**第7章\_子查询**

**1. 什么是子查询**

①子查询的定义

简单的来说，就是嵌套在查询语句里面的查询就是子查询。（换句话说，就是嵌套在查询语句里面的查询语句。）

select 字段名，（子查询） from 表名；

select \* from （子查询）；

select 字段名，字段名 from 表名 where （子查询）；

②子查询嵌套位置

首先我们来看一个完整的SQL结构：

select 字段名1，字段名2.....

from 表名

where 过滤条件

group by 字段名

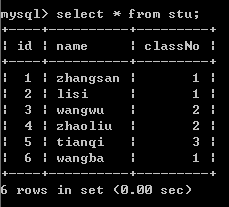
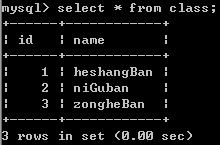
having 过滤条件

order by 字段名

原则上来讲，我们的子查询可以嵌套在上述结构的任何一个位置。但是，用得最多的就是前面三个。（select，where，from），特别是嵌套在where后面，用得尤其的多。

首先有如下两张表，表数据如下：

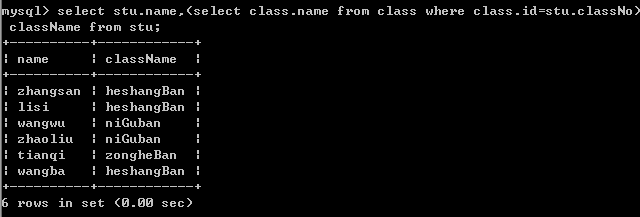
学生表 班级表

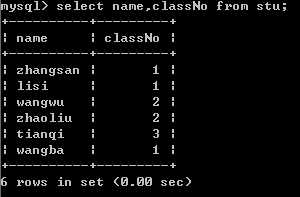
③ 在select中嵌套子查询

查询学生的姓名和所在班级的名称

select stu.name,(select class.name from class where class.id=stu.classNo) className from stu;



以前查询的时候，是单表查询，所以只能查询表里面所拥有的字段



④ 在from中嵌套子查询

from后面跟的是一个表名，所以嵌套的子查询所查询出来的结果被当做一张新表

例如

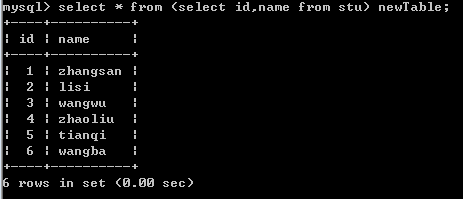
select id,name from stu; //从学生表查询出id和name字段



然后，我们把这个结果当做一张新表

select \* from 表名

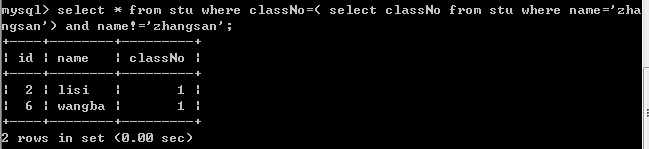
select \* from (select id,name from stu) 别名;（这个地方必须要有别名，否则会报错）



⑤ 在where中嵌套子查询（这是最最最常用的）

例如：查询和zhangsan（张三）在同一个班上的所有同学

select \* from stu where classNo=( select classNo from stu where name='zhangsan') and name!='zhangsan';

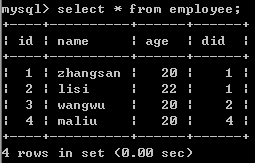
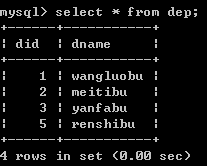


**2. 带关键字的子查询**

在我们的子查询中，常使用的关键字有IN，EXISTS，ANY，ALL这几个关键字。

现在有两张表

员工表 部门表

① IN关键字

例如：查询有年龄为20岁的员工的部门编号和名称

分析：第一步，应该是查询年龄为20岁的员工的部门编号

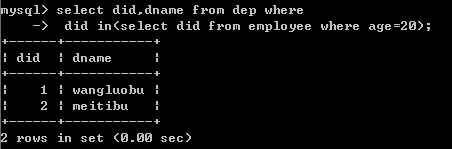
select did from employee where age=20；

查询出来结果为1,2,4

第二步：查询上述部门编号的部门名称

select did,dname from dep where

did in(select did from employee where age=20);

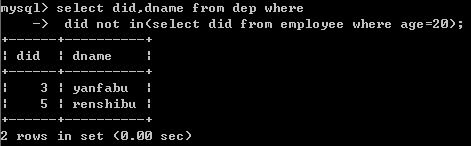


当然，IN关键字是可以和not关键字一起使用的，变为not in

例如：查询没有20岁的员工的部门编号和部门名称

select did,dname from dep where

did not in(select did from employee where age=20);



② EXISTS关键字（存在的意思）

EXISTS这个关键字后面的参数可以是任意一个子查询，这个子查询相当于是一个测试，返回的是一个真假（布尔值），不会产生任何数据。只返回true或者false。当返回值是true的时候，外部查询才会执行，返回值为false，外部查询不执行了。

例如：查询员工中是否存在年龄大于21岁的员工，如果存在，查询部门表中的所有记录

分析：第一步：查询是否有大于21岁的员工

select did from employee where age>21

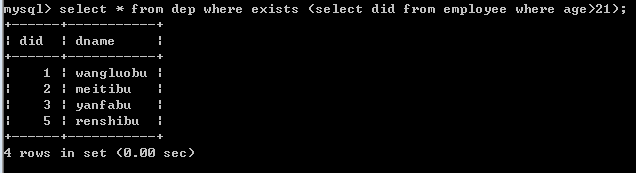
第二步：查询部门所有记录

select \* from dep;

第三部：使用关键字exists將上面两个SQL语句合并起来

select \* from dep where exists (select did from employee where age>21);

返回true的情况



返回false的情况



③any关键字

any关键字表示满足其中任意一个条件，就返回一个结果作为外层查询条件。

例如：

select \* from dep where did>any(select did from employee);

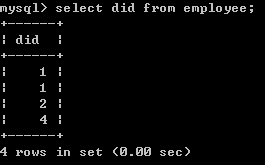
得到以下结果：



分析上面结果如何得来的

①select did from employee;

执行此语句，将会得到员工表里面所有的部门编号（1,1,2,4）



② select \* from dep where 过滤条件（大于1,1,2,4中的任意一个）

④ all关键字

all关键字表示满足全部条件，就返回一个结果作为外层查询条件。

select \* from dep where did>all(select did from employee);



**3. 子查询的分类**

我们子查询大致可以分为2类：一个是相关子查询，二是非相关子查询（又可以称之为：独立查询）

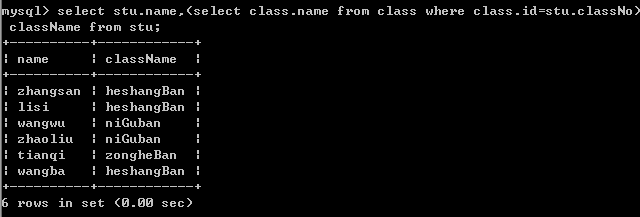
① 相关子查询

所谓相关子查询，就是指内部查询需要依赖外查询

举例：

查询学生的姓名和所在班级的名称

select stu.name,(select class.name from class where class.id=stu.classNo) className from stu;



这里如果你单独执行[select class.name from class where class.id=stu.classNo],会报错，因为找不到stu表，所以这条内部查询语句是需要依赖外部查询的

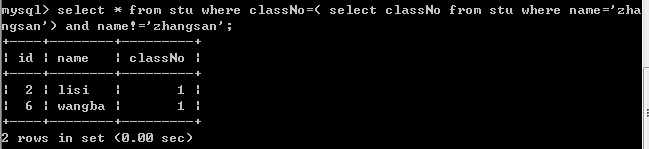
②非相关子查询

所谓非相关子查询，就是指内部查询不依赖外部查询

举例：

查询和zhangsan（张三）在同一个班上的所有同学

select \* from stu where classNo=( select classNo from stu where name='zhangsan') and name!='zhangsan';



这里如果我们单独执行[select classNo from stu where name='zhangsan']，是可以查询出来的，没毛病。所以由此我们可以得知，这条内部查询语句不依赖外部查询语句

**非相关子查询与相关子查询的效率比较：**

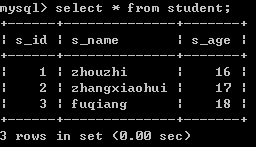
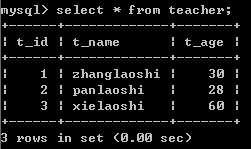
非相关子查询的效率要远远高于相关子查询

4. 集合操作UNION

关键字union的作用就是將两张表纵向的连接起来。但是有一个注意点：就是两张表的字段数必须相同。

现在有如下两张表，表信息如下：

学生表 教师表

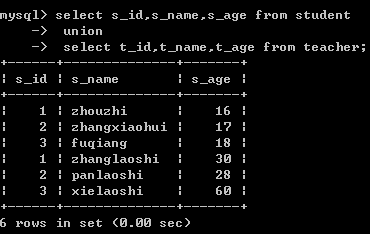
例如：查询所有老师和学生的id，name和age

select s\_id,s\_name,s\_age from student

union

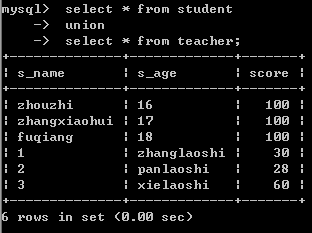
select t\_id,t\_name,t\_age from teacher;

结果如下：



字段名取的是第一个表的字段名

数据是没有关系（只关心字段数，不关心数据类型）



可以多表进行纵向连接

