

《数据库原理》实验报告

实验名称 数据库表/视图的创建与维护实验

班 级 2014211304

学 号 2014211218

姓 名 史文翰

实验二 数据库表/视图的创建与维护实验

一、 实验目的

1. 通过进行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的建立语句。
2. 通过进行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 MYSQL 中建立、维护视图的实验，熟悉 MYSQL 中对视图的操作方法和途径，理解和掌握视图的概念。

二、 实验平台及环境

本次试验采用 MYSQL5.7 作为实验环境。其中命令行方式为 MYSQL 5.7 Command Line，图形化界面及脚本方式为 MYSQL Workbench。操作系统环境为 win10 64bit。

三、 实验内容

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，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）

要求为各数据库表属性选择合适的数据类型。

列名、数据类型（长度列、精度、小数位数项）、是否允许空值等。

(2) 查看和修改表结构。

选择一个数据库表，练习对其表结构进行相应的修改。

(3) 练习用交互式的 SQL 语句分别完成以上操作。

(4) 对学生数据库中的三张表分别插入数据；

(5) 对表中的数据进行修改；

(6) 对表中的数据进行删除操作；

2、视图实验：视图的建立、修改、删除

(1) 创建以下视图：

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、选修的课程编号、课程名称、成绩

(2) 修改以上视图，增加学生所在班级信息。

(3) 对以上视图删除。

四、 实验步骤及结果分析

1、创建数据库表

在 workbench 中建立如下批处理脚本。

```
1  /* 选课系统 */
2
3  /* 学生信息表 */
4  CREATE TABLE `student` (
5      `stu_no` varchar(6) NOT NULL, /* 学生学号, 5位数字, 非空 */
6      `stu_name` varchar(8) DEFAULT NULL, /* 学生姓名, 最多四个汉字, 默认为空 */
7      `sex` varchar(2) DEFAULT NULL, /* 学生性别, 1个汉字, 默认为空 */
8      `b_date` datetime DEFAULT NULL, /* 学生生日, 采用日期类型, 默认为空 */
9      `dept` varchar(8) DEFAULT NULL, /* 学生所属专业, 最多四个汉字, 默认为空 */
10     `class_no` varchar(4) DEFAULT NULL, /* 学生班级, 4位数字, 默认为空 */
11     PRIMARY KEY (`stu_no`) /* 主键为 学生学号 */
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8; /* 使用utf-8编码 */
13
14 CREATE TABLE `sel` (
15     `stu_no` varchar(6) NOT NULL, /* 学生学号, 5位数字, 非空 */
16     `course_no` varchar(3) NOT NULL, /* 课程编号, 3位字符, 非空 */
17     `grade` int DEFAULT NULL, /* 成绩, 0-100, 默认为空 */
18     PRIMARY KEY (`stu_no`,`course_no`) /* 主键为 (学生学号, 课程编号) */
19 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
20
21 CREATE TABLE `course` (
22     `course_no` varchar(3) NOT NULL, /* 课程编号, 3位字符, 非空 */
23     `course_name` varchar(12) DEFAULT NULL, /* 课程名, 最多6个汉字, 默认为空 */
24     `hours` int DEFAULT NULL, /* 课时, 整形, 默认为空 */
25     `credit` int DEFAULT NULL, /* 学分, 整形, 默认为空 */
26     `semester` varchar(2) DEFAULT NULL, /* 课程所在学期, 1个汉字, 默认为空 */
27     PRIMARY KEY (`course_no`) /* 主键为 课程编号 */
28 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

所有的解释都以注释的形式给出。值得说明的是，在 course 表里没有选用课程名作为主键是因为，可能存在同一个课程名由不同的老师讲解，从而出现了不同的课程。因此唯一区别各个课程的是课程 ID 而不是课程名。

至此，我们完成了三张表的建立，在 shell 中查看如下：

```
mysql> show tables;
+-----+
| Tables_in_选课系统 |
+-----+
| course              |
| sel                  |
| student              |
+-----+
3 rows in set (0.00 sec)
```

2、查看和修改表结构

以 student 表为例，增加一个字段为“是否脱单”，名称为 dog。

先插入一条数据：

insert into student values

('31111', '史文翰', '男', '1995-12-5 12:00', '计算机', '2333');

```
mysql> insert into student values
-> ('31111','史文翰','男','1995-12-5 12:00','计算机','2333');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from student
-> ;
```

stu_no	stu_name	sex	b_date	dept	class_no
31111	史文翰	男	1995-12-05 12:00:00	计算机	2333

1 row in set (0.00 sec)

用如下命令增加 dog 字段（默认为 NULL 值）：

```
alter table student add column dog varchar(1);
```

加入 dog 字段后：

```
mysql> alter table student add dog varchar(1);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from student;
```

stu_no	stu_name	sex	b_date	dept	class_no	dog
31111	史文翰	男	1995-12-05 12:00:00	计算机	2333	NULL

1 row in set (0.00 sec)

删除这个列，可用

```
alter table student drop column dog;
```

```
mysql> alter table student drop column dog;
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from student;
```

stu_no	stu_name	sex	b_date	dept	class_no
31111	史文翰	男	1995-12-05 12:00:00	计算机	2333

1 row in set (0.00 sec)

3、插入数据

上图对于“史文翰”的插入即使插入数据的例子，在此不做赘述。

4、修改数据

例如，修改“史文翰”的专业为“通信工程”。可使用如下 SQL 语句：

```
update student
set dept='通信工程'
where stu_name='史文翰';
```

结果如下：

```
mysql> update student
    -> set dept='通信工程'
    -> where stu_name='史文翰';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
```

stu_no	stu_name	sex	b_date	dept	class_no
31111	史文翰	男	1995-12-05 12:00:00	通信工程	2333

```
1 row in set (0.00 sec)
```

5、删除数据

删除史文翰这条信息，可使用如下 SQL 语句：

```
delete from student
where stu_name='史文翰';
```

效果如下：

```
mysql> delete from student
    -> where stu_name='史文翰';
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
Empty set (0.00 sec)
```

由于 student 表中只有这一项，删除之后变为空表。

6、创建视图

首先，我们利用 load 命令将给定的 txt 文件的数据导入“选课系统”的表中。以 student 为例，注意，我们需要将 txt 文件转码为 utf8，假设我们准备好了这样的文件，使用如下命令（默认开始时 student 表为空）：

```
load data local infile 'G:\data\student-utf8.txt' into table student;
load data local infile 'G:\data\sc-utf8.txt' into table sel;
load data local infile 'G:\data\course-utf8.txt' into table course;
```

再利用 select 语句查看完整的 student 表：

```
mysql> select * from student;
```

stu_no	stu_name	sex	b_date	dept	class_no
30201	吴磊	男	1980-01-02 00:00:00	电信	3022
30202	袁青春	男	1980-01-02 00:00:00	电信	3022
30203	唐雷	男	1980-01-02 00:00:00	电信	3022
30204	吴霏	男	1980-01-02 00:00:00	电信	3022
30206	连洪炽	男	1980-01-02 00:00:00	电信	3022
30207	王金柱	男	1980-01-02 00:00:00	电信	3022
30208	苏广学	男	1980-01-02 00:00:00	电信	3022
30209	唐元亮	男	1980-01-02 00:00:00	电信	3022
30210	葛艳杰	男	1980-01-02 00:00:00	电信	3022
30211	张永超	男	1980-01-02 00:00:00	电信	3022
30212	张伟	男	1980-01-02 00:00:00	电信	3022
30213	孙刚	男	1980-01-02 00:00:00	电信	3022
30214	车平跃	男	1980-01-02 00:00:00	电信	3022
30215	张鑫	男	1980-01-02 00:00:00	电信	3022
30216	王凯	男	1980-01-02 00:00:00	电信	3022
30217	马涛	男	1980-01-02 00:00:00	电信	3022
30218	王佳楠	男	1980-01-02 00:00:00	电信	3022
30219	马浩	男	1980-01-02 00:00:00	电信	3022
30220	邵苏杰	男	1980-01-02 00:00:00	电信	3022
30221	郑建华	男	1980-01-02 00:00:00	电信	3022
30222	郑侃彦	男	1980-01-02 00:00:00	电信	3022
30223	宁剑震	女	1980-01-02 00:00:00	电信	3022

类似地，我们导入另外两张表，得到如下所示的结果：

```
mysql> select * from sel;
```

stu_no	course_no	grade
30201	C03	40
30201	C04	88
30201	C05	93
30202	C03	40
30202	C04	40
30203	C03	57
30203	C04	50
30203	C05	40
30204	C03	54
30204	C04	50
30204	C05	40
30206	C03	40
30206	C04	40
30206	C05	50

```
mysql> select * from course
-> ;
```

course_no	course_name	hours	credit	semester
C01	编译原理	51	3	秋
C02	数据库原理	51	3	春
C03	操作系统	51	2	秋
C04	JAVA 程序设计	40	2	秋
C05	计算机组成原理	30	2	春

5 rows in set (0.00 sec)

接下来我们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、选修的课程编号、课程名称、成绩。

```
create view info_sel as
```

```

select
student.stu_no as stu_no,
student.stu_name as stu_name,
student.dept as dept,
course.course_no as course_no,
course.course_name as course_name,
sel.grade as grade
from
student natural join course natural join sel ;

```

得到如下效果：

```

mysql> create view info_sel as
-> select
-> student.stu_no as stu_no,
-> student.stu_name as stu_name,
-> student.dept as dept,
-> course.course_no as course_no,
-> course.course_name as course_name,
-> sel.grade as grade
-> from
-> student natural join course natural join sel
-> ;

```

Query OK, 0 rows affected (0.01 sec)

```
mysql> select * from info_sel;
```

stu_no	stu_name	dept	course_no	course_name	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
30203	唐雷	电信	C03	操作系统	57
30203	唐雷	电信	C04	JAVA 程序设计	50
30203	唐雷	电信	C05	计算机组成原理	40

7、修改视图

利用 alter view 语句，只需要在 create 的基础上增加一列即可。

```

alter view info_sel as
select
student.stu_no as stu_no,
student.stu_name as stu_name,
student.dept as dept,
course.course_no as course_no,
course.course_name as course_name,
sel.grade as grade,
student.class_no as class_no
from
student natural join course natural join sel ;

```

```
mysql> alter view info_sel as
-> select
-> student.stu_no as stu_no,
-> student.stu_name as stu_name,
-> student.dept as dept,
-> course.course_no as course_no,
-> course.course_name as course_name,
-> sel.grade as grade,
-> student.class_no as class_no
-> from
-> student natural join course natural join sel
-> ;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from info_sel;
```

stu_no	stu_name	dept	course_no	course_name	grade	class_no
30201	吴磊	电信	C03	操作系统	40	3022
30201	吴磊	电信	C04	JAVA 程序设计	88	3022
30201	吴磊	电信	C05	计算机组成原理	93	3022
30202	袁青春	电信	C03	操作系统	40	3022
30202	袁青春	电信	C04	JAVA 程序设计	40	3022
30203	唐雷	电信	C03	操作系统	57	3022
30203	唐雷	电信	C04	JAVA 程序设计	50	3022
30203	唐雷	电信	C05	计算机组成原理	40	3022

如上图，我们成功在 info_sel 的视图上增加了 class_no 字段，即学生的班级信息。

8、删除视图

可以使用如下命令删除该视图：

```
drop view info_sel;
```

操作很简单，在此不再赘述。

五、 实验小结

在本次实验中我体会到了一系列的对于 sql 语句的使用。主要包括数据库、表、视图的建立、维护和删除。mysql 的命令行模式非常平易近人，上手难度较低，非常适合数据初学者在 shell 模式下联系 sql 相关语句。对比实验一，我们可以尽可能少用 workbench 而多使用 shell 命令来加深对 sql 的理解。

此外，我们从 mysql 中也体会到了一款数据库软件的魅力，对比 kingbase，mysql 有着更友好的交互界面，更完备的官方文档和说明。由于其在数据库软件市场的地位，网上关于 mysql 的资料也是数不胜数，方便了新手的入门和进一步探索。

然而在实验中也遇到了一些问题，比如初次使用 load 命令时文件路径有中文的问题，还有给定的 txt 文档直接导入会出现乱码的问题。前者可以通过对 load 命令的进一步了解、对转义字符的了解而解决；后者是对编码问题的进一步认识。在本次以及今后的实验中，我们统一采用 utf-8 编码模式，来减少因为编码不一致而可能产生的问题。