

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; 上下文中执行。

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;
```



```

8      -- (2) 查询所有有选修课的学生的编号
9      SELECT DISTINCT C.sid
10     FROM CHOICES as C
11     ORDER BY C.sid;
12
13     -- (3) 查询课时 < 88(小时) 的课程编号
14     SELECT cid
15     FROM COURSES
16     WHERE [hour] < 88;
17
18     -- (4) 找出总分超过 400 分的学生
19     SELECT S.sid AS [学号], SUM(C.score) AS [总分]
20     FROM STUDENTS AS S
21     JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
22     GROUP BY S.sid

```

 输出
  (1) 查询全部课程的详细记录
  (2) 查询所有有选修课的学生的编号 ×
  (3) 查询课时 < 88(小时)











 Tx: 自动 ▾ | DDL |    

| | sid |
|---|-----------|
| 1 | 800001216 |
| 2 | 800002933 |
| 3 | 800005753 |
| 4 | 800006682 |
| 5 | 800006941 |
| 6 | 800007595 |

100,000 行 ▾ | ⋮

(3) 查询课时 < 88 的课程的编号

```





SELECT cid
FROM COURSES
WHERE [hour] < 88;

```

```

12
13 -- (3) 查询课时 < 88(小时) 的课程编号
14 SELECT cid
15 FROM COURSES
16 WHERE [hour] < 88;
17
18 -- (4) 找出总分超过 400 分的学生
19 SELECT S.sid AS [学号], SUM(C.score) AS [总分]
20 FROM STUDENTS AS S
21 JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
22 GROUP BY S.sid
23 HAVING SUM(C.score) > 400;

```

 输出
  (1) 查询全部课程的详细记录
  (2) 查询所有有选修课的学生的编号
  (3) 查询课时 < 88(小时) 的课程编号 ×

     + - ↶ ↷ ↵ Tx: 自动 ▾ DDL    

| | cid |
|---|-------|
| 1 | 10003 |
| 2 | 10004 |
| 3 | 10005 |
| 4 | 10006 |
| 5 | 10007 |
| 6 | 10008 |

48 行 ▾ ⋮

(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;

```

```

18 -- (4) 找出总分超过 400 分的学生
19 SELECT S.sid AS [学号], SUM(C.score) AS [总分]
20 FROM STUDENTS AS S
21 JOIN CHOICES AS C 1<->1..n: ON S.sid = C.sid
22 GROUP BY S.sid
23 HAVING SUM(C.score) > 400;
24
25 -- (5) 查询课程的总数
26 SELECT COUNT(*) FROM COURSES;

```

📄 (4) 找出总分超过 400 分的学生 ×
 📄 (5) 查询课程的总数
📄 (6) 查询所有课程和选修该课

| | 📄 学号 🔽 | 📄 总分 🔽 | |
|---|-----------|--------|--|
| 1 | 886690009 | 415 | |
| 2 | 871307310 | 428 | |
| 3 | 848788630 | 412 | |
| 4 | 802007306 | 439 | |
| 5 | 836636077 | 403 | |
| 6 | 820723618 | 402 | |

3,480 行 ▾ | ⋮

(5) 查询课程的总数

```

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

```

```

25 -- (5) 查询课程的总数
26 SELECT COUNT(*) as [课程总数] FROM COURSES;
27

```

📄 输出
📄 (5) 查询课程的总数 ×

| | 📄 课程总数 🔽 | |
|---|----------|--|
| 1 | 50 | |

(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;
```

```
28 -- (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]
30 FROM COURSES AS C
31 LEFT JOIN CHOICES AS CH ON CH.cid = C.cid
32 GROUP BY C.cid;
33
```

号

(4) 找出总分超过 400 分的学生

(5) 查询课程的总数

(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)

| | cid | [number of students] |
|----|-------|----------------------|
| 6 | 10006 | 6090 |
| 7 | 10007 | 5965 |
| 8 | 10008 | 5985 |
| 9 | 10009 | 5965 |
| 10 | 10010 | 6027 |
| 11 | 10011 | 6086 |
| 12 | 10012 | 5819 |
| 13 | 10013 | 5982 |
| 14 | 10014 | 5866 |

50 行

(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;
```

```
34 -- (7) 查询选修成绩超过 60 的课程超过两门的学生编号
35 SELECT X.sid, MIN(X.score) AS [lowest score]
36 FROM CHOICES AS X
37 WHERE X.score > 60
38 GROUP BY X.sid
39 HAVING COUNT(*) > 2;
40
```

和选修该课程的学生总数（包含无人选的课程）

(7) 查询选修成绩超过 60 的课程超过两门的学生编号 × (8)

| | sid | [lowest score] |
|----|-----------|----------------|
| 1 | 875058993 | 67 |
| 2 | 834984317 | 74 |
| 3 | 813076868 | 66 |
| 4 | 881801448 | 71 |
| 5 | 865564545 | 61 |
| 6 | 875008859 | 72 |
| 7 | 869020586 | 67 |
| 8 | 868562758 | 71 |
| 9 | 861280280 | 72 |
| 10 | 803704669 | 62 |

43,511 行

(8) 统计各个学生的选修课程数目和平均成绩

```
SELECT CH.sid AS [学号],
       COUNT(*) AS [选修课程数目],
       AVG(CH.score) AS [平均成绩]
FROM CHOICES AS CH
GROUP BY CH.sid;
```

```
41 -- (8) 统计各个学生的选修课程数目和平均成绩
42 SELECT CH.sid AS [学号],
43        COUNT(*) AS [选修课程数目],
44        AVG(CH.score) AS [平均成绩]
45 FROM CHOICES AS CH
46 GROUP BY CH.sid;
```

责超过 60 的课程超过两门的学生编号 (8) 统计各个学生的选修课程数目和平均成绩 × (9) 查询选

| | 学号 | 选修课程数目 | 平均成绩 |
|----|-----------|--------|------|
| 1 | 840741597 | 3 | 75 |
| 2 | 852548704 | 3 | 84 |
| 3 | 812917218 | 5 | 77 |
| 4 | 838235261 | 5 | 75 |
| 5 | 831362259 | 3 | 78 |
| 6 | 833074041 | 5 | 88 |
| 7 | 884999156 | 1 | 93 |
| 8 | 847030674 | 5 | 70 |
| 9 | 819452029 | 1 | |
| 10 | 891770368 | 3 | |

100,000 行

(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';
```



```

48  -- (9) 查询选修 Java 的所有学生的编号及姓名
49  SELECT C.sid AS [学号], S.sname AS [姓名]
50  FROM CHOICES AS C
51  JOIN STUDENTS AS S 1..n<->1: ON C.sid = S.sid
52  JOIN COURSES AS K 1..n<->1: ON C.cid = K.cid
53  WHERE K.cname = 'java';
54

```

超过 60 的课程超过两门的学生编号

(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 ×

Tx: 自动 DDL

| | 学号 | 姓名 |
|----|-----------|-----------|
| 1 | 821494745 | ieruur |
| 2 | 821830521 | efwztecfe |
| 3 | 821912373 | txdlp |
| 4 | 821932263 | jhuigkrp |
| 5 | 822025514 | vyxhtwes |
| 6 | 822532579 | alndh |
| 7 | 822688835 | atpmlsv |
| 8 | 823235974 | hcduuu |
| 9 | 823502688 | oxahodeu |
| 10 | 823889103 | zmmke |

6,110 行

(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';

```

```

54
55  -- (10) 查询姓名为 sssht 的学生所选课程的编号和成绩
56  SELECT C.cid AS [课程编号], C.score AS [成绩]
57  FROM CHOICES AS C
58  JOIN STUDENTS AS S 1..n<->1: ON C.sid = S.sid
59  WHERE S.sname = 'sssht';

```

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

(10) 查询姓名为 sssht 的学生所选课程的编号和成绩 ×

Tx: 自动 DDL

| | 课程编号 | 成绩 |
|---|-------|----|
| 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];
```

61

-- (11) 查询课时比课程 C++ 多的课程名称

62

SELECT X.cname

63

FROM COURSES AS X

64

JOIN COURSES AS Y ON Y.cname = 'C++'

65

WHERE X.cname <> 'C++' AND X.[hour] > Y.[hour];

超过两门的学生编号

(11) 查询课时比课程 C++ 多的课程名称

(8) 统计各个学生的选修课程数

📄

📊

🔄

🕒

⬛

+

−

↶

🔄

⬆

Tx: 自动

DDL

🔗

🔍

📄

🔗

📄

cname

🔍

⬆

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。