数据库实验报告(第三周)

班 级: 3班 姓 名: 梁力航 学 号: 23336128

一、实验目的

- 熟悉SQL的数据查询语言,
- 能够使用SQL语句对数据库进行单表查询、 连接查询

二、实验内容

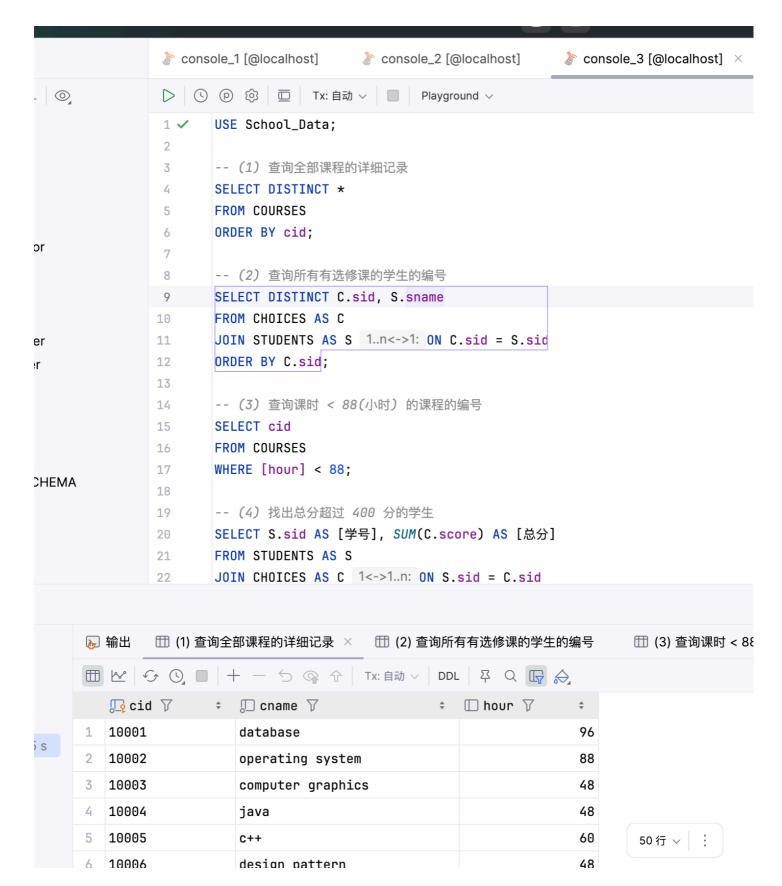
- 1. 查询全部课程的详细记录, 并按课程编号升序
- 2. 查询所有有选修课的学生编号及姓名
- 3. 查询课时 < 88 的课程编号
- 4. 找出总分超过 400 分的学生
- 5. 统计课程总数
- 6. 查询所有课程及其被选修的学生总数
- 7. 查询选修成绩 > 60 的课程数量超过 2 门的学生编号及最低分
- 8. 统计每个学生的选修课程数目与平均成绩
- 9. 查询选修 Java 的所有学生的编号及姓名
- 10. 查询姓名为 sssht 的学生所选课程的编号和成绩
- 11. 查询课时比课程 C++ 多的课程名称

说明: 所有查询均在 USE School_Data; 上下文下执行。

三、实验结果

(1) 查询全部课程的详细记录

USE School_Data;
SELECT DISTINCT *
FROM COURSES
ORDER BY cid;



(2) 查询所有有选修课的学生的编号

```
SELECT DISTINCT C.sid, S.sname
FROM CHOICES AS C
JOIN STUDENTS AS S ON C.sid = S.sid
ORDER BY C.sid;
```

```
-- (2) 查询所有有选修课的学生的编号
        8
        9
             SELECT DISTINCT C.sid
             FROM CHOICES as C
       10
             ORDER BY C.sid;
       11
       12
             -- (3) 查询课时 < 88(小时) 的课程的编号
       13
             SELECT cid
       14
             FROM COURSES
       15
             WHERE [hour] < 88;
       16
       17
             -- (4) 找出总分超过 400 分的学生
       18
       19
             SELECT S.sid AS [学号], SUM(C.score) AS [总分]
             FROM STUDENTS AS S
       20
             JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
       21
             GROUP BY S.sid
       22
№ 输出 (1) 查询全部课程的详细记录
                             Ⅲ (2) 查询所有有选修课的学生的编号 ×
                                                          Ⅲ (3) 查询课时 < 88(小时
∏ sid 🏹
     800001216
 1
     800002933
 3
     800005753
     800006682
 4
 5
    800006941
                                                         100,000 行 ∨
 6 800007595
```

(3) 查询课时 < 88 的课程的编号

SELECT cid FROM COURSES WHERE [hour] < 88;

```
-- (3) 查询课时 < 88(小时) 的课程的编号
             SELECT cid
       15
             FROM COURSES
             WHERE [hour] < 88;
        16
       17
        18
             -- (4) 找出总分超过 400 分的学生
             SELECT S.sid AS [学号], SUM(C.score) AS [总分]
       19
             FROM STUDENTS AS S
        20
             JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
       21
        22
             GROUP BY S.sid
             HAVING SUM(C.score) > 400;
       23
Ⅲ (2) 查询所有有选修课的学生的编号
                                                     Ⅲ (3) 查询课时 < 88(小时) 的课程的编号 ×
ু cid ৴ ÷
1 10003
2 10004
3 10005
4 10006
5 10007
                                                  48 行 🗸 :
6 10008
```

(4) 找出总分超过 400 分的学生

```
SELECT S.sid AS [学号], SUM(C.score) AS [总分]
FROM STUDENTS AS S
JOIN CHOICES AS C ON S.sid = C.sid
GROUP BY S.sid
HAVING SUM(C.score) > 400;
```

```
-- (4) 找出总分超过 400 分的学生
        18
               SELECT S.sid AS [学号], SUM(C.score) AS [总分]
        19
               FROM STUDENTS AS S
        20
               JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
        21
               GROUP BY S.sid
        22
               HAVING SUM(C.score) > 400;
        23
        24
               -- (5) 查询课程的总数
         25
    冊 (4) 找出总分超过 400 分的学生 × 冊 (5) 查询课程的总数
                                                 Ⅲ (6) 查询所有课程和选修该课
□ 学号 ▽ 💠 □ 总分 ▽
    886690009
                              415
    871307310
                              428
3
    848788630
                              412
   802007306
                              439
4
5
  836636077
                              403
                                                         3,480 行 🗸
6 820723618
                              402
```

(5) 查询课程的总数

SELECT COUNT(*) AS [课程总数] FROM COURSES;

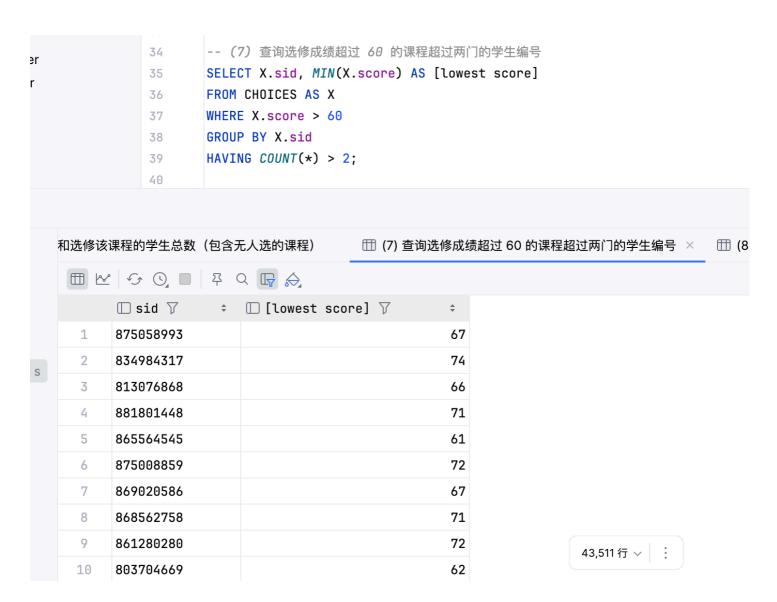


(6) 查询所有课程和选修该课程的学生总数

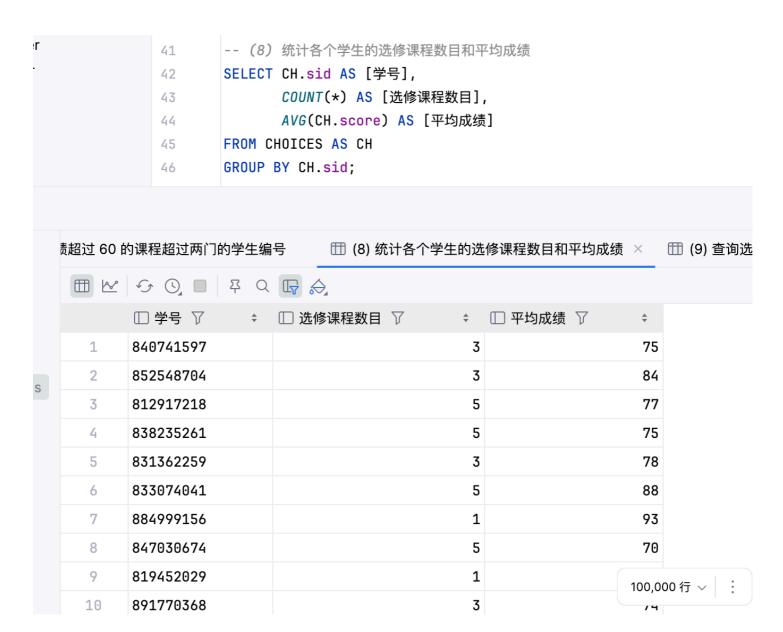
```
SELECT C.cid, COUNT(CH.sid) AS [number of students]
FROM COURSES AS C
LEFT JOIN CHOICES AS CH ON CH.cid = C.cid
GROUP BY C.cid;
                  -- (6) 查询所有课程和选修该课程的学生总数(包含无人选的课程)
           29
                 SELECT C.cid, COUNT(CH.sid) AS [number of students]
                 FROM COURSES AS C
           30
                 LEFT JOIN CHOICES AS CH 1<->0..n: ON CH.cid = C.cid
                  GROUP BY C.cid;
           32
           33
号
      Ⅲ (4) 找出总分超过 400 分的学生
                                   Ⅲ (5) 查询课程的总数
                                                        Ⅲ (6) 查询所有课程和选修该课程的学生总数(包含无人选的课程)
 \square cid \triangledown \Rightarrow \square [number of students] \triangledown
 6 10006
                                                  6090
 7 10007
                                                  5965
 8 10008
                                                  5985
 9
   10009
                                                  5965
10 10010
                                                  6027
                                                  6086
11 10011
12 10012
                                                  5819
13 10013
                                                  5982
                                                                 50 行 🗸 :
14 10014
                                                  5866
```

(7) 查询选修成绩超过 60 的课程超过两门的学生编号

```
SELECT X.sid, MIN(X.score) AS [lowest score]
FROM CHOICES AS X
WHERE X.score > 60
GROUP BY X.sid
HAVING COUNT(*) > 2;
```



(8) 统计各个学生的选修课程数目和平均成绩



(9) 查询选修 Java 的所有学生的编号及姓名

```
SELECT C.sid AS [学号], S.sname AS [姓名] FROM CHOICES AS C
JOIN STUDENTS AS S ON C.sid = S.sid
JOIN COURSES AS K ON C.cid = K.cid
WHERE K.cname = 'java';
```

```
-- (9) 查询选修 Java 的所有学生的编号及姓名
               SELECT C.sid AS [学号], S.sname AS [姓名]
         49
               FROM CHOICES AS C
         50
               JOIN STUDENTS AS S 1..n<->1: ON C.sid = S.sid
         51
               JOIN COURSES AS K 1..n<->1: ON C.cid = K.cid
         52
               WHERE K.cname = 'java';
         54
责超过 60 的课程超过两门的学生编号
                          Ⅲ (8) 统计各个学生的选修课程数目和平均成绩
                                                           Ⅲ (9) 查询选修 Java 的所有学生的编号及姓名
 □ 学号 🍸
                ≑ □姓名 ▽
  1 821494745
                   ieruur
    821830521
                   efwztecfe
    821912373
                   txdlp
    821932263
                   jhuigkrp
    822025514
                   vyxhtwes
    822532579
                   alndh
    822688835
                   atpmlsv
    823235974
                   hcduuo
  9
    823502688
                   oxahodeu
                                                        6,110 行 🗸
 10 823889103
                   zmmke
```

(10) 查询姓名为 sssht 的学生所选的课程的编号和成绩

```
SELECT C.cid AS [课程编号], C.score AS [成绩]
FROM CHOICES AS C
JOIN STUDENTS AS S ON C.sid = S.sid
WHERE S.sname = 'sssht';
               -- (10) 查询姓名为 sssht 的学生所选课程的编号和成绩
               SELECT C.cid AS [课程编号], C.score AS [成绩]
               FROM CHOICES AS C
         57
              JOIN STUDENTS AS S 1..n<->1: ON C.sid = S.sid
         58
               WHERE S.sname = 'sssht';
修课程数目和平均成绩
              Ⅲ (9) 查询选修 Java 的所有学生的编号及姓名
                                              Ⅲ (10) 查询姓名为 sssht 的学生所选课程的编号和成绩 >
 □ 课程编号 ▽
               1 10004
                              76
 2 10030
                              53
 3 10019
                              79
 4 10037
                              84
 5 10037
                              54
```

(11) 查询其他课时比课程 C++ 多的课程的名称

```
SELECT X.cname
 FROM COURSES AS X
 JOIN COURSES AS Y ON Y. cname = 'C++'
 WHERE X.cname <> 'C++' AND X.[hour] > Y.[hour];
er
                      -- (11) 查询课时比课程 C++ 多的课程的名称
               61
               62
                      SELECT X.cname
                      FROM COURSES AS X
               63
                      JOIN COURSES AS Y ON Y.cname = 'C++'
               64
                      WHERE X.cname <> 'C++' AND X.[hour] > Y.[hour];
               65
   召过两门的学生编号
                      Ⅲ (11) 查询课时比课程 C++ 多的课程的名称 × Ⅲ (8) 统计各个学生的选修课程数
     DDL ♀ Q 🖫 쉱
       \square cname 
abla
    1 database
    2 operating system
    3 tcp/ip protocol
    4 algorithm
    5 compiling principle
                                                                     5 行 🗸 📑
```

四、问题&总结

- 第(6)题:一开始用的是内连接join,但是"没人选"的课程就没考虑到了。把它换成 LEFT JOIN,这样就能把所有课程都数进来了。
- 第(3)题: 列名叫 hour, 容易和时间相关的关键词撞车。给它加上方括号 [hour] 更清楚。
- 别名写法:原先用单引号当别名,查询后发现SQL Server 里不太推荐。统一改成方括号,比如 [学号] 、 [总分] ,更清晰也更规范。
- DISTINCT: 第(6)题已经 GROUP BY 了,再写 DISTINCT 就重复了,所以把 DISTINCT 去掉,语句更干净。
- 连接风格:第(9)(10)(11)原来用逗号把表连起来,不太直观。改成显式 JOIN,逻辑一眼就能看懂,也更不容易踩坑。