

# 北京邮电大学数据库系统实验报告

---

## 实验名称: 数据库表/视图的创建与维护实验

计算机科学与技术系 2015211306班

魏晓 学号:2015211301

---

## 实验目的

1. 通过进行数据库表的建立操作, 熟悉并掌握Kingbase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Kingbase数据库数据的操作方法, 巩固SQL中关于数据维护的语句.
  3. 通过对Kingbase中建立、维护视图的实验, 熟悉Kingbase中对视图的操作方法和途径, 理解和掌握视图的概念.
- 

## 实验要求

- 用交互式语句完成以上操作
  - 能够对整个过程进行批命令操作
  - 要求学生独立完成以上内容
  - 根据以上内容确定实验步骤
  - 实验完成后完成要求的实验报告内容
- 

## 实验平台及环境

实验平台:MySQL 14.14

运行环境:Mac OS High Sierra 10.13(17A405)

可视界面:MySQL WorkBench 6.3.10

Xiaos-MacBook-Pro:~ weixiao\$ mysql --version

mysql Ver 14.14 Distrib 5.7.20, for macos10.12 (x86\_64) using EditLine wrapper

## 实验内容

### 数据库表实验：

- **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。

```
#创建表tempt, 并且有结构ID和name, sex等, 是否允许空值等...
CREATE TABLE `mytest`.`tempt` (
  `ID` INT GENERATED ALWAYS AS (),
  `name` VARCHAR(8) NOT NULL,
  `sex` VARCHAR(2) NULL,
  PRIMARY KEY (`ID`));
```

- 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。

```
#查看表结构
DESCRIBE student;

#增加表结构默认设置中国
ALTER table student add 'nation' varchar(8)
NOT NULL COMMENT '中国';
```

- 练习用交互式的SQL语句分别完成以上操作。
  - 在交互式界面下创建如下图所示

Name: tempt 这里设置表名 Schema: mytest

| Column | Datatype   | PK                                  | NN                                  | UQ                       | BIN                      | UN                       | ZF                       | AI                       | G                                   | Default / Expression |
|--------|------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|----------------------|
| ID     | INT        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |                      |
| name   | VARCHAR(8) | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            |                      |
| sex    | VARCHAR(2) | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            |                      |

在这里设置属性值,设置是不是null,是不是主键等...

Column details 'sex'

Column Name: sex

Collation: Table Default

Comments:

Datatype:

Default:

Storage: ☒ VIRTUAL ☐ STORED

☐ Primary Key ☐ Not NULL ☐ Unique

☐ Binary ☐ Unsigned ☐ ZeroFill

☐ Auto Increment ☐ Generated

更多设置

Columns Indexes Foreign Keys Triggers Partitioning Options

点击这里生成

Apply Revert

- 表结构修改在这里,点击修改,最后apply.

| Field Types |          |        |        |         |               |               |
|-------------|----------|--------|--------|---------|---------------|---------------|
| #           | Field    | Schema | Table  | Type    | Character Set | Display Width |
| 1           | cno      | mytest | course | VARCHAR | utf8          |               |
| 2           | cname    | mytest | course | VARCHAR | utf8          |               |
| 3           | lhour    | mytest | course | INT     | binary        |               |
| 4           | credit   | mytest | course | INT     | binary        |               |
| 5           | semester | mytest | course | VARCHAR | utf8          |               |
|             |          |        |        |         |               |               |
|             |          |        |        |         |               |               |
|             |          |        |        |         |               |               |

- 对学生数据库中的三张表分别插入数据；

- insert into dict(class\_mark,mark\_v,mark\_d) values('对应字段的值',对应字段的数字,'对应字段的值')

#在数据表中插入一行数据

```
INSERT INTO course(cno,cname,lhour,credit,semester)
VALUES('C07','代码编译',40,3,'春');
```

- 对表中的数据进行修改；

- update 表名 set 字段=新值,... where 条件

# 更新表student中学号为30201的nation为'china'

```
UPDATE `mytest`.`student` SET `nation`='china' WHERE `sno`='30201';
```

- 对表中的数据进行删除操作

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name  
[WHERE where_definition]
```

```
[ORDER BY ...]
```

```
[LIMIT row_count]
```

- 在代码中删除如下

```
# 删除一行cno=c07的数据  
delete from course  
where cno='C07';
```

## 视图实验：视图的建立、修改、删除

- 建立如下视图： 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

```
# 创建视图  
create view studentable as  
select student.sname,student.sno,student.sex,student.classno,sc.grade  
from student  
left join sc on sc.sno=student.sno;
```

- 图像化的操作如下所示

```
1 • create view studenttable as
2 select student.sname,student.sno,student.sex,student.classno,sc.grade
3 from student
4 left join sc on sc.sno=student.sno;
5
6
7 #update student set nation='zhonuo'
8
9 #scwhere sno=30201;
10 # select * from course;
11
12 • select * from studenttable;
```

100% 36:4

Result Grid Filter Rows: Search Export:

|   | sname | sno   | sex | classno | grade |
|---|-------|-------|-----|---------|-------|
| ▶ | 吴磊    | 30201 | 男   | 302     | 40    |
|   | 吴磊    | 30201 | 男   | 302     | 88    |
|   | 吴磊    | 30201 | 男   | 302     | 93    |
|   | 袁青春   | 30202 | 男   | 302     | 40    |
|   | 袁青春   | 30202 | 男   | 302     | 40    |

Result Grid Form Editor

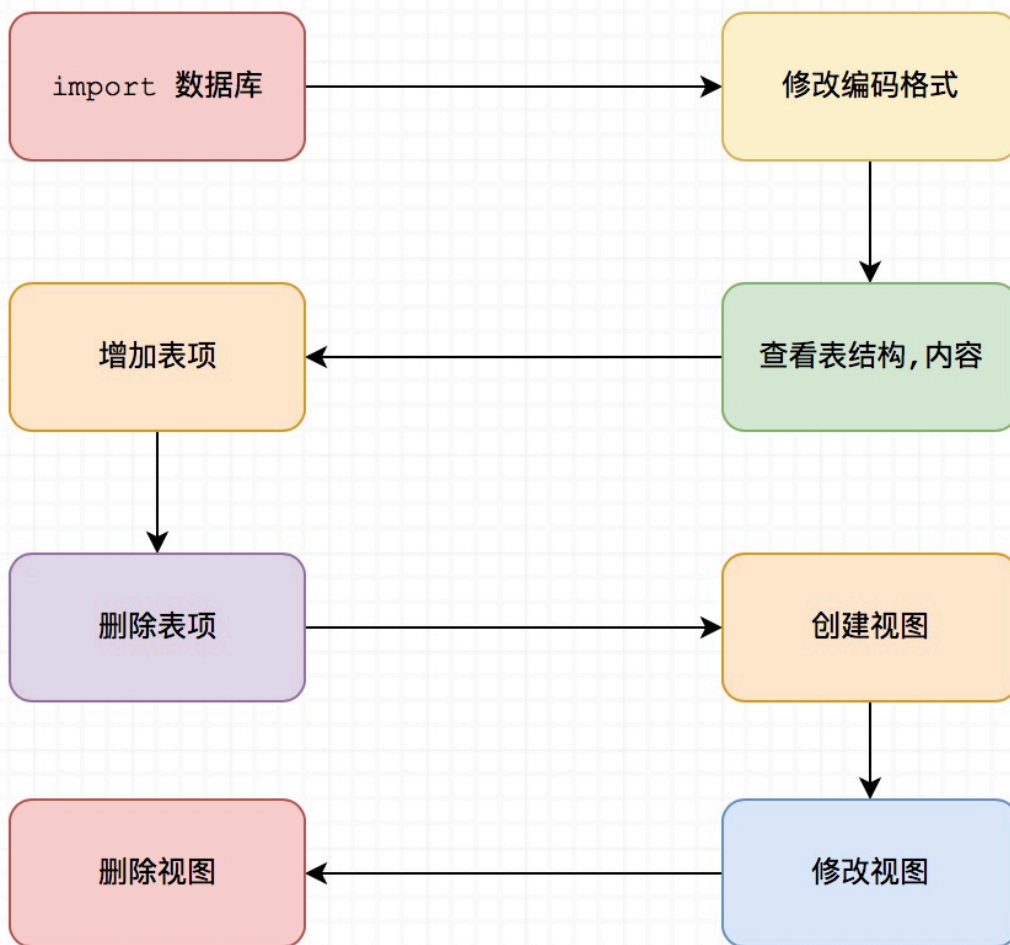
- 修改以上视图，增加学生所在班级信息。

```
# 修改视图
update studenttable set sname='zhaosun' where sno=10;
```

- 对以上视图删除。

```
DROP view tea_view;
```

## 实验步骤



## 实验结果及分析

- 实验结果总结和分析

- 在第二次数据库实验过程中,我首先使用老师给出的数据导入数据库,然后在导入的过程中遇到了没有指定数据库,然后修改sql源文件后才通过

```

-- Table structure for table `student`
--
USE mytest; 这里添加usemytest语句指定数据库
DROP TABLE IF EXISTS `student`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `student` (
  `sno` varchar(6) NOT NULL,
  `sname` varchar(6) DEFAULT NULL,
  `sex` varchar(2) DEFAULT NULL,
  `bdate` datetime DEFAULT NULL,
  `dept` varchar(8) DEFAULT NULL,
  `classno` varchar(4) DEFAULT NULL,
  PRIMARY KEY (`sno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8; 这里修改编码格式
/*!40101 SET character_set_client  = @saved_cs_client */;

--
-- Dumping data for table `student`
--

LOCK TABLES `student` WRITE;
/*!40000 ALTER TABLE `student` DISABLE KEYS */;
INSERT INTO `student` VALUES ('30201','İâÄÜ','ÄÐ','1980-01-02 00:00:00','µçÐÅ','302'),('30202','Ô-Çà´º','ÄÐ','1980-01-02 00:00:
/*!40000 ALTER TABLE `student` ENABLE KEYS */;
UNLOCK TABLES;

```

- 怎么在已建好的数据库上使用脚本语言建立这11张表格？
  - 解决：查阅课本，发现编写脚本时不能直接create table，应先明确使用什么数据库，即要在开始写清use 数据库名称go才可以在正确目录下建表。
- 问题：创建多表上的视图时遇到相同字段名称？
  - 解决：仔细阅读实验讲义，明确其中的关系，加上对于两张表中干扰小区名称相等的判断。
- 在向数据库导入数据时，会出现错误导致不成功情况？
  - 解决：按照老师给出的数据导入出错解决方法，知道在源数据文件中，某些属性列的属性可能被定义成字符串型，但在数据库表中，应该为整型，浮点型等数值类型，则将数据库中的该表删掉，重新导入数据库，即正确。

## • 实验小结

- 通过本次实验对SQL语句的使用更加熟练，掌握了创建/删除数据库表、修改数据库表的结构、向表中插入/删除数据以及数据更新等操作
- 学会了如何创建、修改、删除视图，包括单表、多表视图的增删改查熟悉了利用图形界面操作和SQL语句书写对数据库的操作。不仅如此，也对SQL Server软件编程环境更加熟悉，对于各字段的完整性约束以及整个E-R图的建立都有了更为深入的理解。
- 在使用SQL语句导入数据库时，遇见一些问题，往往跟导入的数据与数据库定义冲突有

关，解决这个问题