

(注意mysql中的正则表达式仅仅是正则表达式中的一个**小部分**而已)

## 一. REGEXP操作符

基本操作如下:

### 1. 基本匹配

WHERE prod\_name REGEXP "1000";(注意这里1000只要在值中**出现**即可)

### 2. 单字符匹配.

(注意这里.是用来匹配任意一个字符, 与**数量无关**!! 与**种类有关**!!)

WHERE prod\_name REGEXP '.000';

### 3. 多种类匹配|

(与下面的内容区别, |是用来分隔多个**整体**的, 这也就是为什么我是用种类一词)

WHERE prod\_name REGEXP '1000|2000';

### 4. 多字符匹配[]

(【】的基本用法之一, 搜索到的**字符**如果在里面**出现过**, 则匹配成功, 注意不是整体)

WHERE prod\_name REGEXP '[123] TON';

### 5. 匹配范围[]

WHERE prod\_name REGEXP '[1-3] TON';

### 6. 反匹配[^]

(用来否定字符集合, 如[^123], 代表**匹配除123以外的任何字符**)

做个小总结, 这里星光最璀璨的就是[]了, []有三种用法, 大家自己总结一下。

下面介绍稍微高级一点的操作:

### 1. 匹配多个实例 (与上面4区分!)

\*, +, ?, {n}, {n, m}

### 2. 匹配字符类 (注意这里的关键在于类)

mysql已经帮你预先定义好了一些合集, 你可以直接使用, 比如[:alpha:], [:digit:]等;

### 3. 定位符

之前已经提到过正则表达式的匹配是部分的匹配 (即匹配字符串中的**任意一个文本**), 为了让位置固定 (比如我想搜以什么字符开头的字符串), 我们需要引入定位符。

1. ^: 文本的开始
2. \$: 文本的结束
3. [[:<:]]: 词的开始
4. [[:>:]]: 词的结束

(注意后面两个是唯一要使用到两对中括号的符号)

## 二. 关于正则表达式的一些注意事项

1. mysql中的正则表达式**不区分大小写（唯一不区分大小写的）**，为区分大小写，我们要在搜索模式前加入**BINARY**(这样做的好处是增加了灵活性)
2. 在匹配特殊字符的时候（如.?\这些已经有定义的符号），我们需要进行转义(escaping)，如\\匹配.,\\\匹配\。这里用两个\的原因是，**mysql要解释一个，正则表达式又要解释一个**。  
(注意使用\的还有一个地方是**元字符**，即不能用键盘键入的字符们)

## 三. LIKE和REGEXP的对比

1. LIKE**区分大小写**，REGEXP**不区分大小写**，但是可以通过添加BINARY使其区分大小写。
2. LIKE直接从文本开始的地方进行匹配，REGEXP则不是，我们可以通过引入定位符来^\$来使REGEXP趋近于LIKE
3. 总的来讲，LIKE所能用的通配符较少，但是使用频率高；REGEXP支持的通配符多，可以通过引入关键字改变功能，因此更加灵活。