

# 中级查询语句

2025 年 3 月 28 日

## 一、主要任务：

1. 各学院有多少教师数，按学院名称排序
2. 列出由多个教师讲授的课程的课程 ID、课程名的前 5 个字符、以及学期学年等信息。
3. 教师号 ID 为 14365 的老师所在学院的学生总数。
4. 成绩均在 A,A+,A- 以上的学生及其所在学院。
5. 获得全体教师中工资的均值和中位数。
6. 实验证明以下两个 SQL 语句的含义是否相同、为什么？
  - a) `select * from student left outer join takes on student. ID = takes. ID`
  - b) `select * from student left outer join takes on true where student. ID= takes. 1`

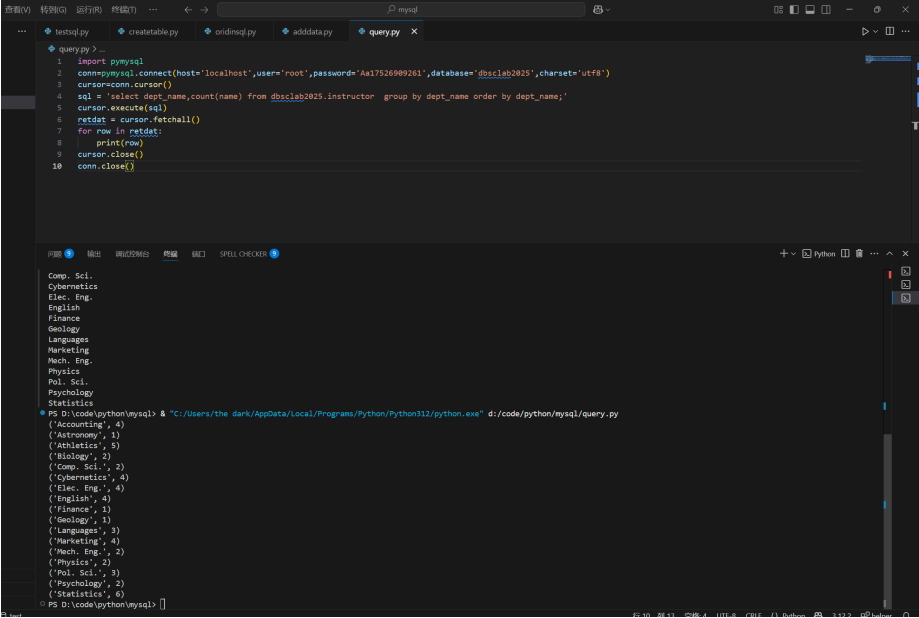
## 二、实现方法：

1. 用 `select count(*)` 语句可以计数，用 `group by` 分组，`order by` 排序。
2. 用 `substr(str, begin, [length])` 语句，可以截断字符串。使用子查询先查询出多个老师讲课的课程 ID，再从 `course` 和 `section` 中找到学期学年等信息。
3. 用子查询先找出当前 ID 老师所在学院，再查找总数即可。
4. 用 `select` 语句查询 `takes` 里的 `id`，然后通过子查询，用这个 `id` 与 `takes` 表做自连接，寻找 `id` 与正在查找的 `id` 相同的课程，并一一检验它们的 `grade` 是不是 A-以上，如果没有不是 A-以上的，则输出
5. 查找平均工资可以用 `avg(salary)`，中位数根据其定义为  $n/2$  个比它大的和  $n/2$  个比它小的，可以用自己笛卡尔积并进行比较计数，最后找到工资排在中间的老师的工资。

6. 尝试执行两个 SQL 语句并获得结论。

三、成果展示：

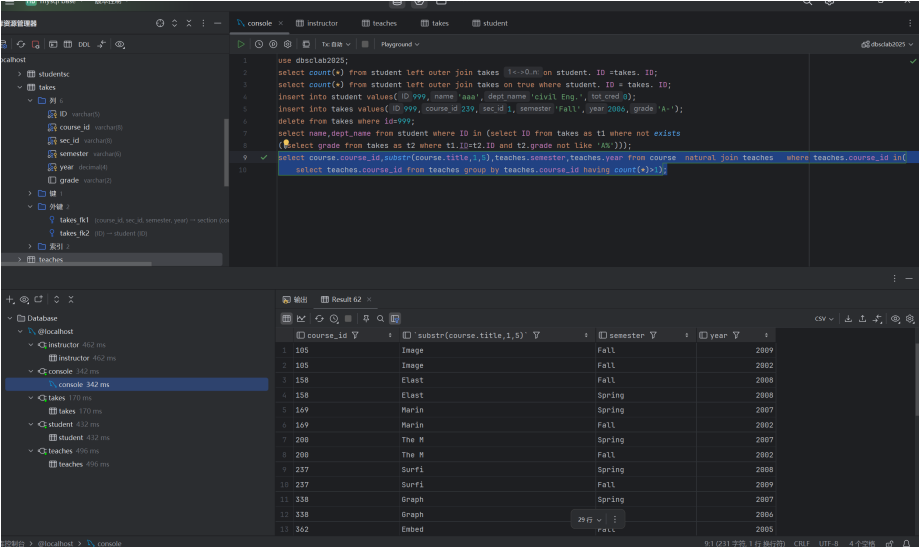
1. 各学院有多少教师数，按学院名称排序



```
1 import pymysql
2 conn=pymysql.connect(host='localhost',user='root',password='Aa17526089261',database='dbclab2025',charset='utf8')
3 cursor=conn.cursor()
4 sql = 'select dept_name,count(name) from dbclab2025.instructor group by dept_name order by dept_name;'
5 cursor.execute(sql)
6 retdata = cursor.fetchall()
7 for row in retdata:
8     print(row)
9 cursor.close()
10 conn.close()
```

Comp. Sci.  
Cybernetics  
Elec. Eng.  
English  
Finance  
Genlog  
Languages  
Marketing  
Mech. Eng.  
Physics  
Pol. Sci.  
Psychology  
Statistics  
(Accounting', 4)  
(Astronomy', 1)  
(Athletics', 5)  
(Biology', 2)  
(Comp. Sci.', 2)  
(Cybernetics', 4)  
(Elec. Eng.', 4)  
(English', 4)  
(Finance', 1)  
(Genlog', 1)  
(Languages', 3)  
(Marketing', 4)  
(Mech. Eng.', 2)  
(Physics', 2)  
(Pol. Sci.', 2)  
(Psychology', 2)  
(Statistics', 6)

2. 列出由多个教师讲授的课程课程 ID、课程名的前 5 个字符、以及学期学年等信息。



```
1 use dbclab2025;
2 select count(*) from student left outer join takes on student_ID = takes_ID;
3 select count(*) from student left outer join takes on true where student_ID = takes_ID;
4 insert into student values(10999,'aaa',dept_name 'civil Eng.',100,cred 0);
5 insert into takes values(10999,course_id 239,sec_id 1, semester 'Fall', year 2006, grade 'A-');
6 delete from takes where lastrow;
7 select name,dept_name from student where ID in (select ID from takes as t1 where not exists
8 (select grade from takes as t2 where t1.ID=t2.ID and t2.grade not like 'A%'));
9 select course_id,substr(course_title,1,5),teaches.semester,teaches.year from course natural join teaches where teaches.course_id in
10 (select teaches.course_id from teaches group by teaches.course_id having count(*)>1);
```

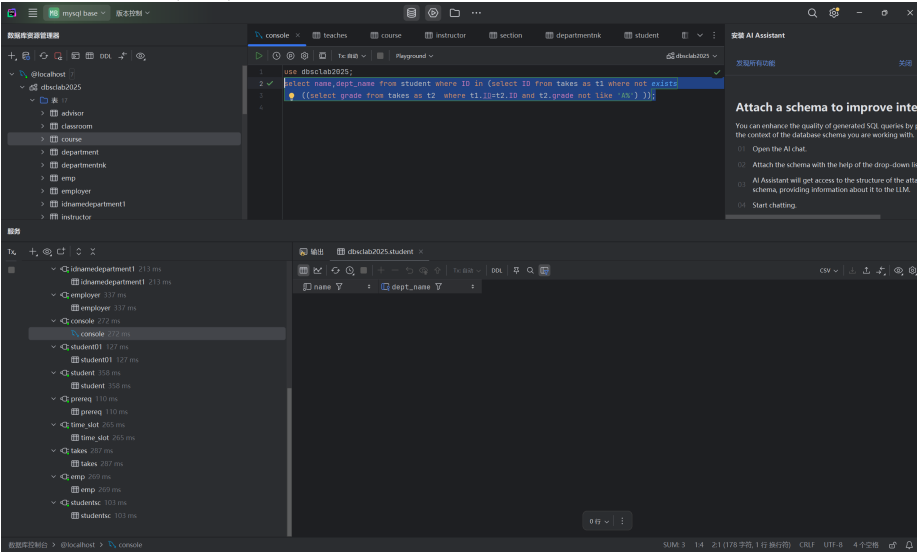
course_id	substr(course_title,1,5)	semester	year
105	Image	Fall	2009
105	Image	Fall	2002
158	Elast	Fall	2008
158	Elast	Spring	2008
169	Marlin	Spring	2007
169	Marlin	Fall	2002
208	The H	Spring	2007
208	The H	Fall	2002
237	Surfi	Spring	2008
237	Surfi	Fall	2009
318	Graph	Spring	2007
318	Graph	Fall	2004
362	Embed	Fall	2005

3. 教师号 ID 为 14365 的老师所在学院的学生总数

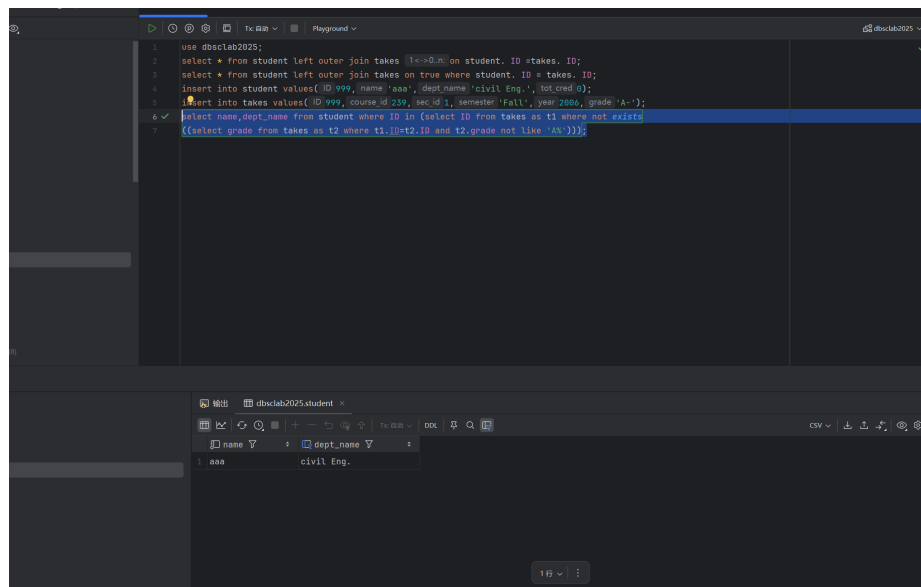
```
query.py > ...
1 import pymysql
2 conn=pymysql.connect(host='localhost',user='root',password='Aa1752698261',database='dbclab2025',charset='utf8')
3 cursor=conn.cursor()
4 sql = 'select dept_name,count(*) from student where dept_name=(select dept_name from instructor where instructor.ID=14365) group by dept_name'
5 cursor.execute(sql)
6 retlist = cursor.fetchall()
7 for row in retlist:
8     print(row)
9 cursor.close()
10 conn.close()

PS D:\code\python\mysql> & "C:/Users/the_dark/AppData/Local/Programs/Python/Python312/python.exe" d:/code/python/mysql/query.py
('Accounting', 99)
PS D:\code\python\mysql>
```

4. 成绩均在 A,A+,A- 以上的学生及其所在学院。

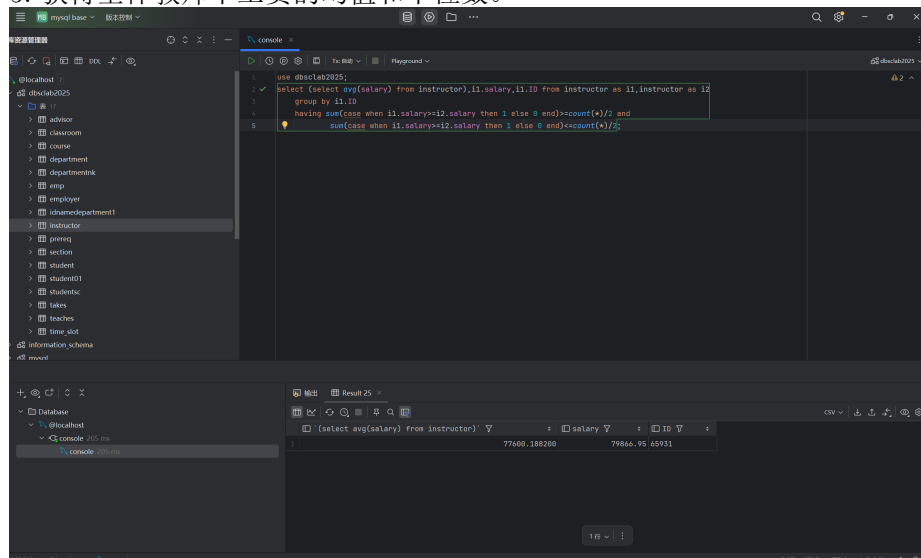


可以看到无人能达到这一成就，于是向其中加一组数据

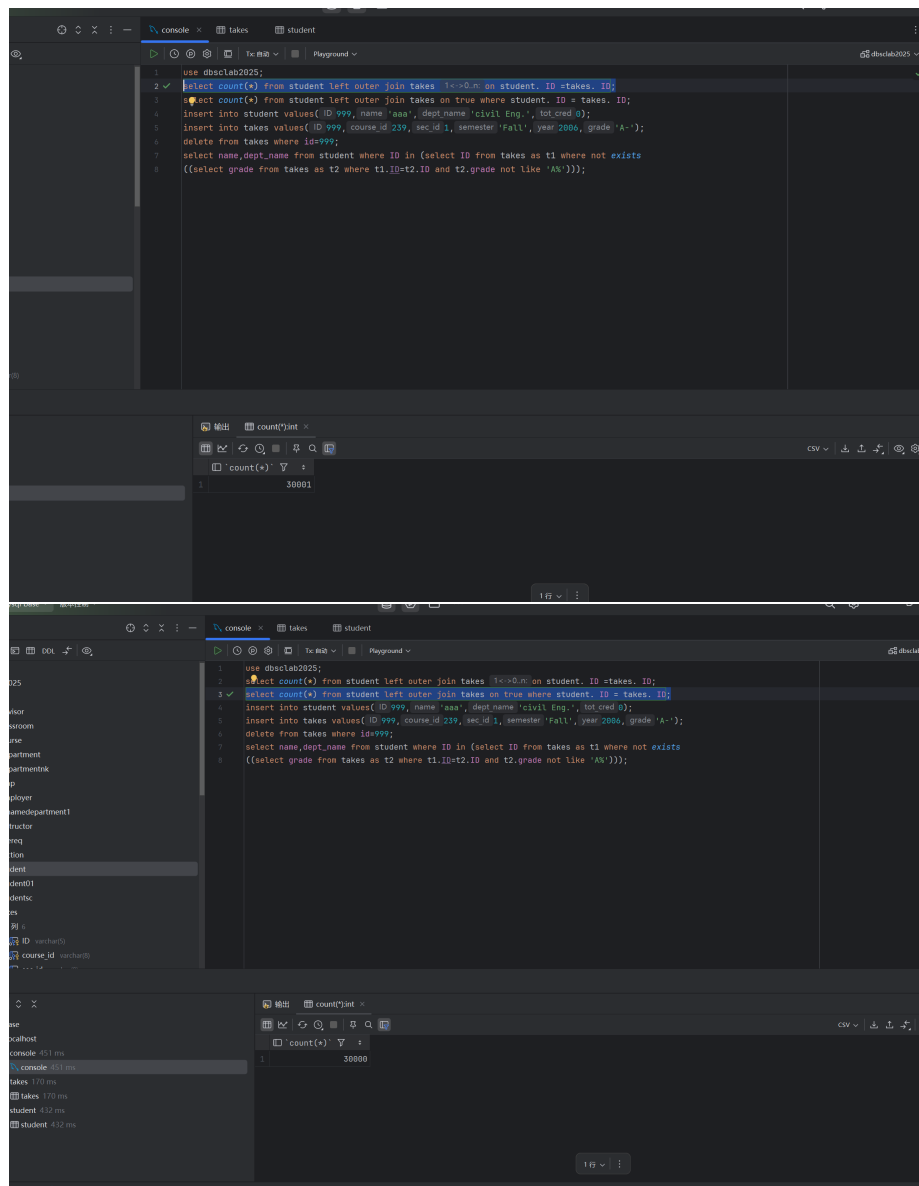


可见可以正确找到

5. 获得全体教师中工资的均值和中位数。



6. 实验证明以下两个 SQL 语句的含义是否相同、为什么？



由于之前添加了一组数据，可以看到两组的统计数不一样，因为第一条语句的 on 是做外连接的条件，无论如何前面的表里所有的数据都会保留，而第二条语句是做外连接后用 where 判断，这样的话如果有学生没有选课的话就不会被统计进去

#### 四、结论：

1. 在多步骤问题中，可以用子查询方法，将问题分步解决

2. 计算数据较多时，可以创建临时表，将大问题分割为小问题解决。
3. 多使用各种语句排列组合寻找可能的解决方案, 要熟练掌握各种基础语句的用法
4. 用 like 进行模糊匹配是很高效的
5. 用自连接的方法可以查询某个重复 id 下面的所有记录