

# 数据库第五次作业

22373386 高铭

WEEK 5 & 6

## 书P130

---

### 5. 完成查询操作

#### (1) 找出所有供应商的姓名和所在城市

```
1  SELECT SNAME, CITY
2  FROM S;
```

#### (2) 找出所有零件的名称、颜色、重量

```
1  SELECT PNAME, COLOR, WEIGHT
2  FROM P;
```

#### (3) 找出使用供应商S1所供应零件的工程号码

```
1  SELECT JNO
2  FROM SPJ
3  WHERE SNO = 'S1';
```

#### (4) 找出工程项目J2使用的各种零件的名称及其数量

```
1  SELECT P.PNAME, SPJ.QTY
2  FROM P, SPJ
3  WHERE P.PNO = SPJ.PNO AND SPJ.JNO = 'J2';
```

#### (5) 找出上海厂商供应的所有零件号码

```
1  # 写法1
2  SELECT DISTINCT PNO
3  FROM SPJ
4  WHERE SNO IN
5      (SELECT SNO
6       FROM S
7       WHERE CITY = '上海');
```

```

1  # 写法2
2  SELECT DISTINCT SPJ.PNO
3  FROM S, SPJ
4  WHERE S.SNO = SPJ.SNO AND S.CITY = '上海';

```

## (6) 找出使用上海产的零件的工程名称

```

1  SELECT JNAME
2  FROM S, J, SPJ
3  WHERE S.CITY = '上海' AND S.SNO = SPJ.SNO AND SPJ.JNO = J.JNO;

```

## (7) 找出没有使用天津产的零件的工程号码

```

1  SELECT JNO
2  FROM J
3  WHERE NOT EXISTS
4      (SELECT *
5       FROM SPJ, S
6       WHERE S.CITY = '天津' AND S.SNO = SPJ.SNO AND SPJ.JNO = J.JNO);

```

# WEEK 5 - SQL 测验题

—

现有关系模式如下：

学生（学号，姓名，性别，年龄）；课程（课程号，课程名，教师姓名）；选课表（课程号，学号，成绩）

```
S(SID, SNAME, SEX, AGE); C(CID, CNAME, TEACHER); SC(CID, SID, SCORE)
```

## 1. 检索年龄大于20岁的男生的学号和姓名。

```

1  SELECT SID, SNAME
2  FROM S
3  WHERE AGE > 20 AND SEX = 'MALE';

```

## 2. 检索选修了姓刘的老师所教授的课程的女学生的姓名。

```

1  SELECT SNAME
2  FROM S, C, SC
3  WHERE C.TEACHER LIKE '刘%' AND C.CID = SC.CID AND SC.SID = S.SID AND S.SEX = 'FEMALE';

```

3. 检索李想同学不学的课程的课程号和课程名。

```
1  SELECT CID,CNAME
2  FROM C
3  WHERE NOT EXISTS
4      (SELECT *
5       FROM S,SC
6       WHERE S.SNAME='李想' AND S.SID=SC.SID AND SC.CID=C.CID);
```

4. 检索至少选修了两门课程的学生的学号。

```
1  SELECT SID
2  FROM SC
3  GROUP BY SID
4  HAVING COUNT(*)>=2;
```

5. 求刘老师所教授课程的每门课的平均成绩。

```
1  SELECT AVG(SC.SCORE)
2  FROM C,SC
3  WHERE C.TEACHER LIKE '刘%' AND C.CID=SC.CID
4  GROUP BY C.CID;
```

6. 假设不存在重修的情况，请统计每门课的选修人数(选课人数超过两人的课程才统计)。要求显示课程号和人数，查询结果按人数降序排列，若人数相同，按课程号升序排列。

```
1  SELECT CID,COUNT(SID)
2  FROM SC
3  GROUP BY CID
4  HAVING COUNT(SID)>2
5  ORDER BY COUNT(SID) DESC, CID;
```

7. 求年龄大于所有女生年龄的男性的姓名和年龄。

```
1  SELECT SNAME,AGE
2  FROM S
3  WHERE SEX='MALE' AND AGE>ALL
4      (SELECT AGE
5       FROM S
6       WHERE SEX='FEMALE');
```

8. 假定不存在重修的情况，求选修了所有课程的学生的学号姓名。(可以不用相关子查询做)

```
1  SELECT SID,SNAME
2  FROM S
3  WHERE NOT EXISTS
4      (SELECT *
5       FROM C
6       WHERE NOT EXISTS
7           (SELECT *
8            FROM SC
9            WHERE SC.SID=S.SID AND SC.CID=C.CID));
```

9. 查询重修次数在2次以上的学生学号，课程号，重修次数

```
1  SELECT SID,CID,COUNT(*)
2  FROM SC
3  GROUP BY SID,CID
4  HAVING COUNT(*)>2;
```

10. 查询重修学生人数最多的课程号，课程名，教师姓名

```
1  SELECT C.CID,C.CNAME,C.TEACHER
2  FROM C,SC
3  WHERE C.CID=SC.CID
4  GROUP BY SC.SID,SC.CID
5  HAVING COUNT(*) >= ALL
6      (SELECT COUNT(*)
7       FROM SC
8       GROUP BY SC.SID,SC.CID);
```

## 二

关系模式：学生（学号，姓名，年龄，性别，班级）

课程（课程号，课程名，先修课程号，学分）注意：此表的主键是(课程号)

选课（学号，课程号，教师号，成绩）

教师（教师号，教师名称）

```
1  S(SID, SNAME, AGE, SEX, CLASS)
2  C(CID, CNAME, CPID, CREDIT)
3  SC(SID, CID, TID, SCORE)
4  T(TID, TNAME)
```

### 1. 查找李力的所有不及格的课程名称和成绩，按成绩降序排列

```
1  SELECT C.CNAME, SC.SCORE
2  FROM S, C, SC
3  WHERE S.SNAME='李力' AND S.SID=SC.SID AND C.CID=SC.CID AND SC.SCORE<60
4  ORDER BY SC.SCORE DESC;
```

### 2. 列出每门课的学分，选修的学生人数，及学生成绩的平均分

```
1  SELECT CREDIT, CNT, AVGSORE
2  FROM C
3  JOIN
4      (SELECT CID, COUNT(SID) AS CNT, AVG(SCORE) AS AVGSORE
5       FROM SC
6       GROUP BY SC.CID) AS NEW
7  ON NEW.CID=C.CID;
```

### 3. 选出所修课程总学分在10分以下的学生（注：不及格的课程没有学分）。

```
1  SELECT S.*
2  FROM SC, C, S
3  WHERE C.CID=SC.CID AND SC.SCORE>=60 AND SC.SID=S.SID
4  GROUP BY SC.SID
5  HAVING SUM(C.CREDIT)<10;
```

### 4. 选出选课门数最多的学生学号及选课数量

```
1  SELECT SID, COUNT(CID)
2  FROM SC
3  GROUP BY SID
4  HAVING COUNT(CID)>ALL
5      (SELECT COUNT(CID)
6       FROM SC
7       GROUP BY SID);
```

### 5. 列出每门课的最高分及获得该分数的学生

```
1  SELECT CID, SCORE, S.*
2  FROM SC, S
3  WHERE SC.SID=S.SID AND SCORE=
4      (SELECT MAX(SCORE)
5       FROM SC AS SC2
6       WHERE SC.CID=SC2.CID
7       GROUP BY CID);
```

## 6. 选出物理课得分比所有男学生的物理课平均分高的学生姓名

```
1  SELECT S.SNAME
2  FROM S, SC, C
3  WHERE S.SID=SC.SID AND C.CID=SC.CID AND C.CNAME='物理' AND SCORE >
4         (SELECT AVG(SCORE)
5          FROM S, SC, C
6          WHERE S.SID=SC.SID AND C.CID=SC.CID AND C.CNAME='物理' AND S.SEX='MALE');
```

## 7. 选出修习过物理课的直接先修课的学生

```
1  SELECT S.*
2  FROM SC, C
3  WHERE SC.CID=C.CPID AND S.SID=SC.SID AND C.CPID=
4         (SELECT CID
5          FROM C
6          WHERE C.CNAME='物理');
```

## 8. 选出有两门以上先修课的课程（包括直接先修课、间接先修课）（用课程表）

```
1  SELECT *
2  FROM C
3  WHERE EXISTS
4         (SELECT *
5          FROM C C1, C C2
6          WHERE C.CPID=C1.CID AND C1.CPID=C2.CID);
```

# WEEK 6 - SQL练习

```
1  S(SID, SNAME, AGE, SEX, CLASS)
2  C(CID, CNAME, CPID, CREDIT)
3  SC(SID, CID, TID, SCORE, TIME)
4  T(TID, TNAME)
```

## 关系代数

### 1. 查找选修了物理课的学生姓名

$$\Pi_{\text{SNAME}}(\sigma_{\text{CNAME}='物理'}(C \bowtie S))$$

### 2. 查找教的学生的成绩都大于60分的教师（给出教师号即可）

$$\Pi_{\text{TID}}(T) - \Pi_{\text{TID}}(\sigma_{\text{SCORE} \leq 60}(T \bowtie SC \bowtie S))$$

### 3. 查找没有选修张三老师教的所有课的学生

$$\Pi_{SID}(S) - \Pi_{SID}(\sigma_{TNAME='张三'}(T \bowtie SC))$$

## SQL

### 1. 创建一个表，查询每个学生选修的课程数量，将结果插入表中

```
1 CREATE TABLE NEWTAB
2 (SID CHAR(20) NOT NULL PRIMARY KEY,
3  SNAME CHAR NOT NULL,
4  CNUM INT);
5 INSERT
6 INTO NEWTAB(SID, SNAME, CNUM)
7 SELECT S.SID, S.SNAME, COUNT(SC.CID)
8 FROM S
9 WHERE S.SID=
10 (SELECT SC.SID
11 FROM SC
12 GROUP BY SC.SID
13 );
```

### 2. 找出所有姓诸的学生姓名（排除姓‘诸葛’的学生）

```
1 SELECT SNAME
2 FROM S
3 WHERE SNAME LIKE '诸%' AND SNAME NOT LIKE '诸葛%';
```

### 3. 检索至少得过一次课程最高分的学生学号姓名（不考虑重修的情况）

```
1 SELECT DISTINCT SID, SNAME
2 FROM S
3 WHERE SID =
4 (SELECT SID
5 FROM SC SC1
6 WHERE SC1.SCORE =
7 (SELECT MAX(SC2.SCORE)
8 FROM SC SC2
9 GROUP BY SC2.CID));
```

### 4. 查询如下内容（学生ID，课程ID，时间），列出每个学生第一次选某课程的时间（即非重修的选课时间）

```
1 SELECT SC.SID, SC.CID, MIN(SC.TIME)
2 FROM SC
3 GROUP BY SC.SID, SC.CID;
```

## 5. 将学生的重修课程成绩都改成60分

```
1  # 理解1: 首修仍是原来分数, 重修那次的选课改成60分
2  UPDATE SC
3  SET SCORE = 60
4  WHERE (SID, CID, TIME) IN (
5      SELECT SID, CID, MAXTIME
6      FROM(
7          SELECT SID, CID, MAX(TIME) AS MAXTIME, COUNT(*)
8          FROM SC
9          GROUP BY SID,CID
10         HAVING COUNT(*) > 1
11     )
12 );
```

```
1  # 理解2: 对于有过重修的课, 成绩记录都改成60分
2  UPDATE SC
3  SET SCORE = 60
4  WHERE (SID, CID) IN (
5      SELECT SID, CID
6      FROM (
7          SELECT SID, CID, COUNT(*)
8          FROM SC
9          GROUP BY SID, CID
10         HAVING COUNT(*) > 1
11     )
12 );
```

## 6. 查找每个学生当前可选修的课程列表 (即该学生没有选该课程, 且该学生已经修完了该课程的先修课)

```
1  SELECT S.SID, C.CID, C.CNAME
2  FROM S
3  CROSS JOIN C          # 创建所有学生和课程的组合
4  LEFT JOIN SC ON S.SID=SC.SID AND C.CID=SC.CID
5  WHERE SC.CID IS NULL  # 该学生没有选该课程
6  AND EXISTS (
7      SELECT *
8      FROM C AS C2
9      JOIN SC AS SC2 ON C2.CID=SC2.CID
10     WHERE C2.CPID=C.CID AND SC2.SID=S.SID
11 );                    # 该学生已经修完了该课程的先修课
```