**环境：win7，IBM DB2**

先连接数据库

CONNECT TO DBSTUDY;

**（一）数据定义**

**1. 建立基本表**

[1] 创建学生表Student，由以下属性组成：学号SNO（INT型，主键），姓名SNAME（CHAR型，长度为8，非空），性别SEX（CHAR型，长度为2），所在系DEPTNO（INT型）。

CREATE TABLE STUDENT ( SNO INTEGER NOT NULL,

SNAME CHARACTER (8) NOT NULL ,

SEX CHARACTER (2) ,

DEPTNO INTEGER NOT NULL , CONSTRAINT SNO PRIMARY KEY ( SNO) ) ;

[2] 创建课程表Course，由以下属性组成：课程号CNO（INT型），课程名CNAME（CHAR型，长度为20，非空），授课教师编号TNO（INT型），学分CREDIT（INT型）。其中（CNO，TNO）为主键。

CREATE TABLE COURSE ( CNO INTEGER NOT NULL,

CNAME CHARACTER(20) NOT NULL,

TNO INTEGER,

CREDIT INTEGER,

CONSTRAINT CNO PRIMARY KEY(CNO));

[3] 创建学生选课表SC，由以下属性组成：学号SNO，课程号CNO，成绩GRADE。所有属性均为INT型，其中（SNO，CNO）为主键。

CREATE TABLE SC(SNO INTEGER NOT NULL,

CNO INTEGER NOT NULL,

GRADE INTEGER,

CONSTRAINT SNO\_CNO PRIMARY KEY(SNO,CNO));

[4] 创建教师表Teacher，由以下属性组成：教师编号TNO（INT型，主键），教师姓名TNAME（CHAR型，长度为8，非空），所在系DEPTNO（INT型）。

CREATE TABLE TEACHER(TNO INTEGER NOT NULL,

TNAME CHARACTER(8) NOT NULL,

DEPTNO INTEGER,

CONSTRAINT TNO PRIMARY KEY(TNO));

[5] 创建系表Dept，由以下属性组成：系号DEPTNO（INT型，主键），系名DNAME（CHAR型，长度为20，非空）。

CREATE TABLE DEPT(DEPTNO INTEGER NOT NULL,

DNAME CHARACTER(20) NOT NULL,

CONSTRAINT DEPTNO PRIMARY KEY(DEPTNO));

**2. 修改基本表**

[1] 在Student表中加入属性AGE（INT型）。

ALTER TABLE STUDENT ADD COLUMN AGE INT;

//不可为NOT NULL，否则出错

[2] 将Student表中的属性AGE类型改为SMALLINT型。

ALTER TABLE STUDENT DROP AGE;

ALTER TABLE STUDENT ADD COLUMN AGE SMALLINT;

//db2下没有找到好的办法，如果是mysql,网上说可以

ALTERＴTABLE STUDENT MODIFY AGE SMALLINT;

**3. 删除基本表**

[1] 在所有操作结束后删除Student表。

DROP TABLE STUDENT;

[2] 在所有操作结束后删除Course表。

DROP TABLE COURSE;

[3] 在所有操作结束后删除SC表。

DROP TABLE SC;

[4] 在所有操作结束后删除Teacher表。

DROP TABLE TEACHER;

[5] 在所有操作结束后删除Dept表。

DROP TABLE DEPT;

**（二）数据操作**

**1. 插入数据**

[1] 向Student表插入下列数据：

（1001，张天，m，10，20）、（1002，李兰，f，10，21）、

（1003，陈铭，m，10，21）、（1004，刘茜，f，20，21）、

（1005，马阳，m，20，22）。

INSERT

INTO STUDENT

VALUES(1001, '张天', 'm ',10,20),( 1002 '李兰', 'f ',10,21),

//word里面不好输入'，下面的’全部为'。

(1003,’陈铭’,’m’,10,21),(1004,’刘茜’,’f’,20,21),(1005,’马阳’,’m’,20,22);

[2] 向Course表插入下列数据：

（1，数据结构，101，4）、（2，数据库，102，4）、

（3，离散数学，103，4）、（4，C语言程序设计，101，2）。

INSERT

INTO COURSE

VALUES(1,’数据结构’,101,4), (2,’数据库’,102,4),(3,’离散数学’,103,4),(4,‘C语言程序设计’,101,2);

[3] 向SC表插入下列数据：

（1001，1，80）、（1001，2，85）、（1001，3，78）、

（1002，1，78）、（1002，2，82）、（1002，3，86）、

（1003，1，92）、（1003，3，90）、

（1004，1，87）、（1004，4，90）、

（1005，1，85）、（1005，4，92）。

INSERT

INTO SC

VALUES(1001,1,80),(1001,2,85),(1001,3,78),

(1002,1,78),(1002,2,82),(1002,3,86),

(1003,1,92),(1003,3,90),(1004,1,87),

(1004,4,90),(1005,1,85),(1005,4,92);

[4] 向Teacher表插入下列数据：

（101，张星，10）、（102，李珊，10）、

（103，赵应，10）、（104，刘田，20）。

INSERT

INTO TEACHER

VALUES(101,’张星’,10),(102,’李珊’,10),(103,’赵应’,10),(104,’刘田’,20);

[5] 向Dept表插入下列数据：

（10，计算机）、（20，信息）。

INSERT

INTO DEPT

VALUES(10,’计算机’),(20,’信息’);

**2. 单表查询**

[1] 查询所有学生的信息。

SELECT \* FRＯＭ STUDENT;

[2] 查询所有女生的姓名。

SELECT SNAME

FROM STUDENT

WHERE SEX=‘f’;

[3] 查询成绩在80到89之间的所有学生的选课记录，查询结果按成绩的降序排列。

SELECT \*

FROM SC

WHERE GRADE >= 80 AND GRADE <=89

ORDER BY GRADE DESC;

[4] 查询各个系的学生人数。

SELECT DEPTNO,COUNT(SNO)

FROM STUDENT

GROUP BY DEPTNO;

**3. 连接查询**

查询信息系年龄在21岁以下（含21岁）的女生姓名及其年龄。

SELECT SNAME,AGE

FROM STUDENT,DEPT

WHERE STUDENT.AGE <= 21 AND STUDENT.SEX = ‘f’AND STUDENT.DEPTNO = DEPT.DEPTNO AND DEPT.DNAME = ‘信息’;

**4. 嵌套查询**

[1] 查询修课总学分在10分以下的学生姓名。

SELECT SNAME

FROM STUDENT

WHERE SNO IN

(SELECT SNO

FROM SC,COURSE

WHERE SC.CNO = COURSE.CNO

GROUP BY SC.SNO

HAVING SUM(COURSE.CREDIT)<10);

[2] 查询各门课程取得最高成绩的学生姓名及其成绩。

SELECT SNAME,GRADE

FROM STUDENT,SC SC1

WHERE STUDENT.SNO = SC1.SNO AND GRADE >= ALL

(SELECT GRADE

FROM SC SC2

WHERE SC1.CNO = SC2.CNO)

[3] 查询选修了1001学生选课的全部课程的学生学号。?

//实际上是除法的实现

SELECT SNO

FROM SC

WHERE (SELECT SC1.CNO

FROM SC SC1

WHERE SC.SNO = SC1.SNO)

CONTAINS (SELECT SC2.CNO

FROM SC SC2

WHERE SC2.SNO = ‘1001’);

CONTAINS是不能执行滴。

考虑到CONTAINS的语义，通过两个NOT EXISTS来实现。

对于符合条件的学生，因为其课程contain1001号学生的课程，则其不上的课程不在1001号学生课程中。

SELECT DISTINCT SNO

FROM SC

WHERE NOT EXISTS

(SELECT \*

FROM SC SCx

WHERE SCx.SNO = 1001 AND NOT EXISTS

(SELECT \*

FROM SC SCy

WHERE SCy.SNO = SC.SNO AND SCy.cno = SCx.cno)); //就是一个Natural Join

考虑到SC中学号是有重复的，可以优化为

SELECT SNO

FROM Student

WHERE NOT EXISTS

(SELECT \*

FROM SC SCx

WHERE SCx.SNO = 1001 AND NOT EXISTS

(SELECT \*

FROM SC SCy

WHERE SCy.SNO = Student.SNO AND SCy.cno = SCx.cno));

[4] 查询选修了张星老师开设的全部课程的学生姓名。

//为第三题的加强版

SELECT SNAME

FROM Student

WHERE NOT EXISTS

(SELECT \*

FROM Course

WHERE TNO IN

(SELECT TNO

FROM Teacher

WHERE TNAME = '张星') AND NOT EXISTS

(SELECT \*

FROM SC

WHERE SC.SNO =Student.SNO AND SC.CNO = Course.CNO));

**5. 修改数据**

将张星老师数据结构课的学生成绩全部加2分。

UPDATE SC

SET GRADE = GRADE+2

WHERE ‘张星’=(SELECT TNAME

FROM TEACHER,COURSE

WHERE TEACHER.TNO = COURSE.TNO AND COURSE.CNAME = ‘数据结构’AND COURSE.CNO = SC.CNO);

**6. 删除数据**

删除马阳同学的所有选课记录。

DELETE

FROM SC,STUDENT

WHERE SC.SNO = STUDENT.SNO AND STUDENT.SNAME =‘马阳’;

**（三）视图操作**

**1. 建立视图**

在插入数据的Student基本表上为计算机科学与技术系的学生记录建立一个视图CS\_STUDENT。

CREATE VIEW CS\_STUDENT(SNO,SNAME,SEX,DEPTNO,AGE)

AS(SELECT STUDENT.\*

FROM STUDENT,DEPT

WHERE STUDENT.DEPTNO = DEPT.DEPTNO AND DEPT.DNAME = ‘计算机’)

**2. 删除视图**

在操作结束后删除视图CS\_STUDENT。

DROP VIEW CS\_STUDENT;