

MySQL 正则表达式

在前面的章节我们已经了解到MySQL可以通过 **LIKE ...%** 来进行模糊匹配。

MySQL 同样也支持其他正则表达式的匹配，MySQL中使用 REGEXP 操作符来进行正则表达式匹配。

如果您了解PHP或Perl，那么操作起来就非常简单，因为MySQL的正则表达式匹配与这些脚本的类似。

下表中的正则模式可应用于 REGEXP 操作符中。

| 模式 | 描述 |
|----------|---|
| ^ | 匹配输入字符串的开始位置。如果设置了 RegExp 对象的 Multiline 属性，^ 也匹配 '\n' 或 '\r' 之后的位置。 |
| \$ | 匹配输入字符串的结束位置。如果设置了RegExp 对象的 Multiline 属性，\$ 也匹配 '\n' 或 '\r' 之前的位置。 |
| . | 匹配除 "\n" 之外的任何单个字符。要匹配包括 '\n' 在内的任何字符，请使用象 '[\n]' 的模式。 |
| [...] | 字符集合。匹配所包含的任意一个字符。例如， '[abc]' 可以匹配 "plain" 中的 'a'。 |
| [^...] | 负值字符集合。匹配未包含的任意字符。例如， '[^abc]' 可以匹配 "plain" 中的 'p'。 |
| p1 p2 p3 | 匹配 p1 或 p2 或 p3。例如， 'z food' 能匹配 "z" 或 "food"。'(z f)ood' 则匹配 "zood" 或 "food"。 |
| * | 匹配前面的子表达式零次或多次。例如， zo* 能匹配 "z" 以及 "zoo"。* 等价于 {0,}。 |
| + | 匹配前面的子表达式一次或多次。例如， 'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。 |
| {n} | n 是一个非负整数。匹配确定的 n 次。例如， 'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。 |
| {n,m} | m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。 |

实例

了解以上的正则需求后，我们就可以根据自己的需求来编写带有正则表达式的SQL语句。以下我们将列出几个小实例(表名：person_tbl)来加深我们的理解：

查找name字段中以'st'为开头的所有数据：

```
mysql> SELECT name FROM person_tbl WHERE name REGEXP '^st';
```

查找name字段中以'ok'为结尾的所有数据：

```
mysql> SELECT name FROM person_tbl WHERE name REGEXP 'ok$';
```

查找name字段中包含'mar'字符串的所有数据：

```
mysql> SELECT name FROM person_tbl WHERE name REGEXP 'mar';
```

查找name字段中以元音字符开头或以'ok'字符串结尾的所有数据：

```
mysql> SELECT name FROM person_tbl WHERE name REGEXP '^[aeiou]|ok$';
```

[← MySQL NULL 值处理](#)

[MySQL 事务 →](#)

 [点我分享笔记](#)