# 2实验报告

#### 实验目的

实验环境

实验过程中主要使用的SQL语句类型

实验内容

自我实践

常见错误

# 实验目的

- 理解和掌握数据库查询语言(SQL)的基本概念和操作。
- 学习如何使用SQL语句进行数据检索、过滤、排序和聚合。
- 练习编写复杂的查询以解决实际问题。
- 查询的目标表达式为所有列、指定列或指定列的运算。
- 使用DISTINCT保留字消除重复行。
- 对查询结果排序和分组。
- 集合分组使用集函数进行各项统计。

# 实验环境

- 数据库管理系统 PostgreSQL
- 包含所需表和数据的数据库实例。

在该数据库中存在4张表格,分别为:

STUDENTS(sid,sname,email,grade)

TEACHERS(tid,tname,email,salary)

COURSES(cid,cname,hour)

CHOICES(no,sid,tid,cid,score)

# 实验过程中主要使用的SQL语句类型

1. SELECT: 基本的查询语句,用于检索数据。

2. WHERE: 用于设置查询条件。

3. ORDER BY: 用于对结果进行排序。

4. JOIN:用于连接多个表。

5. GROUP BY: 用于对结果进行分组。

6. HAVING: 用于对分组后的结果进行过滤。

7. AVG: 计算平均值。

8. COUNT: 计算数量。

9. SUM: 计算总和。

10. DISTINCT: 用于去除重复的记录。

# 实验内容

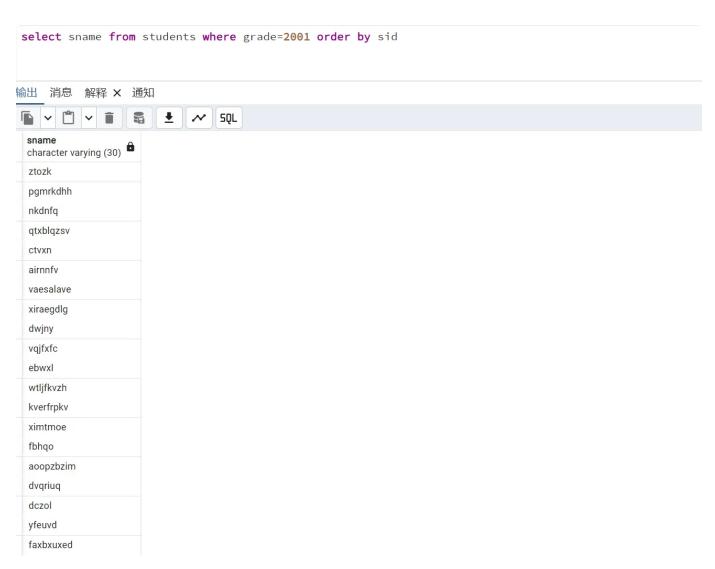
在数据库中,存在这样的关系:学生可以选择课程。一个课程对应一个教师。在表CHOICES中保存学生的选课记录。

1. 查询年级为2001的所有学生的名称,按编号升序排列;

select sname from students where grade=2001 order by sid

消息

已成功运行。查询运行时间总数: 84 毫秒。 6754 行受影响。



2. 查询学生的选课成绩合格的课程成绩,并把成绩换算为积点(60分对应积点为1,每增加1分,积点增加0.1);

select no,sid, tid,cid,(score-60)\*0.1 from choices where score>=6

#### 消息

已成功运行。查询运行时间总数: 348 毫秒。 239594 行受影响。

3.

3 select no,sid, tid,cid,(score-60)\*0.1 from choices where score>=60

#### 数据输出 消息 解释 🗙 通知

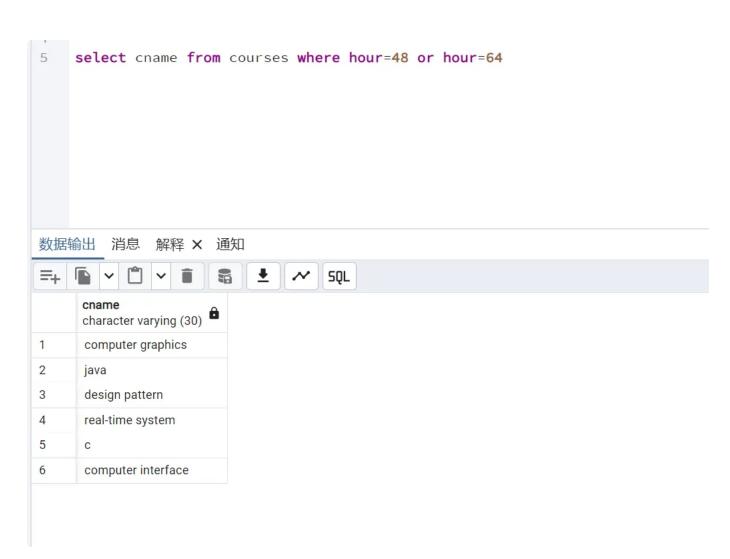
=+ [	~ <u></u>		<b>~</b> SQL		
	no [PK] integer	sid character (9)	tid character (9)	cid character (5)	?column? numeric
1	500000058	823069829	249596497	10037	1.6
2	500000253	829348273	202560416	10010	2.7
3	500001270	847061074	292043491	10025	3.2
4	500002143	860635914	238811498	10039	2.2
5	500002371	829785562	273189968	10028	1.7
6	500002715	822137137	218922066	10011	0.7
7	500003262	826310502	267846042	10005	3.0
8	500004144	817636568	253205179	10047	0.0
9	500005935	801967882	234419511	10021	1.0
10	500005987	875434315	223646385	10048	2.2
11	500006353	830180555	218440500	10016	1.6
12	500006805	848035070	208952048	10007	2.8
13	500006824	834091581	201353263	10049	1.2
14	500007004	809548802	210870137	10002	0.4
15	500007637	833961570	209860626	10004	2.0
16	500007901	894256303	209831209	10018	1.5
17	500008311	826412145	200207128	10009	0.3
18	500008620	836887807	294083883	10031	1.4
10	500000000	004520000	200001701	10022	2 0

4. 查询课时是48或64的课程的名称;

#### select cname from courses where hour=48 or hour=64

#### 消息

已成功运行。查询运行时间总数: 65 毫秒。 6 行受影响。



5. 查询所有课程名称中含有data的课程编号;

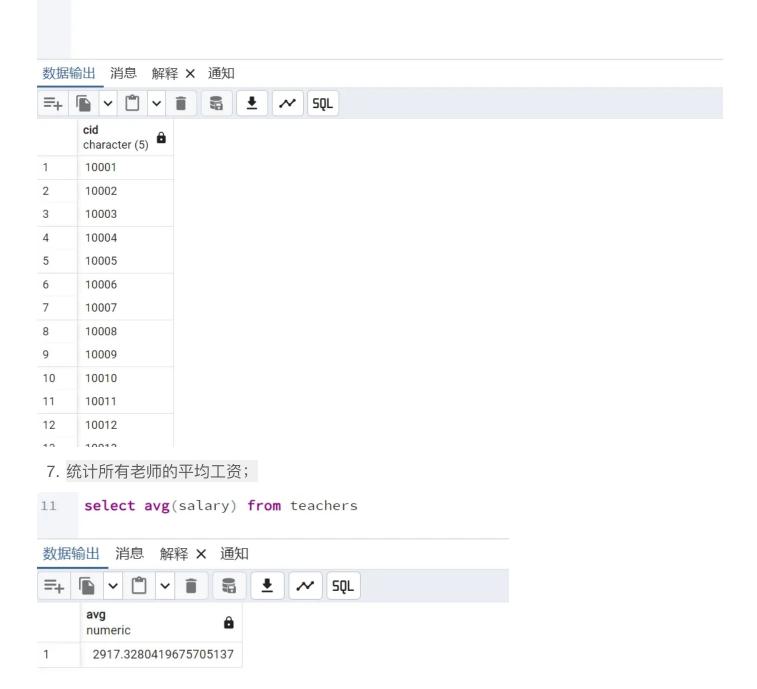
6. 查询所有选课记录的课程号(不重复显示);

select distinct cid from choices

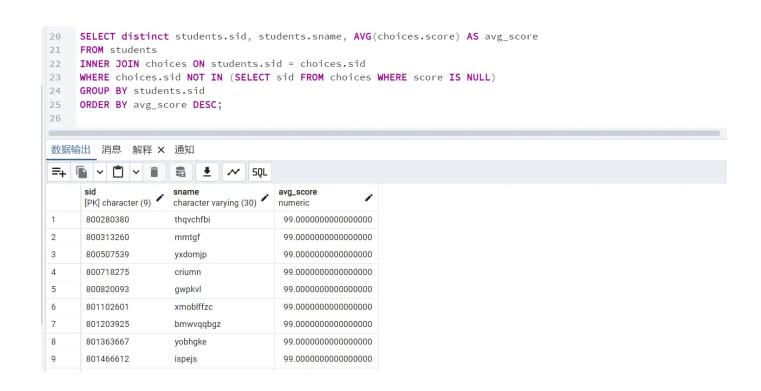
#### 消息

已成功运行。查询运行时间总数: 256 毫秒。 50 行受影响。

9 select distinct cid from choices



8. 查询所有学生的编号, 姓名和平均成绩, 按总平均成绩降序排列;



# 9. 统计各个课程的选课人数和平均成绩;

```
32 v select count(distinct sid),cid ,avg(score)
      from choices
33
      group by cid
34
      order by cid
35
36
数据输出
          消息
                 解释×通知
                                          SQL
=+
      count
                                 avg
                                                   ۵
                  character (5)
                                 numeric
      bigint
                  10001
1
            5757
                                  75.9724011039558418
2
            5853
                  10002
                                  75.8944020356234097
3
                  10003
                                  75.9430183870380484
            5811
            5952
                  10004
                                  76.1287462316013478
4
            5876
                  10005
                                  76.0171449196895867
5
                                  76.0699625200785294
6
            5913
                  10006
7
            5825
                  10007
                                  75.9119038951583546
8
            5825
                  10008
                                  75.6591074020319303
9
            5807
                  10009
                                  76.2119555312556953
            5881
                  10010
                                  75.8274111675126904
10
            5937
                  10011
                                  76.3341042519124711
11
            5669
                  10012
                                  76.1280852655198205
12
13
            5833
                  10013
                                  75.6476068688344903
                  10014
                                  76.1030146106898465
14
            5712
```

75.8496732026143791

76.0415129151291513

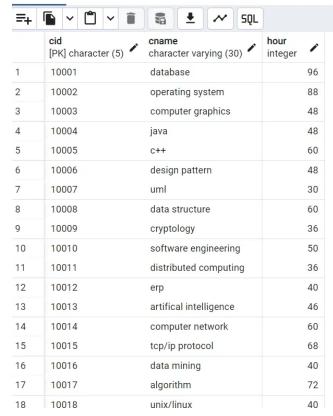
### 10. 查询至少选修了三门课程的学生编号。

```
38
      select count(cid),sid
39
      from choices
40
      group by sid
      having count(cid)>=3
41
数据输出 消息 解释 × 通知
                              <u>*</u>
                                       SQL
                                   ~
       count
                 character (9)
       bigint
                  805139598
1
2
                  862064110
               4
3
               4
                  892437507
                  810913195
4
               3
5
               4
                  801893096
                  889788630
6
               3
7
                  869459336
8
                  881707540
9
               4
                  899016218
                  872267782
10
               5
               3
                  808645017
11
               5
                  837195907
12
                  881336524
13
                  853084873
14
15
               5
                  830128474
                  885085902
16
                  004400000
```

# 自我实践

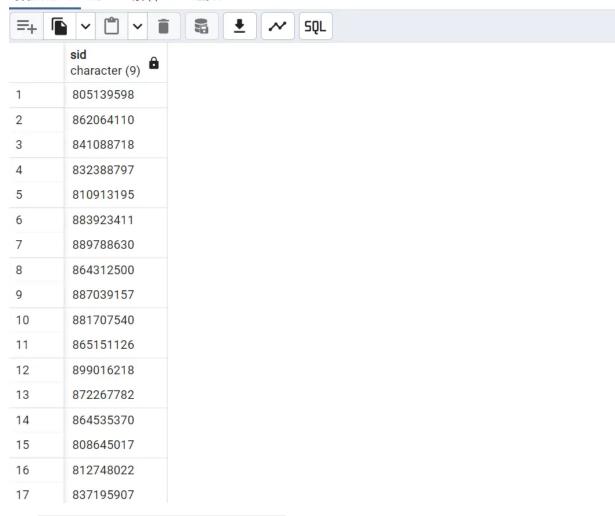
1. 查询全部课程的详细记录;

#### 数据输出 消息 解释 × 通知

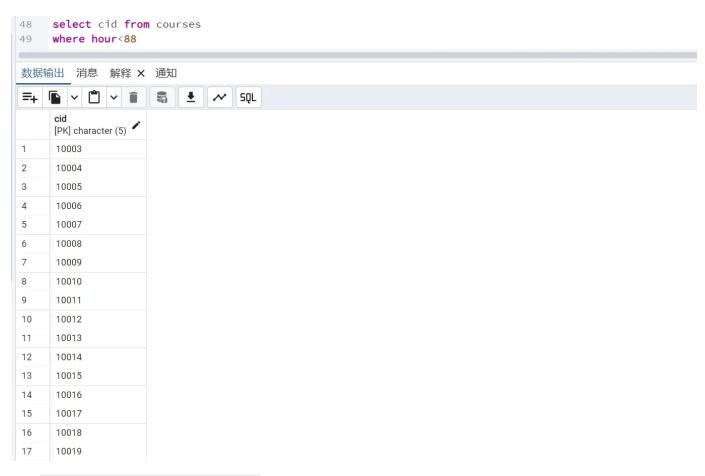


# 2. 查询所有有选修课的学生的编号;

# 数据输出 消息 解释 × 通知



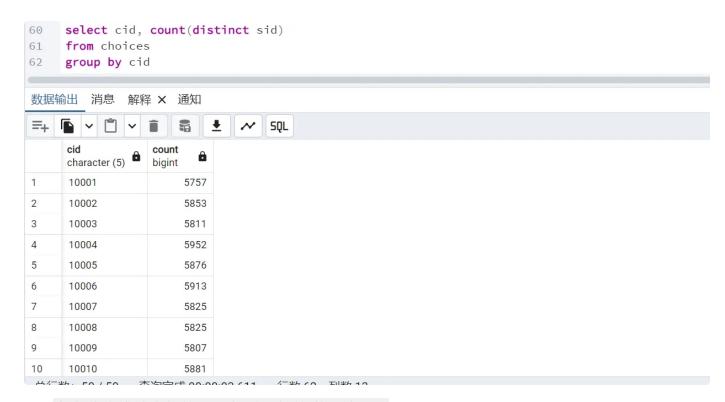
3. 查询课时<88(小时)的课程的编号;



4. 请找出总分超过400分的学生;

51 select sid ,sum(score) from choices group by sid having sum(score)>400 数据输出 消息 解释 🗙 通知 **~** | 5QL =+ **\$ ±** sid sum character (9) â bigint 5. 查询课程的总数; select count(distinct cid) from courses 数据输出 消息 解释× 通知 =+ SQL count bigint

6. 查询所有课程和选修该课程的学生总数;

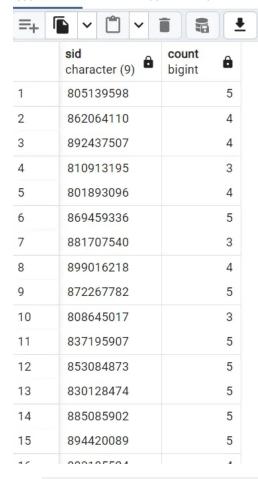


7. 查询选修成绩合格的课程超过两门的学生编号;

- 64 select sid,count(cid)
- 65 from choices
- 66 group by sid
- 67 having count(score>60)>2

SQL

#### 数据输出 消息 解释 × 通知



8. 统计各个学生的选修课程数目和平均成绩。

```
select sid,avg(score),count(cid)
69
      from choices
70
      group by sid
71
72
73
数据输出
          消息
                 解释×通知
                                          SQL
                       avg
                                             count
                                                     â
        character (9)
                       numeric
                                             bigint
1
        837785588
                        72.5000000000000000
                                                      5
2
        838336767
                        77.00000000000000000
                                                      1
3
        890901698
                        68.0000000000000000
                                                      2
4
        805139598
                        78.40000000000000000
                                                      5
5
        813450043
                        64.00000000000000000
                                                      1
6
        862064110
                        86.00000000000000000
                                                      4
7
        862219775
                        74.3333333333333333
                                                      3
        892437507
8
                        59.666666666666667
                                                      4
9
        883923411
                        62.50000000000000000
                                                      2
10
        801893096
                        86.3333333333333333
                                                      4
```

1

5

3

5

4

行数 71, 列数 13

常见错误

11

12 13

14

15

在使用 GROUP BY 子句进行SQL查询时经常发生错误,对实验过程中出现的错误简单总结

1. 未在 SELECT 列表中包含聚合函数:

69.00000000000000000

74.80000000000000000

89.666666666666667

77.20000000000000000

68.00000000000000000

查询完成 00:00:00.376

- 当使用 GROUP BY 时, SELECT 列表中的每个非聚合列必须在 GROUP BY 子句中指 定。
- 错误示例:

816552513

816612565

881707540

872267782

839148293

总行数: 1000 / 100000

- 1 SELECT column1, column2
- 2 FROM table\_name
- 3 GROUP BY column1;
  - 正确示例:

```
1 SELECT column1, AVG(column2) -- 使用聚合函数
2 FROM table_name
3 GROUP BY column1;
```

#### 非常常见的错误!!!

### 2. 在选择列表中使用未在 GROUP BY 中分组的列:

- 任何在 SELECT 列表中出现的列,如果没有在 GROUP BY 子句中使用聚合函数,那么它 也必须出现在 GROUP BY 子句中。
- 错误示例:

```
SELECT column1, column2, SUM(column3)
FROM table_name
GROUP BY column1;
```

#### ○ 正确示例:

```
SELECT column1, SUM(column3)
FROM table_name
GROUP BY column1;
```

#### 或者

```
1 SELECT column1, AVG(column2) -- 使用聚合函数
2 FROM table_name
3 GROUP BY column1;
```

## 3. 在 WHERE 子句中使用聚合函数:

- WHERE 子句不能使用聚合函数,聚合函数应该在 SELECT 列表或 HAVING 子句中使用。
- 错误示例:

```
1  SELECT column1
2  FROM table_name
3  WHERE SUM(column2) > 100;
```

#### ○ 正确示例:

```
1  SELECT column1
2  FROM table_name
3  GROUP BY column1
4  HAVING SUM(column2) > 100;
```

## 4. 在 GROUP BY 子句中使用子查询:

- 子查询不能直接在 GROUP BY 子句中使用,除非子查询返回一个列。
- 错误示例:
- 1 SELECT column1
- 2 FROM (SELECT column1, column2 FROM table\_name) AS subquery
- 3 GROUP BY column1;

### 5. 在 GROUP BY 子句中使用 DISTINCT:

- DISTINCT 关键字通常与 SELECT 子句一起使用,而不是与 GROUP BY 一起使用。
- 错误示例:
- 1 SELECT DISTINCT column1
- 2 FROM table\_name
- 3 GROUP BY column1;