

SQL 必知必会 5 - 高级数据过滤

SQL 必知必会 5 - 高级数据过滤

5.1 组合 WHERE 子句

5.1.1 AND 操作符

5.1.2 OR 操作符

5.1.3 求值顺序

5.2 IN 操作符

5.3 NOT 操作符

5.1 组合 WHERE 子句

SQL 允许使用多个 WHERE 子句，有两种方式：

- AND 子句
- OR 子句

操作符 operator：用来联结或改变 WHERE 子句中的子句的关键字，也称为逻辑操作符 **logical operator**。

5.1.1 AND 操作符

要通过不止一个列进行过滤，要使用 AND 操作符给 WHERE 子句附加条件。

```
SELECT prod_id, prod_price, prod_name
FROM Products
WHERE vend_id = 'DLL01' AND prod_price <= 4
ORDER BY prod_price DESC, prod_name;
```

AND

用在 WHERE 子句中的关键字，用来指示检索满足所有给定条件的行。

可以增加多个过滤条件，每个条件之间都要使用 AND 关键字。

ORDER BY 子句要放在 WHERE 子句之后。

5.1.2 OR 操作符

OR 操作符指示 DBMS 查询匹配任一条件的行。

许多 DBMS 有短路逻辑，第一个条件满足的时候，不管第二个条件是否满足，相应的行都会被查询出来。

```
SELECT prod_name, prod_price
FROM Products
WHERE vend_id = 'DLL01' OR vend_id = 'BRS01';
```

5.1.3 求值顺序

像多数语言一样，SQL 先处理 AND 后处理 OR。

假如要查询由供应商 DLL01 或 BRS01 制造的，且价格在 10 美元以上的所有产品。那么不能这么写：

```
SELECT prod_name, prod_price
FROM Products
WHERE vend_id = 'DLL01' OR vend_id = 'BRS01'
      AND prod_price >= 10;
```

这样先处理 AND 后处理 OR，查出来的是：

- 供应商 BRS01 的价格大于 10 美元的产品
- 以及供应商 DLL01 的所有产品

解决方法：使用圆括号对操作符进行明确分组。

```
SELECT prod_name, prod_price
FROM Products
WHERE (vend_id = 'DL01' OR vend_id = 'BRS01')
      AND prod_price >= 10;
```

不要过分依赖默认求值顺序，使用圆括号没有什么坏处，它能消除歧义。

5.2 IN 操作符

IN 操作符用来指定条件范围，范围中的每个条件都可以进行匹配。

IN 取一组由逗号分隔、括在圆括号中的合法值。

```
SELECT prod_name, prod_price
FROM Products
WHERE vend_id IN ('DLL01', 'BRS01')
ORDER BY prod_name;
```

IN 操作符完成了与 **OR** 相同的功能。

```
SELECT prod_name, prod_price
FROM Products
WHERE vend_id = 'DLL01' OR vend_id = 'BRS01'
ORDER BY prod_name;
```

为什么选 **IN**，优点如下：

- 有很多选项时，语法更清楚**直观**。
- 与 **AND** 和 **OR** 操作符组合使用的时候，**求值顺序更容易管理**。
- **IN** 操作符一般比一组 **OR** 操作符执行的**更快**
- **IN** 最大的有点是**可以包含其它 **SELECT** 子句**，能够更动态地建立 **WHERE** 子句。

5.3 NOT 操作符

NOT 否定其后所跟的任何条件。

除了 **DLL01** 之外的所有供应商制造的产品。

```
SELECT prod_name
FROM Products
WHERE NOT vend_id = 'DLL01'
ORDER BY prod_name;
```

也可以使用 **<>** 操作符来完成。

```
SELECT prod_name
FROM Products
WHERE vend_id <> 'DLL01'
ORDER BY prod_name;
```

在复杂的子句中，`NOT` 更有优势。比如和 `IN` 搭配使用。

`MariaDB` 允许 `NOT` 否定 `IN`、`BETWEEN` 和 `EXISTS` 子句。
大多数 `DBMS` 允许使用 `NOT` 否定任何条件。