* SQL多表查询  
  Select  
  From  
  Where AND
* SQL聚集查询  
  Select Count/AVG/MAX(\*)  
  From  
  Where\
  1. Count() ：聚集函数，元组个数
  2. AVG()：平均值
  3. MAX()：最大值
* SQL分组聚集  
  Select Groups, AVG(Num)  
  From  
  Group By Groups  
  Having AVG(Num)>90

|  |  |
| --- | --- |
| Groups | AVG(Num) |
| Group1 |  |
| Group2 |  |
| Group2 |  |

1. 分组不聚集的参数非法

**第1题：考虑以下4个查询**

**Q1:   
Select a,b   
From T1, T2   
Where T1.c=T2.c And T1.c=100;   
Q2:   
Select a,b   
From T1, T2   
Where T1.c=100 And T2.c=100;   
Q3:   
Select a,b   
From T1, T2   
Where T1.c=T2.c And T1.c>100;   
Q4:   
Select a,b   
From T1, T2   
Where T1.c>100 And T2.c>100;**说法正确的是：

A：Q1和Q2结果相同；Q3和Q4结果不同； √

B：Q1和Q2结果不同；Q3和Q4结果相同；

C：Q1和Q2结果相同；Q3和Q4结果相同；

D：4个查询结果均不同。

**第*2*题：在学生表Student(s\_no, s\_name, birthday, gender)和学生选课表SC(s\_no, c\_no, grade)中求每个学生的姓名和平均成绩。哪个查询表达正确？**

A：Select s\_name, AVG(grade)

From Student, SC

Where Student.s\_no=SC.s\_no

Group By Student.s\_no

B：Select s\_name, AVG(grade)

From Student, SC

Where Student.s\_no=SC.s\_no

Group By SC.s\_no

C：Select s\_name, AVG(grade)

From Student, SC

Group By SC.s\_no

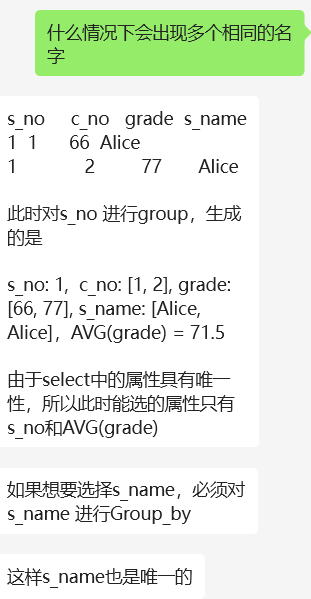
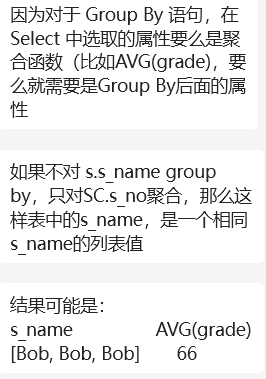
D：Select s\_name, AVG(grade)

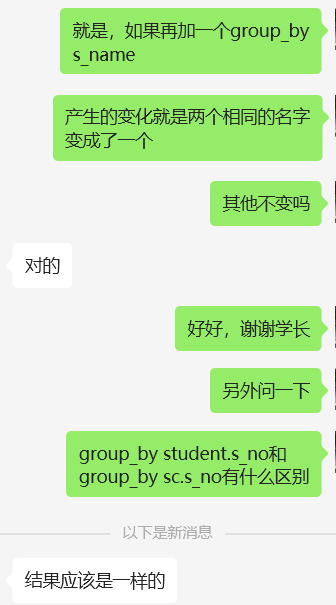
From Student, SC

Where Student.s\_no=SC.s\_no

Group By SC.s\_no, s\_name

√ 如果不同时Group By SC.s\_no, s\_name，可能无法保证唯一性





**第*3*题：以下哪些查询是等价的？**

A：Select Count(\*)

From T

Where b>100

Group By a

和

Select Count(\*)

From T

Group By a

Count(\*) 表示您想要计算满足查询条件的行数。

具体地，它将返回表 "T" 中所有满足后续条件的行的总数。

B：Select Count(\*)

From T

Where b>100

Group By a

和

Select Count(\*)

From T

Group By a

Having AVG(b)>100

C：Select AVG(a)

From T

Where a<10

Group By a

和

Select AVG(a)

From T

Group By a

Having AVG(a)<10 √

D：Select a,AVG(b)

From T

Where b>100

Group By a

这个查询首先会筛选出表 "T" 中满足条件 "b > 100" 的行，然后再按照列 "a" 的值进行分组，最后计算每个分组中列 "b" 的平均值。

这意味着只有那些 "b" 列的值大于 100 的行会被包括在平均值的计算中。

和

Select a,AVG(b)

From T

Group By a

Having AVG(b)>10

这个查询首先会根据列 "a" 的值进行分组，然后计算每个分组中列 "b" 的平均值。接着，使用 HAVING 子句筛选出那些平均值大于 10 的分组。

这意味着在计算平均值之后，只有那些平均值大于 10 的分组会被包括在结果中，但不会限制行的选择。

假设我们有三个关系（同课程中使用的例子一样），其模式分别为

Student(s\_no, s\_name, birthday, gender)

Course(c\_no, c\_name, credit)

SC(s\_no, c\_no, grade)

请写出以下信息需求的SQL查询：

1. 在数学课上成绩超过90分的男生姓名；
2. 数学课成绩超过历史课成绩的女生姓名；
3. 平均成绩超过90分的女生姓名。
4. Select s\_name  
   From Student, Course, SC  
   Where Student.s\_no = SC.s\_no AND SC.c\_no = Course.c\_no AND c\_\_name = “Math” AND grade > 90 AND gender = “male”
5. Math:  
   Select s\_name, SC.s\_no, SC.c\_no, grade   
   From Student, Course, SC  
   Where Student.s\_no = SC.s\_no AND SC.c\_no = Course.c\_no AND c\_name = “Math” And gender = “female”  
     
   History:  
   Select s\_name, SC.s\_no, SC.c\_no, grade  
   From Student, Course, SC  
   Where Student.s\_no = SC.s\_no AND SC.c\_no = Course.c\_no AND c\_name = “History” And gender = “female”  
     
   Select Math.s\_name  
   From Math, History  
   Where Math.s\_no = History.s\_no AND Math.grade > History.grade  
     
   Select s\_name

From Student, SC SC\_1, SC SC\_2, Course C\_1, Course C\_2

Where Student.gender = “female”

AND C\_1.c\_name=“Math” AND C\_2.c\_name=“History”

AND Student.s\_no = SC\_1.s\_no AND Student.s\_no = SC\_2.s\_no

AND SC\_1.c\_no = C\_1.c\_no AND SC\_2.c\_no = C\_2.c\_no

And SC\_1.grade > SC\_2.grade

1. Select s\_name  
   From Student, SC  
   Where Student.s\_no = SC.s\_no And gender=”female”  
   Group By SC.s\_no, s\_name  
   Having AVG(grade) > 90

最好成绩超过平均成绩20分以上的课程名称

Select c\_name

From course,sc

Where c\_no=sc\_no

Having max(grade)-avg(grade)>20

Group by sc.c\_no,c\_name/group by course.c\_no,c\_name

共同选了三门以上课程的所有学生组合（学好的两两组合）

Select

s1.s\_no AS s\_no1,s1.s\_name AS s\_name1,

s2.s\_no AS s\_no2,s2.s\_name AS s\_name2

From student ,students s1,students s2,course, course c1,course c2

Where course

SELECT

s1.s\_no AS s\_no1,

s2.s\_no AS s\_no2,

COUNT(\*) AS common\_course\_count

FROM SC s1

JOIN SC s2

ON s1.s\_no < s2.s\_no AND s1.c\_no = s2.c\_no

GROUP BY s1.s\_no, s2.s\_no

HAVING common\_course\_count >= 3;