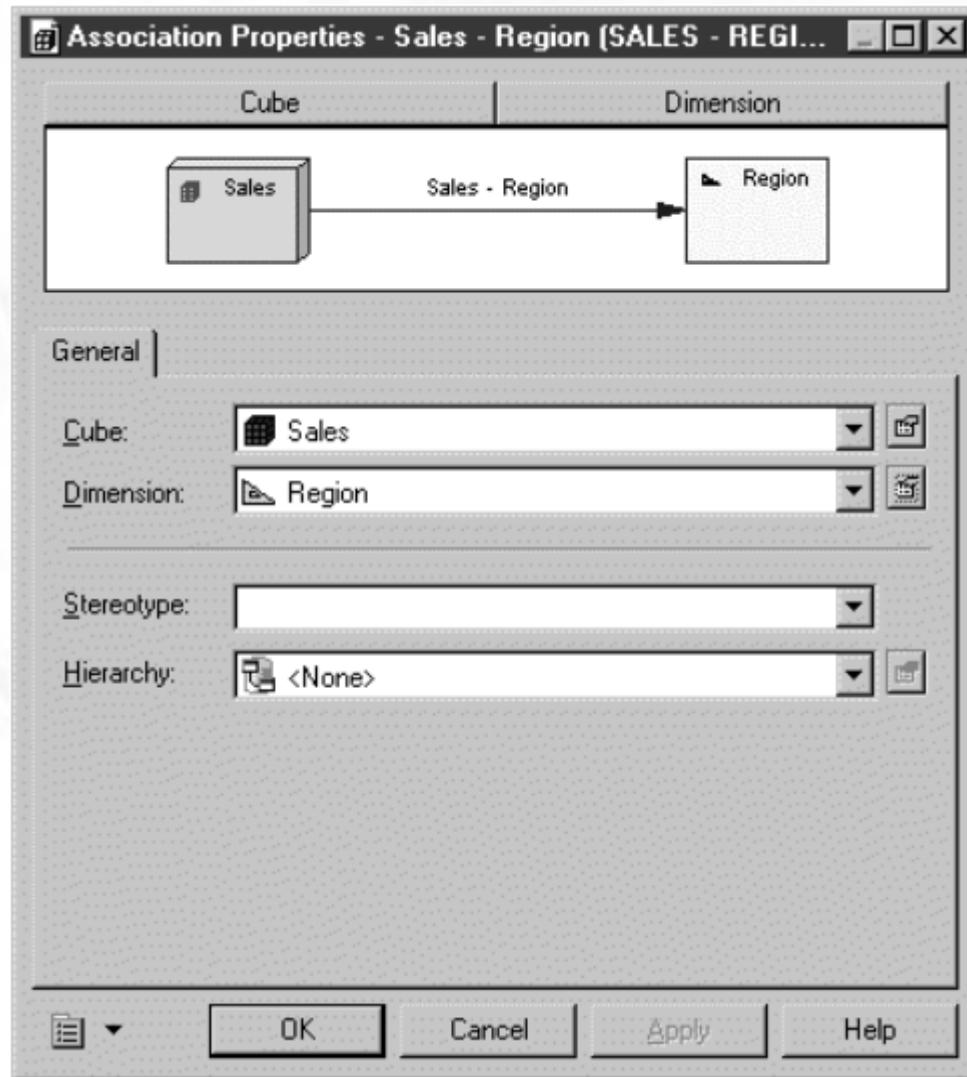
An association relates a cube to the dimension that defines it. It shows the axis of investigation of the dimension in the cube.

For example, the Sale cube is linked to the Time dimension by the Sale - Time association to analyze sales through the time dimension.



There can be only one association between a cube and a dimension.

4.4.8. Định nghĩa các liên kết trong PDM đ.chiều (2)



122

122/1

4.5. Làm việc với mô hình VẬT LÝ DỮ LIỆU

4.5.1. Kiểm tra PDM

4.5.2. Các **đối tượng tham số** được kiểm tra

4.5.3. Mapping các **đối tượng** trong PDM

4.5.4. Retrieving các **đối tượng** trong PDM

4.5.5. Rebuilding các **khối** (cube) trong PDM

4.5.6. Tạo sinh các **script** (script) trong PDM

4.5.1. Kiểm tra PDM (1)

Checking a PDM

You can use the Check Model command to check the validity of a PDM at any time. A valid PDM conforms to the following rules:

- ◆ Each object name must be unique
- ◆ Model constraints must be compatible with database constraints
- ◆ Each table must have at least one column
- ◆ Each index must have a column
- ◆ Each reference must have at least one column pair

4.5.1. Kiểm tra PDM (2)

PDM check options

When you check a PDM, you can define levels of severity for problems that Check model finds and have certain problems automatically corrected.

Levels of problem severity

You can set a level of problem severity for each object parameter verified by Check model. This severity level depends on the degree of normalization that you want to achieve in your model.

When you check a PDM, you can display an invalid parameter with one of two types of messages, depending on the degree of problem severity you assigned to that parameter:

Message	Description
Error	Major problem that impedes database generation
Warning	Minor problem or recommendation

4.5.1. Kiểm tra PDM (3)

Automatic correction

You can choose to have certain problems corrected automatically by PowerDesigner. However, before using automatic correction, make sure you understand how it will affect your model.

For example, if a column code length is longer than the length specified in the **MaxColumnLen** field in the DBMS, then PowerDesigner can automatically truncate the code length to be within the specified length.

Tool	Indicates	Description
	Error	Indicates a major problem that prevents database generation
	Warning	Indicates a minor problem or recommendation
	Automatic correction	Allows automatic error correction

4.5.2. Các đối tượng TS được kiểm tra (1)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

Horizontal partitioning check

During a horizontal partitioning check, the following object controls are made.

Horizontal partitioning name and code uniqueness

Existence of partitions

Unavailable target table

4.5.2. Các đối tượng TS được kiểm tra (2)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

Table collapsing check

During a table collapsing check, the following object controls are made.

Table collapsing name and code uniqueness

Existence of target table

Unavailable target table

4.5.2. Các đối tượng TS được kiểm tra (3)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

Vertical partitioning check

During a vertical partitioning check, the following object controls are made.

Vertical partitioning name and code uniqueness

Existence of partitions

Unavailable target table

4.5.2. Các đối tượng TS được kiểm tra (4)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

Business Rule check in a PDM

During a business rule check, the following object controls are made.

Business rule name and code uniqueness

Unused business rule

4.5.2. Các đối tượng TS được kiểm tra (5)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

Package check

During a package check, the following object controls are made.

Circular reference

Constraint name uniqueness

Constraint name maximum length

Circular dependencies

Shortcut code uniqueness

4.5.2. Các đối tượng TS được kiểm tra (6)

Object parameters verified during check model

The Check Model command verifies the validity of PDM objects.

4.5.3. Mapping các đối tượng trong PDM (1)

Mapping objects in a PDM

Object mapping is the ability to establish a correspondence between objects belonging to heterogeneous models and diagrams.

In a PDM, you can link physical and multidimensional objects for the following purpose:

- ◆ **Relational to relational** mapping between physical objects to generate extraction scripts to populate a data warehouse with operational data
- ◆ **Relational to multidimensional** mapping between physical and multidimensional objects to generate text files containing cube data to fill the OLAP database

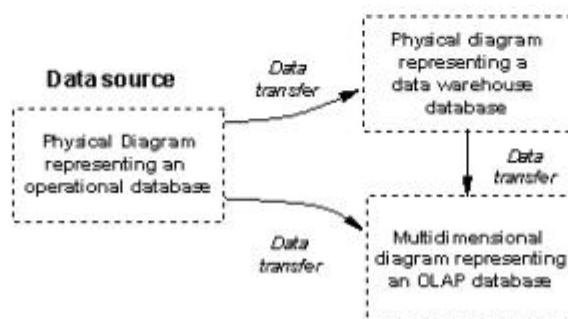
4.5.3. Mapping các đối tượng trong PDM (2)

Understanding object mapping

You create a mapping between PDM objects to setup a structure for data movement and transformation. Data comes from a **data source** and is loaded in another database.

In PowerDesigner, the data source and the destination database can be designed in physical data models.

The following schema illustrates the transfer of data between the different types of databases:



4.5.3. Mapping các đối tượng trong PDM (3)

Data transfer

The above diagram shows the different paths for data transfer:

- ◆ From operational to data warehouse databases
- ◆ From data warehouse to OLAP databases
- ◆ From Operational to OLAP databases. This path is less common.
Usually operational data need to be stored in a data warehouse database before they can be used in an OLAP engine. However, when there are no large amounts of data, it is possible to skip the data warehouse step and have a data transfer directly from the operational database to the OLAP engine. For clarity purpose, we will not develop this aspect of data transfer in this section

4.5.3. Mapping các đối tượng trong PDM (4)

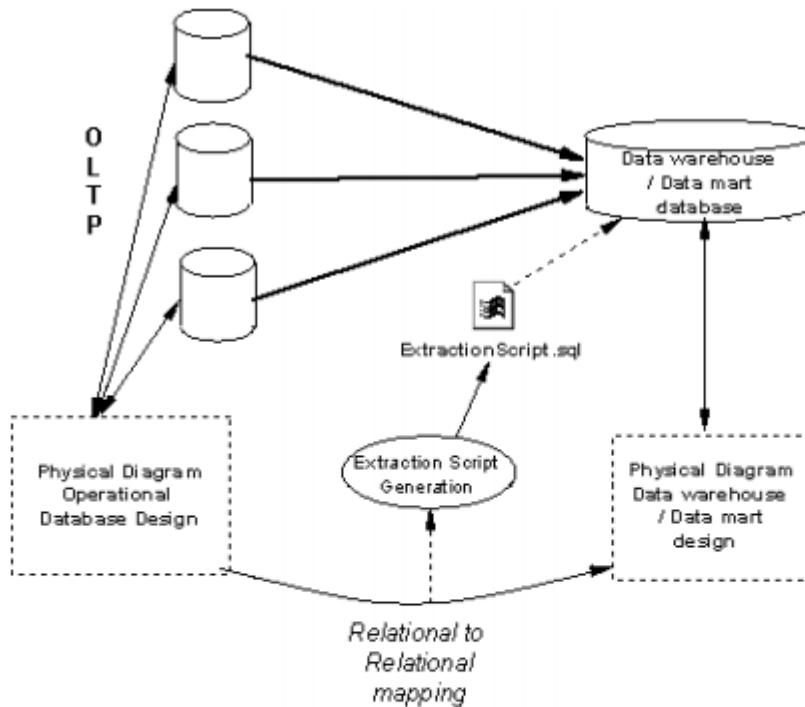
Defining data sources in a PDM

You create a data source to define where (in which database) data should be extracted to be transferred to another database.

When you define a data source, you have to declare physical data models in the list of data source models. A data source can contain several models, you can select the source models among a list of models opened in the current workspace. All the selected models represent the same database that contain operational data.

A model can contain several data sources.

4.5.3. Mapping các đối tượng trong PDM (5)



The following table lists object mapping in a relational to relational mapping:

Operational object	Data warehouse object
Table	Table (Fact or Dimension type)
Column	Column

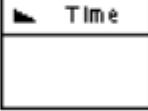
4.5.4. Retrieving các đối tượng trong PDM

Retrieving multidimensional objects

You will need to identify fact and dimension tables and views during the design of a data warehouse. Tables and views support multidimensional types, in this section both are referred to as **fact and dimension tables**.

A **fact table** stores variable numerical values related to aspects of a business. For example, sales, revenue, budget. These are usually the values you want to obtain when you carry out a decision support investigation.

A **dimension table** stores data related to the axis of investigation of a fact. For example, geography, time, product. A dimension table should be connected to a central fact table.

Fact table	Dimension table
	

4.5.5. Rebuilding các khối (cube) trong PDM

Rebuilding cubes

The Rebuild Cubes feature allows you to create cubes and dimensions from fact and dimension tables and views. Tables and views support multidimensional types, in this section both are referred to as **fact and dimension tables**

The rebuild cubes functionality modifies objects according to the following rule:

Physical object	After rebuild cubes, creates
Fact table	One fact with the same name as the fact table + One Cube
Column in a Fact table (except foreign keys)	Measure
Dimensions tables attached to the fact table	One dimension, the name of the dimension corresponds to the concatenation of the dimension tables along the path to the child table, from the furthest to the closest + One hierarchy that becomes the default hierarchy. This hierarchy contains the attributes corresponding to the primary key columns of the tables converted into a dimension
Column in a Dimension table (except foreign keys)	One attribute, the name of the attribute is the concatenation of the dimension table name and column name if column names are ambiguous. Otherwise the name is identical to the name of the column.
References between a fact and a dimension table	One cube dimension association

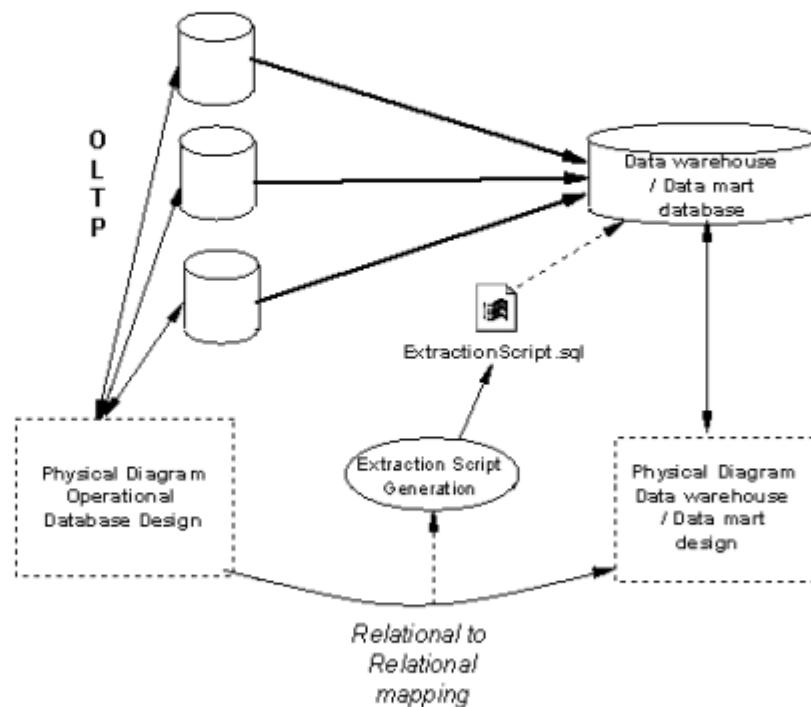
139

139/1

4.5.6. Tạo sinh các **script** (script) trong PDM (1)

Generating extraction scripts

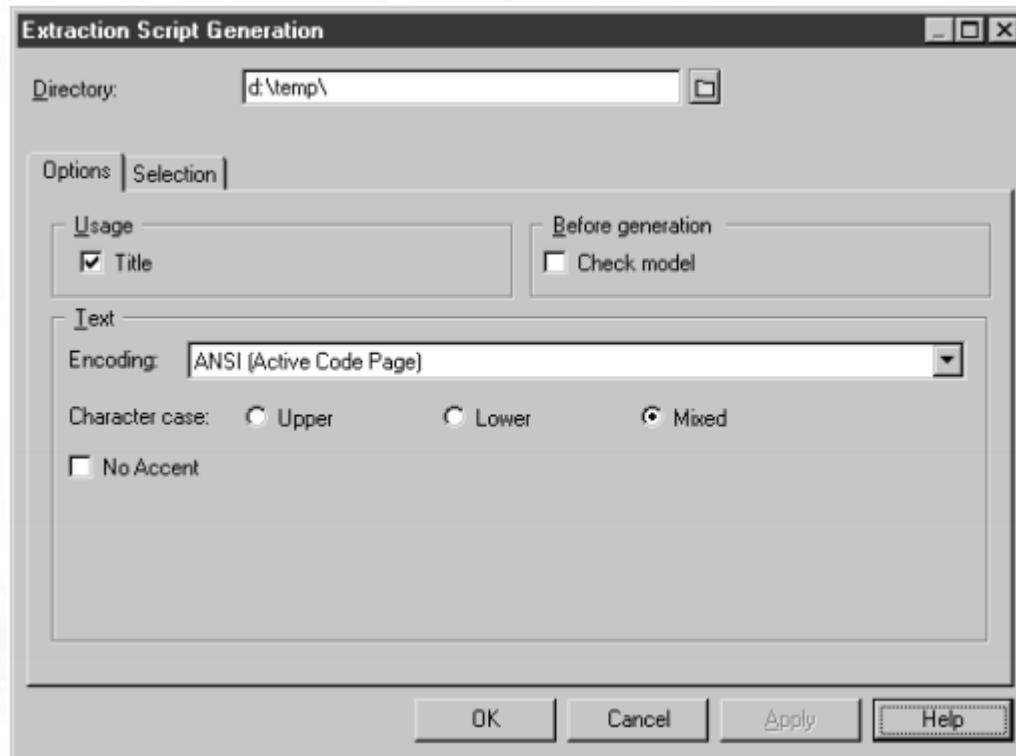
The Generate Extraction Script feature allows you to generate script files that will be used to fill and update tables in a data warehouse or data mart database.



The link between the operational database and the data warehouse or data mart database is a relational to relational mapping.

140

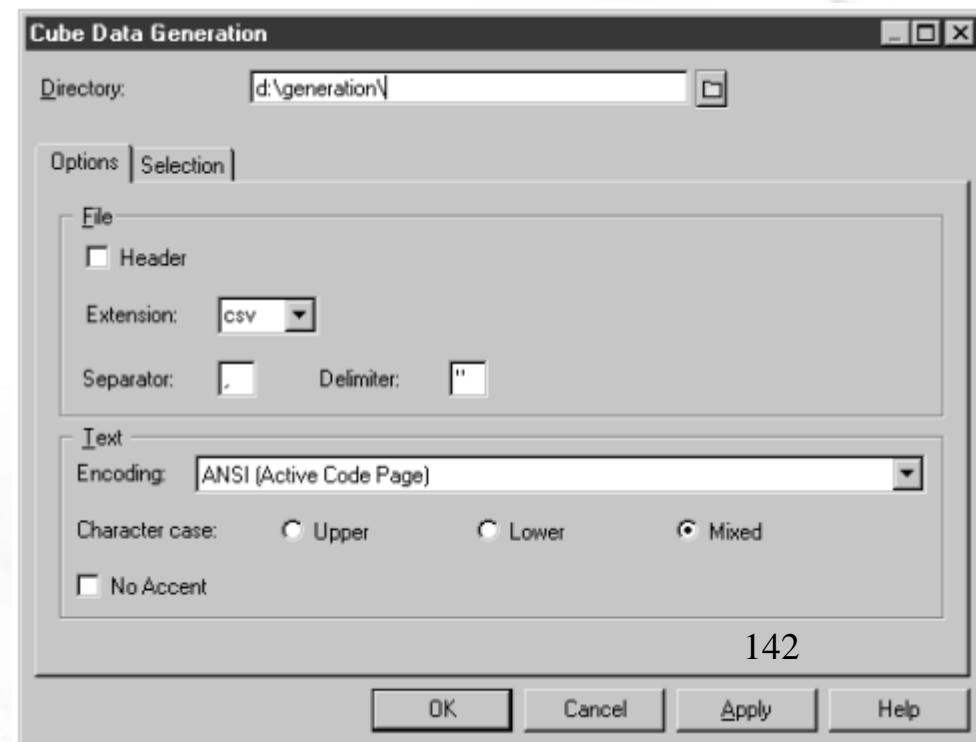
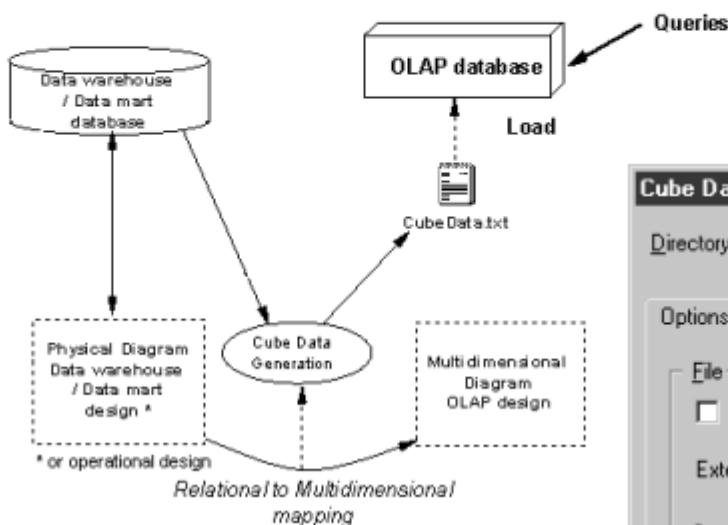
4.5.6. Tạo sinh các **script** (script) trong PDM (2)



4.5.6. Tạo sinh các script (script) trong PDM (3)

Generating Cube Data

The cube data generation feature allows the generation of a text file that will be used by an OLAP tool to create and populate cubes using data from operational sources.



4.6. Tạo và điều chỉnh CSDL

4.6.1. Sử dụng giao diện ODBC

4.6.2. Truy xuất CSDL

4.6.3. Cấu hình lưu trữ

4.6.4. Ước tính kích thước CSDL

4.6.5. Tùy chỉnh các đoạn mã script

4.6.6. Tạo sinh ra CSDL

4.6.7. Sử dụng dữ liệu test

4.6.1. Sử dụng giao diện ODBC (1)

Using the ODBC interface

The Open Database Connectivity (ODBC) interface allows PowerDesigner to access information in different database management systems (DBMS) using Structured Query Language (SQL) as a standard for accessing data.

There are three types of data sources:

Data source	Description
Machine	Data source is created on the client machine, and is available to the user currently logged onto the system. Machine data sources are stored in the part of the registry containing settings for the current user.
System	Data source is created on the client machine, and is available to all users regardless of whether a user is logged onto the system or not. System data sources are stored in the part of the registry containing settings for the current machine.
File	Data source is stored as a file. A file data source has the extension .dsn. It can be used by different users if it is placed in the default location for file data sources. File data sources are usually managed by database administrators.

4.6.1. Sử dụng giao diện ODBC (2)

Using the ODBC interface

The Open Database Connectivity (ODBC) interface allows PowerDesigner to access information in different database management systems (DBMS) using Structured Query Language (SQL) as a standard for accessing data.

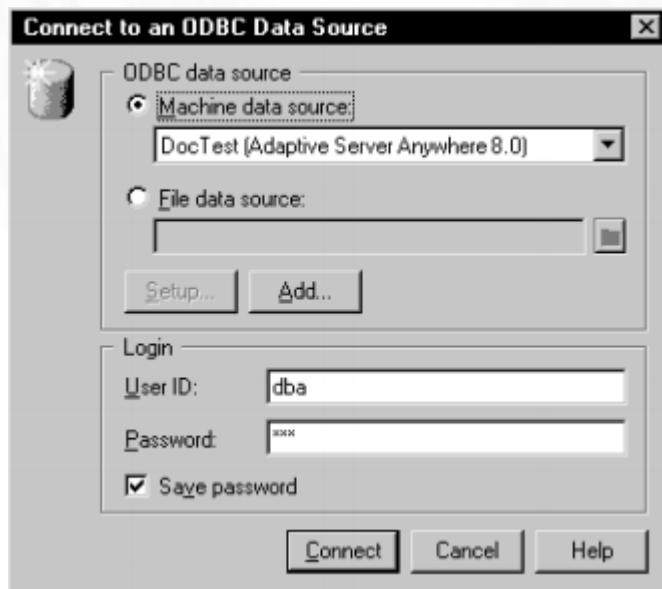
There are three types of data sources:

Data source	Description
Machine	Data source is created on the client machine, and is available to the user currently logged onto the system. Machine data sources are stored in the part of the registry containing settings for the current user.
System	Data source is created on the client machine, and is available to all users regardless of whether a user is logged onto the system or not. System data sources are stored in the part of the registry containing settings for the current machine.
File	Data source is stored as a file. A file data source has the extension .dsn. It can be used by different users if it is placed in the default location for file data sources. File data sources are usually managed by database administrators.

4.6.2. Truy xuất CSDL

Accessing a database

PowerDesigner allows you to change the target DBMS of your model, display data from the database that corresponds to your model, and send SQL queries to a connected data source.



The window displays several SQL alter statements:

```
alter table PROJ.EMPLOYEE
    delete foreign key FK_EMPLOYEE_CHIEF_EMPLOYEE;

alter table PROJ.COMPOSE
    delete foreign key FK_COMPOSE_COMPOSED_MATERIAL;

alter table PROJ.COMPOSE
    delete foreign key FK_COMPOSE_COMPOSES_MATERIAL;

alter table PROJ.PARTICIPATE
    delete foreign key FK_PARTICIP_IS_DONE_B_TASK;

alter table PROJ.MEMBER
    delete foreign key FK_MEMBER_IS_MEMBER_EMPLOYEE;

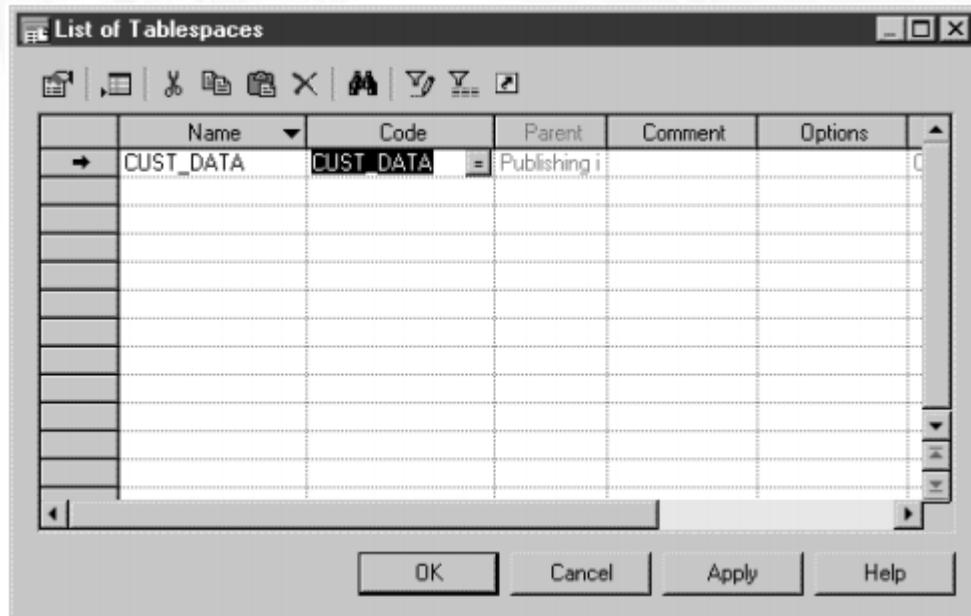
alter table PROJ.PROJECT
```

At the bottom are 'Run', 'Close', and 'Help' buttons. The status bar shows 'Ln 25, Col 1' and '146'.

4.6.3. Cấu hình lưu trữ

Configuring tablespace and storage

Tablespace and storage indicate the physical location of tables and indexes in a database or storage device. The definition of tablespace and storage is DBMS specific. Different DBMS can use different terms for tablespace and storage.



4.6.4. Ước tính kích thước CSDL (1)

Estimating database size

You can estimate the size of a database for all the tables in the model or for only selected tables throughout the model.

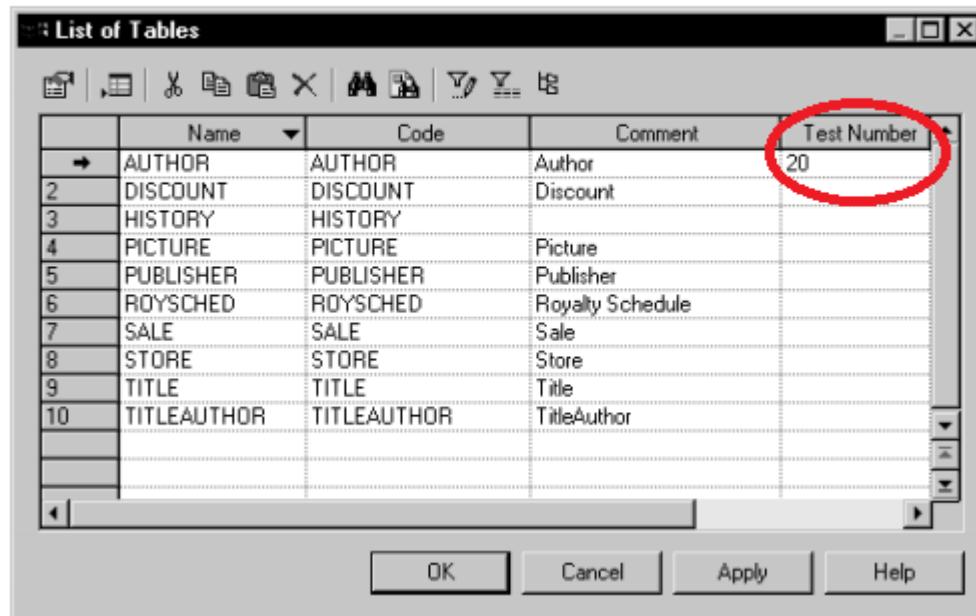
Basis of calculation	<p>You obtain an estimate based on the following elements:</p> <ul style="list-style-type: none">◆ Estimated number of records in tables◆ Columns in each table◆ Indexes in the model◆ Tablespaces in the model◆ Database storage options
----------------------	---

4.6.4. Ước tính kích thước CSDL (2)

Columns	<p>The estimated database size for a column is based on the following:</p> <ul style="list-style-type: none">◆ Size of fixed length data types◆ Average size of variable length data types
Indexes	<p>The estimate of the database size includes all indexes including primary key indexes, foreign key indexes, alternate key indexes and database-specific indexes such as IQ join indexes.</p>
Tablespace size	<p>The size of a tablespace associated with a table is estimated by default, and is displayed as a total of the following:</p> <ul style="list-style-type: none">◆ All tables in the tablespace◆ All indexes in the tablespace
Storage options	<p>Data storage options are DBMS specific. You define data storage options supported by the target database. These are included in the estimate of database size.</p>

4.6.4. Ước tính kích thước CSDL (3)

Indicating the number of records in a table



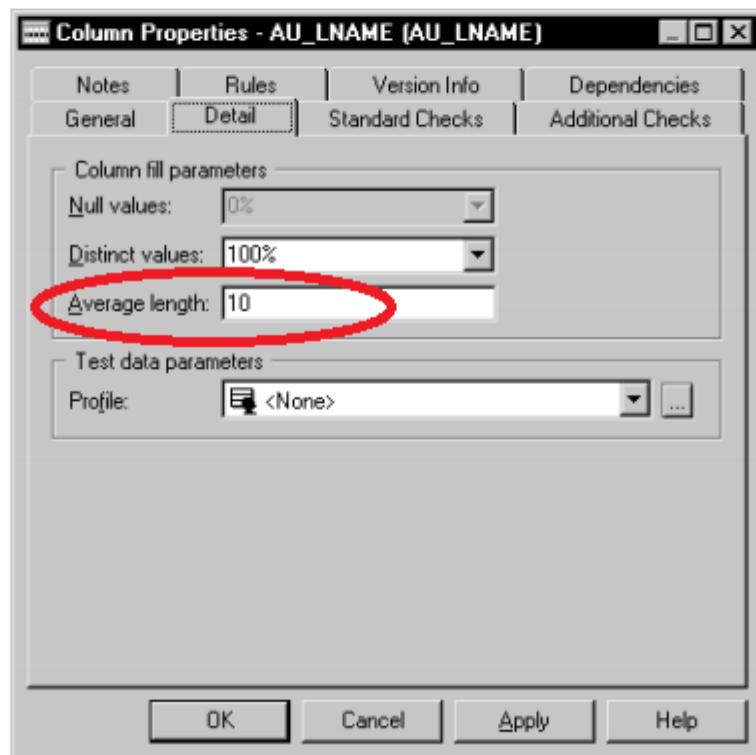
The screenshot shows a Windows-style dialog box titled "List of Tables". The main area is a table with columns labeled "Name", "Code", "Comment", and "Test Number". The "Test Number" column contains the value "20" for the first row, which is highlighted with a red circle. The other rows show standard table names like AUTHOR, DISCOUNT, HISTORY, etc., with their respective codes and comments. At the bottom of the dialog are buttons for "OK", "Cancel", "Apply", and "Help".

	Name	Code	Comment	Test Number
1	AUTHOR	AUTHOR	Author	20
2	DISCOUNT	DISCOUNT	Discount	
3	HISTORY	HISTORY		
4	PICTURE	PICTURE	Picture	
5	PUBLISHER	PUBLISHER	Publisher	
6	ROYSCHED	ROYSCHED	Royalty Schedule	
7	SALE	SALE	Sale	
8	STORE	STORE	Store	
9	TITLE	TITLE	Title	
10	TITLEAUTHOR	TITLEAUTHOR	TitleAuthor	

4.6.4. Ước tính kích thước CSDL (4)

Indicating average data type length

You can indicate the estimated average data type length for variable length data types. This average length value is then used instead of the maximum data type length when you estimate database size.

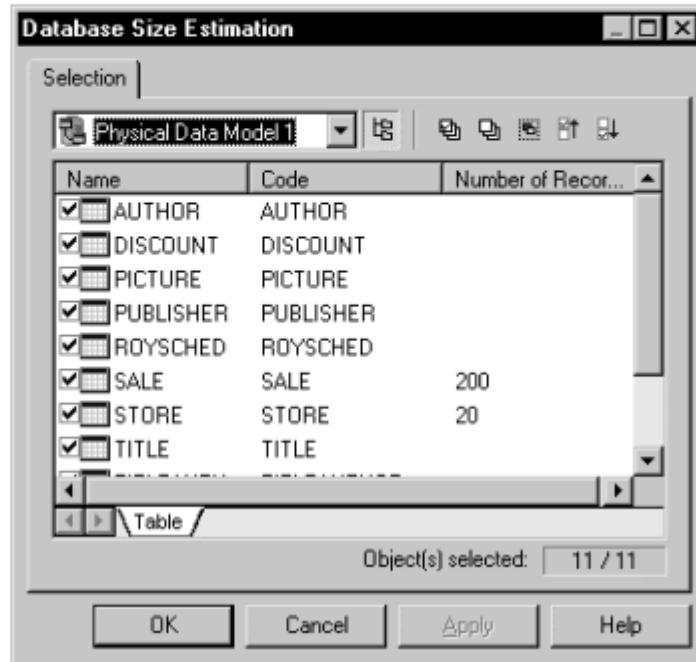


4.6.4. Ước tính kích thước CSDL (5)

Estimating the database size of a model

You can estimate the database size of all the tables in the model.

Select Database→Estimate Database Size.



4.6.5. Tùy chỉnh các đoạn mã script

Customizing scripts

You can customize scripts as follows:

- ◆ Insert scripts at the beginning and end of database creation script
- ◆ Insert scripts before and after a table creation command

Examples

If a development project archives all the database creation scripts that are generated, a header script can be inserted before each creation script, which may indicate the date, time, and any other information specific to the generated script.

If an organization requires that generated scripts are filed using a naming system which may be independent from a script name, a header script could direct a generated script to be filed under a different name than the name indicated in the creation script.

4.6.6. Tạo sinh ra CSDL (1)

Generating a database

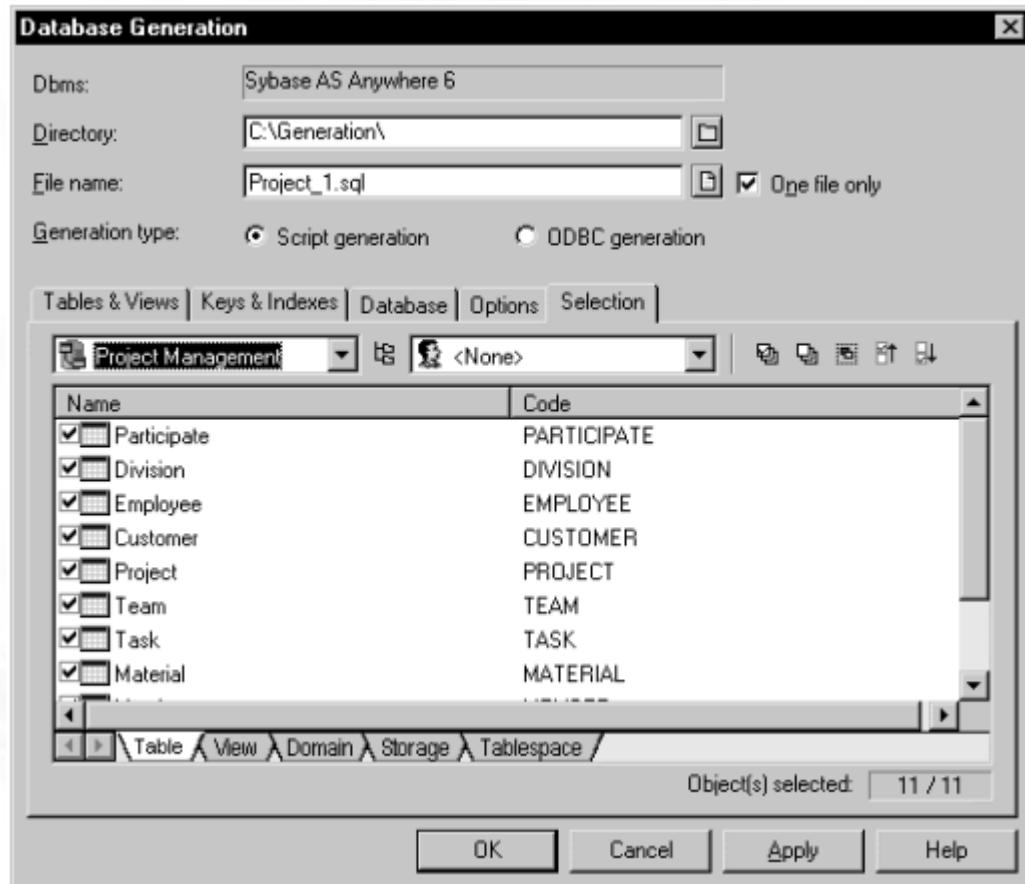
You can create or modify a database in two ways:

- ◆ Directly execute a script on a connected data source using an ODBC driver
- ◆ Generate a script to be executed on a DBMS at a later time

In both cases, the database generation commands are saved in a script file. You must always provide the following information about the script file:

Parameter	Description
Directory	Destination directory for the script file
File name	Destination filename for the script file
One file only	When selected creates one script file, instead of a separate script file for each table
Generation type	Database generation by script or using ODBC

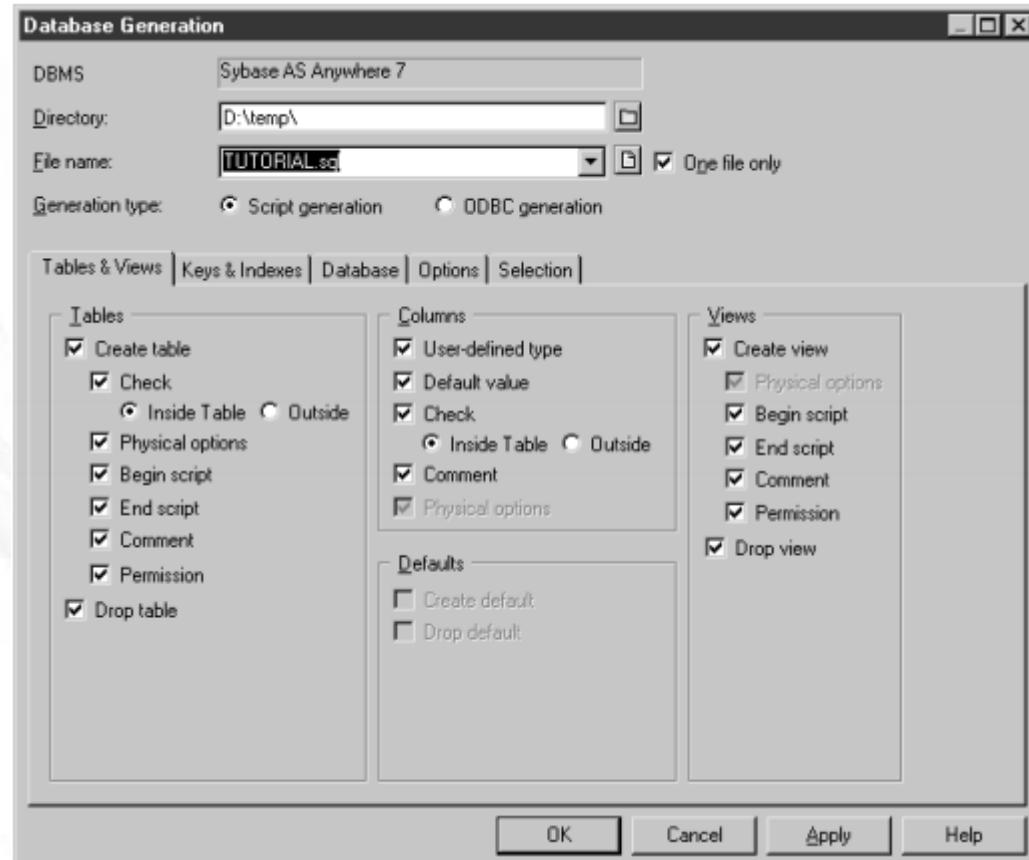
4.6.6. Tạo sinh ra CSDL (2)



4.6.6. Tạo sinh ra CSDL (3)

Generating a database creation script

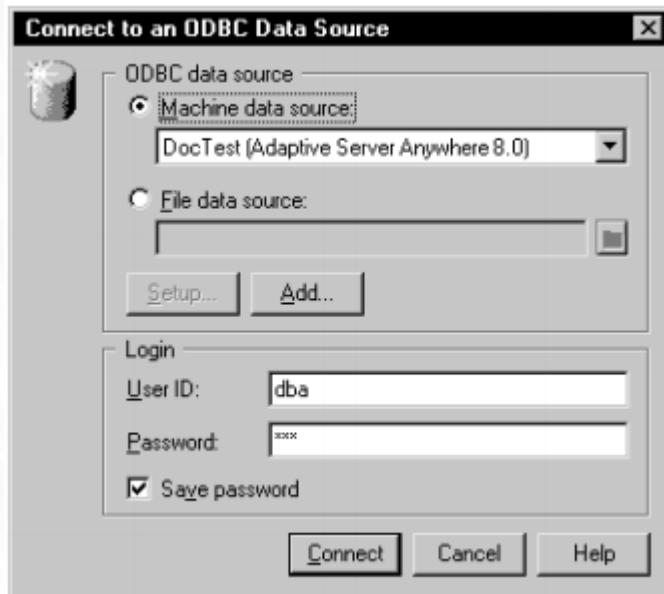
PowerDesigner can generate a database creation script that you can run in your DBMS environment.



4.6.6. Tạo sinh ra CSDL (4)

Creating a database directly

PowerDesigner can generate database structure directly. To do so, you must connect to a data source using an ODBC driver.



4.6.6. Tạo sinh ra CSDL (5)

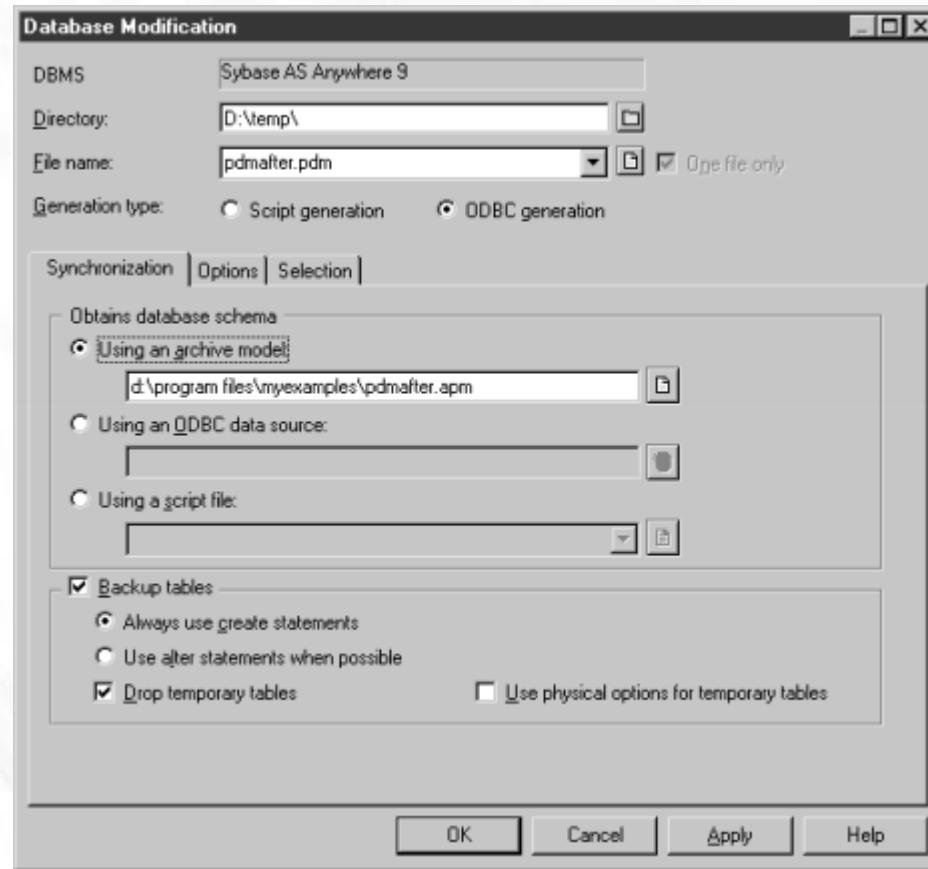
Modifying a database

You can generate a database by synchronizing a modified PDM with one of the following existing database schema:

- ◆ Archived PDM
- ◆ ODBC data source
- ◆ Generation script file

The modified PDM and the existing database schema are merged using a database synchronization window which displays an object tree view of the modified PDM (source model) and the existing database schema (target model). You can choose which objects are added, deleted, or updated in the target database relative to the modified PDM.

4.6.6. Tạo sinh ra CSDL (6)



4.6.7. Sử dụng dữ liệu test (1)

Using 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, PowerDesigner automatically generates rows of data in database tables.

You normally use test data to verify the performance of a database that is being developed. You can use test data to do the following:

- ◆ Verify the **performance of the database** when it is filled with large amounts of data
- ◆ Verify the performance of the database when it is accessed by different applications or users
- ◆ Verify the **operational performance** of the database when it is accessed by different applications or users
- ◆ Estimate the **amount of memory space** the database will take up
- ◆ Examine data formatting in the database

You can generate test data for selected tables in a PDM, or for all the tables in a PDM.

You can add test data to an empty database or to a database that already contains data.

4.6.7. Sử dụng dữ liệu test (2)

Using 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, PowerDesigner automatically generates rows of data in database tables.

A test data profile can use one of the following:

- ◆ Number
- ◆ Character
- ◆ Date/Time

Test data restrictions

The following objects are **not taken into account** when you generate test data:

- ◆ Alternate keys
- ◆ Foreign keys
- ◆ Business and validation rules
- ◆ Binary, sequential, OLE, text or image data types
- ◆ Trigger contents

4.6.7. Sử dụng dữ liệu test (3)

Select Model→Test Data Profiles.

The List of Test Data Profiles appears.

	Name	Code	Comment	Profile Class
1	Profile_1	Profile_1		Number
2	Profile_2	Profile_2		Character
3	Profile_3	Profile_3		Date & Time

General | Detail | Notes | Rules | Version Info | Dependencies

Name: Profile_1

Code: Profile_1

Comment:

Class: Number

Generation source:

Automatic List ODBC File

OK Cancel Apply Help

4.6.7. Sử dụng dữ liệu test (4)

The image displays three overlapping windows titled "Test Data Profile Properties".

- Profile_2 (Left Window):** Contains sections for "Verify" (Valid characters: "[a-z,A-Z,0-9,]", Invalid characters: "", Mask: ""), "Case" (Upper, First uppercase, Lower, Mixed), and "Length" (Exact: 0, From: 0, to: 0). Buttons at the bottom include OK, Cancel, and Apply.
- Profile_3 (Top Right Window):** Contains sections for "Date range" (From: Y [] M [] D [], To: Y [] M [] D []), "Time range" (From: H [] M [] S [], To: H [] M [] S []), and "Step" (Date step: Y [0] M [0] D [0], Time step: H [0] M [0] S [0]). A "Values" section on the right has radio buttons for Random (selected) and Sequential.
- Profile_1 (Bottom Center Window):** Contains a "List of values" table with columns "Value" and "Label". The data is:

	Value	Label
1	1	Engineering
2	2	Accounting
3	3	HR

Buttons at the bottom include OK, Cancel, Apply, and Help.

163

163/1

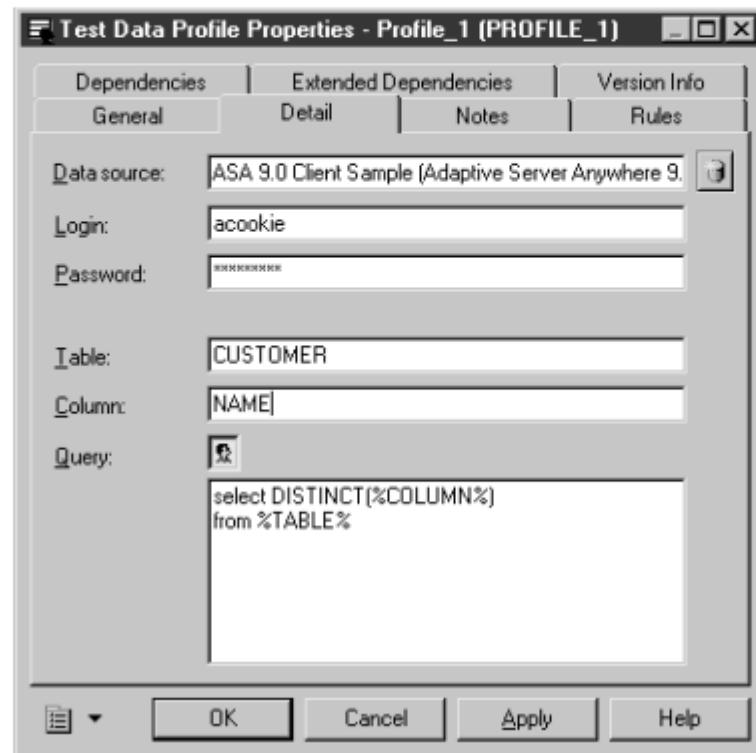
4.6.7. Sử dụng dữ liệu test (5)

Defining an ODBC data source as a test data generation source

You can use information from an ODBC data source to define a test data generation source for a data profile.

Select Model→Test Data Profiles.

The List of Test Data Profiles appears.



4.7. Các mô hình được sinh ra từ MH VẬT LÝ DL

4.7.1. Giới thiệu tổng quan

4.7.2. Sinh tạo PDM từ PDM

4.7.3. Sinh tạo CDM từ PDM

4.7.4. Sinh tạo OOM từ PDM

4.7.5. Sinh tạo XSM từ PDM

4.6.1. Giới thiệu tổng quan (1)

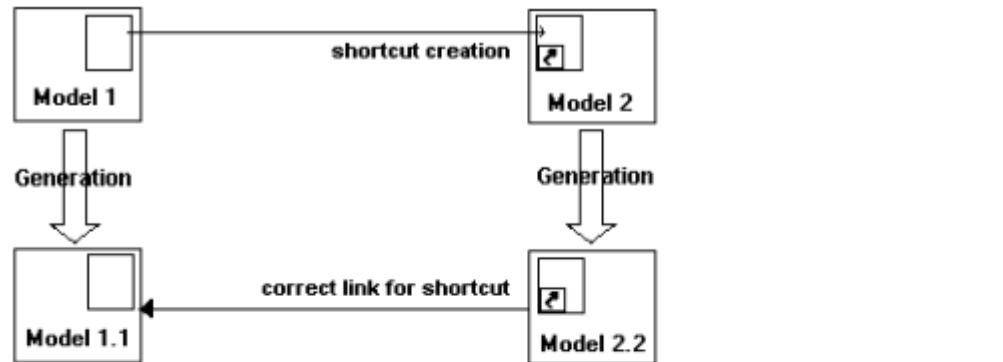
Generation basics

When you generate from a Physical Data Model you can generate to a PDM, a CDM, or an OOM. You can generate a model from a global PDM or from a package within the model. Limiting model generation to a single package is useful when different designers own packages of the same PDM. Designers can generate their packages independently from others. Generating a package results in an independent model.

You can generate from a PDM in two ways:

Generate	Description
New model	Creates a new model containing the objects translated from the PDM
Updated model	Creates a default model with the objects translated from the PDM that is merged with an existing model. You can update, delete, or add objects in the existing model (model to be merged in the right pane) based on modifications made in the default model (in the left pane)

4.6.1. Giới thiệu tổng quan (2)



For example, here is the proper sequence of events for external shortcuts generation:

- ◆ Model 1 is the target model of a shortcut in Model 2
- ◆ Model 1 is generated to Model 1.1
- ◆ Model 2 is prepared for generation to Model 2.2 by associating appropriate parameters in the Target Models page:

4.7.2. Sinh tạo PDM từ PDM (1)

Generating a Physical Data Model from a Physical Data Model

This section explains how to generate a Physical Data Model from a Physical Data Model.

Why generate a PDM into a PDM?

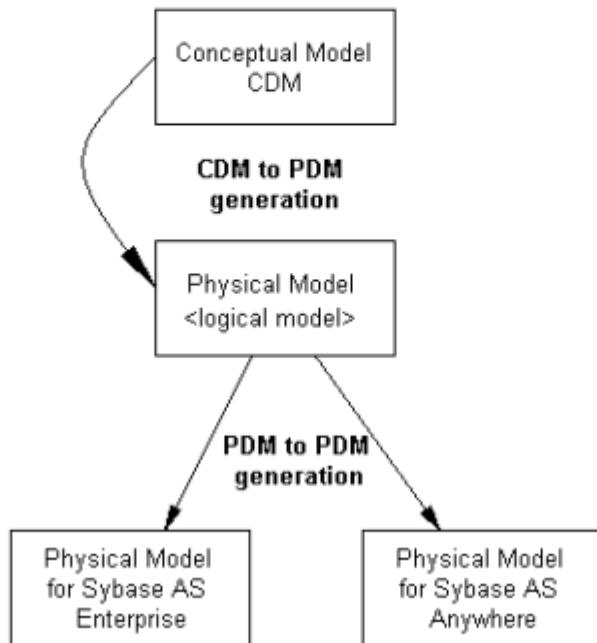
You can generate a PDM into a PDM when you need to keep two models synchronized during the design process.

This kind of generation allows you to create a copy of a given model and define generation links between objects in the source PDM and their equivalent in the generated PDM. When changes are made to the source model, they can then be easily propagated to the generated models using the Update Existing Model generation mode.

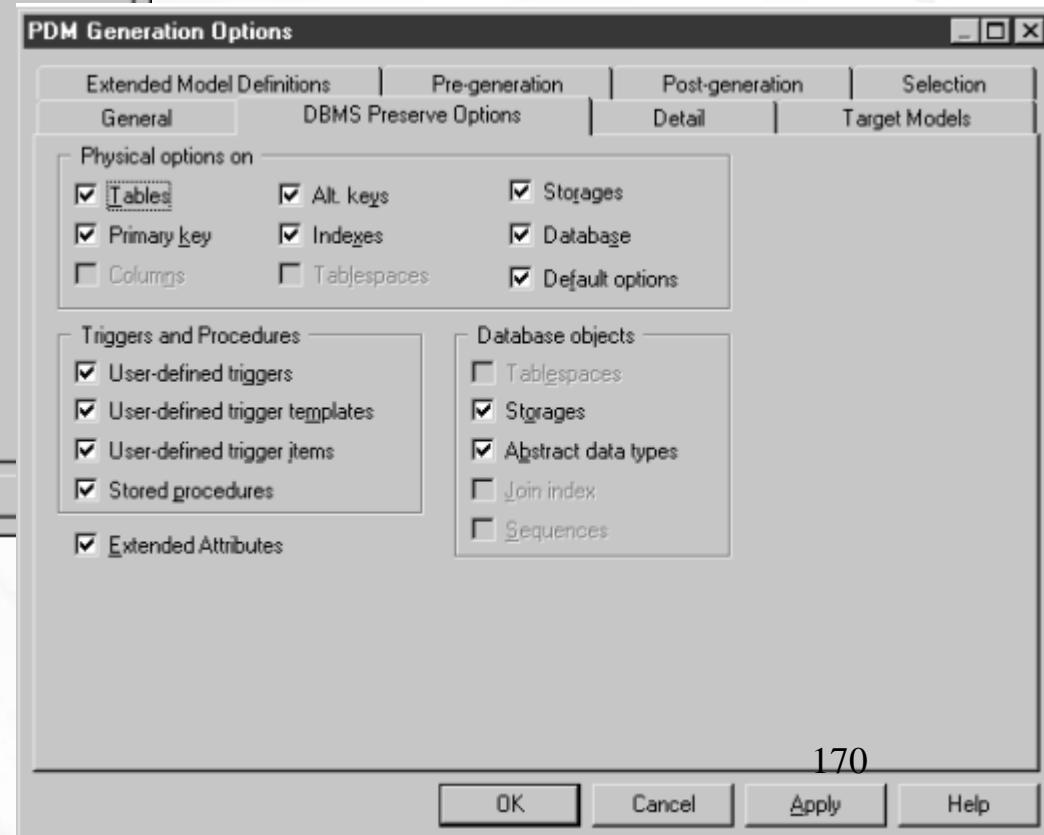
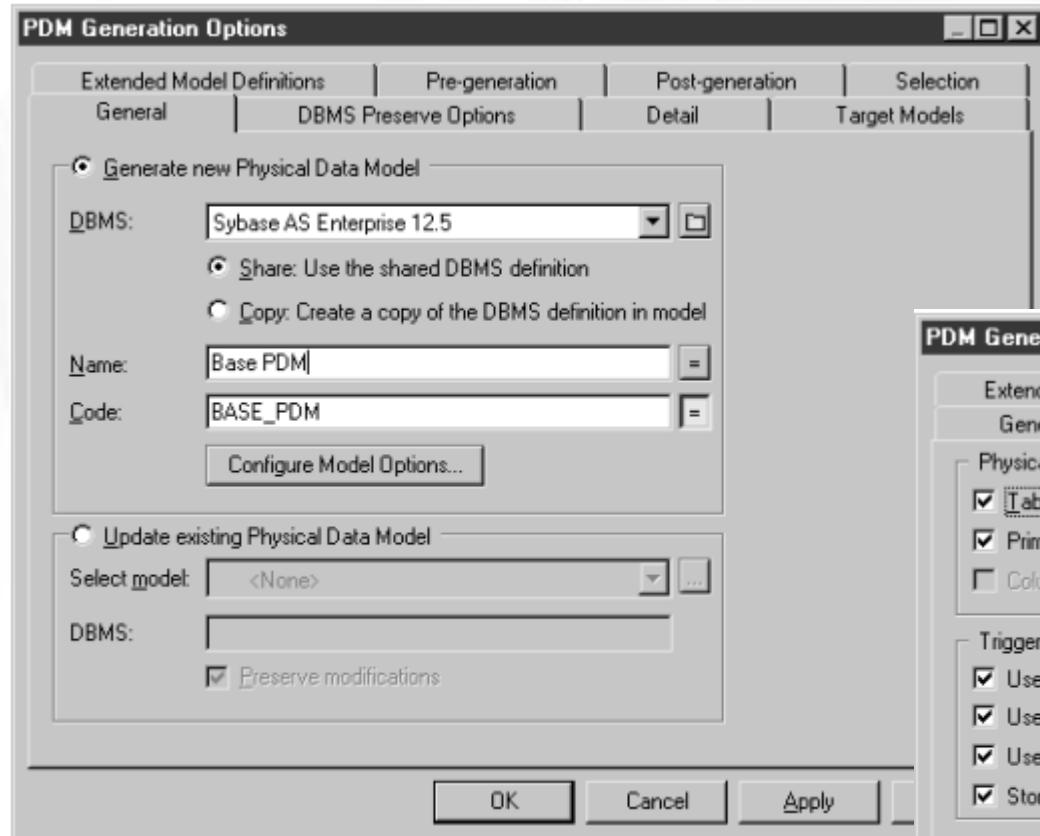
4.7.2. Sinh tạo PDM từ PDM (2)

Multi DBMS environment

If you are working in a multi DBMS environment, you can use the PDM to PDM generation to avoid generating a CDM into different PDM based on different DBMS. You generate a PDM from the CDM, this PDM uses the <logical model> DBMS. Then you can use this PDM to generate different PDM for the different DBMS required as shown in the following figure:



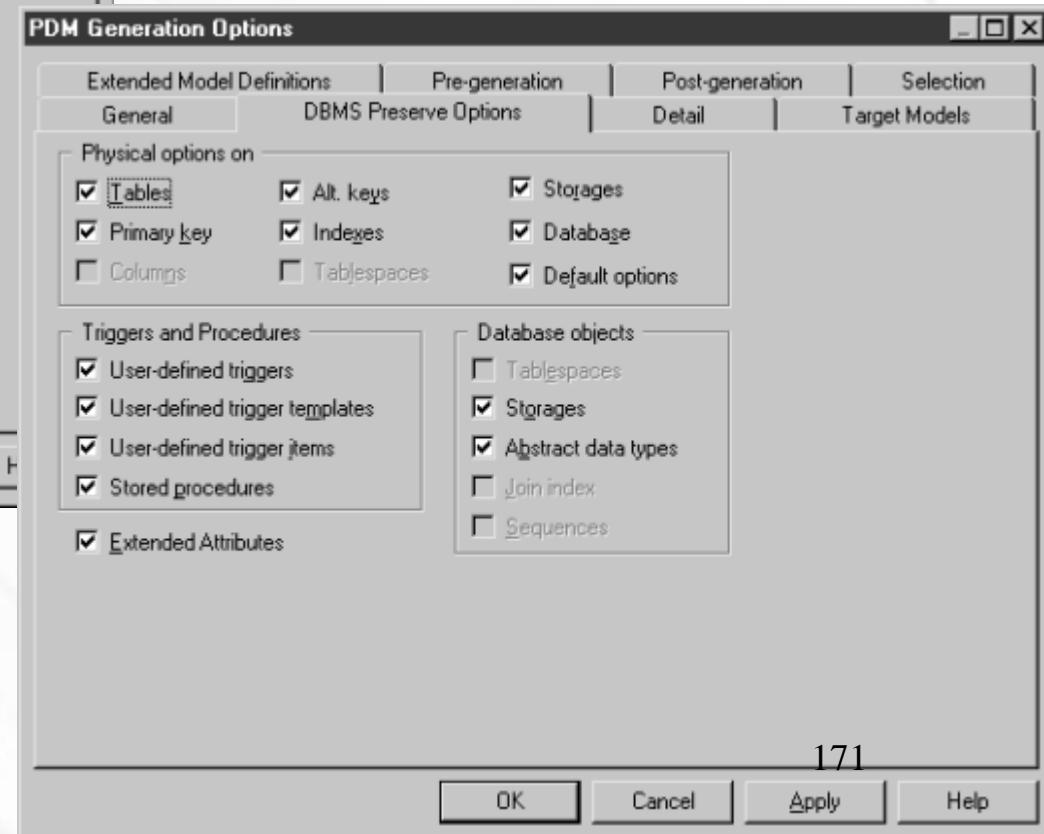
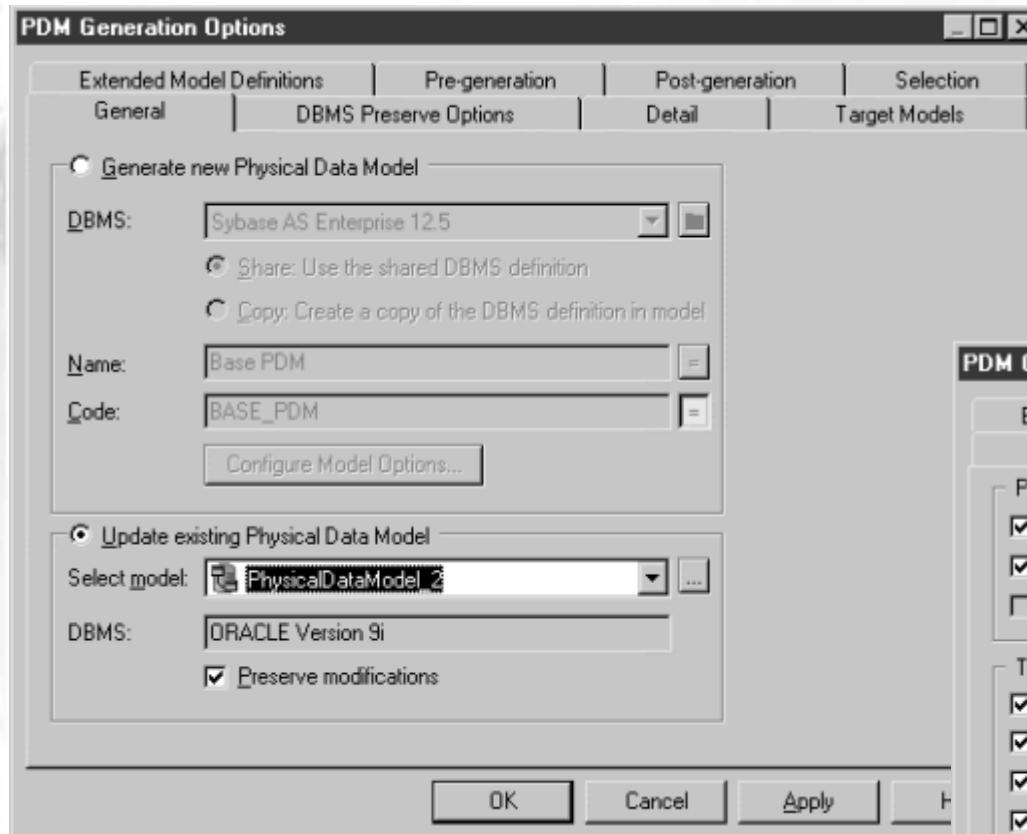
4.7.2. Sinh tạo PDM từ PDM (3)



170

170/1

4.7.2. Sinh tạo PDM từ PDM (4)



4.7.3. Sinh tạo CDM từ PDM (1)

Generating a Conceptual Data Model from a Physical Data Model

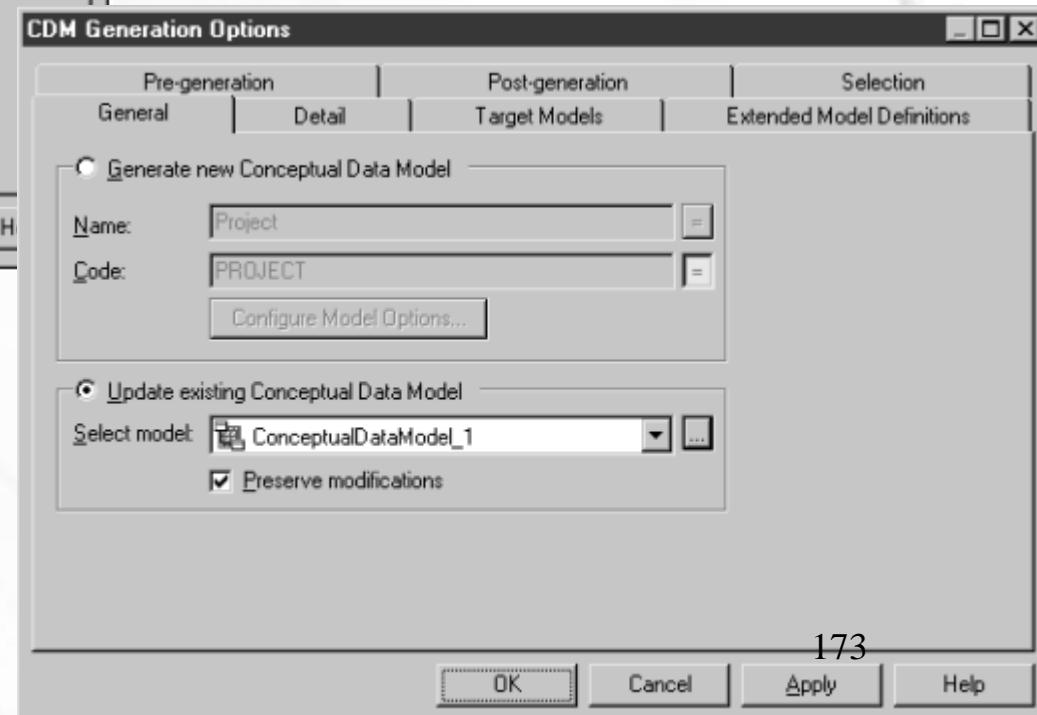
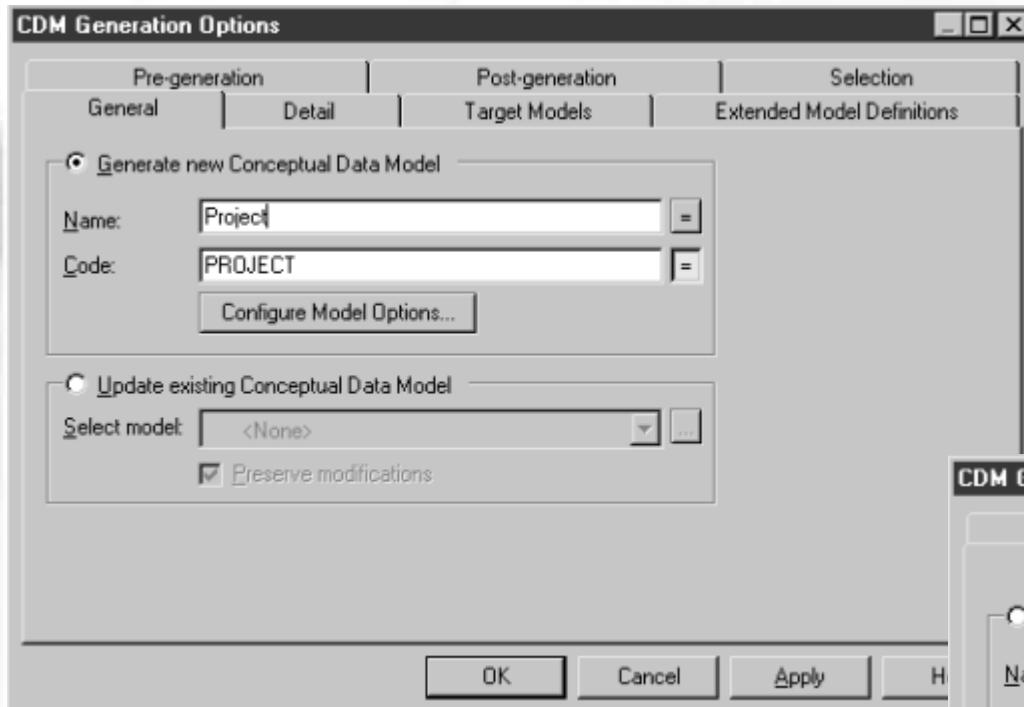
This section explains how to generate a Conceptual Data Model from a Physical Data Model.

Generating CDM objects

When you generate a Conceptual Data Model (CDM) from a PDM, PowerDesigner translates physical objects to conceptual objects as follows:

Object in a PDM	Generated object in a CDM
Table	Entity
Table column	Entity attribute
Primary key	Primary identifier
Alternate key	Identifier
Reference	Relationship

4.7.3. Sinh tạo CDM từ PDM (2)



173

173/1

4.7.4. Sinh tạo OOM từ PDM (1)

Generating an Object-Oriented Model from a Physical Data Model

This chapter describes how you can generate an Object-Oriented Model (OOM) from a Physical Data Model (PDM).

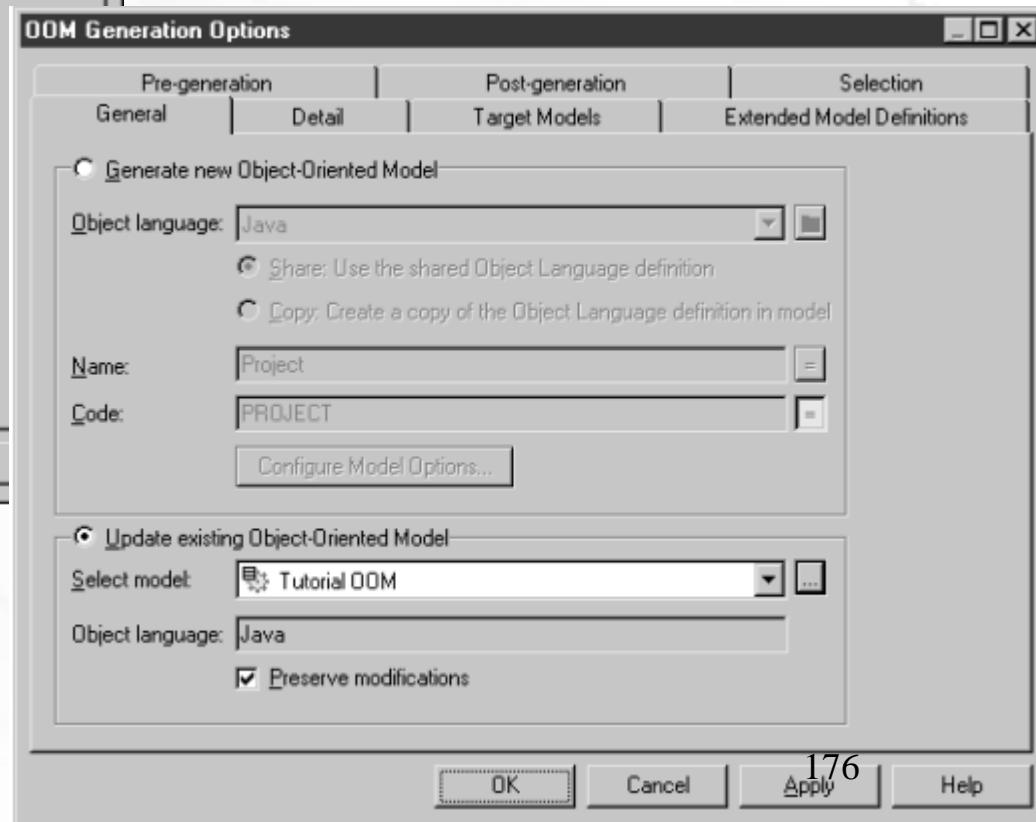
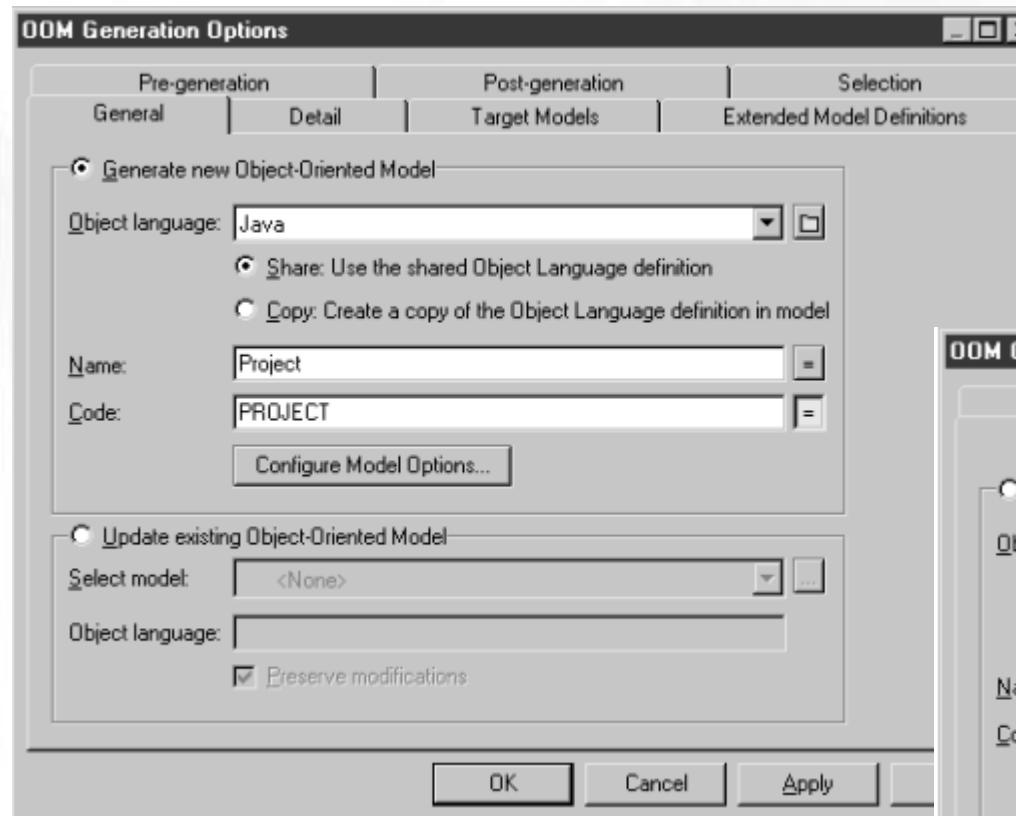
4.7.4. Sinh tạo OOM từ PDM (2)

Generating OOM Objects

When you generate an Object-Oriented Model (OOM) from a PDM, PowerDesigner translates PDM objects into specified object language objects as follows:

PDM Object	Generated object in an OOM
Domain	Domain
Table	Class. The cardinality of a class is translated from the number of estimated records in a table
Table with migrated keys from only two other tables	Class linked with an association class between the two classes generated by the two parent tables
Column	Attribute
Key	Identifier
Primary key	Primary identifier
Stored-Procedures and stored functions attached to selected table	If the parent table is generated as a class, the stored procedure or the stored function is generated as an operation attached to the class
Reference	Association

4.7.4. Sinh tạo OOM từ PDM (3)



4.7.5. Sinh tạo XSM từ PDM (1)

Generating an XML Model from a Physical Data Model

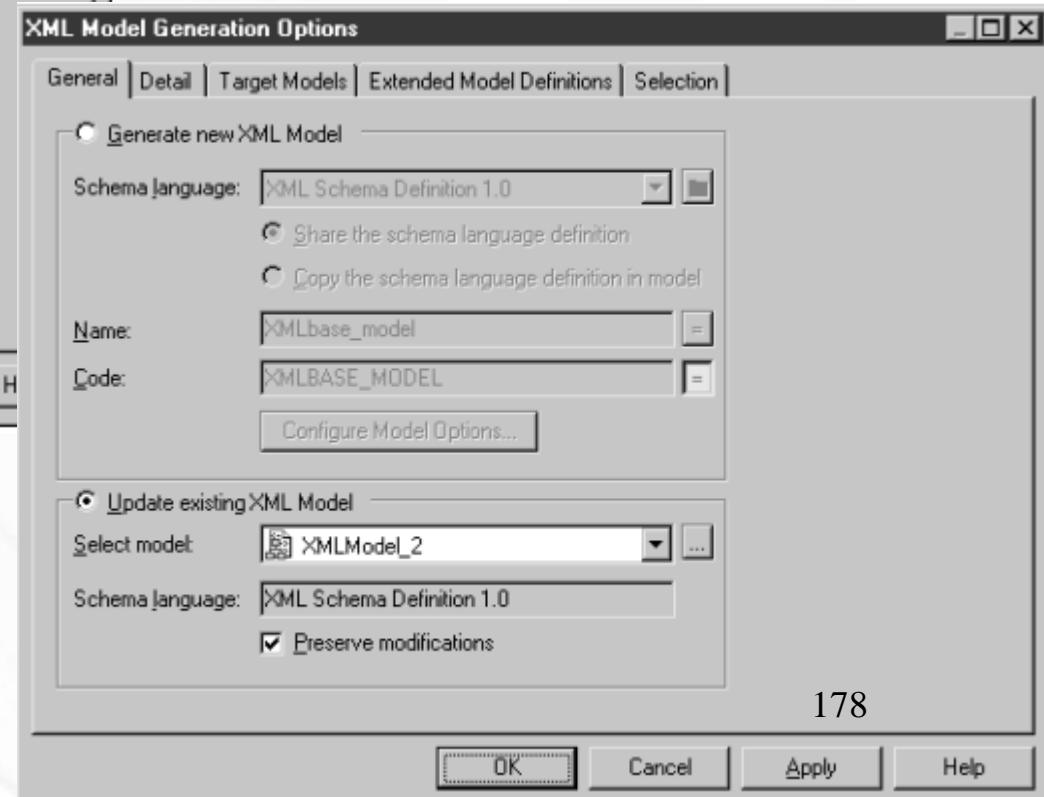
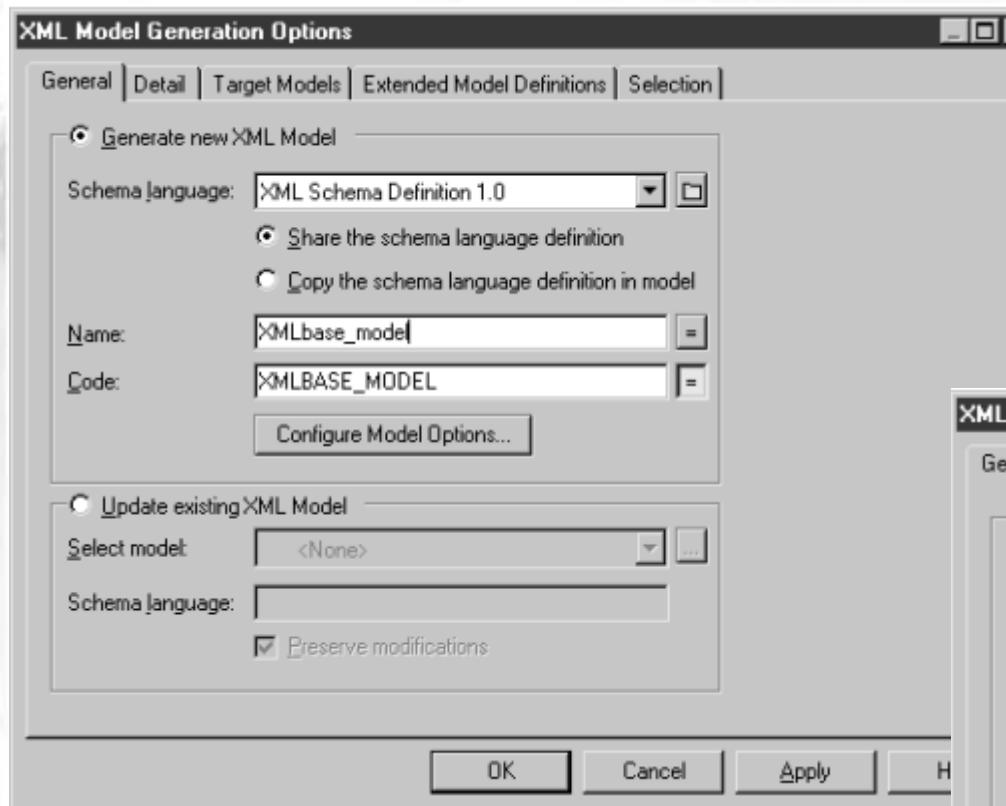
This section explains how to generate an XML Model from a Physical Data Model.

Generating XML Model objects

When you generate an XML model from a PDM, PowerDesigner translates PDM objects into specified XML model objects as follows:

PDM Object	Generated object in an XML model
Domain	Simple Type
Table	Element
Column	Attribute or element (see generation options)
View	Element
View column	Attribute
Key	Key
Index	Unique
Abstract data type	Complex Type

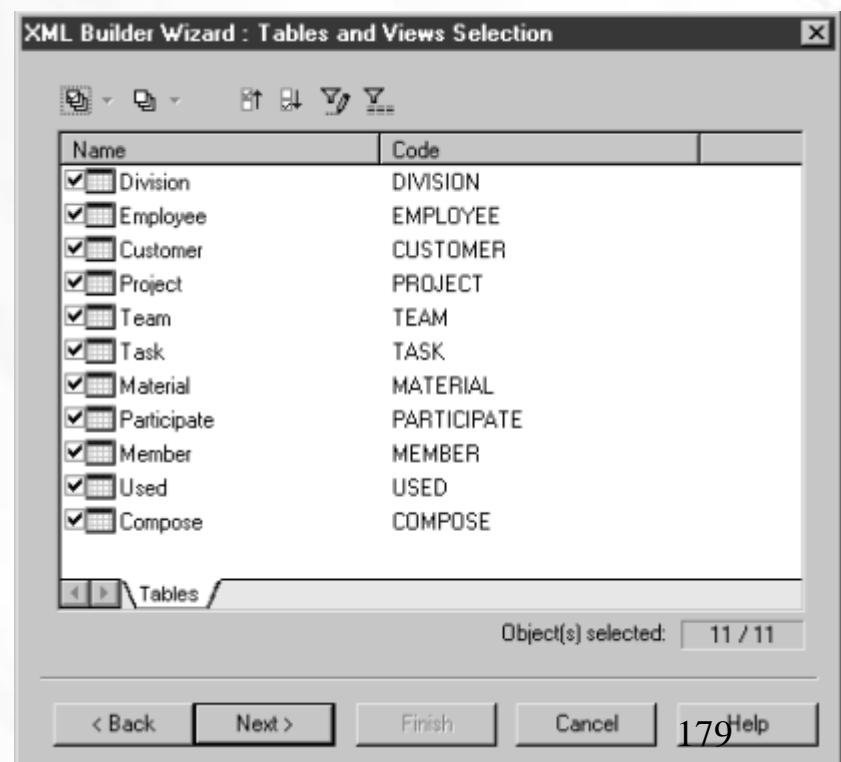
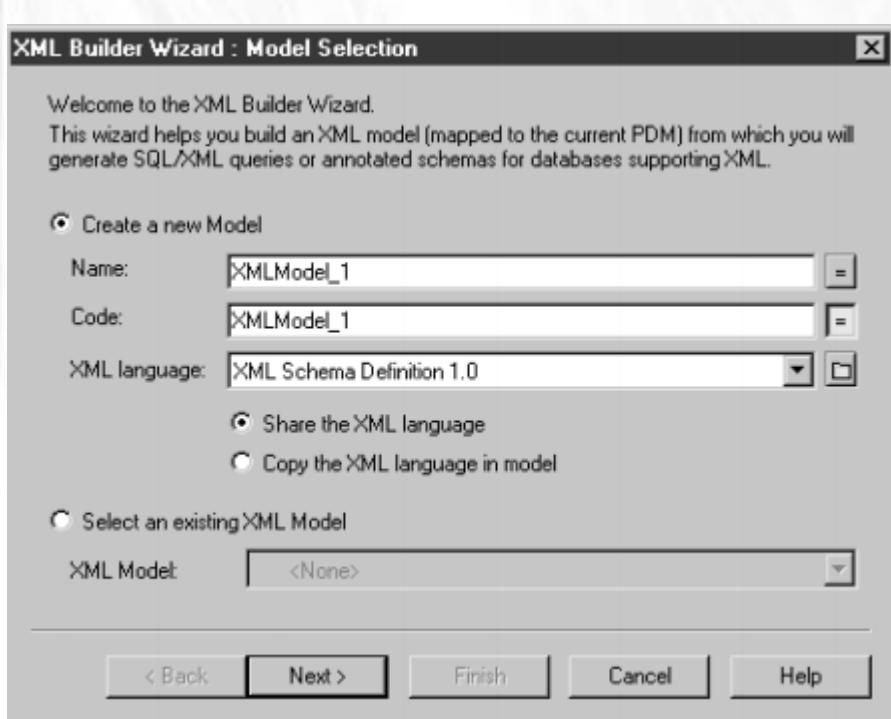
4.7.5. Sinh tạo XSM từ PDM (2)



4.7.5. Sinh tạo XSM từ PDM (3)

Generating an XML model via the XML Builder Wizard

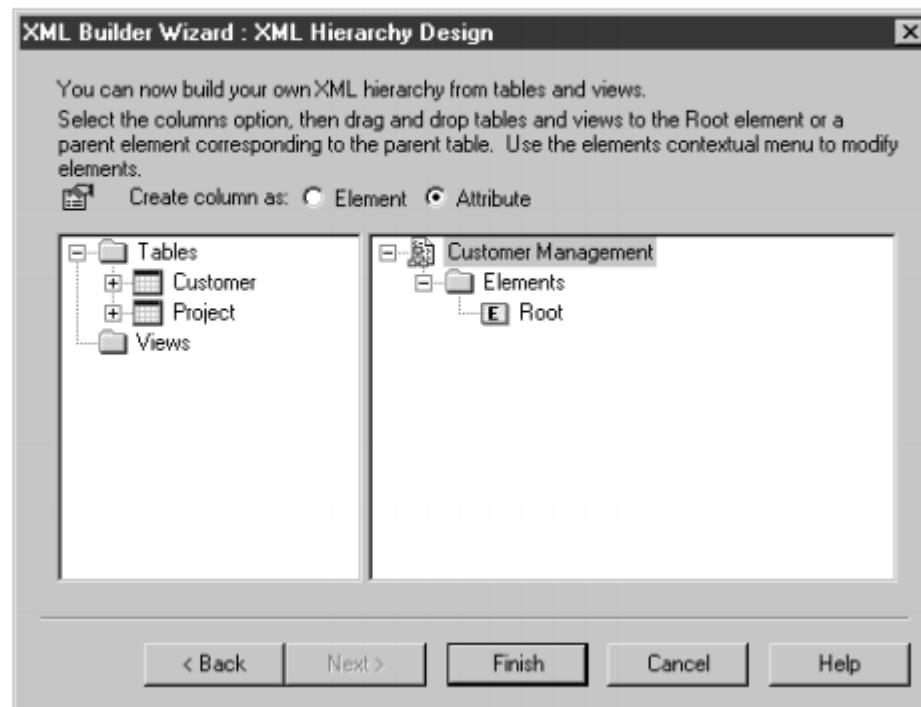
The XML Builder Wizard helps you build an XML model that will be used to generate SQL/XML queries for retrieving data from databases. Unlike the standard PDM to XML model generation, the XML Builder Wizard facilitates the next operations:



4.7.5. Sinh tạo XSM từ PDM (4)

Building an XML hierarchy

In the XML Hierarchy Design page, you build the structure of the XML model.



4.7.5. Sinh tạo XSM từ PDM (5)

XML Builder Wizard : XML Hierarchy Design

You can now build your own XML hierarchy from tables and views.

Select the columns option, then drag and drop tables and views to the Root element or a parent element corresponding to the parent table. Use the elements contextual menu to modify elements.

Create column as: Element Attribute

Tables

- Customer
 - Columns
 - Customer number
 - Customer name
 - Customer address
 - Customer activity
 - Customer telephone
 - Customer fax
- Project
- Views

Customer Management

- Elements
- Root
- sequence
- Customer
- sequence
- Customer number
- Customer name
- Customer address
- Customer activity
- Customer telephone
- Customer fax

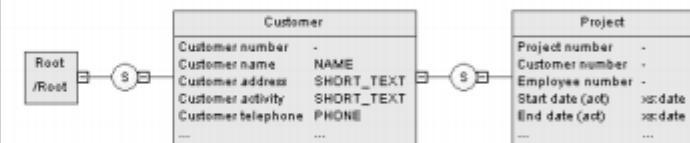
Workspace

- Project Management [PDM] ×
- Customer Management *
- Diagram_1
- Simple Types
- Elements
- Root
- sequence
- Customer
- sequence
- Project
- Attributes
- Customer number
- Customer name
- Employee number
- Start date (act)
- End date (act)
- Project name
- Project label
- Attributes
- Customer number
- Customer name
- Customer address
- Customer activity
- Customer telephone
- Customer fax

Data Sources

- Project Management [PDM]
- Extended Model Definitions
- SQL/XML

< Back Next > Finish Cancel



4.8. Tổng kết chương & Bài tập

Hãy vẽ mô hình PDM cho mô tả hệ thống.
Sau đó, chuyển đổi sang các mô hình
CDM, OOM và XSM.
(sẽ được cung cấp tại lớp học)

Môn học:

Công cụ Thiết Kế Hệ Thống Thông Tin

Giảng viên: ThS. VÕ THỊ KIM-ANH

(2021)