

# SQL 经典题型解答 (3)

数据库

11、查询没有学全所有课程的同学的信息

```
SELECT
    b.*
FROM
    student b
WHERE
    b.S IN (
        SELECT
            a.s
        FROM
            ( SELECT a.S s, COUNT( a.C ) NUM FROM sc a GROUP BY a.s ) a
        WHERE
            a.NUM NOT IN ( SELECT COUNT( Cname ) FROM course )
    )
```

详解：

首先在表 sc 中查找出每个学生学习的课程数，然后查找课程数不够总课程数的学生编号，最后在表 student 中查出学生信息。

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	Sage	Ssex
05	周梅	1990-12-01	女
06	吴兰	1992-03-01	女
07	郑竹	1989-07-01	女

12、查询至少有一门课与学号为"01"的同学所学相同的同学的信息

```
SELECT
    *
FROM
```

```

student
WHERE
    S IN ( SELECT DISTINCT ( s ) FROM sc WHERE s != 01 AND C IN ( SELECT
c FROM sc WHERE s = 01 ) )

```

详解：

首先从表 **sc** 中找到“01”同学的课程编号，然后再从表 **sc** 中找到学生编号不是”01“ 且课程编号在 “01“同学课程编号中的学生编号，最后从表 **student** 中找到学生信息。

**SELECT DISTINCT** 用于返回唯一不同的值，在这里也可以用 **GROUP BY s** 选择。  
**SELECT DISTINCT** 用法

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	Sage	Ssex
▶ 02	钱电	1990-12-21	男
03	孙风	1990-05-20	男
04	李云	1990-08-06	男
05	周梅	1990-12-01	女
06	吴兰	1992-03-01	女
07	郑竹	1989-07-01	女

13、查询和”01”号的同学学习的课程完全相同的其他同学的信息

```

SELECT
    *
FROM
    student
WHERE
    s IN (
        SELECT
            s
        FROM
            sc
        WHERE
            s != 01
            AND C IN ( SELECT c FROM sc WHERE s = 01 )
        GROUP BY
            s
        HAVING
            COUNT( C ) = ( SELECT COUNT( c ) FROM sc WHERE S = '01' )
    )

```

详解：

上一题中我们可以找到至少有一门课与学号为"01"的同学所学相同的同学的学生编号,再加一个两个同学的学习课程数相同的条件,即可证明这两个同学的学习课程相同。

程序中运用的 **HAVING** 函数,所以不能用 **DISTINCT** 函数, **HAVING** 的用法与 **WHERE** 用法类似,不过 **WHERE** 关键字无法与聚合函数一起使用, **HAVING** 子句可以让我们筛选分组后的各组数据。

**HAVING** 通常和 **GROUP BY** 连用,

**WHERE** 子句的作用是在对查询结果进行分组前,将不符合**where**条件的行去掉,即**在分组之前过滤数据**,条件中不能包含聚合函数,使用 **WHERE** 条件显示特定的行。

**HAVING** 子句的作用是筛选满足条件的组,即**在分组之后过滤数据**,条件中经常包含聚合函数,使用 **HAVING** 条件显示特定的组,也可以使用多个分组标准进行分组。

具体用法可以参考：

- [SQL中GROUP BY语句与HAVING语句的使用](#)
- [where、having之间的区别和用法](#)
- [SQL HAVING 子句](#)
- [SQL WHERE 子句](#)

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	Sage	Ssex
02	钱电	1990-12-21	男
03	孙风	1990-05-20	男
04	李云	1990-08-06	男

14、查询没学过"张三"老师讲授的任一门课程的学生姓名

```
SELECT
    *
FROM
    student
WHERE
    S NOT IN (
        SELECT
            S
        FROM
            SC
        WHERE
```

```
C = ( SELECT C FROM course WHERE T = ( SELECT T FROM teacher WHERE Tname = '张三' ) )
)
```

详解：

首先在表 `teacher` 中找到“张三”老师的编号，然后在表 `course` 中找到其课程编号，再在表 `sc` 中找到学习该课程的学生编号，最后在表 `student` 中找出不在这些编号的学生信息。

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	Sage	Ssex
06	吴兰	1992-03-01	女
08	王菊	1990-01-20	女

15、查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

```
SELECT
    a.S,
    a.Sname,
    b.avgscore
FROM
    student AS a,
    (
        SELECT
            AVG( score ) avgscore,
            s
        FROM
            sc
        WHERE
            sc.score < 60 GROUP BY s HAVING COUNT( s ) >= 2
    ) AS b
WHERE
    a.S = b.S
```

详解：

首先通过表 `sc` 找到两门及其以上不及格课程的同学的学号和平均成绩，然后通过表 `student` 找到学生姓名。

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	avgscore	
04	李云	3.3333333333333336	
06	吴兰	32.5	

16、检索“01”课程分数小于60，按分数降序排列的学生信息

```
SELECT
    a.*
FROM
    student a
    INNER JOIN ( SELECT S, score FROM sc WHERE c = '01' AND score < 60 )
    b ON a.s = b.s
ORDER BY
    b.score DESC
```

详解：

首先找到“01”课程分数小于 60 的学生的编号和成绩，然后与 student 表连接后通过 score 降序排列

[SQL ORDER BY 关键字用法](#)

不能再表 b 内排序后与表 student 连接，b 表内的排序并不影响最终结果。

程序运行结果：

信息	结果 1	剖析	状态
S	Sname	Sage	Ssex
04	李云	1990-08-06	男
06	吴兰	1992-03-01	女

17、按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

```
SELECT
    a.s,
    a.Sname,
    b.c,
```

```
        b.score,  
        c.avgscore  
FROM  
    student a  
LEFT JOIN  
    sc b  
    ON a.s = b.s  
LEFT JOIN  
    ( SELECT s, AVG( score ) AS avgscore FROM sc GROUP BY s ) c  
    ON  
        a.s = c.s  
ORDER BY  
    c.avgscore DESC
```

详解：

这道题题目有点看不懂，我的理解是先求出每个同学的平均成绩，然后根据平均成绩进行排列，本来是 `WHERE` 语句进行的，结果发现竟然还有一个 8 号学生没有在表 `sc` 中，所以采用了 `LEFT JOIN` 语句。

首先求出每个学生的平均出成绩，然后与表 `sc` 和 表 `student` 连接，通过 `ORDER BY` 语句排列，用 `LEFT JOIN` 语句可以包含所有的学生信息。

以前的题都没有考虑这一因素。。。。。

程序运行结果：

信息	结果 1	剖析	状态		
s	Sname	c	score	avgscore	
▶ 07	郑竹	02	89	93.5	
07	郑竹	03	98	93.5	
01	赵雷	01	80	89.66666666666667	
01	赵雷	02	90	89.66666666666667	
01	赵雷	03	99	89.66666666666667	
05	周梅	02	87	81.5	
05	周梅	01	76	81.5	
03	孙风	01	80	80	
03	孙风	02	80	80	
03	孙风	03	80	80	
02	钱电	03	80	70	
02	钱电	01	70	70	
02	钱电	02	60	70	
04	李云	01	50	33.333333333333336	
04	李云	02	30	33.333333333333336	
04	李云	03	20	33.333333333333336	
06	吴兰	01	31	32.5	
06	吴兰	03	34	32.5	
08	王菊	(Null)	(Null)	(Null)	

部分答案参考自：[SQL 经典五十道题](#)