

Database HW2

王章瀚 PB18111697

1

给定基本表

```
-- 学生表(sno 是主键):
Student(sno:char(10),sname: varchar(50),birthdate: date)
-- 课程表(cno 是主键):
Course(cno:char(10),cname: varchar(50),type: int,credit: float)
-- 选课表(sno, cno, term):
SC(sno,cno,score: float,term:int)
-- type: 类型, 0 表示必修课, 1 表示选修课, 2 表示通识课, 3 表示公选课
-- credit: 课程学分
-- 第几学期, 取值范围为 1-8
```

用 SQL 语句回答下面查询:

(1)

查询姓名中含有 "科" 字的学生学号和姓名.

```
Select sno, sname From Student
Where sname Like '%科%';
```

(2)

查询学分不低于3分的必修课课程号和课程名

```
Select cno, cname From Course
Where Course.type = 0 and Course.credit >= 3;
```

(3)

查询选修了公选课但是缺少成绩的学生学号和姓名

```
Select Distinct s.sno, s.sname
from Student s, Course c, SC sc
Where s.sno = sc.sno and sc.cno = c.cno
      and c.type = 3 and sc.score = NULL; -- 选了公选课 且 缺少成绩
```

(4)

查询年龄大于20的学生学号、姓名和年龄

```
Select sno, sname,
      timeStampDiff(YEAR, birthdate, now()) As age
From Student
Where age > 20;
```

(5)

查询已选必修课总学分大于16并且所选通识课成绩都大于75分的学生姓名

```
Select s.sname
From Student s, (
  Select sno -- 必修课总学分大于16的学生
  From Course c, SC sc
  Where s.sno = sc.sno and c.cno = sc.cno -- 对应人和课
        and c.type = 0 -- 必修课
  Group By s.sno
  Having sum(sc.credit) > 16) sc2
Where sc2.sno = s.sno
      and NOT EXISTS ( -- s.sno 不存在小于等于75分的通识课成绩
  Select *
  From Course c, SC sc
  Where s.sno = sc2.sno -- 特定学生 s.sno
        and c.type = 2 and sc.score <= 75 -- 通识课成绩 <= 75
);
```

(6)

查询已经修完所有必修课且成绩合格的学生学号和姓名

```

Select sno, sname
From Student s
Where NOT EXISTS ( -- 该学生不存在:
    Select * From Course -- 某门课
    Where type = 0 -- 是必修课
        and cno NOT IN ( -- 且在该学生已选课程中分数大于等于60
            Select Distinct cno From SC
            Where sno = Student.sno and sc.score >= 60
        )
);

```

(7)

查询总平均成绩排名在前50%（向上取整）的学生中必修课平均分最高的前10位同学，要求返回这些学生的学号、姓名、必修课平均分以及课程总平均成绩（不足10位时则全部返回）

不用 join , 则以 (学号, 姓名) 为唯一区分标志:

```

Select s.sno, s.sname, allAvg, avg(sc.score) As required_avg
From (
    Select sno, avg(sc.score) As all_avg -- 总平均成绩排名前 50% (ceil) 的学生
    From Student s, Course c, SC sc
    Where s.sno = sc.sno and c.cno = sc.cno
        and all_avg >= (
            Select all_avg_50 From (
                Select avg(sc.score) As all_avg_50,
                    row_number() over (Order By all_avg_50 DESC)
                From Student s, Course c, SC sc
                Where s.sno = sc.sno and c.cno = sc.cno
                Group By sno
            )
            Where r = (Select ceil(count(*)*0.5) From Student) -- 排名恰为 50% 上取整的排名序号
        )
    Group By sno
) sno_all_avg50, Student s, Course c, SC sc
Where sno_all_avg50.sno = sc.sno and c.cno = sc.cno
    and c.type = 0
Group By sno, sname, all_avg
Order By required_avg
Limit 10;

```

若使用 join :

```

Select s.sno, s.sname, allAvg, required_avg
From ( -- 总平均成绩排名前 50% (ceil) 的学生及其成绩表
    Select sno, avg(sc.score) As all_avg
    From Student s, Course c, SC sc
    Where s.sno = sc.sno and c.cno = sc.cno
    and all_avg >= (
        Select all_avg_50 From (
            Select avg(sc.score) As all_avg_50,
                row_number() over (Order By all_avg_50 DESC)
            From Student s, Course c, SC sc
            Where s.sno = sc.sno and c.cno = sc.cno
            Group By sno
        )
        Where r = (Select ceil(count(*)*0.5) From Student) -- 排名恰为 50% 上取整的排名序号
    )
    Group By sno
) all_avg50
Natural Join ( -- 各学生的必修课平均成绩表
    Select s.sno, avg(sc.score) As required_avg
    From Student s , Course c, SC sc
    Where c.cno = sc.cno and c.type = 0
    Group By sno
) total_required
Natural Join Student s -- 连入学生信息
Order By required_avg
Limit 10

```

(8)

查询每门课程的课程名、课程类型、最高成绩、最低成绩、平均成绩和不及格率，要求结果按通识课、必修课、选修课、公选课顺序排列（提示：课程名可能有重名）

```

Select c.cname, c.type, max(sc.score) max_score, min(sc.score) min_score, avg(sc.score) avg_
From Course c, SC sc, (
    Select c.cno, failed_count
    From (
        Select c.cno, count(*) As failed_count
        From Course c, SC sc
        Where sc.score < 60
        Group By c.cno
    ) failed_count_tb_part
    Union (
        Select c.cno, 0 As failed_count
        Where not exists (
            Select * from Sc sc
            Where sc.cno = c.cno and sc.score < 60;
        )
    ) failed_count_tb
Where c.cno = failed_count_tb.cno
Group By c.cno, c.cname, c.ctype, failed_count_tb.failed_count
Order By (
    Case c.type
    When 2 Then 0    -- 通识课
    When 0 Then 1    -- 必修课
    When 1 Then 2    -- 选修课
    Else 3           -- 公选课
)

```

(9)

查询存在课程重修不及格情况的学生学号、姓名以及重修不及格的课程号和课程名

```

Select Distinct s.sno, s.sname, c.cno, c.cname
From Student s, Course c, SC sc
Where s.sno = sc.sno and c.cno = sc.cno
Group By s.sno, s.sname, c.cno, c.cname
Having count(*) > 1 and max(sc.score) < 60 -- 重修不及格

```

(10)

SC表中重复的sno和cno意味着该学生重修了课程（在不同的学期里），现在我们希望删除学生重复选课的信息，只保留最近一个学期的选课记录以及成绩，请给出相应的SQL语句

```
Delete From SC sc
Where Exists (
    Select sc.sno, sc.cno From (
        Select sc.sno, sc.cno, max(sc.term) As latest_term
        where sc.score is not null -- 必须要有成绩
        Group By sc.sno, sc.cno
    ) latest_retake
Where sc.sno = latest_retake.sno and sc.cno = latest_retake.cno
    and sc.term < latest_retake.latest_term -- 学期早于最近学期
)
```