**上海大学 计算机学院**

**《数据库原理》实验一**

**姓名 学号**

**时间 指导教师 李晓强**

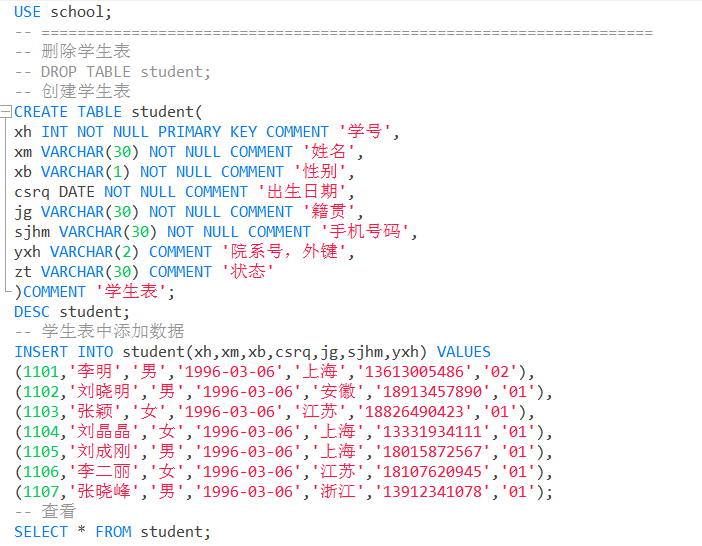
## 一、实验要求：

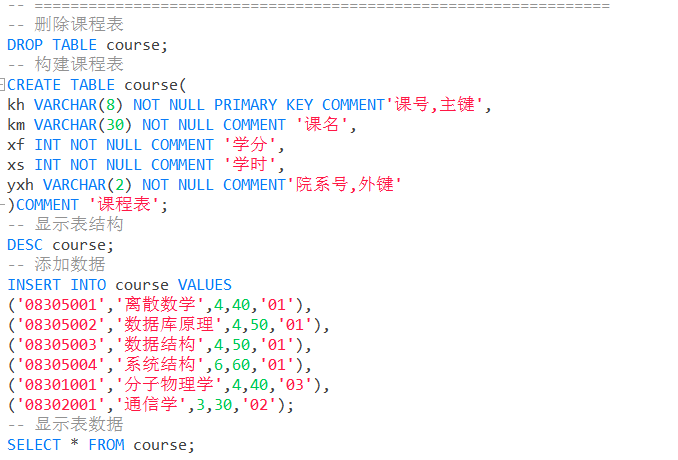
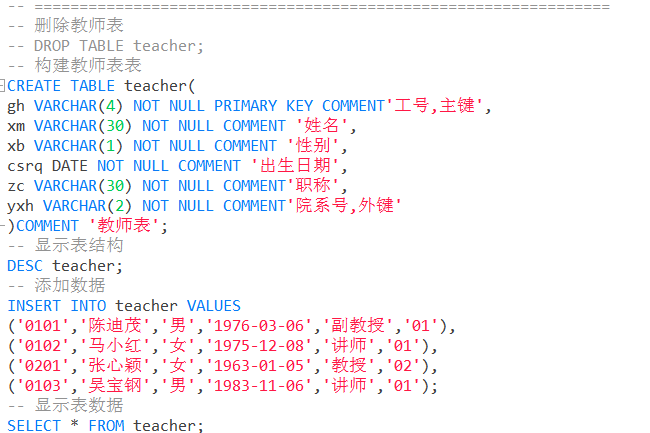
1. 为某学校教务管理系统数据库建立school数据库。
2. 在school下建立如下数据库表，根据表中数据选取合适的数据类型及宽度，设置各表的主键及表间外键联系；

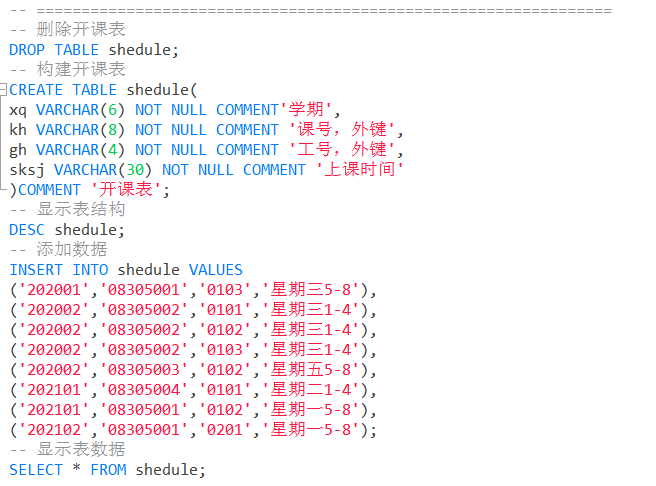
注意： 字段名是对应汉字字段名的汉语拼音第一个字母组合而成

学期 1901 表示 2019-2020 学年第一学期

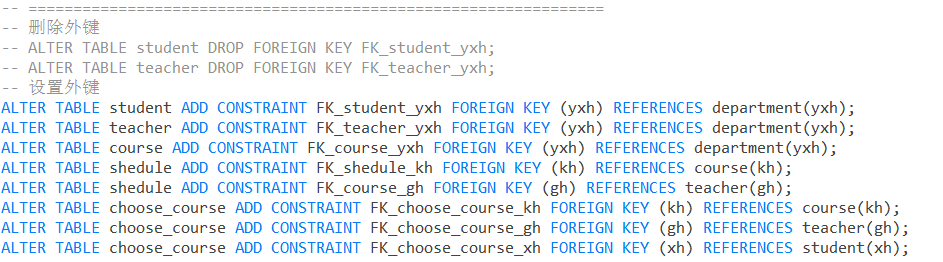
## **实验过程**

=

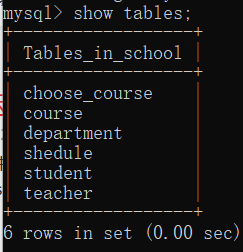
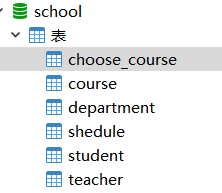




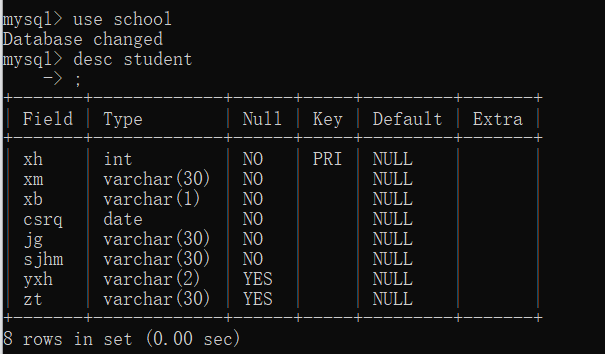




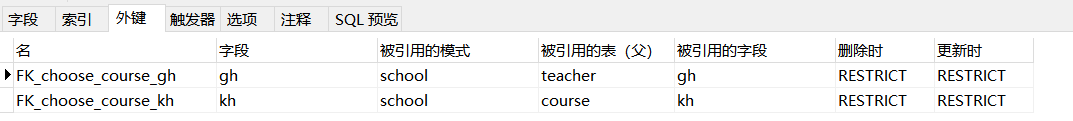
## **实验结果**

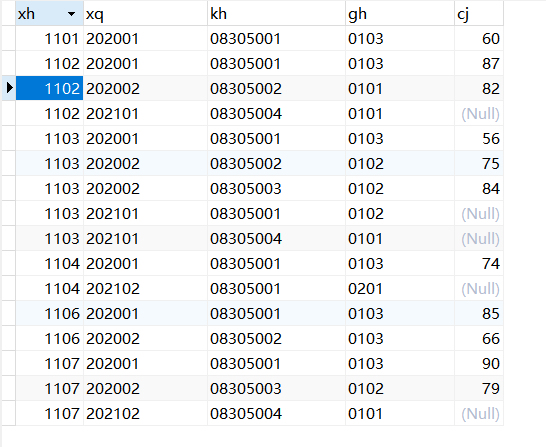


1. student表

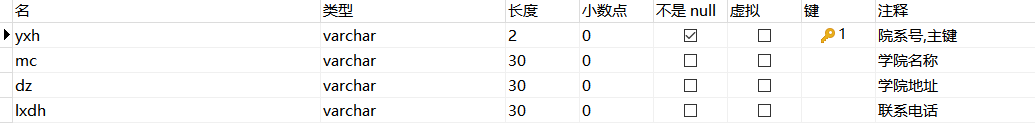


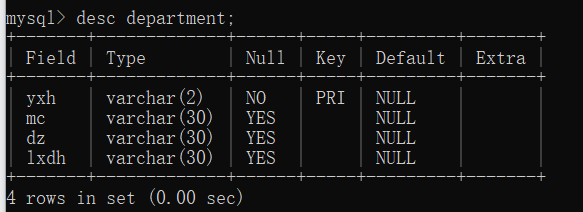
外键

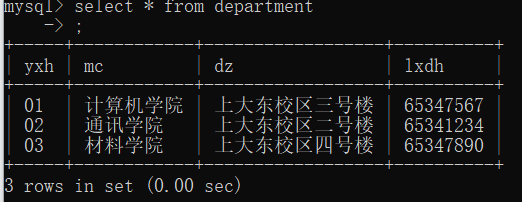


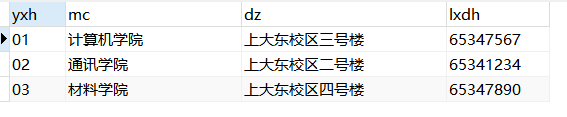


1. 院系表



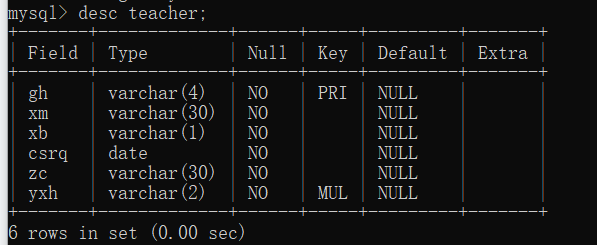






1. 教师表

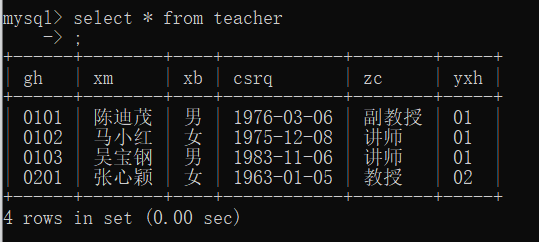




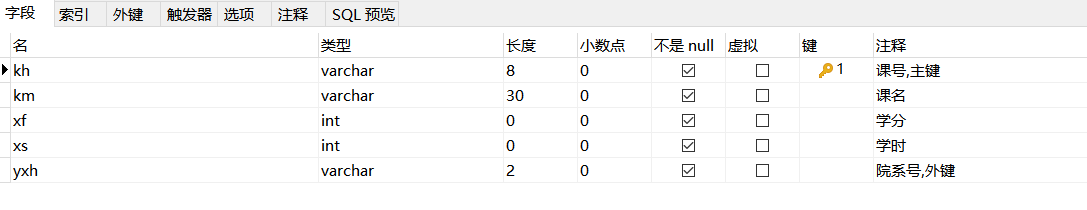
外键

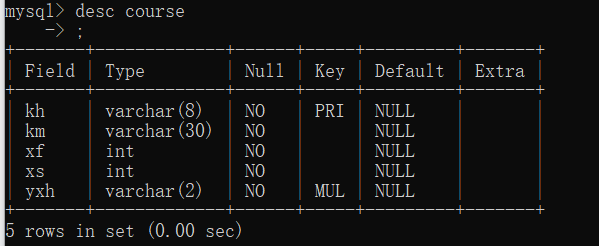




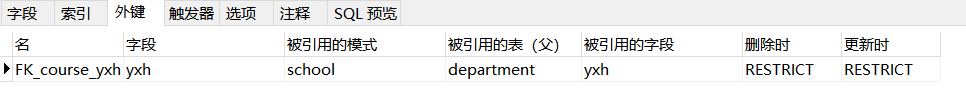


1. 课程表

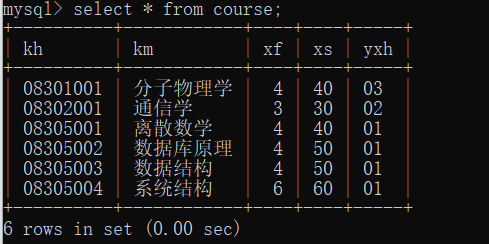




外键

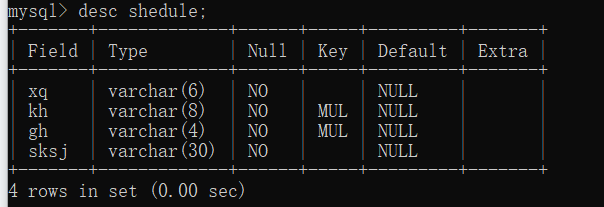




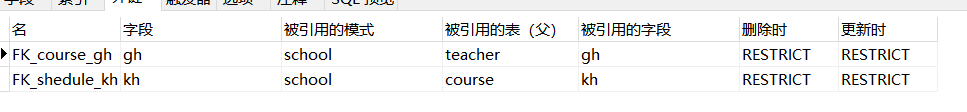


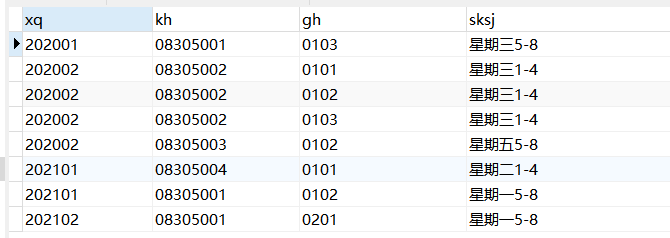
1. 开课表





外键：

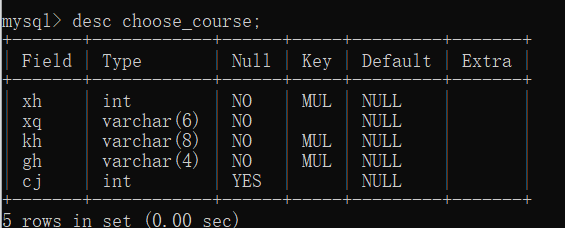




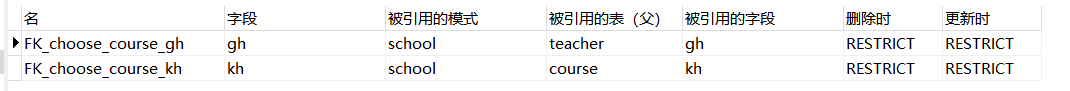


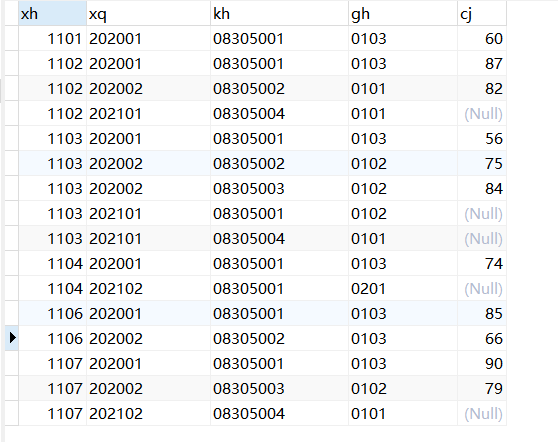
1. 选课表

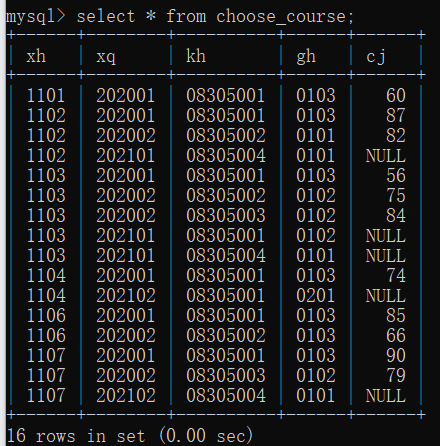




外键：

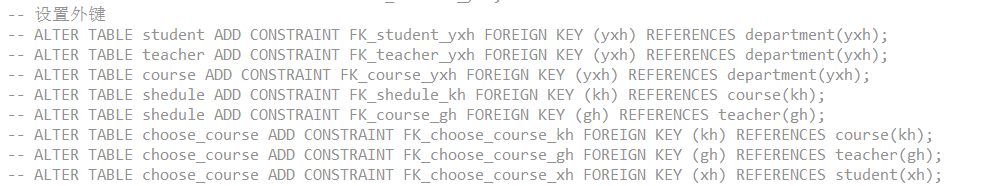






## **实验研讨**

1. 请说明school数据库中各表的主键和外键。同时说明：在学生表中的yxh字段输入“03”、“05”或在院系表中删除yxh为“02”、“03”记录，会有什么结果？为什么？



答：在学生表中主键是学号，外键是院系号。

在教师表中主键是工号，外键是院系号。

在课程表中，主键是课号，外键是院系号。

在开课表中，没有主键，外键是课号和工号。

在选课表中，没有主键，外建史课号、工号和学号。

1. 如果不定义外键会怎样？以school数据库为例说明。如果定义了外键，添加数据和删除数据是否有顺序限制，请以school数据库举例说明。

答：不定义外键会导致当某个表中的主键修改或删除时，其他表中仍存在，例如school数据库中若删除了xh（学号）字段，在选课表中的学号字段不会痛不删除，仍需要手动删除，导致重复劳动，效率低。

若定义了外键，在添加数据时，需要先在主表中添加相应字段，然后再在从表中添加。

例如在school数据库中，若想要在学生表中添加一个院系号为04的学生，则需要先在院系表中添加一个院系号为04的记录。

1. 如果O表加一个开课编号(标识码)会带来哪些影响(优缺点)?

答：优点： 1、能提高系统性能。

1. 保证数据唯一性。
2. 加速表和表之间连接的速度。
3. 减少检索时分组和排序的时间。

缺点： 1、维护索引需要时间。

1. 索引需要占物理空间，浪费空间。
2. 修改表时需要同步修改索引，浪费时间。
3. 什么是关系型数据库？什么是非关系型数据库？它们有什么区别？国产和国外分别各举1个典型产品简单介绍他们特点？

答：

关系型数据库是由多张能互相联接的二维行列表格组成的数据库。

非关系型数据库是不遵循关系代数模式的数据库。

扩展方式不同：关系数据库可纵向扩展，可提高处理能力，非关系数据库为天然分布式，通过更多的数据服务器来分担负载。关系型数据天然就是表格式的，因此存储在数据表的行和列中；数据表可以彼此关联协作存储，也很容易提取数据。而非关系型数据不适合存储在数据表的行和列中，而是大块组合在一起；非关系型数据通常存储在数据集中，就像文档、键值对或者图结构。

国产数据库有OpenGauss他是一款由华为开源的关系型数据库管理系统，它具有多核高性能、全链路安全性、智能运维等企业级特性，采用自底向上搜索路径算法。

国外数据库有SQL Server 是Microsoft 公司推出的关系型数据库管理系统，是一个全面的数据库平台，使用集成的商业智能 (BI)工具提供了企业级的数据管理。Microsoft SQL Server 数据库引擎为关系型数据和结构化数据提供了更安全可靠的存储功能，使得可以构建和管理用于业务的高可用和高性能的数据应用程序。

**上海大学 计算机学院**

**《数据库原理》实验二**

**姓名 学号**

**时间 指导教师**

## 一、实验要求：

1.（实验二）SQL Server 2016 操纵数据和使用视图（见教材P.300的查询部分）

2.对school数据库做如下查询：

1. 验证在1000万个以上记录时在索引和不索引时的查询时间区别。
2. 查询2011年进校年龄大于20岁的男学生的学号与姓名。
3. 检索刘晓明不学的课程的课程号。
4. 检索马小红老师所授课程的学年，学期，课程号，上课时间。
5. 查询计算机学院男生总评成绩及格、教授开设的课程的课程号、课名、开课教师姓名，按开课教师升序，课程号降序排序。
6. 检索学号比张颖同学大，年龄比张颖同学小的同学学号、姓名。
7. 检索同时选修了“08305001”和“08305002”的学生学号和姓名。

## **实验过程**

### 实验一：SQL Server 2016 操纵数据和使用视图（见教材P.300的查询部分）

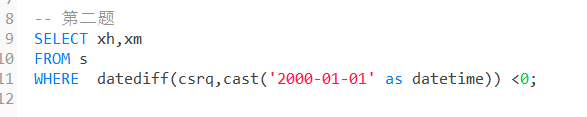
### 查询题目：

1. 检索马小红老师所授课程的课程号和课程名。



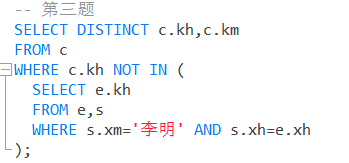


1. 检索年龄大于22岁的男学生的学号与姓名



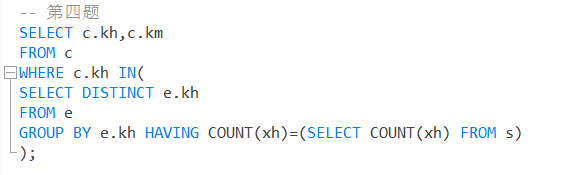


1. 检索李明不学的课程的课程号



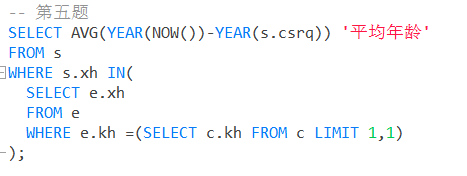


1. 检索全部学生都选修的课程号与课程名



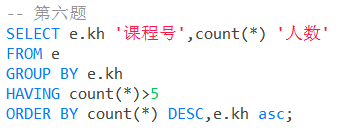


1. 求选修C2课程的学生的平均年龄



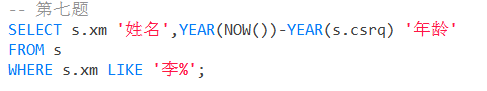


1. 统计每门课的学生选修人数（超过5人才统计）要求输出课程号和人数，查询结果按人数降序排列，若人数相同，按课程号升序排列



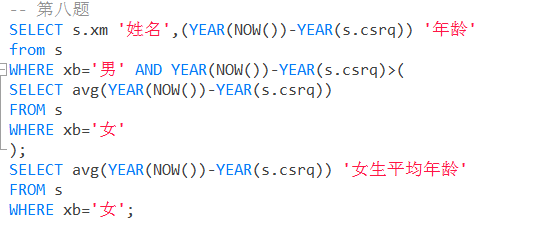


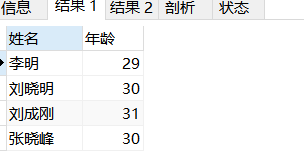
1. 检索姓名以李开头的学生的姓名与年龄





1. 求年龄大于女同学平均年龄的男同学的姓名与年龄





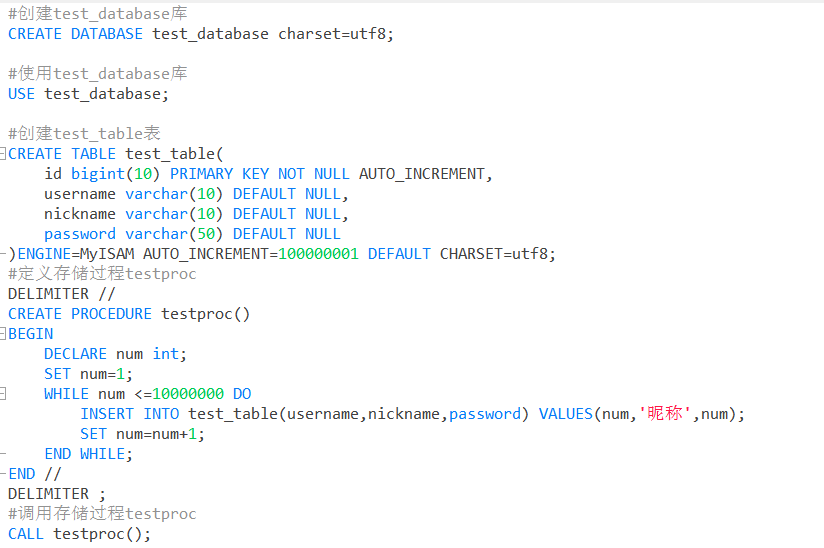


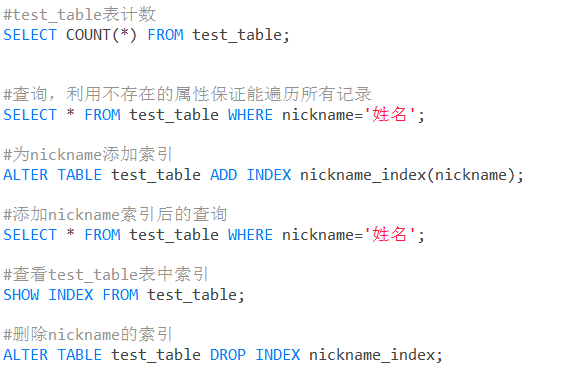
实验二：2.对school数据库做如下查询：

1. 验证在1000万个以上记录时在索引和不索引时的查询时间区别。

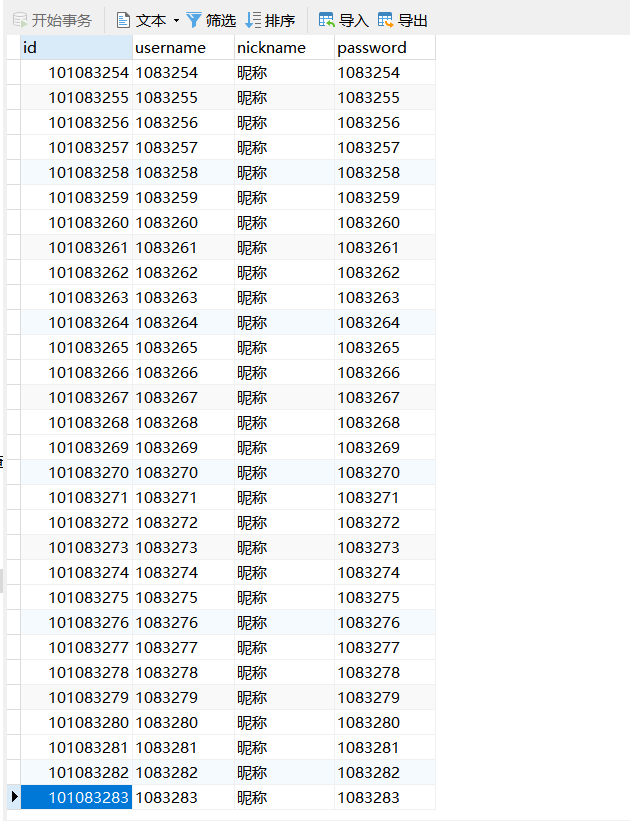
由于电脑原因，创建100万个数据需要半个小时，因此用100万数据进行验证

脚本：



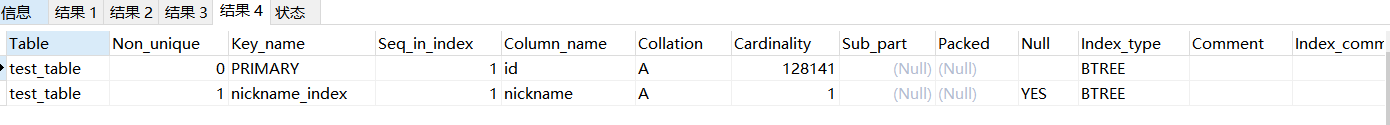
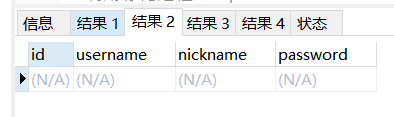


产生的数据





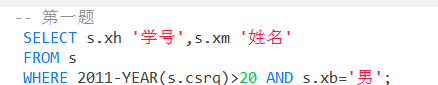
可以看到总共1092275条数据，添加了索引后，查询时间从5.288s骤降到0s，提升巨大



可以发现，在某些场合下，合理的适用索引能够大幅度的降低查询时间

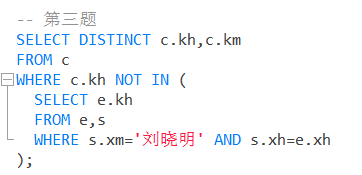
1. 查询2011年进校年龄大于20岁的男学生的学号与姓名。

无2011年进校年龄大于20岁的男学生



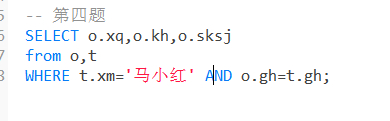


1. 检索刘晓明不学的课程的课程号。





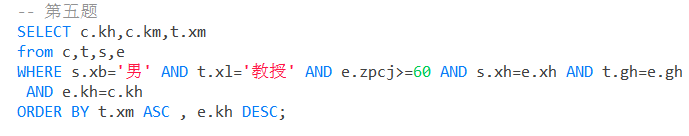
1. 检索马小红老师所授课程的学年，学期，课程号，上课时间。





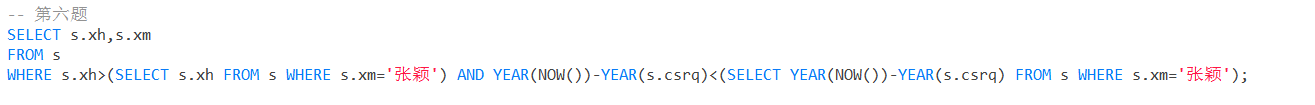
1. 查询计算机学院男生总评成绩及格、教授开设的课程的课程号、课名、开课教师姓

名，按开课教师升序，课程号降序排序。



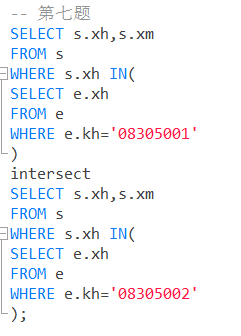


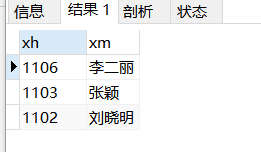
1. 检索学号比张颖同学大，年龄比张颖同学小的同学学号、姓名。





1. 检索同时选修了“08305001”和“08305002”的学生学号和姓名。





**上海大学 计算机学院**

**《数据库原理》实验三**

**姓名 学号**

**时间 指导教师**

## 一、实验要求：

1.（实1.（实验二）SQL Server 2016 操纵数据和使用视图（见教材P.300的更新部分）

2.对school数据库做如下操作：

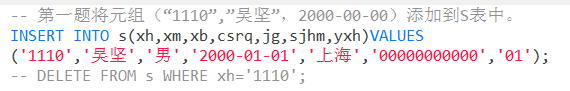
1. 查询每个学生选课情况（包括没有选修课程的学生）。
2. 检索所有课程都选修的的学生的学号与姓名。
3. 检索选修课程包含1106同学所学全部课程的学生学号和姓名。
4. 查询每门课程中分数最高的学生学号和学生姓名。
5. 查询年龄小于本学院平均年龄，所有课程总评成绩都高于所选课程平均总评成绩的学生学号、姓名和平均总评成绩，按年龄排序。

## **实验过程**

### 实验一：SQL Server 2016 操纵数据和使用视图（见教材P.300的更新部分）

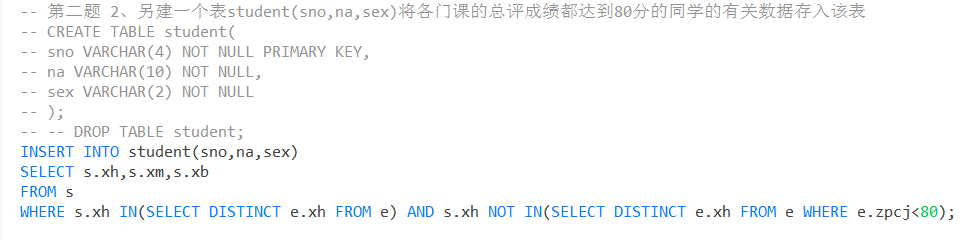
### 更新题目：

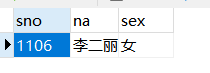
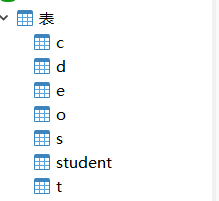
1. 将元组('1110','吴坚','男','2000-01-01','上海','00000000000','01')添加到S表中。





1. 另建一个表student(sno,na,sex)将各门课的总评成绩都达到80分的同学的有关数据存入该表

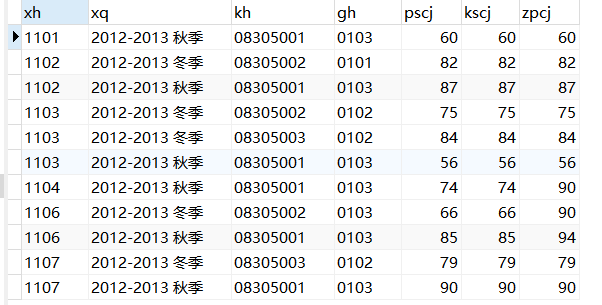




经过确认，发现选过课的并且所有课均80分以上的学生只有1106，李二丽。

1. 从e表中删除没有考试成绩的元组





1. 删除学生李明的选修课程和成绩





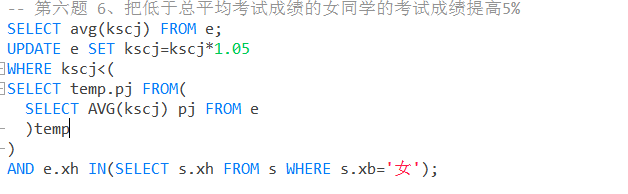
发现学号为1101的李明的元组不见了

1. 把数据结构课程考试成绩不合格的总评成绩改成空值





1. 把低于总平均考试成绩的女同学的考试成绩提高5%

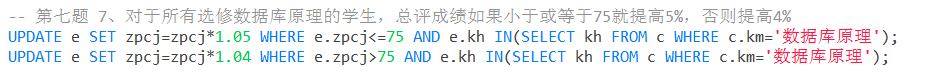


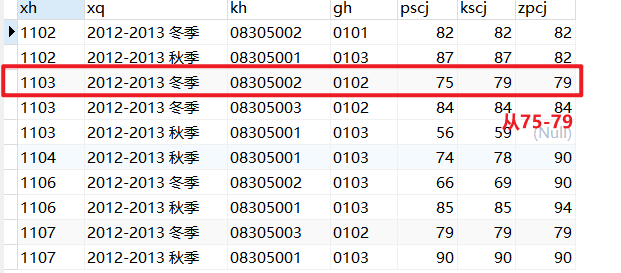




考试成绩低于77.8的都提升了5%。

1. 对于所有选修数据库原理的学生，总评成绩如果小于或等于75就提高5%，否则提高4%

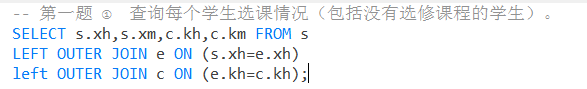






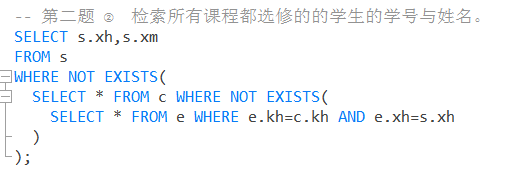
实验二：2.对school数据库做如下查询：

1. 查询每个学生选课情况（包括没有选修课程的学生）。





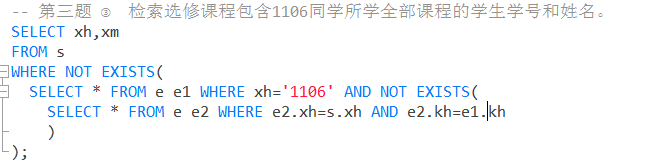
1. 检索所有课程都选修的的学生的学号与姓名。

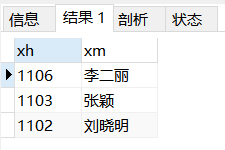




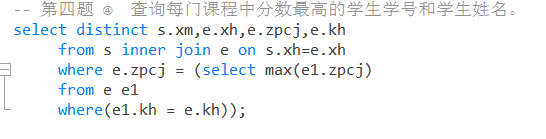
不存在所有课程都选修的学生。

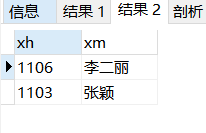
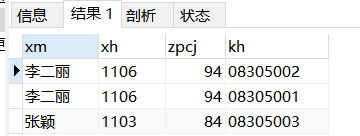
1. 检索选修课程包含1106同学所学全部课程的学生学号和姓名。





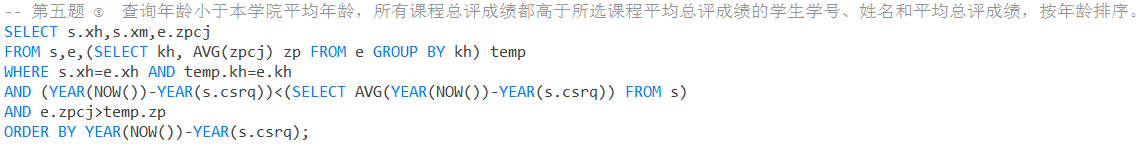
1. 查询每门课程中分数最高的学生学号和学生姓名。





验证：经过判断，发现08305001和08305002的最高峰均为94，是1106李二丽同学，而08305003则是1103张颖同学。

1. 查询年龄小于本学院平均年龄，所有课程总评成绩都高于所选课程平均总评成绩的学生学号、姓名和平均总评成绩，按年龄排序。





只有一个人符合

**上海大学 计算机学院**

**《数据库原理》实验四**

**姓名 学号**

**时间 指导教师**

## 一、实验要求：

1.（实验二）SQL Server 2016 操纵数据和使用视图（见教材P.300的视图部分）

2.对school数据库做如下操作：

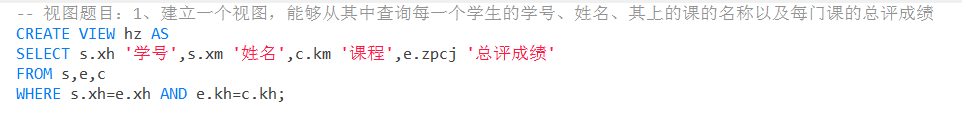
1. 建立计算机学院总评不及格成绩学生的视图，包括学生学号、姓名、性别、手机、所选课程和成绩。
2. 在E表中插入记录，把每个学生没学过的课程都插入到E表中，使得每个学生都选修每门课。
3. 求年龄大于所有女同学年龄的男学生姓名和年龄。
4. 在E表中修改08305001课程的平时成绩，若成绩小于等于75分时提高5%，若成绩大于75分时提高4%。
5. 删除没有开课的学院。
6. 查询优、良、中、及格、不及格学生人数

## **实验过程**

### 实验一：SQL Server 2016 操纵数据和使用视图（见教材P.300的视图部分）

### 视图题目：

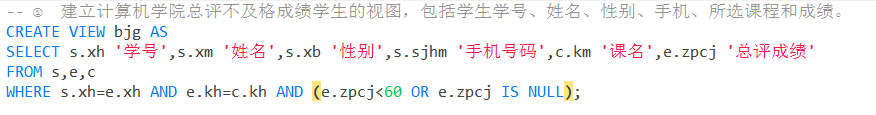
1. 建立一个视图，能够从其中查询每一个学生的学号、姓名、其上的课的名称以及每门课的总评成绩。

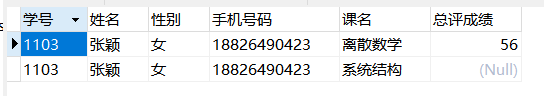




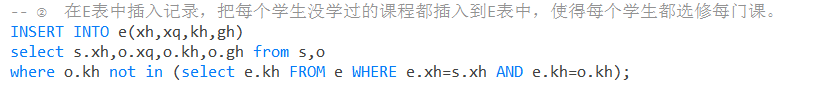
## 2.对school数据库做如下操作：

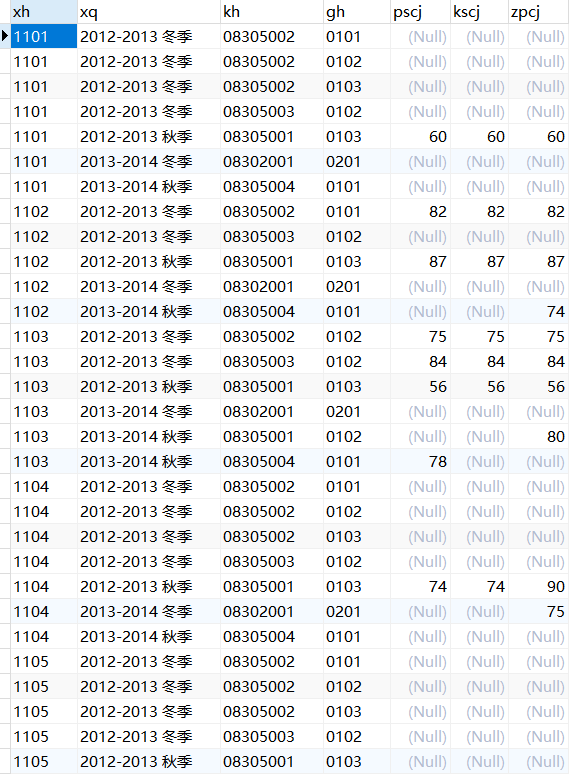
1. 建立计算机学院总评不及格成绩学生的视图，包括学生学号、姓名、性别、手机、所选课程和成绩。





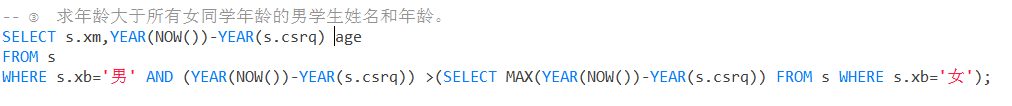
1. 在E表中插入记录，把每个学生没学过的课程都插入到E表中，使得每个学生都选修每门课。

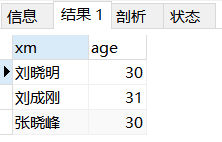




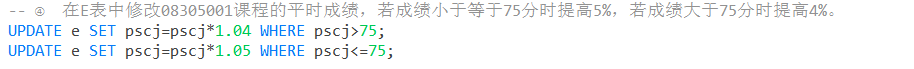
可以看到，所有学生都选修了全部的课程

1. 求年龄大于所有女同学年龄的男学生姓名和年龄。

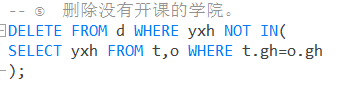


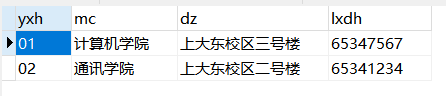
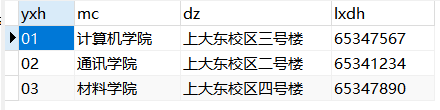


1. 在E表中修改08305001课程的平时成绩，若成绩小于等于75分时提高5%，若成绩大于75分时提高4%。



1. 删除没有开课的学院。





可以看到，材料学院被删除了

1. 查询优、良、中、及格、不及格学生人数

