



# 数据库系统实验报告

学 院 计算机学院

专 业

班 级

学 号

姓 名

指导教师 谢 锐

**2021 年 9 月 26 日**

计算机 学院 专业 班 学号：

姓名： 教师评定：

实验 **一** 题目

实验 **二** 题目

实验 **三** 题目

实验四题目

实验五题目

实验六题目

实验七题目

实验八题目

**实验平台：SQL SERVER 2019**

计算机 学院 专业 班 学号：

姓名： 协作者： 教师评定：

实验题目

# 实验二 数据库及基本表的建立

**实验项目名称：数据库及基本表的建立**

**实验项目性质：验证性**

**所属课程名称：《数据库系统概论》**

**实验计划学时：课内3学时**

## 一、实验目的

1、掌握 SQL SERVER 的查询分析器和企业管理器的使用；2、掌握创建数据库和表的操作；

## 二、实验内容和要求

## 1、分别使用 SQL 语句、企业管理器（Enterprise Manager）创建数据库；

## 2、使用 SQL 语句、企业管理器（Enterprise Manager）创建数据库表；

## 三、实验主要仪器设备和材料

1、计算机及操作系统：PC机，Windows10

2、数据库管理系统：SQL Server2019

## 四、实验方法、步骤及结果测试

## 创建一个教学管理数据库 SC，其描述的信息有：学生信息、课程信息、教师信息、学生选课成绩、授课信息、班级信息、系部信息、专业信息。

## 创建：student 表(学生信息表)、course 表（课程信息表)、teacher 表（教师信息表)、

## student \_course 表（学生选课成绩表)、teacher\_course 表（教师上课课表）等。

## 五、实验过程

## 创建数据库：

2、创建基本表

利用查询分析器，使用 SQL 语句方式创建方式将下面各表建立到教学管理数据库中。

实现代码及截图：

## create table department(

## dno char(6),

## dept\_name char(20)not null,

## header char(8)

## primary key(dno))

## create table speciality(

## spno char(8),

## dno char(6)not null,

## spname char(20)not null

## primary key(spno)

## foreign key(dno)references department)

## create table teacher(

## t\_no char(8),

## t\_name char(8)not null,

## t\_sex char (2),

## t\_birthday datetime,

## dno char(6),

## tech\_title char(10)

## primary key(t\_no),

## foreign key(dno)references department)

## create table student(

## s\_no char(8),

## sname char(8)not null,

## sex char(2),

## sbirthday datetime,

## dno char(6),

## spno char(8),

## class\_no char(4),

## primary key(s\_no),

## foreign key(dno) references department,

## foreign key(spno) references speciality )

## create table course(

## cno char(10),

## cname char(20)not null,

## spno char(8),

## ctno tinyint,

## lecture tinyint,

## experiment tinyint,

## semester tinyint,

## credit tinyint

## primary key(cno),

## foreign key(spno)references speciality)

## create table student\_course(

## s\_no char(8),

## tcid smallint,score tinyint

## primary key(tcid,s\_no)

## foreign key(s\_no)references student on delete cascade)

## create table teacher\_course(

## tcid smallint,

## t\_no char(8),

## spno char(8),

## class\_no char(4),

## cno char(10)not null,

## semester char(6),

## schoolyear char(10)

## primary key(tcid),

## foreign key(t\_no)references teacher,

## foreign key(spno)references speciality,

## foreign key(cno)references course)

## create table class(

## spno char(8),

## class\_no char(4),

## header char(8)

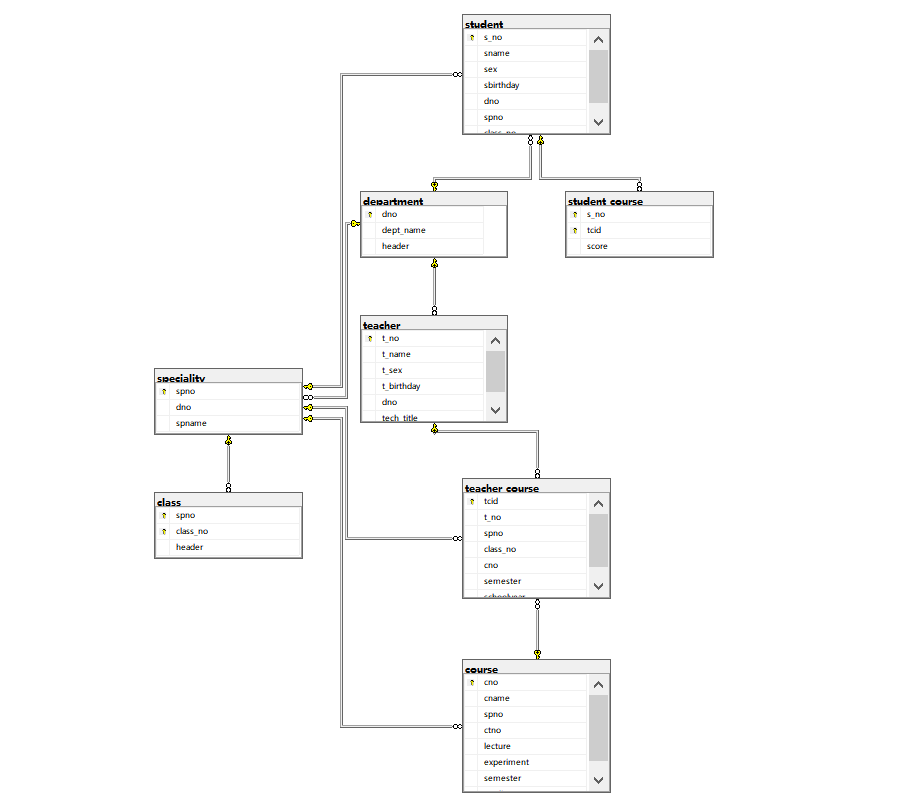
## primary key(spno,class\_no)

## foreign key(spno)references speciality)

## SQL 语句及执行结果截图显示

## 

1. 查看各数据表之间的关系，生成数据库关系图。
2. 生成数据库关系图截图显示



5、利用查询分析器修改上述各表。

(1)、用 INSERT 语句向各个表中插入数据录入 5 条记录。录入时注意体会外键约束。

实现代码及截图

insert into department values

('01','Com.Sci','Wu'),

('02','Music','John'),

('03','Bology','Sam'),

('04','Dance','Liu'),

('05','Esport','Gogo')

insert into speciality values

('01','01','计算机'),

('02','02','音乐'),

('03','03','生物'),

('04','04','舞蹈'),

('05','05','电竞')

insert into teacher values

('51','Xie','男','2000-01-01 01:01:01','01','DBMS')

('52','Li','女','1990-02-02 02:02:02','02','pop')

('53','Yang','男','1992-03-03 03:03:03','03','zom')

('54','Zhang','女','1996-04-04 04:04:04','04','tridition')

('55','邹佳豪','男','2000-05-05 05:05:05','05','lolm')

insert into student values

('311901','Jassisse','男','2000-01-01','01','01','01'),

('311902','Sam','男','2001-02-02','02','02','02'),

('311903','Gold','男','2001-02-05','03','03','03'),

('311904','Mishel','女','2000-04-01','04','04','04'),

('311905','Mache','女','2001-05-02','05','05','05')

insert into course values

('1001','Basic','01','1','11','66','66','66'),

('1002','Chinese','02','2','22','77','77','77'),

('1003','Math','03','3','33','88','88','88'),

('1004','ENG','04','4','44','99','99','99'),

('1005','BIO','05','5','55','98','98','98')

insert into student\_course values

('311901','50','100'),

('311902','49','99'),

('311903','48','98'),

('311904','47','97'),

('311903','46','96')

insert into teacher\_course values

('21','51','01','101','1001','5555','2021'),

('22','52','02','102','1002','5556','2021'),

('23','53','03','103','1003','5557','2020'),

('24','54','04','104','1004','5558','2021'),

('25','55','05','105','1005','5559','2022')

insert into class values

('01','101','Levis'),

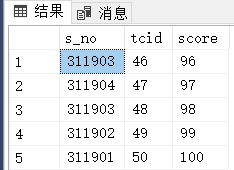
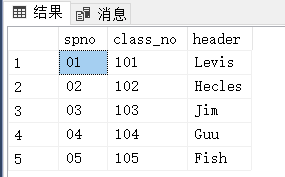
('02','102','Hecles'),

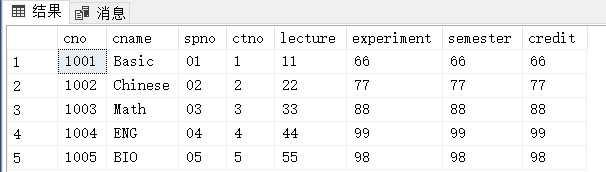
('03','103','Jim'),

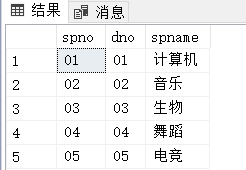
('04','104','Guu'),

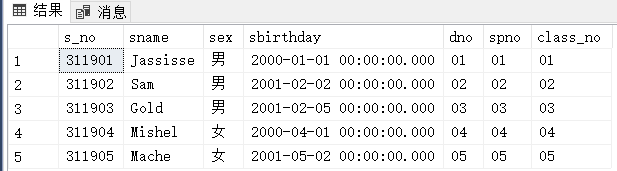
('05','105','Fish')

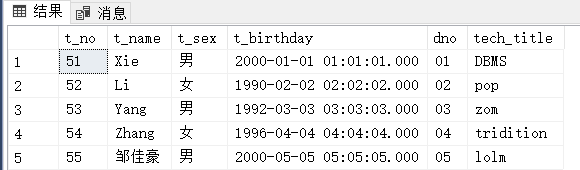
实验结果截图显示（插入数据的表格）

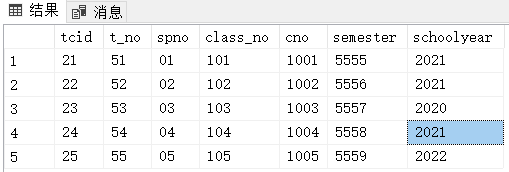










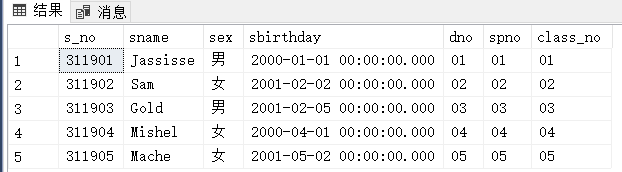


1. 、用 UPDATE 语句更改 student 表中数据；

修改student表中学号为311902的学生性别为女

Sql实现代码：update student set sex='女' where s\_no='311902'



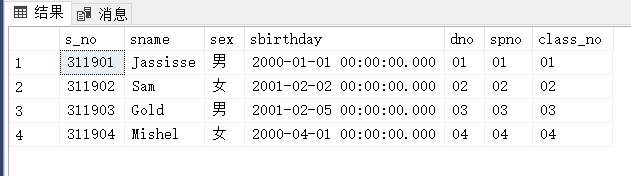


1. 、用 DELETE 语句删除 student 表中数据；

删除student表中class\_no为05的学生数据

Sql实现代码delete from student where class\_no='05'

## 



## 实验中出现的问题及解决办法

问题：实验建表时，一开始没有按照逻辑顺序建表，导致主键外键发生错误

解决办法：建表前要理清建表的逻辑顺序，整理为建系表，专业表，老师表，学生表，课程表，选课表，老师选课表，班级表

1. 思考题
2. 说明数据库中的表和数据文件的关系

答：数据表是DB中建立的table。数据文件是存储整个数据库，除了表数据，还有视图、过程、函数、登录用户信息等。

数据文件存放着数据库中存储的数据，且一个数据库只有一个数据文件。表作为数据库中的存储单位，存储在数据文件中。

1. 数据库中的日志文件能否单独修改

答：不能，因为日志文件记录了数据库的数据，单独修改可能会造成日志文件和数据库的不一致，影响正常运行。

计算机 学院 专业 班 学号：

姓名： 教师评定：

实验题目

# 实验三 查询数据库

**实验项目名称：使用SQL语言查询管理数据**

**实验项目性质：基础性**

**所属课程名称：《数据库系统概论》**

**实验计划学时：课内3学时**

## 一、实验目的

1、掌握 SQL 语句的基本使用方法，学习如何使用SQL语句来查询

2、掌握基本的SELECT语句

3、掌握复杂的SELECT查询

## 二、实验内容和要求

使用SQL查询分析器查询数据，掌握SELECT语句的完整结构，包括简单查询，嵌套查询，连接查询等基本实现方法。

## 三、实验主要仪器设备和材料

1、计算机及操作系统：PC机，Windows10

2、数据库管理系统：SQL Server2019

## 四、实验方法、步骤及结果测试

## 1、对实验二建立的数据库表进行查询

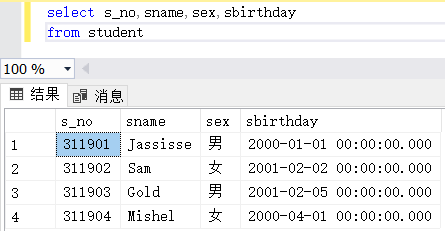
（1）查询全部学生的学号，姓名，性别和出生年月

SQL语句及结果显示：

select s\_no,sname,sex,sbirthday

from student

查询结果截图显示：



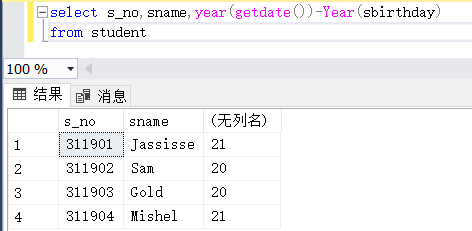
1. 查询全体学生的学号姓名和年龄

SQL语句及结果显示：

select s\_no,sname,year(getdate())-Year(sbirthday)

from student

查询结果截图显示：



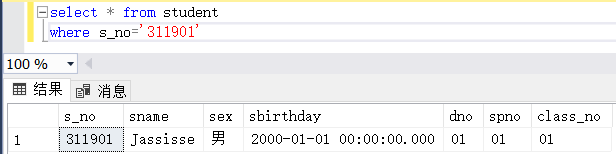
（3）查询院系编号为‘XXXXX’的学生信息

SQL语句及结果显示：

select \* from student

where s\_no='311901'

查询结果截图显示：



（4）查询XXX专业（专业代码为XXX）并且班级代码为XXX的学生学号，姓名和出生日期。

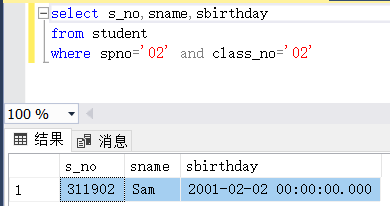
SQL语句及结果显示：

select s\_no,sname,sbirthday

from student

where spno='02' and class\_no='02'

查询结果截图显示：



1. 查询在XXX年月后出生的XX学院（院系编号为XX）的学生学号，姓名，性别和出生日期。

SQL语句及结果显示：

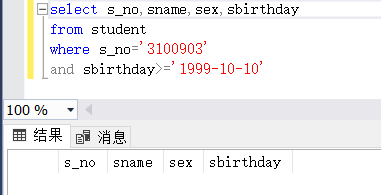
select s\_no,sname,sex,sbirthday

from student

where s\_no='3100903'

and sbirthday>='1999-10-10'

查询结果截图显示：未插入符合的数据



1. 查询全部学生的学号，姓名，性别和出生日期按照出生日期升序排列

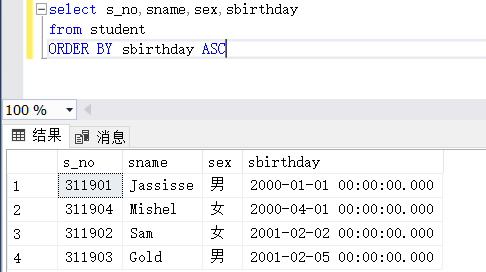
SQL语句及结果显示：

select s\_no,sname,sex,sbirthday

from student

ORDER BY sbirthday ASC

查询结果截图显示：



1. 嵌套查询：
2. 查询全部学生的学号，姓名，性别，所在院系名称和专业名称

SQL语句及结果显示：

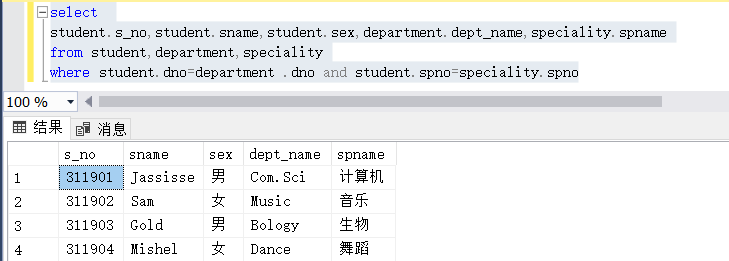
select

student.s\_no,student.sname,student.sex,department.dept\_name,speciality.spname

from student,department,speciality

where student.dno=department .dno and student.spno=speciality.spno

查询结果截图显示：



1. 查询选修了课程1（上课编号）的学生的学号，姓名，专业名称和成绩

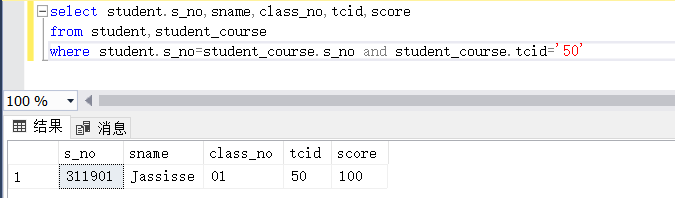
SQL语句及结果显示：

select student.s\_no,sname,class\_no,tcid,score

from student,student\_course

where student.s\_no=student\_course.s\_no and student\_course.tcid='50'

查询结果截图显示：



1. 查询学生不及格的情况列出不及格学生的学号，姓名和不及格的课程名称

注：此处未插入分数低于60分的学生，故改为了查询分数低于99的学生

SQL语句及结果显示：

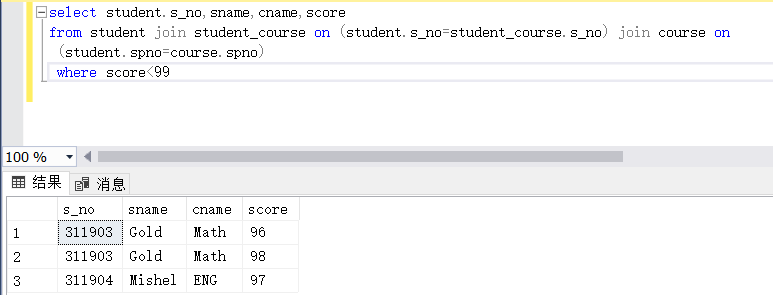
select student.s\_no,sname,cname,score

from student join student\_course on (student.s\_no=student\_course.s\_no) join course on

(student.spno=course.spno)

where score<99

查询结果截图显示：



1. 嵌套查询：
2. 查询没有选修了课程48的学生，列出学生的学号和姓名。

SQL语句及结果显示：

select student.s\_no,sname

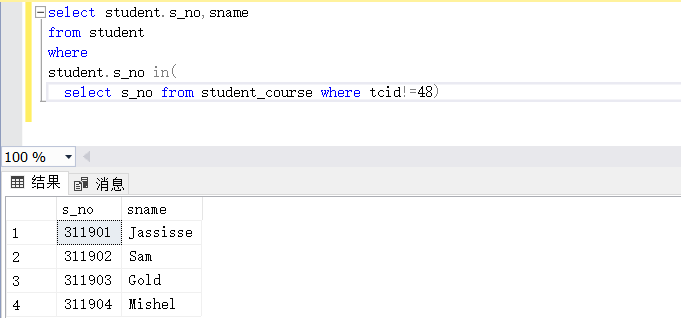
from student

where

student.s\_no in(

select s\_no from student\_course where tcid!=48)

查询结果截图显示：



1. 查询每门课都是80分以上的学生的学号和姓名

SQL语句及结果显示：

select student.s\_no,sname

from student

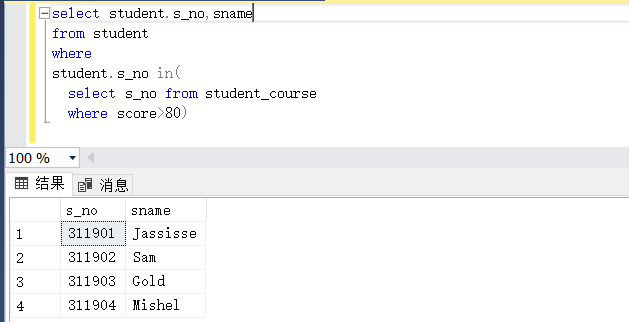
where

student.s\_no in(

select s\_no from student\_course

where score>80)

查询结果截图显示：



2、选用 Northwind 数据库进行查询

（1）、对 NothWind. Products 表进行简单查询；在查询分析器在窗口下用 SELECT 语句完成单表查询：

查询所有 Products 的详细记录；

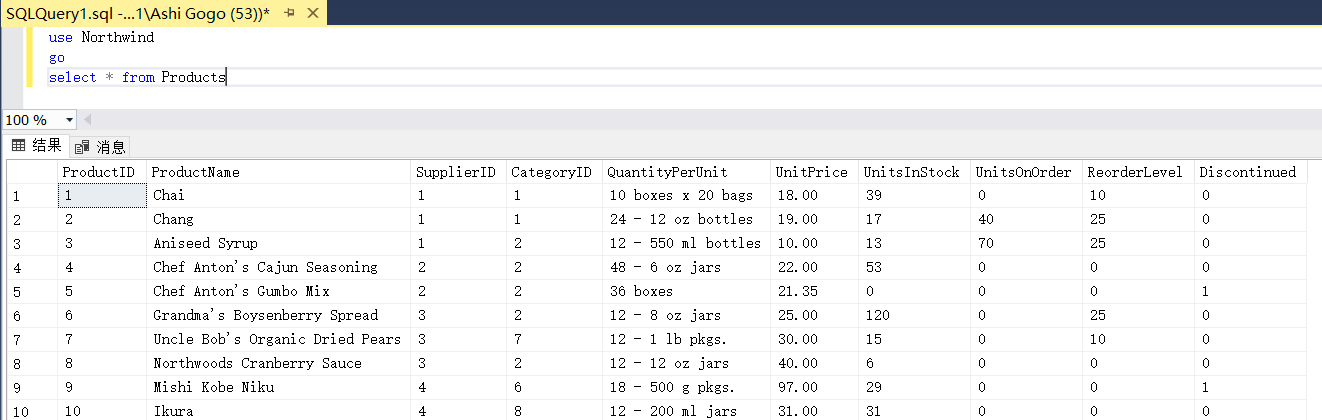
SQL语句及结果显示：

use Northwind

go

select \* from Products

查询结果截图显示：



查询单价（UnitPrice）小于 20 的 Products；

SQL语句及结果显示：

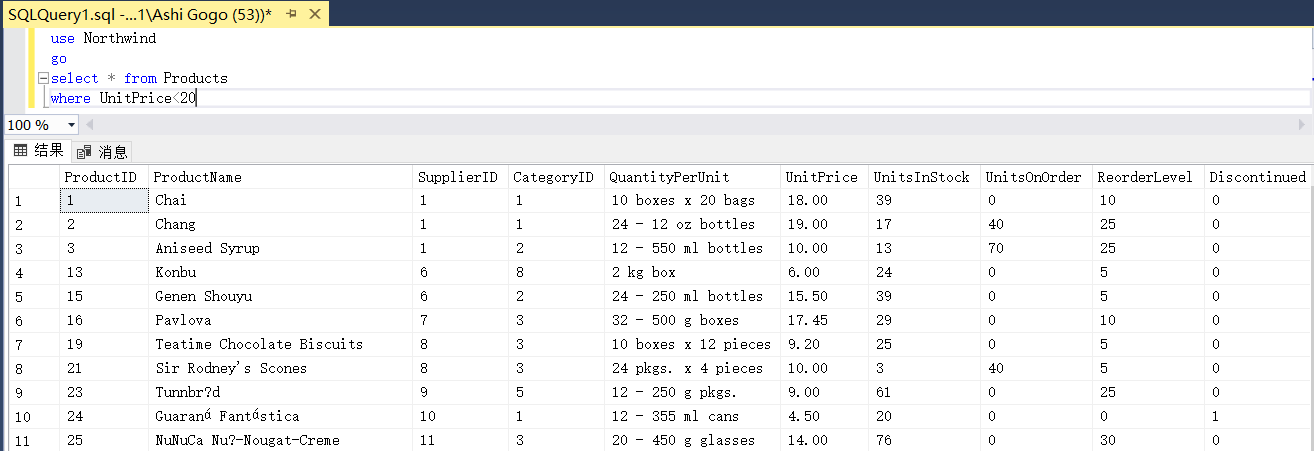
use Northwind

go

select \* from Products

where UnitPrice<20

查询结果截图显示：



查询 Products 中最高单价（UnitPrice）是多少；

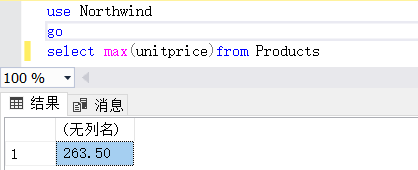
SQL语句及结果显示：

use Northwind

go

select max(unitprice)from Products

查询结果截图显示：



（2）、在查询分析器在窗口下用 SELECT 语句完成连接（嵌套）查询：

查询所有被订购过得 Products 的 ProductsID 和 ProductName；

SQL语句及结果显示：

use Northwind

go

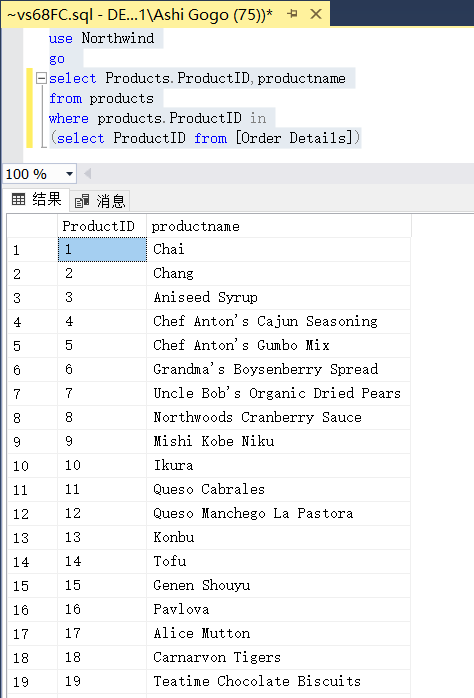
select Products.ProductID,productname

from products

where products.ProductID in

(select ProductID from [Order Details])

查询结果截图显示：



查询所有被CustomerID为AROUT客和订购过的ProductsID和ProductName；

SQL语句及结果显示：

use Northwind

go

select Products.ProductID,productname

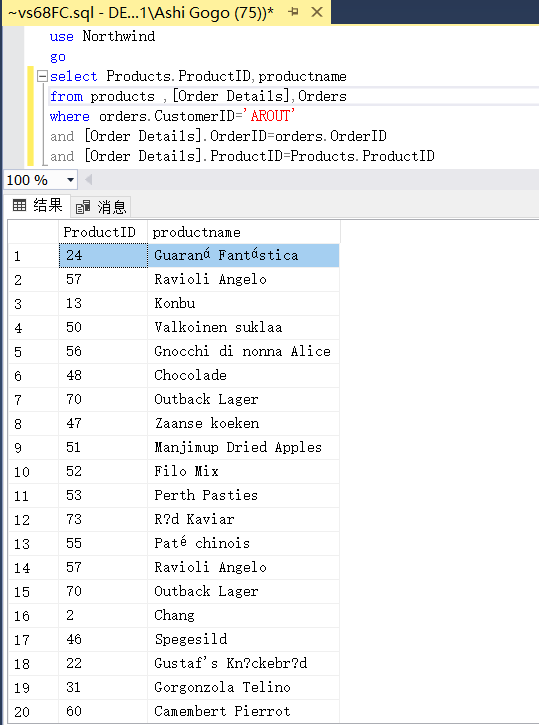
from products ,[Order Details],Orders

where orders.CustomerID='AROUT'

and [Order Details].OrderID=orders.OrderID

and [Order Details].ProductID=Products.ProductID

查询结果截图显示：



查询所有被CustomerID为AROUT客和订购过，且单价在20 以上的ProductsID

和 ProductName；

SQL语句及结果显示：

use Northwind

go

select Products.ProductID,productname

from products ,[Order Details],Orders

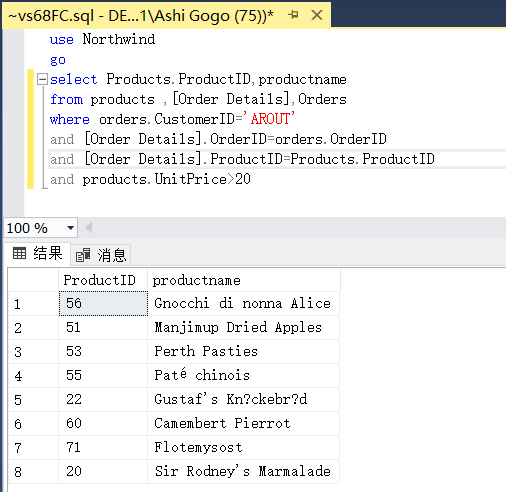
where orders.CustomerID='AROUT'

and [Order Details].OrderID=orders.OrderID

and [Order Details].ProductID=Products.ProductID

and products.UnitPrice>20

查询结果截图显示：



查询 Products 中单价（UnitPrice）最高的 Products 的资料；

SQL语句及结果显示：

use Northwind

go

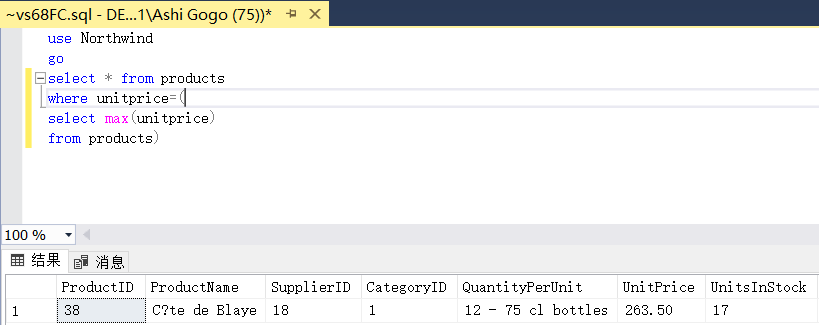
select \* from products

where unitprice=(

select max(unitprice)

from products)

查询结果截图显示：



五、实验中出现的问题及解决方案

答：

在本次实验中，前半部分实验使用了上一个实验的实验数据，与此次实验部分要求不符，故适配要求做了一些数据的修改，以达到查询目的，不影响实验进行。

插入数据库northwind时，直接导入数据库较慢，采用了还原数据库的办法进行导入。

六、思考题

1、连接查询分哪几类？各有什么特点？

答：

连接查询有：  
1.交叉连接  
交叉连接即笛卡儿乘积，是指两个关系中所有元组的任意组合。一般情况下，交叉查询是没有实际意义的。  
2.内连接查询  
内连接是一种最常用的连接类型。内连接查询实际上是一种任意条件的查询。使用内连接时，如果两个表的相关字段满足连接条件，则从这两个表中提取数据并组合成新的记录，也就是说，在内连接查询中，只有满足条件的元组才能出现在结果关系中。  
3.自连接查询  
如果在一个连接查询中，涉及到的两个表都是同一个表，这种查询称为自连接查询。同一张表在FROM字句中多次出现，为了区别该表的每一次出现，需要为表定义一个别名。自连接是一种特殊的内连接，它是指相互连接的表在物理上为同一张表，但可以在逻辑上分为两张表。  
4.外连接查询  
内连接的查询结果都是满足连接条件的元组。但是，有时我们也希望输出那些不满足连接条件的元组的信息。比如，我们想知道每个学生的选课情况，包括已经选课的学生（这部分学生的学号在学生表中有，在选课表中也有，是满足连接条件的），也包括没有选课的学生（这部分学生的学号在学生表中有，但在选课表中没有，不满足连接条件），这时就需要使用外连接。外连接是只限制一张表中的数据必须满足连接条件，而另一张表中的数据可以不满足连接条件的连接方式。

2、进行连接查询时应注意哪些问题？

答：

连接多个表时，可以认为他们被连成了一个表。尽管没有创建一个物理表，SQL引擎创建了很多虚拟表，当连接表时，可以在每个表中选择任何一列。

连接表的数量问题。连接表的数量取决于具体的数据库管理系统，有的规定最多连接25个，有的则没有限制。使用时，需要查看具体的数据库运行环境。连接表越多，响应的时间相对越长。

进行多表连接时，要注意查看查询条件。如果不指明连接条件，系统将对多表进行笛卡尔连接，产生巨大记录。

计算机 学院 专业 班 学号：

姓名： 教师评定：

实验题目

# 实验四 **创建和使用视图、索引**

实验项目名称：创建和使用视图

实验项目性质：基础型

所属课程名称：《数据库应用》实验计划学时：3

一、实验目的

1、理解视图、索引的定义、索引的优点与、索引的工作原理；

2、掌握在查询分析器和企业管理器中创建、修改及删除视图、索引；

3、掌握创建视图、索引的 SQL 语句的用法；并能够熟练利用视图向表中插入、删除和修改数据。

4、掌握使用视图来查询数据。

二、实验主要仪器设备和材料

1．计算机及操作系统：PC 机，Windows；

2．数据库管理系统：SQL Server 2019；

三、实验方法、步骤及结果测试

（一）、视图

1、建立“计算机学院”的学生基本情况视图 ies\_student\_view，该视图包括计算机学院

所有学生的学号、姓名、性别、出身年月、专业名称。

SQL语句及结果显示：

Create view ies\_student\_view

AS

SELECT distinct s.s\_no 学号,S.sname 姓名,S.sex 性别,S.sbirthday 出生年月,SP.spname 专业名称

FROM student as s,speciality as sp,department as d

where s.dno=

( select dno

from department

where dept\_name='Com.Sci')

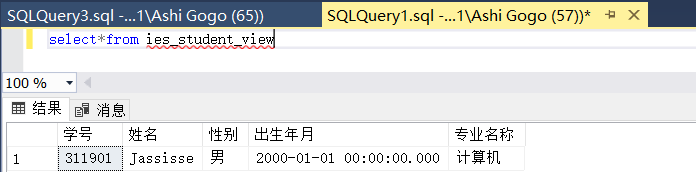
and s.spno=sp.spno

2、执行 ies\_student\_view 视图并观察结果。

SQL语句及结果显示：

select\*from ies\_student\_view

查询结果截图显示：



3、建立课程 1（上课编号）的学生名册的视图，该名册包括学生的学号、姓名、专业名

称和这门课的成绩；并查询结果。

SQL语句及结果显示：

Create view ies\_student\_course1\_view

as

select s.s\_no,s.sname,s.sex,p.spname,sc.score

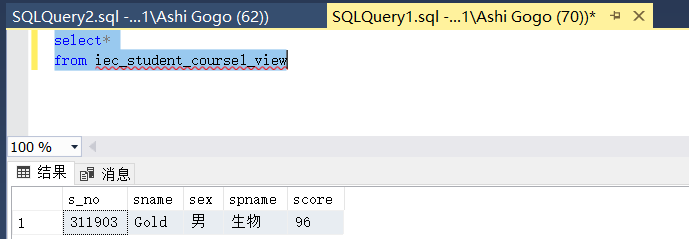
from student s,speciality p,department d,student\_course sc

where s.dno=d.dno and p.dno=d.dno and s.s\_no=sc.s\_no and sc.tcid=46 and s.spno=p.spno

查询结果截图显示：

select\*

from iec\_student\_course1\_view



4、建立统计不及格情况的视图，列出不及格学生的学号、姓名和不及格的课程代码；

SQL语句及结果显示：

create view iec\_student\_fail\_view

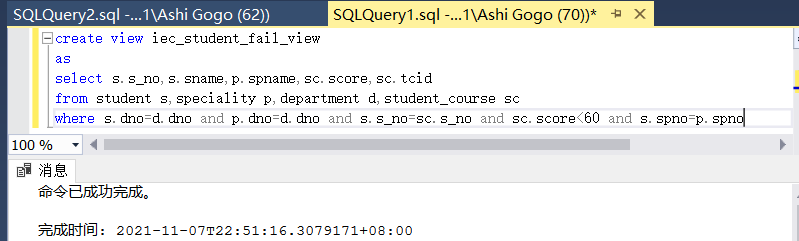
as

select s.s\_no,s.sname,p.spname,sc.score,sc.tcid

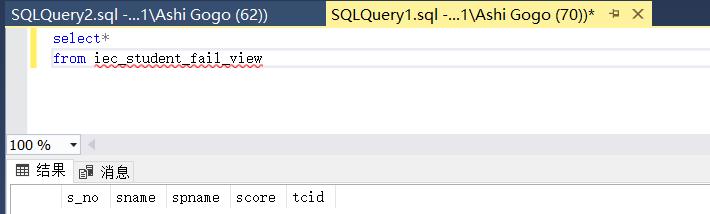
from student s,speciality p,department d,student\_course sc

where s.dno=d.dno and p.dno=d.dno and s.s\_no=sc.s\_no and sc.score<60 and s.spno=p.spno

查询结果截图显示：



5、执行视图并观察结果。



6、修改视图 ies\_student\_view，使该视图包括所有学生的学号、姓名、性别、出身年月、

学院名称、专业名称。

SQL语句及结果显示：

alter view ies\_student\_view

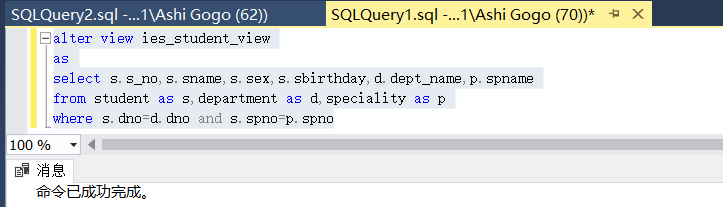
as

select s.s\_no,s.sname,s.sex,s.sbirthday,d.dept\_name,p.spname

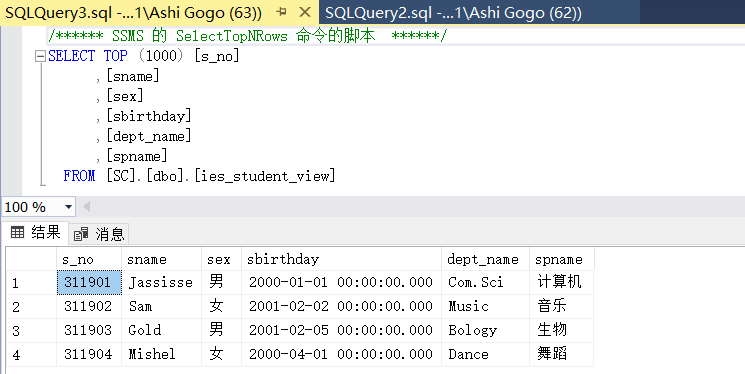
from student as s,department as d,speciality as p

where s.dno=d.dno and s.spno=p.spno

查询结果截图显示：



7、执行 ies\_student\_view 视图并观察结果。

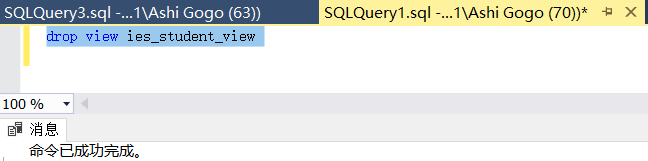


8、删除视图 ies\_student\_view。并查询结果。

SQL语句及结果显示：

drop view ies\_student\_view

查询结果截图显示：



（二）、索引

（1）、在 student\_course 表（学生选课表）的学生学号（sno）列上创建索引

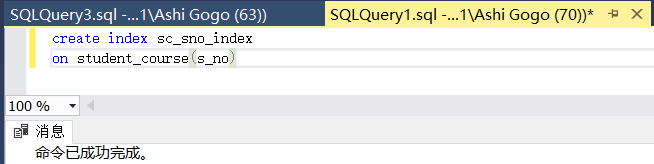
sc\_sno\_index

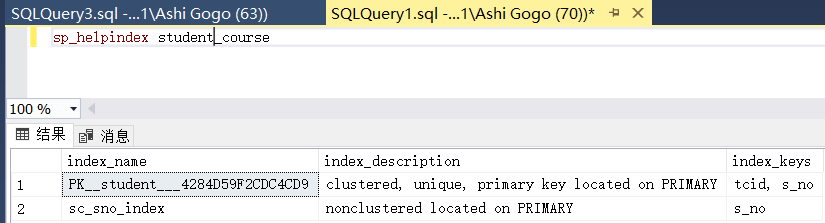
SQL语句及结果显示：

create index sc\_sno\_index

on student\_course(s\_no)

查询结果截图显示：





**四、思考题**

1、视图更新的优点

第一点：使用视图，可以定制用户数据，聚焦特定的数据。

第二点：使用视图，可以简化数据操作。

第三点：使用视图，基表中的数据就有了一定的安全性

第四点：可以合并分离的数据，创建分区视图

2、哪些视图是可以更新的？哪些视图是不可以更新的？针对实验举例说明。

答：一般来说，行列子集视图可以更新，视图的属性来自于聚集函数和表达式，则该视图是不可更新的。在本次实验中，ies\_student\_vie视图是可以更新的，其视图更新既是对student表的更新。视图tcid1\_student\_view和fail\_student\_view都是由两个基本表组成的，那么此视图不可以更新。