一、数据定义

1、创建数据库ST 【文档作业一、1.】

CREATE DATABASE IF NOT EXISTS ST;

2、使用数据库（指定数据库）【文档作业一、2.】

USE ST;

3、创建基本表-创建student，course和sc表，插入数据 【文档作业一、3.】

CREATE TABLE student

(Sno CHAR(9) PRIMARY KEY NOT NULL, /\*列级完整性约束 PPT3-1 32页\*/

Sname CHAR(20) UNIQUE NOT NULL,

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHAR(20)

);

CREATE TABLE course

(Cno CHAR(4) PRIMARY KEY,

Cname CHAR(40),

Cpno CHAR(4),

Ccredit SMALLINT,

FOREIGN KEY (Cpno) REFERENCES Course(Cno) /\*表级完整性约束 PPT3-1 32页\*/

);

CREATE TABLE sc

(Sno CHAR(9),

Cno CHAR(4),

Grade SMALLINT,

PRIMARY KEY (Sno,Cno),

/\* 主码由两个属性构成，必须作为表级完整性进行定义\*/

FOREIGN KEY (Sno) REFERENCES Student(Sno),

/\* 表级完整性约束条件，Sno是外码，被参照表是Student \*/

FOREIGN KEY (Cno) REFERENCES Course(Cno)

/\* 表级完整性约束条件， Cno是外码，被参照表是Course\*/

);

-- 实际项目中，并不严格遵从数据库设计规范，通常先不定义外键，留有灵活度。在应用程序中以下拉框的方式进行有限制的选择。

/\*补充：在已有的表中，⼀次性插⼊多⾏数据格式如下。

INSERT INTO table\_name VALUES (value1,value2,value3,...)，(value1,value2,value3,...);\*/

4、【3-1 PPT 33页-35页】

[例3.5] 建立“学生”表Student。学号是主码，姓名取值唯一。

CREATE TABLE Student

(Sno CHAR(9) PRIMARY KEY,   
 /\* 列级完整性约束条件,Sno是主码\*/

Sname CHAR(20) UNIQUE, /\* Sname取唯一值\*/

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHAR(20)

);

[例3.6 ] 建立一个“课程”表Course

CREATE TABLE Course

(Cno CHAR(4) PRIMARY KEY,

Cname CHAR(40),

Cpno CHAR(4),

Ccredit SMALLINT，

FOREIGN KEY (Cpno) REFERENCES Course(Cno)

);

[例3.7] 建立一个学生选课表SC

CREATE TABLE SC

(Sno CHAR(9),

Cno CHAR(4),

Grade SMALLINT，

PRIMARY KEY (Sno,Cno),

/\* 主码由两个属性构成，必须作为表级完整性进行定义\*/

FOREIGN KEY (Sno) REFERENCES Student(Sno),

/\* 表级完整性约束条件，Sno是外码，被参照表是Student \*/

FOREIGN KEY (Cno)REFERENCES Course(Cno)

/\* 表级完整性约束条件， Cno是外码，被参照表是Course\*/

);

5、数据更新 3-3PPT - 插入数据【作业文档】

在已有的表中，⼀次性插⼊多⾏数据格式如下：

INSERT INTO table\_name VALUES (value1,value2,value3,...)，(value1,value2,value3,...);

1) 在student表中插入数据

解1：

insert into STUDENT(Sno,Sname,Ssex,Sage,Sdept)

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');

解2：

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');

2) 在course表中插入数据

出现的问题：course表定义了依赖于表自身的外键约束，在插入数据时受到限制（系统自动检查）

-- 方案一

--（1）表先不插入Cpno列的数据，按如下顺序插入数据，后再更改列值：

insert into COURSE(Cno,Cname,Ccredit)values('1','数据库',4);

insert into COURSE(Cno,Cname,Ccredit)values('2','数学',2);

insert into COURSE(Cno,Cname,Ccredit)values('3','信息系统',4);

insert into COURSE(Cno,Cname,Ccredit)values('4','操作系统',3);

insert into COURSE(Cno,Cname,Ccredit)values('5','数据结构',4);

insert into COURSE(Cno,Cname,Ccredit)values('6','数据处理',2);

insert into COURSE(Cno,Cname,Ccredit)values('7','PASCAL语言',4);

-- （2)更改列值

UPDATE COURSE SET Cpno='5' WHERE Cno='1';

UPDATE COURSE SET Cpno=NULL WHERE Cno='2';

UPDATE COURSE SET Cpno='1' WHERE Cno='3';

UPDATE COURSE SET Cpno='6' WHERE Cno='4';

UPDATE COURSE SET Cpno='7' WHERE Cno='5';

UPDATE COURSE SET Cpno=NULL WHERE Cno='6';

UPDATE COURSE SET Cpno='6' WHERE Cno='7';

-- 方案二 如果已加外键参照约束，须按如下顺序插入数据：

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('2','数学',NULL,2);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('6','数据处理',NULL,2);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('4','操作系统','6',3);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('7','PASCAL语言','6',4);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('5','数据结构','7',4);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('1','数据库','5',4);

insert into COURSE(Cno,Cname,Cpno,Ccredit)values('3','信息系统','1',4);

3) 在sc表中插入数据，由于外键引用了course表的cno,必须在course表插入数据后进行

INSERT INTO SC(Sno,Cno,Grade) values('201215121','1',92);

INSERT INTO SC(Sno,Cno,Grade) values('201215121','2',85);

INSERT INTO SC(Sno,Cno,Grade) values('201215121','3',88);

INSERT INTO SC(Sno,Cno,Grade) values('201215122','2',90);

INSERT INTO SC(Sno,Cno,Grade) values('201215122','3',80);

6、【3-3PPT 8页-10页】

[例3.69]将一个新学生元组（学号：201215128;姓名：陈冬;性别：男;所在系：IS;年龄：18岁）插入到Student表中。

INSERT

INTO Student (Sno,Sname,Ssex,Sdept,Sage)

VALUES ('201215128','陈冬','男','IS',18);

[例3.71] 插入一条选课记录（ '200215128','1 '）。

INSERT

INTO SC(Sno,Cno)

VALUES ('201215128 ',' 1 ');

关系数据库管理系统DBMS将在新插入记录的Grade列上自动赋空值。

或者：

INSERT

INTO SC

VALUES (' 201215128 ',' 1 ',NULL);

[例3.70]将学生张成民的信息插入到Student表中。

INSERT

INTO Student

VALUES ('201215126','张成民','男’,18,'CS');

7、修改基本表【3-1PPT41页-语句格式】

向student表增加入学时间，数据类型为日期型类型（删除这个属性）【作业文档】

ALTER TABLE student ADD S\_entrance DATE;

ALTER TABLE student DROP COLUMN S\_entrance;

注意： 1) insert 日期型数据时，格式为 '2023-04-02'

2)在WorkBench的表中直接填数值时，不能加引号'',直接写2023-04-02

[例3.8] 向Student表增加“入学时间”列，其数据类型为日期型【3-1PPT 43页】

ALTER TABLE Student ADD S\_entrance DATE;

不管基本表中原来是否已有数据，新增加的列一律为空值

8、将年龄的数据类型改为int【作业文档】

ALTER TABLE student MODIFY Sage INT;

[例3.9] 将年龄的数据类型由字符型（假设原来的数据类型是字符型）改为整数。【3-1PPT 44页】

ALTER TABLE Student ALTER COLUMN Sage INT;

9、增加课程名的唯一约束条件【作业文档】

ALTER TABLE course ADD UNIQUE(Cname);

[例3.10] 增加课程名称必须取唯一值的约束条件。【3-1PPT 44页】

ALTER TABLE Course ADD UNIQUE(Cname);

二、单表简单查询

1、查询全体学生的姓名、学号和所在系【作业文档】

SELECT Sname,Sno,Sdept

FROM student;

2、查询所有学生的情况【作业文档】

SELECT \* FROM student;

[例3.18] 查询全体学生的详细记录【3-1PPT 65页】

SELECT Sno,Sname,Ssex,Sage,Sdept

FROM Student;

或

SELECT \*

FROM Student;

3、查询全体学生的姓名及其出生年份【作业文档】

SELECT Sname,'Year of Birth: ',2023-Sage

FROM student;

[例3.19] 查全体学生的姓名及其出生年份。【3-1PPT 66页】

SELECT Sname,2014-Sage /\*假设当时为2014年\*/

FROM Student;

[例3.20] 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。【3-1PPT 67页】

SELECT Sname,'Year of Birth: ',2014-Sage,LOWER(Sdept)

FROM Student;

使用列别名改变查询结果的列标题:【3-1PPT 68页】

SELECT Sname NAME,'Year of Birth:' BIRTH,

2014-Sage BIRTHDAY,LOWER(Sdept) DEPARTMENT

FROM Student;

4、 查询选修了课程的学生学号（去掉重复的值）【作业文档】

SELECT DISTINCT Sno FROM sc;

指定DISTINCT关键词，去掉表中重复的行

[例3.21] 查询选修了课程的学生学号。【3-1PPT 71页】

SELECT DISTINCT Sno

FROM SC;

5、查询计算机系的学生的名单【作业文档】

SELECT Sname FROM student

WHERE Sdept=’CS’

6、查询年龄在20以下的学生姓名及年龄【作业文档】

SELECT Sname,Sage

FORM student

WHERE Sage<20;

7、查询成绩不及格的学生的学号【作业文档】

SELECT DISTINCT Sno

FROM sc

WHERE Grade<60;

[例3.22] 查询计算机科学系全体学生的名单。【3-1PPT 73页】

SELECT Sname

FROM Student

WHERE Sdept=‘CS’;

[例3.23]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;

[例3.24]查询考试成绩有不及格的学生的学号。

SELECT DISTINCT Sno

FROM SC

WHERE Grade<60;

四、查询条件中含集合、字符串模糊匹配等

1、查询年龄在20~23之间的学生姓名、系和年龄【作业文档】

SELECT Sname,Sdept,Sage

FROM student

WHERE Sage BETWEEN 20 AND 23;

[例3.25] 查询年龄在20~23岁（包括20岁和23岁）之间的学生的姓名、系别和年龄【3-1PPT 74页】

SELECT Sname, Sdept, Sage

FROM Student

WHERE Sage BETWEEN 20 AND 23;

[例3.26] 查询年龄不在20~23岁之间的学生姓名、系别和年龄

SELECT Sname, Sdept, Sage

FROM Student

WHERE Sage NOT BETWEEN 20 AND 23;

2、查询计算机系、数学系和信息系的学生姓名和年龄【作业文档】

SELECT Sname,Sage FROM student

WHERE Sdept IN ('CS','MA’,'IS' );

3、查询不是计算机系、数学系和信息系的学生姓名和年龄

SELECT Sname,Sage

FROM student

WHERE Sdept NOT IN ('CS','MA’,'IS' );

[例3.27]查询计算机科学系（CS）、数学系（MA）和信息系（IS）学生的姓名和性别。【3-1PPT 75页】

SELECT Sname, Ssex

FROM Student

WHERE Sdept IN ('CS','MA’,'IS' );

[例3.28]查询既不是计算机科学系、数学系，也不是信息系的学生的姓名和性别。

SELECT Sname, Ssex

FROM Student

WHERE Sdept NOT IN ('IS','MA’,'CS' );

匹配串为含通配符的字符串

[例3.29] 查询学号为201215121的学生的详细情况。【3-1PPT 77页】

SELECT \*

FROM Student

WHERE Sno LIKE ‘201215121';

等价于：

SELECT \*

FROM Student

WHERE Sno = ' 201215121 ';

改写：

SELECT \*

FROM Student

WHERE Sno LIKE ‘2022%';

4、查询所有姓刘的学生的姓名、学号和性别【作业文档】

SELECT Sname,Sno,Sage

FROM student

WHERE Sname LIKE ‘刘%';

5、查询欧阳且全名为三个汉字的学生姓名

SELECT Sname

FROM student

WHERE Sname LIKE ‘欧阳\_';

[例3.30] 查询所有姓刘学生的姓名、学号和性别。【3-1PPT 78页】

SELECT Sname, Sno, Ssex

FROM Student

WHERE Sname LIKE '刘%';

[例3.31] 查询姓"欧阳"且全名为三个汉字的学生的姓名。

SELECT Sname

FROM Student

WHERE Sname LIKE '欧阳\_\_';

[例3.32] 查询名字中第2个字为"阳"字的学生的姓名和学号。【3-1PPT 79页】

SELECT Sname，Sno

FROM Student

WHERE Sname LIKE '\_\_阳%';

[例3.33] 查询所有不姓刘的学生姓名、学号和性别。

SELECT Sname, Sno, Ssex

FROM Student

WHERE Sname NOT LIKE '刘%';

[例3.34] 查询DB\_Design课程的课程号和学分。【3-1PPT 80页】

SELECT Cno，Ccredit

FROM Course

WHERE Cname LIKE 'DB\\_Design' ESCAPE '\ ' ;

[例3.35] 查询以"DB\_"开头，且倒数第3个字符为 i的课程的详细情况。

SELECT \*

FROM Course

WHERE Cname LIKE 'DB\\_%i\_ \_' ESCAPE '\ ' ;

6、谓词： IS NULL 或 IS NOT NULL【3-1PPT 81页】

注意：IS 不能用 = 代替

[例3.36] 某些学生选修课程后没有参加考试，所以有选课记录，但没有考试成绩。查询缺少成绩的学生的学号和相应的课程号。

SELECT Sno，Cno

FROM SC

WHERE Grade IS NULL

[例3.37] 查所有有成绩的学生学号和课程号。

SELECT Sno，Cno

FROM SC

WHERE Grade IS NOT NULL;

7、逻辑运算符：AND和 OR来连接多个查询条件【3-1PPT 82页】

[例3.38] 查询计算机系年龄在20岁以下的学生姓名。

SELECT Sname

FROM Student

WHERE Sdept= 'CS' AND Sage<20;

[例3.27]改写 查询计算机科学系（CS）、数学系（MA）和信息系（IS）学生的姓名和性别。【3-1PPT 83页】

SELECT Sname, Ssex

FROM Student

WHERE Sdept IN ('CS ','MA ','IS')

可改写为：

SELECT Sname, Ssex

FROM Student

WHERE Sdept= ' CS' OR Sdept= ' MA' OR Sdept= 'IS ';

8、练习order by子句ORDER BY子句

可以按一个或多个属性列排序

升序：ASC ;降序：DESC; 缺省值为升序

[例3.39]查询选修了3号课程的学生的学号及其成绩，查询结果按分数降序排列。【3-1PPT 86页】

SELECT Sno, Grade

FROM SC

WHERE Cno= ' 3 '

ORDER BY Grade DESC;

[例3.40]查询全体学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

SELECT \*

FROM Student

ORDER BY Sdept, Sage DESC; /\*可以按多个属性列排序\*/

五、分组查询group by+聚集函数+having

1、[例3.41] 查询学生总人数。【3-1PPT 89-90页】

SELECT COUNT(\*)

FROM Student;

[例3.42] 查询选修了课程的学生人数。

SELECT COUNT(DISTINCT Sno)/\*注意消除重复，这里必须用DISTINCT\*/

FROM SC;

[例3.43] 计算1号课程的学生平均成绩。

SELECT AVG(Grade)

FROM SC

WHERE Cno= ' 1 ';

[例3.44] 查询选修1号课程的学生最高分数。

SELECT MAX(Grade)

FROM SC

WHERE Cno='1';

[例3.45 ] 查询学生201215012选修课程的总学分数。

SELECT SUM(Ccredit)

FROM SC,Course

WHERE Sno='201215012' AND SC.Cno=Course.Cno;

2、[例3.46] 求各个课程号及相应的选课人数。【3-1PPT 93页】

SELECT Cno，COUNT(Sno) 注意：不能用DISTINCT消重，cno与sno是多对多关系

FROM SC

GROUP BY Cno;

3、按性别统计各系学生的人数【作业文档】

参考分析步骤：

1）select 统计学生人数 count(Sno) ，有时不能用count(\*) ,怕不能保证记录的唯一性

2）查询结果集应包含用来分组的列Sdept，Ssex，否则只显示学生人数，看不出意思

select sdept,ssex,count(Sno)

from student

group by sdept,ssex;

4、统计计算机系男生和女生的人数 【作业文档】

select ssex,count(Sno)

from student

where sdept='CS'

group by ssex;

select ssex,count(Sno)

from student

group by sdept,ssex

having sdept='CS';

以上两个答案都正确

注意：

1）并非凡是使用group by 就一定只用having，不用where，本题就是反例

2）二者区别在于：WHERE子句作用于基表（或视图）的所有元组，从中选择满足条件的元组，再分组

HAVING短语作用于每一组，从中选择出满足条件的组。即，筛选出的组都满足HAVING条件。

3）where子句与having可以同时使用

4）FROM - WHERE - GROUP BY - HAVING的顺序不能改变

5、统计各科的课号，以及最高分和最低分 【作业文档】

按课号分组，求最大，最小值

select Cno,MAX(Grade) ,MIN(Grade)

from SC

group by Cno;

-- 为MAX(Grade) 重命名为maxGrade ,MIN(Grade) 重命名为 minGrade

-- as可省略

select Cno,MAX(DISTINCT Grade) as maxGrade ,MIN(DISTINCT Grade) as minGrade

from SC

group by Cno;

6、[例3.47] 查询选修了3门以上课程的学生学号。【3-1PPT 94页】

SELECT Sno

FROM SC

GROUP BY Sno

HAVING COUNT(\*) >3;

7、[例3.48 ]查询平均成绩大于等于90分的学生学号和平均成绩【3-1PPT 94页】

SELECT Sno, AVG(Grade) 课程成绩按学号分组

FROM SC

GROUP BY Sno

HAVING AVG(Grade)>=90;

7、统计平均分超过90的学生的学号 【作业文档】

按学号分组，求平均分

select Sno

from SC

group by Sno

having AVG(Grade)>=90;

8、统计男生人数超过5的系 【作业文档】

select Sdept

from Student

group by Sdept,Ssex

having Ssex='男' and count(Sno)>=5;

六、连接查询

1、[例 3.49] 查询每个学生及其选修课程的情况 【3-2PPT 6页】

SELECT Student.\*, SC.\*

FROM Student, SC

WHERE Student.Sno = SC.Sno;

2、[例 3.51 ]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。【3-2PPT 13页】

SELECT Student.Sno, Sname

FROM Student, SC

WHERE Student.Sno=SC.Sno AND

SC.Cno=' 2 ' AND SC.Grade>90;

2、查询选修了2号课程且成绩在90分以上的所有学生的学号和姓名【作业文档】

select Student.Sno,Sname

from student,SC

where Cno='2' and Grade>=90 and Student.Sno=SC.Sno;

3、【自连接】

[例 3.52]查询每一门课的间接先修课（即先修课的先修课）【3-2PPT 15页】

SELECT FIRST.Cno, SECOND.Cpno

FROM Course FIRST, Course SECOND

WHERE FIRST.Cpno = SECOND.Cno;

select T1.Cno,T2.Cpno【作业文档】

from Course T1,Course T2

where T2.Cno=T1.Cpno;

4、【外连接】使用左外连接、右外连接和左右外连接（创建专业表major（mid，mname），

mid为专业编号，主码，mname为专业名。删除student中的属性，填加专业号mid，定义为外码，引用专业表中的专业号。？）【作业文档】

[例 3. 53] 改写[例 3.49] 【3-2PPT 20页】

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

FROM Student LEFT OUT JOIN SC ON

(Student.Sno=SC.Sno);

七、多表连接

1、[例3.54]查询每个学生的学号、姓名、选修的课程名及成绩【3-2PPT 23页】【作业文档】

SELECT Student.Sno, Sname, Cname, Grade

FROM Student, SC, Course /\*多表连接\*/

WHERE Student.Sno = SC.Sno

AND SC.Cno = Course.Cno;

八、连接和嵌套查询

1、SELECT Sname /\*外层查询/父查询\*/【3-2PPT 25页】

FROM Student

WHERE Sno IN

( SELECT Sno /\*内层查询/子查询\*/

FROM SC

WHERE Cno= ' 2 ');

2、[例 3.55] 查询与“刘晨”在同一个系学习的学生。【3-2PPT 32页】

SELECT Sno, Sname, Sdept

FROM Student

WHERE Sdept IN

(SELECT Sdept

FROM Student

WHERE Sname= ' 刘晨 ');

由于一个学生只可能在一个系学习，可以用 = 代替IN ：【3-2PPT 37页】

SELECT Sno,Sname,Sdept

FROM Student

WHERE Sdept =

(SELECT Sdept

FROM Student

WHERE Sname= '刘晨');

3、[例 3.56]查询选修了课程名为“信息系统”的学生学号和姓名【3-2PPT 34页】【作业文档】

1）嵌套查询

SELECT Sno,Sname

FROM Student

WHERE Sno IN

(SELECT Sno

FROM SC

WHERE Cno IN /\*此句也可写为 where cno = \*/

(SELECT Cno

FROM Course

WHERE Cname= '信息系统') );

2）连接查询

select Student.Sno,Sname

from Student,SC,Course

where Cname='信息系统' and SC.Cno=Course.Cno and Student.Sno=SC.Sno ;

4、【特殊情形：表自身嵌套】[例 3.57 ]找出每个学生 超过他选修课程平均成绩的 课程号。【3-2PPT 38页】

SELECT Sno, Cno

FROM SC x

WHERE Grade >=(SELECT AVG（Grade）

FROM SC y

WHERE y.Sno=x.Sno);

5、[例 3.58] 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【3-2PPT 45页】

SELECT Sname,Sage

FROM Student

WHERE Sage < ANY (SELECT Sage

FROM Student

WHERE Sdept= ' CS ')

AND Sdept != ‘CS ' ; /\*父查询块中的条件 \*/

用聚集函数实现[例 3.58] 【3-2PPT 47页】

SELECT Sname,Sage

FROM Student

WHERE Sage <(SELECT MAX（Sage）

FROM Student

WHERE Sdept= 'CS ')

AND Sdept <> ' CS ';

6、[例 3.59] 查询非计算机科学系中 比计算机科学系所有学生年龄都小的 学生姓名及年龄。【3-2PPT 48-49页】

方法一：用ALL谓词

SELECT Sname,Sage

FROM Student

WHERE Sage < ALL

(SELECT Sage

FROM Student

WHERE Sdept= ' CS ')

AND Sdept <> ' CS ’;

方法二：用聚集函数

SELECT Sname,Sage

FROM Student

WHERE Sage <

(SELECT MIN(Sage)

FROM Student

WHERE Sdept= ' CS ')

AND Sdept <>' CS ';

7、查找计算机系年龄最大的学生的学号和姓名(ALL)【作业文档】

思路：年龄=查询出来的最大年龄 嵌套查询

select Sno,Sname

from Student

where sdept='CS' and Sage=(select Max(Sage)

from Student

where sdept='CS');

select Sno,Sname

from Student

where Sage>=ALL(select Sage

from Student

where sdept='CS') and sdept='CS';

8、查找 选修了最多课程的 学生的姓名【作业文档】

1）

select sname

from student

where (select count(cno)

from sc

where sno=student.sno)>=ALL(select count(cno)

from sc

group by sno);

注意：聚集函数不能嵌套，MAX(count(Cno))执行报错，只能用count()>=ANY(select count()...)实现

2）查找选修了最多课程的学生的学号和课程数目（变化题目）

select sno，count(cno)

from sc

group by sno

having count(cno)>=ALL(select count(Cno)

from sc

group by Sno);

九、带有EXISTS/NOT EXISTS谓词的子查询

（一）EXISTS

1、[例 3.60]查询所有选修了1号课程的学生姓名。【PPT3-2 56页】

SELECT Sname

FROM Student

WHERE EXISTS

(SELECT \*

FROM SC

WHERE Sno=Student.Sno AND Cno= ' 1 ');

2、 选修了“数据库“的学生的学号（使用相关子查询，使用EXISTS以及不使用EXISTS）【作业文档】

1）

select sno

from sc

where exists(select cno

from course

where cname='数据库' and cno=sc.cno);

2）

select sc.Sno

from sc,course

where Cname='数据库' and sc.Cno=course.Cno;

3、 [例 3.55]查询与“刘晨”在同一个系学习的学生。【PPT3-2 58页】

可以用带EXISTS谓词的子查询替换：

SELECT Sno,Sname,Sdept

FROM Student S1

WHERE EXISTS

　 (SELECT \*

FROM Student S2

WHERE S2.Sdept = S1.Sdept AND

S2.Sname = '刘晨');

（二）NOT EXISTS

4、[例 3.61] 查询没有选修1号课程的学生姓名。【PPT3-2 56页】

SELECT Sname

FROM Student

WHERE NOT EXISTS

(SELECT \*

FROM SC

WHERE Sno = Student.Sno AND Cno='1');

5、没有选修“操作系统”的学生的学号和姓名。【作业文档】

select sno,sname

from student

where not exists(select sno

from course,sc

where cname='操作系统' and sno=student.sno and sc.cno=course.cno);

（三）全称量词（全部）的实现-双重NOT EXISTS【难点】

6、[例 3.62] 查询选修了全部课程的学生姓名。【PPT3-2 59页】【作业文档】

select sname

from student

where NOT EXISTS (select \*

from course

where NOT EXISTS(select \*

from sc

where cno=sc.cno and sno=student.sno) );

解2：

select sname

from student

where NOT EXISTS (select \*

from course

where cno NOT IN (select cno

from sc

where sno=student.sno) );

[例 3.63]查询至少选修了学生201215122选修的全部课程的学生号码。【PPT3-2 62-64页】

用NOT EXISTS谓词+表自身嵌套表示：

SELECT DISTINCT Sno

FROM SC SCX

WHERE NOT EXISTS /\*不存在某个cno,201215122选修了，却没有当前sno选修该cno的记录\*/

(SELECT \*

FROM SC SCY

WHERE SCY.Sno = ' 201215122 ' AND

NOT EXISTS

(SELECT \*

FROM SC SCZ

WHERE SCZ.Sno=SCX.Sno AND

SCZ.Cno=SCY.Cno));

4、查询选了全部学分大于4的课程的学生的学号【作业文档】

1)

select DISTINCT sno

from sc sc1

where NOT EXISTS( select \*

from course

where Ccredit>=4 and

NOT EXISTS(select \*

from sc

where sno =sc1.sno and cno=course.cno));

2)

select DISTINCT sno

from sc sc1

where NOT EXISTS( select \*

from course

where Ccredit>=4 and cno NOT IN(select cno

from sc

where sno=sc1.sno));

5.查询没有参加数据库这门课程考试学生的学号，姓名和专业【实验报告】

1)选修了数据库课程，有数据库选课记录，但grade is NULL

【注意】必须写 is NULL,不能写 = NULL !

解1（连接）：

select student.sno,sname,sdept

from student,sc,course

where cname='数据库' and grade is NULL

and student.sno=sc.sno and sc.cno=course.cno;

解2（嵌套）：

select sno,sname,sdept

from student

where sno IN(select sno

from sc

where cno=(select cno

from course

where cname='数据库') and grade is NULL);

解3：

select sno,sname,sdept

from student

where EXISTS (select \*

from sc

where cno=(select cno

from course

where cname='数据库')

and sno=student.sno and grade is NULL);

2)只要没参加数据库考试的学生，无论是否选修了数据库都统计

select sno,sname,sdept

from student

where NOT EXISTS(select \*

from sc

where sno=student.sno and grade is not NULL

and cno=(select cno

from course

where cname='数据库'));

6、1）查询没有选修数据库这门课程考试学生的学号，姓名和专业【题目改写】

解1-多重嵌套：

嵌套查询:选修数据库这门课程=存在学生的数据库选修记录

没选修=不存在该学生的数据库记录 。NOT EXISTS 。

select sno,sname,sdept

from student

where NOT EXISTS (select \*

from sc

where cno=(select cno

from course

where cname='数据库')

and sno=student.sno );

解2-嵌套+连接：

select sno,sname,sdept

from student

where NOT EXISTS (select \*

from sc,course

where cname='数据库' and sno=student.sno and sc.cno=course.cno);

7、查询所选全部学分总和大于100的学生的学号(改写的题目，注意区别)【作业文档】

select sno

from sc,course

where sc.cno=course.cno

group by sno

having sum(Ccredit)>100;

八、综合

1、查询李勇所选课程的课程号，课程名（分别使用连接和子查询）【实验报告】【作业文档】

1）连接

select course.cno,cname

from student,sc,course

where sname='李勇' and student.sno=sc.sno and sc.cno=course.cno;

2)子查询

select cno,cname

from course

where cno IN(select cno

from sc

where sno IN(select sno

from student

where sname='李勇'));

3)子查询+连接【PPT中无混合使用的例子】

select cno,cname

from course

where cno IN(select cno

from sc,student

where sname='李勇' and student.sno=sc.sno);

2.查询和李勇年龄相同的学生的学号，姓名和专业【实验报告】【作业文档】

select sno,sname,sdept

from student

where sage=(select sage

from student

where sname='李勇') and sname!='李勇';

3、查询李勇选课的总学分【实验报告】【作业文档】

解1（连接）：

select SUM(ccredit)

from student,sc,course

where sname='李勇' and student.sno=sc.sno and sc.cno=course.cno;

解2（嵌套）：

select SUM(ccredit)

from course

where cno IN(select cno

from sc

where sno=(select sno

from student

where sname='李勇'));

解3（嵌套+连接）：

select sum(Ccredit)

from course

where cno IN(select cno

from sc,student

where sname='李勇' and student.sno=sc.sno);

5、统计每门课程的选课人数，输出课号、课程名称和人数【实验报告】【作业文档】

select course.cno,cname,count(sno) /\*注意：select course.cno,cname,count(\*) 也正确\*/

from sc,course

where sc.cno=course.cno

group by course.cno;

6、统计平均成绩超过90的学生的姓名，学号和专业【实验报告】【作业文档】

解1:

select sname,sno,sdept

from student,sc

where student.sno=sc.sno

group by student.sno

having AVG(grade)>=90;

解2：

select sname,student.sno,sdept

from student

where (select AVG(grade)

from sc

where sno=student.sno) >=90;

7、数据库这门课取得最高成绩的学生学号、姓名和专业【实验报告】【作业文档】

解1：

注意：在查询某个学生的数据库成绩和查询数据库最高成绩时，条件都要包含cname='数据库'

select student.sno,sname,sdept

from student,sc,course

where cname='数据库' and grade=(select MAX(grade)

from sc

where cno=(select cno

from course

where cname='数据库'))

and student.sno=sc.sno and sc.cno=course.cno;

解2：

注意：最高分可能有多个学生并列，一定要用IN

select sno,sname

from student

where sno IN(select sno

from sc,course

where cname='数据库' and grade=(select MAX(grade)

from sc,course

where sc.cno=course.cno and cname='数据库')

and sc.cno=course.cno);

九、集合查询

1、[例 3.64] 查询计算机科学系的学生及年龄不大于19岁的学生。【PPT3-2 67页】

SELECT \*

FROM Student

WHERE Sdept= 'CS'

UNION /\*相当于 or \*/

SELECT \*

FROM Student

WHERE Sage<=19;

2、[例 3.65] 查询选修了课程1或者选修了课程2的学生。【PPT3-2 68页】

SELECT Sno

FROM SC

WHERE Cno=' 1 '

UNION

SELECT Sno

FROM SC

WHERE Cno= ' 2 ';

3、[例3.66] 查询计算机科学系的学生与年龄不大于19岁的学生的交集。【PPT3-2 69页】

SELECT \*

FROM Student

WHERE Sdept='CS'

INTERSECT

SELECT \*

FROM Student

WHERE Sage<=19

4、[例 3.67]查询既选修了课程1又选修了课程2的学生。【PPT3-2 71页】

SELECT Sno

FROM SC

WHERE Cno=' 1 '

INTERSECT

SELECT Sno

FROM SC

WHERE Cno='2 ';

注意：不能简单用AND代替

解2：SELECT Sno 【PPT3-2 72页】

FROM SC

WHERE Cno='1' AND Sno IN

(SELECT Sno

FROM SC

WHERE Cno='2');

5、[例 3.68] 查询计算机科学系的学生与年龄不大于19岁的学生的差集。【PPT3-2 73页】

SELECT \*

FROM Student

WHERE Sdept='CS'

EXCEPT

SELECT \*

FROM Student

WHERE Sage <=19;

十、基于派生表的查询

1、[例3.57]找出每个学生超过他自己选修课程平均成绩的课程号【PPT3-2 76页】

SELECT Sno, Cno

FROM SC, (SELECT Sno, Avg(Grade)

FROM SC

GROUP BY Sno)

AS Avg\_sc(avg\_sno,avg\_grade)

WHERE SC.Sno = Avg\_sc.avg\_sno

and SC.Grade >=Avg\_sc.avg\_grade

2、[例3.60]查询所有选修了1号课程的学生姓名，可以用如下查询完成：【PPT3-2 77页】

SELECT Sname

FROM Student,

(SELECT Sno FROM SC WHERE Cno=' 1 ') AS SC1

WHERE Student.Sno=SC1.Sno;

十一、数据更新

（一）插入数据

1、[例3.69]将一个新学生元组（学号：201215128;姓名：陈冬;性别：男;所在系：IS;年龄：18岁）插入到Student表中。【PPT3-3 8页】

INSERT

INTO Student (Sno,Sname,Ssex,Sdept,Sage)

VALUES ('201215128','陈冬','男','IS',18);

2、[例3.71] 插入一条选课记录（ '200215128','1 '）。【PPT3-3 8页】

INSERT

INTO SC(Sno,Cno)

VALUES ('201215128 ',' 1 ');

关系数据库管理系统将在新插入记录的Grade列上自动地赋空值NULL。

或者：

INSERT

INTO SC

VALUES (' 201215128 ',' 1 ',NULL);

3、[例3.70]将学生张成民的信息插入到Student表中。【PPT3-3 10页】

INSERT

INTO Student

VALUES ('201215126','张成民','男’,18,'CS');

4、[例3.72] 对每一个系，求学生的平均年龄，并把结果存入数据库【PPT3-3 12页】

第一步：建表

CREATE TABLE Dept\_age

( Sdept CHAR(15) , /\*系名\*/

Avg\_age SMALLINT); /\*学生平均年龄\*/

第二步：插入数据

INSERT

INTO Dept\_age(Sdept,Avg\_age)

SELECT Sdept，AVG(Sage)

FROM Student

GROUP BY Sdept;

5、创建表t1(sdept,total)，使用子查询将插入系的名称以及人数【实验报告】【作业文档】

create table t1

(sdept CHAR(15),

total SMALLINT);

insert into t1

select sdept,count(sno)

from student

group by sdept;

（二）修改数据 update

6、 [例3.73] 将学生201215121的年龄改为22岁【PPT3-3 17页】

UPDATE Student

SET Sage=22

WHERE Sno=' 201215121 ';

7、 [例3.74] 将所有学生的年龄增加1岁。【PPT3-3 22页】

UPDATE Student

SET Sage= Sage+1;

8、[例3.75] 将计算机科学系全体学生的成绩置零。

UPDATE SC

SET Grade=0

WHERE Sno IN

(SELETE Sno

FROM Student

WHERE Sdept= 'CS' );

9、将计算机的学生的成绩提高5%【实验报告】【作业文档】

update sc

set grade=grade\*(1+0.05)

where sno IN(select sno

from student

where sdept='CS');

（二）删除数据 delete

1、 [例3.76] 删除学号为201215128的学生记录。【PPT3-3 28页】

DELETE

FROM Student

WHERE Sno= 201215128 ';

2、[例3.77] 删除所有的学生选课记录。【PPT3-3 29页】

DELETE

FROM SC;

3、[例3.78] 删除计算机科学系所有学生的选课记录。【PPT3-3 30页】

DELETE

FROM SC

WHERE Sno IN

(SELETE Sno

FROM Student

WHERE Sdept= 'CS') ;

3、从sc表删除没参加考试的学生【作业文档】

1）运行错误：

delete from sc

where grade is NULL;

报错原因：现在是在mysql的safe-updates模式中，如果where后跟的条件不是主键，就会出现这种错误。

where后面加上主键的条件，仍然报错:

delete from sc

where sno is not NULL and grade is NULL;

2）改模式：

SET SQL\_SAFE\_UPDATES = 0;

delete from sc

where grade is NULL;

SET SQL\_SAFE\_UPDATES = 1;

/\*注意：sno!=NULL或sno=NULL的用法不对，有关null的判断是 is null 和 is not null两种\*/

十二、视图

1、[例3.84] 建立信息系学生的视图。【PPT3-3 49页】

CREATE VIEW IS\_Student

AS

SELECT Sno,Sname,Sage

FROM Student

WHERE Sdept= 'IS';

2、[例3.85]建立信息系学生的视图，并要求进行修改和插入操作时仍需保证该视图只有信息系的学生 。【PPT3-3 50页】

CREATE VIEW IS\_Student

AS

SELECT Sno,Sname,Sage

FROM Student

WHERE Sdept= 'IS'

WITH CHECK OPTION;

3、[例3.86] 建立信息系选修了1号课程的学生的视图（包括学号、姓名、成绩）。【PPT3-3 52页】

CREATE VIEW IS\_S1(Sno,Sname,Grade)

AS

SELECT Student.Sno,Sname,Grade

FROM Student,SC

WHERE Sdept= 'IS' AND

Student.Sno=SC.Sno AND

SC.Cno= '1';

4、基于视图的视图

[例3.87] 建立信息系选修了1号课程且成绩在90分以上的学生的视图。【PPT3-3 53页】

CREATE VIEW IS\_S2

AS

SELECT Sno,Sname,Grade

FROM IS\_S1

WHERE Grade>=90;

5、带表达式的视图

[例3.88] 定义一个反映学生出生年份的视图。【PPT3-3 54页】

CREATE VIEW BT\_S(Sno,Sname,Sbirth)

AS

SELECT Sno,Sname,2014-Sage

FROM Student;

6、创建计算机的学生视图【作业文档】

create view CSstu

as

select \*

from student

where sdept='CS';

select \* from CSstu;

5、创建选修了数据库的学生视图，属性包括学号，姓名和成绩【实验报告】【作业文档】

create view stuDB(sno,name,grade)

as

select student.sno,sname,grade

from student,sc,course

where cname='数据库' and sc.cno=course.cno and student.sno=sc.sno;

select \* from stuDB;

6、查询计算机系选修了数据库这门课程的学生学号、姓名和成绩【实验报告】【作业文档】

从5、创建的视图stuDB查询

select \*

from stuDB

where sno IN(select sno

from student

where sdept='CS');

7、查询数据库这门课成绩超过平均分的同学的学号、姓名和成绩【实验报告】【作业文档】

select \*

from stuDB

where grade>=(select AVG(grade)

from stuDB);

十三、空值处理

1、[例 3.79]向SC表中插入一个元组，学生号是”201215126”，课程号是”1”，成绩为空。【PPT3-3 33页】

INSERT INTO SC(Sno,Cno,Grade)

VALUES('201215126 ','1',NULL); /\*该学生还没有考试成绩，取空值\*/

或

INSERT INTO SC(Sno,Cno)

VALUES(' 201215126 ','1'); /\*没有赋值的属性，其值为空值\*/

2、[例3.80] 将Student表中学生号为”201215200”的学生所属的系改为空值。【PPT3-3 34页】

UPDATE Student

SET Sdept = NULL

WHERE Sno='201215200';

3、判断一个属性的值是否为空值，用IS NULL或IS NOT NULL来表示。

[例 3.81] 从Student表中找出漏填了数据的学生信息 【PPT3-3 35页】

SELECT \*

FROM Student

WHERE Sname IS NULL OR Ssex IS NULL OR Sage IS NULL OR Sdept IS NULL;

4、[例3.82] 找出选修1号课程的不及格的学生。【PPT3-3 39页】

SELECT Sno

FROM SC

WHERE Grade < 60 AND Cno='1';

注意： 查询结果不包括缺考的学生，因为他们的Grade值为 null。

5、[例 3.83] 选出选修1号课程的不及格的学生以及缺考的学生。【PPT3-3 39页】

SELECT Sno

FROM SC

WHERE Grade < 60 AND Cno='1'

UNION

SELECT Sno

FROM SC

WHERE Grade IS NULL AND Cno='1'

或者

SELECT Sno

FROM SC

WHERE Cno='1' AND (Grade<60 OR Grade IS NULL);