create table 导师 (导师号 char(10),姓名char(20))

create table 专业 (专业号 char(10),专业名char(40))

create table 研究生 (研究生号 char(10),姓名 char(20))

INSERT INTO 导师 values('1111','张清逸')

INSERT INTO 导师 values('2222','刘逸')

INSERT INTO 专业 values('aaaa','计算机专业')

INSERT INTO 专业 values('bbbb','信息专业')

INSERT INTO 研究生 values('AAAA','李勇')

INSERT INTO 研究生 values('BBBB','刘晨')

INSERT INTO 研究生 values('CCCC','王敏')

SELECT 导师.姓名, 专业.专业名, 研究生.姓名FROM 导师, 专业, 研究生 ORDER BY 导师.姓名DESC , 专业.专业名;

create table 学生(学号 char(10) primary key ,姓名 char(20),性别 char(4),专业号 char(20), 年龄 number)

create table 选修(学号 char(10) ,课程号 char(10),成绩 number,primary key (学号,课程号))

alter table 学生 add constraint 专业号 foreign key(专业号) references 专业(专业号);

create table 课程 (课程号 char(10) primary key ,课程名char(40),学分 number)

insert into课程(课程号,课程名,学分) values('2', '数学', 4);

insert into课程 values('3', '信息系统', '1', 4);

alter table 选修 add constraint 选修学号 foreign key(学号) references 学生(学号);

alter table 选修 add constraint 选课程号 foreign key(课程号) references 课程(课程号);

create table course (课程号 char(10) primary key ,课程名 char(40),先行课 char(10),学分 number)

create table student(学号 char(10) primary key ,姓名 char(20),性别 char(4),年龄 number, 所在系 char(20))

create table sc(学号 char(10) ,课程号 char(10),成绩 number,primary key (学号,课程号))

alter table sc add constraint 选修学号 foreign key(学号) references student(学号);

alter table sc add constraint 选课程号 foreign key(课程号) references course (课程号);

INSERT INTO student VALUES ('201215121', '李勇', '男', 20, 'CS');

INSERT INTO student VALUES ('201215122', '刘晨', '女', 19, 'CS');

INSERT INTO student VALUES ('201215123', '王敏', '女', 18, 'MA');

INSERT INTO student VALUES ('201215125', '张立', '男', 19, 'IS');

insert into course values('1', '数据库', '5', 4);

insert into course(课程号,课程名,学分) values('2', '数学', 4);

insert into course values('3', '信息系统', '1', 4);

insert into course values('4', '操作系统', '6', 4);

insert into course values('5', '数据结构', '7', 4);

insert into course(课程号,课程名,学分) values('6', '数据处理', 4);

insert into course values('7', 'pascal 语言', '6', 4);

insert into sc values('201215121', '1',92);

insert into sc values('201215121', '2',85);

insert into sc values('201215121', '3',88);

insert into sc values('201215122', '2',90);

insert into sc values('201215122', '3',88);

所在系为IS的学生

SELECT student.\* FROM student where 所在系='IS'

查询年龄小于20岁的学生

SELECT student.\*, student.年龄 FROM student WHERE (((student.年龄)<20));

查询学生的姓名和所在系

SELECT student.姓名, student.所在系 FROM student;

交：

SELECT course.课程号, course.课程名, course.先行课, course.学分

FROM (course INNER JOIN 课程 ON course.课程号 = 课程.课程号)

select course.课程号, course.课程名 from course where 课程号 in (select 课程号 from 课程)

SELECT course.课程号, course.课程名, 课程.课程号 AS Expr1, 课程.课程名 AS Expr2

FROM (course LEFT OUTER JOIN

课程 ON course.课程号 = 课程.课程号)

C:\Users\Admin\AppData\Local\Temp\ksohtml16220\wps1.pngC:\Users\Admin\AppData\Local\Temp\ksohtml16220\wps2.png

减：

SELECT course.课程号, course.课程名, 课程.课程号 AS Expr1

FROM (course LEFT OUTER JOIN

课程 ON course.课程号 = 课程.课程号)

WHERE (课程.课程号 IS NULL)

SELECT 课程号, 课程名

FROM course

WHERE (课程号 NOT IN

(SELECT 课程号

FROM course course\_1

WHERE (课程号 = '2')))

SELECT 课程号, 课程名 FROM course UNION SELECT 课程号, 课程名 FROM 课程

SELECT \* FROM 学生 UNION ALL SELECT \* FROM student;

INTERSECT

SELECT \* FROM 学生 INTERSECT SELECT \* FROM student;

**第3章**

**创建学生表**

**CREATE TABLE Student**

**(Sno CHAR(9) PRIMARY KEY,**

**Sname CHAR(20) UNIQUE,**

**Ssex CHAR(2),**

**Sage SMALLINT,**

**Sdept CHAR(20)**

**)**

**创建课程表**

CREATE TABLE Course(

Cno CHAR(4) PRIMARY KEY,

Cname CHAR(40),

Cpno CHAR(4),

Ccredit SMALLINT,

FOREIGN KEY (Cpno) REFERENCES Course(Cno) )

**创建选课表**

**CREATE TABLE SC**

**(Sno CHAR(9),**

**Cno CHAR(4),**

**Grade SMALLINT,**

**PRIMARY KEY (Sno,Cno),**

**FOREIGN KEY (Sno) REFERENCES Student(Sno),**

**FOREIGN KEY (Cno)REFERENCES Course(Cno)**

**)**

**创建视图**

**CREATE VIEW IS\_Student AS SELECT Sno,Sname,Sage**

**FROM Student WHERE Sdept='IS';**

**向Student表增加“入学时间”列，其数据类型为日期型**

alter table Student add S\_entrance DATE

**将学生表中年龄的数据类型改为整数。**

ALTER TABLE Student ALTER COLUMN Sage INT

**增加课程名称必须取唯一值的约束条件**

ALTER TABLE Course ADD UNIQUE(Cname)

**删除主键**

alter table TABNAME drop primary key

**添加主键**

alter table sc add primary key(Sno,Cno)

**条件删除学生表**

DROP TABLE Student **RESTRICT**;

**级联删除学生表**

DROP TABLE Student **CASCADE**;

**建立索引**

CREATE UNIQUE INDEX Stusno ON Student(Sno)

CREATE UNIQUE INDEX Coucno ON Course(Cno)

CREATE UNIQUE INDEX SCno ON SC(Sno ASC,Cno DESC)

**删除索引**

**DROP INDEX** Stusno **on Student;**

**DROP INDEX Student.** Stusno**；**

**插入数据至学生表**

INSERT INTO student(**Sno , Sname, Ssex ,Sage ,Sdept**) VALUES ('201215121', '李勇', '男', 20, 'CS');

INSERT INTO student VALUES ('201215122', '刘晨', '女', 19, 'CS');

INSERT INTO student VALUES ('201215123', '王敏', '女', 18, 'MA');

INSERT INTO student(Sno,Sname,Ssex,Sage,Sdept) VALUES ('201215121', '李勇', '男', 20, 'CS');

INSERT INTO student(Sno,Sname,Ssex,Sage,Sdept) VALUES ('201215122', '刘晨', '女', 19, 'CS');

INSERT INTO student(Sno,Sname,Ssex,Sage,Sdept) VALUES ('201215123', '王敏', '女', 18, 'MA');

INSERT INTO student(Sno,Sname,Ssex,Sage,Sdept) VALUES ('201215125', '张立', '男', 19, 'IS');

**insert into course(Cno,Cname) values('1', '数据库');**

**insert into course(Cno,Cname) values(**'2', '数学'**);**

**insert into course(Cno,Cname) values(**'3', '信息系统'**);**

**insert into course(Cno,Cname) values(**'4', '操作系统'**);**

**insert into course(Cno,Cname) values(**'5', '数据结构'**);**

**insert into course(Cno,Cname) values(**'6', '数据处理'**);**

**insert into course(Cno,Cname) values(**'7', 'pascal 语言'**);**

**update course set Cpno='5',Ccredit=4 where Cno='1'**

**update course set Ccredit=2 where Cno='2'**

**update course set Cpno='1',Ccredit=4 where Cno='3'**

**update course set Cpno='6',Ccredit=3 where Cno='4'**

**update course set Cpno='7',Ccredit=4 where Cno='5'**

**update course set Ccredit=2 where Cno='6'**

**update course set Cpno='6',Ccredit=4 where Cno='7'**

insert into sc values('201215121', '1',92);

insert into sc values('201215121', '2',85);

insert into sc values('201215121', '3',88);

insert into sc values('201215122', '2',90);

insert into sc values('201215122', '3',88);

**单表查询**

**查询指定列 SELECT Sno,Sname FROM Student;**

**查询所有列 SELECT \* FROM Student;**

**查询经过计算的值 SELECT Sname, 2019 - Sage FROM Student**

**查询经过计算的值 SELECT Sname,'Year of Birth: ',2014-Sage FROM Student**

**查询结果使用别名 SELECT Sname, 2019 - Sage AS 出生年份 FROM Student**

**去掉表中重复的行 SELECT DISTINCT Sno FROM SC**

**查询满足条件的元组：**

**SELECT Sname,Sage FROM Student WHERE Sage < 20;**

**SELECT DISTINCT Sno FROM SC WHERE Grade<60;**

**SELECT Sname, Sdept, Sage FROM Student WHERE Sage (not)BETWEEN 20 AND 23**

**SELECT Sname, Ssex FROM Student WHERE Sdept (not)IN ('CS','MA','IS' );**

**SELECT Sname, Sno, Ssex FROM Student WHERE Sname LIKE '刘%';**

**SELECT Sname FROM Student WHERE Sname LIKE '欧阳\_\_';**

**SELECT Sno, Cno FROM SC WHERE (Grade IS NULL)**

**SELECT COUNT(\*) FROM Student**

**SELECT COUNT(DISTINCT Sno) FROM SC //Mysql上能执行**

**SELECT MAX(Grade) FROM SC WHERE (Cno = '1')**

**SELECT SUM(Ccredit) FROM SC,Course WHERE Sno='201215012' AND SC.Cno=Course.Cno**

**SELECT Cno, COUNT(Sno) FROM SC GROUP BY Cno**

**SELECT Cno as 课程号, COUNT(Sno) as 选课人数 FROM SC GROUP BY Cno**

**SELECT Sno FROM SC GROUP BY Sno //此时group by 作用类似于多个count**

**SELECT Sno FROM SC GROUP BY Sno HAVING COUNT(\*) >3**

**SELECT Sno, AVG(Grade) FROM SC GROUP BY Sno HAVING (AVG(Grade) >= 90)**

**连接查询**

**SELECT Student.\*, SC.\* FROM Student, SC WHERE Student.Sno = SC.Sno //等值连接**

SELECT Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade FROM Student,SC

WHERE Student.Sno = SC.Sno //自然连接，去除重复列

**SELECT FIRST.Cno, SECOND.Cpno FROM Course FIRST, Course SECOND WHERE FIRST.Cpno = SECOND.Cno //自身连接，取别名区别**

SELECT FIRST.Cno, SECOND.Cpno

FROM Course FIRST inner join Course SECOND

on FIRST.Cpno = SECOND.Cno //使用JOIN关键字

SELECT Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade FROM Student LEFT JOIN SC ON (Student.Sno=SC.Sno) //左外连接

**SELECT Student.Sno, Sname, Cname, Grade FROM Student, SC, Course**

**WHERE Student.Sno = SC.Sno AND SC.Cno = Course.Cno /\*多表连接\*/**

SELECT Sno, Sname, Sdept FROM Student WHERE Sdept IN (SELECT Sdept FROM Student WHERE Sname= ' 刘晨 ') //**查询与“刘晨”在同一个系学习的学生**

**SELECT Sno,Sname FROM Student WHERE Sno IN (SELECT Sno FROM SC WHERE Cno IN (SELECT Cno FROM Course WHERE Cname= '信息系统' )) //查询选修了课程名为“信息系统”的学生学号和姓名**

SELECT Sno, Cno FROM SC x WHERE Grade >=(SELECT AVG（Grade）FROM SC y WHERE y.Sno=x.Sno) //找出每个学生超过他选修课程平均成绩的课程号(相关查询)

**SELECT Sname, Sage FROM Student WHERE Sage < ANY (SELECT Sage　FROM Student WHERE Sdept= 'CS') AND Sdept != 'CS' //查询非计算机科学系中比计算机科学系任意一个学生年龄小的学生姓名和年龄 (也可用聚集函数形式)**