# 子查询

## 概述

当一个查询是另一个查询的条件时，称之为子查询。

子查询和连接一样，提供了使用单个查询访问多个表中的数据的方法。子查询可以使用在SELECT、INSERT、UPDATE和DELETE语句中，使它们能够利用子查询返回的结果。

需要注意的是，子查询必须包含括号。

## 特点

通常来讲，使用子查询的好处如下：

子查询允许结构化的查询，这样就可以把一个查询语句的每个部分隔开。

子查询提供了另一种方法来执行有些需要复杂的JOIN和UNION来实现的操作。

在许多人看来，子查询可读性较高。而实际上，这也是子查询的由来。

## 分类

### 单值子查询

### 多值子查询

一个子查询除了可以产生一个单一值外，也可以产生一个关系，该关系可以包含若干元组。SQL提供了若干对于关系的操作符，并产生一个布尔型的结果，这些操作符主要用在子查询的结果关系上，它主要包括：IN、EXISTS、SOME（ANY）、ALL、UNIQUE等。

### 独立子查询

### 相关子查询

### 嵌套子查询

大多数情况下，子查询出现在查询的HAVING子句或WHERE子句中。子查询自身也可以包含WHERE子句和或HAVING子句，同样，子查询也可以出现在其他子查询中。位于其他子查询内的子查询被称为嵌套的子查询。

### 树查询

## 使用场景

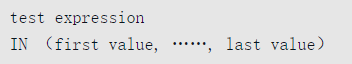
使用子查询可以避免由于子查询中的数据产生的重复

使用子查询更符合语意，更好理解

# 原理

## IN子查询

IN运算符允许根据一行记录中是否有一列包括在一系列值之中而选择该行。语法如下。

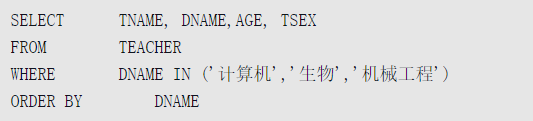


所有的条件在IN运算符后面罗列，并以括号“()”包括起来，条件中间用逗号分开。当testexpression处于括号中列出的一系列值之中时，IN运算符求值为True。

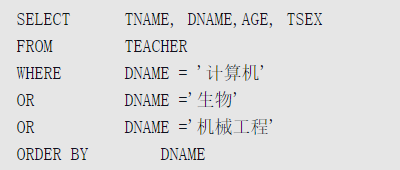
与IN运算符相对的还有NOT IN运算符，其判别结果刚好与IN运算符相反，即当test expression不处于括号中列出的一系列值之中时，NOT IN运算符求值为True。

**在大多数情况下，OR运算符与IN运算符可以实现相同的功能**。

例如：



等价于：



我们可以发现二者结果完全相同。那么为什么要提出IN运算符呢？这是因为与OR运算符相比，IN运算符具有以下两个优点。

• 当选择的条件很多时，采用IN运算符就显得很简便，只需在括号内用逗号间隔依次罗列即可，运行效率比OR运算符要高。

• 使用IN运算符，其后面所列的条件可以是另一条SELECT语句，即子查询。

## EXIST子查询

**IN和EXIST区别：**

mysql中的in语句是把外表和内表作hash 连接，而exists语句是对外表作loop循环，每次loop循环再对内表进行查询。一直大家都认为exists比in语句的效率要高，这种说法其实是不准确的。这个是要区分环境的。

1、如果查询的两个表大小相当，那么用in和exists差别不大。

2、如果两个表中一个较小，一个是大表，则子查询表大的用exists，子查询表小的用in。

3、not in 和not exists如果查询语句使用了not in 那么内外表都进行全表扫描，没有用到索引；而not extsts的子查询依然能用到表上的索引。所以无论那个表大，用not exists都比not in要快。

## NOT运算符

NOT运算符用于对搜索条件的布尔值求反。本节将讨论NOT运算符的使用、应用场合及其与<>运算符的区别。

注：**NULL值进行取反，结果仍是NULL**。

对于简单的条件查询，NOT运算符与<>运算符的功能几乎没有什么区别，那么NOT运算符的优势体现在哪里呢？答案是它可以与其他运算符组合使用，这一点是<>运算符所不能实现的。

注：MySQL数据库系统不支持NOT运算符。在MySQL数据库系统中，NOT运算符只能被用在EXISTS运算符的前面，形如NOT EXISTS。

## SOME/ANY/ALL子查询

# 应用

## 多例过滤

子查询匹配两个值，机MySQL独有的多例过滤方式：

SELECT a.user\_name,b.timestr,kills

FROM user1 a

JOIN user\_kills b ON a.id = b.user\_id

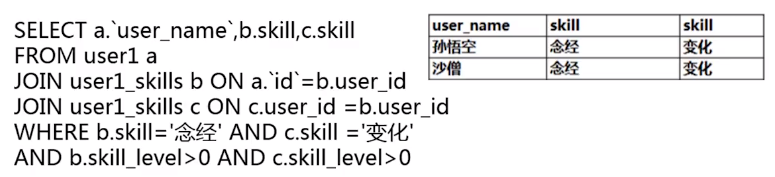
WHERE (b.user\_id,b.kills) IN (

SELECT user\_id,MAX(kills) FROM user\_kills GROUP BY user\_id

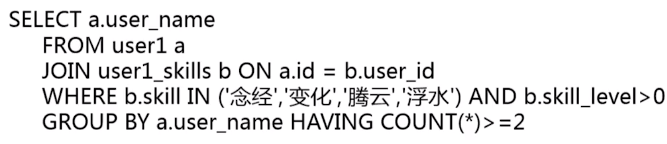
);

## 同属性多值过滤

### JOIN方式实现



### GROUP BY方式实现



## 多属性查询