



3.3.5 查询注意事项

👉 查询综合举例及注意事项

一、别名的使用

(1) 别名用于对输出属性列的重命名

例 查询全体学生的姓名、出生年份和所在系，并将输出字段依次更名为：NAME、BIRTH、BIRTHDAY、DEPARTMENT。

```
SELECT Sname NAME, 'Year of Birth:' BIRTH, 2007-Sage  
BIRTHDAY, LOWER(Sdept) DEPARTMENT  
FROM Student;
```



3.3.5 查询注意事项

(2) 别名用于自身连接查询和对同一表的相关子查询中，用于区别对同一表的不同引用

例：查询每一门课的间接先修课（即先修课的先修课）。

```
SELECT FIRST.Cno, SECOND.Cpno  
FROM Course FIRST, Course SECOND  
WHERE FIRST.Cpno = SECOND.Cno ;
```

例：查询选修了课程1和选修了课程2的学生学号。

```
SELECT SCX.Sno  
FROM SC SCX  
WHERE SCX.Cno = '1' AND  
      EXISTS ( SELECT *  
              FROM SC SCY  
              WHERE SCX.Sno = SCY.Sno  
                    AND SCY.CNO = '2' );
```



3.3.5 查询注意事项

(3) 对不相关子查询可以不使用别名

例：查询“数据库”课程成绩最高的学生学号。

```
SELECT Sno
FROM SC
WHERE Grade =
    ( SELECT MAX(Grade)
      FROM SC
      WHERE Cno = ( SELECT Cno
                   FROM Course
                   WHERE Cname = '数据库' )
    )
AND Cno = ( SELECT Cno
            FROM Course
            WHERE Cname = '数据库' );
```



3.3.5 查询注意事项

二、DISTINCT的使用

DISTINCT用于区分相同的**记录**，将多条相同的记录作为一条处理。

例：查询成绩不及格的学生学号。

```
SELECT DISTINCT Sno
FROM SC
WHERE Grade < 60 ;
```

例：查询有成绩不及格的学生人数。

```
SELECT COUNT ( DISTINCT Sno)
FROM SC
WHERE Grade < 60 ;
```

注：如果不用DISTINCT加以限定，则同一个人有多门课不及格按多次计算，不符合题意。



3.3.5 查询注意事项

三、集函数的使用

集函数只能用于 SELECT子句和 HAVING短语之中，而绝对不能出现在 WHERE子句中。

例：查询年龄最大的学生。

```
SELECT *  
FROM Student  
WHERE Sage = MAX (Sage) ;
```



正确的查询语句：

```
SELECT *  
FROM Student  
WHERE Sage = ( SELECT MAX (Sage)  
                FROM Student );
```



3.3.5 查询注意事项

例：查询计算机系年龄最大的学生。

```
SELECT *
FROM Student
WHERE Sage = ( SELECT MAX (Sage)
                FROM Student
                WHERE Sdept = 'CS' )
AND Sdept = 'CS' ;
```

例：查询平均成绩最高的学生学号。

```
SELECT Sno
FROM SC
GROUP BY Sno
HAVING AVG(Grade) >= ALL ( SELECT AVG(Grade)
                            FROM SC
                            GROUP BY Sno) ;
```

注：集函数没有复合功能，即不能写成 MAX(AVG(Grade)) 。



3.3.5 查询注意事项

四、GROUP BY的使用

使用了分组的查询语句，其SELECT子句中只能出现分组属性和集函数，而不能有在GROUP BY没有出现的属性。

原因在于SELECT子句是对分组后的结果集进行输出，参与分组的属性在同一组中取相同值，而不参与分组的属性(不在GROUP BY子句中出现的属性)则可能取不同的值，分组输出通常是对组的整体描述，而不是组内的详细记录，不参与分组的属性取值不同，故无法输出。

例：查询各系的学生人数。

```
SELECT Sdept, Ssex, COUNT(*)
```

```
FROM Student
```

```
GROUP BY Sdept ;
```

原因在于组内的性别值不同，输出值无法确定。



3.3.5 查询注意事项

GROUP BY子句在复合查询中的应用

GROUP BY子句用来对查询结果进行分组，通常用作对各组的统计，可用于子查询。

例：查询平均成绩“优秀”(≥ 90 分)的学生学号和姓名，并按学号升序输出。

```
SELECT Sno, Sname
FROM Student
WHERE Sno IN ( SELECT Sno
                FROM SC
                GROUP BY Sno
                HAVING AVG(Grade)  $\geq$  90 )
ORDER BY Sno ;
```




3.3.5 查询注意事项

五、ORDER BY子句在复合查询中的应用

ORDER BY子句用于对查询结果进行排序后再输出，故只用于最外层的查询，而子查询中不应该出现ORDER BY子句。

例：查询学习“数据库”课程的学生学号和成绩，并按成绩由高到低进行排序。

```
SELECT Sno, Grade
FROM SC
WHERE Cno= ( SELECT Cno
              FROM Course
              WHERE Cname = '数据库' )
ORDER BY Grade DESC ;
```



3.3.5 查询注意事项

例：查询计算机系年龄不大于19岁的学生信息，并按学号升序输出。

```
SELECT *  
FROM Student  
WHERE Sdept = 'CS'  
ORDER BY Sno  
INTERSECT  
SELECT *  
FROM Student  
WHERE Sage <= 19  
ORDER BY Sno ;
```



正确写法：

```
SELECT *  
FROM Student  
WHERE Sdept = 'CS'  
INTERSECT  
SELECT *  
FROM Student  
WHERE Sage <= 19  
ORDER BY 1 ;
```



3.3.5 查询注意事项

六、输出多个表的属性的查询

查询的输出只能取自最外层查询所使用的表，对于子查询中的属性是不能作为最终的输出的。如果输出的属性涉及多个表，则最外层查询只能使用这些表的连接查询。

例：查询成绩超过80分的学生的学号、姓名、课程号和成绩。

```
SELECT DISTINCT Sno, Sname, Cno, Grade
FROM Student
WHERE Sno IN ( SELECT Sno
                FROM SC
                WHERE Grade >= 80 );
```



正确的查询语句：

```
SELECT DISTINCT SC.Sno, Sname, Cno, Grade
FROM Student, SC
WHERE Student.Sno = SC.Sno AND Grade >= 80 ;
```



3.3.5 查询注意事项

七、导出关系作为数据源的使用

子查询的结果集可以看作基本关系一样作为查询的数据源，需要对结果集命名别名以便引用。

例：查询全体学生的学号、姓名和平均成绩。

```
SELECT Student.Sno, Sname, avg_grade
FROM Student, ( SELECT Sno, AVG(Grade) AS avg_grade
                FROM SC
                GROUP BY Sno ) AS SC_AVG
WHERE Student.Sno = SC_AVG.Sno ;
```

实际上平均成绩的计算应该使用如下子查询：

```
SELECT Sno, SUM(Grade*Ccredit)/SUM(Ccredit) AS avg_grade
FROM SC, Course
WHERE SC.Cno=Course.Cno
GROUP BY Sno
```



3.3.5 查询注意事项

[例46] 查询至少选修学生95002选修的全部课程的学生号码。

不用EXISTS谓词实现的思路：

只用关心95002选修过的课程，其他课程可以忽略不及，结果可以用子查询实现。换言之，查询即是要求：在此子查询结果集中，学生的选课数量等于95002的选课数量。

```
SELECT Sno FROM
( SELECT X.Sno AS Sno,Y.Cno AS Cno
  FROM SC X, SC Y
  WHERE X.Cno=Y.Cno AND Y.Sno='95002') AS M
GROUP BY Sno
HAVING COUNT(*)=(
                SELECT COUNT(*) FROM SC
                WHERE Sno='95002');
```