

# 《数据库设计与应用开发》 实验指导书

新疆大学软件学院

钱育蓉

# 目录

实验一：数据库的基本查询与更新.....	4
1 数据库软件的安装与启动.....	6
1.1 基于 MySQL 平台的数据库基本查询与更新.....	12
2 通过 SQL 语句创建和操作表.....	12
2.1 使用 SQL 语句创建表.....	12
2.2 使用 SQL 语句修改表.....	13
2.3 使用 SQL 语句添加、修改和删除表数据.....	13
2.4 使用 SQL 语句查询表内容.....	13
3 通过图形化工具创建和操作表.....	15
3.1 通过 SQLyog 向导创建表.....	15
实验二：数据库的高级查询与统计.....	18
1 实验数据表及其导入.....	18
1.1 数据表的结构.....	18
1.2 数据表的数据实例.....	20
1.3 实验数据的导入.....	22
2 数据的高级查询与统计.....	24
2.1 数据查询与统计的实例.....	24
2.2 数据查询与统计的练习.....	26
附录 1：建表 SQL 语句.....	26
附录 2：插入数据 SQL 语句.....	28
实验三：完整性约束、视图、存储过程和数据安全.....	31
1 完整性约束.....	32
1.1 实体完整性约束.....	32
1.2 参照完整性约束.....	33
1.3 业务规则约束.....	33
2 视图、存储过程与索引.....	34
2.1 视图的创建和查询.....	34
2.2 存储过程.....	35
2.3 索引.....	36
3 数据安全.....	36
3.1 用户管理.....	36
3.2 视图.....	36

4 数据备份与恢复 .....	37
4.1 数据备份 .....	37
4.2 数据恢复 .....	38
实验四：使用 PowerDesigner设计数据库.....	39
1 安装与破解 PowerDesigner 软件.....	40
1.1 安装 PowerDesigner 软件.....	40
1.2 破解 PowerDesigner 软件.....	40
2 数据库需求分析 .....	42
3 创建 CDM 概念数据模型 .....	43
3.1 创建 CDM 文件 .....	43
3.2 添加实体对象和联系对象.....	44
3.3 确定对象之间的关系.....	46
3.4 完成 CDM 图 .....	48
附件.....	49
实验五：使用 PowerDesigner 创建 MySQL 数据库.....	50
1 生成 PDM 图.....	50
2 生成数据库脚本文件.....	53
3 配置 MySQL 数据源 .....	55
3.1 安装 MySQL 的 ODBC 驱动 .....	55
3.2 在 PowerDesigner 中配置数据源.....	55
4 在 PowerDesigner 中建立数据库.....	59
4.1 连接 MySQL 数据库 .....	59
4.2 执行 SQL 脚本.....	60
附件.....	61
实验六/七：数据库综合实验（一）/（二） .....	63
[实验内容及要求].....	63
Case -高校实验室管理统.....	64

## 实验一：数据库的基本查询与更新

### [目的和意义]

了解 MySQL 数据库的安装、启动、关闭过程，掌握数据库用户的创建方法;熟练掌握通过基本 SQL 语句来创建和操作表;了解通过图形化数据库管理工具对数据库进行基本操作。

### [实验内容]

安装 MySQL 数据库，向数据库中增加学院 ( department ) 和学生 ( student ) 两个数据库表。

#### ◇ 数据表的结构

表1-S1 department 表 ( 院系信息表结构 )

字段名称	类型	宽度	允许空值	主键	说明
dno	char	6	NOT NULL	是	院系编号
dname	varchar	20	NOT NULL		院系名称
dean	varchar	8	NULL		院系负责人
addr	varchar	40	NULL		院系办公地址
phone	varchar	40	NULL		联系电话

表1-S2 student 表 ( 学生信息表结构 )

字段名称	类型	宽度	允许空值	主键	说明
Sno	char	8	NOT NULL	是	学生学号

sname	varchar	8	NOT NULL	学生姓名
sex	char	2	NULL	学生性别
native	varchar	20	NULL	籍贯
birthday	date		NULL	学生出生日期
dno	char	6	NULL	学生所在院系编号 ( 外键 )
classno	char	4	NULL	班级号
entime	date		NULL	学生入校时间
homeAddr	varchar	48	NULL	学生家庭住址
Phone	varchar	11	NULL	学生联系

◇ 数据表的数据实例

表 1-D1 department 表 ( 院系信息表数据 )

dno	dname	header	addr	phone
500000	人文学院	刘东妮	长沙市	8305424
510000	外国语学院	万世平	长沙市	8304551
520000	艺术与设计学院	周曼玲	长沙市	8304530
530000	法学院	李平	长沙市	8304578
540000	经济与管理学院	刘小明	长沙市	8304517
560000	生命科学学院	王林林	长沙市	8305204
590000	软件学院	张中华	长沙市	8305682
600000	建筑学院	马力	长沙市	8304501

表 1-D2 student 表 ( 学生信息表数据 )

sno	sname	sex	birthday	dno	classno
20080001	李毅	男	1985-1-2	590000	0801
20080002	厉超	男	1984-3-8	590000	0801
20080003	梁冰龙	男	1984-7-9	590000	0801
20080004	罗煦	男	1985-10-12	590000	0801
20080005	马晓睿	男	1983-11-1	590000	0802
20080006	聂立业	男	1984-6-3	590000	0802
20080007	王实	男	1984-2-11	590000	0802
20080008	伍卉	女	1985-1-3	590000	0802
20080009	伍琼	男	1985-12-11	590000	0803
20080010	曾文超 3	男	1984-7-5	590000	0803
20080011	黄文杰	男	1984-7-5	590000	0803
20080012	龙 淼	男	1985-12-11	590000	0803
20080013	何晓明	男	1984-8-13	590000	0804
20080014	刘 寅	男	1984-5-7	590000	0804
20081001	邢晓静	女	1984-8-13	590000	0804
20080016	李明伟	女	1986-3-9	590000	0804
03080001	童熙	女	1985-10-16	590000	0805
20080018	朱俊华	男	1985-12-3	590000	0805

## [实验步骤]

### 1 数据库软件的安装与启动

#### 1.1 基于 MySQL 平台的数据库基本查询与更新

##### 1.1.1 MySQL 的安装与配置

1. 下载MySQL安装文件，双击安装，下图中点击“Next”

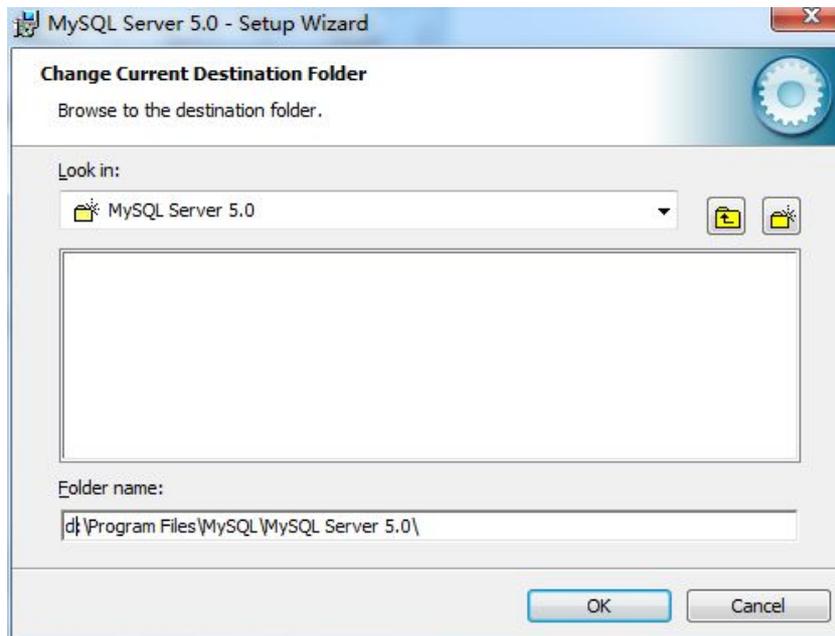
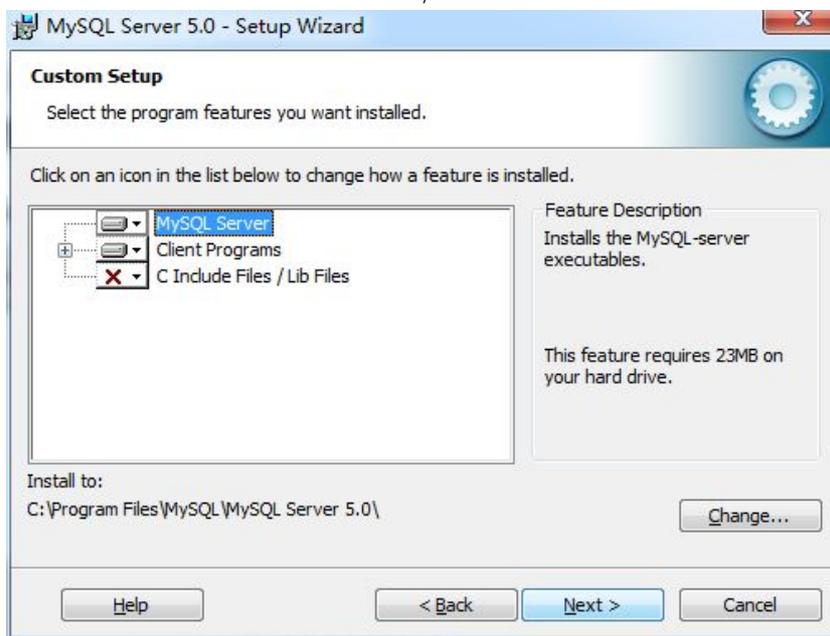


2. 选择“Custom”，单击“Next”

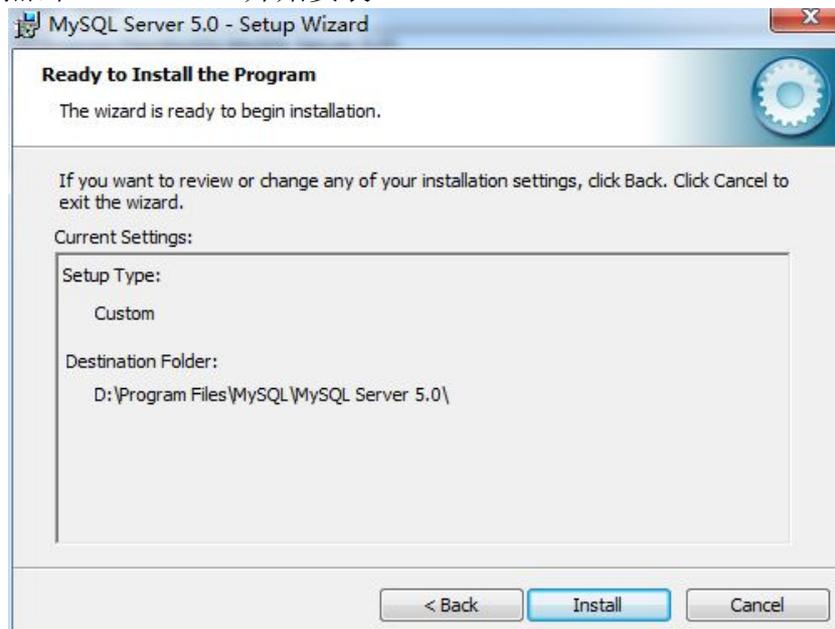


注意：选择“Custom”是为了修改安装目录，为了数据安全，不要把MySQL安装在系统盘

3. 点击“Change”修改安装目录，单击“OK”，然后“Next”



#### 4. 点击“Install”开始安装



#### 5. 一直“Next”

MySQL Enterprise



**Enterprise**

A MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services, and support to ensure your business achieves the highest levels of reliability, security, and uptime.

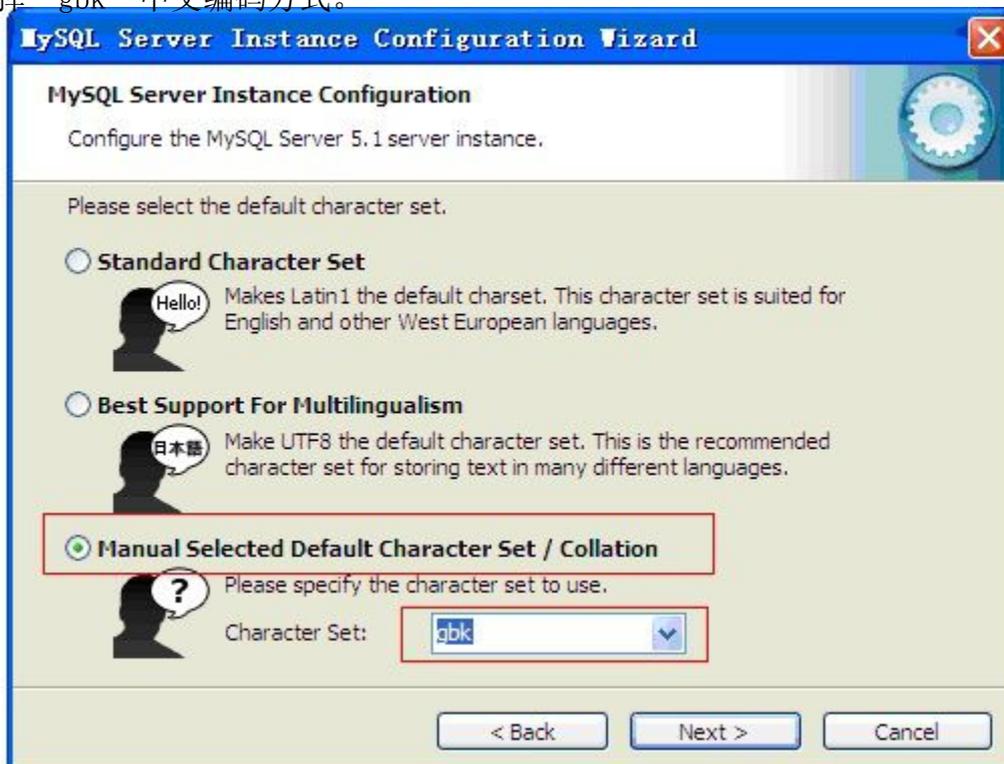
An Enterprise Subscription includes:

- 1. The MySQL Enterprise Server** - The most reliable, secure, and up-to-date version of the world's most popular open source database.
- 2. MySQL Enterprise Monitor Service** - An automated virtual database assistant.
- 3. MySQL Production Support** - Technical and consultative support when you need it, along with service packs, hot-fixes, and more.

For more information click [\[More..\]](#) or visit [www.mysql.com/enterprise](http://www.mysql.com/enterprise)



6. 配置mysql。点击上图中“Finish”进入配置阶段，而后一直“next”至下图界面，选择单选项“ManualSelectedDefaultCharacterSet/ Collation”，在“CharacterSet:”中选择“gbk”中文编码方式。



7. 而后“next”至下图，选中“IncludeBinDirectoryinWindowsPath”，用来支持在WindowsDOS中登陆MySQL。



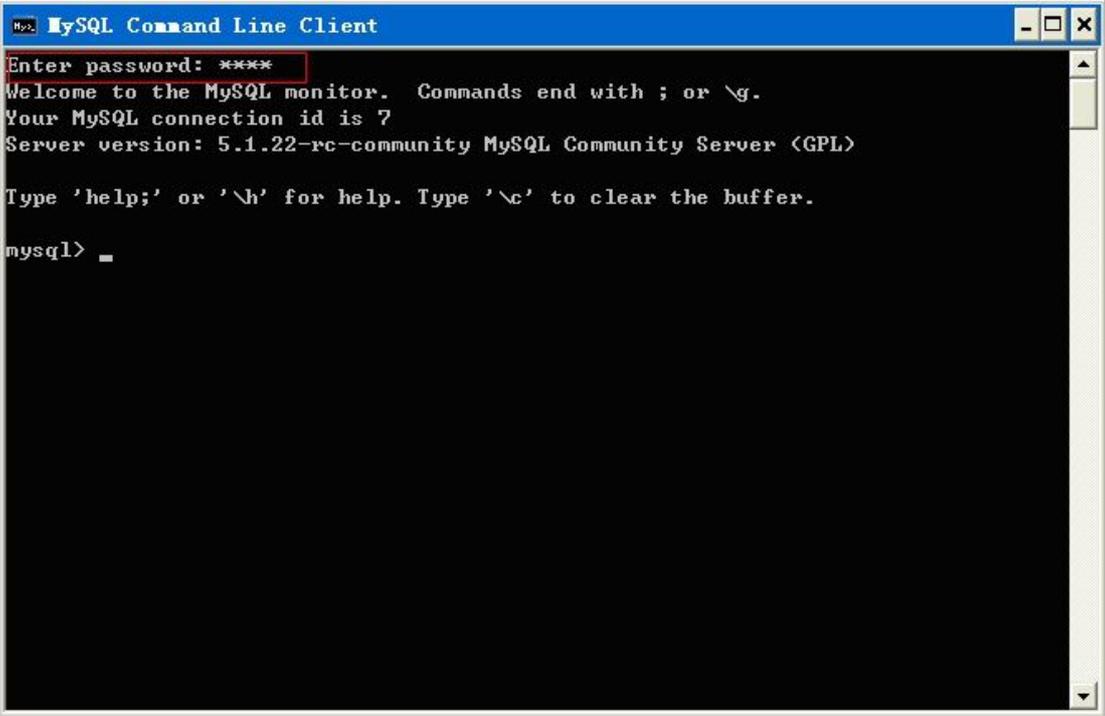
8. 而后“next”至下图，设置登陆密码，建议设置为“root”，系统的默认根用户名即为“root”，方便记忆。



9. 最后“next”至配置完成。

### 1.1.2 MySQL 的启动

选择“开始”→“所有程序”→“MySQL”→“MySQL Server 5.1”→点击“MySQL Command Line Client”运行如下命令行界面，输入密码（root）即进入系统，如下图：



```

MySQL Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.22-rc-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _

```

MySQL暂不创建新用户，我们直接用root用户来进行基本操作。

- MySQL中的基本命令：

show databases/ tables； 显示用户下面的所有数据库/表的名字。

create database/ table NAME； 创建新数据/表。

drop database/ table NAME； 删除某一具体数据库/表。

use database\_name； 登入某一具体数据库。

desc table\_name； 显示表中的字段属性。

- 通过Windows DOS登陆.

在桌面选择“开始”→“运行”→输入“cmd”点击“确认”进入 DOS 界面；

而后使用命令“mysql -u用户名 -p密码”，如：mysql -uroot -proot ，注意：该命令后不要用分号。

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.1.22-rc-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _

```

## 2 通过 SQL 语句创建和操作表

### 2.1 使用 SQL 语句创建表

```

DROP TABLE department;
CREATE TABLE department
(
    dno char (6) PRIMARY KEY, #院系编号(主键)
    dname varchar (20) NOT NULL, #院系名称
    dean varchar (8), #院系负责人
    addr varchar (40), #院系办公地址
    phone varchar (40) #联系电话
)
;
DROP TABLE student;
CREATE TABLE student
(
    sno char (8) PRIMARY KEY, #学生学号(主键)
    sname varchar (8) NOT NULL, #学生姓名
    sex char (2), #学生性别
    native varchar (20), #籍贯
    birthday date, #学生出生日期
    dno char (6), #学生所在院系编号 (外键)
    classno char (4), #班级号
    entime date, #学生入校时间

```

```

homeAddr varchar (48), #学生家庭住址
phone varchar (11) #学生联系电话
)
;

```

(MySQL中的行注释为#)

## 2.2 使用 SQL 语句修改表

- ◇ 修改表中字段属性：  
ALTER TABLE student MODIFY phone varchar(50);
- ◇ 修改表名：  
ALTER TABLE student RENAME TO stu;

## 2.3 使用 SQL 语句添加、修改和删除表数据

- ◇ 添加数据：  
INSERT INTO department  
( dno,dname,dean,addr,phone )  
VALUES('590000','软件学院','张中华','长沙市','8305682')  
;  
  
INSERT INTO student  
(sno,sname,sex,birthday,dno,classno)  
VALUES('20080001','李毅','男','1985-1-2','590000','0801') ;
- ◇ 更新表中的数据：  
例如，将李毅的性别改为女，如下：  
UPDATE student SET sex='女' where sname='李毅';
- ◇ 删除表中的数据：  
例如，删除student表中sno为20080001的学生信息：  
DELETE FROM student WHERE sno='20080001';

## 2.4 使用 SQL 语句查询表内容

- ◇ 选出 student 表中的所有数据：  
SELECT \* FROM student;

- ◇ 选出 student 表中的部分属性 :  

```
SELECT sno,sname,sex,birthday FROM student;
```
- ◇ 选出student表中所有性别为男的学生的学号，姓名，性别和出生日期 :  

```
SELECT sno,sname,sex,birthday FROM student WHERE sex='男';
```
- ◇ 选出student表中1985年1月1日以后出生的学生 :  

```
SELECT sno,sname,sex,birthday FROM student
WHERE birthday >'1985/01/01';
```
- ◇ 以下两个使用了模糊查询：%代替任意个字符，\_代表一个字符。  

```
SELECT sno,sname FROM student WHERE sname LIKE '李%';
SELECT sno,sname FROM student WHERE sname LIKE '李_';
```
- ◇ 多个条件查询 :  

```
SELECT sno,sname,sex,birthday
FROM student
WHERE (dno='590000')AND (birthday>'1985/01/01');
```
- ◇ 选出籍贯为空的数据（null是不存在，不能用“=”来比较） :  

```
SELECT * FROM student WHERE native is null;
```
- ◇ 以下三句为排序，DESC为降序，ASC为升序 :  

```
SELECT sno,sname,sex,birthday,dno FROM student
ORDERBY dno,birthdayDESC;
#降序
SELECT sno,sname,sex,birthday FROM student
ORDERBY birthdayASC;
#升序
SELECT sno,sname,sex,birthday,dno FROM student
ORDERBY 5,4DESC;
```

#5，4对应select后面的顺序，这里即为dno,birthday。

- ◇ 使用DISTINCT消除重复值：

```
SELECT DISTINCT sex FROM student;
```

- ◇ 以下两句在Oracle/MySQL中等价：

```
SELECT sno AS 学号, sname AS 姓名, sex AS 性别  
FROM student;  
SELECT sno 学号, sname 姓名, sex 性别  
FROM student;
```

- ◇ 得出各院男女生人数：

```
SELECT dno 学院, COUNT(*) 学生人数 FROM student  
GROUP BY sex;
```

- ◇ 连接查询：

```
SELECT sno, d.dno, dname  
FROM students, department d  
WHERE s.dno = d.dno;
```

- ◇ 嵌套查询

```
SELECT sno, sname, birthday  
FROM student  
WHERE dno = (SELECT dno FROM student WHERE sname = '刘寅');
```

### 3 通过图形化工具创建和操作表

### 3.1 通过 SQLyog 向导创建表

双击软件图标，进入登陆界面“连接到MySQL主机”，单击“新建”，如下图：



在“NewConnection”对话框点击“确定”，回到“连接到MySQL主机”界面，输入“密码”，点击“连接”登入软件。





## 实验二：数据库的高级查询与统计

### [目的和意义]

进一步掌握基本 SQL 命令和语法，熟练运行子查询、连接查询、嵌套查询、组函数等技术查询出需要的结果，实现对数据库的高级查询与统计。

### [实验内容]

本实验的主要内容：

1. 根据给定的表结构和表数据信息，通过 SQL 语言建立相应的数据表，并导入数据；
2. 通过 SQL 查询语句，实现对数据信息的高级查询与统计。

### [实验步骤]

#### 1 实验数据表及其导入

本实验的数据是模拟一个简单学校信息系统的数据库，共 6 个数据表，其中 4 个实体表和 2 个关系表，实体表为：学院表 ( department )、学生表 ( student )、老师表 ( teacher )、课程表 ( course )；关系表为：老师开课表 ( teacher\_course )，学生选课表 ( student\_teacher\_course )。各表的结构和数据实例都在下面列出。

#### 1.1 数据表的结构

表 2-S1 department 表 ( 院系 信息表 )

字段名称	类 型	宽 度	允许空值	主 键	说 明
dno	char	6	NOTNULL	是	院系 编号
dname	varchar	20	NOTNULL		院系 名称
dean	varchar	8	NULL		院系 负责人
addr	varchar	40	NULL		院系 办公地址
phone	varchar	40	NULL		联系电话

表 2-S2 student 表 ( 学生信息表 )

字段名称	类 型	宽 度	允许空值	主 键	说 明
sno	char	8	NOTNULL	是	学生学号
sname	varchar	8	NOTNULL		学生姓名
sex	char	2	NULL		学生性别

native	varchar	20	NULL		籍贯
birthday	date		NULL		学生出生日期
dno	char	6	NULL		学生所在院系编号 ( 外键 )
classno	char	4	NULL		班级号
entime	date		NULL		学生入校时间
homeAddr	varchar	48	NULL		学生家庭住址
phone	varchar	11	NULL		学生联系电话

表 2-3 teacher 老师表

字段名称	类 型	宽 度	允许空值	主 键	说 明
tno	varchar	8	NOT NULL	是	教师编号
tname	varchar	8	NOT NULL		姓名
rank	varchar	6	NOT NULL		职称
dno	varchar	6	NOT NULL		所在院系编号 ( 外键 )

表 2-S4 course 表 ( 课程信息表 )

字段名称	类 型	宽 度	允许空值	主 键	说 明
cno	varchar	10	NOT NULL	是	课程编号
cname	varchar	20	NOT NULL		课程名称
hours	integer		NOT NULL		授课学时
term	integer		NOT NULL		开课学期
credit	integer		NOT NULL		课程学分
dno	varchar	8	NOT NULL		课程归属院系代码 ( 外键 )

表 2-S5 teacher\_course 老师开课表

字段名称	类 型	宽 度	允许空值	主 键	说 明
tno	varchar	8	NOT NULL	是	教师编号 ( 外键 )
cno	varchar	10	NOT NULL	是	课程编号 ( 外键 )
semester	varchar	7	NOT NULL	是	开课年度和学期

表 2-S6 student\_course\_teacher 表 ( 学生选课表 )

字段名称	类 型	宽 度	允许空值	主 键	说 明
tno	varchar	8	NOT NULL	是	教师编号 ( 外键 )
cno	varchar	10	NOT NULL	是	课程编号 ( 外键 )
semester	varchar	7	NOT NULL	是	开课年度和学期
sno	char	8	NOT NULL	是	学生学号 ( 外键 )
score	integer		NULL		学生成绩

## 1.2 数据表的数据实例

表 2-D1 department 表 ( 院系信息表 )

dno	dname	dean	addr	phone
500000	人文学院	刘东妮	长沙市	8305424
510000	外国语学院	万世平	长沙市	8304551
520000	艺术与设计学院	周曼玲	长沙市	8304530
530000	法学院	李平	长沙市	8304578
540000	经济与管理学院	刘小明	长沙市	8304517
560000	生命科学学院	王林林	长沙市	8305204
590000	软件学院	张中华	长沙市	8305682
600000	建筑学院	马力	长沙市	8304501

表 2-D2 student 表 ( 学生表 )

sno	sname	sex	birthday	dno	classno
20080001	李毅	男	1985-1-2	590000	0801
20080002	厉超	男	1984-3-8	590000	0801
20080003	梁冰龙	男	1984-7-9	590000	0801
20080004	罗煦	男	1985-10-12	590000	0801
20080005	马晓睿	男	1983-11-1	590000	0802
20080006	聂立业	男	1984-6-3	590000	0802
20080007	王实	男	1984-2-11	590000	0802
20080008	伍卉	女	1985-1-3	590000	0802
20080009	伍琼	男	1985-12-11	590000	0803
20080010	曾文超	男	1984-7-5	590000	0803
20080011	黄文杰	男	1984-7-5	590000	0803
20080012	龙淼	男	1985-12-11	590000	0803
20080013	何晓明	男	1984-8-13	590000	0804
20080014	刘寅	男	1984-5-7	590000	0804
20081001	邢晓静	女	1984-8-13	590000	0804
20080016	李明伟	女	1986-3-9	590000	0804
03080001	童熙	女	1985-10-16	590000	0805
20080018	朱俊华	男	1985-12-3	590000	0805

表 2-D3 teacher 表 ( 教师信息表 )

tno	tname	rank	dno
-----	-------	------	-----

2004-213	杨金民	副教授	590000
2002-048	戴壮红	副教授	590000
2001-021	尹庚	讲师	590000
2006-117	杨柳	讲师	590000
2005-187	周波	讲师	590000
2000-093	林亚平	教授	590000
2001-051	杨贯中	教授	500000
2003-011	边耐政	副教授	500000
2002-127	陈浩	副教授	590000

表 2-D4 course 表 ( 课程信息表 )

cno	cname	hours	term	credit	dno
H61030006	数据结构	64	4	4	590000
H61030007	计算器组成原理	32	5	2	590000
H61030008	数据库系统	48	5	3	590000
H61030009	操作系统	48	4	3	590000
H61030012	软件工程	48	6	3	590000
H61030013	计算机网络	64	7	4	590000
T53030001	毛泽东思想概论	32	1	2	500000
T53030002	邓小平理论	32	2	2	500000
X61030005	面向对象程序设计言	32	2	2	590000
X61030012	Web 程序设计	32	6	2	590000
X61030019	数据挖掘	32	8	2	590000
X61030022	网络安全概论	32	7	2	590000
X80002002	软件产品市场营销	16	7	1	590000
X80002003	知识管理	16	3	1	500000

表 2-D5 teacher\_course ( 老师开课表 )

tno	cno	semester
2004-213	H61030008	2010/01
2002-048	H61030008	2010/01
2001-021	H61030008	2010/01
2006-117	X61030005	2010/01
2005-187	X61030005	2010/01
2000-093	H61030013	2009/01
2001-051	H61030013	2009/01
2003-011	X61030012	2009/01
2002-127	X61030012	2009/01
2004-213	X61030019	2010/02
2002-048	X61030019	2010/02

表 2-D6 student\_course\_teacher 表 ( 学生选课表 )

sno	tno	cno	semester	score
20080001	2004-213	H61030008	2010/01	92
20080001	2006-117	X61030005	2010/01	88
20080001	2004-213	H61030008	2009/02	78
20080002	2004-213	H61030008	2010/01	90
20080002	2006-117	X61030005	2010/01	85
20080002	2004-213	H61030008	2009/02	82
20080003	2004-213	H61030008	2010/01	90
20080003	2006-117	X61030005	2010/01	83
20080003	2004-213	H61030008	2009/02	89
20080005	2004-213	H61030008	2010/01	95
20080005	2006-117	X61030005	2010/01	86
20080005	2004-213	H61030008	2009/02	91
20080006	2004-213	H61030008	2010/01	83
20080006	2006-117	X61030005	2010/01	86
20080006	2004-213	H61030008	2009/02	75
20080012	2004-213	H61030008	2010/01	88
20080012	2006-117	X61030005	2010/01	90
20080012	2004-213	H61030008	2009/02	78
20080011	2002-048	H61030008	2010/01	87
20080011	2006-117	X61030005	2010/01	62
20080007	2002-048	H61030008	2010/01	93
20080007	2006-117	X61030005	2010/01	88
20080007	2004-213	H61030008	2009/02	87
20080008	2002-048	H61030008	2010/01	77
20080008	2006-117	X61030005	2010/01	74
20080008	2004-213	H61030008	2009/02	89
20080009	2002-048	H61030008	2010/01	90
20080009	2006-117	X61030005	2010/01	81
20080009	2004-213	H61030008	2009/02	84
20080010	2002-048	H61030008	2010/01	73
20080010	2006-117	X61030005	2010/01	71
20080010	2004-213	H61030008	2009/02	50

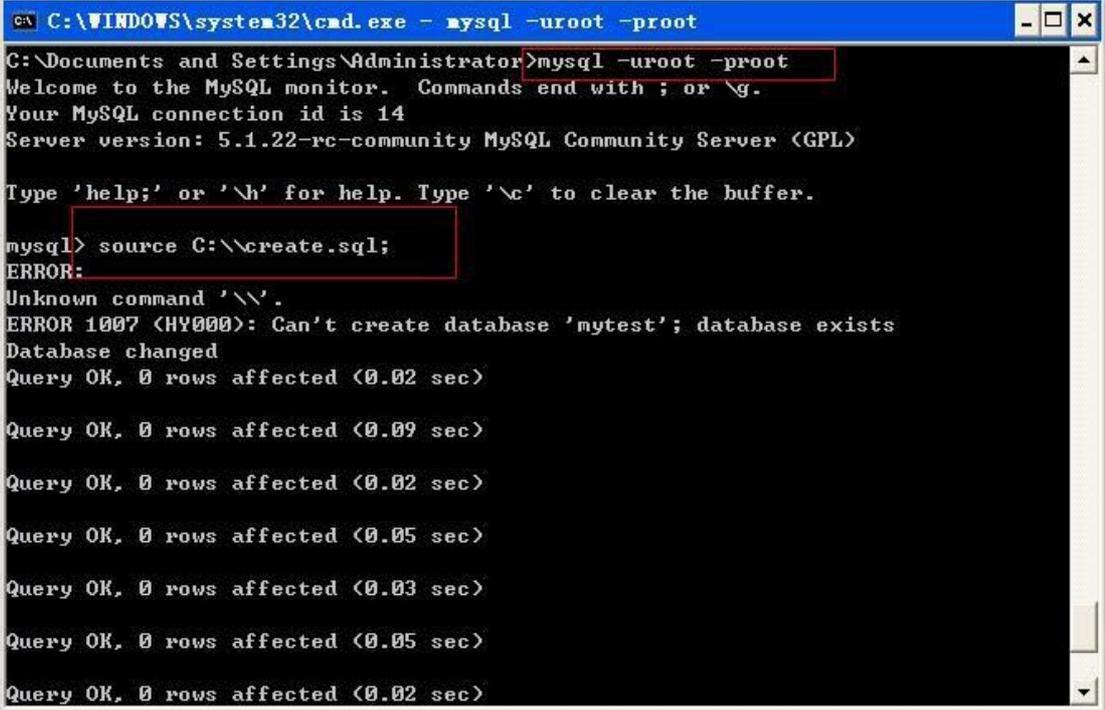
### 1.3 实验数据的导入

本实验中创建表和添加数据的 SQL 语句已经在附录 ( 1,2 ) 中提供，同学们只需要将其分别建成两文件：create.sql 和 populate.sql 文件，然后再依下面

方法导入数据库。

通过命令行方式（或图形界面方式）登入 mysql 数据库，而后我们可以直接用 source 命令来导入表和所有数据，命令格式：source 文件名（包括绝对路径和文件名）。

先导入 create.sql 文件创建 6 个表，如下图：



```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot
C:\Documents and Settings\Administrator>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.1.22-rc-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> source C:\\create.sql;
ERROR:
Unknown command '\'.
ERROR 1007 (HY000): Can't create database 'mytest'; database exists
Database changed
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.09 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.02 sec)
```

然后导入 populate.sql 文件添加各表的实例数据，如下图：

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot
Query OK, 0 rows affected (0.08 sec)

mysql> source C:\populate.sql;
ERROR:
Unknown command '\'.
Database changed
Query OK, 8 rows affected (0.02 sec)
Records: 8 Duplicates: 0 Warnings: 0

Query OK, 18 rows affected (0.02 sec)
Records: 18 Duplicates: 0 Warnings: 0

Query OK, 9 rows affected (0.03 sec)
Records: 9 Duplicates: 0 Warnings: 0

Query OK, 14 rows affected (0.01 sec)
Records: 14 Duplicates: 0 Warnings: 0

Query OK, 11 rows affected (0.02 sec)
Records: 11 Duplicates: 0 Warnings: 0

Query OK, 32 rows affected (0.03 sec)
Records: 32 Duplicates: 0 Warnings: 0

mysql>

```

create.sql 文件中创建新的数据库 mytest, populate.sql 文件导入完毕后, 使用 use mytest 进入 mytest 数据库, 对数据进行查询和统计。

**备注：**

2. 在数据库的图形界面操作工具上也能实现同样的功能, 可以用命令操作, 也可以用菜单操作。

3. create.sql 文件和 populate.sql 文件中的 SQL 语句是基于 MySQL 平台写的, 如果要运行在 Oracle 平台上须进行部分修改。

## 2 数据的高级查询与统计

本部分包括两小节内容, 第一小节为数据查询与统计的4个实例;第二小节为数据查询与统计的8个练习。

### 2.1 数据查询与统计的实例

1、在 2010/01 学期选了杨金民老师开设的“数据库系统”课程的学生姓名、学号、班级清单;

```

selects.sno 学号,sname 姓名,classno 班级
from students,
     teachert,
     coursec,
     student_course_teachersct
where t.tname='杨金民'
and c.cname='数据库系统'
and sct.semester='2010/01'

```

```

and s.sno=sct.sno
and t.tno=sct.tno
and c.cno=sct.cno;

```

2、2010/01 学期“数据库系统”课程的班级平均分清单;

```

select classno 班级, avg(score) 平均分
from student s, course c, student_course_teacher sct
where c.cname= '数据库系统'
and sct.semester= '2010/01'
and s.sno=sct.sno
and c.cno=sct.cno
group by classno;

```

3、2010/01 学期“数据库系统”课程，在软件学院的 2008 级学生中没有选修该课程的同学清单; ( 连接符：外连接 - 左连接、右连接;交集、差集)

```

select s_d.*
from (select s.*
      from student s, department d
      where d.dname = '软件学院'
            and s.snolike '2008%'
            and s.dno=d.dno) as s_d
left join
(select sct.*
 from course c, student_course_teacher sct
 where c.cname= '数据库系统'
       and sct.semester = '2010/01'
       and c.cno=sct.cno) as s_c_sct
using(sno)
where s_c_sct.cno is null;

```

4、2008 级软件学院毕业班，输出其学生成绩排名;输出前 5 名，作为研究生保送生; ( limit

```

select 学号,姓名,avg_score 平均分 from
(select s.sno 学号,sname 姓名, avg(score) as avg_score

```

```

from students, department d, student_course_teachers sct
where s.sno like '2008%'
and d.dname = '软件学院'
and s.dno = d.dno
and s.sno = sct.sno
group by s.sno, s.sname
order by avg_score, s.sno desc) as soft_student
limit 0,5;

```

备注：

常用(分)组函数：SUM( )求总和;AVG( )求平均值;COUNT( )记录的数量;MAX( )求最大值;MIN( )求最小值。除了COUNT(\*),其他所有的组函数都忽略空值。

## 2.2 数据查询与统计的练习

- 1、输出 2010/01 学期每门课程的最高分、最低分;
- 2、输出 2010/01 学期“数据库系统”课程，成绩优秀(90分以上)的学生的清单;
- 3、统计软件学院 2009 年度教师的教学工作量清单;
- 4、软件学院 2009 年度教学工作量未达到要求(150 学时)的教师清单;
- 5、输出软件学院 2008 级毕业班“李毅”同学的成绩清单(课程名称，学分，成绩);
- 6、输出软件学院 2010/01 学期每门课的选修人数清单;
- 7、统计湖南大学各个学院 2009 年度的教学工作量清单;
- 8、对软件学院每个学生，给其家长输出 2010/01 学期成绩单;

## 附录 1: 建表 SQL 语句

◇ create.sql 文件内容 :

```
create database mytest;
use mytest;
```

```
drop table if exists department ;
CREATE TABLE department
(
dno char(6) PRIMARY KEY, #院系编号(主键)
dname varchar(20) NOT NULL, #院系名称
dean varchar(8), #院系负责人
addr varchar(40), #院系办公地址
phone varchar(40) #联系电话
);
```

```
drop table if exists student ;
CREATE TABLE student
(
sno char(8) PRIMARY KEY, #学生学号(主键)
sname varchar(8) NOT NULL, #学生姓名
sex char(2), #学生性别
native varchar(20), #籍贯
birthday date , #学生出生日期
dno char(6), #学生所在院系编号 (外键)
classno char(4), #班级号
entime date, #学生入校时间
homeAddr varchar(48), #学生家庭住址
phone varchar(11) #学生联系电话
);
```

```

drop table if exists teacher ;
CREATE TABLE teacher
(
tno varchar(8) PRIMARY KEY, #教师编号
tname varchar(8) NOT NULL, #姓名
rank varchar(6) NOT NULL, #职称
dno varchar(6) NOT NULL #所在院系编号 (外键)
);

```

```

drop table if exists course ;
CREATE TABLE course
(
cno varchar(10) PRIMARYKEY, #课程编号
cname varchar(20) NOT NULL, #课程名称
hours integer NOT NULL, #授课学时
term integer NOT NULL, #开课学期
credit integer NOT NULL, #课程学分
dno varchar(8) NOT NULL #课程归属院系代码 (外键)
);

```

```

drop table if exists teacher_course ;
CREATE TABLE teacher_course
(
tno varchar(8) NOT NULL, #教师编号 (外键)
cno varchar(10) NOT NULL, #课程编号 (外键)
semester varchar(7) NOT NULL #开课年度和学期
);

```

```

drop table if exists student_course_teacher ;
CREATE TABLE student_course_teacher
(
tno varchar(8) NOT NULL, #教师编号 (外键)
cno varchar(10) NOT NULL, #课程编号 (外键)
semester varchar(7) NOT NULL, #开课年度和学期
snochar(8) NOT NULL, #学生学号 (外键)
score integer #学生成绩
);

```

## 附录 2: 插入数据 SQL 语句

◇ populate.sql 文件内容 :

```
use mytest;
```

```
#loading department
```

```
INSERT INTO department(dno,dname,dean,addr,phone)
VALUES
```

```
('500000','人文学院','刘东妮','长沙市','8305424'),
('510000','外国语学院','万世平','长沙市','8304551'),
('520000','艺术与设计的学院','周曼玲','长沙市','8304530'),
('530000','法学院','李平','长沙市','8304578'),
('540000','经济与管理学院','刘小明','长沙市','8304517'),
('560000','生命科学学院','王林林','长沙市','8305204'),
('590000','软件学院','张中华','长沙市','8305682'),
('600000','建筑学院','马力','长沙市','8304501');
```

```
#loading student
```

```
INSERT INTO student(sno,sname,sex,birthday,dno,classno)
VALUES
```

```
('20080001','李毅','男','1985-1-2','590000','0801'),
('20080002','厉超','男','1984-3-8','590000','0801'),
('20080003','梁冰龙','男','1984-7-9','590000','0801'),
('20080004','罗煦','男','1985-10-12','590000','0801'),
('20080005','马晓睿','男','1983-11-1','590000','0802'),
('20080006','聂立业','男','1984-6-3','590000','0802'),
('20080007','王实','男','1984-2-11','590000','0802'),
('20080008','伍卉','女','1985-1-3','590000','0802'),
('20080009','伍琼','男','1985-12-11','590000','0803'),
('20080010','曾文超','男','1984-7-5','590000','0803'),
('20080011','黄文杰','男','1984-7-5','590000','0803'),
('20080012','龙淼','男','1985-12-11','590000','0803'),
('20080013','何晓明','男','1984-8-13','590000','0804'),
('20080014','刘寅','男','1984-5-7','590000','0804'),
('20081001','邢晓静','女','1984-8-13','590000','0804'),
('20080016','李明伟','女','1986-3-9','590000','0804'),
('03080001','童熙','女','1985-10-16','590000','0805'),
('20080018','朱俊华','男','1985-12-3','590000','0805');
```

```
#loading teacher
```

```
INSERT INTO teacher(tno,tname,rank,dno)
VALUES
```

```
('2004-213','杨金民','副教授','590000'),
('2002-048','戴牡红','副教授','590000'),
('2001-021','尹庚','讲师','590000'),
('2006-117','杨柳','讲师','590000'),
('2005-187','周波','讲师','590000'),
('2000-093','林亚平','教授','590000'),
('2001-051','杨贯中','教授','500000'),
('2003-011','边耐政','副教授','500000'),
```

```
('2002-127','陈浩','副教授','590000');
```

```
#loading course
```

```
INSERT INTO course(cno,cname,hours,term,credit,dno)
```

```
VALUES
```

```
('H61030006','数据结构','64','4','4','590000'),
('H61030007','计算器组成原理','32','5','2','590000'),
('H61030008','数据库系统','48','5','3','590000'),
('H61030009','操作系统','48','4','3','590000'),
('H61030012','软件工程','48','6','3','590000'),
('H61030013','计算机网络','64','7','4','590000'),
('T53030001','毛泽东思想概论','32','1','2','500000'),
('T53030002','邓小平理论','32','2','2','500000'),
('X61030005','面向对象程序设计言','32','2','2','590000'),
('X61030012','Web程序设计','32','6','2','590000'),
('X61030019','数据挖掘','32','8','2','590000'),
('X61030022','网络安全概论','32','7','2','590000'),
('X80002002','软件产品市场营销','16','7','1','590000'),
('X80002003','知识管理','16','3','1','500000');
```

```
#loading teacher_course
```

```
INSERT INTO teacher_course(tno,cno,semester)
```

```
VALUES
```

```
('2004-213','H61030008','2010/01'),
('2002-048','H61030008','2010/01'),
('2001-021','H61030008','2010/01'),
('2006-117','X61030005','2010/01'),
('2005-187','X61030005','2010/01'),
('2000-093','H61030013','2009/01'),
('2001-051','H61030013','2009/01'),
('2003-011','X61030012','2009/01'),
('2002-127','X61030012','2009/01'),
('2004-213','X61030019','2010/02'),
('2002-048','X61030019','2010/02');
```

```
#loading student_course_teacher
```

```
INSERT INTO student_course_teacher(sno,tno,cno,semester,score)
```

```
VALUES
```

```
('20080001','2004-213','H61030008','2010/01','92'),
('20080001','2006-117','X61030005','2010/01','88'),
('20080001','2004-213','H61030008','2009/02','78'),
('20080002','2004-213','H61030008','2010/01','90'),
('20080002','2006-117','X61030005','2010/01','85'),
('20080002','2004-213','H61030008','2009/02','82'),
```

```
('20080003','2004-213','H61030008','2010/01','90'),
('20080003','2006-117','X61030005','2010/01','83'),
('20080003','2004-213','H61030008','2009/02','89'),
('20080005','2004-213','H61030008','2010/01','95'),
('20080005','2006-117','X61030005','2010/01','86'),
('20080005','2004-213','H61030008','2009/02','91'),
('20080006','2004-213','H61030008','2010/01','83'),
('20080006','2006-117','X61030005','2010/01','86'),
('20080006','2004-213','H61030008','2009/02','75'),
('20080012','2004-213','H61030008','2010/01','88'),
('20080012','2006-117','X61030005','2010/01','90'),
('20080012','2004-213','H61030008','2009/02','78'),
('20080011','2002-048','H61030008','2010/01','87'),
('20080011','2006-117','X61030005','2010/01','62'),
('20080007','2002-048','H61030008','2010/01','93'),
('20080007','2006-117','X61030005','2010/01','88'),
('20080007','2004-213','H61030008','2009/02','87'),
('20080008','2002-048','H61030008','2010/01','77'),
('20080008','2006-117','X61030005','2010/01','74'),
('20080008','2004-213','H61030008','2009/02','89'),
('20080009','2002-048','H61030008','2010/01','90'),
('20080009','2006-117','X61030005','2010/01','81'),
('20080009','2004-213','H61030008','2009/02','84'),
('20080010','2002-048','H61030008','2010/01','73'),
('20080010','2006-117','X61030005','2010/01','71'),
('20080010','2004-213','H61030008','2009/02','50');
```

### 实验三：完整性约束、视图、存储过程和数据安全

## [目的和意义]

掌握创建各种完整性约束的方法;掌握创建和使用视图、存储过程和索引的方法;掌握实现数据安全的基本方法;掌握数据库的基本备份和恢复技术。

## [实验内容]

本实验在 mysql 数据库平台上进行操作，实验中的各种数据都来自于实验二中的各个表。

实验主要内容如下：

1. 学习创建完整性约束的方法。
2. 学习创建视图、存储过程和索引的方法。
3. 学习利用用户管理和视图实现数据安全。
4. 学习 mysql 数据库的基本备份和恢复技术。

## [实验步骤]

### 1 完整性约束

#### 1.1 实体完整性约束

主键一定是非空的，而且具有唯一性，即不能重复。能够唯一标识实体，保证实体完整性。虽然，上面已经成功创建了主键，但是，我们还是要学习一下主键的写法。

第一种：在建表的时候就加上主键：

还是以上面 course 表的创建为例：

```
CREATE TABLE course (
cno VARCHAR(10) NOTNULL,
dno CHAR(8) NOTNULL,
cname VARCHAR(20) NOTNULL,
hours INTEGER NOTNULL,
term INTEGER NOTNULL,
credit INTEGER NOTNULL,
CONSTRAINT pk_course PRIMARY KEY(cno)
);
```

其中最后一句话就是将 cno 设为主键。如果主键包含多个字段，可以在括号内添加所包含字段名，如取 cno 和 dno 为主键，则语句为：CONSTRAINT pk\_course

PRIMARYKEY(cno,dno)。

第二种：若建表的时候没有将 cno 设为主键，表建立成功以后，可以加上主键约束：

```
ALTER TABLE course ADD CONSTRAINT pk_course PRIMARY
KEY(cno) ;
```

## 1.2 参照完整性约束

建立外键：例如，在 course 表中建立和 department 表之间的外键，语句如下：

```
ALTER TABLE course
ADD CONSTRAINT fk_course_department FOREIGN KEY(dno)
REFERENCES department(dno);
```

在 student 表中建立和 department 表之间的外键，语句如下：

```
ALTER TABLE student
ADD CONSTRAINT fk_student_department FOREIGN KEY(dno)
REFERENCES department(dno);
```

## 1.3 业务规则约束

有时候，不同系统会有一些特定的业务规则；

例如这里，要求在删除 teacher 表之前一定要先删除 teacher\_course 与 student\_course\_teacher 中的相关信息，这里，我们用触发器实现一下：

```
DELIMITER//      #修改分隔符
CREATE TRIGGER tri_del_teacher
AFTER DELETE ON teacher
FOR EACH ROW
BEGIN
DELETE FROM teacher_course
WHERE tno=old.tno;
DELETE FROM student_course_teacher
WHERE tno=old.tno;
END//
DELIMITER;      #修改回原分隔符
```

建立成功以后，如果我们在 teacher 表中删除老师编号为 '2004-213' 的老师的

信息，那么teacher\_course和student\_course\_teacher两表中包含此老师信息的相关数据都会删除掉（慎用！）。

```
delete from teacher where tno='2004-213';
```

**备注：**

1. mysql中在编写函数、存储教程和触发器时，要进行语句分隔符转变，mysql的默认分隔符为“;”，上面语句中我们将分隔符修改为“//”，在语句完毕后，我们再将分隔符还原成“;”。

2. mysql中不支持 check 约束，需要用触发器来完成类似于 check 约束的功能。

## 2 视图、存储过程与索引

### 2.1 视图的创建和查询

我们以实验二中的例题1为例：

1、在 2010/01 学期选了杨金民老师开设的“数据库系统”课程的学生姓名、学号、班级清单；

```
SELECT s.sno 学号, sname 姓名, classno 班级
FROM student s,
     teacher t,
     course c,
     student_course_teacher sct
WHERE t.tname = '杨金民'
AND c.cname = '数据库系统'
AND sct.semester = '2010/01'
AND s.sno = sct.sno
AND t.tno = sct.tno
AND c.cno = sct.cno;
```

将它作为视图：

```
CREATE VIEW myview_course AS SELECT s.sno 学号, sname 姓名, classno
班级
FROM student s,
     teacher t,
     course c,
```

```

        student_course_teacher sct
WHERE t.tname = '杨金民'
AND c.cname = '数据库系统'
AND sct.semester = '2010/01'
AND s.sno = sct.sno
AND t.tno = sct.tno
AND c.cno = sct.cno;

```

然后 `SELECT * FROM myview_course;`

其它语句大家可以自己实验。

### 备注

1. 创建视图语句: `create view view_name as select_ clause ;`
2. 删除视图语句: `drop view view_name;`

## 2.2 存储过程

创建存储过程：

以上次实验中的第二句为例：

2010/01 学期“数据库系统”课程的班级平均分清单；

```

select classno 班级, avg(score) 平均分
from student s, course c, student_course_teacher sct
where c.cname = '数据库系统'
and sct.semester = '2010/01'
and s.sno = sct.sno
and c.cno = sct.cno
group by classno;

```

创建存储过程：

```

delimiter //
create procedure myproc(
in cname1 varchar(20), in semester1 varchar(7))
begin
        # in 为 mysql 存储过程中的参数类型
select classno 班级, avg(score) 平均分
from student s, course c, student_course_teacher sct
where c.cname = cname1
and sct.semester = semester1
and s.sno = sct.sno
and c.cno = sct.cno

```

```
group by classno;
end //
delimiter ;
```

创建成功以后，使用存储过程:

```
CALL myproc('数据库系统','2010/01');
```

就可以实验上述语句的效果，但这里只显示出了一项数据。

**备注：**

1. 调用存储过程语句：`call procedure_name( );`
2. 删除存储过程语句：`drop procedure procedure_name;`
3. mysql支持 IN（传递给存储过程，如上面使用）、OUT（从存储过程传出）和 INOUT（对存储过程传入和传出）类型的参数。

## 2.3 索引

当表中数据量很大时，索引能够提高查询速度，改善处理性能。

索引的创建方法，例如对 STUDENT 表中的 SEX 建立索引：

```
CREATE INDEX my_student_idx ON student(sex);
```

## 3 数据安全

### 3.1 用户管理

mysql中用户管理包括创建、修改用户，用户的授权和权限的收回。

在前面的实验中，我们都是使用 root 用户进行登陆操作。root 是一个超级用户，它对整个 mysql 服务器具有完全的控制，如果只是利用 mysql 进行实验操作，这样很好;但如果在现实世界的日常工作过程中，决不能使用 root，应该创建一系列的账号，有的供用户使用，有的供开发人员使用，等等。

mysql中的用户账号和信息存储在名为 mysql 的数据库中，可以使用 `use mysql; select user from user;` 来查看所有的用户账号。

创建用户 test,密码为 tiger,命令为:

```
CREATE USER test IDENTIFIED BY 'tiger';
```

此时，用户 test 可以登陆，且只能查看 information\_schema 数据库，用户要进行其它操作必须先进行授权。

授权语句格式为：

```
GRANT {privilegeList | ALL PRIVILEGES}
ON ObjectName
TO {AuthorizationIdList | PUBLIC}
[WITH GRANT OPTION]
```

例如:我们让 test用户能够查看前面实验中在用户 mytest 中创建的表 teacher\_course,并且能够修改其中的 semester 字段,则命令为：

```
GRANT select,update(semester) ON mytest.teacher_course TO test;
```

然后登入 test 用户，SELECT \* FROM mytest.teacher\_course;查看一下结果;

现在，我们修改一下semester字段，这里，我们将 cno='X61030005'的数据中的 semester 由'2001/01'修改为'2008/01';命令为：

```
UPDATE mytest.teacher_course SET semester='2008/01' WHERE
cno='X61030005';
```

这里，可以试下修改 tno,会出现权限不足的提示;

然后，我们将修改 semester 字段的权限收回，回到 root 用户。

输入下面语句：

```
REVOKE UPDATE ON mytest.teacher_course FROM test;
```

这时，再连接到test用户下，输入上面修改 semester 字段的语句，会提示权限不足。

删除用户的命令为 DROP USER username;慎用！

## 3.2 视图

视图也能增强数据安全性：

例如：这里我们想让用户 test 不能查看学生的详细信息，只对 test 显示学生的学号，姓名，方法如下：

首先，建立视图:

```
CREATE VIEW myview_student AS SELECT sno,sname FROM mytest.student;
```

然后，把查看此视图的权限授予 test:

```
GRANT select ON myview_student TO test;
```

现在，test 就只能查看 sno,sname 此两项信息。

#### 4 数据备份与恢复

本章主要学习使用命令进行数据库的逻辑备份与恢复：

##### 4.1 数据备份

备份命令格式：

```
MYSQLDUMP -u用户名 -p 数据库名 [表名] > 目标文件名。
```

举例说明：我们现在将 student\_course\_teacher 表进行备份和恢复。

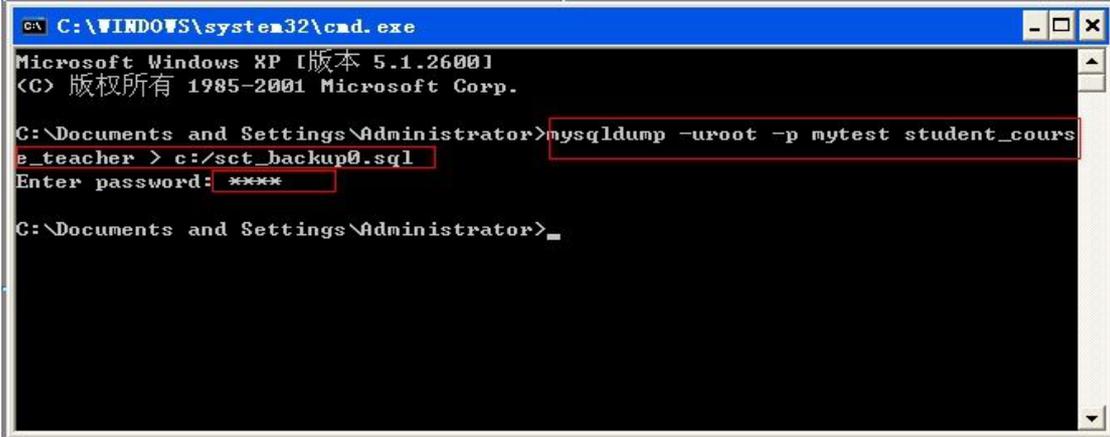
首先，运行“cmd”，进入 DOS 窗口界面，输入如下命令：

```
MYSQLDUMP -uroot -pmytest student_course_teacher >c:/sct_backup.sql
```

回车运行，提示输入密码，输入完成后即完成了对 student\_course\_teacher 表的备份操作。

(如果报错则用如下语句：`mysqldump -uroot -p #default-character-set=utf8 #max_allowed_packet=48M mytest student_course_teacher >c:/sct_backup.sql`)

图示如下：



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>mysqldump -uroot -p mytest student_course_teacher > c:/sct_backup.sql
Enter password: ****

C:\Documents and Settings\Administrator>_
  
```

操作成功后可以在 C 盘下查看 sct\_backup.sql 文件。

## 4.2 数据恢复

接着下来，我们试验一下恢复表 student\_course\_teacher:

命令是我们已经使用过的 source 命令。首先登陆 mysql 数据库系统，而后导入文件：

```
SOURCE c:/sct_backup.sql
```

## 实验四：使用 PowerDesigner 设计数据库

### [目的和意义]

掌握将现实世界的事物转换成E-R图的基本技术，能够正确判断给定表的关键词;了解将E-R图转换成关系表的技术;掌握使用PowerDesigner工具创建CDM概念数据模型的方法，以及完成数据库表的创建和修改等工作。

### [实验内容]

本实验的主要内容：

使用 PowerDesigner 按下列要求创建用户数据库模型：

5. PowerDesigner 软件的安装
6. 数据库需求分析
7. 设计数据库并创建CMD概念数据模型

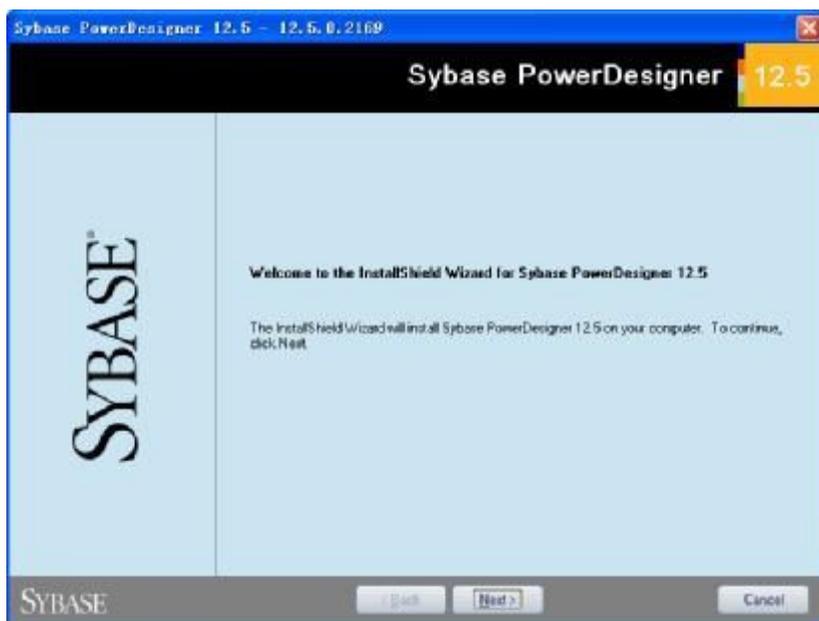
### [实验步骤]

## 1 安装与破解 PowerDesigner 软件

目前主要的建模工具厂商有 Sybase PowerDesigner, IBM Rational Rose, Computer Associates ERWin 等。PowerDesigner 是 Sybase 公司的 CASE 工具集, 使用它可以方便地对信息系统进行分析设计, 其中主要受关注的有数据设计模型、对象模型、和业务流程模型等, 它几乎包括了数据库模型设计的全过程。目前的最新版本是 PowerDesigner 15, 我们实验安装的版本为 PowerDesigner v12.5。

### 1.1 安装 PowerDesigner 软件

在给定的安装文件夹 Sybase.PowerDesigner.v12.5-EDGE 下找到安装文件 powerdesigner125\_eval.exe, 点击直接安装。



点击“Next”继续, 接受默认设置, 直至安装完成。

### 1.2 破解 PowerDesigner 软件

安装文件夹子目录中的两文件 pdf1m12.dll 和 license.lic 用于破解。

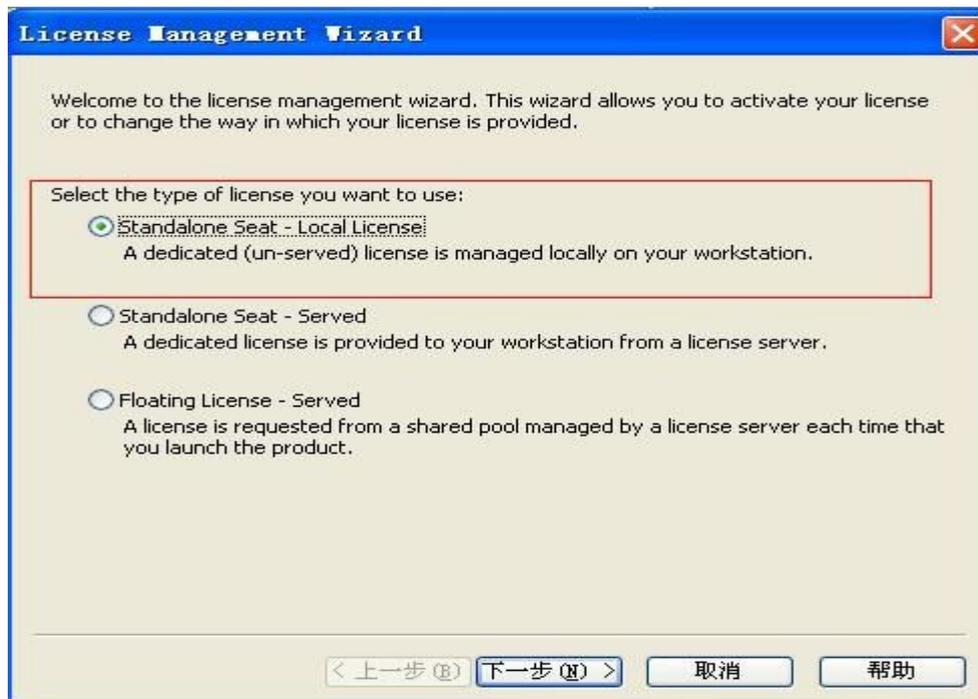


安装完成后，在 安装文 件下找到 pdf1m12.dll 文 件，用它去替换 PowerDesigner 安装目录下的 pdf1m12.dll 同名文 件。然后在 “开 始” ->“所在程序”->“Sybase”->PowerDesigner12 中点击“PowerDesigner”打开软件。

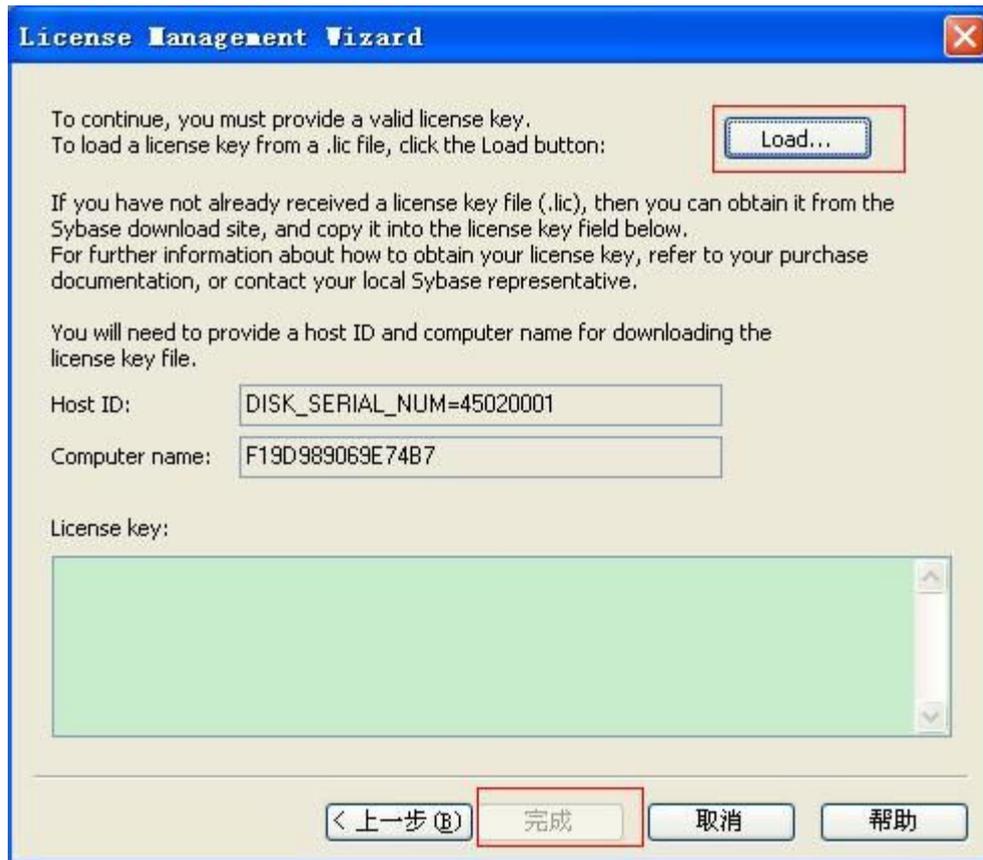
打开软件后，在“Tool”菜单下点击“License Parameters”;



进入“License Management Wizard”对话框;



选择第一个单选框，然后“下一步”

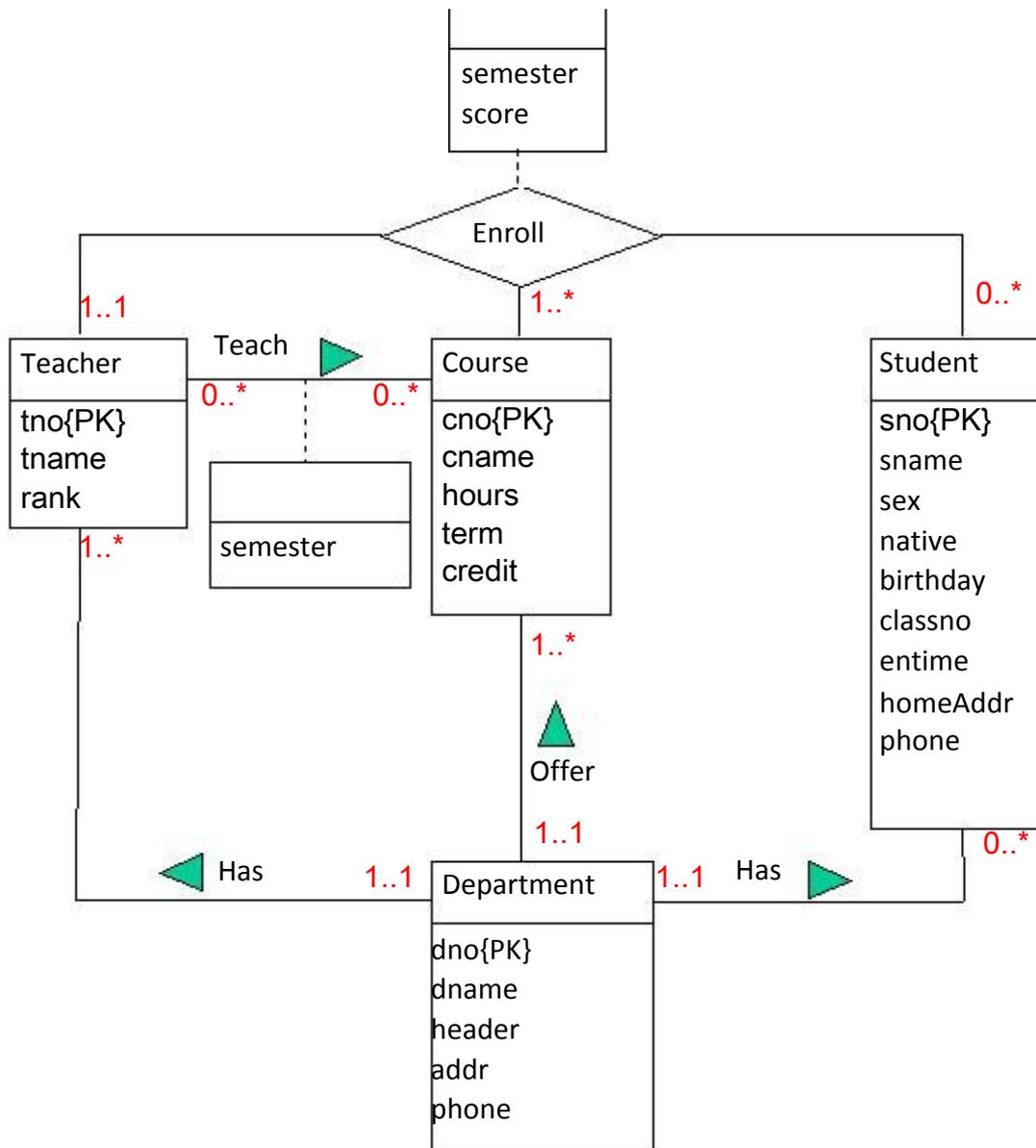


点击“Load”在安装目录找到 license.lic 文件并导入，点击“完成”进入下一页，选择“Location”为 HongKong，同意协议，完成破解。

## 2 数据库需求分析

本次实验是基于实验二中的学校信息系统数据库进行建模操作，该数据库有6个数据表，其中4个实体表和2个关系表，实体表为：学院表（department）、学生表（student）、老师表（teacher）、课程表（course）；关系表为：老师开课表（teacher\_course），学生选课表（student\_teacher\_course）

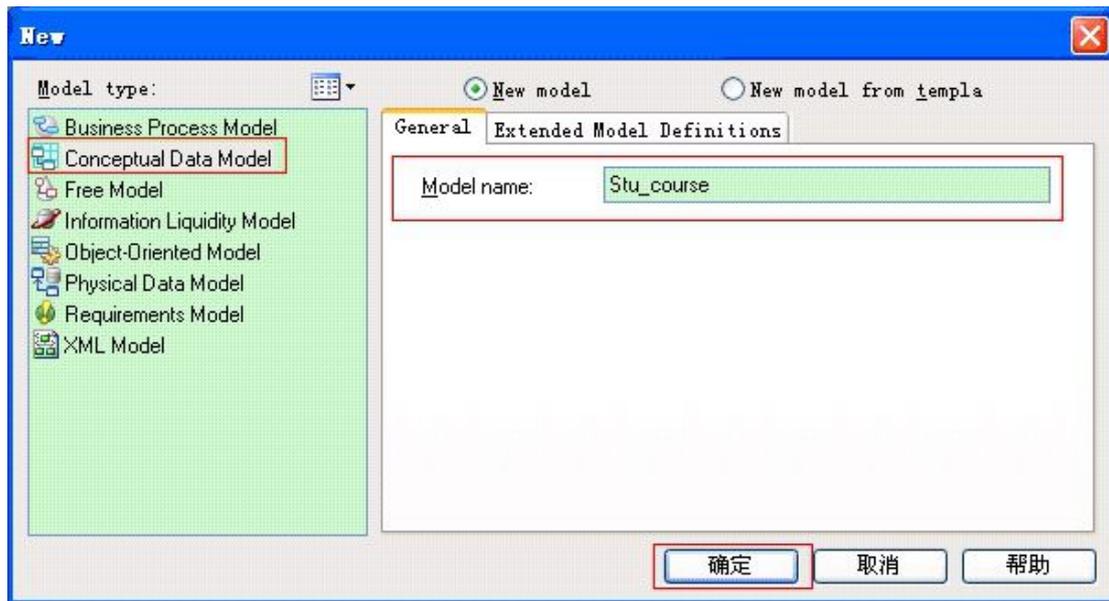
通过分析数据表单及业务功能，可得出初步模型图：



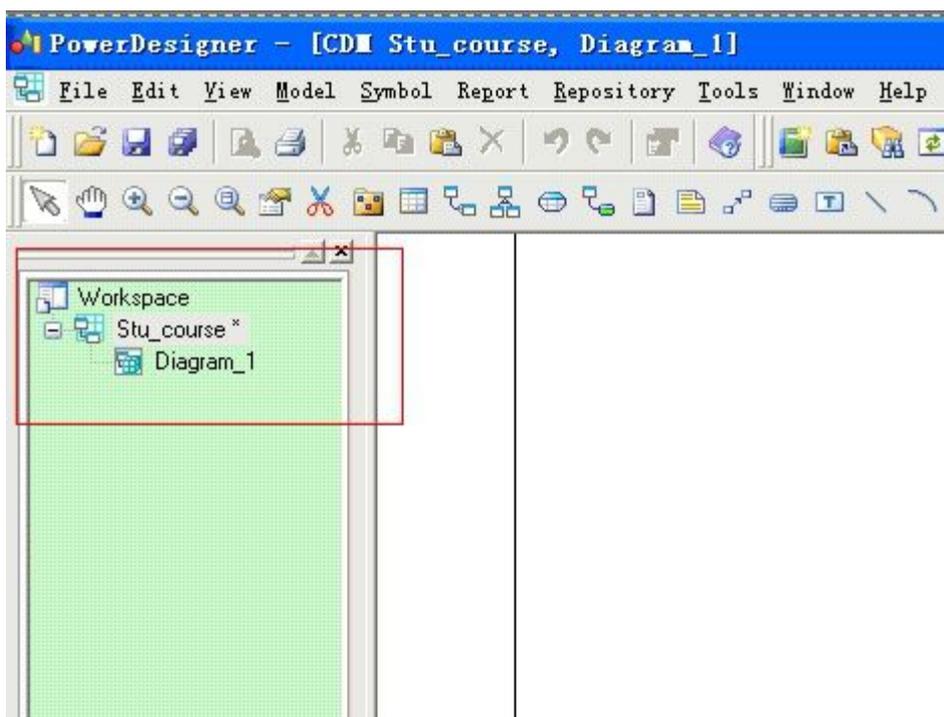
### 3 创建 CDM 概念数据模型

#### 3.1 创建 CDM 文件

打开 PowerDesigner 工具，选择“文件”中的“NEW”子菜单，选 CDM 数据库模型，即“Conceptual Data Model”，如下图：

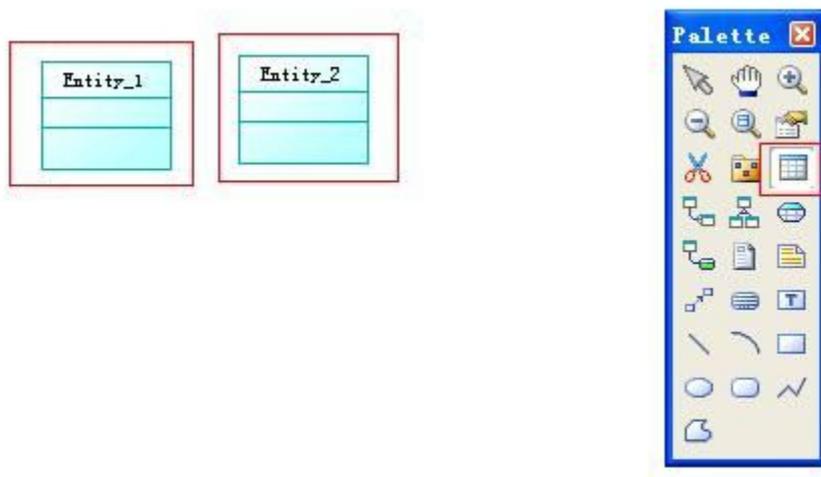


在“Model name”输出入要建立的 cdm 模型的名字“Stu\_course”，按确定按钮，系统出现如下的屏幕：

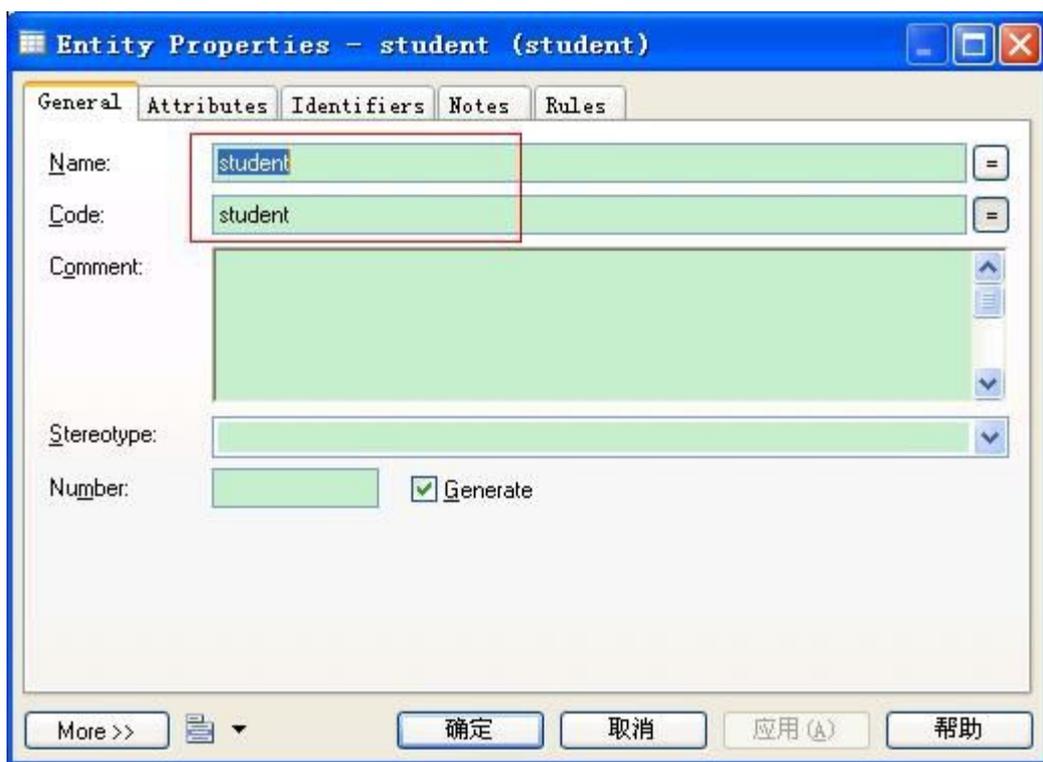


### 3.2 添加实体对象和联系对象

在工具栏或 Palette 面板中选取 Entity 图标,然后在画布中放置，如下图：

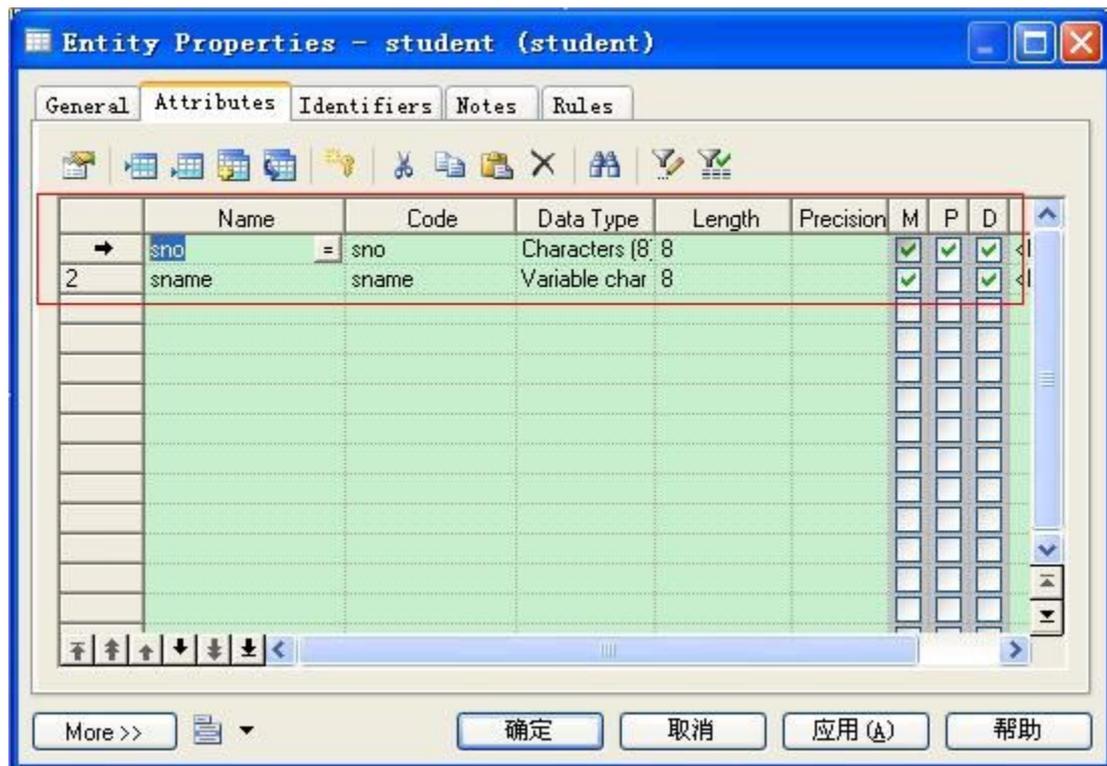


双击一个 Entity，在弹出“Entity Properties”中输入 Name、Code。其中 Name 是在显示时使用的名称，而 Code 是在代码中使用的名称。



点击“Attributes”，建立相应的属性，如下图：

( Name : 显示名称，Code : 代码名称，Data Type: 数据类型，Length: 数据长度，Precision : 数据精度，M : 必须的，P : 主键，D : 显示 )

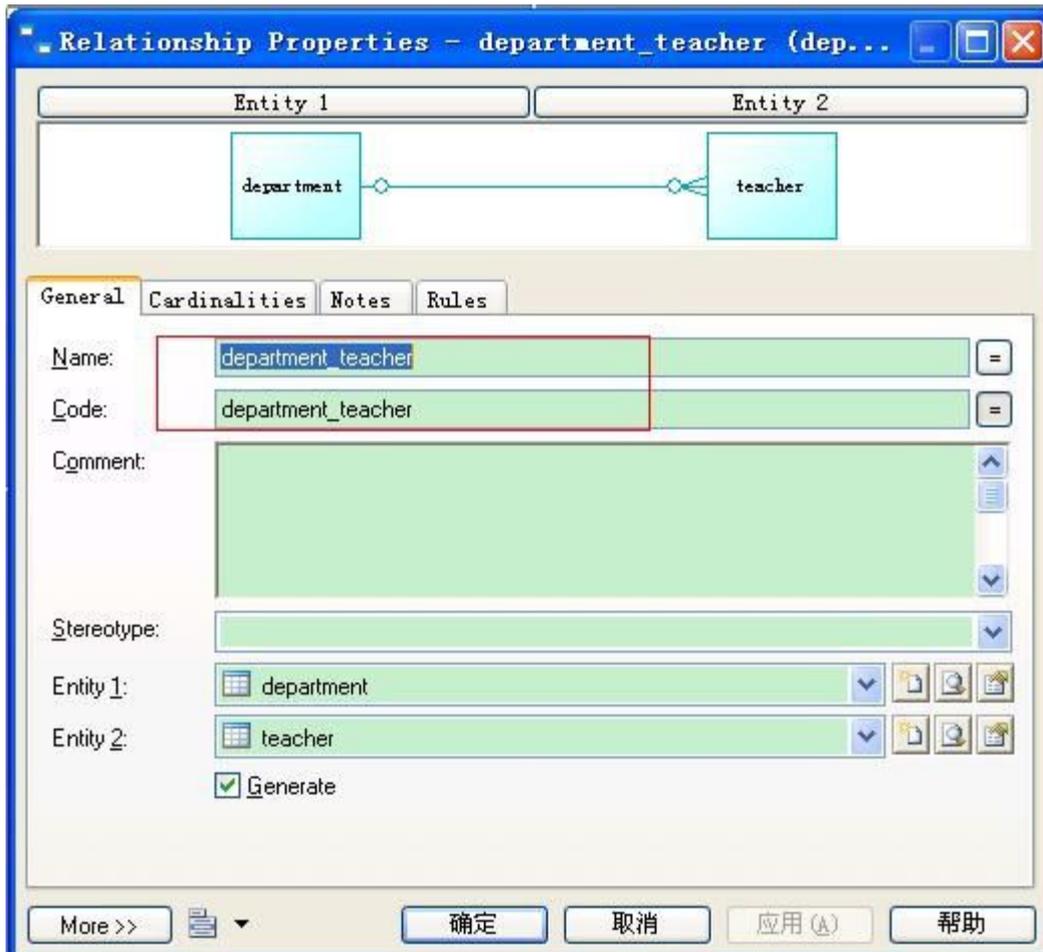


使用上面的方法建立学院（department）、学生（student）、老师（teacher）、课程（course）4 个 Entity 对象，及老师开课（teacher\_course），学生选课（student\_teacher\_course）两个association联系对象。

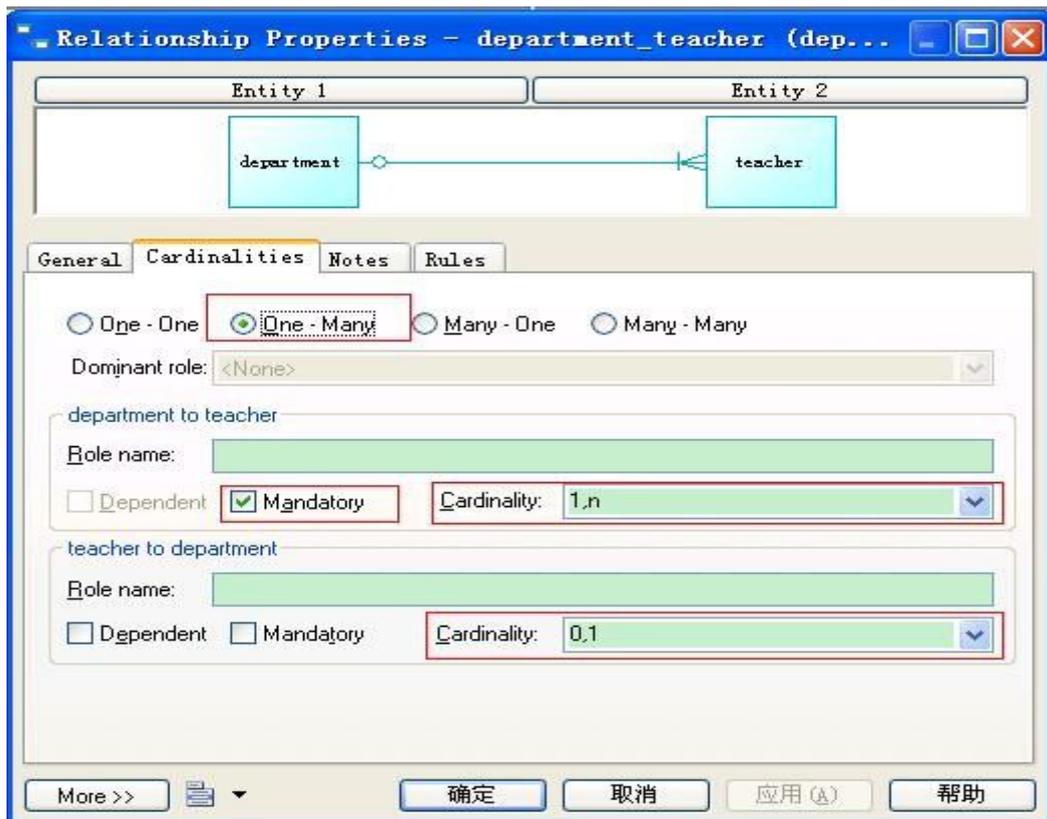
### 3.3 确定对象之间的关系

点击工具栏中的关系工具 Relationship，在学院Entity上点击并拖至教师Entity。

双击刚刚建立的 Relationship 连线.修改 Name 及 Code，如下图：



在“Cardinalities”中选择相应的 Cardinality，对关系进行相关设置，如下图：

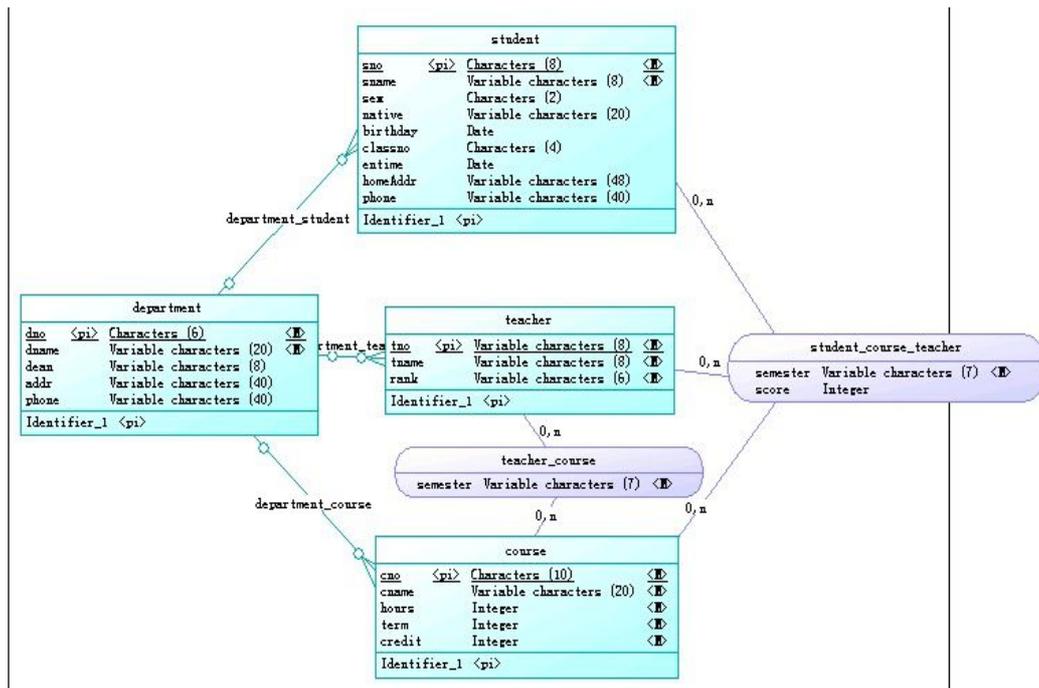


然后就可以查看到 department 和 teacher 之间的关系 department\_teacher，  
如下图：



### 3.4 完成 CDM 图

最终画出的 CDM 图应该类似于下图：



概念数据模型 ( CDM ) 图

## 附件

## PowerDesigner 技巧

- PowerDesigner 12和Office Word 2003有冲突，会导致Word 2003鼠标在文文件中无效，并且不能打开两个文档的问题。

将\$:\Program Files\Sybase\PowerDesigner 12\Add-ins\目录下的 Microsoft Word Import Export 目录删除

## 实验五：使用PowerDesigner创建Mysql数据库

### [目的和意义]

掌握 CDM 概念模型转换成 PDM 物理数据模型方法;掌握 PDM 物理数据模型生成数据库脚本文件的基本操作;了解 MySQL 的 ODBC 驱动程序的作用和安装方法;掌握在 PowerDesigner 中配置数据源和建立数据库的基本方法。

### [实验内容]

本实验的主要内容：

使用 PowerDesigner 按下列要求创建用户数据库：

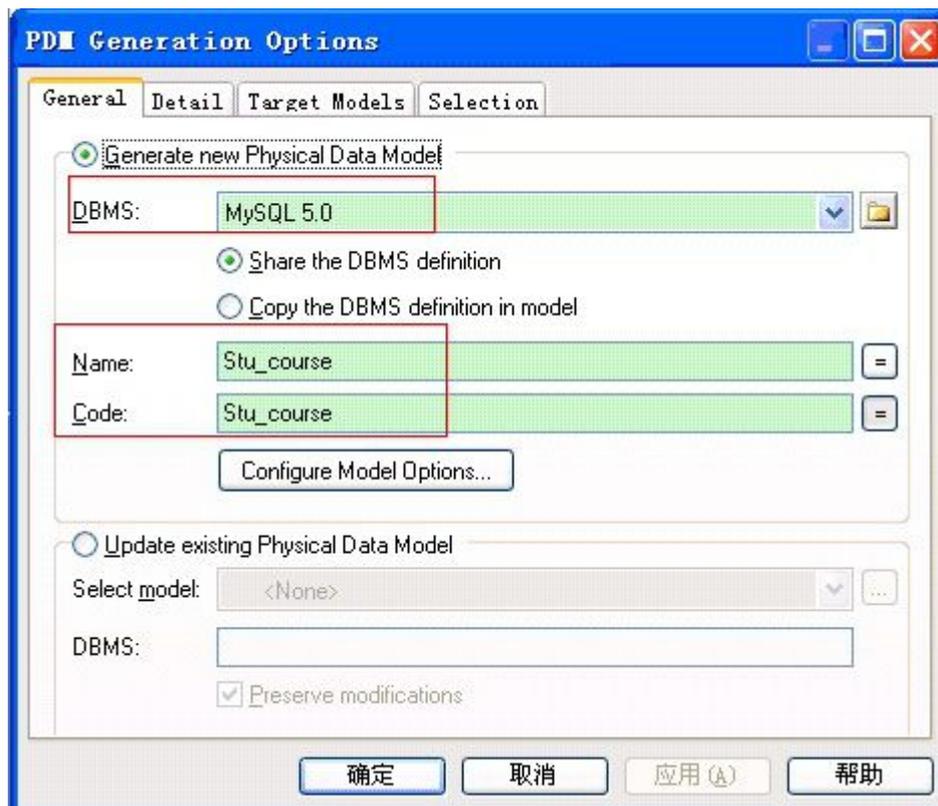
8. 将 CDM 概念模型转换成PDM 物理数据模型;
9. 基于 PDM 物理数据模型生成 MySQL数据库脚本文件;
10. 安装 MySQL 的 ODBC 驱动并配置MySQL数据源;
11. 在 PowerDesigner中建立数据库。

### [实验步骤]

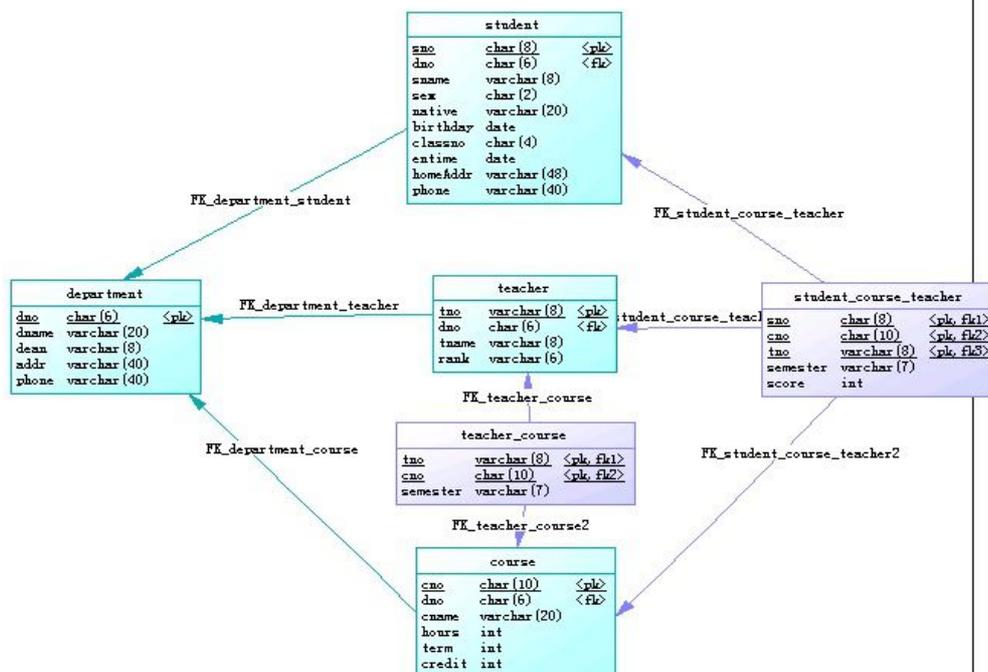
#### 1 生成 PDM 图

在 powerdesigner 菜单栏中选择“Tools”菜单，并单击“Generate Physical Data Model”子菜单。

系统弹出“PDM Generation Options”对话框，在“General”标签下的“DBMS”选项中选择“M y SQL5.0”，Name、Code可为默认值“Stu\_course”，如下图：

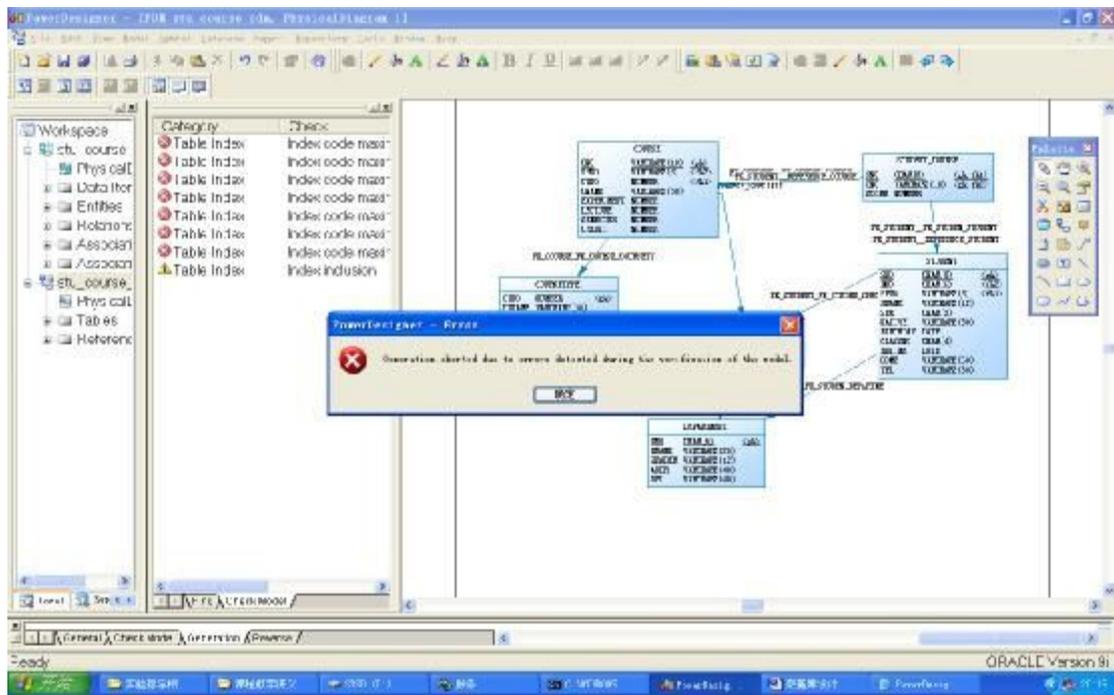


填好后，点击“确定”，生成 PDM 物理数据模型，转化成的 PDM 图，注意其中外键自动生成了。如下：



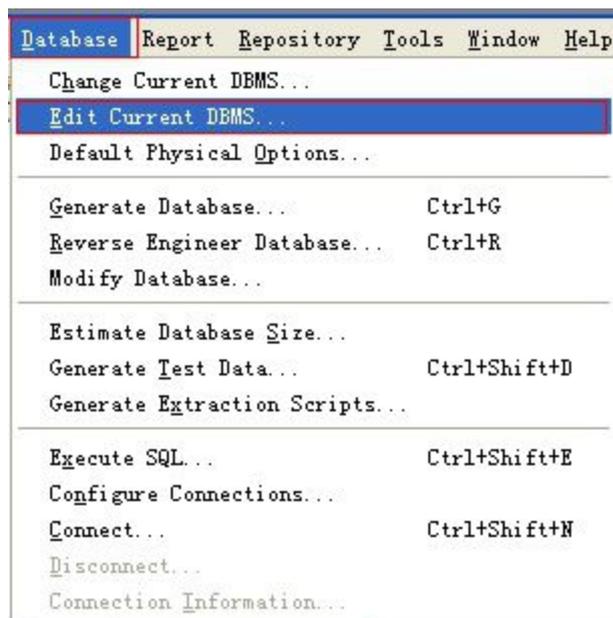
物理数据模型 ( PDM ) 图

在转换的过程中有可能出现下面的错误信息：

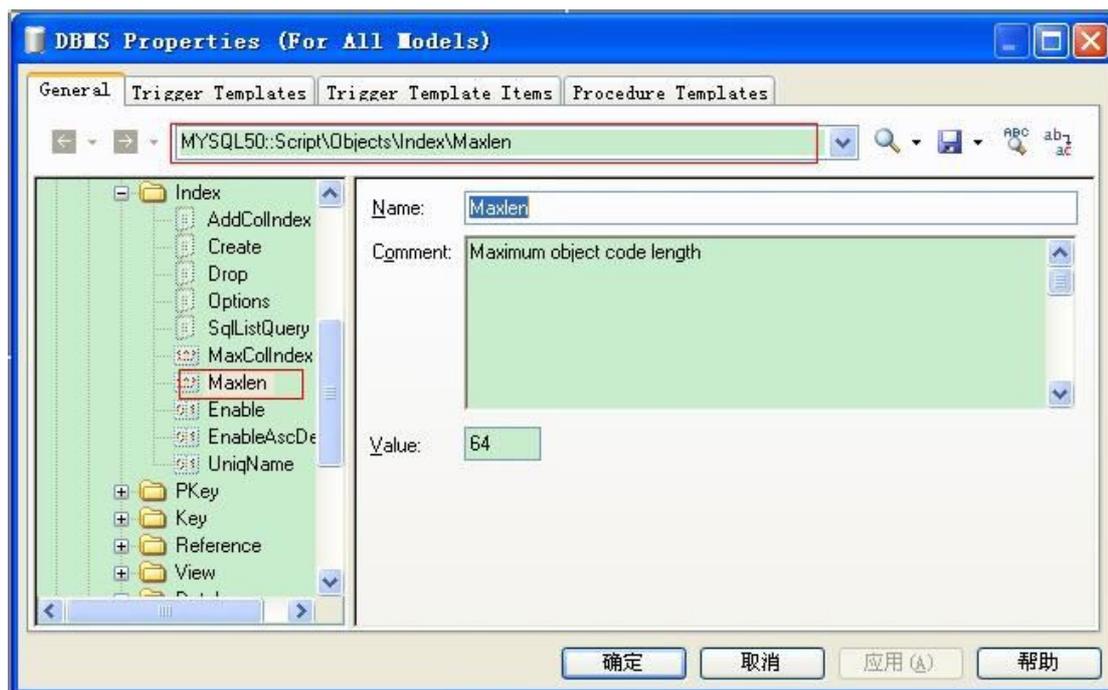


如果是这样的话，需要设置 PowerDesigner 中的 model 参数：修改 Index 等对象的 MaxLen 为一个较大的值。具体操作如下：

打开 PDM 模型图，选择“Database”菜单下的“Edit Current DBMS”；

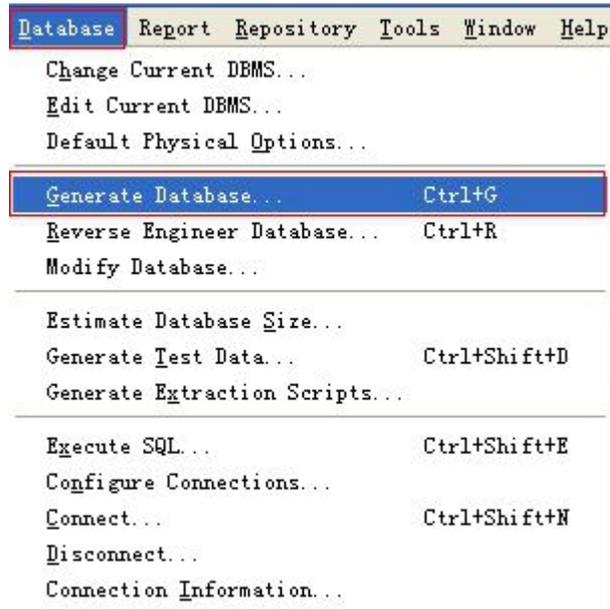


弹出“DBMS Properties (For All Models)”对话框。在 Script\Objects\ 下，修改 Index 等对象的 MaxLen 为一个较大的值。

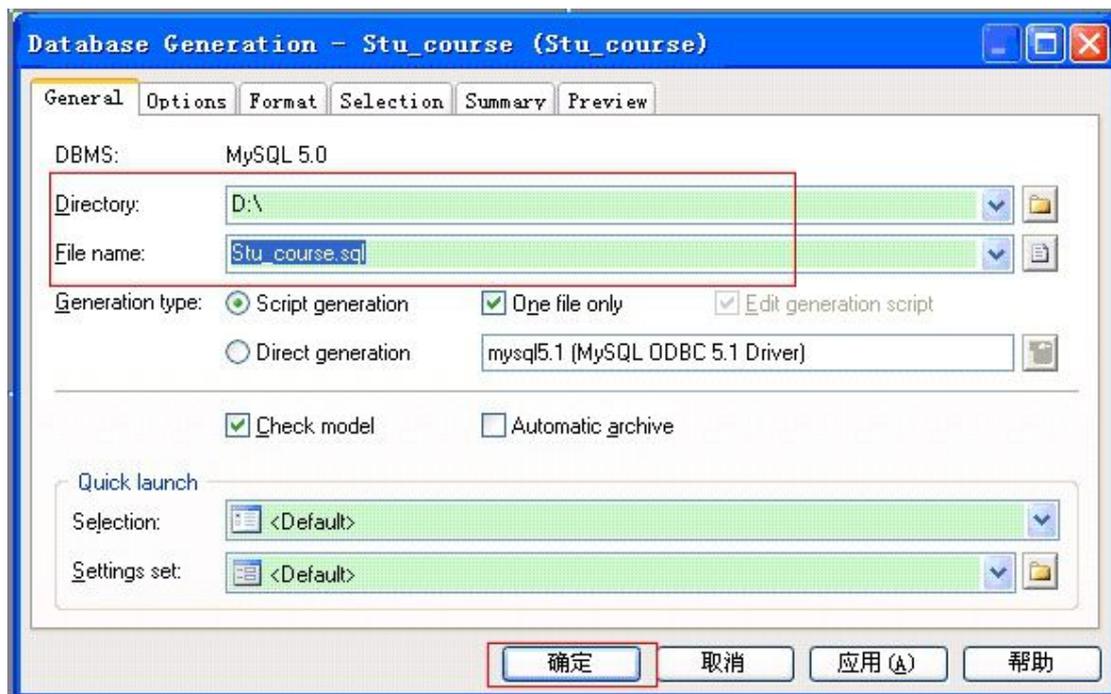


## 2 生成数据库脚本文件

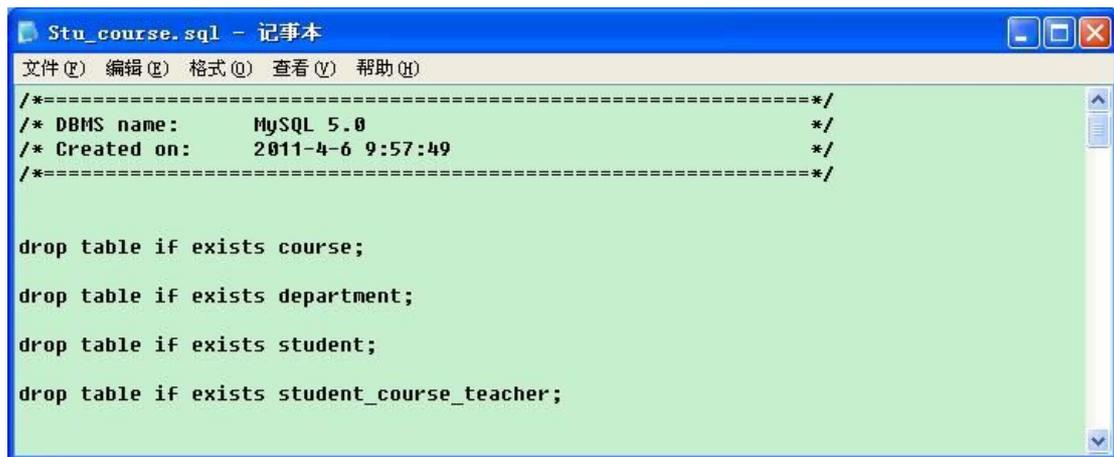
用 powerdesigner 打开 PDM 图后，在菜单栏中选择“Database”菜单，单击“Generate Database”子菜单；



弹出“Database Generation”对话框;



填好保存路径，和文件名称，点击“确定”，就生成了脚本文件 **Stu\_course.sql**;



```
Stu_course.sql - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

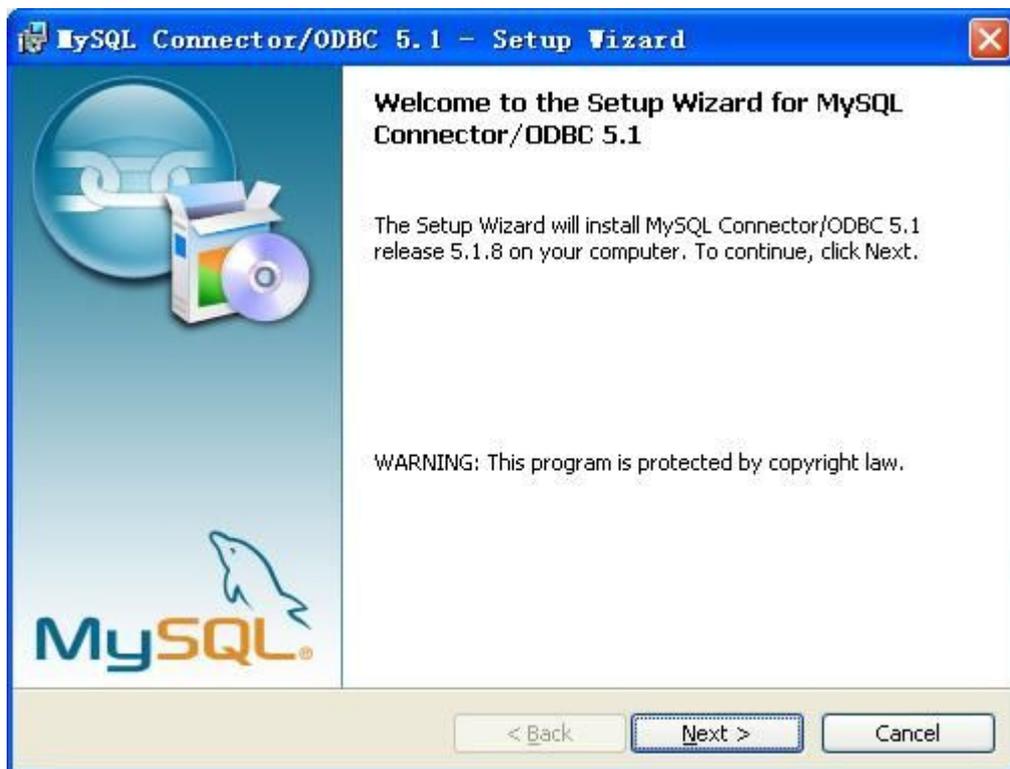
/*=====*/
/* DBMS name:      MySQL 5.0          */
/* Created on:     2011-4-6 9:57:49   */
/*=====*/

drop table if exists course;
drop table if exists department;
drop table if exists student;
drop table if exists student_course_teacher;
```

### 3 配置 MySQL 数据源

#### 3.1 安装 MySQL 的 ODBC 驱动

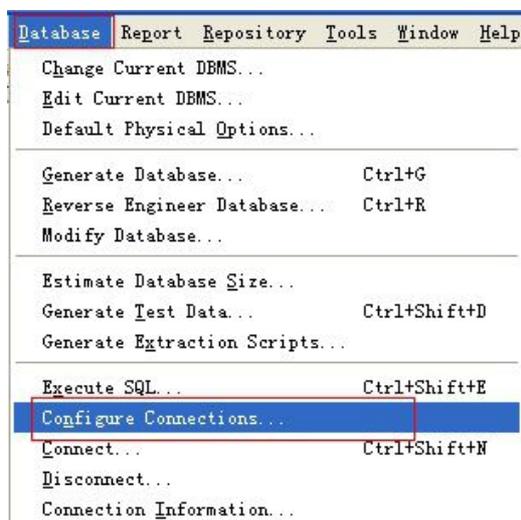
在 PowerDesigner 配置数据库连接之前，先要安装 MySQL 的 ODBC 连接驱动程序 mysql-connector-odbc-5.1.8-win32.msi，点击安装。



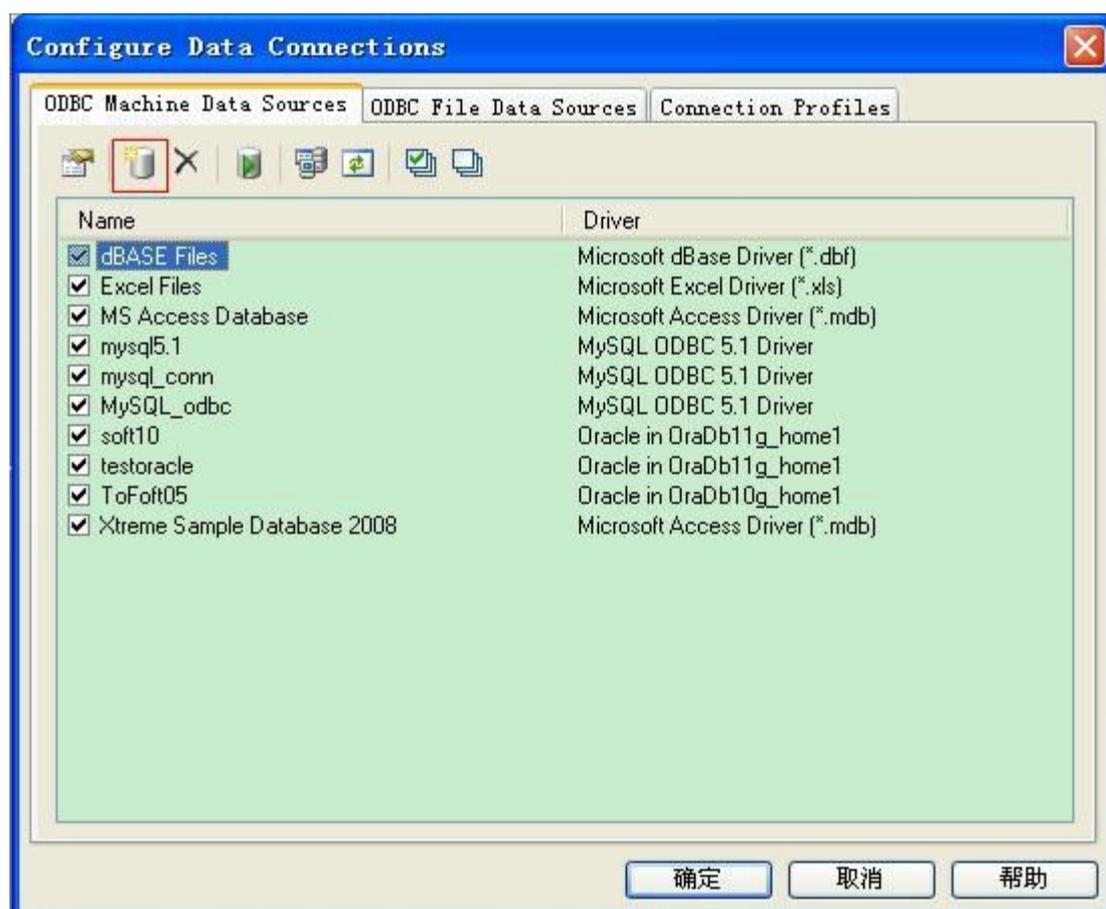
按“Next”继续，接受默认设置，直至安装完成。

### 3.2 在 PowerDesigner 中配置数据源

在 PowerDesigner 的“DataBase”菜单下点击“Configure Connections”;



进入“Configure Data Connections”对话框;



点击添加数据源图，打开“创建新数据源”对话框;



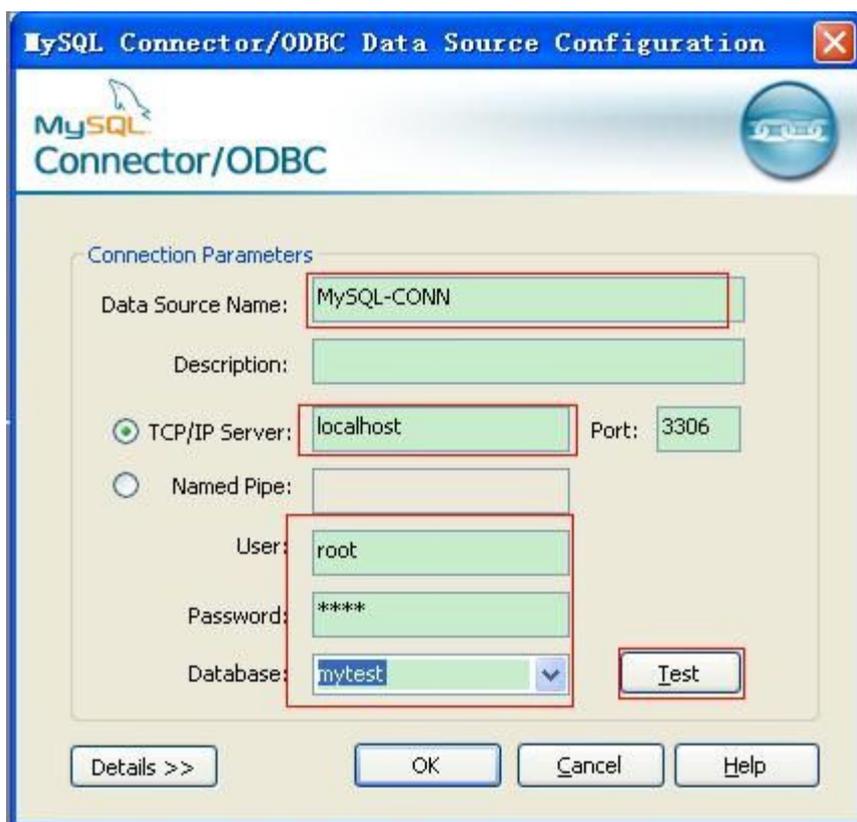
在“选择数据源类型”中选择“系统数据源”，点“下一步”；



选择“MySQL ODBC 5.1 Driver”数据源驱动程序，点击“下一步”；



点击“完成”，进入数据源配置界面；



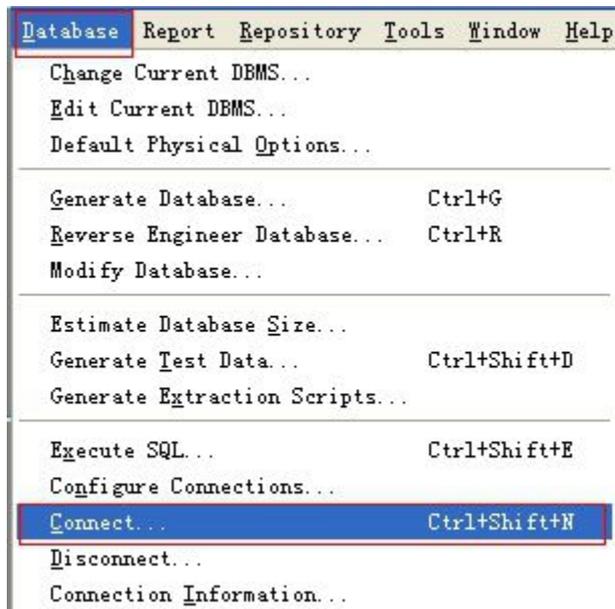
在连接列表中输入相应的 MySQL 参数;点击“Test”测试数据源的连接，连接成功，提示“Connection successful”。



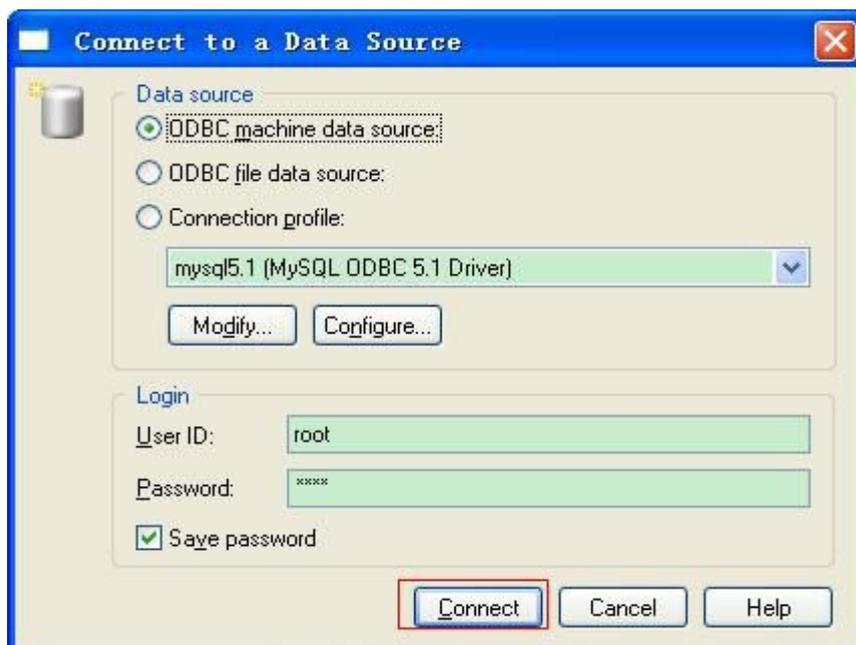
4 在 PowerDesigner 中建立数据库

#### 4.1 连接 MySQL 数据库

在 PowerDesigner 的进入“DataBase”菜单，点击“Connect”；

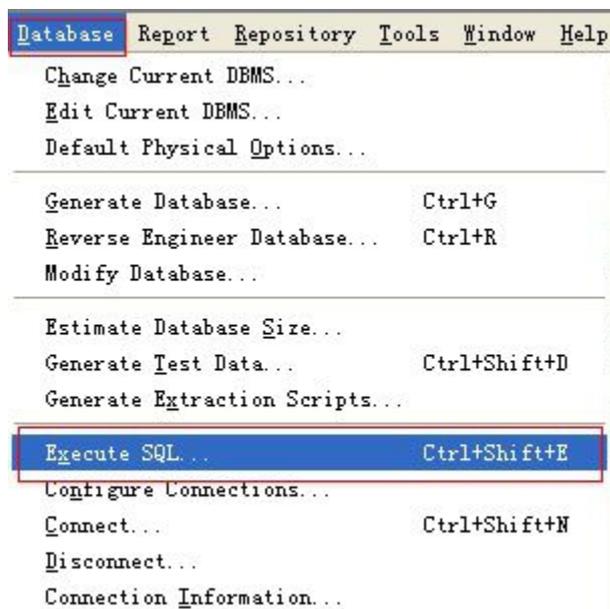


弹出“Connect to Data Source”对话框，接受默认配置，点击“Connect”

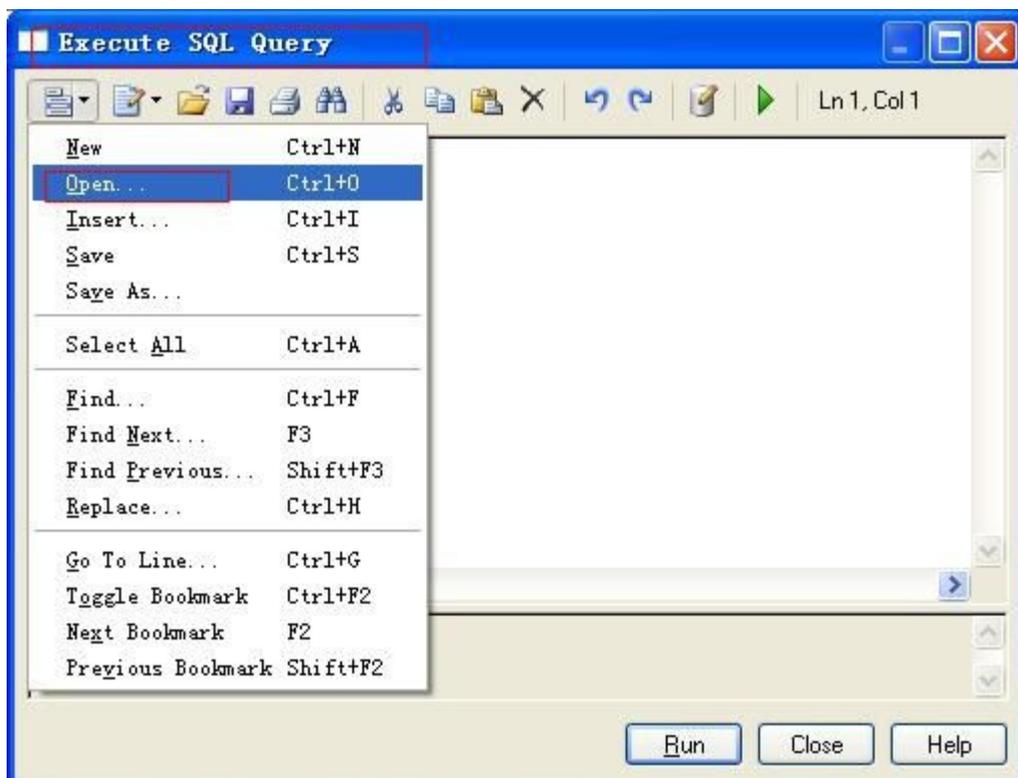


## 4.2 执行 SQL 脚本

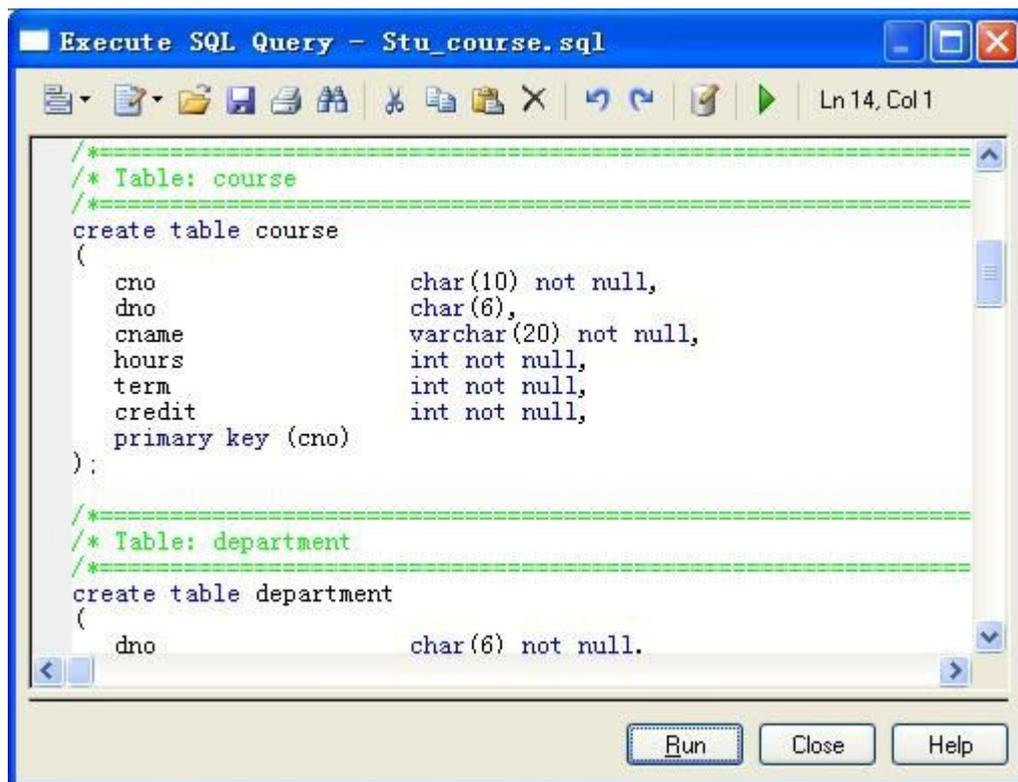
在 PowerDesigner 的进入“DataBase”菜单，点击“Execute SQL”；



弹出“ExecuteSQLQuery”对话框；



通过“Open”打开并导入第 2 节中生成的数据库脚本文件“Stu\_course.sql”；



在上面的窗口中按“Run”按钮，如果提示错误，按提示进行处理相关问题或按“Ignore All”按钮忽略错误。



解决问题后再“Run”一遍，看看还有没有问题！成功后登陆 MySQL 数据库确认是否正确建立相关数据表。

附件

### PowerDesigner 技巧

**b. 由pdm生成建表脚本时，字段超过15字符就发生错误（oracle）：**

解决办法是打开PDM后，会出现Database的菜单栏，进入Database - Edit Current DBMS - script - objects - column - maxlen，把value值调大，比如改成60。出现表或者其他对象的长度也有这种错误的话都可以选择对应的objects照此种方法更改！

**c. 生成的脚本中如何把对象的双引号去掉？**

打开cdm的情况下，进入Tools - Model Options - Naming Convention，把Name和

Code的卷标的Charcter case选项设置成Uppercase或者Lowercase，只要不是Mixed Case就行！

或者选择Database->Edit current database->Script->Sql->Format，有一项CaseSensitivityUsingQuote，它的comment为“Determines if the case sensitivity for identifiers is managed using double quotes”，表示是否适用双引号来规定标识符的大小写，可以看到右边的values默认值为“YES”，改为“No”即可！

或者在打开pdm的情况下，进入Tools - Model Options—Naming Convention，把Name和Code的卷标的Charcter case选项设置成Uppercase就可以！

#### d. 建立一个表后, 为何检测出现Existence of index的警告

A table should contain at least one column, one index, one key, and one reference.

可以不检查 Existence of index 这项，也就没有这个警告错误了！意思是说没有给表建立索引，而一个表一般至少要有一个索引，这是一个警告，不用管也没有关系！

## 实验六/七：数据库综合实验（一）/（二）

### [实验内容及要求]

本实验包括包括两次综合实验：实验七和实验八，两实验的案例材料相同，为两上连续实验，实验七主要完成数据库设计，实验八主要完成相应应用程序开发。

本次实验共需 5 周时间/20 课时完成。实验内容为选择下面的四个案例之一，分析并设计数据库，开发 PHP 应用系统。每班选择一个案例，各班不能重复，每组 1-3 人。前两周进行数据库设计、后两周进行 PHP 应用设计，最后一周检查评审。

数据库设计要求：

- 1、实体关系模型
- 2、表实例图
- 3、建表语句
- 4、样例数据入库（insert 语句）
- 5、业务逻辑实现（select 语句）

PHP 开发要求：

参照案例的内容，自主设计应用页面，要求界面友好，能够满足案例中的基本业务需求，对相关数据进行增删改查。

综合实验最终提交文档：

C D M、P D M 文件，S Q L 代码，关键抽取文档，数据库设计说明书文档，以及 PHP 应用程序。

## Case - 高校实验室管理系统

在我国，每所高校里一定数量的实验室，实验室大小不一，但其内部日常维护和管理事务烦杂。在这种情况下，设计一个高校实验室管理系统用于对高校实验室内研究人员、科研成果、工作任务、日常事务进行协调和管理是当务之急。于是通过和实验室管理人员沟通，获得如下业务信息：

- 1、实验室的主要成员包括参加该实验室的老师和参加该实验室的各年级研究生。
- 2、实验室的老师中有一部分是硕士生导师和博士生导师，每个导师可以带多名研究生，每个研究生只有一个导师，在校研究生都参加到导师所在的实验室。
- 3、实验室需要记录和管理所有研究生从入校到毕业的所有流程，包括毕业论文的开题、中期检查、答辩和小论文的发表及项目的参与情况;实验室要记录教师和实验项目的参与情况。
- 4、研究生的毕业论文开题、中期检查、答辩三个主要环节都有一定的先决条件，学生必须修满学分才能开题，每一个环节都需要学生申请，导师审查同意才能进行，三个环节逐一进行。实验室每年组织四次时间供学生开题、中期检查和答辩，如果一个环节没通过得在下次重新申请进行。
- 5、每个研究生毕业都至少必须发表一篇小论文，发表小论文以后才能进行毕业论文答辩。研究生在校可以参加到实验室中的多个项目，并在其中负责一定的工作内容和工作量。
- 6、实验室的一个具体项目一般以一个老师作为主要负责人，再带领另外一些老师和学生来开发项目，不同的人负责不同的工作，有不同的工作量。

通过和实验室人员详细面谈，了解更详细的业务的信息：

- 1、研究生分在校和毕业，硕士和博士之分，毕业研究生要记录你们的在校信息及就业单位信息。
- 2、论文分毕业论文和小论文，毕业论文不需要发表，小论文需要发表，小论文信息必须包含论文的名称、发表日期、发表的杂志、作者等信息。任何一篇论文可以有多名作者，分第一作者和合作者，第一作者只能有一个;在校研究生发表论文时应将其导师列为合作者。
- 3、项目信息包括项目的编号、名称、类型、经费、状态、启动/结束时间、项目负责人、参与者等基本信息。
- 4、老师信息包括姓名、年龄、学历、职称(讲师、副教授、教授)、头衔(博导、硕导)、

教龄等基本信息，其中博导、硕导必须为副教授及以上职称。

业务逻辑实现：

- 1、查询出实验室所有老师的姓名以及该老师所参与过的项目名称。
- 2、查询出实验室的某一位老师所带的所有学生的姓名、年级。
- 3、查询出实验室的某一位老师发表的所有论文信息，包括以第一作者和合作者身份发表的。
- 4、查询出某一研究生所参与过的实验室项目名称及其所负责的工作和工作量。
- 5、查询出近现两年内实验室发表的所有论文的信息。
- 6、查询出实验室在某一具体杂志上发表过所有论文信息。
- 7、查询出实验室的所有项目的基本信息。
- 8、查询出实验室去年实验项目中工作量最大的老师和学生。
- 9、查询出实验室某一具体项目中参与的老师、硕士研究生有。
- 10、查询实验室研三学生中未通过中期检查的学生姓名及其导师姓名。