

北京邮电大学 计算机学院
《数据库系统原理》实验报告

姓名 王睿嘉

学号 2015211906

班级 2015211307

实验一 数据库的安装、创建与维护

一、实验内容 and 环境描述

1. 实验目的

通过 MySQL 的安装和简单使用：

- 1) 了解安装 MySQL 的软硬件环境和安装方法；
- 2) 熟悉 MySQL 相关使用；
- 3) 通过 MySQL 的使用来理解数据库系统的基本概念。

通过创建数据库，并进行相应维护，了解并掌握 MySQL 数据库创建和维护的不同途径，并由具体的数据库理解数据库所包含的各要素。

2. 实验内容

1) MySQL 安装：

在 windows 10 上安装并运行 MySQL；

练习启动和停止数据库服务。

2) 数据库创建与维护：

创建学生选课数据库；

对数据库属性和参数进行查询、修改及维护，内容包括最大连接数、服务器端口、共享缓存数等；

练习数据库的删除等。

3. 实验要求

- 1) 在微机上安装 MySQL 数据库系统，为后续各实验搭建环境；
- 2) 实验要求面向具体应用领域，利用 MySQL 相关机制，创建并维护数据库系统，为后续各实验提供前期准备；
- 3) 根据需求确定实验步骤，独立完成以上实验内容，并熟悉 MySQL 的各种运行管理；
- 4) 实验完成后，撰写相应实验报告。

4. 实验环境

数据库系统：5.7.21 MySQL Community Server (GPL)

二、实验结论和心得

1. 实验结论

1.1 启动和停止数据库服务

在 DOS 窗口中，使用 net start 命令，进行 Windows 系统已启动服务的查询；

开启 MySQL 服务：使用 net start MySQL 命令，需以管理员身份执行；

关闭 MySQL 服务：通常情况下使用 net stop MySQL 即可，但由于本人的数据库设置了密码，因而采用 mysqladmin -u root -p shutdown 命令，并键入密码。

具体流程如下图所示：



```

管理用: 命令提示符
Windows Update
Windows 应用商店安装服务
Windows 推送通知用户服务_14542623
Windows 推送通知系统服务
Windows 许可证管理服务
WinHTTP Web Proxy Auto-Discovery Service
WLAN AutoConfig
Workstation
XLNSService
XLServicesPlatform
付款和 NFC/SE 管理器
同步主机_14542623
数据使用量
无线电管理服务
连接设备平台服务
连接设备平台用户服务_14542623

命令成功完成。

C:\WINDOWS\system32>mysqladmin -u root -p shutdown
Enter password: *****

C:\WINDOWS\system32>net start MySQL
MySQL 服务正在启动。
MySQL 服务已经启动成功。

C:\WINDOWS\system32>

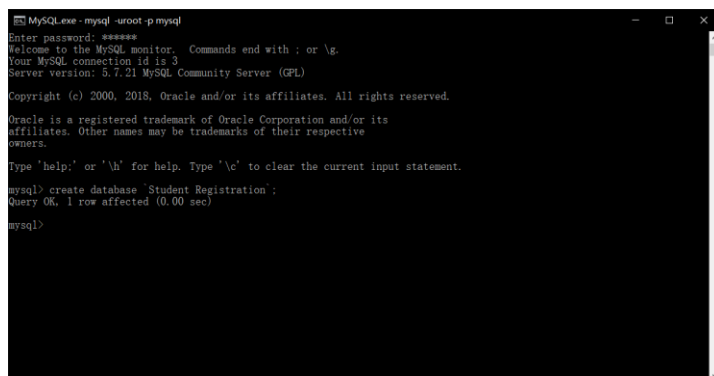
```

1.2 创建学生选课数据库

使用 create database 数据库名 命令，进行数据库的创建，且需使用引号。但非单引号，而是反引号；

在本实验中，具体命令为 create database `Student Registration`。

如下图所示：



```

MySQL.exe - mysql -uroot -p mysql
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database `Student Registration`;
Query OK, 1 row affected (0.00 sec)

mysql>

```

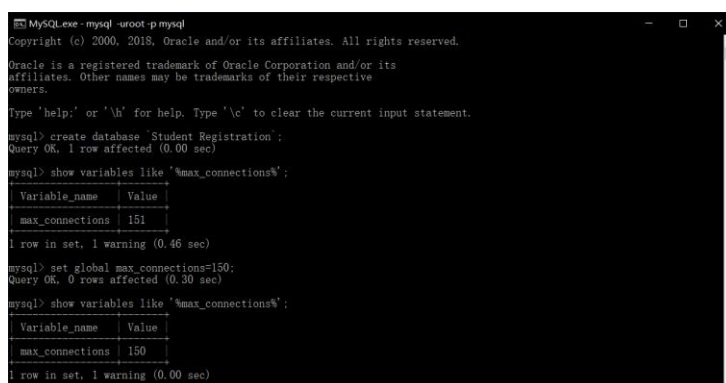
1.3 数据库属性和参数的查询及修改

最大连接数

使用 show variables like '%max_connections%' 命令，进行最大连接数的查询，结果为 151；

使用 set global max_connections = 最大连接数 命令，进行最大连接数的修改。

如下图所示：



```

MySQL.exe - mysql -uroot -p mysql
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database `Student Registration`;
Query OK, 1 row affected (0.00 sec)

mysql> show variables like '%max_connections%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 151 |
+-----+-----+
1 row in set, 1 warning (0.46 sec)

mysql> set global max_connections=150;
Query OK, 0 rows affected (0.30 sec)

mysql> show variables like '%max_connections%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 150 |
+-----+-----+
1 row in set, 1 warning (0.00 sec)

```

服务器端口

使用 `show variables like '%port%'` 命令，进行服务器端口的查询，结果为 3306；

服务器端口的修改，需更改配置文件。

共享缓存数

使用 `show variables like '%query_cache%'` 命令，进行共享缓存数的查询。其中，`have_query_cache` 代表是否开启，结果为 YES，`query_cache_limit` 代表单个查询能够使用的缓存区大小，结果为 1M，`query_cache_min_res_unit` 代表为系统分配的最小缓存块大小，结果为 4KB，`query_cache_size` 代表缓存大小，结果为 1M；

共享缓存数的修改，需更改配置文件。

如下图所示：

```

MySQL.exe - mysql -uroot -p mysql
mysql> show variables like '%port%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_support_xa | ON |
| large_files_support | ON |
| port | 3306 |
| report_host | |
| report_password | |
| report_port | 3306 |
| report_user | |
| require_secure_transport | OFF |
+-----+-----+
8 rows in set, 1 warning (0.00 sec)

mysql> show variables like '%query_cache%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_query_cache | YES |
| query_cache_limit | 1048576 |
| query_cache_min_res_unit | 4096 |
| query_cache_size | 1048576 |
| query_cache_type | OFF |
| query_cache_wlock_invalidate | OFF |
+-----+-----+
6 rows in set, 1 warning (0.00 sec)

mysql>

```

1.4 数据库的删除

使用 `drop database 数据库名` 命令，进行数据库的删除，且需使用反引号；

在本实验中，具体命令为 `drop database `Student Registration``。

如下图所示：

```

MySQL.exe - mysql -uroot -p mysql
mysql> show variables like '%port%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_support_xa | ON |
| large_files_support | ON |
| port | 3306 |
| report_host | |
| report_password | |
| report_port | 3306 |
| report_user | |
| require_secure_transport | OFF |
+-----+-----+
8 rows in set, 1 warning (0.00 sec)

mysql> show variables like '%query_cache%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_query_cache | YES |
| query_cache_limit | 1048576 |
| query_cache_min_res_unit | 4096 |
| query_cache_size | 1048576 |
| query_cache_type | OFF |
| query_cache_wlock_invalidate | OFF |
+-----+-----+
6 rows in set, 1 warning (0.00 sec)

mysql> drop database `Student Registration`;
Query OK, 0 rows affected (0.49 sec)

mysql>

```

2. 实验心得

在本次实验中，遇到的主要问题有以下三点：

- ① 使用 `net start MySQL` 命令，进行数据库的启动时，提示拒绝访问。随后，以管理员身份运行 cmd，再重复以上操作，该问题得到解决；
- ② 使用 `net stop MySQL` 命令，进行数据库的关闭时，提示拒绝访问。经查阅资料，得知在数据库设有密码的情况下，需使用 `mysqladmin -u root -p shutdown` 命令，并键入密码，方可关闭；

③ 创建数据库时，名称需用反引号。未实践前，仅由书本判断，误以为是单引号。

实践出真知，本次数据库的安装、创建与维护实验是对课堂和书本所学知识的补充。实际情形与已了解到的原理大体一致，但又有许多细节未曾注意。通过自己动手、亲力亲为编写 SQL 语言，加深了对数据库各要素的理解和记忆，收获颇丰。

实验二 数据库表/视图的创建与维护

一、实验内容 and 环境描述

1. 实验目的

- 1) 通过数据库表的建立操作, 熟悉并掌握 MySQL 数据库表的建立方法, 理解关系数据库表的结构, 巩固 SQL 标准中关于数据库表建立的语句;
- 2) 通过数据库表数据的增加、删除和插入等维护操作, 熟悉并掌握 MySQL 数据库数据的操作方法, 巩固 SQL 中关于数据维护的语句;
- 3) 通过在 MySQL 中建立、维护视图, 熟悉对视图的操作方法, 理解和掌握视图的概念。

2. 实验内容

2.1 数据库表实验

1) 创建数据库表

创建学生选课数据库中所包含的数据库表, 结构如下:

学生 (学号, 姓名, 性别, 年龄, 系别, 班级);

课程 (课程编号, 课程名称, 学分);

选修 (学号, 课程编号, 学生成绩);

要求为各属性选择合适的数据类型。

2) 查看和修改表结构

选择一个数据库表, 练习对其表结构的修改。

3) 对三张表分别插入数据;

4) 对表中数据进行修改;

5) 对表中数据进行删除。

2.2 视图实验

1) 建立视图

建立学生选修课程信息视图, 包括以下内容: 学生学号、姓名、所在系、选修课程编号、课程名称和成绩。

2) 修改视图

增加学生所在班级信息。

3) 删除视图

3. 实验环境

数据库系统: 5.7.21 MySQL Community Server (GPL);

集成开发环境: PyCharm;

解释器版本: 3.6.1。

二、实验结论和心得

1. 实验结论

1.1 数据库表

- ① 使用 use `Student Registration System` 命令，进入指定数据库；
- ② 使用 source 路径 命令，导入 sql 文件。其中，路径的斜杠方向与操作系统的写法相反；
- ③ 使用 desc 数据表名 命令，进行表结构的查询；
- ④ 使用 select * from 数据表名 命令，进行表内容的查询，但中文均为乱码；
- ⑤ 使用 delete from 数据表名 命令，进行表内容的删除，并返回已删除记录数，但不删除表的结构。
若再次使用 select 访问，将会显示 empty set。如果使用 drop table 数据表名 命令，会同时删除表的数据和结构；
使用 delete from 数据表名 where 命令，进行行的删除；
- ⑥ 使用 set names utf8 命令，进行数据库编码的设置；
- ⑦ 使用 create table 数据表名() 命令，进行数据库表的创建。其中，not null 代表不允许空值，若缺省，则允许空值。在本实验中，具体命令为 create table sc(studentID varchar(6) not null, courseID varchar(3) not null, grade int(11), primary key(studentID,courseID));
- ⑧ 使用 insert into 数据表名 value() 命令，进行表的插入，且数据均使用单引号。在本实验中，具体命令为 insert into course value('C04','JAVA 程序设计','2');
- ⑨ 使用 alter table 数据表名 change column 原属性名 现属性名 命令，进行属性名的更改。在本实验中，具体命令为 alter table student change column sname name varchar(6);
使用 alter table 数据表名 add column 属性名 命令，进行属性的增加；
使用 alter table 数据表名 drop column 属性名 命令，进行属性的删除；
使用 alter table 数据表名 drop primary key 命令，进行主键的删除；
使用 alter table 数据表名 add primary key() 命令，进行主键的增加；
- ⑩ 使用 update 数据表名 set 更改 where 命令，进行数据的修改。

部分流程如下图所示：

```

MySQL.exe -mysql -uroot -pmysql
mysql> use `Student Registration System`;
mysql> desc student;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| sID | varchar(6) | NO | PRI | NULL | |
| sname | varchar(6) | YES | | NULL | |
| sex | varchar(2) | YES | | NULL | |
| age | datetime | YES | | NULL | |
| dept | varchar(10) | YES | | NULL | |
| class | varchar(4) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.15 sec)

mysql> select * from course;
+----+-----+-----+
| cID | cname | credit |
+----+-----+-----+
| C01 | 编译原理 | 3 |
| C02 | 数据库原理 | 3 |
| C03 | 操作系统 | 3 |
| C04 | JAVA 程序设计 | 3 |
| C05 | 计算机组成原理 | 3 |
+----+-----+-----+
5 rows in set (0.30 sec)

mysql>
MySQL.exe -mysql -uroot -pmysql
mysql> set names utf8;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into course value(
-> '1','哈',1);
Query OK, 1 row affected (0.36 sec)

mysql> select * from course;
+----+-----+-----+
| cID | cname | credit |
+----+-----+-----+
| 1 | 哈 | 1 |
| C01 | 编译原理 | 3 |
| C02 | 数据库原理 | 3 |
| C03 | 操作系统 | 3 |
| C04 | JAVA 程序设计 | 3 |
| C05 | 计算机组成原理 | 3 |
+----+-----+-----+
6 rows in set (0.00 sec)

mysql> delete from course where cID='1';
Query OK, 1 row affected (0.00 sec)

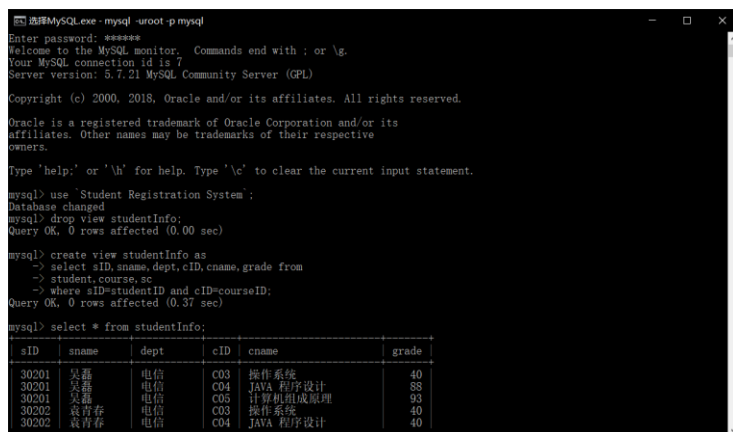
mysql> select * from course;
+----+-----+-----+
| cID | cname | credit |
+----+-----+-----+
| C01 | 编译原理 | 3 |
| C02 | 数据库原理 | 3 |
| C03 | 操作系统 | 3 |
| C04 | JAVA 程序设计 | 3 |
| C05 | 计算机组成原理 | 3 |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql>
  
```

1.2 视图

- ① 使用 create view 视图名 as select 属性名 from 数据表 where 命令，进行视图的创建，在本实验中，具体命令为 create view studentInfo as select sID,sname,dept,cID,cname,grade from student,course,sc where sID=studentID and cID=courseID ；
- ② 使用 alter view 视图名() as select 属性名 from 数据表 where 命令，进行视图的修改；
- ③ 使用 drop view 视图名，进行视图的删除。

部分流程如下图所示：



```

mysql> use 'Student Registration System';
Database changed
mysql> drop view studentInfo;
Query OK, 0 rows affected (0.00 sec)

mysql> create view studentInfo as
  -> select sID,sname,dept,cID,cname,grade from
  -> student,course,sc
  -> where sID=studentID and cID=courseID;
Query OK, 0 rows affected (0.37 sec)

mysql> select * from studentInfo;
+----+-----+-----+----+-----+-----+
| sID | sname | dept | cID | cname | grade |
+----+-----+-----+----+-----+-----+
| 30201 | 吴磊 | 电信 | C03 | 操作系统 | 40 |
| 30201 | 吴磊 | 电信 | C04 | JAVA 程序设计 | 88 |
| 30201 | 吴磊 | 电信 | C05 | 计算机组成原理 | 93 |
| 30202 | 袁青春 | 电信 | C03 | 操作系统 | 40 |
| 30202 | 袁青春 | 电信 | C04 | JAVA 程序设计 | 40 |
+----+-----+-----+----+-----+-----+
  
```

2. 实验心得

在本次实验中，遇到的主要问题有以下两点：

- ① 直接导入所提供的 sql 文件，中文均为乱码。首先，使用 set names utf8 命令，将数据库的中文字符集设置为 UTF-8。然后，将 student.txt 另存为 UTF-8 编码格式。最终，利用 python，导入 PySQL 库，编写相应代码将数据输入数据库；
- ② 对于 SQL 命令，相关记忆存在诸多模糊点。经网上查阅，顺利完成实验。

实践出真知，本次数据库表/视图的创建与维护实验是对课堂和书本所学知识的补充。由于数据版本不匹配，进一步提高了对中文字符集的处理能力。通过自己动手、亲力亲为编写 SQL 语句，加深了对查询语言的理解和记忆，收获颇丰。