



链滴

MySQL 全文检索基础

作者: [umeone](#)

原文链接: <https://ld246.com/article/1516948965353>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

MySQL全文检索基础

1. 数据库引擎

数据库引擎使用MyISAM类型，InnoDB引擎不支持全文检索功能。此外，不同版本对全文检索的支持可能有差异，请尽量保证使用4.X以上的版本。

2. 数据库表的创建

数据库表在创建的时候需要用到FULLTEXT关键字进行指定需要创建索引的列，其中列可以包含多个，用进行分割。如：

```
CREATE TABLE product
(
  id INT NOT NULL AUTO INCREMENT,
  name VARCHAR(10) NOT NULL,
  number INT NOT NULL,
  description VARCHAR(255) NOT NULL,
  PRIMARY KEY(id),
  FULLTEXT(description)
) ENGINE=MyISAM;
```

其中FULLTEXT(description)就是为name列简历索引。

3. 索引查询

MySQL通过Match()和Against()两个函数进行索引查询。示例如下：

```
select name,description from product where Match(description) Against('描述关键词');
```

在如上示例中即可搜索出关键词相关数据。

4. 查询拓展

在查询过程中，如果我们启用了拓展查询，那么在查询数据的时候，及时有些数据没有包含我们查询关键字也同样会一并返回。大致规则可以理解成：当我们查询的关键字有匹配数据的时候，如果其他据中存在有匹配数据中出现过的关键字（不是查询关键字）也同样会一并返回，区别仅仅是优先级没那么高，也就是索引评分没那么高而已。示例如下：

```
select name,description from product where Match(description) Against('描述关键词' WITH QUERY EXPANSION);
```

可以看到如上示例仅仅是在普通查询上多了几个关键字WITH QUERY EXPANSION。

5. 布尔文本检索

布尔文本检索通过关键词IN BOOLEAN MODE来实现，示例如下：

```
select name,description from product where Match(description) Against('操作符表达式' IN BOOLEAN MODE);
```

其中关键点在操作符表达式这里，通过不同的表达式实现不同的结果筛选

操作符说明如下：

操作符

+
-
>

说明

包含，词必须存在
排除，词必须不出现
包含，而且增加等级值

<	包含, 且减少等级值
()	把词组成子表达式 (允许这些子表达式作为
个组被包含、排除、排列等)	
~	取消一个词的排序值
*	词尾的通配符
""	定义一个短语 (与单个词的列表不一样, 它匹
整个短语以便包含或排除这个短语)	

通过以上操作符的不同组合实现不同的效果

以上则为MySQL全文检索的一些基本操作说明