

《数据库原理(MySQL 16 学时版本)》

实 验 指 导 书

吴思远 编著

重庆邮电大学计算机科学与技术学院

专业实验中心

2018 年 03 月

目录

实验一 Navicat for MySQL 使用基础与 SQL 语言的 DDL	3
实验二 SQL 语言数据操纵语言 DML.....	16
实验三 SQL 语言数据查询语言 DQL.....	21
实验四 MySQL 应用开发环境的建立	27
实验五 MySQL 的存储过程	42
实验六 MySQL 的触发器	54
实验七 PowerDesigner 的 CDM 应用.....	62
实验八 PowerDesigner 的 PDM 应用	71

实验一 Navicat for MySQL 使用基础与 SQL 语言的

DDL

一、实验目的：

Navicat for MySQL 是一套专为 MySQL 设计的高性能数据库管理及开发工具。它可以用于任何版本 3.21 或以上的 MySQL 数据库服务器，并支持大部份 MySQL 最新版本的功能，包括触发器、存储过程、函数、事件、视图、管理用户等。

SQL (Structured Query Language) 语言是关系数据库的标准语言。是一种介于关系代数与关系演算之间的结构化查询语言，其功能并不仅仅是查询，SQL 语言是一个通用的、功能极强的关系数据库语言。

本实验首先了解并掌握如何使用 Navicat for MySQL 连接数据库、浏览数据库对象、运行 SQL 语句和 SQL 脚本，以及如何编辑和运行 SQL 语句。Navicat for MySQL 可以提高工作效率并简化数据库开发任务。

本次实验还了解 SQL 语言中 DDL 语言的 CREATE、DROP、ALTER 对表、索引、视图的操作，掌握在 Navicat for MySQL 中用 DDL 语言进行对表、索引、视图的增加、删除和改动。掌握使用 SQL 语句增加或删除约束，加深对完整性概念的理解，达到灵活应用的目的。掌握使用 SQL 语句定义和删除同义词。

二、实验要求：

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点：

1、重点：

- (1) Navicat for MySQL 的安装及启动。
- (2) Navicat for MySQL 连接数据库、浏览数据库对象、运行 SQL 语句和 SQL 脚本。
- (3) 用 SQL 语句定义表结构（创建、修改和删除）。

- (4) 用 SQL 语句定义索引（创建、删除）。
- (5) 用 SQL 语句定义视图（创建、删除）。
- (6) 用 SQL 语句定义同义词（创建、删除）。

2、难点：

Navicat for MySQL 连接数据库的参数配置，包括用户名、密码、主机名、端口。

完整性约束的定义、增加及删除。

同义词的定义与删除。

四、仪器设备及用具：

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：本机已安装 MySQL5.X 数据库平台。

五、教学过程：

（一）实验预习

- (1) 熟悉 SQL 中的 CREATE、DROP、ALTER 语句的格式及所用的关键字含义及用法。
- (2) 掌握完整性约束定义、增加和删除的一般用法。
- (3) 掌握同义词定义、删除的一般用法。

（二）实验原理

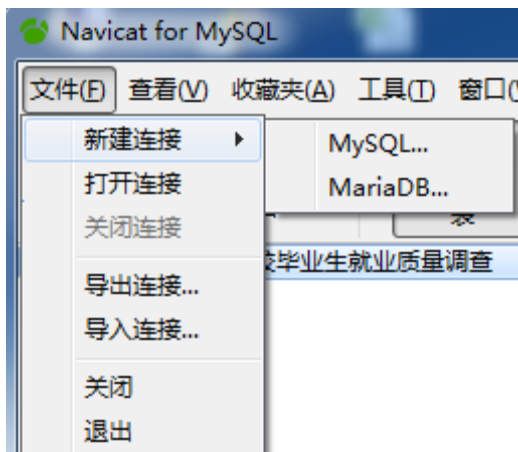
观看 Navicat for MySQL 连接配置教学视频，在 Navicat for MySQL 中完成连接参数配置，并进行数据库连接、浏览数据库对象、运行 SQL 语句和 SQL 脚本等操作。

在 Navicat for MySQL 中使用 CREATE 命令完成对表、索引、视图、同义词的创建，使用 DROP 命令完成对表、索引、视图、同义词的删除，使用 ALTER 命令对表结构进行修改及完整性约束的增加、删除。

（三）实验内容

使用 Navicat for MySQL 管理数据库对象首先要创建数据库连接。

1. 双击桌面 Navicat for MySQL 快捷方式。
2. 在文件菜单中，先选择新建连接，再选择 MySQL...

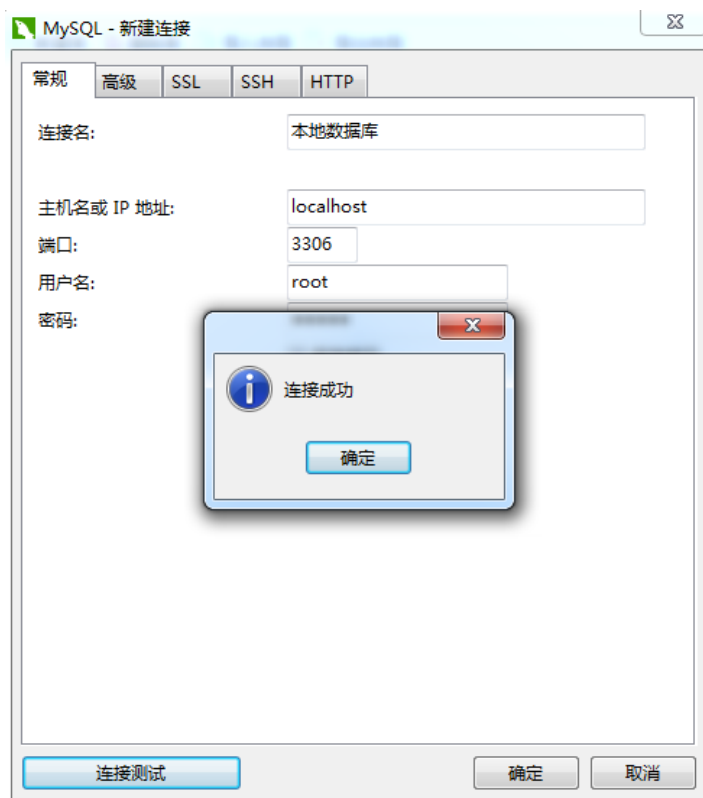


3. 在新建连接常规设置页面下，连接名中输入“本地数据库”，主机名或 IP 地址输入“localhost”，端口保持 3306 不变，用户名保持 root 不变，密码输入“mysql”，然后单击连接测试按钮。



4. 测试连接状态已成功，但没有保存该连接。要保存该连接，请单击连接成功提示框的确定按钮后，再单击 MySQL – 新建连接的确定按钮。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

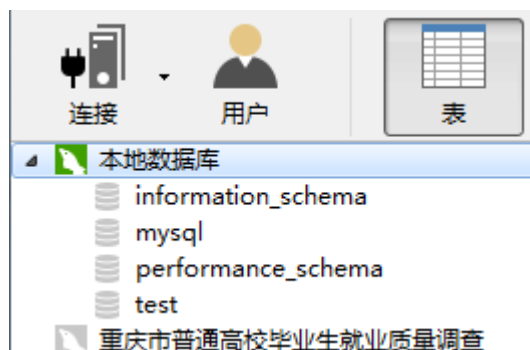


5.保存了该连接，可以在列表中看到该数据库。

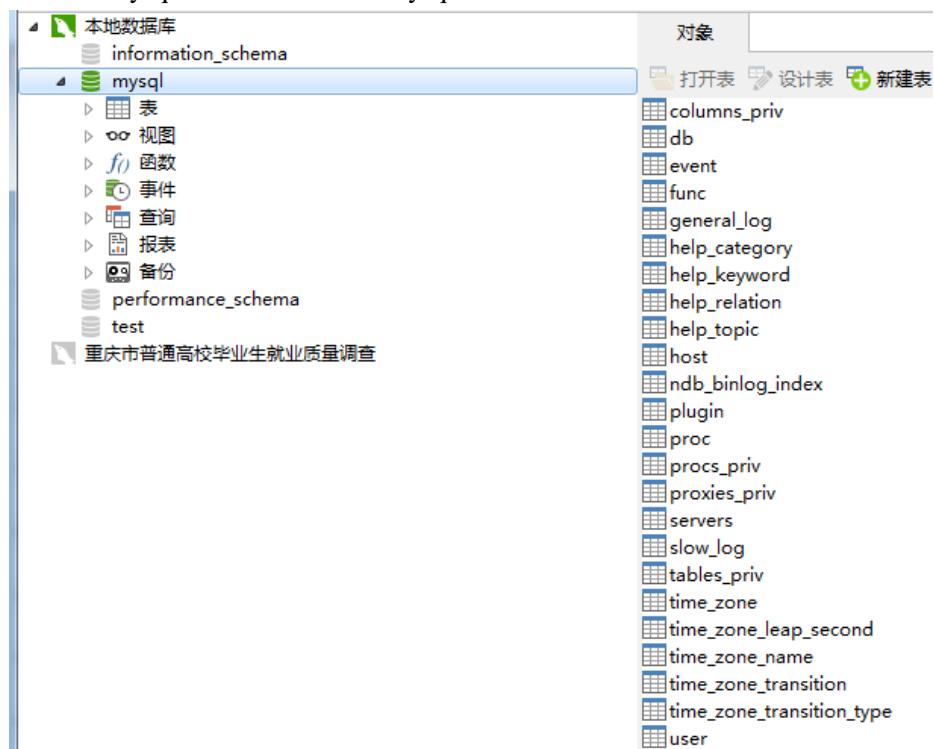


6.双击本地数据库连接名，连接到本地 MySQL 数据库服务器上。我们可以看到本地数据库服务器上有四个不同名称的数据库。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

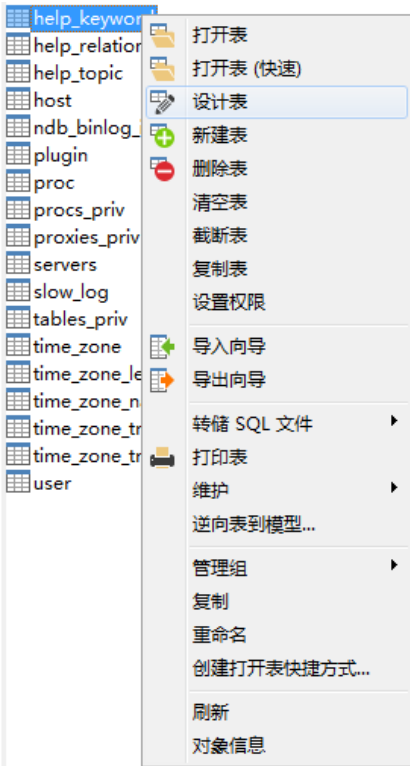


7. 双击 mysql 数据库，连接到 mysql 数据库上。



8. 选择 help_keyword 表，单击右键选择设计表。

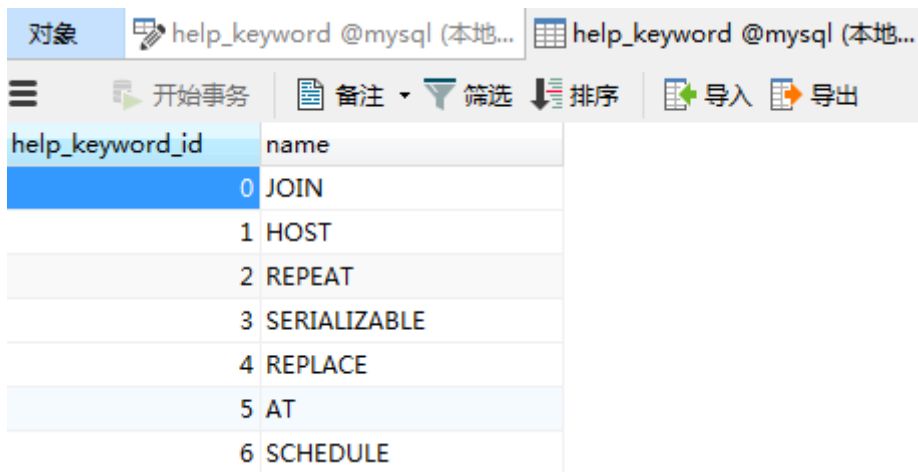
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



9.查看表定义、索引、外键、触发器。

对象 help_keyword @mysql (本地...)						
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移						
栏位 索引 外键 触发器 选项 注释 SQL 预览						
名	类型	长度	小数点	不是 null		
help_keyword_id	int	10	0	<input checked="" type="checkbox"/>	1	
name	char	64	0	<input checked="" type="checkbox"/>		

10.要查看该表数据，请在对象页，单击 help_keyword 表，单击右键选择打开表。



help_keyword_id	name
0	JOIN
1	HOST
2	REPEAT
3	SERIALIZABLE
4	REPLACE
5	AT
6	SCHEDULE

11.数据随即显示。

12.运行 Navicat for MySQL，连接到 test 数据库，用如下语句进行表操作，详细的语法格式如下：

CREATE TABLE 表名字

(column1 datatype [DEFAULT expression],
column1 datatype [DEFAULT expression],
.....)

| [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)

| KEY [index_name] [index_type] (index_col_name,...)

| INDEX [index_name] [index_type] (index_col_name,...)

| [CONSTRAINT [symbol]] UNIQUE [INDEX]

[index_name] [index_type] (index_col_name,...)

| [FULLTEXT|SPATIAL] [INDEX] [index_name] (index_col_name,...)

| [CONSTRAINT [symbol]] FOREIGN KEY

[index_name] (index_col_name,...) [reference_definition]

| CHECK (expr)

建立表主要指定义下列信息：

列定义、主键定义、键定义、索引定义 、完整性约束、外键定义、表达式检查，例如在新建查询中输入如下语句：

CREATE TABLE NEW_DEPT92150033

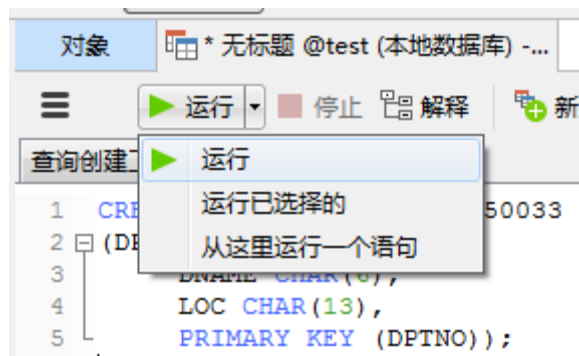
(DPTNO DECIMAL(10,2),

 DNAME CHAR(6),

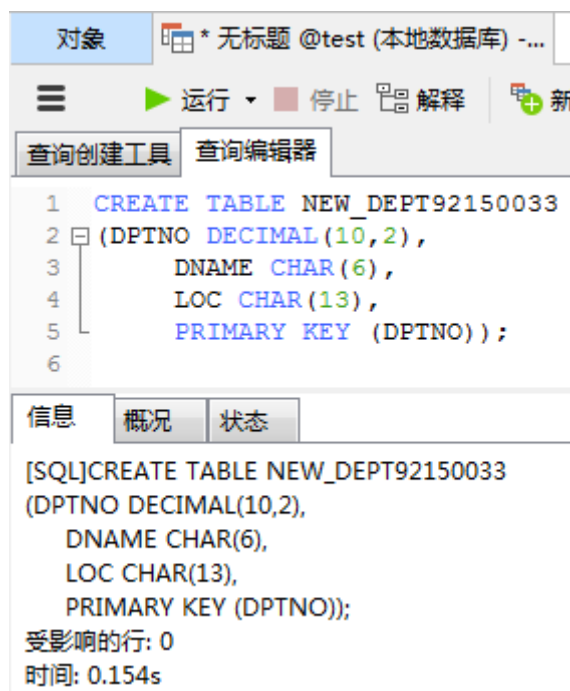
 LOC CHAR(13),

 PRIMARY KEY (DPTNO));

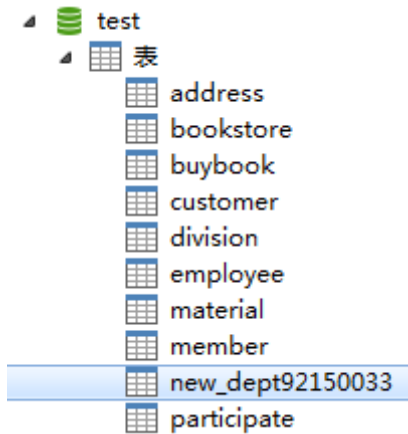
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



点击运行，创建表 NEW_DEPT92150033，如下图所示：



选中表,单击右键，执行刷新进行表刷新，这时你可以看到新建的表。



更改表详细的语法格式如下：

增加一个列：

```
alter table 表名字 ADD [COLUMN] column_definition [FIRST | AFTER  
col_name ],...);
```

修改一个列：

```
alter table 表名字 MODIFY [COLUMN] column_definition [FIRST | AFTER  
col_name ],...);
```

删除一个列：

```
alter table 表名字 DROP [COLUMN] col_name;
```

在查询编辑器中执行 create table 语句首先建立一个 test92150033 表,然后分别使用 alter table add、alter table modify、alter table drop column 在表 test92150033 上来增加两个列、修改一个列和删除一个列。SQL 语句如下所示：

```
create table test92150033 (id varchar(20) not null);
```

```
alter table test92150033 ADD (name varchar (30) default '无名氏' not null);
```

```
alter table test92150033 ADD (age integer not null);
```

```
alter table test92150033 MODIFY name varchar(16);
```

```
alter table test92150033 drop column age;
```

删除表语法：

```
Drop table 表名字;
```

例如在查询编辑器中执行如下语句删除表：

```
Drop table test92150033;
```

13. 用如下语句进行视图操作，详细的语法格式如下：

```
CREATE VIEW 视图名 AS SELECT ... FROM ...;
```

视图是一个逻辑表，它允许操作者从其它表或视图存取数据，视图本身不包含数据。视图所基于的表称为基表。

引入视图有下列作用：

提供附加的表安全级，限制存取基表的行或/和列集合。

隐藏数据复杂性。为数据提供另一种观点。

例如在查询编辑器中执行如下语句建立视图：

先建立基表：

```
CREATE TABLE t (qty INT, price INT);
```

插入记录

```
INSERT INTO t VALUES(3, 50);
```

```
INSERT INTO t VALUES(5, 10);
```

在基表 t 的基础上，创建视图 v。

```
CREATE VIEW v AS SELECT qty, price, qty*price AS value FROM t;
```

从视图 v 检索数据

```
SELECT * FROM v;
```

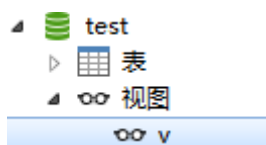
结果如图所示。



The screenshot shows a SQL query editor with the query 'SELECT * FROM v;' entered. Below the query, there is a table with three columns: 'qty', 'price', and 'value'. The table contains two rows of data: (3, 50, 150) and (5, 10, 50).

qty	price	value
3	50	150
5	10	50

点击视图可以看到新建立的视图 v,如图所示。



删除视图语法：

```
Drop View 视图名;
```

例如在查询编辑器中执行如下语句删除视图：

```
drop view v;
```

14.用如下语句对索引进行操作，详细的语法格式如下：

```
建立索引：CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name  
[USING index_type]
```

ON tbl_name (index_col_name,...)

index_col_name:

col_name [(length)] [ASC | DESC]

索引是种数据库对象。对于在表或聚集的索引列上的每一值将包含一项，为行提供直接的快速存取。

例如，在 ADDRESS 表的 CUSTOMERID 上建立索引 IX_ADD_CUST:

CREATE INDEX IX_ADD_CUST ON ADDRESS(CUSTOMERID);

删除索引语法:

DROP INDEX index_name ON tbl_name;

例如在查询编辑器中执行如下语句删除索引:

Drop Index IX_ADD_CUST on ADDRESS;

(四) 注意事项

- 1、在配置主机名或 IP 地址时，如果访问数据库服务器，可以用本机的 IP 地址替代 localhost。
- 2、MySQL 的端口值默认为 3306。
- 3、建立视图 v 之前，必须创建表 t 和插入记录。

六、实验步骤:

按照五、教学过程中(三)实验内容所示步骤，执行以下内容:

- 1.启动 Navicat for MySQL。
- 2.在 MySQL – 新建连接中完成连接参数配置，并完成连接测试和保存连接参数。
- 3.查看 mysql 数据库的 help_category 表的定义、表中数据与索引、完整性约束等。
- 4.查看 mysql 数据库的其它数据库对象，如视图、索引、存储过程、函数、触发器。
- 5.在 Navicat for MySQL 中打开查询，新建查询，运行简单的 SQL 语句，如 select * from help_keyword where help_keyword_id<=3; 观察下结果。
- 6.登录到本地数据库服务器后，连接到 test 数据库上。
- 7.用 SQL 语句，建立如下所示的表 student:

属性名	类型	长度
studentid	Varchar	10
name	Varchar	20
sex	Varchar	2
age	Integer	
Fee	DECIMAL	10, 2

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

address	Varchar	50
memo	Varchar	300

8. 用 Create Table 语句建表 CourseAa，表结构如下所示：

属性名	类型	长度
Aa1	Varchar	20
Aa2	Integer	
Aa3	DECIMAL	10

9. 用 Create Table 语句建表 ChooseBb，表结构如下所示：

属性名	类型	长度
Bb1	Varchar	30
Bb2	Integer	
Bb3	DECIMAL	6

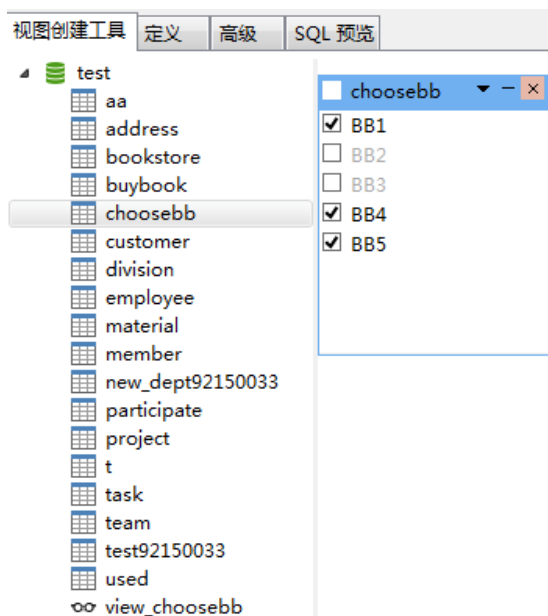
10. 用 Drop Table 语句删除表 CourseAa。

11. 用 Alter Table 语句更新表 ChooseBb，添加一个属性名 Bb4，类型 Varchar，长度 20，完整性约束设为非空值，缺省值设为字符“系统测试值”。

12. 用 Alter Table 语句更新表 ChooseBb，添加一个属性名 Bb5，类型 Varchar，长度 10，完整性约束设为主码。完成后，表 ChooseBb 的设计如下所示。

对象	choosebb @test (本地数据库...)					
	新建	保存	另存为	添加栏位	插入栏位	删除栏位
栏位	索引	外键	触发器	选项	注释	SQL 预览
名	类型	长度	小数点	不是 null		
BB1	decimal	10	0	<input type="checkbox"/>		
BB2	char	6	0	<input type="checkbox"/>		
BB3	int	11	0	<input type="checkbox"/>		
BB4	varchar	30	0	<input checked="" type="checkbox"/>		
BB5	varchar	30	0	<input checked="" type="checkbox"/>		1

13. 用 Create View 语句建立一个视图 View_Choosebb，生成的视图属性名（View_bb1，View_bb2，view_bb3），其中 View_bb1 对应于基表 ChooseBb 的 Bb1、View_bb2 对应于基表 ChooseBb 的 Bb4、view_bb3 对应于基表 ChooseBb 的 Bb5。完成后，视图 View_Choosebb 的设计如下所示。



14. 用 Drop View 语句删除视图 View_Chosebb。

15. 用 Create Index 语句对表 ChooseBb 的 Bb2 属性建立一个升序索引，索引名 Index_bb2。用 Create Index 语句对表 ChooseBb 的 Bb4 属性建立一个降序索引，索引名 Index_bb4。

16. 用 Drop Index 语句删除索引 Index_bb2。

七、思考与练习：

- 1、还可以用什么方法启动 Navicat for MySQL？
- 2、主机名或 IP 地址在什么时候用 localhost？什么时候用本机 IP 地址？
- 3、端口值能更改成其它值吗？
- 4、实体完整性通过什么机制保证？
- 5、唯一性约束与主键有何区别？

八、实验报告要求：

参见数据库实验报告模版

参考文献：

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验二 SQL 语言数据操纵语言 DML

一、实验目的

SQL 语言的数据操纵功能通过 DML(数据维护语言)实现。DML 包括数据查询和数据更新两种数据操纵语句。其中，数据查询指对数据库中的数据查询、统计、分组、排序等操作；数据更新指数据的插入、更新和删除等数据维护操作。

本次实验了解 DML 语言的 INSERT、UPDATE、DELETE 等数据维护语言，掌握在 Navicat for MySQL 中用 DML 语言的 INSERT、UPDATE、DELETE 对表进行数据插入、更新和删除。

二、实验要求

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，实验时注意每种 SQL 语句的基本命令及各个关键字的含义，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点

1、重点：

- (1) 用 INSERT 语句向表中插入数据。
- (2) 用 UPDATE 语句更新(修改)表中已有数据。
- (3) 用 DELETE 语句删除表中数据。

2、难点：

INSERT 语句中，使用从子表插入数据的 FROM 子句。

UPDATE 语句和 DELETE 语句的 WHERE 子句。

四、仪器设备及用具

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：本机已安装 MySQL 5.X 数据库平台。

五、教学过程

（一）实验预习

（1）熟悉 SQL 中的 INSERT、UPDATE、DELETE 语句的格式及所用的关键字含义及用法。

（2）掌握 INSERT 语句的两种不同插入数据方式。

（3）掌握 UPDATE 语句和 DELETE 的一般用法。

（二）实验原理

在 Navicat for MySQL 中使用 INSERT 语句向表中插入数据。使用 UPDATE 语句更新(修改)表中已有数据。使用 DELETE 语句删除表中数据。

（三）实验内容

1. 启动 Navicat for MySQL, 用 INSERT 语句对表进行单条数据插入操作, 语法格式如下:

```
Insert Into <表名> [(<属性列 1>[,<属性列 2>]...)] Values (<常量 1>[,<常量 2>]...);
```

其中, INSERT INTO 指明要插入的表以及表中的属性列(字段), VALUES 指明要插入相应属性列(字段)的值。

例如:

```
Insert into test (name,age,score,address) values ('吴思远',35,590.00,'重庆邮电大学计算机学院');
```

用 INSERT 语句对表进行批量数据插入操作, 语法格式如下:

```
Insert Into <表名 1> [(<属性列 1>[,<属性列 2>]...)] Select [(<属性列 1>[,<属性列 2>]...)] From <表名 2>;
```

INSERT 语句的批量插入，把从其他表中查询出来数据插入到当前表中。

例如：

```
Insert      into      test      (name, age, score, address)      Select  
name, age, score, address from test_temp;
```

2. 在 Navicat for MySQL 中，用 UPDATE 语句对表中已有的数据进行修改，语法格式如下：

```
Update <表名> Set <属性列>=<表达式>[, <属性列>=<表达式>][, ...  
n] [Where <条件>];
```

其中，表名指定要更新的表，SET 指定要更新的属性列(字段)及其相应的值，WHERE 指定更新条件。

例如：

```
Update test set age = 37 where name=' 吴思远' and address=' 重庆  
邮电大学计算机科学与技术学院' ;
```

3. 在 Navicat for MySQL 中，用 DELETE 语句对表中已有的数据进行删除，语法格式如下：Delete From <表名> [Where <条件>];

其中，FROM 指定要删除数据的表，WHERE 指定要删除数据的条件。

例如：Delete From test Where age>=35;

(四) 注意事项

- 1、INSERT 语句中，无论是哪一种用法，都应该注意要插入的值与要插入的字段相互对应。
- 2、UPDATE 语句中，如果没有指定更新条件 WHERE 子句，则对表中所有记录进行更新。
- 3、DELETE 语句中，如果没有指定删除条件 WHERE 子句，则对表中所有记录进行删除。使用 DELETE 语句删除表中数据时，并不能释放被占用的数据块空间，将来还可以使用回退 (Rollback) 操作。

六、实验步骤：

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

执行以下内容：

1. 启动 Navicat for MySQL，在 MySQL - 新建连接中完成连接参数配置。

2. 登录到本地数据库服务器后，连接到 test 数据库上。

以下操作请全部使用 SQL 语句完成

3. 首先用 Create Table 语句建立 test 表，test 表定义如下所示：

字段名	类型	长度
Name	Varchar	20
Age	Integer	
Score	Numeric	10, 2(总共长 10 位，小数部分 2 位)
Address	Varchar	60

4. 用 INSERT 语句对表 test, 插入如下 6 条记录：

Name	Age	Score	Address
赵一	20	580.00	重邮宿舍 12-3-5
钱二	19	540.00	南福苑 5-2-9
孙三	21	555.50	学生新区 21-5-15
李四	22	505.00	重邮宿舍 8-6-22
周五	20	495.50	学生新区 23-4-8
吴六	19	435.00	南福苑 2-5-12

5. 再用 Create Table 语句建立 test_temp 表，test_temp 表定义如下所示：

字段名	类型	长度
Name	Varchar	20
Age	Integer	
Score	Numeric	10, 2(总共长 10 位，小数部分 2 位)
Address	Varchar	60

6. 用 INSERT 语句对表 test_temp, 插入如下 3 条记录：

Name	Age	Score	Address
------	-----	-------	---------

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

郑七	21	490.50	重邮宿舍 11-2-1
张八	20	560.00	南福苑 3-3-3
王九	10	515.00	学生新区 19-7-1

7. 用 INSERT INTO ... SELECT ... FROM 语句，将 test_temp 表中的数据，插入到 test 表中。

8. 用 UPDATE 语句将 test 表中年龄小于等于 20 的数据，将成绩更新为原来的成绩加 5 分。

9. 用 UPDATE 语句将 test 表中居住在南福苑所有学生的年龄减少 1 岁。

10. 用 DELETE 语句将 test 表中年龄大于等于 21 并且成绩大于等于 500 的学生数据删除掉。

11. 用 DELETE 语句将 test 表中成绩小于 550 并且居住在重邮宿舍的学生数据删除掉。

以上 SQL 语句请保存，以备老师检查和写实验报告之用。

七、思考与练习

- 1、如何确保前面执行的 DML 操作物理存储到数据库中？
- 2、如何取消前面执行的 DML 操作？

八、实验报告要求

参见数据库实验报告模版

参考文献

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验三 SQL 语言数据查询语言 DQL

一、实验目的

数据查询语言指对数据库中的数据查询、统计、分组、排序等操作。查询语句可以分为简单查询、连接查询、嵌套查询和组合查询等。

本次实验了解 SQL 语言的 SELECT 语句对数据的查询,学会在 Navicat for MySQL 中用 SELECT 语句对表中的数据进行简单查询、连接查询、嵌套查询和组合查询。

二、实验要求

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，实验时注意每种 SQL 语句的基本命令及各个关键字的含义，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点

1、重点：

- (1) 查询结果的排序和分组
- (2) 聚合函数的使用
- (3) 内连接查询
- (4) 内外层不相关子查询

2、难点：

连接查询时，两个表或者多个表之间连接条件的使用。

嵌套查询时，In、Exist 操作符的使用。

进行组合查询时，集合操作符的使用。

SELECT 语句的 GROUP BY... HAVING 子句、ORDER BY 子句的使用。

自身连接和内外层相关子查询。

四、仪器设备及用具

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：本机已安装 MySQL 5.X 数据库平台。

五、教学过程

（一）实验预习

- （1）熟悉 SQL 中的 SELECT 语句的格式及所用的关键字含义及用法。
- （2）掌握用 SELECT 语句进行简单查询的方法。
- （3）掌握用 SELECT 语句进行连接查询的方法。
- （4）掌握用 SELECT 语句进行嵌套查询的方法。
- （5）掌握用 SELECT 语句进行组合查询的方法。

（二）实验原理

在 Navicat for MySQL 中使用 SELECT 语句从表中查询数据、统计数据及对数据进行分组和排序等操作。

（三）实验内容

1. 启动 Navicat for MySQL，用 SELECT 语句对表进行简单查询操作，整个查询过程只涉及到一个表，是最基本的查询语句。语法格式如下：

Select <目标列组> From <表名> [Where <元组选择条件>] [Group by <分组列> [Having <组选择条件>]] [Order by <排序列 1> <排序要求> [, ...n]]

2. 用 SELECT 语句对表进行连接查询操作，连接查询涉及被连接和连接两个表，所以数据源一般为多个表。用来连接两个表的条件称为连接条件，一般格式为：

[<表名 1>.<列名 1> <比较运算符> [<表名 2>.<列名 2>

其中比较运算符主要有：=、>、<、>=、<=、!=。

3. 用 SELECT 语句对表进行嵌套查询操作，一个 Select...From...Where 语句称为一个查询块，将一个查询块嵌套在另一个查询块的 Where 子句或 Having 短语的条件中的查询，就是嵌套查询。主要有使用 In 操作符、Exist 操作符、使用比较运算符的嵌套查询方式。

4. 用 SELECT 语句对表进行集合查询操作，将 SELECT 语句的查询结果集再进行集合运算就构成了 SQL 的集合查询。集合查询操作符有 Union(并操作)、Intersect(交操作)和 Minus(差操作)。目前 MySQL 仅支持 Union(并操作)。

(四) 注意事项

- 1、对表指定别名后，在 where 条件中要引用表名时，应引用别名。
 - 2、对于空值，只能用 is null 或 not is null 进行比较，而不能用=比较。
- 例： select * from sc where grade is null
- 3、在作 like 进行模糊查询时，注意%和_的区别，同时要注意汉字所占用字节。
 - 4、having 只能用在 group by 子句后面，不能用在 where 子句后面。
 - 5、自身连接必须为表取别名。
 - 6、在连接查询中，如果一个字段来源于两个及两个以上的表，需要指明字段的来源，即需加上表名前缀，格式为“表名. 字段名”。
 - 7、在嵌套查询中，order by 只能用于最外层嵌套。
 - 8、内外层相关查询中，应对表取别名。

六、实验步骤：

执行以下内容：

1. 启动 Navicat for MySQL，在 MySQL - 新建连接中完成连接参数配置。
2. 登录到本地数据库服务器后，连接到 test 数据库上。
3. 用 Create Table 建立 Student 表，表结构如下所示：

属性名	类型	长度
SNO	Varchar	20
Name	Varchar	10

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

Age	Integer	
College	Varchar	30

4. 用 Create Table 建立 Course 表，表结构如下所示：

字段名	类型	长度
CourseID	Varchar	15
CourseName	Varchar	30
CourseBeforeID	Varchar	15

5. 用 Create Table 建立 Choose 表，表结构如下所示：

字段名	类型	长度
SNO	Varchar	20
CourseID	Varchar	30
Score	DECIMAL	5, 2

6. 用 INSERT 语句向 Student 表中插入 3 个元组：

注意：每条元组一个完整的 INSERT 语句，3 个元组需要写 3 个 INSERT 语句。

SNO	Name	Age	College
S00001	张三	20	计算机学院
S00002	李四	19	通信学院
S00003	王五	21	计算机学院

7. 用 INSERT 语句向 Course 表中插入 3 个元组：

注意：每条元组一个完整的 INSERT 语句，3 个元组需要写 3 个 INSERT 语句。

CourseID	CourseName	CourseBeforeID
C1	计算机引论	空值 (NULL)
C2	C 语言	C1
C3	数据结构	C2

8. 用 INSERT 语句向 Choose 表中插入 7 个元组：

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

注意：每条元组一个完整的 INSERT 语句，7 个元组需要写 7 个 INSERT 语句。

SNO	CourseID	Score
S00001	C1	95
S00001	C2	80
S00001	C3	84
S00002	C1	80
S00002	C2	85
S00003	C1	78
S00003	C3	70

9. 用 SELECT 语句，查询计算机学院学生的学号和姓名。

10. 用 SELECT 语句的 between...and...表示查询条件，查询年龄在 20~23 岁的学生信息。

11. 用 SELECT 语句的 COUNT() 聚集函数，统计 Student 表中学生总人数。

12. 分别用 SELECT 语句的 max()、min()、sum()、avg() 四个聚集函数，计算 Choose 表中 C1 课程的最高分、最低分、总分、平均分。

13. 用 SELECT 语句对空值 (NULL) 的判断条件，查询 Course 表中先修课称编号为空值的课程编号和课程名称。

14. 用 SELECT 语句的连接查询，查询学生的学号、姓名、选修的课程名及成绩。

15. 用 SELECT 的存在量词 EXISTS，查询与“张三”在同一个学院学习的学生信息。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

16. 用 SELECT 语句的嵌套查询，查询选修 C1 课程的成绩低于“张三”的学生的学号和成绩。

17. 用 SELECT 语句的组合查询(UNION)，查询选修了 C1 课程或者选修了 C3 课程的学生学号。

18. 用 SELECT 语句的组合查询(UNION)与 DISTINCT 短语，查询选修了 C1 课程或者选修了 C3 课程的学生学号，并取消重复的数据。

以上 SQL 语句请保存，以备老师检查和写实验报告之用。

七、思考与练习

- 1、Where 与 Having 语句有何差别？
- 2、Count (*) 与 Count (字段名) 有何差别？
- 3、LIKE '[^s][^y][^s]%' 与 NOT LIKE 'sys%' 有无区别？请加以验证？
- 4、全连接、左连接、右连接有什么不同？
- 5、IN 与 Exists 语句有何区别？
- 6、不相关子查询、相关子查询有何区别？
- 7、UNION 与 UNION ALL 有何差别，谁的性能高？

八、实验报告要求

参见数据库实验报告模版

参考文献

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验四 MySQL 应用开发环境的建立

一、实验目的

通过在 Windows 平台下，搭建 Apache Httpd Web 服务器，PHP 脚本支持，访问 MySQL 数据库等，学习了解 MySQL 应用开发环境的建立过程。

本次实验通过安装 Apache HTTPD Server 2.2、PHP 5.3.28，配置 httpd.conf 和 PHP.ini 文件，完成 MySQL 的 PHP 应用开发环境的建立。

二、实验要求

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，实验时注意每种 SQL 语句的基本命令及各个关键字的含义，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点

1、重点：

- (1) Apache HTTPD Server 的安装
- (2) PHP 5.3.28 的安装

2、难点：

- httpd.conf 文件的配置。
- PHP.ini 文件的配置。
- mysqlconn.php 文件的编写和测试。

四、仪器设备及用具

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：本机已安装 MySQL 5.X 数据库平台。

五、教学过程

（一）实验预习

- （1）熟悉 Windows 平台下最常用的 MySQL 应用开发环境。
- （2）掌握 HTTP Server 和 PHP 各自的用途。


（二）实验原理

安装 HTTP Server 和 PHP 并进行配置。

（三）实验内容

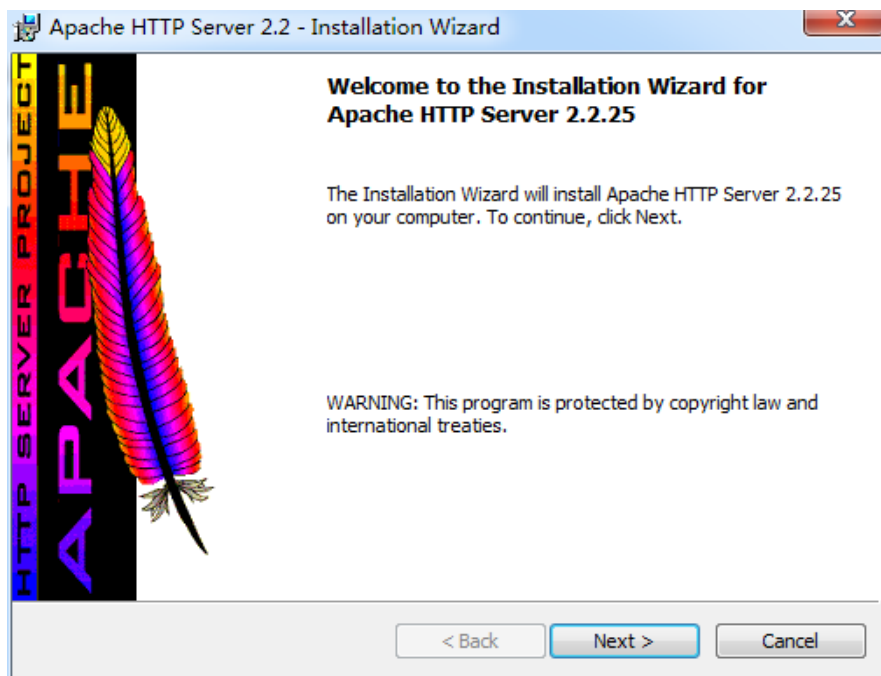
本次实验所用软件为 Apache 的 `httpd-2.2.25-win32-x86-openssl-0.9.8y.exe` 和 PHP 的 `php-5.3.28-Win32-VC9-x86.zip` 均在 [ftp://202.202.43.106/1.实验类课程\(含随课实验、独立实验等\)/A2040100-数据库原理/软件](ftp://202.202.43.106/1.实验类课程(含随课实验、独立实验等)/A2040100-数据库原理/软件) 目录下, 请先下载到本地电脑备用。

1. 安装 Apache httpd 服务器

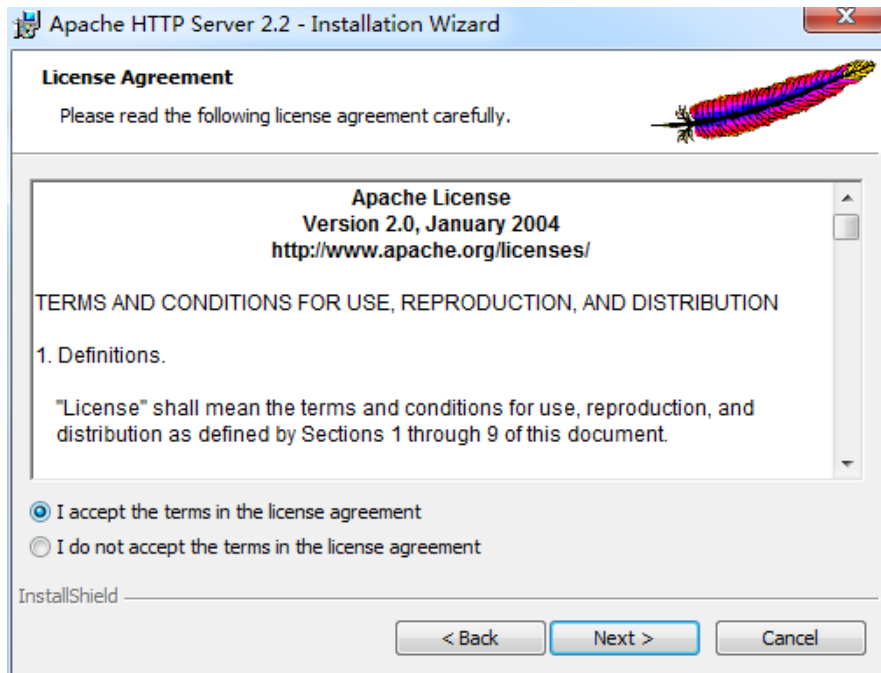
双击  `httpd-2.2.25-win32-x86-openssl-0.9.8y` , 出现 Apache HTTP Server

2.2.25 的安装向导界面, 点 “Next” 继续。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

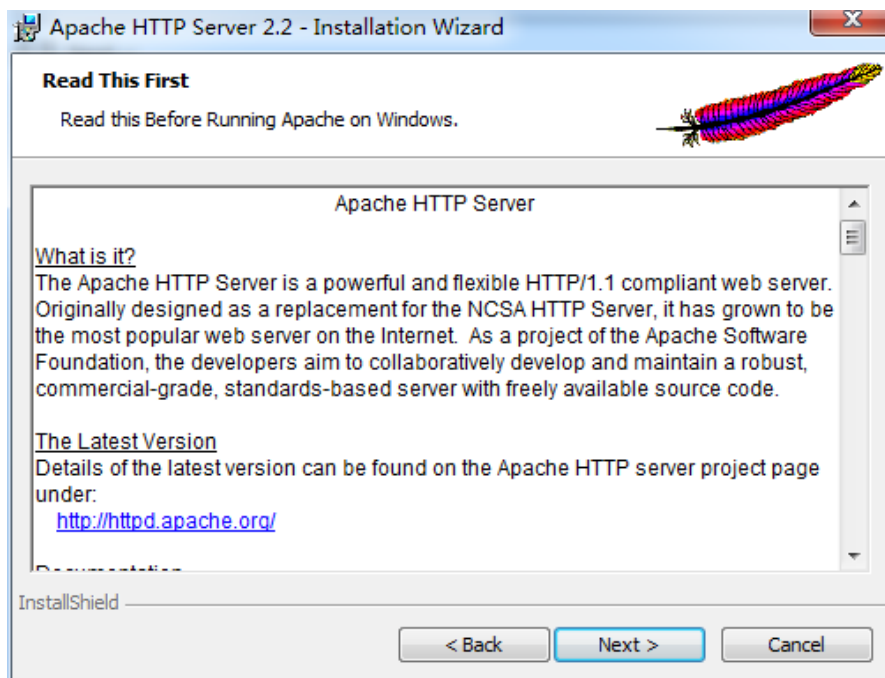


确认同意软件安装使用许可条例，选择“I accept the terms in the license agreement”，点“Next”继续。



将 Apache 安装到 Windows 上的使用须知，请阅读完毕后，按“Next”继续。

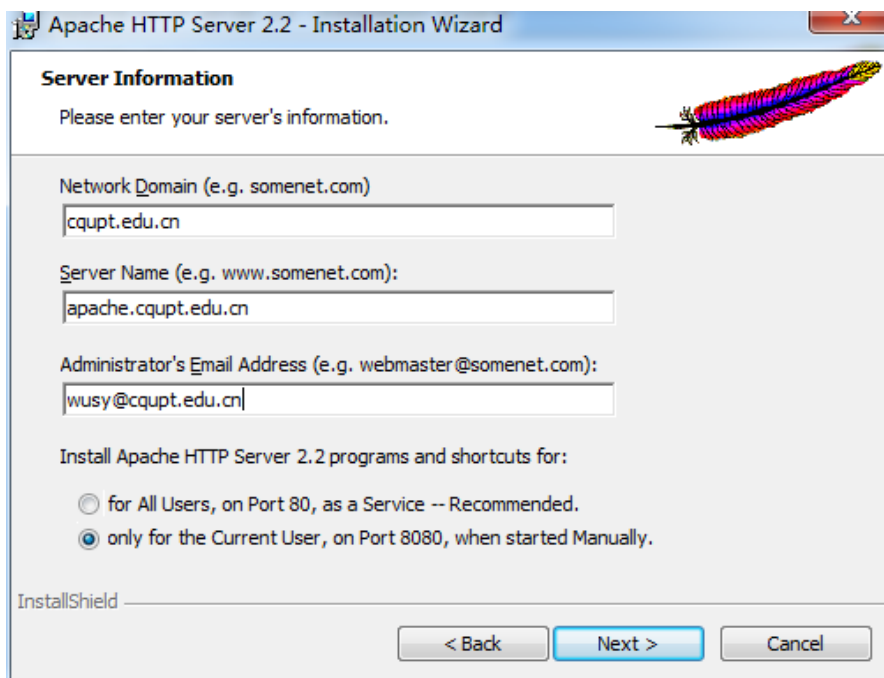
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



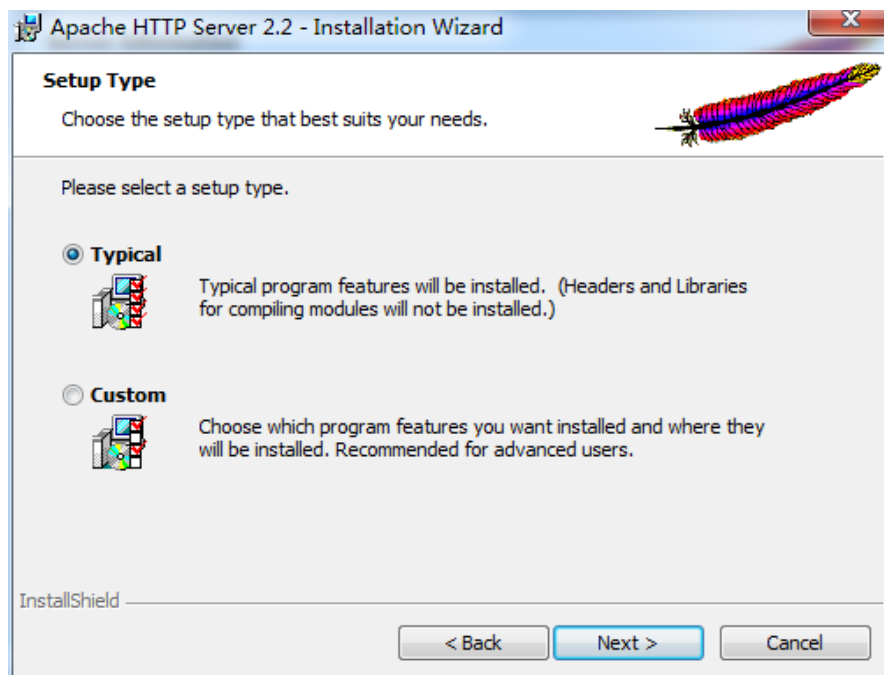
设置系统信息，在 Network Domain 下填入您的域名（比如：cqupt.edu.cn），在 Server Name 下填入您的服务器名称（比如：apache.cqupt.edu.cn，也就是主机名加上域名），在 Administrator's Email Address 下填入系统管理员的联系电子邮件地址（比如：wusy@cqupt.edu.cn），上述三条信息仅供参考，其中联系电子邮件地址会在当系统故障时提供给访问者，三条信息均可任意填写，无效的也行。下面有两个选择，图片上选择的是为系统所有用户安装，使用默认的 80 端口，并作为系统服务自动启动；另外一个是为当前用户安装，使用端口 8080，手动启动。

本次实验选择端口 8080，如图所示。按“Next”继续。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



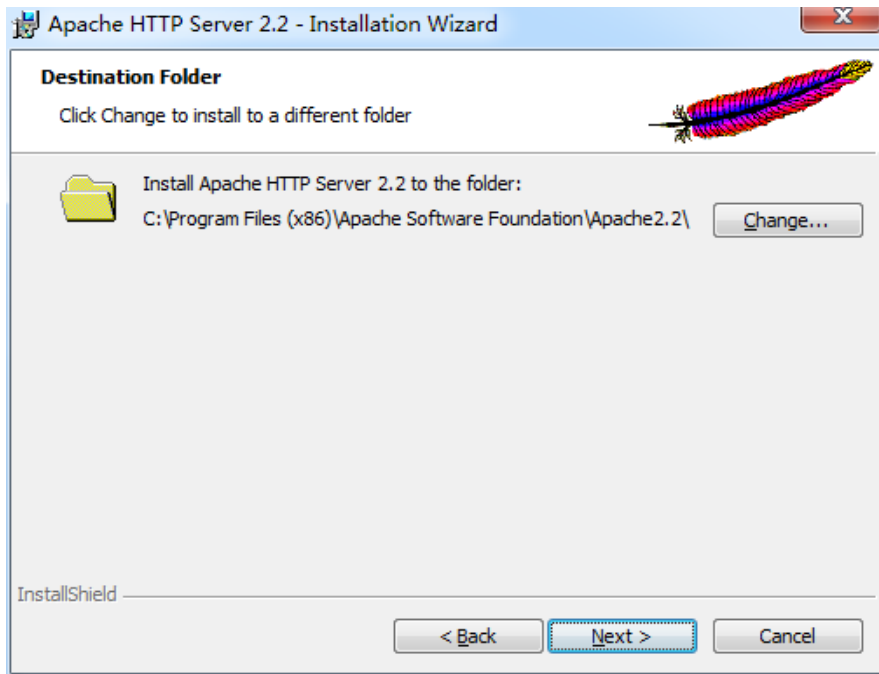
选择安装类型，Typical 为默认安装，Custom 为用户自定义安装，我们这里选择 Typical。按“Next”继续。



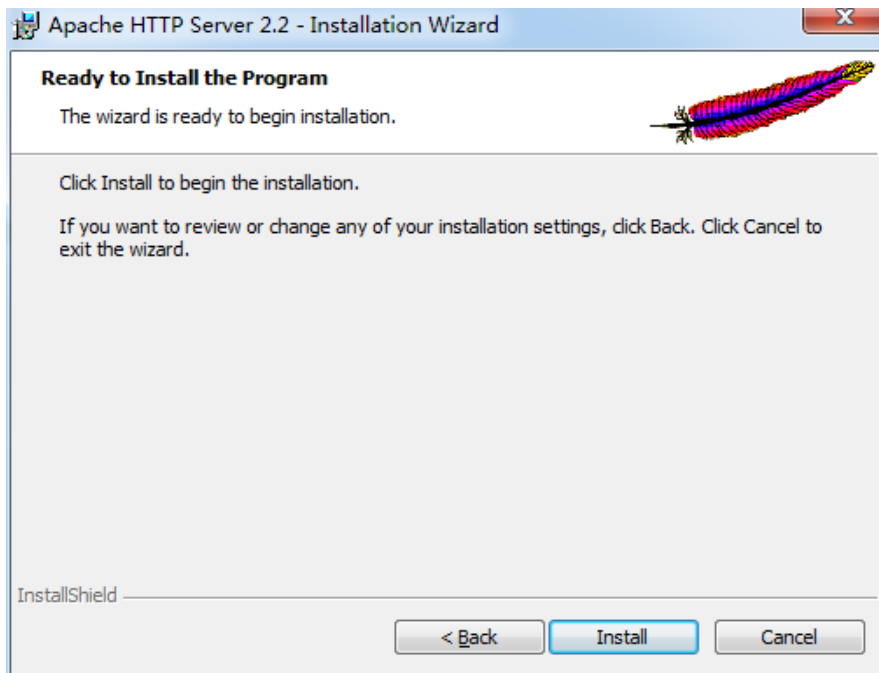
使用默认的安装目录 C:\Program Files (x86)\Apache Software

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

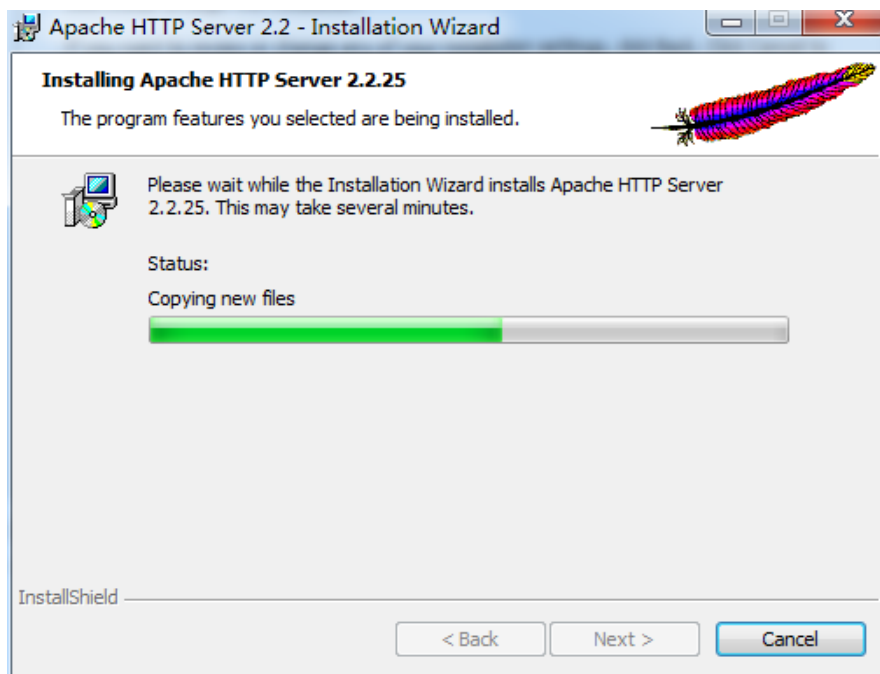
Foundation\Apache2.2，选“Next”继续。



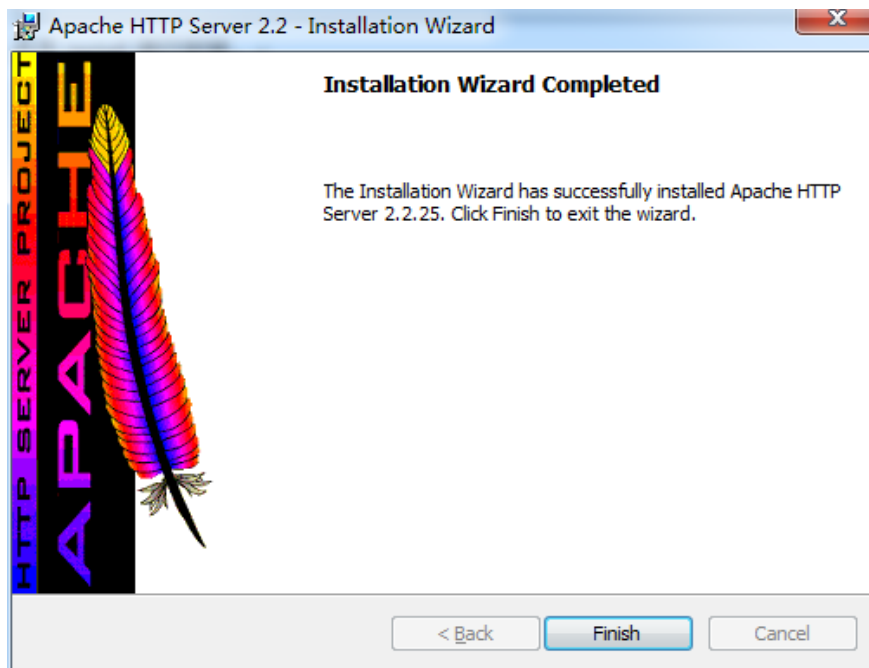
确认安装选项无误，如果您认为要再检查一遍，可以点“Back”一步步返回检查。点“Install”开始按前面设定的安装选项安装。



本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



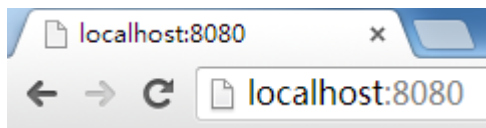
正在安装界面，请耐心等待，直到出现下面的画面。



按“Finish”按钮，结束 Apache 的软件安装。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

打开浏览器，输入 `http://localhost:8080`



It works!

如果看到 It works!，表明 Apache http Server 安装成功。

注意：

如果在 XP 某些版本上不能看到 It works!，请输入如下命令手动安装 Apache http 服务和手动启动 Apache http 服务。

- (1) 点击开始，运行，输入 `cmd`，回车，进入命令行模式；
- (2) 输入 `cd\`回车 切换到根目录；
- (3) 输入 `cd program files` 回车
- (4) 输入 `cd Apache software foundation` 回车
- (5) 输入 `cd Apache2.2` 回车
- (6) 输入 `cd bin` 回车
- (7) 输入 `httpd -k install` 回车，安装 Apache http 服务
- (8) 输入 `httpd -k start` 回车，启动 Apache http 服务

完整操作如下图所示：

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

```
C:\WINDOWS\system32\cmd.exe
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd \

G:\>cd "Program Files"

C:\Program Files>cd "Apache Software Foundation"

C:\Program Files\Apache Software Foundation>cd Apache2.2

C:\Program Files\Apache Software Foundation\Apache2.2>cd bin

G:\Program Files\Apache Software Foundation\Apache2.2\bin>httpd -k install
Installing the Apache2.2 service
The Apache2.2 service is successfully installed.
Testing httpd.conf....
Errors reported here must be corrected before the service can be started.
httpd: Could not reliably determine the server's fully qualified domain name,
ing 172.16.38.111 for ServerName

G:\Program Files\Apache Software Foundation\Apache2.2\bin>httpd -k start
httpd: Could not reliably determine the server's fully qualified domain name,
ing 172.16.38.111 for ServerName

C:\Program Files\Apache Software Foundation\Apache2.2\bin>
```

打开浏览器，输入 <http://localhost:8080>，如果看到 It works!，表明 Apache http Server 安装成功。

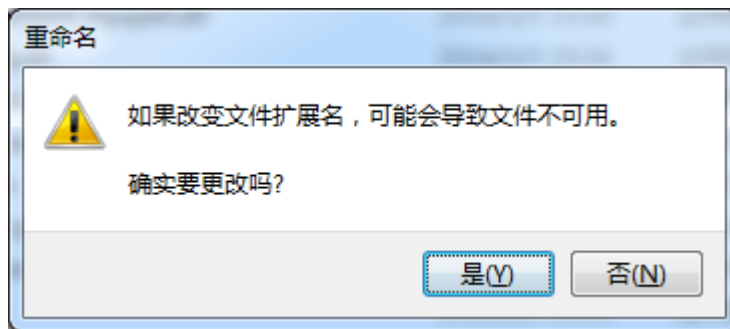
2. 安装 MySQL 数据库服务器

MySQL 数据库服务器安装已经完成。

3. php5.3.28 安装

将文件  [php-5.3.28-Win32-VC9-x86](#) 解压到 D:\php5.3.28 目录，找到 D:\php5.3.28 目录下的 php.ini-development 文件，将其重名为 php.ini，出现对话框，点击“是”。

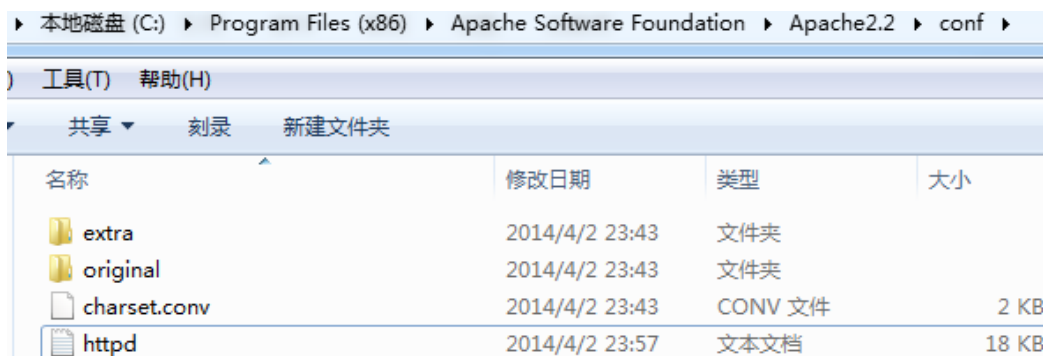
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



Php5. 3. 28 安装完成。

配置 Apache 服务器支持 PHP 文件解析。

在 目 录 C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf 中，找到 httpd.conf 文件，用记事本打开。



在其文件最后面添加如下代码，并保存文件。

```
PHPIniDir "D:/php5.3.28"
```

```
LoadModule php5_module "D:/php5.3.28/php5apache2_2.dll"
```

```
AddType application/x-httpd-php .php .html
```

注意一定要使用"/"，最后一行的.php和.html前均有空格。

#增加以下内容支持PHP脚本

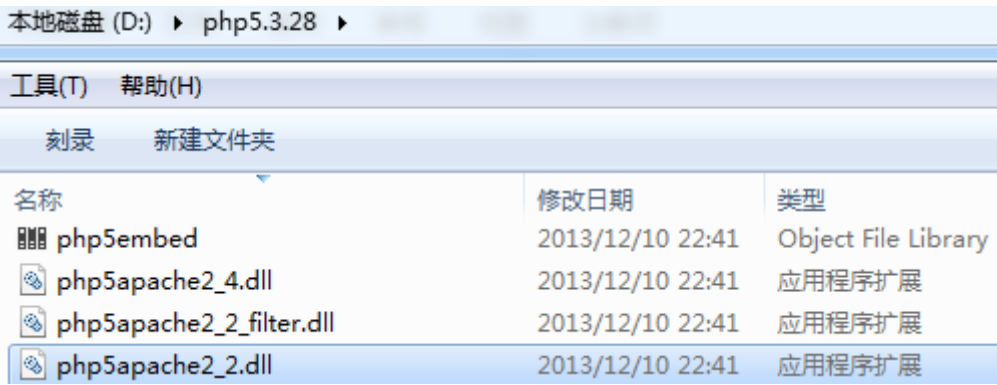
```
PHPIniDir "D:/php5.3.28"
```

```
LoadModule php5_module "D:/php5.3.28/php5apache2_2.dll"
```

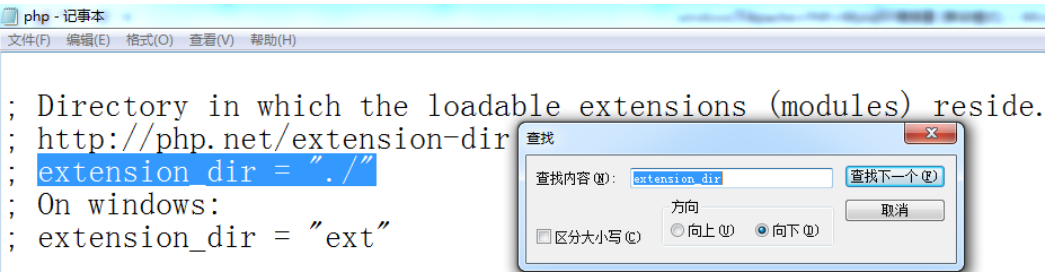
```
AddType application/x-httpd-php .php .html
```

其中的 php5apache2_2.dll 就是在 D:\php5.3.28 目录下，如图所示。

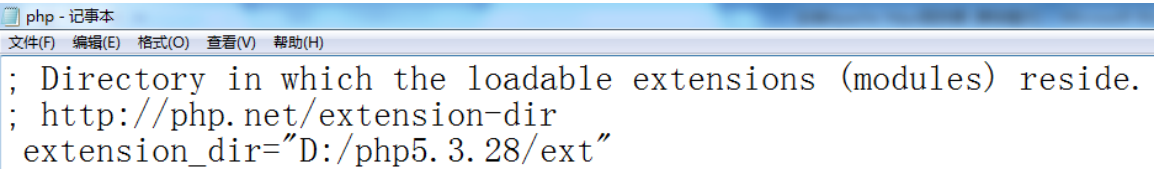
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。




打开 D:\php5. 3. 28 文件夹下的 php. ini 配置文件，查找到 extension_dir，如下图所示位置。



将刚刚查找到的 extension_dir 前面的； 去掉， 并将其设置为 extension_dir="D:/php5. 3. 28/ext"， 如下图所示。



保存 php. ini 文件， 点击右下角， 找到 Apache 2.2  图标， 单击右键， 选择 Open Apache Monitor， 出现 Apache Service Monitor 对话框， 点击 Restart 按钮重启 Apache 让刚刚修改的配置生效， 或者在前面的命令行窗口中， 输入 httpd -k restart 回车， 重新启动 Apache http 服务。 如果有错误， 根据错误信息对前面的配置文件进行修改。

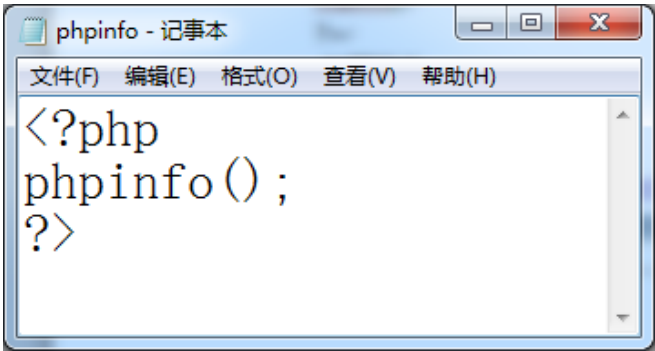
用记事本建立 php 测试文件 phpinfo. php ， 将起放置到 Apache Httpd 的文件 目 录 C:\Program Files (x86)\Apache Software

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

Foundation\Apache2.2\htdocs\下，phpinfo.php 内容如下

```
<?php
Phpinfo();
?>
```

如图所示。



保存文件时，注意文件后缀为 php，不是 txt。此处很容易出错。



在浏览器中打开 <http://localhost:8080/phpinfo.php>

phpinfo0

localhost:8080/phpinfo.php

PHP Version 5.3.28

System	Windows NT 吴思远-AF 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
Build Date	Dec 10 2013 22:26:06
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	escript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mysql" "--without-pdo-mysql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze"
Server API	Apache 2.0 Handler


此时，Apache 已经支持 PHP 了。

配置 PHP 扩展支持 MySQL 数据库。

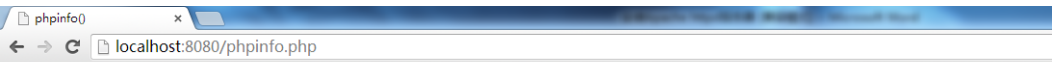
用 记 事 本 打 开 php5.3.28 之 下 的 php.ini 文 件 ，
把;extension=php_mysql.dll 和;extension=php_mysqli.dll 之前的;去掉。

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

;extension=php_mbstring.dll
;extension=php_exif.dll
extension=php_mysql.dll
extension=php_mysqli.dll
;extension=php_oci8.dll
```

保存 php.ini 文件，点击右下角，找到 Apache 2.2  图标，单击右键，选择 Open Apache Monitor，出现 Apache Service Monitor 对话框，点击 Restart 按钮重启 Apache 让刚刚修改的配置生效。

此时的 PHP 已经支持 MySQL 了。在浏览器中打开 <http://localhost:8080/phpinfo.php>



mysql

MySQL Support	enabled	
Active Persistent Links	0	
Active Links	0	
Client API version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 731e5b87ba42146a687c29995d2af8b4e40b325 \$	

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	127.0.0.1	127.0.0.1
mysql.default_password	no value	no value
mysql.default_port	3306	3306
mysql.default_socket	no value	no value
mysql.default_user	root	root
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

mysqli

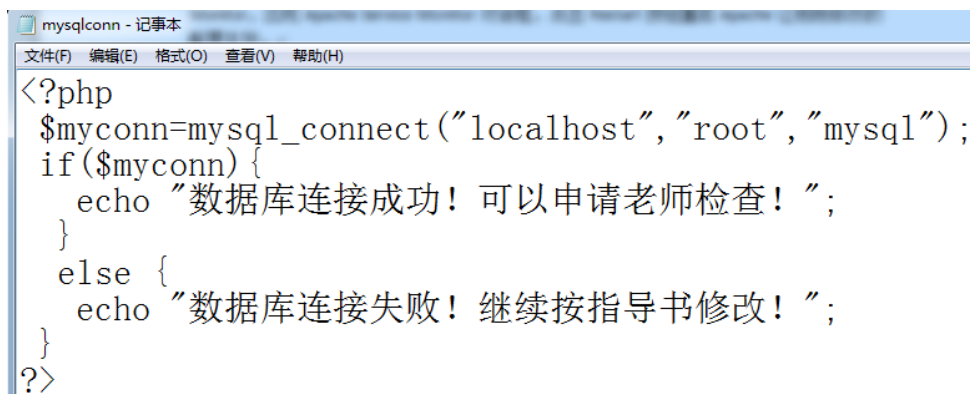
MySQLi Support		enabled	
Client API library version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 731e5b87ba42146a687c29995d2af8b4e40b325 \$		
Active Persistent Links	0		
Inactive Persistent Links	0		
Active Links	0		

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	127.0.0.1	127.0.0.1
mysqli.default_port	3306	3306

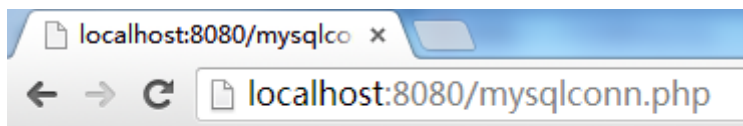
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

用记事本建立 php 测试文件 mysqlconn.php，将起放置到 Apache Httpd 的文件目录 C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs\下，mysqlconn.php 内容如下：

```
<?php
$myconn=mysql_connect("localhost","root","mysql");
if($myconn){
    echo "数据库连接成功！可以申请老师检查！";
}
else {
    echo "数据库连接失败！继续按指导书修改！";
}
?>
```



在浏览器中打开 <http://localhost:8080/mysqlconn.php>



数据库连接成功！可以申请老师检查！

六、实验步骤：

完成以上软件的安装与配置，并看到成功信息。

七、思考与练习

- 1、Apache HTTP Server 对 html 和 php 的支持有何差别？

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

2、PHP 如果要支持 MS SQL SERVER 和 Oracle 数据库，需要如何配置？

八、实验报告要求

参见数据库实验报告模版

参考文献

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验五 MySQL 的存储过程

一、实验目的

存储过程是一组为了完成特定功能的 SQL 语句集，经编译后存储在数据库中。存储过程分为两类：1. 系统提供的存储过程；2. 用户自定义存储过程。

存储过程具有的优点：1. 存储过程允许标准组件式编程；2. 存储过程能够实现较快的执行速度；3. 存储过程能够减少网络流量；4. 存储过程可被作为一种安全机制来充分利用。存储过程是可复用的组件！想象一下如果你改变了主机的语言，这对存储过程不会产生影响，因为它是数据库逻辑而不是应用程序。存储过程是可以移植的！

本次实验了解 MySQL 存储过程的创建、修改和删除的方法和步骤，掌握在 Navicat for MySQL 中对存储过程的进行创建、修改和删除，掌握在 MySQL Command Line Client 中调用带参数和不带参数的存储过程。

二、实验要求

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，实验时注意每种 SQL 语句的基本命令及各个关键字的含义，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点

1、重点：

- (1) 存储过程的创建、修改和删除。
- (2) 存储过程的调用。

2、难点：

MySQL 的分隔符定义

存储过程的创建

带参数的存储过程调用。

四、仪器设备及用具

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：本机已安装 MySQL 5.X 数据库平台。

五、教学过程

（一）实验预习

（1）熟悉 SQL 中的 CREATE PROCEDURE 语句的格式及所用的关键字含义及用法。

（2）掌握存储过程创建、修改和删除的一般方法。

（3）掌握存储过程调用的一般方法。

（二）实验原理


在 Navicat for MySQL 中查询数据库版本，创建示例数据库库、表、数据。在 MySQL Command Line Client 使用 CREATE PROCEDURE 语句创建存储过程，对存储过程的修改、删除等操作。在 MySQL Command Line Client 中调用存储过程。

（三）实验内容

1. 创建及调用不带参数的存储过程：

（1）双击桌面 Navicat for MySQL 快捷方式，连接到 mysql 数据库上。点击查询->新建查询，输入如下语句，确定 MySQL 数据库版本：

```
Select VERSION();
```

然后点击  按钮，运行查询语句。如图 8-1 所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

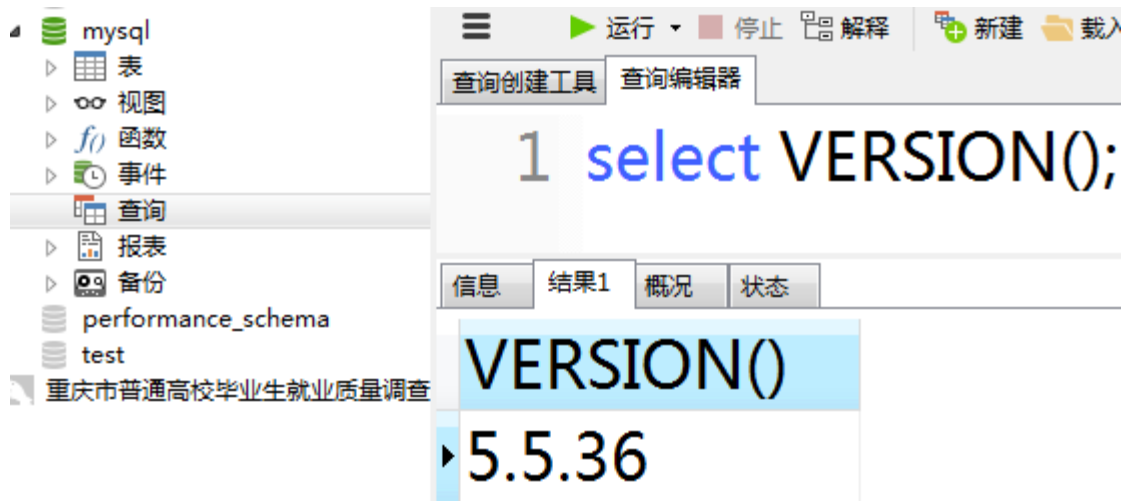



图 8-1

输入如下语句，创建示例数据库：

```
CREATE DATABASE db5536;
```

```
USE db5536;
```

然后点击  按钮，运行查询语句。如图 8-2 所示。

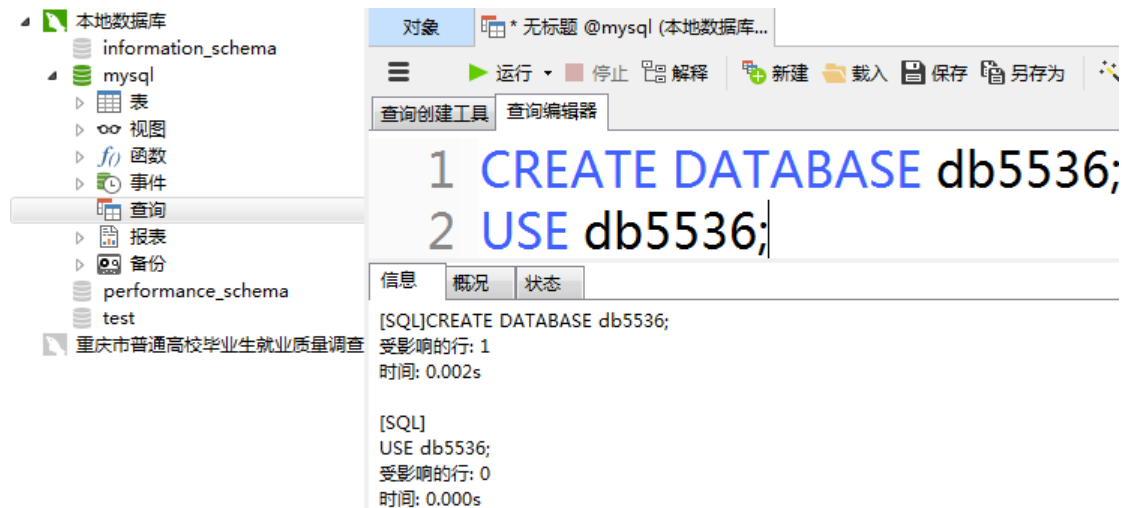


图 8-2

选择本地数据连接，单击右键，选择刷新，如图 8-3 所示。

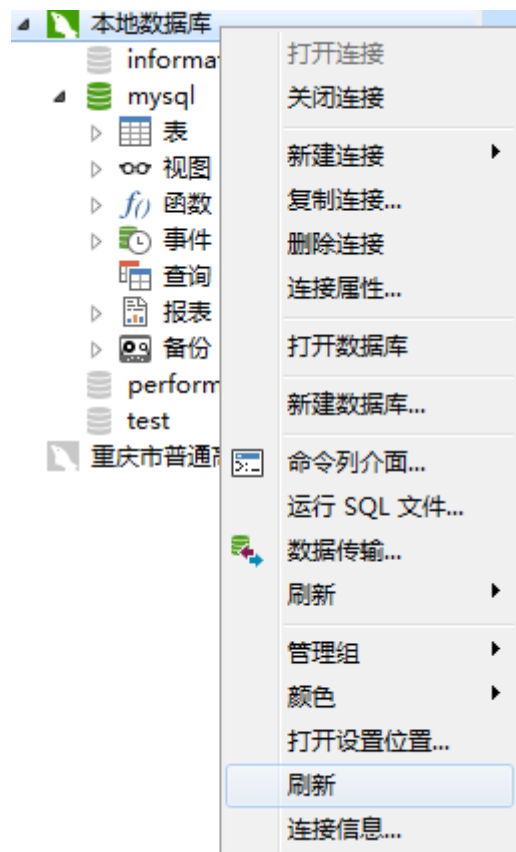


图 8-3

刷新后，新建立的 db5536 出现在数据库列表中，**双击 db5536 连接到该数据库上**，以后的操作都在 db5536 上进行，如图 8-4 所示。

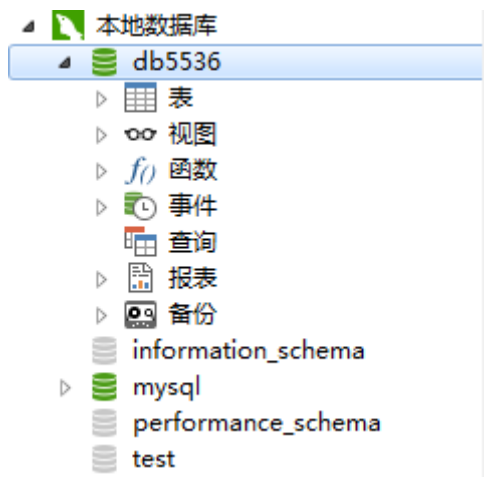



图 8-4

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

点击查询->新建查询，输入如下语句，创建示例表和插入演示数据：

```
CREATE TABLE cqupt (s1 INTEGER);
```

```
INSERT INTO cqupt VALUES (5);
```

然后点击  按钮，运行查询语句。如图 8-5 所示。

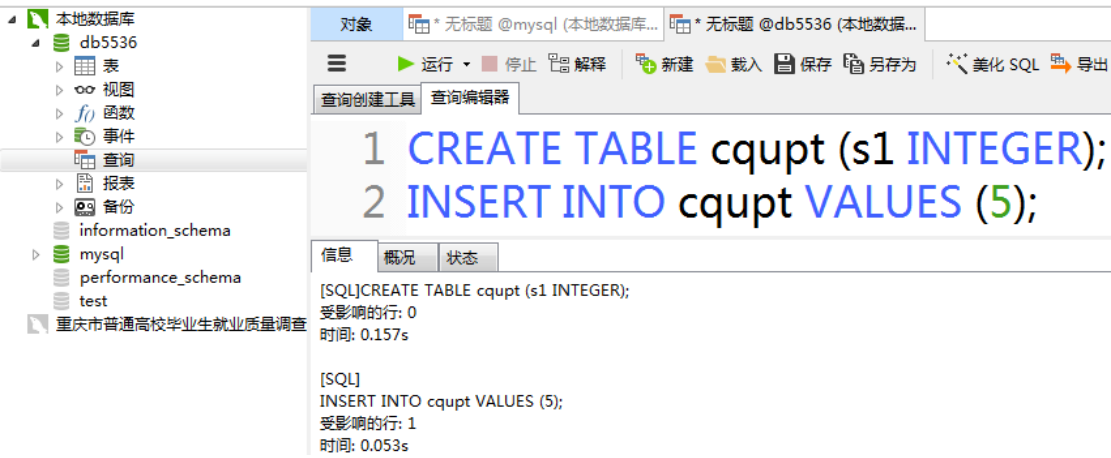


图 8-5

(2)MySQL Command Line Client 中创建存储过程：

可以运行 MySQL 的命令行客户端软件，如图 8-6 所示。

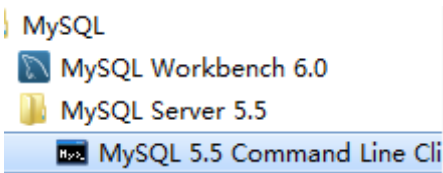


图 8-6

在登录界面输入密码 mysql，如图 8-7 所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

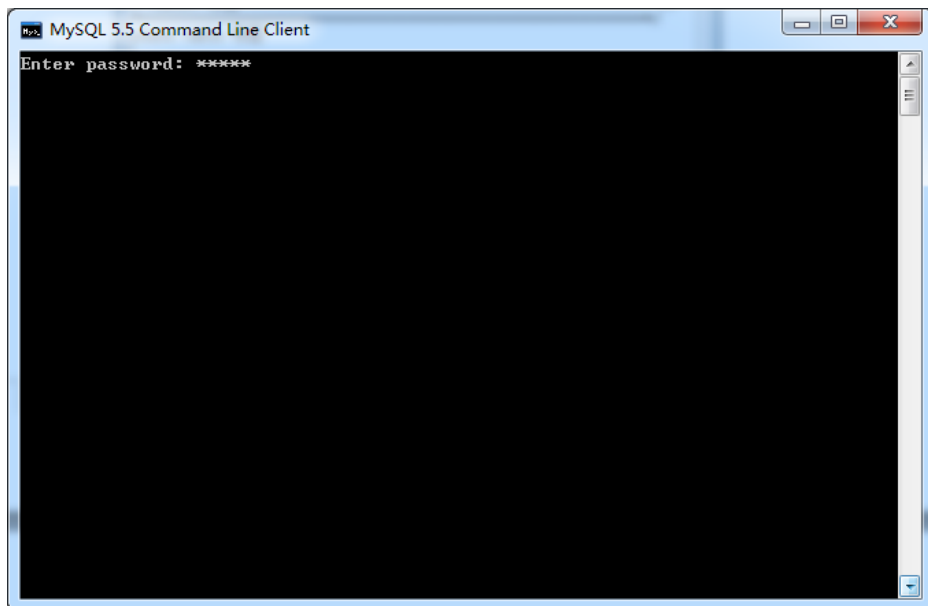


图 8-7

使用 Use db5536; 语句打开 db5536 数据库，
使用 DELIMITER // 语句设置 // 为新的分割符（**DELIMITER 与 // 之间有一个空格**），如图 8-8 所示。

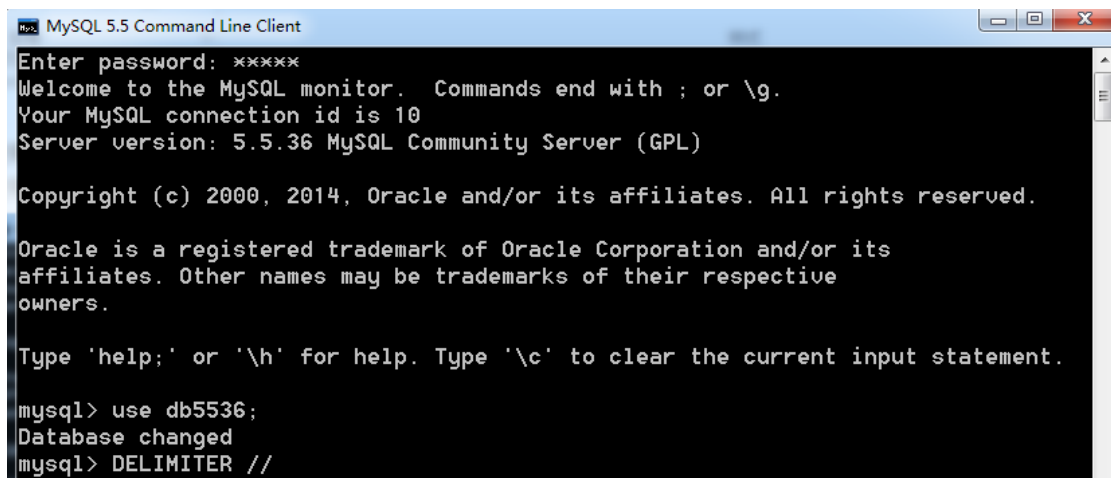


图 8-8

输入如下语句，创建第一个存储过程，注意引号为英文半角状态下单引号：

```
Create procedure first_p()  
Select 'Hello MySQL, This is my first procedure'  
//
```

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

创建成功，如图 8-9 所示。

```
mysql> Create procedure first_p()  
-> Select 'Hello MySQL,This is my first procedure '  
-> //  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> _
```

图 8-9

输入如下语句，创建第二个存储过程,注意引号为英文半角状态下单引号：

```
Create procedure second_p()  
Select current_date from cqupt  
//
```

创建成功，如图 8-10 所示。

```
mysql> Create procedure second_p()  
-> Select current_date from cqupt  
-> //  
Query OK, 0 rows affected (0.00 sec)
```

图 8-10

(3)MySQL Command Line Client 中调用存储过程：

使用如下命令 CALL 存储过程名字，对刚才建立的第一个存储过程进行调用，如图 8-11 所示：

CALL first_p()//

```
mysql> CALL first_p()//  
+-----+  
| Hello MySQL,This is my first procedure |  
+-----+  
| Hello MySQL,This is my first procedure |  
+-----+  
1 row in set (0.03 sec)  
  
Query OK, 0 rows affected (0.04 sec)
```

图 8-11

使用如下命令 CALL 存储过程名字，对刚才建立的第二个存储过程进行调用，

如图 8-12 所示：

CALL second_p() //

```
mysql> CALL second_p()//
+-----+
| current_date |
+-----+
| 2014-04-21   |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

图 8-12

2. 创建及调用带参数的存储过程。

(1)MySQL Command Line Client 中创建存储过程：

输入如下语句，创建带输出参数的第三个存储过程，结果如图 8-13 所示：

```
CREATE PROCEDURE simpleproc (OUT param1 INT)
SELECT COUNT(*) INTO param1 FROM cqupt
//
```

```
mysql> CREATE PROCEDURE simpleproc (OUT param1 INT)
-> SELECT COUNT(*) INTO param1 FROM cqupt
-> //
Query OK, 0 rows affected (0.00 sec)
```

图 8-13

输入如下语句，创建带输入参数的第四个存储过程，结果如图 8-14 所示：

```
CREATE PROCEDURE simpleproc1 (IN param1 INT)
SET @x = param1 - 100
//
```

```
mysql> CREATE PROCEDURE simpleproc1 (IN param1 INT)
-> SET @x = param1 - 100
-> //
Query OK, 0 rows affected (0.03 sec)
```

图 8-14

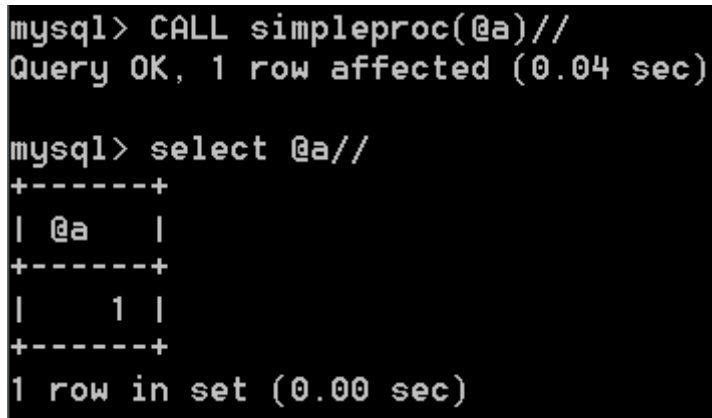
(2) MySQL Command Line Client 中调用存储过程：

输入如下语句，调用第三个带传出参数的存储过程，结果如图 8-15 所示：

```
CALL simpleproc(@a) //
```

```
Select @a //
```

上述语句的@a 为定义的变量



```
mysql> CALL simpleproc(@a) //  
Query OK, 1 row affected (0.04 sec)  
  
mysql> select @a //  
+-----+  
| @a    |  
+-----+  
|      1 |  
+-----+  
1 row in set (0.00 sec)
```

图 8-15

输入如下语句，调用第四个带传入参数的存储过程，结果如图 8-16 所示：

```
CALL simpleproc1(5500) //
```

```
Select @x //
```

上述语句的@x 为定义的变量

```
mysql> CALL simpleproc1(5500)//
Query OK, 0 rows affected (0.00 sec)

mysql> select @x//
+-----+
| @x    |
+-----+
| 5400  |
+-----+
1 row in set (0.00 sec)
```

图 8-16

(四) 注意事项

- 1、DELIMITER //是用于设置在创建存储过程时，出现的多行语句问题的特殊分隔符。完成创建后，应该使用 DELIMITER ;语句，将分隔符设置回分号。
- 2、调用 MySQL 存储过程时，应使用如@a, @x 这样的变量来完成参数的传入和传出。

六、实验步骤：

执行以下内容：

1. 启动 Navicat for MySQL，在 MySQL - 新建连接中完成连接参数配置。
2. 登录到本地数据库服务器后，连接到 db5536 数据库上。
3. 执行五、教学过程中（三）实验内容的 1 部分的（2）、（3）步骤创建及调用不带参数的存储过程，验证不带参数的存储过程的创建和调用方法。
4. 在步骤 3 的基础上，创建存储过程 third_p()，显示表 cqupt 中的数据，并对其进行调用测试。结果如图所示：

```
mysql> call third_p()//
+-----+
| s1    |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

5. 执行五、教学过程中（三）实验内容的第 2 部分创建及调用带参数的存储过程，验证带参数的存储过程的创建和调用方法。

6. 在步骤 5 的基础上，创建存储过程 simpleproc2(), 对传入参数进行乘以 10 的处理，变量定义@x, 并对其进行调用测试。结果如图所示：

```
mysql> CALL simpleproc2(50)//
Query OK, 0 rows affected (0.00 sec)

mysql> select @x//
+-----+
| @x    |
+-----+
|  500  |
+-----+
1 row in set (0.00 sec)

mysql> CALL simpleproc2(0)//
Query OK, 0 rows affected (0.00 sec)

mysql> select @x//
+-----+
| @x    |
+-----+
|     0 |
+-----+
1 row in set (0.00 sec)
```

7. 用 Drop Procedure 语句删除存储过程 first_p。如图所示：

```
mysql> drop procedure first_p//
Query OK, 0 rows affected (0.01 sec)
```

以上 SQL 语句请保存，以备老师检查和写实验报告之用。

七、思考与练习

1、如何修改存储过程？

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

2、如果雇员数据重复插入，如何修改存储过程来正确显示结果？

八、实验报告要求

参见数据库实验报告模版

参考文献

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验六 MySQL 的触发器

一、实验目的

触发器是特定事件出现的时候，自动执行的代码块，类似于存储过程。触发器与存储过程的区别在于：存储过程是由用户或应用程序显式调用的，而触发器是不能被直接调用的，由 DBMS 自动执行。

触发器具有这些功能：1、允许/限制对表的修改；2、自动生成派生列，比如自增字段；3、强制数据一致性；4、提供审计和日志记录；5、防止无效的事务处理；6、启用复杂的业务逻辑。

本次实验了解 MySQL 触发器的创建、修改和删除的方法和步骤，掌握在 Navicat for MySQL 中对触发器的进行创建、修改和删除，掌握触发器触发测试的方法。

二、实验要求

- 1、实验前：预习实验内容，学习相关知识。
- 2、实验中：按照实验内容要求进行实验，实验时注意每种 SQL 语句的基本命令及各个关键字的含义，做好实验记录。
- 3、实验后：分析实验结果，总结实验知识，得出结论，按格式写出实验报告。
- 4、在整个实验过程中，要独立思考、独立按时完成实验任务，不懂的要虚心向教师或同学请教。
- 5、要求按指定格式书写实验报告，且报告中应反映出对本次实验的总结，下次实验前交实验报告。

三、实验的重点与难点

1、重点：

- (1) 触发器的创建、修改和删除。
- (2) 触发器的触发测试。

2、难点：

触发器的定义。

触发器的触发测试。

四、仪器设备及用具

硬件：投影仪、每位同学分配已连接校园网 PC 机一台。

软件：数据库服务器已安装 MySQL 5.X 数据库平台。

五、教学过程

（一）实验预习

（1）熟悉 SQL 中的 CREATE TRIGGER 语句的格式及所用的关键字含义及用法。

（1）掌握触发器创建、修改和删除的一般方法。

（2）掌握触发器测试的一般方法。


（二）实验原理

在 Navicat for MySQL 中使用 CREATE TRIGGER 语句创建存储过程，及在 Navicat for MySQL 中对触发器的修改、删除等操作。在 Navicat for MySQL 中测试触发器是否工作正常。

（三）实验内容

1. 创建触发器 audit_student，该触发器对修改 student 表的时间、用户名记录到日志表中。

双击桌面 Navicat for MySQL 快捷方式，**连接到 test 数据库上(本次实验如果连接到 mysql 数据库上，无法正常完成)**，按顺序运行以下语句，然

后点击  按钮，运行 SQL 语句。

（1）. 输入如下语句，建立实验表 student，如图 9-1 所示：

```
create table student
(SNO varchar(10),
SNAME varchar(30),
Sage Int);
```

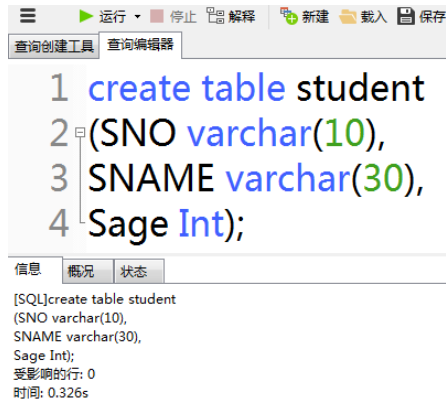


图 9-1

注意，语句前面的行号 1、2、3、4 为自动生成的，不需要手工录入。

(2). 输入如下语句，建立日志表 student_log，并查看日志表是否为空，如图 9-2 所示：

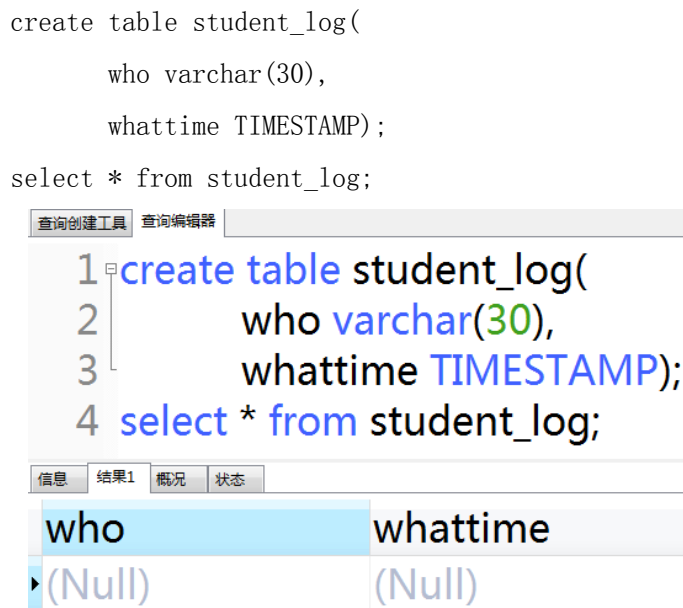


图 9-2

我们看到，现在的 student_log 表中没有记录，全部为 Null 值。

(3). 输入如下语句，在 student 表上建立语句触发器，在触发器中将时间、用户记录到 student_log 表中。红色 Before 表示检测的时间为 insert 操作前，也可以为 After，表示在操作后。如图 9-3 所示：

```
use test;
```



```
Create trigger audit_student
Before insert On student
For each row
Begin
    Insert into student_log(Who,whattime) Values
(current_user,current_timestamp);
End;
```

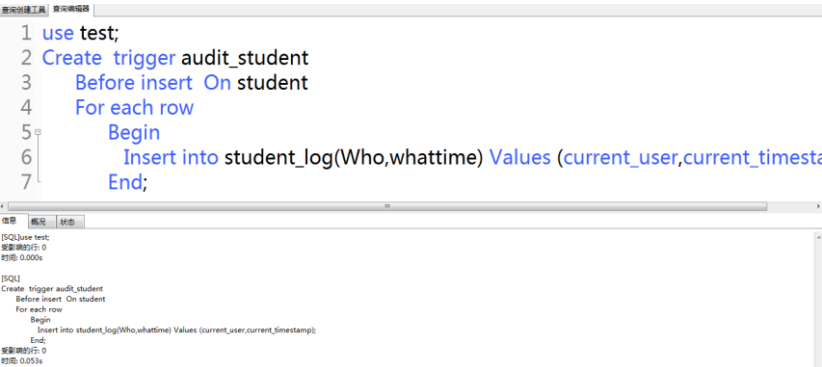


图 9-3

选中 student 表，单击右键，选择设计表，点击触发器选项卡，如图 9-4 所示：

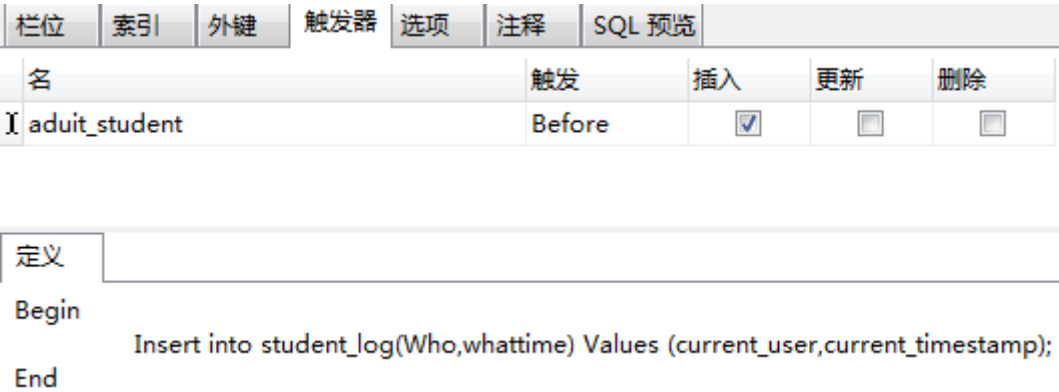


图 9-4

我们可以看到，在 INSERT 操作前，会触发触发器，触发器执行的脚步就是向 student_log 表中，写入触发的用户和时间戳。

- (4). 输入如下语句对触发器进行 INSERT 触发测试，如图 9-5 所示：
- 对 student 表做一个 INSERT 操作：

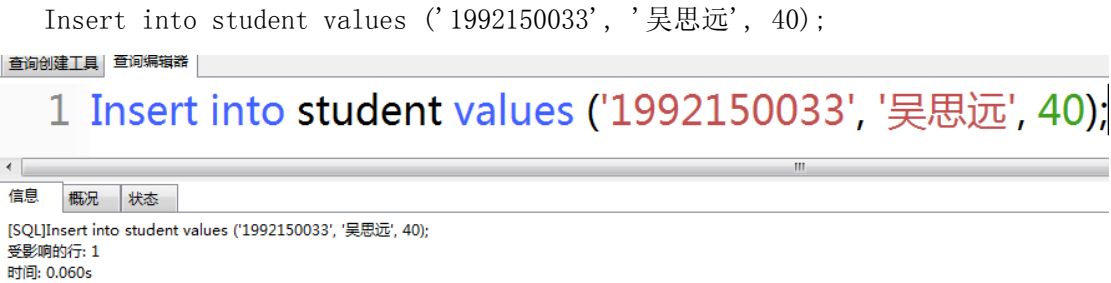


图 9-5

输入如下语句查看表 student_log, 检查触发器工作结果，表 student_log 中数据如图 9-6 所示：

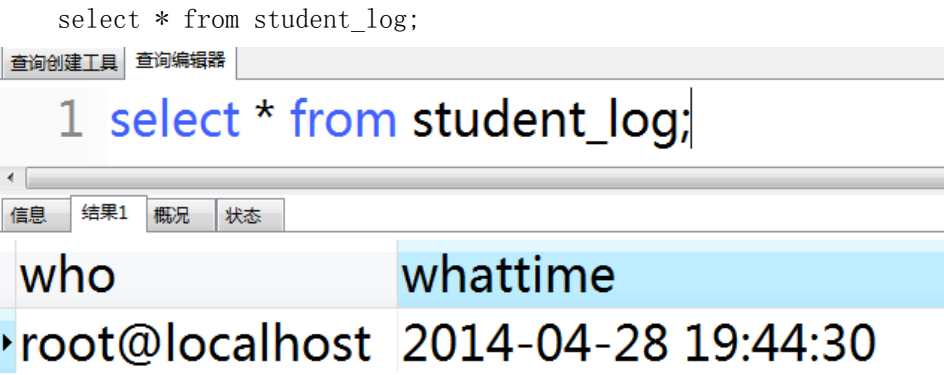


图 9-6

我们看到，执行插入语句的是 student 表，但现在的 student_log 表中由触发器自动生成了 1 条记录。

2. 是哪个 DML 操作起作用的呢？要搞清楚这个问题，我们必须在前面触发器的基础上进行改进。

(1). 输入如下语句，修改日志表结构，增加一个属性存放触发语句类型。如图 9-7 所示：

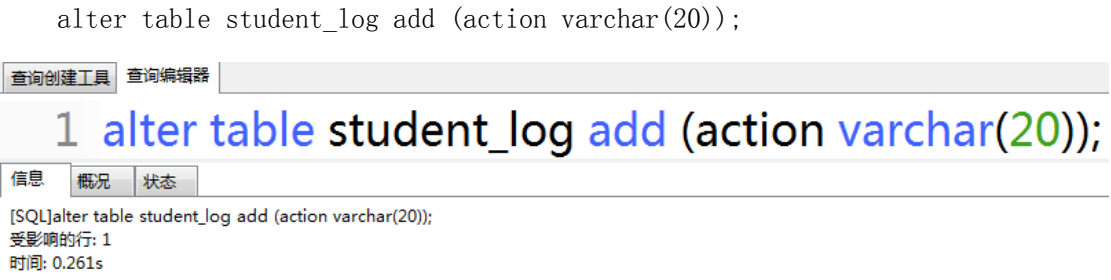


图 9-7

(2) 修改触发器，记录触发语句类型。选中 student 表，单击右键，选择设计表，点击触发器选项卡，在定义中对 INSERT 进行编辑, 如图 9-8 所示：

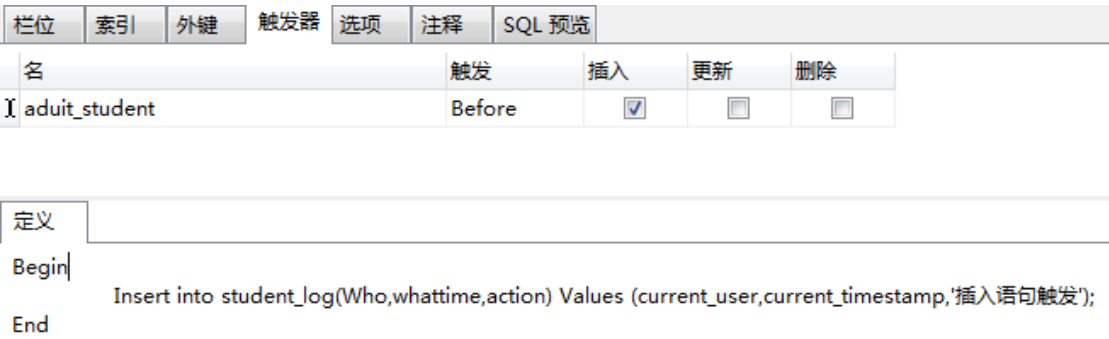



图 9-8

修改完毕后，点击  按钮，对触发器进行保存。

(3). 输入如下语句对触发器进行 INSERT 触发测试，如图 9-9 所示：

用你自己的学号、姓名、年龄，对 student 表再做一个 insert 操作：

Insert into student values ('2012XXXXXXX', 'XXXXX', 20);

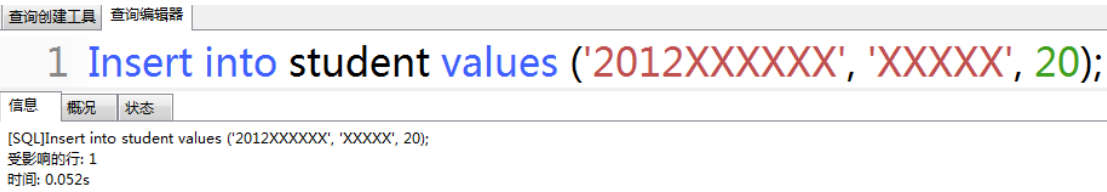


图 9-9

输入如下语句检查触发器工作结果，查看表 student_log，表 student_log 中数据如图 9-10 所示：

select * from student_log

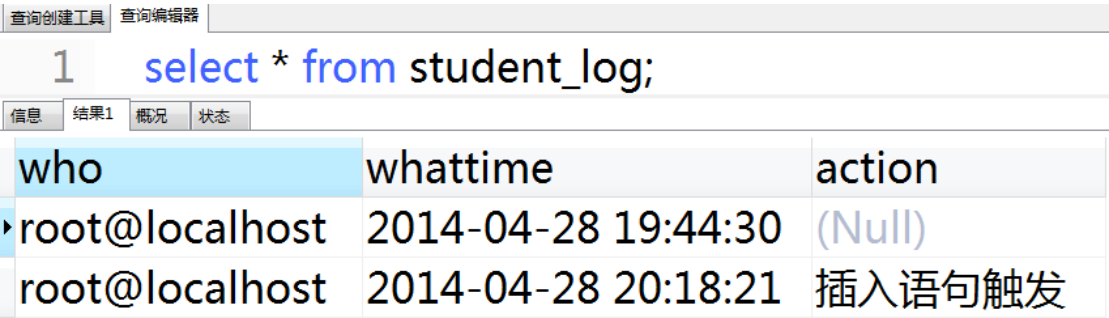


图 9-10

我们看到，执行插入语句的是 student 表，但现在的 student_log 表中由触发器再次自动生成了 1 条记录。

触发器使用完毕后，可以使用 `drop trigger audit_student;` 将其删除。

（四）注意事项

1、在修改能检测触发类型的触发器前，应先修改表 `alter table student_log add (action varchar2(20));`

六、实验步骤：

执行以下内容：

1. 启动 Navicat for MySQL，在 MySQL - 新建连接中完成连接参数配置。
2. 登录到本地数据库服务器后，连接到 test 数据库上。
3. 执行五、教学过程中（三）实验内容的第一部分创建简单触发器，验证触发器的创建和触发器的触发测试。
4. 执行五、教学过程中（三）实验内容的第二部分改造简单触发器，增加触发类型判断功能，验证触发器的改造和改造后的触发器触发测试。
5. 自行编写 student 表的触发器 `aduit_student_update`，在对 student 表进行更新操作**前**触发，触发时，向 student_log 表写入触发用户、触发时间和触发类型。并对该新建进行触发测试。
6. 自行编写 student 表的触发器 `aduit_student_delete`，在对 student 表进行删除操作**后**触发，触发时，向 student_log 表写入触发用户、触发时间和触发类型。并对该新建进行触发测试。

以上 SQL 语句请保存，以备老师检查和写实验报告之用。

七、思考与练习

- 1、如何删除触发器？
- 2、如何启用和禁用触发器？

八、实验报告要求

参见数据库实验报告模版

参考文献

- [1] 《数据库原理与设计》，王国胤等，电子工业出版社，2011
- [2] 《数据库系统及应用》，崔巍，高等教育出版社，2005

实验七 PowerDesigner 的 CDM 应用

一、实验目的

数据建模把设计分成如下两个级别：

概念级，CDM 代表业务信息需求，不考虑在数据库上的物理实现；物理级，PDM指定了物理实现的具体细节，这些细节包括：目标RDBMS的特征、改善性能的措施、数据库的可用性。

本次实验中，我们重点介绍 CDM 的使用。

概念数据模型（CDM）设计是建模过程的关键阶段，此阶段把现实世界中需要保存的信息抽象成信息世界中的实体和联系，产生实体联系图。这一阶段可以为高质量的应用提供坚实的基础。

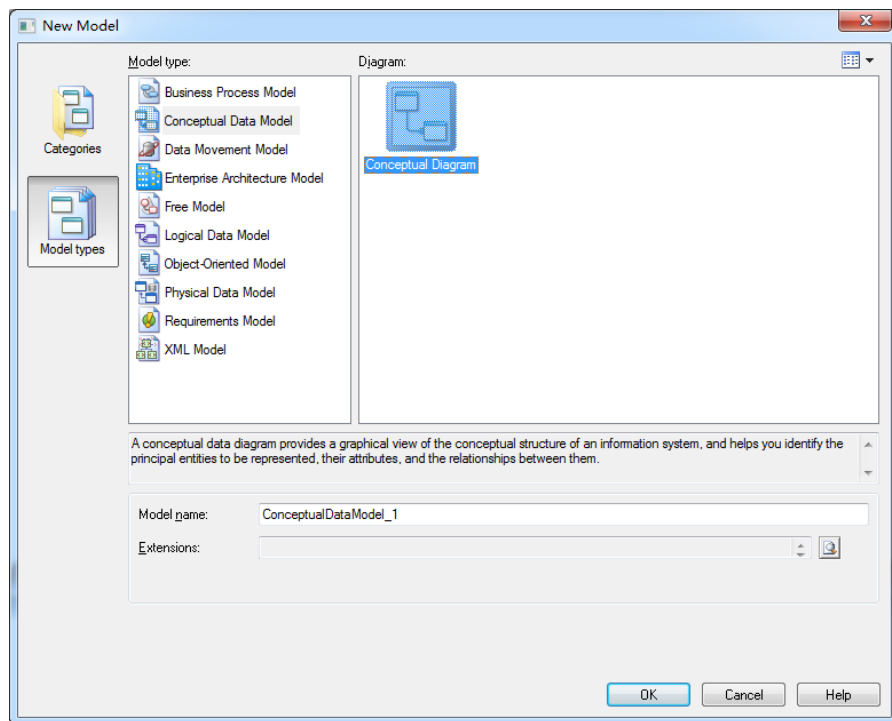
建立概念数据模型是一项综合性的工作。这项工作应由具有业务领域知识的专家和数据模型专家参加。业务领域专家和数据模型专家共同合作，完成一个清晰的、包括全部业务问题的综合文本，然后由数据模型专家把这些原始数据转化成数据流程图和概念数据模型。

本次实验熟悉PowerDesinger的CDM模块的使用，完成这个实验后，应该能够熟悉PowerDesinger的CDM工作区，使用工具选项板进行CDM对象设计。

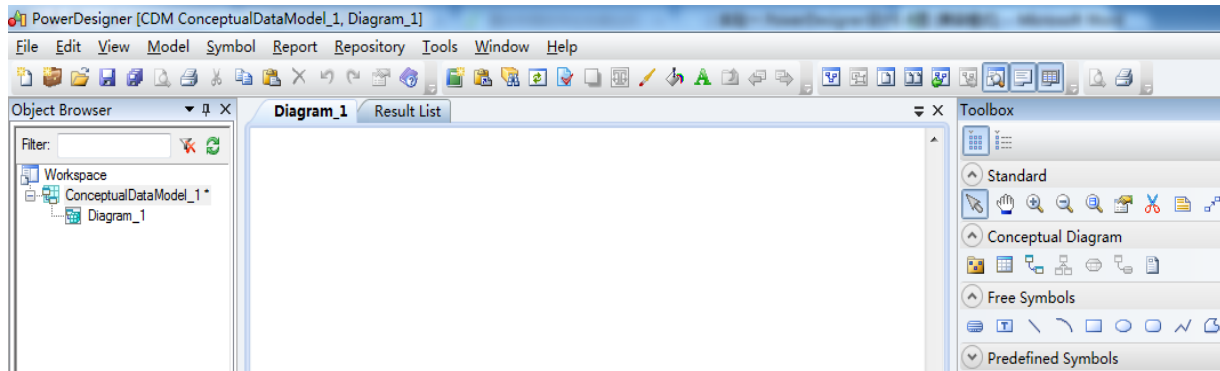
二、实验内容

启动 PowerDesigner，选择“File/New Model”命令，在弹出的对话框中选择 Model types→Conceptual Data Model(概念数据模型)→Conceptual Diagram，如下图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

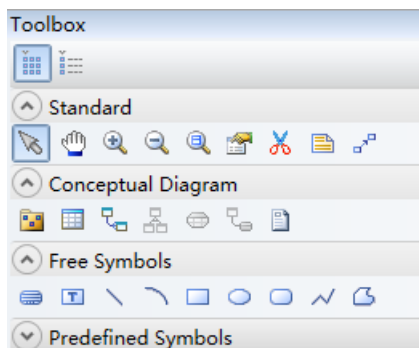


单击“OK”按钮，进入概念模型设计界面。

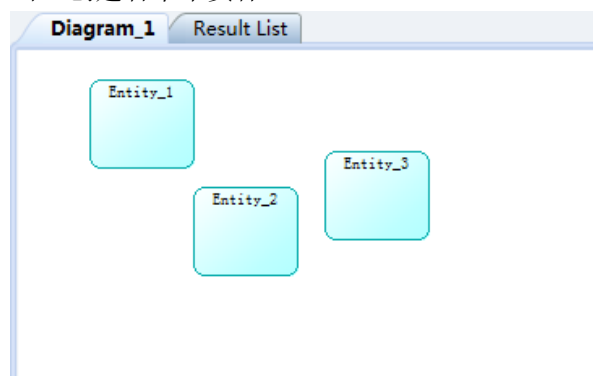



概念设计阶段是现实世界到信息世界的第一层抽象，通过对现实世界的具体事物抽象、组织为数据模型，确定实体、属性及它们之间的联系。概念数据模型是对实体和实体间的关系的定义(即数据库的逻辑模型)，是独立于数据库和数据库管理系统的。

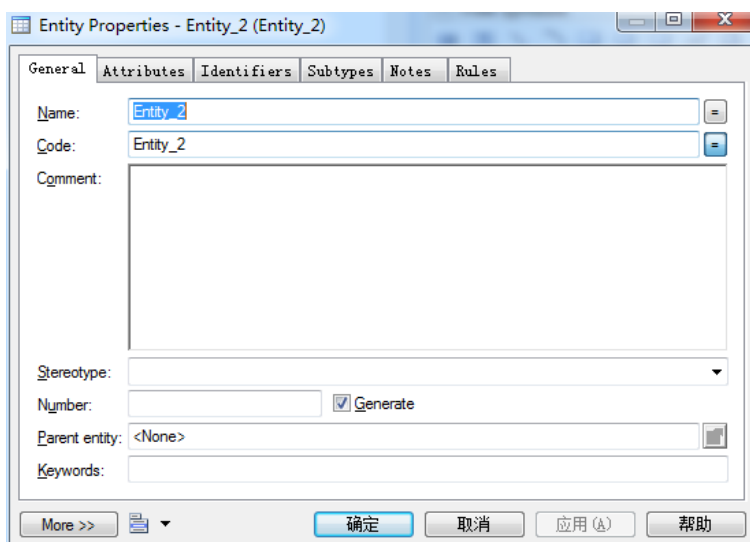
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



单击 Toolbox 上 Conceptual Diagram 中的实体 ，然后在 Diagram_1 的画布上分别单击，创建若干个实体。

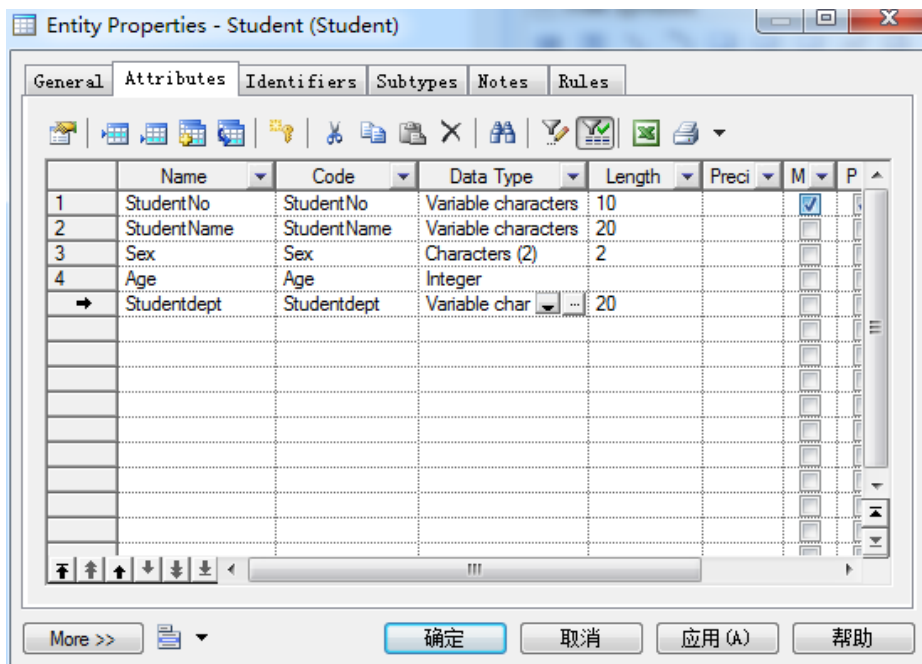


创建实体后，单击 Toolbox 上 Standard 的光标图标 ，设置实体的属性。双击刚才创建的实体，如“Entity_2”，弹出“实体属性”对话框。

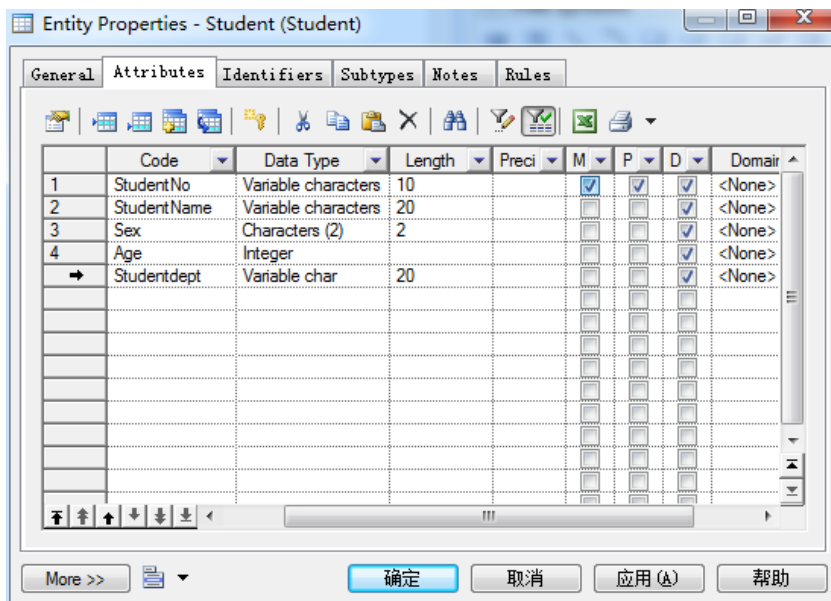


本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

继续设置“StudentName”属性为 Variable Character 类型，长度为 20 位；“Sex”属性为 Character 类型，长度为 2 位；“Age”属性为 Integer；“Studentdept”属性为 Variable Character 类型，长度为 20 位，如下图所示。单击“确定”按钮。

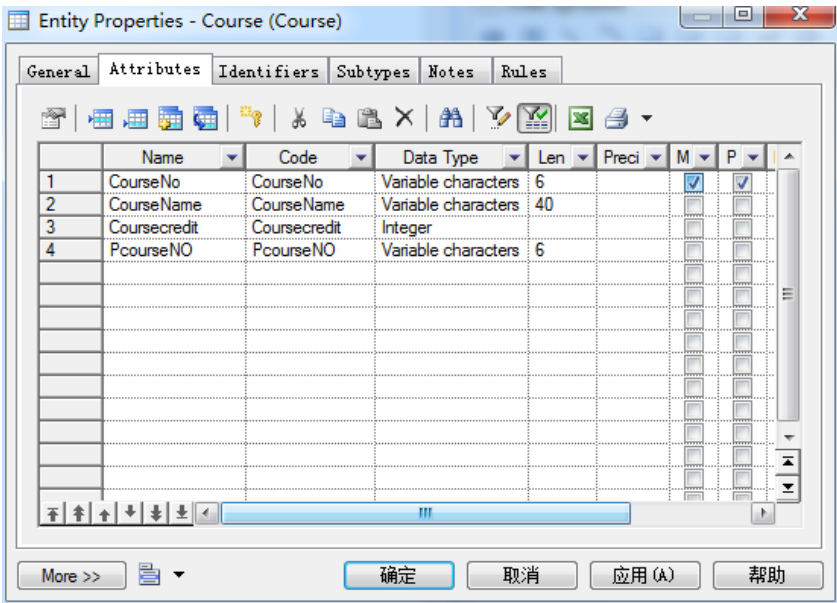


设置实体主码的方法是，向右拖动最下边的滚动条，能看到 M、P、D 三个带有复选框的列。例如：设置“StudentNO”为主码，则在其后边的 P 复选框中打勾，如下图所示。

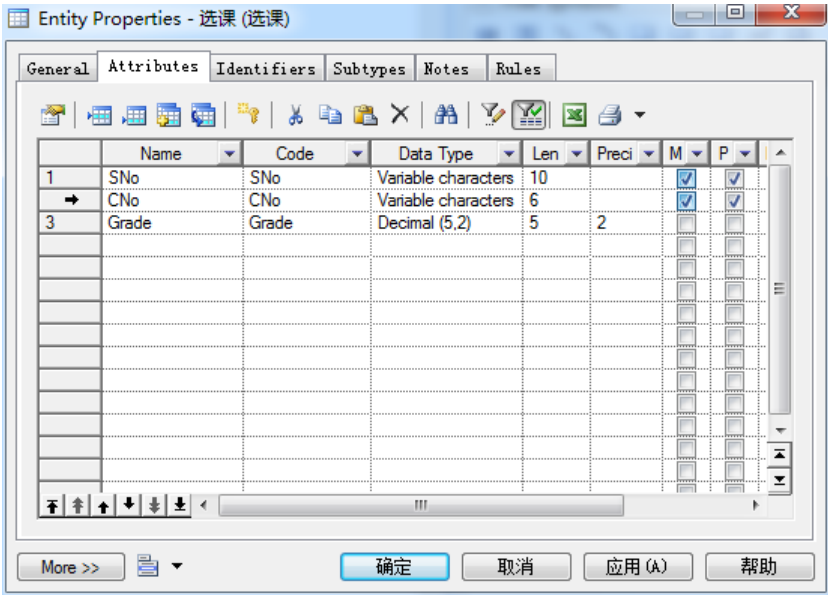


本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

按前面介绍的方法，创建课程实体及其属性，设置“CourseNO”为主码，设置完成后，如下图所示。




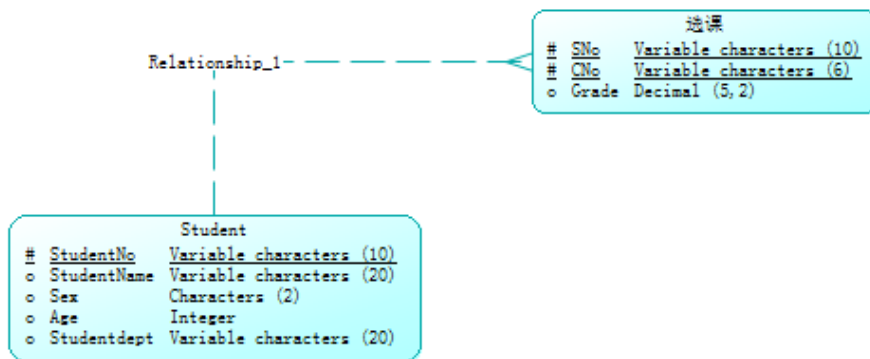
继续创建选课实体及其属性，设置“SNO”、“CNO”为主码，设置完成后，如下图所示。




实体定义完成后，接下来创建实体间的关系。单击 Toolbox 上 Conceptual

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

Diagram 中的 ，在实体 Student 上单击鼠标左键，然后拖至另外一个实体选课，结果如下图所示。



创建关系后，单击面板上的光标图标 ，进一步设置实体间的关系。例如设置“Student”实体和“选课”实体的关系，双击关系“Relationship_1”，弹出“关系属性”对话框，如下图所示。在“常规”选项卡中，修改关系的名称为“选课”，在“cardinalities”中选择“One-Many”，表示“Student”实体和“选课”实体之间是一对多的对应关系。

The dialog box is titled 'Relationship Properties - 选课 (选修)'. It shows the relationship between 'Entity 1' (Student) and 'Entity 2' (选课). The 'Cardinalities' tab is selected, showing the relationship 'Student to 选课' with a cardinality of '0..n' and '选课 to Student' with a cardinality of '0..1'. The 'General' tab shows the relationship name '选课' and the cardinalities 'One - many' and 'Many - one'.

Entity 1: Student

Entity 2: 选课

Relationship: Relationship_1

General

Each Student may have one or more 选课.

Each 选课 may have at most one Student.

Cardinalities

One - one ☒ One - many ☐ Many - one ☐ Many - many ☐

Dominant role: <None>

Student to 选课


Role name:

☐ Dependent ☒ Mandatory Cardinality: 0..n

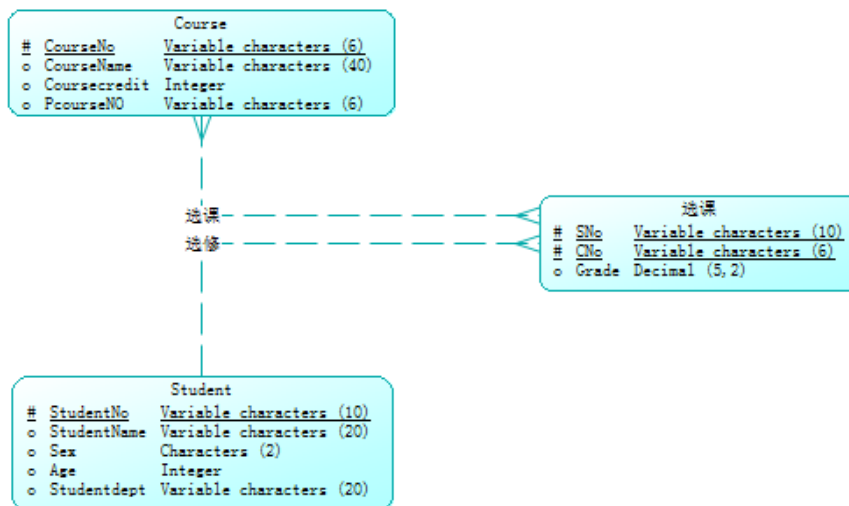
选课 to Student


Role name:

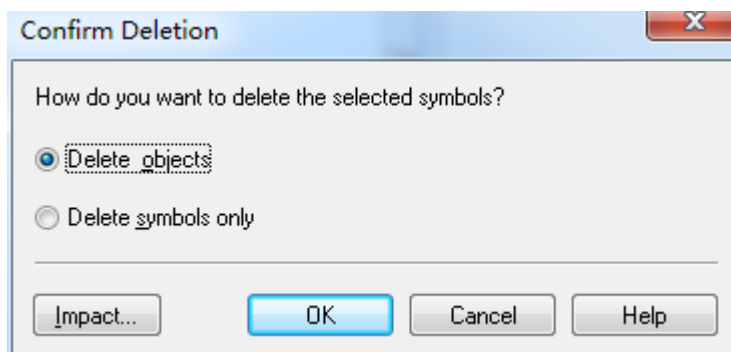
☐ Dependent ☒ Mandatory Cardinality: 0..1

More >>  确定 取消 应用 (A) 帮助

按前面介绍的方法，设置实体 Course 和实体选课间的关系“选课”，联系类型为“Many-many”。至此，概念模型设计工作完成。设计好的概念数据模型如下图所示。



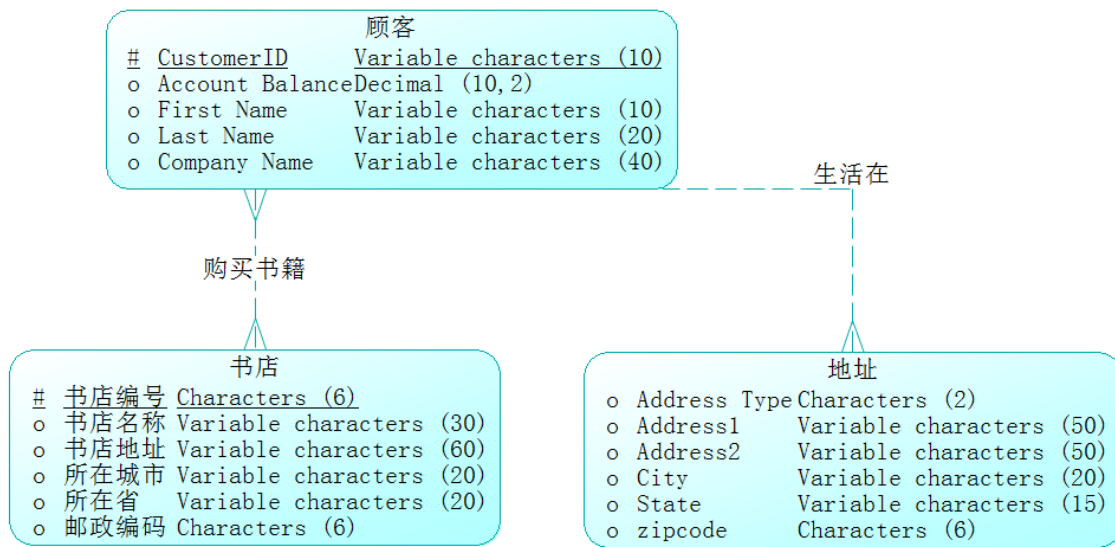
要删除实体或者联系，单击 Toolbox 上 Standard 的光标图标，选中你要删除的实体或者联系，按下 DEL 键，系统弹出确认删除的消息框。



你如果选择删除对象 (Delete Objects)，则把图形符号和模型中的对象都删除了；如果选择只删除符号 (Delete Symbols only)，则只删除图形符号，模型中的对象仍然存在。点击“Ok”键，删除一个实体或者联系。

三、实验任务

建立一个“实验一练习”命名的 Conceptual Data Model (CDM)文件，其中应包括的实体、实体的属性及实体之间的联系如图所示。请把文件保存好，以备实验二使用。



学有余力的同学，可以将教材相关部分的 E-R 图，在 PowerDesigner 中进行实现。

实验八 PowerDesigner 的 PDM 应用

二、实验目的

物理数据模型（PDM）充分考虑了模型的物理实现细节，包括：PDM 所选定的目标RDBMS 的特征；修改PDM 的特性以改善模型实现后的性能，增加系统可用性和安全性。

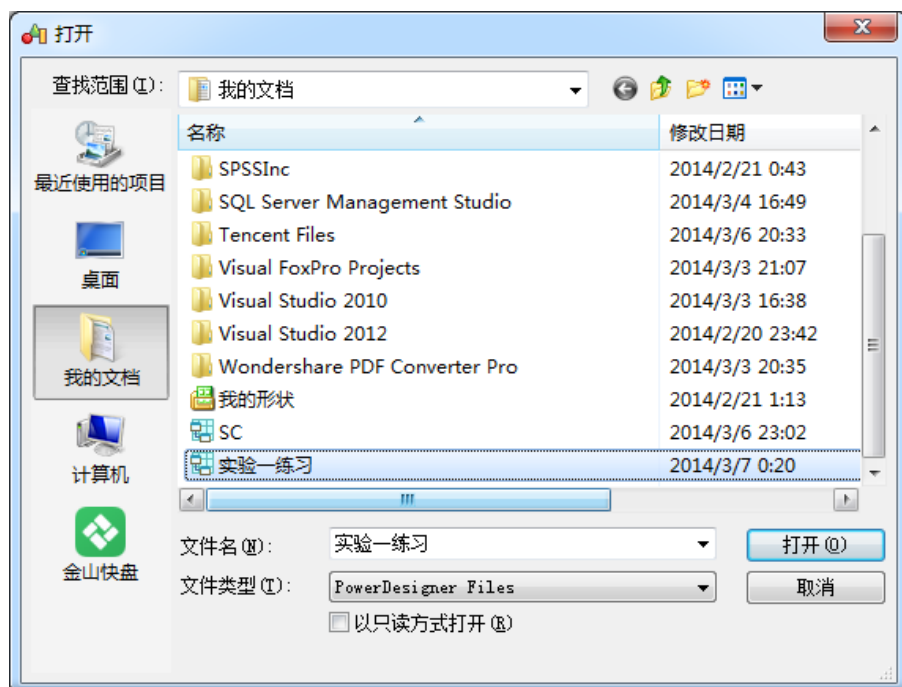
本次实验了解用PowerDesinger的工具将CDM生成PDM过程，熟悉PDM工作区，为生成数据库中的物理的表打下基础。完成这个实验后，应该能够了解从CDM到PDM的转换过程，熟悉PowerDesinger的PDM工作区，使用工具选项板进行PDM对象设计。

二、实验内容

从 CDM 生成 PDM，了解 CDM 生成 PDM 的过程。

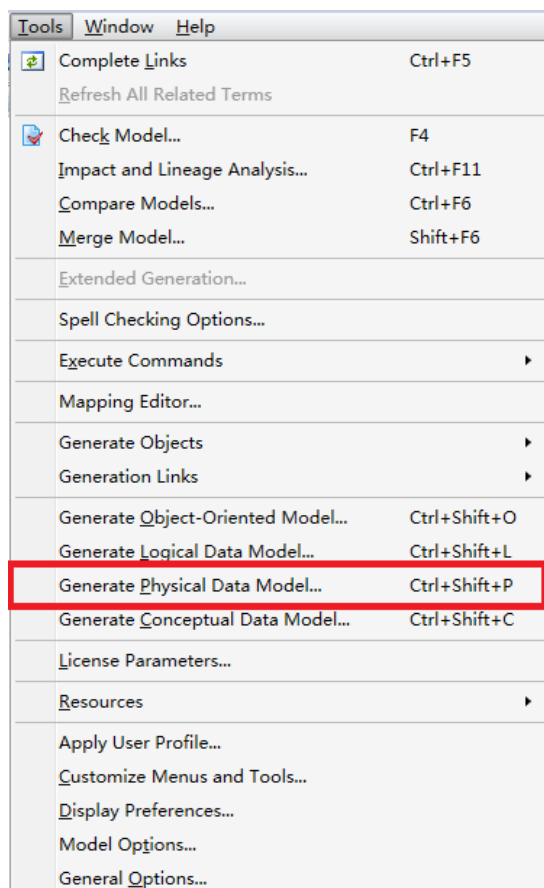
启动 PowerDesigner，选择 File→Open 菜单，打开“实验一练习.CDM”文件。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



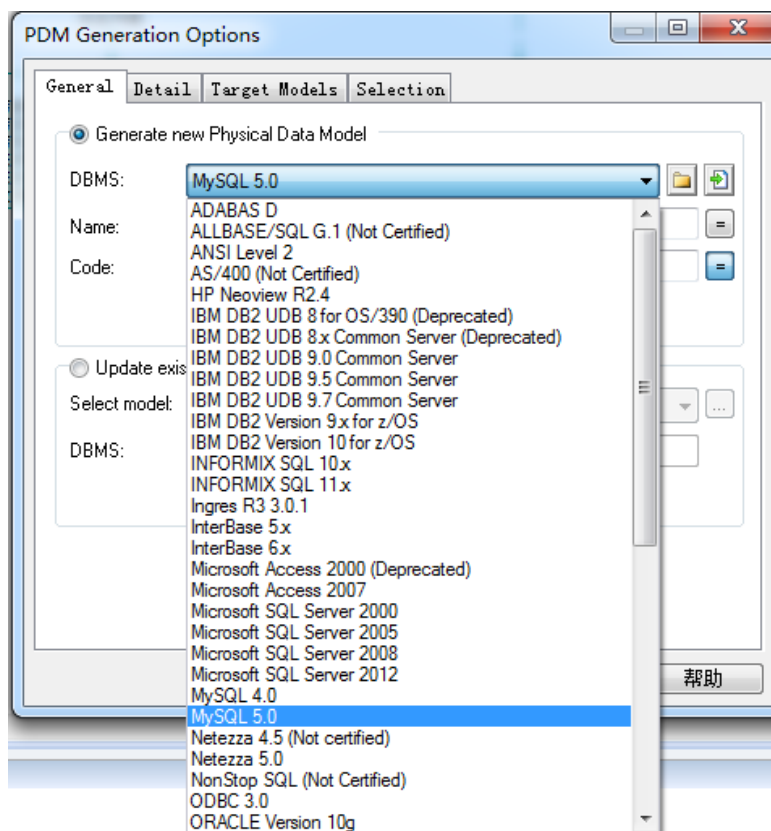
选择 Tools→Generate Physical Data Model，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

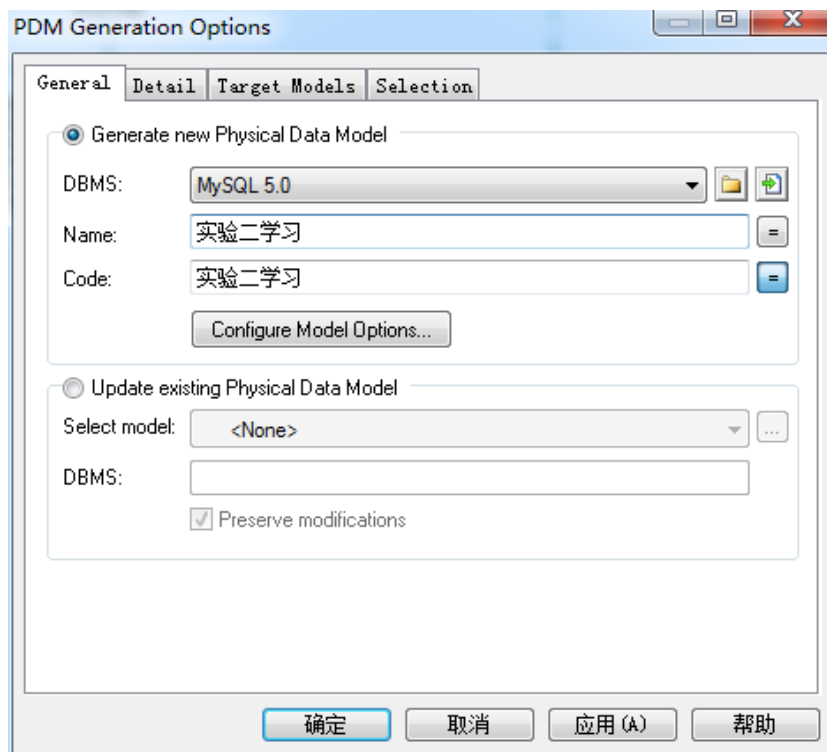


弹出 PDM 生成选择项窗口，DBMS 选择 MySQL 5.0，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

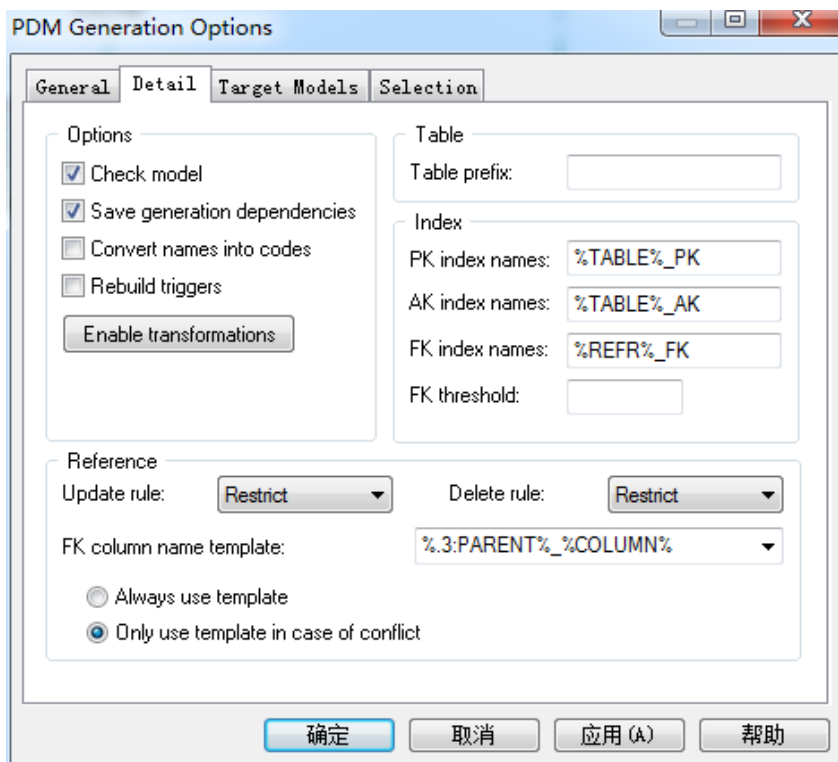


并把 Name 设置为实验二学习，如图所示。

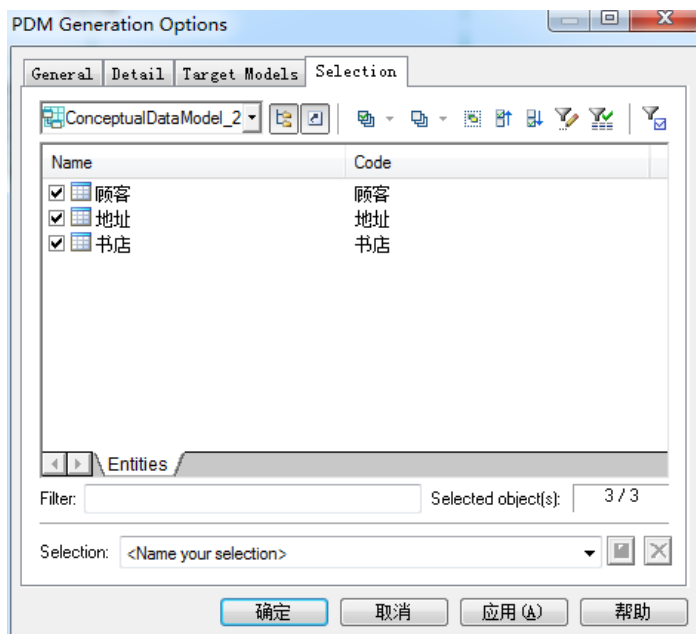


点击 Detail 页面，将选项中的 Check model 和 Save generation dependencies 两个选择框前打勾。如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

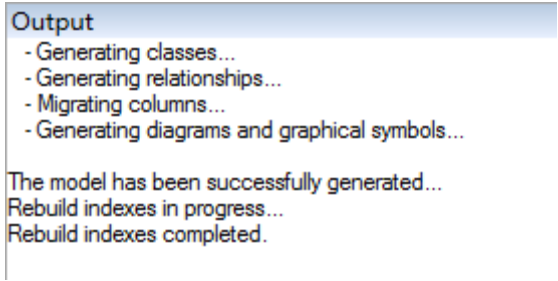


然后点击“Selection”页面，如图所示。

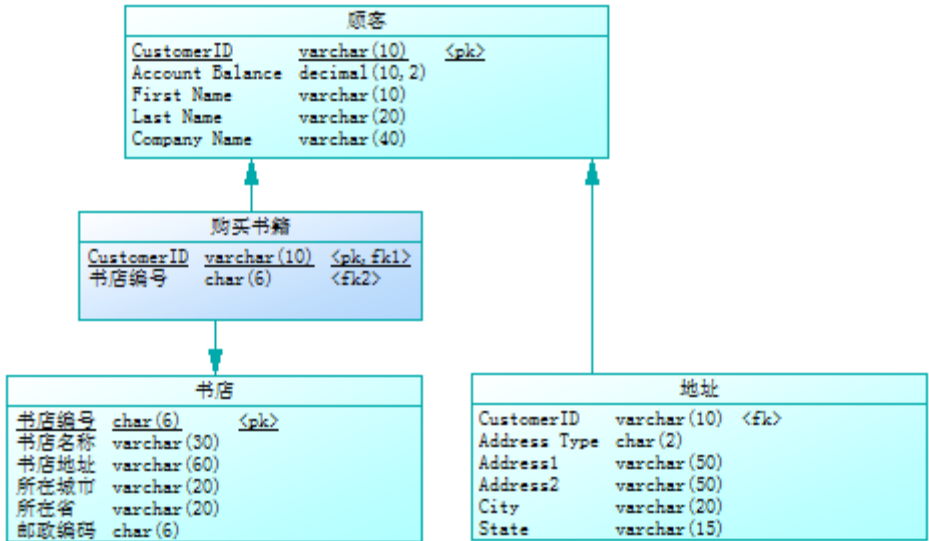


在这个页面里面包含了 CDM 中所有的对象，缺省的前面都打了勾，即都要被转换成 PDM。你可以将不需要转换的对象前的勾取掉。

点“Ok”按钮，系统将完成从 CDM 到 PDM 的转换，在输出窗口中可以看到转换产生的信息，如图所示。



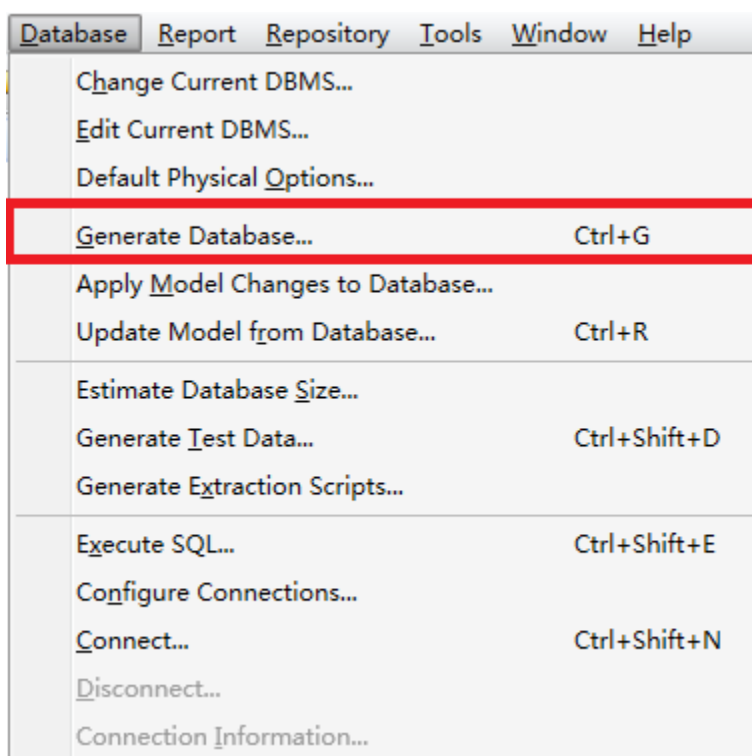
生成的 PDM 也出现在工作区当中，如图所示。



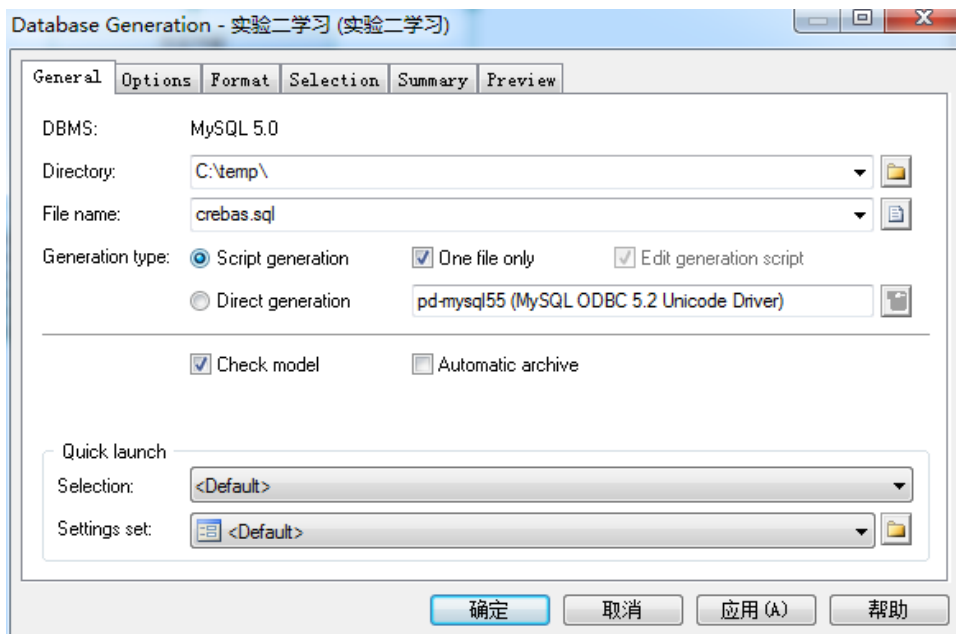
至此，从 CDM 生成 PDM 就完成了。在 PDM 中，根据 CDM 的定义，自动添加了 PK（主键）、FK（外键）等参照完整性约束条件。请大家认真查看和比较，特别是购买书籍这个表。

PDM 设计结束后，利用 PowerDesigner 的“生成数据库”功能，产生数据库中各数据对象的定义。操作方法是选择 Database→Generate Database 菜单，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



在出现的“数据库生成”对话框中，选择 Script Generation，单击“确定”按钮。



系统将完成从 PDM 生成创建 MySQL 数据库基本表的脚本，在输出窗口

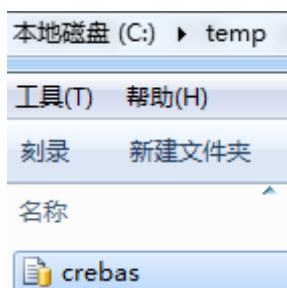
本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

中可以看到转换产生的信息，如图所示。

```
Output
Generation successful

Usage:
(1) Start command prompt
(2) Go to the directory C:\temp\
(3) Start the SQL interpreter:
    mysql.exe
(4) Run the database creation script:
    mysql> source crebas.sql
```

在 C 盘 temp 目录下，生成 crebas.sql，可打开我的电脑进行查看，如图所示。



文件内容如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

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

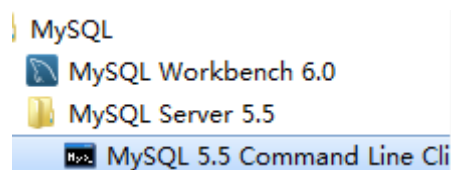
/*=====*/
/* DBMS name:      MySQL 5.0                      */
/* Created on:     2014/3/7 0:50:08                */
/*=====*/

drop table if exists 书店;
drop index 生活在_FK on 地址;
drop table if exists 地址;
drop index 购买书籍2_FK on 购买书籍;
drop index 购买书籍_PK on 购买书籍;
drop table if exists 购买书籍;
drop index 顾客_PK on 顾客;
drop table if exists 顾客;

/*=====*/
/* Table: 书店                                     */
/*=====*/
create table 书店
(
    书店编号          char(6) not null,
    书店名称          varchar(30),
    书店地址          varchar(60),
    所在城市          varchar(20),
    所在省            varchar(20),
    邮政编码          char(6)
);
```

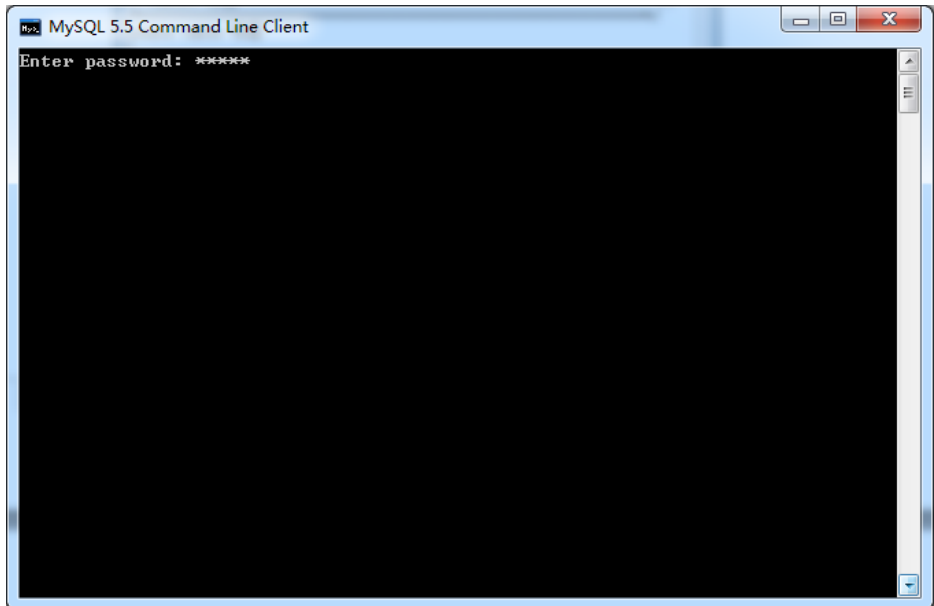
PowerDesigner 几乎能够产生所有常用数据库管理系统的 SQL 脚本，设计人员完全可以不经过手工编写 SQL 脚本直接在 DBMS 中生成数据库。使用记事本等文本编辑器打开生成的 SQL 脚本文件。将脚本复制到查询分析器中执行，无须手工创建表、视图等数据对象。

可以运行 MySQL 的命令行客户端软件，如图所示。

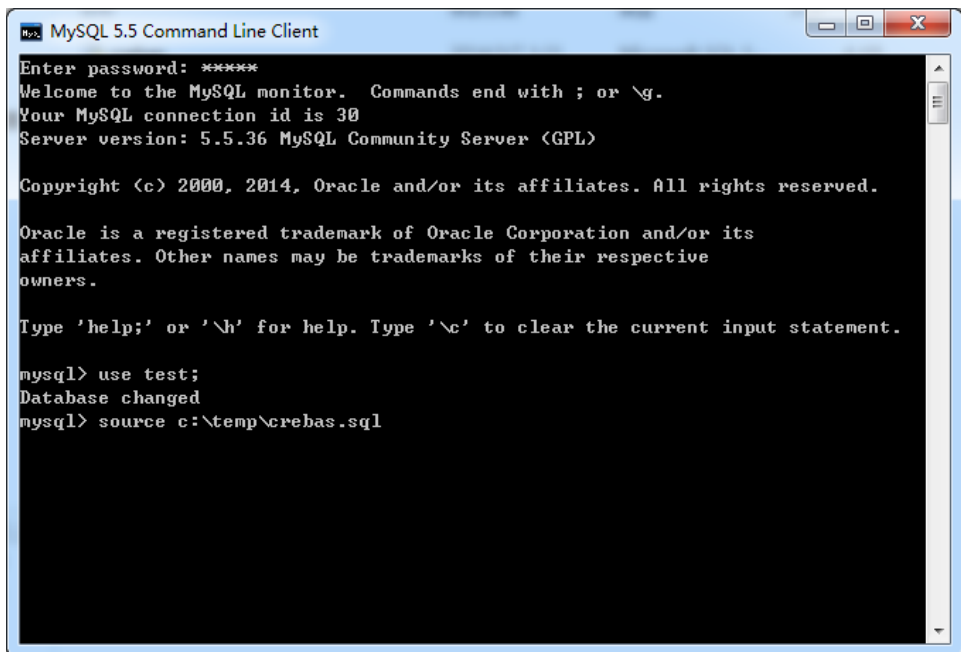


在登录界面输入密码 mysql，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

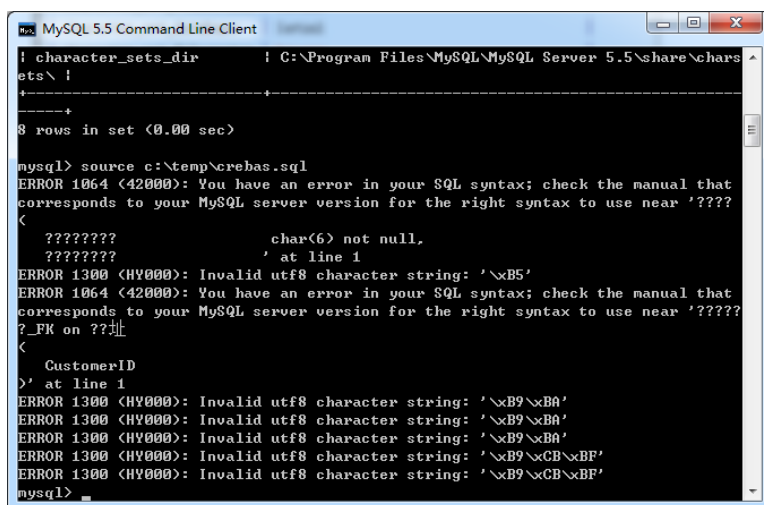


Use test; 打开 test 数据库，执行 source c:\temp\crebas.sql, 如图所示。



由于 SQL 脚本中含有中文，所以执行出错，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

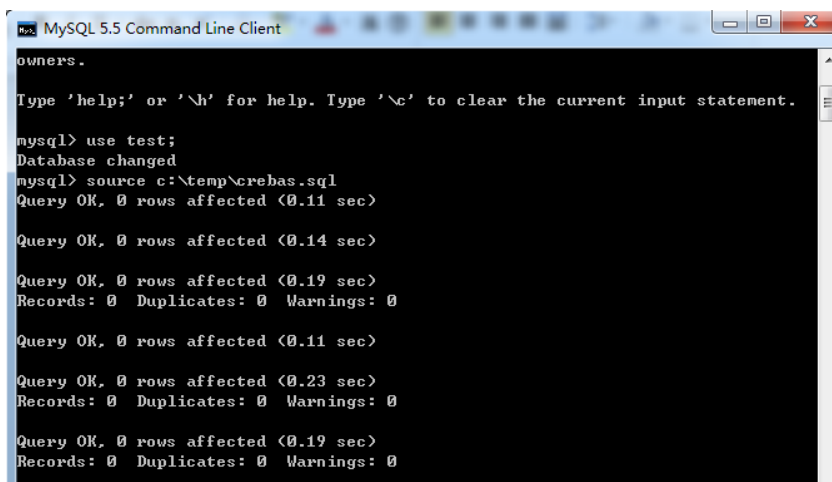


```
MySQL 5.5 Command Line Client
! character_sets_dir      ! C:\Program Files\MySQL\MySQL Server 5.5\share\chars
ets\ !
+-----+
+-----+
8 rows in set (0.00 sec)

mysql> source c:\temp\crebas.sql
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '????
(
????????? char(6) not null,
????????? ' at line 1
ERROR 1300 (HY000): Invalid utf8 character string: '\xB5'
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '?????
?_FK on ??址
(
CustomerID
>' at line 1
ERROR 1300 (HY000): Invalid utf8 character string: '\xB9\xBA'
ERROR 1300 (HY000): Invalid utf8 character string: '\xB9\xBA'
ERROR 1300 (HY000): Invalid utf8 character string: '\xB9\xBA'
ERROR 1300 (HY000): Invalid utf8 character string: '\xB9\xCB\xBF'
ERROR 1300 (HY000): Invalid utf8 character string: '\xB9\xCB\xBF'
mysql>
```

三、实验任务

将 crebas.sql 中的中文表名用英文替换，类似"Address Type"的字段名称用 AddressType 替换(去掉双引号和中间的空格)，顾客表中的"Account Balance"、"First Name"、"Last Name"、"Company Name"等字段，也去掉双引号和中间的空格。修改完毕后，保存 crebas.sql 文件，可按↑复制前一条命令，在 MySQL 命令行客户端中再次执行，成功结果如图所示。



```
MySQL 5.5 Command Line Client
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use test;
Database changed
mysql> source c:\temp\crebas.sql
Query OK, 0 rows affected (0.11 sec)

Query OK, 0 rows affected (0.14 sec)

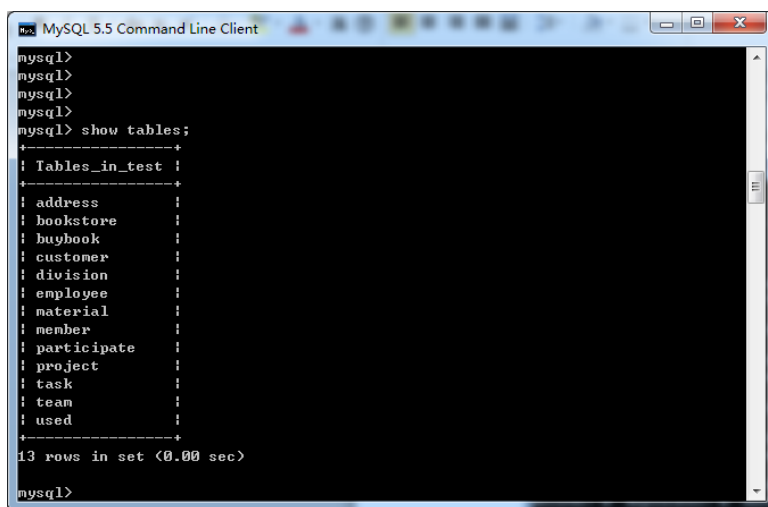
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.11 sec)

Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0

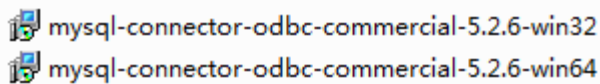
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

使用 show table 命令查看，结果如图所示。



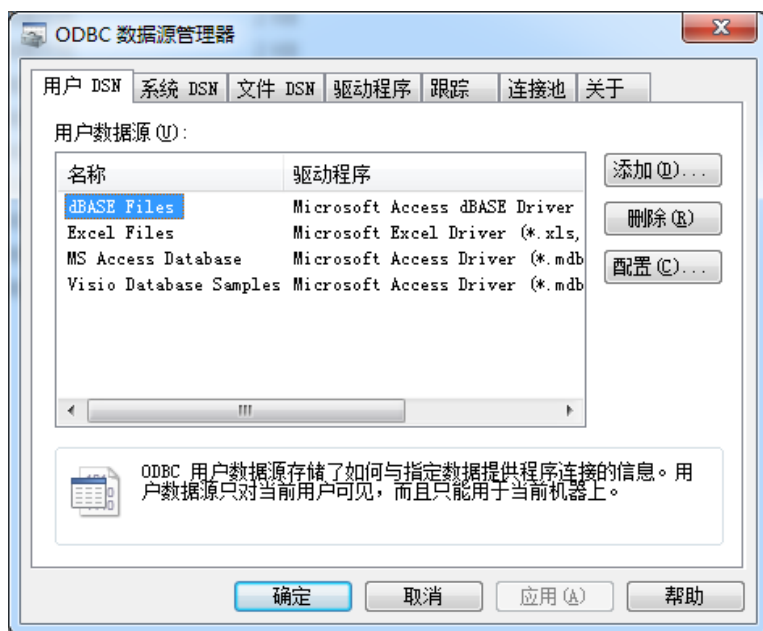
四、实验提升

学有余力的同学，根据操作系统选择安装 MySQL 的 ODBC 驱动程序，32 位操作系统安装 32 位驱动“mysql-connector-odbc-commercial-xxxx-win32”，64 位操作系统安装 64 位驱动“mysql-connector-odbc-commercial-xxxx-win64”，如图所示。

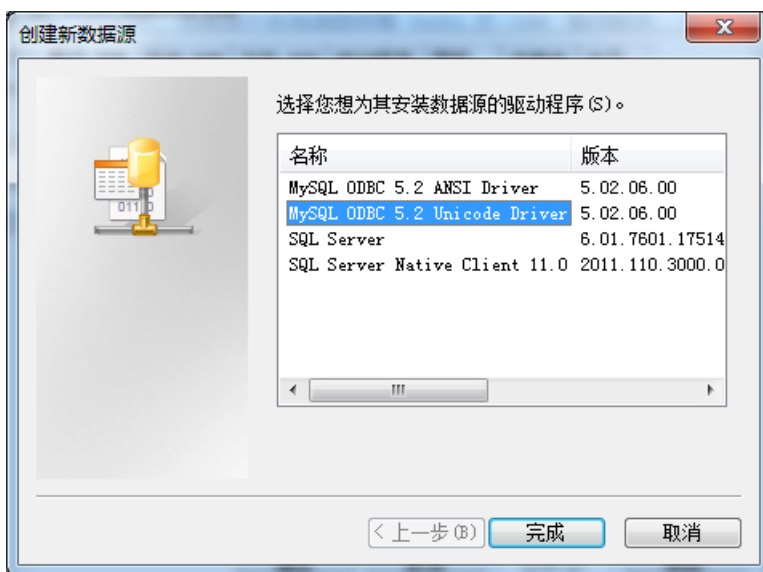


打开控制面板→系统和安全→管理工具→数据源(ODBC)，如图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

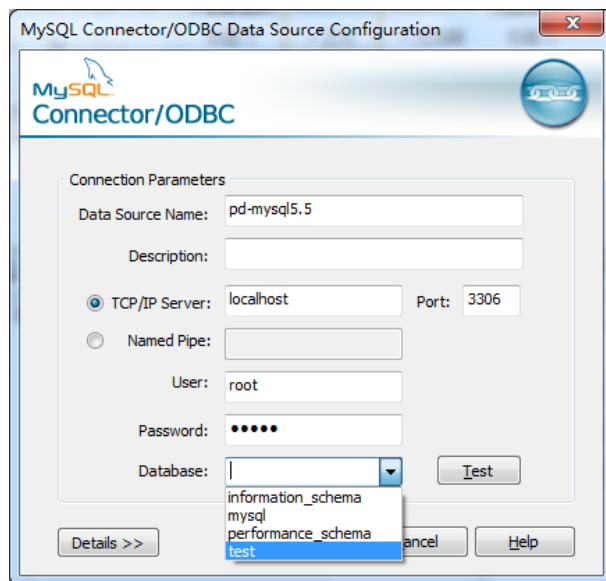


点击“添加”按钮，创建新的数据源，如图所示。



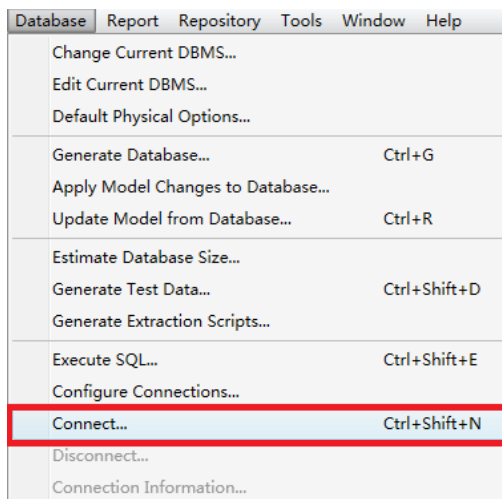
选择 MySQL ODBC x.x Unicode Driver，点击“完成”按钮。在弹出的 MySQL 数据源配置中，设置数据源名称“pd-mysql5.5”，TCP/IP Server 设置为 localhost，User 设置为 root，password 设置为 mysql，database 选择 test，如下图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



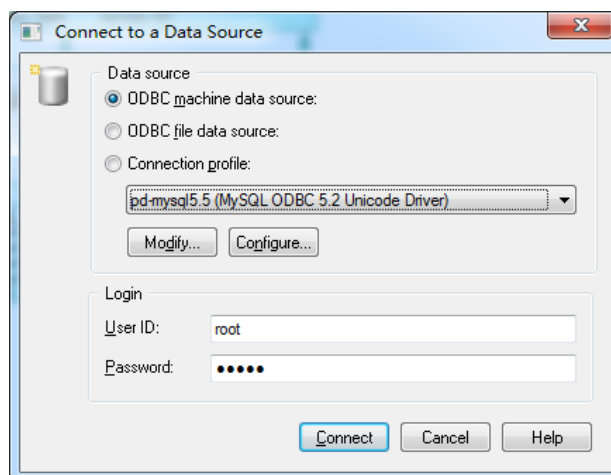
点击“ok”按钮，再点击“确定”按钮。

在 PowerDesigner 中，选择 Database 菜单中的 Connect，如下图所示。

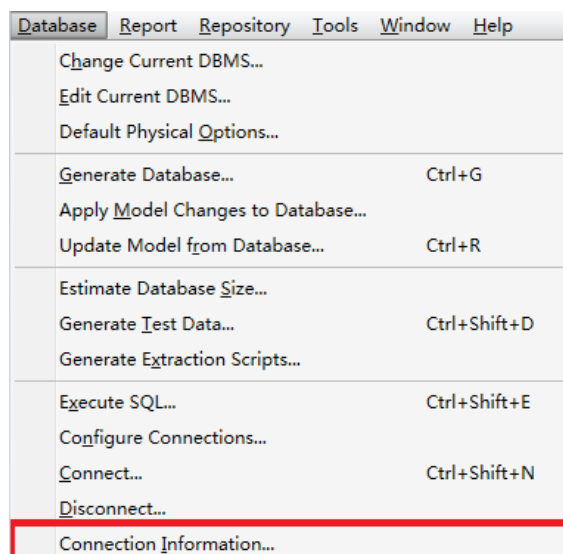


在连接数据源窗口中，选择刚才创建的数据源“pd-mysql5.5”，点击“connect”按钮。如下图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

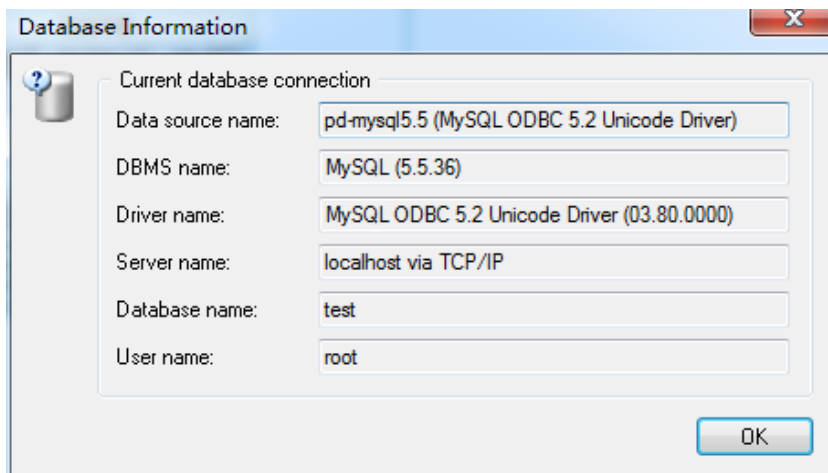


选择 Database 菜单中的 Connection Information 按钮，如下图所示。



如果数据库连接正确的话，会看到如下图所示画面。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

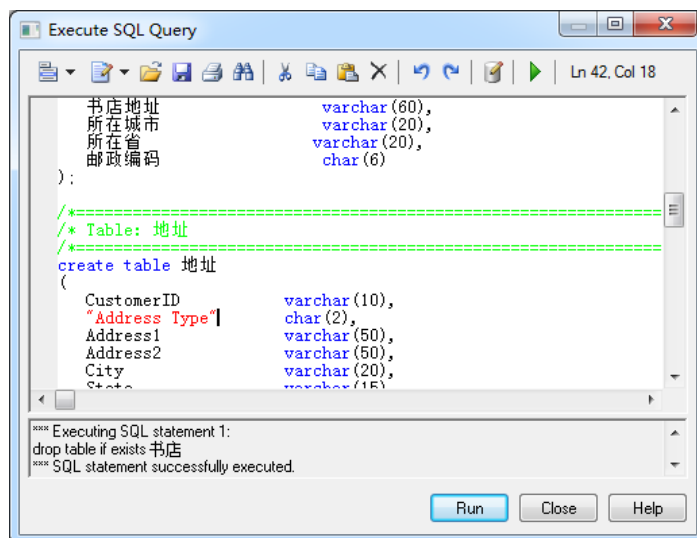


选择 Database 菜单下的, Generate Database...按钮, 选择 Direct generation, 点击“确定”按钮, 如下图所示。



在 Execute SQL Query 窗口中, 修改红色有错误的字体部分 (去掉双引号和中间的空格), 点击 Run 按钮执行 SQL 语句执行脚本, 生成数据库表、完整性约束等。如下图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。



成功执行后，可以使用其他图形化数据库管理工具，如 Navicat，连接到数据库 test 上，查看建立的中文表，如下图所示。

本实验内容为即将出版的教材之有效内容，请勿传播到互联网上。违者必究。

