Document Name		Confidentiality Level
Database Design and Modeling		Only for Recipients' Reference
Template Code	Version	Document Code
BP	1.0	BP

Database Design and Modeling (Student Information Management System)

Database design, Data dictionary, E-R and PD Modeing

Copyright Declaration

Contents

1 Teaching Tips

Practice targets

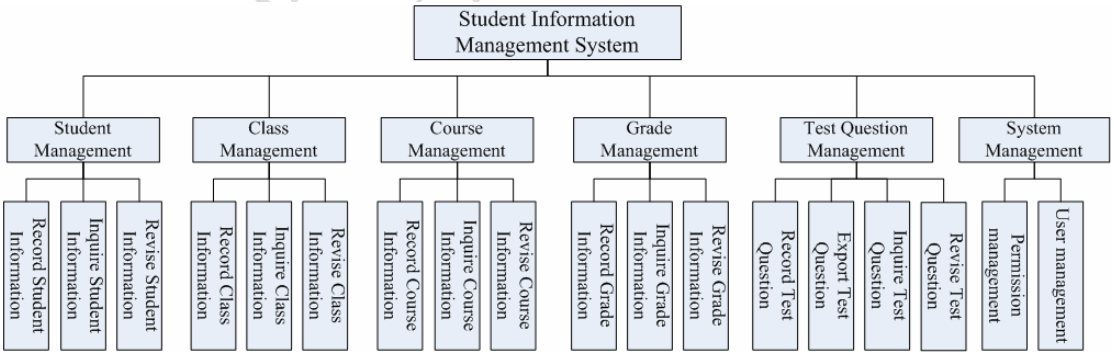
(1) Getting to know the concept and purpose of database design;
(2) Getting to know the procedure and tool of database design;
(3) Getting to know Data Dictionary, as well as its analysis and description;
(4) Getting to know the relationship of entities, grasping E-R Diagram structure and the application;
(5) Grasping the use of database modeling tool PowerDesigner;
(6) And conducting database design of Student Information Management System by applying skills learned under this subject.

2 Functional Requirements

According to the function of Student Information Management System, we abstract Data Dictionary model by analyzing data, business procedure and processing method that relate to the system.
And based on Data Dictionary, we identify the relationship between entities by drawing E-R Diagram, and carry out both database design and modeling to generate SQL script supported by DBMS SQL Server 2008, so as to enable construction and configuration of system database.

1. Function

The Student Information Management System realizes a unified management over student information through automatic processing by the introduction of computer and network technologies. The functional architecture is like this.



Major functions

(1) **Student Information Management.** To record, inquire and revise student basic information, class, course, grade, etc.
(2) **Test Question Management.** To establish test bank for each course, so as to enable student to check learning by taking tests. Grades should be provided after test immediately by the system. Functions such as recording, inquiry, exporting and revision should be provided as well.
(3) **System Management**. Roles of system user are defined as administrator, teacher, student, etc.

Access authorization should be identified by the system to all roles, and basic user profiles are to be maintained as well.

2. Data Dictionary

System data are Student, Class, Teacher, Course, Faculty, Selective course, Grade, Teaching schedule, Test question, etc.

Data Dictionary of student is described in the following table.

student					
Data element	Internal name	Domain	Description	Null	Type/Length
Student id	id		Mark each unique student	N	int
Name	name			N	string/50
Gender	gender	1-2	1 male，2 female	N	int
Nationality	nationality			N	string/20
Address	address				string/100
Date of birth	birthday			N	string/50
Class number	class_id			N	int
Class name	class_name			N	string/50

For more detail, please see file **Data Dictionary (Student Information Management System).xls.**

3 Development Environment

Type	Tool and environment
Developing tool	Database modeling tool: PowerDesigner.

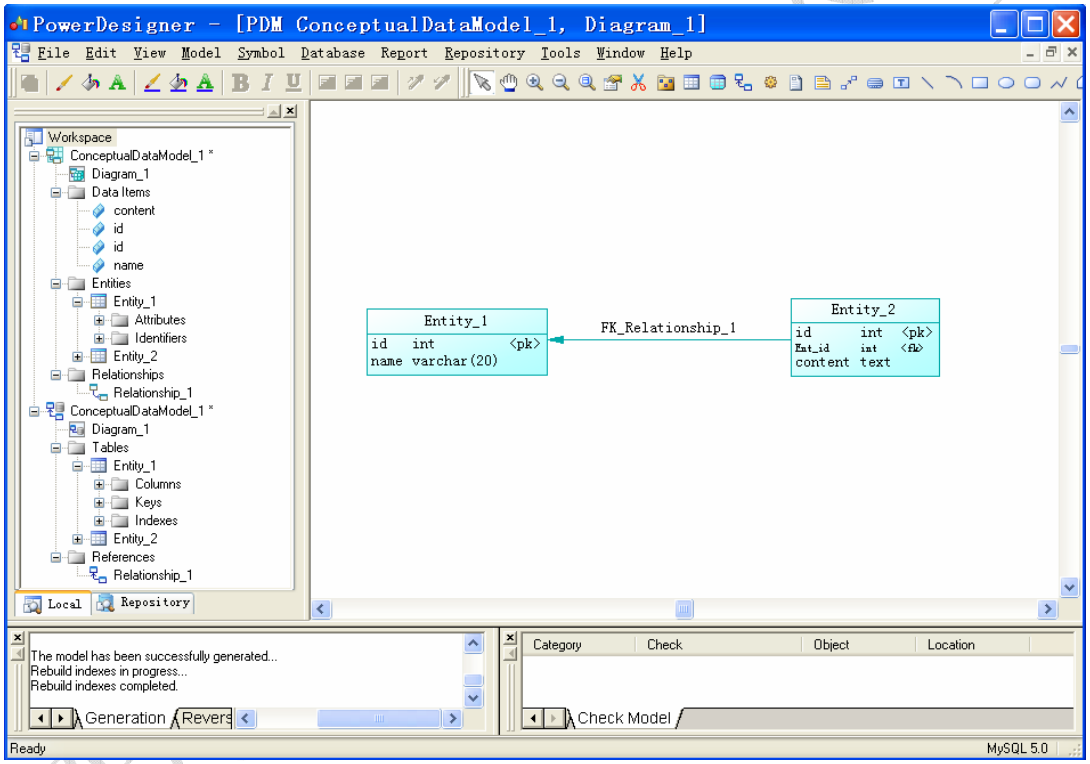
1. PowerDesigner introduction

PowerDesigner is a CASE tool set released by Sybase. By PowerDesigner, you can make data procedure chart, conceptual data model and physical data model, you can generate applications of multiple client development tools, and even you can produce structural model for data warehouse, as well as control team equipment model.

PowerDesigner is a commonly used database modeling tool among developers, which helps in conducting database design of high efficiency. The downloading address is http://www.sybase.com/products/modelingdevelopment/powerdesigner/

2. PowerDesigner workspace

When PowerDesigner is used in modeling conceptual data model, the workspace is like this.



3. In this application

In this application, we use PowerDesign to model Conceptual Data Model and Physical Data Model, and to generate database SQL script.

4 Design Idea

By combining database design procedure and skills, we carry out database design and modeling. Data model is relational, and a two-dimensional table is structured. And basing on that, we compile SQL script on the basis of SQL Server 2008.

The overall design idea and procedure is like this.

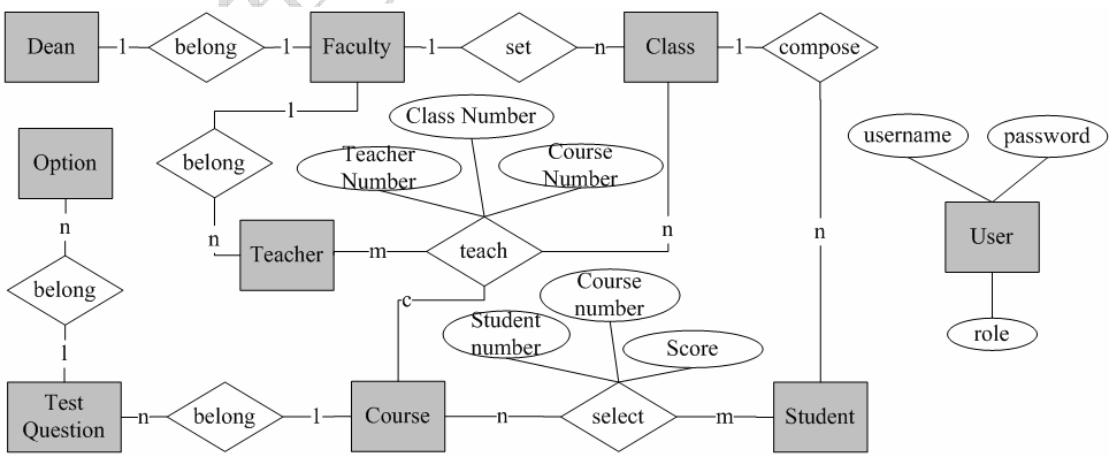
Requirement analysis	To collect and analyze requirements, in order to identify data items to abstract **data dictionary**.
Conceptual structure design	To identify entity, properties of entity and relationship between entity, and then build E-R model.
Logic Structure design	Based on E-R model, according to the practical DBMS products to build corresponding logical structure. Build the relational model, form a two-dimensional table.
Physical structure design	Design data storage structure and accessing method, build SQL script.
Database implementation	Select the storage location of database, use SQL script to build data and table.

Firstly, you need to identify the relationship between entities by drawing E-R Diagram and optimize it in the process according to Data Dictionary. Secondly, you need to establish conceptual model and physical model by using PowerDesigner, and generate SQL script.

1. Draw E-R Diagram

Obtain system entities by analyzing data in Data Dictionary, naming Student, Class, Teacher, Course, Faculty, Dean, Test question, Option and User.

The E-R Diagram is shown below. In this diagram, some but key entity attributes are identified.



2. Two-dimensional table structure

A relational database model is created on SQL Server 2008 by E-R Diagram. It's a two-dimensional table.

Descriptions of student table are listed here below.

t_student				
Name	Code	Type/Length	Instruction	Null
id	id	int	Primary key, auto-increase by 1 each time	N
Student id	student_id	int	Mark each unique student	N
Name	name	varchar/20		N
Gender	gender	int	1 MALE，2 FEMALE	N
Nationality	nationality	varchar/20		N
Political status	politics_status	int	1 Folks，2 League member，3 Party member	N
Address	address	varchar/100		
Date of birth	birthday	varchar/50		N
Class number	cla_id	int	Foreign key, quote id in table t_class.	N
Delete Mark	del	int	0 Deleted Not，1 Deleted	N

For a complete table structure, please see file **Database Table Structure (Student Information Management System).**

5 Involving knowledge and technology

5.1 Database Design Procedure

We'll get the knowledge of database design procedure and contents in two parts. One is "The Designed Lifetime of Database", and the other is "Contents in Database Design Stage".

1. The Designed Lifetime of Database

Database design is divided into six stages according to the complete developing process of database and the system.

(1) Requirement analysis

To collect and analyze requirements, in order to get data requirements to be described in Data Dictionary.

(2) Conceptual structure design

To integrate, induce and abstract requirements, in order to formulate a conceptual model free from any specific DBMS.

(3) Logic structure design

To convert conceptual structure into data model supported by DMBS, like a relational model. And then optimize.

(4) Physical structure design

To select a best-suitable physical structure for logic data model, including both storage structure and accessing method.

(5) Database implementation

To establish database based on the result of logic and physical design by deploying data language like SQL and the host language like C, which are provided in DBMS. And to compile and debug program, import data into database and conduct a trial running.

(6) Database running and maintenance

Database can be put in formal operation after trial running. A continuous assessment, adjustment and revision are recommended in the process of operation.

Please note that an excellent database isn't completed just in one time. It's usually a continuous repeating of the above six stages.

2. Contents in Database Design Stage

The design procedure includes not only database design, but also the design of database application system. Now we give a detailed description of all design stages.

5.2 The Use of PowerDesigner

1. About PowerDesigner

(1) Introduction

PowerDesigner is a CASE tool set released by Sybase. By PowerDesigner, you can make data procedure chart, conceptual data model and physical data model, you can generate applications of multiple client development tools, and even you can produce structural model for data warehouse, as well as control team equipment model.

PowerDesigner is a standardized tool and is very popular in database design.

(2) Advantages

To write SQL script by hand is not easy and is inclined to have error. The script does not show the relationships of entities visually. It's complex to modify script if there is any change in model.

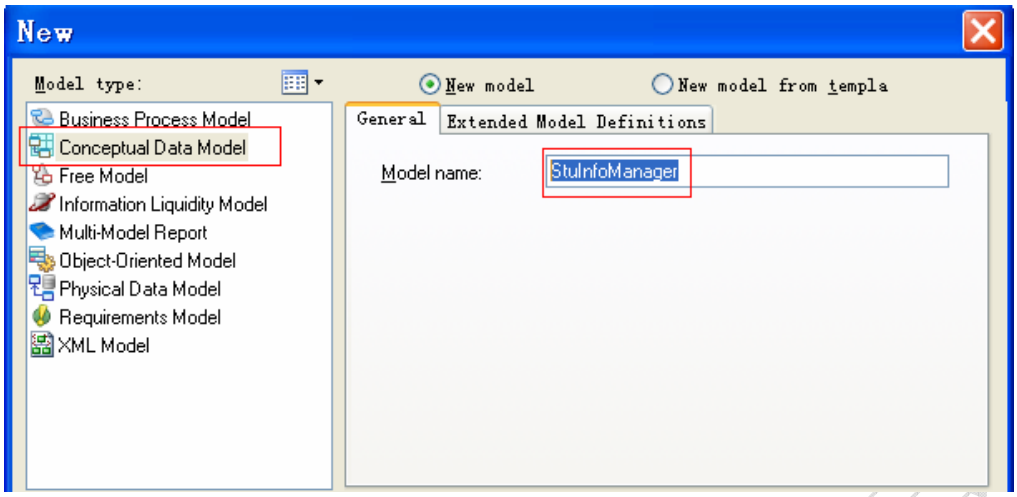
While modeling by PowerDesigner enables us to view the relationship clearly. And conversion can be made freely among CDM, PDM and SQL. When data information changes or we need to move to a new database, for example when SQL Server is replaced by MySQL, what you do is merely to modify the model to generate a new script. It's much easier to maintain and manage.

2. PowerDesigner modeling

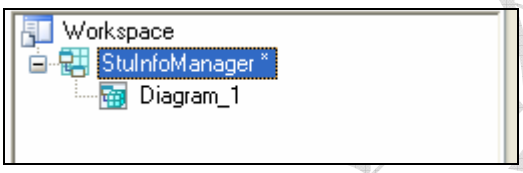
The key operating steps of modeling by PowerDesigner and script exporting are listed below.

(1) Create conceptual model

Select File->New from menu bar for a New dialog. Choose Conceptual Data Model in the dialog and then enter name StuInfoManager.

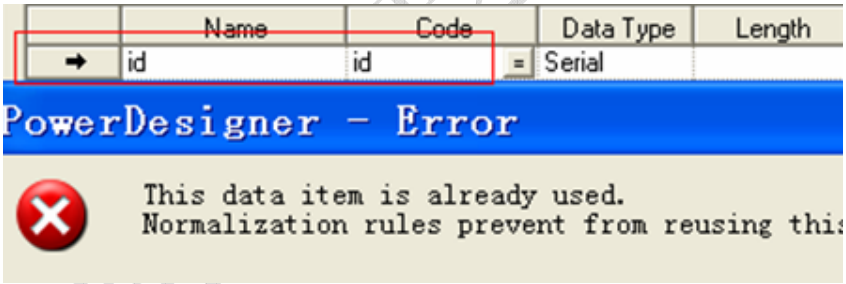


You can see the newly created conceptual model in Workspace.

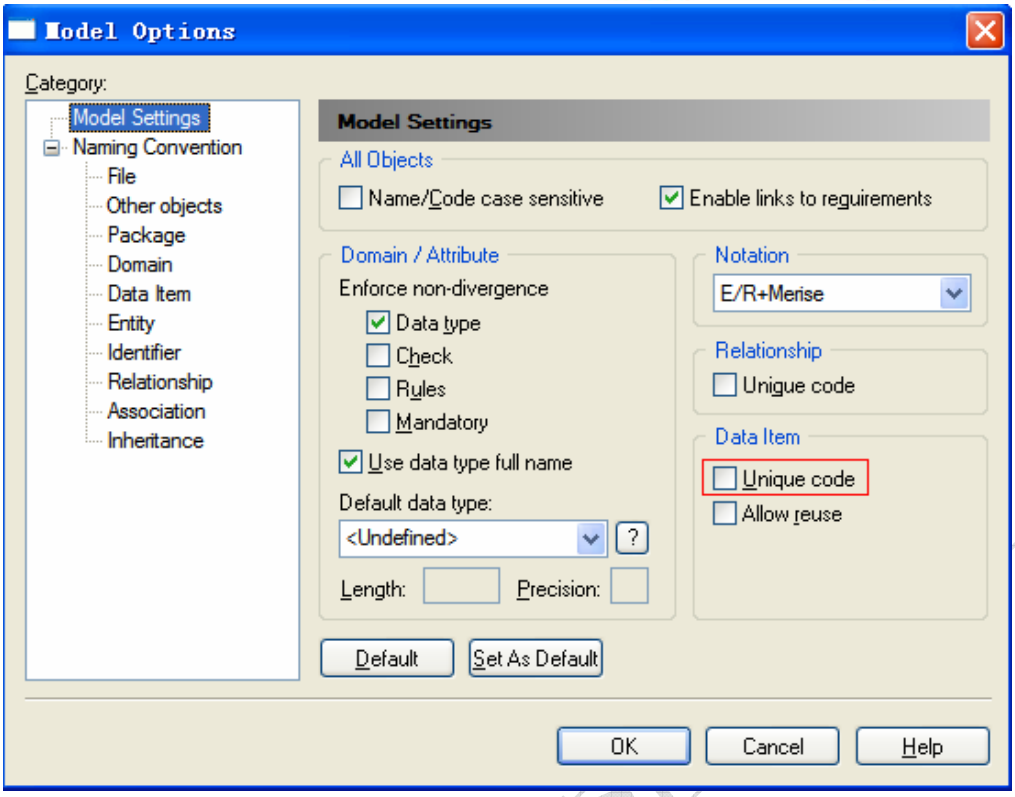


(2) To avoid reusing names

When setting attributes for the entity, sometimes you need same attributes in multiple entities, like name, id. But the attribute code is not allowed to be reused by default in PowerDesigner . The errant message is like this.

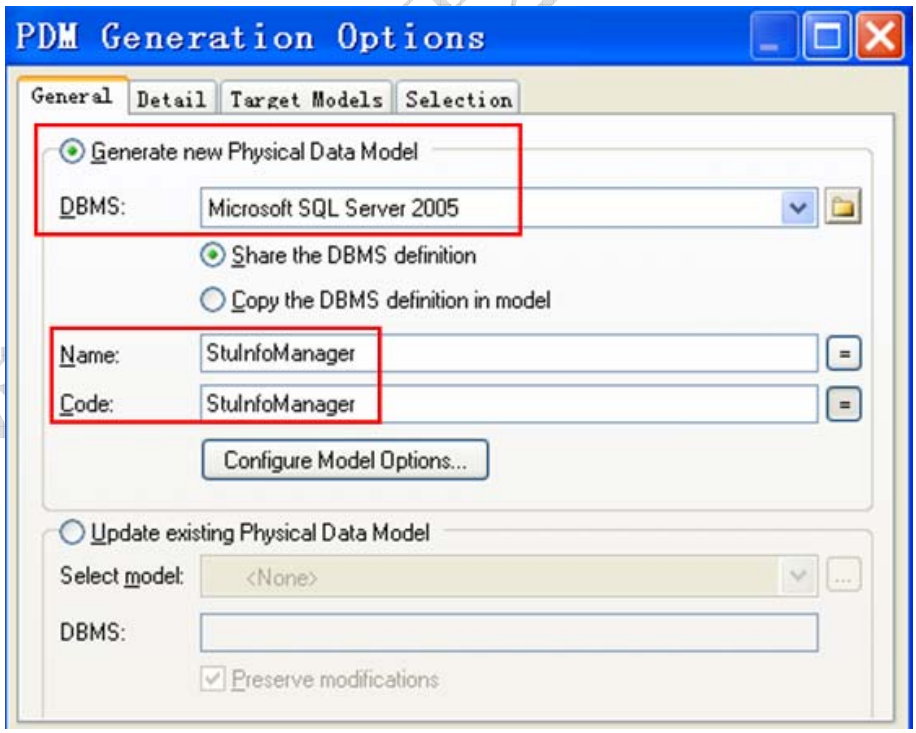


Select “Tools -> Model Options” from menu bar to uncheck “Unique code”.



(3) Convert Conceptual Model into Physical Model

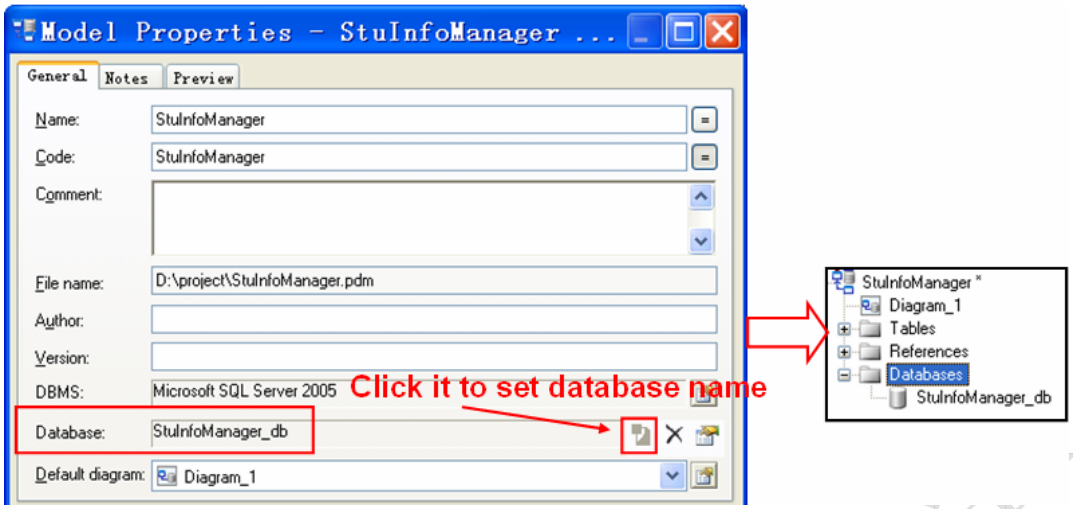
Select Tools->Generate Physical Data Model for a dialog. Fill in Name and Code and select database SQL Server 2008 for DBMS.



(4) Set database name

Right click Physical Model and select Properties for a dialog window. Enter database name

StuInfoManager_db needed for exporting script.

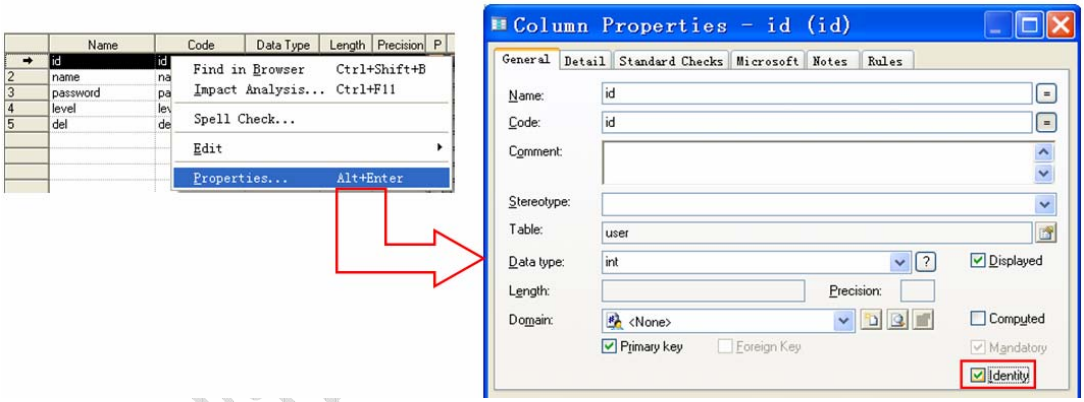


(5) Alter primary key id of Physical Model into auto-increment

Double click the physical model name in workspace to view it.

Double click physical model table to open properties window, and select Columns.

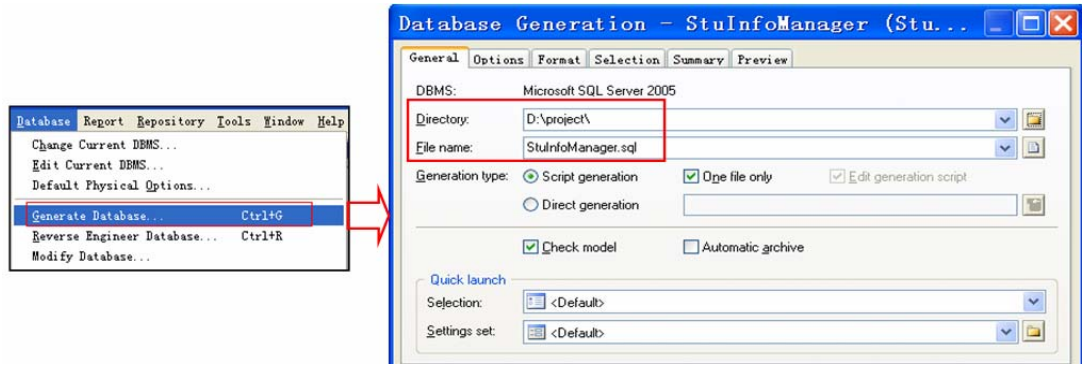
Select Columns label and choose id line. Right click and select Properties. Check “Identity” for the id property.



(6) Export PDM into SQL script

Choose “Database-> Generate Database” from menu for a database script generation dialog.

Select Directory and File name for the script and confirm by clicking “Yes” to complete the process of exporting SQL script.



5.3 Database Design Principle and Skills

Database design is to be conducted on the basis of a full understanding of requirements.
A good database design not only fulfills software requirements, but also is easy to maintain and upgrade. Besides, issues like data consistency, redundancy and accessing efficiency are to be taken into consideration at design.

1. A good database design needs to be "Readable".
Name each filed meaningfully.
Context is to be taken into consideration in naming fields, for example, Name is much better than UserName for a user name's field.
No inter-quotations between recording items in one same table. This is to reduce complexity.
To name an associate table, use table names with an underline in between. For example, name associate table student and course by StudentInfo_CourseInfo, or briefly Stu_Cou.

2. A good database design satisfies demands on both space and efficiency.
For example, lower space occupation and instant response time.
As for data item of variable length, you can use varchar to reduce space occupation in half. For data item of fixed length, char is better than varchar to be used in storage.
Do not use BLOB field to save large data. Large binary data storage causes problems of database management and waste of space.
Do not use the default field length. You should specify field length by manual. For example, only 20 characters are needed for the length of name field, but the defaulted length might be as long as 255 characters.

3. A good database design simplifies the design of business logic.
All null fields must be set based on the need of user requirement, rather than design convenience.

4. Data in database table are both static and dynamic.
There are static data and dynamic data in the table. Two different tables are used to store them.
(1) Data like student id, date of birth and name are rarely changing, so we put them in student profile table.
(2) Data like course, contents and grades are inclined to be completed by calculation and are to be manipulated dynamically. So they are put in a specified table like selective course table, instead of student profile table.

5. No business field for key words.
Use meaningful code as primary key of the table. For example, use student id as primary key of student table. But when the generation rule of student id varies, you need to enlarge the length of data by hand in case of a shortage. That's a lot of synchronous modification and maintenance works.
So it is usually suggested that a field irrelevant to the business is used as a primary key for the

table. A unique value is necessary for the field to identify each information item, which has nothing to do with the business at all.

6. No physical deletion will be made on database.

When Deletion is to be made on data in database, it's not true that data is physically removed from table. A logic deletion field del is added in the table and the data is set as Deleted. When manipulating data table, del is used to identify the status of each data item.

In this way that the history data is kept. This is for later statistics and analysis, and providing data support for system optimization.

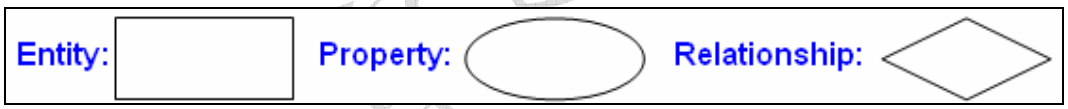
6 Implementation Idea

First you need to identify the relationship of entities by drawing E-R Diagram based Data Dictionary, and optimize. Use PowerDesigner to conduct modeling of conceptual model and physical model. And last generate SQL script for database and table, which is based on SQL Server 2008.

6.1 Conceptual Structure Design (E-R Diagram)

1. E-R Diagram

Entity-Relationship Approach is a way to represent conceptual model, which gives a method to represent entity model, attribute and relationship.



Rhombus shape is used to associate entities. Entities are linked by undirected edges, marking with the link type. The link types are 1:1, 1:n and m:n.

(1) 1:1

At most only one entity in set B is linked with each entity of set A and vice versa.

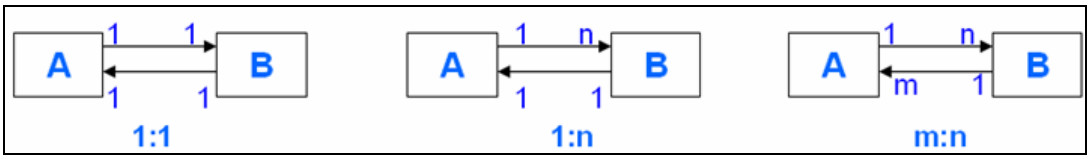
(2) 1:n

There are n entities (n≥0) at most in set B linked with each of the entity in set A. Inversely, there is only one entity at most in set A linked with each of the entity in set B.

(3) m:n

There are n entities (n≥0) at most in set B linked with each of the entity in set A. Inversely there are m entities in set A linked with each of the entity in set B.

Please refer to the following diagrams for the three types.



2. Define entities

It has 9 entities and they are Student, Class, Teacher, Course, Faculty, Dean, Test question, Option and User.

3. Identify the relationship of entities.

(1) Dean<> Faculty: one versus one (1:1), one dean belongs to one certain faculty and one faculty owns one dean.

(2) Faculty<> Teacher: one versus multiple (1:n), one faculty owns multiple teachers but one teacher belongs to only one certain faculty.

(3) Faculty<>Class: one versus multiple (1:n), one faculty opens multiple classes but one class belongs to only one certain faculty.。

(4) Class<> Student: one versus multiple (1:n), one class owns multiple students but one certain student belongs to only one certain class.

(5) Class<> Teacher: multiple versus multiple (n:m), one teacher teaches multiple classes and one class can be taught by multiple teachers.

(6) Class <> Course: multiple versus multiple (n:m), one class can be set with multiple courses and one course can be selected by multiple classes.

(7) Teacher<> Course: multiple versus multiple(n:m), one teacher can teach multiple courses and one course can be taught by multiple teachers.

(8) Student<> Course: multiple versus multiple (n:m), one student can select multiple courses and one course can be selected by multiple students.

(9) Course<> Test question: one versus multiple (1:n), one course can has multiple test questions but one test question belongs to only one certain course.

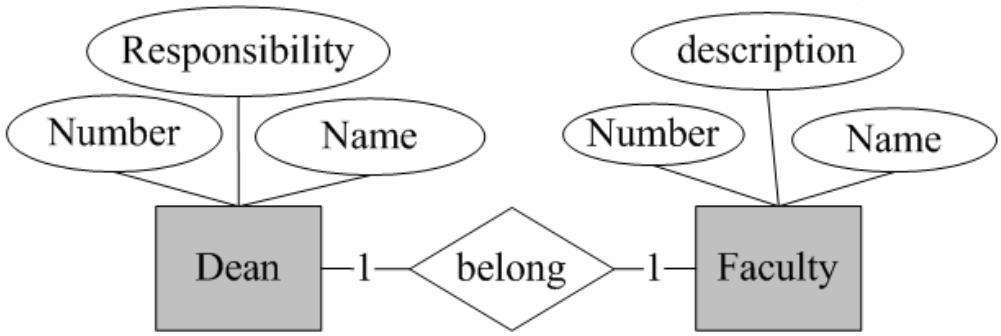
(10) Test question<> Option: one versus multiple (1:n), one test question can has multiple options or answers but one option belongs to only one certain test question.

4. Draw conceptual model(E-R Diagram)

E-R diagram is used to describe. For example, Dean<>Faculty, Class<>Student, Student<>Course.

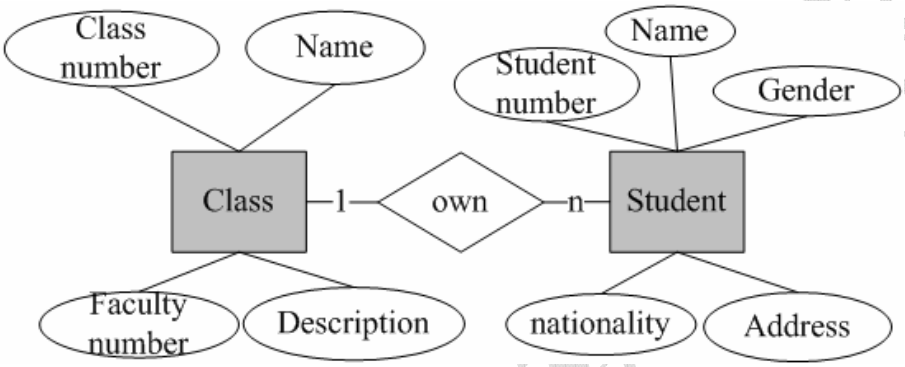
(1) Dean<> Faculty

Relationship is named Belong To. From a dean's perspective, a dean belongs to a certain faculty.



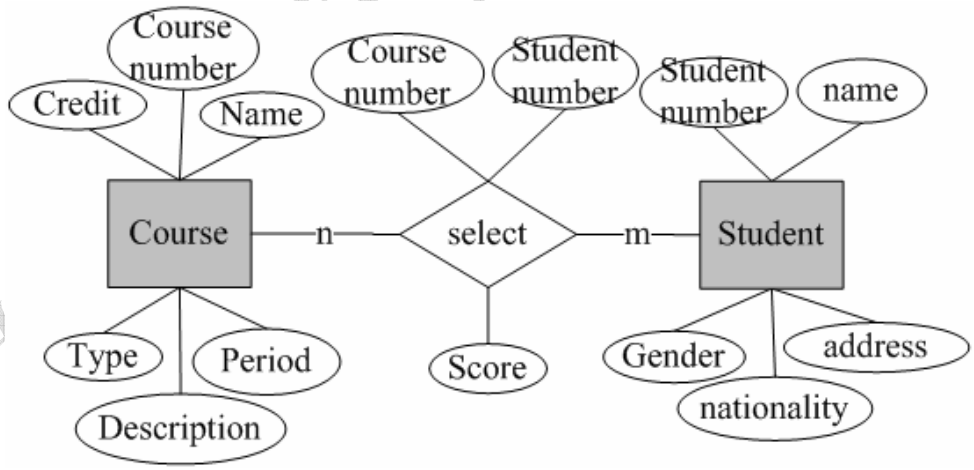
(2) Class<> Student

Relationship is named Own. From a class's perspective, a class owns multiple students.



(3) Student<> Course

Relationship is named Select Course. From a student's perspective, Select Course has an attribute of grade.

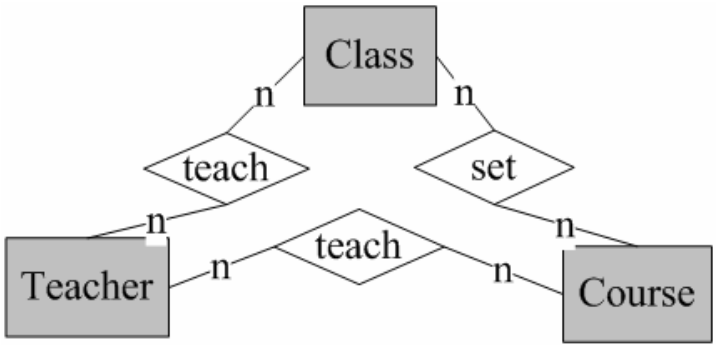


5. Optimize Conceptual Model

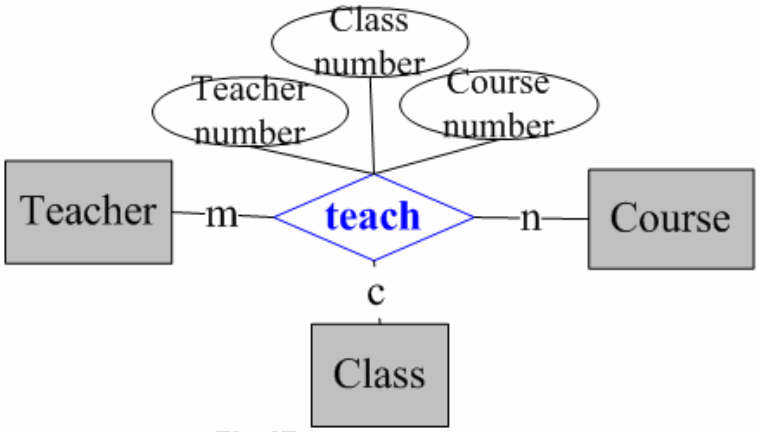
We analyze and draw local E-R Diagram. It focuses on the relationship of two entities. In the local E-R diagram, there might be redundant data and relationships, and parts to be consolidated.

Now we are to consolidate E-R Diagram into a whole integration. In this process, adjustment, revision and optimization will be made to the original local E-R Diagram.

In this conceptual model, you can see they are associated relations between any of the two entities of Teacher, Class and Course. Please see the picture below.



The relationship of 3 entities is actually not independent. When a teacher teaches a class, the course is already assigned. The same is true when a course is assigned to a class, the teacher is designated too. Therefore, an associated relationship exists between 3 entities. That is to describe in a ternary relationship. In order to find the associated information between teacher, class and course, the associated relationship should be designated with attributes such as Teacher number, Class number and Course number.



6.2 Logical Structure Design (Database Table Structure)

Relational data mode is applied here. First convert E-R diagram into relational model. And then convert it into a specific data model according to DBMS, the SQL Server 2008. Finally, optimize it.

1. To convert E-R diagram into relational model

Convert E-R diagram into relational mode and mark relation code with underline _. Please see the following.

(1) User (Name, Password, Role)

(2) Faculty (Faculty number, Faculty name, Dean number, Profile)

(3) Dean (Dean number, Name, Duty)

(4) Class (Class number, Class name, Faculty number, Profile)

(5) Teacher (Teacher number, Name, Gender, Date of birth, Political status, Contact, Faculty number, Title)

(6) Student (Student id, Name, Gender, Nationality, Political status, Address, Date of birth, Class number)
(7) Course (Course number, Course name, Credit, Credit hour, Type, Brief)
(8) Teaching (Teacher number, Course number, Class number, Teaching schedule)
(9) Selective course (Student id, Course number, Grade)
(10) Test question (Item number, Course number, Type, Question, Analysis)
(11) Option (Option number, Test item number, Options, Content, Identifier)

2. Draw two-dimensional table

Relational model can be described in a two-dimensional table, with explicit model data like data table, column, data type, length, null, primary key and foreign key.

The structure of database is based on SQL Server 2008. For example, the description of student table.

t_student				
Name	Code	Type/Length	Instruction	Null
id	id	int	Primary key, auto-increase by 1 each time	N
Student id	student_id	int	Mark each unique student	N
Name	name	varchar/20		N
Gender	gender	int	1 male, 2 female	N
Nationality	nationality	varchar/20		N
Political status	politics_status	int	1 folks，2 league member，3 party member	N
Address	address	varchar/100		
Date of birth	birthday	varchar/50		N
Class number	cla_id	int	Foreign key,quote id in table t_class.	N
Mark	del	int	0 Deleted Not, 1 Deleted	N

For the complete database table structure, please see "**Database Table Structure (Student Information Management System)**".

6.3 Modeling by PowerDesigner

After two-dimensional table design, database and table script can be compiled based on SQL Server 2008.

1. Disadvantages of compiling script by hand

To write script by hand is not easy and is inclined to have error. The script does not show the relationship of entities visually. It's complex to modify script if there is any change in model.

2. Modeling by PowerDesigner

While modeling by PowerDesigner enables us to view the relationship clearly. And conversion

can be made freely among CDM, PDM and SQL. When data information changes or we need to move to a new database, for example when SQL Server is replaced by MySQL, what you do is merely to modify the model to generate a new script. It's much easier to maintain and manage.

Tip:

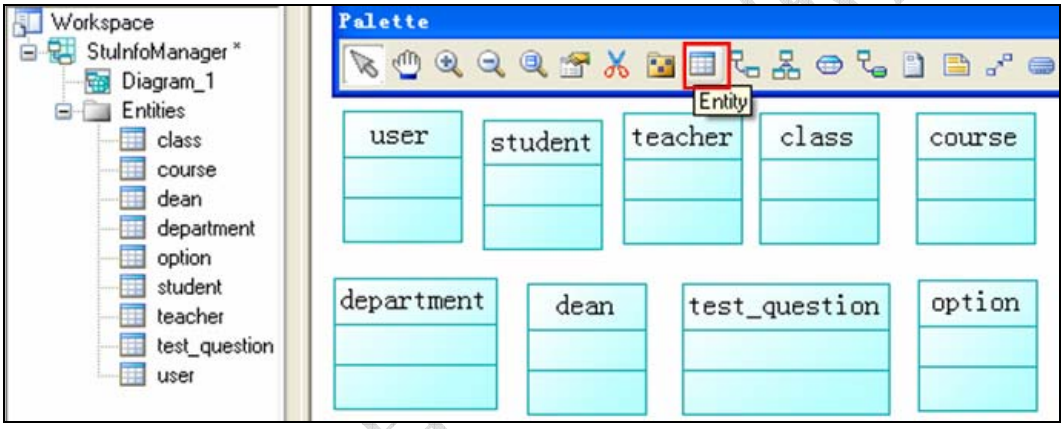
PowerDesigner is usually the tool used for database design in project development. First establish conceptual model, and then physical model based on like SQL Server. Last generate database script of any concrete database by using the physical model.

6.3.1 Establish Conceptual Model by PowerDesigner

1. Open PowerDesigner and create conceptual model named StuInfoManager.

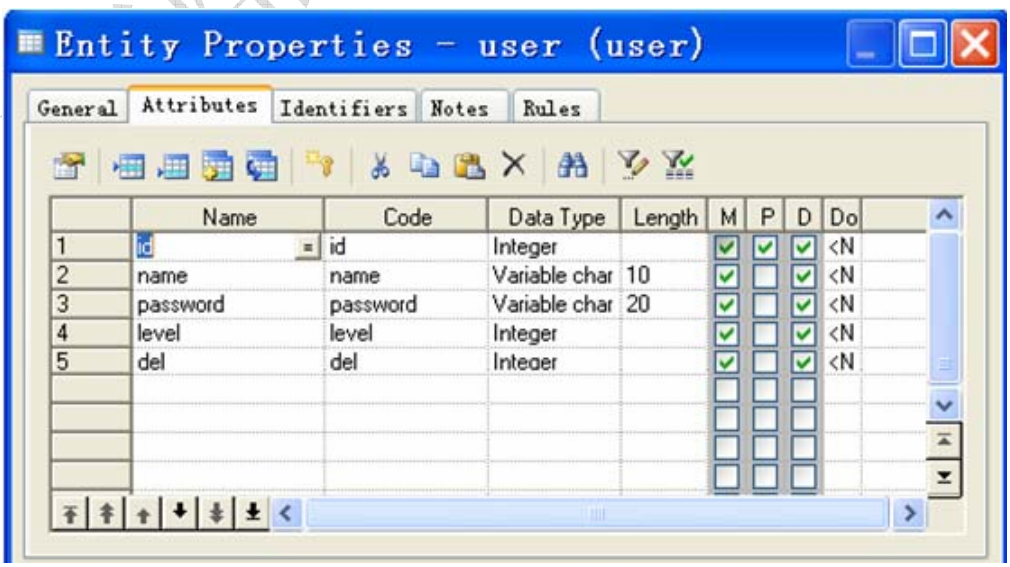
2. Add 9 entities into it.

"user", "student", "teacher", "class", "course", "department", "dean", "test_question" and "option".

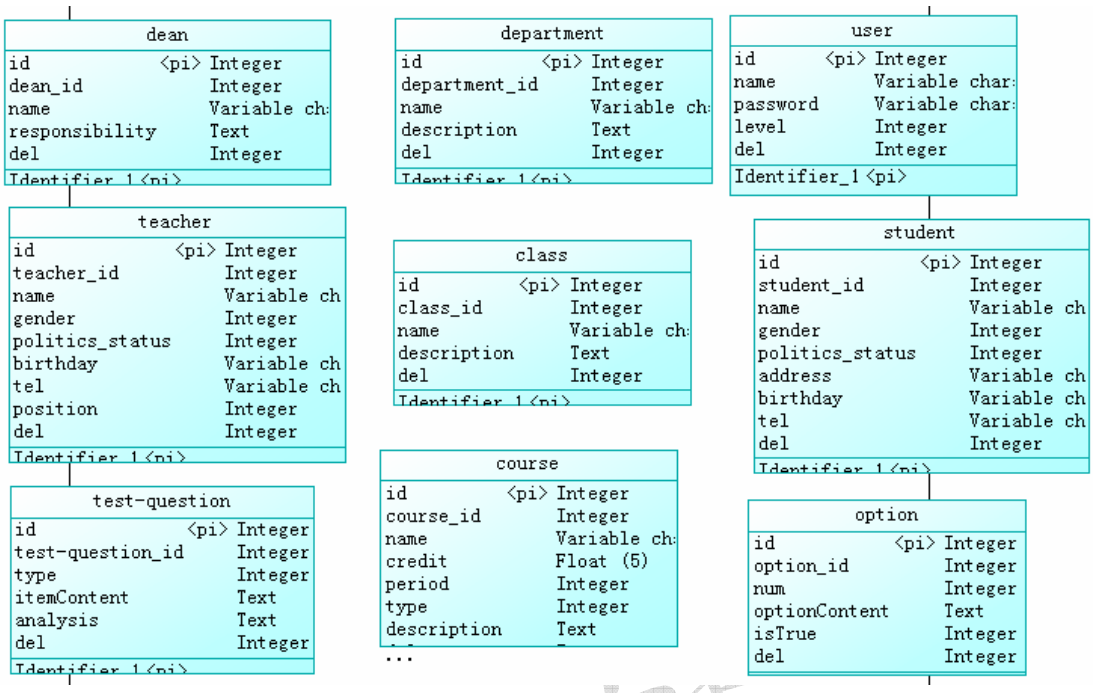


3. Assign attributes to entities.

Double click the entity and fill in columns like name, code, etc. in General labels.

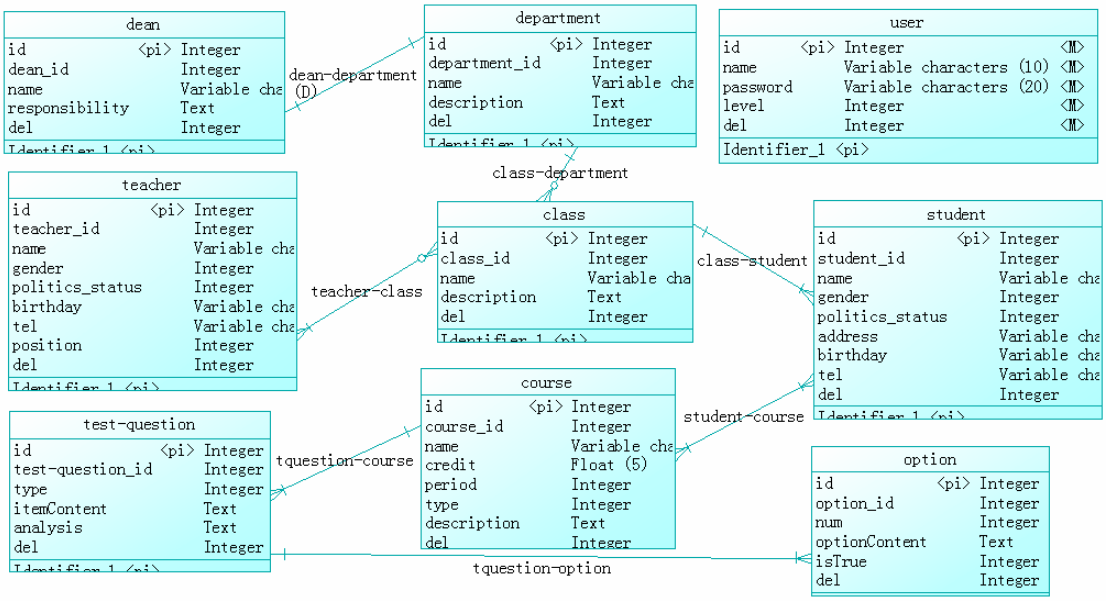


Assign attributes to all entities respectively. Please note that you need to uncheck “Unique code” in “Model Options”, in order to avoid reusing names.



4. Establish relationship of entities.

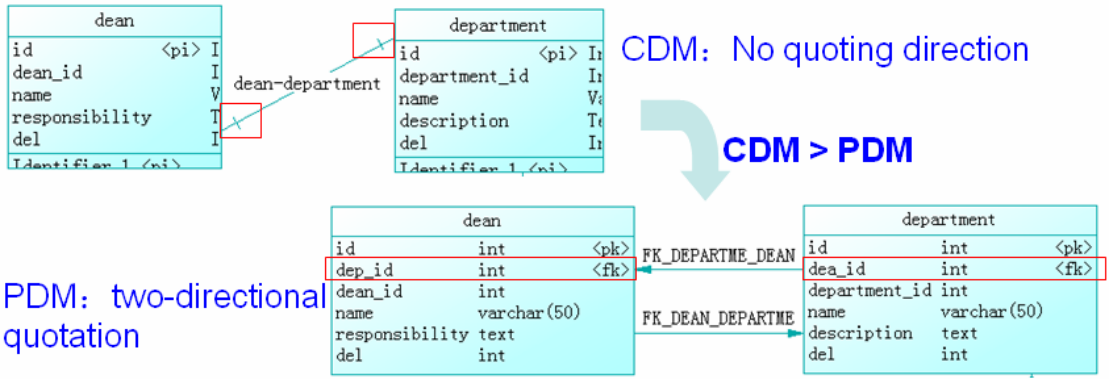
Please set according to relationship of entities defined in E-R diagram.



Remarks

(1) To prevent dual quotations in one versus one relationship (foreign key)

No quoting direction is specified in PowerDesigner by default for the one versus one relationship. So a two-directional quotation can be generated. That is to say, in PDM, two tables would quote primary keys of each other and that forms a looping dependency.



Problems of looping dependency:

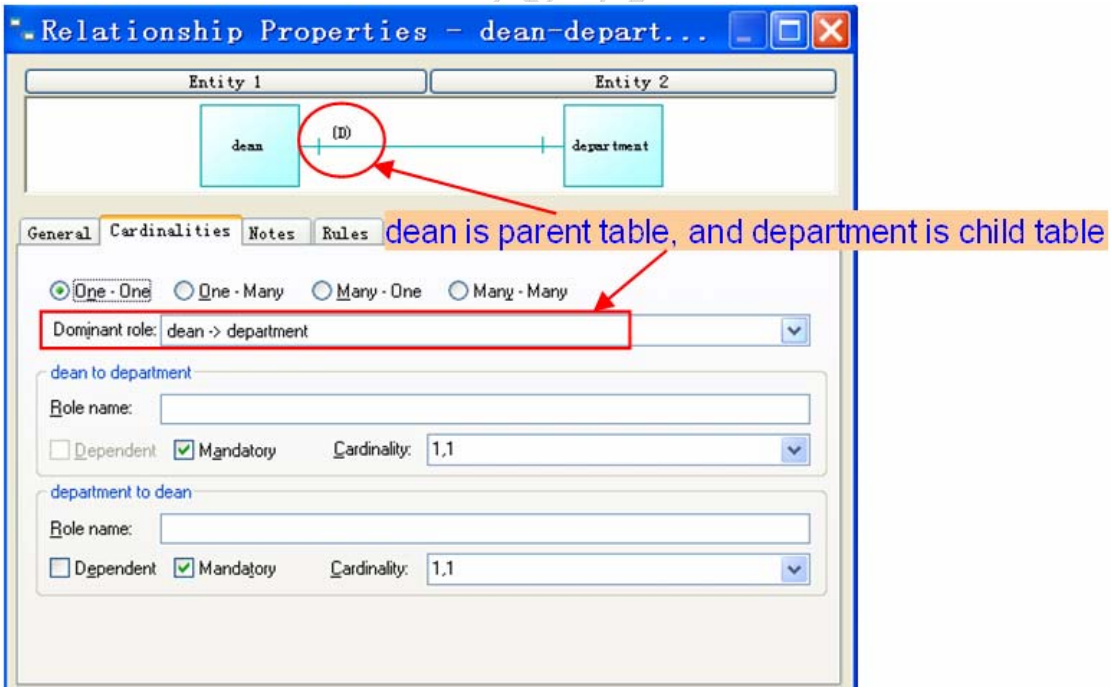
It's impossible to find out which is parent table or which is child table, because they both quoted primary key of the other one as foreign key. No valid constrain relationship is generated, nor valid SQL script is to be made.

Solutions

You need to alter model into one-directional dependency way, relationship of parent table and child table, to generate a genuine SQL script.

Define the dominating direction of relationship in CMD. The dominating entity marked with D converts into parent table in the process of PDM generation.

Double click one versus one relation-> **Detail-> Cardinalities -> Dominant** role to select the dominating relationship.



(2) Ternary relationship design between "teacher", "class" and "course".

Solution 1, design the relationship as an entity.

The "teaching" is the associate link between "teacher", "class" and "course". Design a relational

entity “teaching” first in CMD before defining relations among 4 entities.

Solution 2, establish relationship in PDM.

Establish relationship of two entities first like “teacher” and “class”. And establish the relationship of another table, which is transformed from entity, with the associate table later when design PDM.

We choose solution 2 in this application.

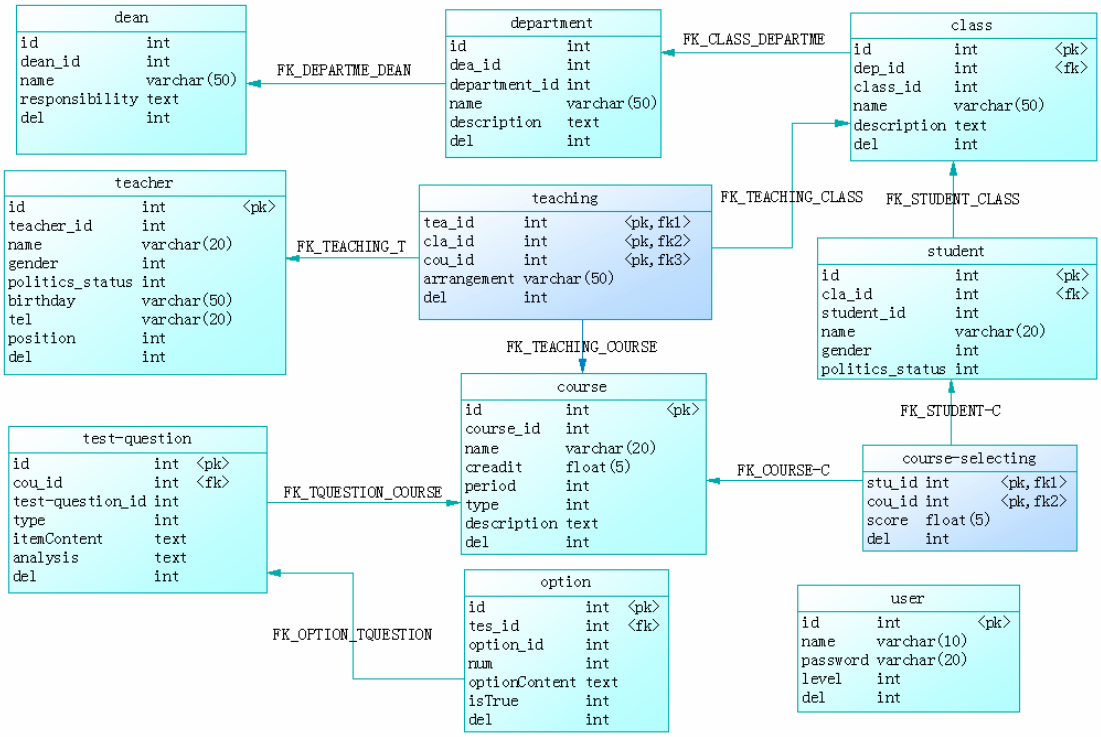
6.3.2 Establish Physical Model by PowerDesigner

1. Convert conceptual model into physical model

Select conceptual model **StuInfoManager**, and set both name and code of the generated physical model PDM as **StuInfoManager**. Select SQL Server 2008 for DBMS.

If any error occurs in the generation of PDM, you need to check whether any conflict or error in CDM. After PDM is successfully generated, you can make adjustment or modification on the table like name or field of the associate table.

A unified naming rule for the table is t_table name. For example, student > t_student. Name tables in PDM according to this rule.



Note: Primary key of “t_course” table should be quoted by “t_teaching” table. The “t_course_selecting” table contains fields of “score” and “del”.

2. Set database name

Select physical model and right click it. Choose Properties for a dialog window. Add database name **StuInfoManager_db** needed for exporting script in Database bar.

3. Alter primary key id of the table in physical model into auto-increment

Because a field irrelevant to business is taken as primary key of the table, it can be managed by DBMS itself by setting auto-increment.

6.3.3 Export SQL Script by PowerDesigner

Set storage path and script name StuInfoManager.sql for the script file and export it.

Note: Model examination will be conducted at script generation. If problem or error occurs, you need to modify PDM.

Tip: Database script generated by modeling tool needs adjustment according to enterprise requirement and relative standard for a final optimized version.

Part of the script is shown here.

```
drop database StuInfoManager_db
go

create database StuInfoManager_db
go

use StuInfoManager_db
go

/*==============================================================*/
/* Table: t_dean                                                */
/*==============================================================*/
create table t_dean (
   id int not null identity(1, 1) primary key,
   dean_id              int                 not null,--Mark each unique dean
   name                 varchar(50)         not null,
   responsibility       text                null,
   del                  int                 not null,--0: Deleted not, 1: Deleted
)
go
```

