1.创建3张表

//学生表创建

CREATE table student(

Sno CHAR(9) PRIMARY KEY,

Sname CHAR(20) UNIQUE,

Ssex char(2),

Sage SMALLINT,

Sdept char(20)

);

//课程表创建

CREATE table course(

Cno char(4) PRIMARY KEY,

Cname char(40) not NULL,

Cpno char(4),

Ccredit SMALLINT

);

//学生选课表创建

CREATE table SC(

Sno char(9),

Cno char(4),

Grade SMALLINT

);

2.向表中添加数据

也可使用图形化工具Navicat或其他进行输入，

//向学生表中添加数据

INSERT into Student values(

201215121,'李勇','男',20,'CS'),

(201215122,'刘晨','女',19,'CS'),

(201215123,'王敏','女',18,'MA'),

(201215125,'张立','男',19,'IS'

);

//向课程表中添加数据

insert into course VALUES(

'1','数据库','5',4),

'2','数学','',2),

'3','信息系统','1',4),

('4','操作系统','6',3),

('5','数据结构','7',4),

('6','数据处理','',2),

('7','Java语言','6',4)

//向学生选课表中添加数据

insert into sc values

(201215121,1,92),

(201215121,2,85),

(201215121,3,88),

(201215122,2,58),

(201215122,3,80)

3.数据查询

3.1单表查询

3.1.1查询全体学生的学号与姓名

select Sno,Sname

from student

3.1.2查询全体学生的姓名，学号，所在系

select Sname,Sno,Sdept

from student

3.1.3查询全体学生的详细记录

select \* from student

3.1.4查询全体学生的姓名及其出生年份

select Sname,2020-Sage

from Student

3.1.5查询全体学生的姓名，出生年份和所在的院系，要求用小写字母表示系名

select Sname,2020-Sage,lower(Sdept)

from student

3.1.6查询选修了课程的学生学号

select Sno

from SC

3.1.7查询计算机科学系全体学生的名单

select Sname

from student

where Sdept='CS'

3.1.8查询所有年龄在20岁以下的学生姓名及其年龄

select Sname,Sage

from student

where Sage<20

3.1.9查询考试成绩不及格的学生的学号

select Sno

from sc

where Grade<60

3.1.10查询年龄在20-23岁（包括20和23）之间的学生的姓名，系别，年龄

select Sname,Sdept,Sage

from student

where Sage between 20 and 23

3.1.11查询年龄不中20-23之间的学生姓名，系别，年龄

select Sname,Sdept,Sage

from student

where Sage not between 20 and 23

3.1.12查询计算机科学系（CS），数学系（MA），信息系（IS）学生的姓名和性别

select Sname,Ssex

from student

where Sdept in('CS','MA','IS')

3.1.13查询既不是CS，MA，也不是IS的学生的姓名和性别

select Sname,Ssex

from student

where Sdept not in('CS','MA','IS')

3.1.14查询学号为201215121的学生的详细情况

select \*

from student

where Sno='201215121'

3.1.15查询所有姓刘的学生的姓名，学号，性别

select Sname,Sno,Ssex

from student

where Sname like '刘%'

3.1.16查询姓欧阳且全名为3个汉字的学生的姓名

select Sname

from student

where Sname like '欧阳\_'

3.1.17查询名字中第二个字为“阳”的学生的姓名和性别

select Sname,Ssex

from student

where Sname like '\_阳%'

3.1.18查询所有不姓刘的学生的姓名，学号和性别

select Sname,Sno,Ssex

from student

where Sname not like '刘%'

3.1.19查询缺少成绩的学生的学号和响应的课程号

select Sno,Cno

from sc

where grade is null

3.1.20查询所有有成绩的学生的学号和课程号

select Sno,Cno

from sc

where grade is not null

3.1.21查询计算机科学系且年龄在20岁以下的学生的姓名

select Sname

from student

where Sdept='CS' and Sage<=20

3.1.22查询选修了3号课程的学生的学号及其成绩，查询结果按分数的降序排列

//order by 默认升序，ASC是升序，DESC是降序

select Sno,Grade

from sc

where Cno='3'

order by Grade desc

3.1.23查询全体学生情况，查询结果按所在系的系号升序排列，同一系的学生按年龄降序排列

select \*

from student

order by Sdept,Sage DESC

3.1.24查询学生总人数

select count(\*)

from student

3.1.25查询选修了课程的学生人数

//学生可以选多门课程，避免重复需在count函数里加distinct短语

select count(distinct Sno)

from sc

3.1.26计算选修1号课程的学生平均成绩

select avg(Grade)

from sc

where Cno='1'

3.1.27查询选修1号课程的学生最高分数

select max(Grade)

from sc

where Cno='1'

3.1.28查询学生201215121选修课程的总学分数

select sum(Grade)

from sc

where Sno='201215121'

3.1.29求各个课程号及相应的选课人数

//group up 是将查询结果按某个属性进行分组

select Cno ,Count(Sno)

from sc

group by Cno

3.1.30查询选修了3门以上课程的学生学号

//having作用于组，这里先用group by按Sno进行分组，再用聚集函数count对每一组进行计数，用having提取出满足条件的组

select Sno

from sc

group by Sno

having count(\*)>3

3.1.31查询平均成绩大于等于90分的学生学号和平均成绩

//where句中不能用聚集函数作为条件表达式

select Sno,avg(Grade)

from sc

group by Sno

having avg(Grade)>=90

3.2连接查询

3.2.1查询每个学生及其选修课的情况

select student.\*,sc.\*

from Student,sc

where student.Sno=sc.Sno

3.2.2将上面 的例子用自然连接完成

select student.Sno,Sname,Ssex,Sage,Sdept,Cno,grade

from student,sc

where student.sno=sc.sno

3.2.3查询选修2号课程且成绩在90分以上的所有学生的学号和姓名

select student.Sno,Sname

from Student,sc

where student.Sno=sc.Sno and

sc.Cno='2' and

sc.Grade>=90

3.2.4查询每一门课的间接先修课（先修课的先修课）

//先对一门课找到其先修课，再按此先修课的课程号查找它的先修课，

//将表与自身连接，就要取别名

select FIRST.Cno,second.Cpno

from Course first,Course SECOND

where `first`.Cpno=`SECOND`.Cno

3.2.5查询每个学生的学号，姓名，选修的课程名及成绩

select student.Sno,Sname,Cname,Grade

from student,sc,course

where student.Sno=sc.Sno AND

sc.Cno=course.Cno

3.3嵌套查询

3.3.1查询与刘晨在同一个系学习的学生

select Sno,Sname,Sdept

from student

where Sdept in(

select Sdept

from student

where Sname='刘晨'

)

3.3.2查询选修了课程名为“信息系统”的学生学号和姓名

//嵌套查询太多了，用连接查询呈现出来

select student.Sno,Sname

from student,sc,course

where student.Sno=sc.Sno and

sc.Cno=course.Cno and

course.Cname='信息系统'

3.3.3找出每个学生超过他自己选修课程平均成绩的课程号

select Sno,Cno

from sc x

where Grade >=(

select avg(Grade)

from sc y

where y.Sno=x.Sno

)

3.3.4查询非计算机系中比计算机系任意学生年龄小的学生姓名和年龄

//任意：any 所有:all

select Sname,Sage

from student

where Sage<any(

select Sage

from student

where Sdept='CS'

)

3.3.5查询非计算机系中比计算机系所有学生年龄都小的学生姓名和年龄

select Sname,Sage

from student

where Sage<all(

select Sage

from student

where Sdept='CS'

)

3.3.6查询所有选修了1号课程的学生姓名

select Sname

from student,sc

where student.Sno=sc.Sno and

sc.Cno='1'

3.3.7查询选修了全部课程的学生姓名

select Sname

from student

where not exists(

select \*

from course

where not exists(

SELECT \*

from sc

where Sno=student.Sno AND

Cno=course.Cno

)

)

3.4集合查询

3.4.1查询选修了1号课程或则2号课程的学生

select Sno

from sc

where Cno='1'

UNION

select Sno

from sc

where Cno='2'

3.4.2查询既选修了课程1又选修了课程2的学生

select Sno

from sc

where Cno='1'

intersert

select Sno

from sc

where Cno='2'

4.数据更新

4.1插入数据

4.1.1将一个新学生元组（学号：201215128，姓名：陈东，性别：男，系别：IS，年龄：18岁）

insert into student

values ('201215128','陈东','男',18,'IS')

4.1.2插入一条选课记录

insert into sc(Sno,Cno) VALUES('201215128','1')

4.1.3对每一个系，求学生的平均年龄，并把结果存入数据库

//首先创建一张表来存放数据

create table Deptage(

Sdept char(15),

avg\_age smallint

)

//计算数据，存放到表中

insert into Deptage(Sdept,avg\_age)

select Sdept,avg(Sage)

from student

group by Sdept

4.2修改数据

4.2.1将学生201215121的年龄改为22岁

update student

set Sage=22

where Sno='201215121'

4.2.2将所有学生的年龄增加1岁

update student

set Sage=Sage+1

4.2.3将计算机系全体学生的成绩置0

update student

set Sage=0

where Sdept='CS'

4.3删除数据

4.3.1删除学号为201215128的学生记录

delete

from student

where Sno='201215128'

4.3.2删除计算机系所有学生的选课记录

delete

from sc

where Sno in(

select Sno

from student

where Sdept='CS'

)

4.3.3删除学生表所有记录

truncat student //该语句是删除该张表，重新创建表，不是一条一条删除表中数据；且truncat只能作用于表，delete，drop可作用于表，视图

5.常用关键字总结

5.1 distinct

5.1.1 作用于单列

select distinct 列名 //去重

5.1.2 作用多列

select distinct 列名1，列名2,列名3 //对3列都去重

5.1.3 与函数一起使用

select count(distinct 列名) //计算不重复的列名个数