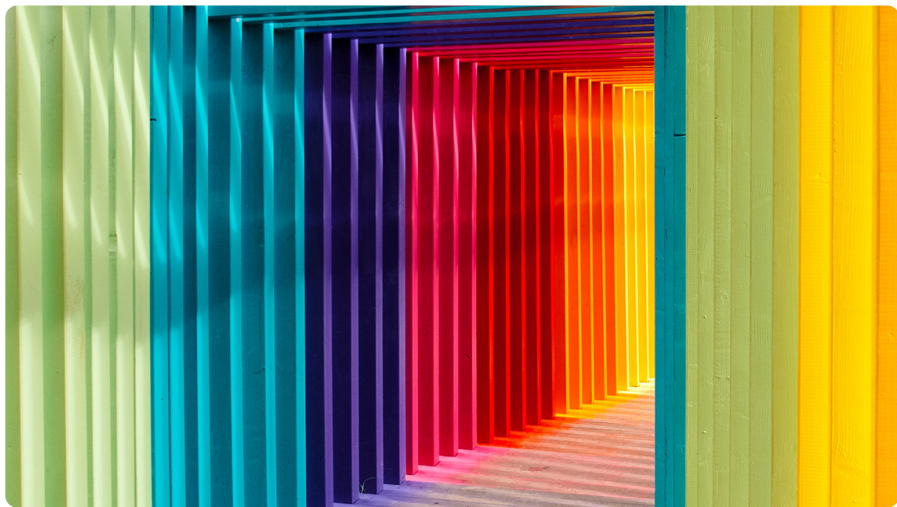


## 06 | 数据过滤：SQL数据过滤都有哪些方法？

2019-06-24 陈旸

SQL必知必会

[进入课程 >](#)



**讲述：陈旸**

时长 08:46 大小 8.04M



我在上篇文章中讲到过，提升查询效率的一个很重要的方式，就是约束返回结果的数量，还有一个很有效的方式，就是指定筛选条件，进行过滤。过滤可以筛选符合条件的结果，并进行返回，减少不必要的数据行。

那么在今天的内容里，我们来学习如何对 SQL 数据进行过滤，这里主要使用的就是 WHERE 子句。

你可能已经使用过 WHERE 子句，说起来 SQL 其实很简单，只要能把满足条件的内容筛选出来即可，但在实际使用过程中，不同人写出来的 WHERE 子句存在很大差别，比如执行效率的高低，有没有遇到莫名的报错等。

在今天的学习中，你重点需要掌握以下几方面的内容：

1. 学会使用 WHERE 子句，如何使用比较运算符对字段的数值进行比较筛选；
2. 如何使用逻辑运算符，进行多条件的过滤；
3. 学会使用通配符对数据条件进行复杂过滤。

## 比较运算符

在 SQL 中，我们可以使用 WHERE 子句对条件进行筛选，在此之前，你需要了解 WHERE 子句中的比较运算符。这些比较运算符的含义你可以参见下面这张表格：


含义	运算符
等于	=
不等于	<>或!=
小于	<
小于等于（大不予）	<=或!>
大于	>
大于等于（不小于）	>=或!<
不小于	!<
在指定的两个数值之间	BETWEEN
为空值	IS NULL

实际上你能看到，同样的含义可能会有多种表达方式，比如小于等于，可以是（<=），也可以是不大于（!>）。同样不等于，可以用（<>），也可以用（!=），它们的含义都是相同的，但这些符号的顺序都不能颠倒，比如你不能写（=<）。需要注意的是，你需要查看使用的 DBMS 是否支持，不同的 DBMS 支持的运算符可能是不同的，比如 Access 不支持（!=），不等于应该使用（<>）。在 MySQL 中，不支持（!>）（!<）等。

我在上一篇文章中使用了 heros 数据表，今天还是以这张表格做练习。下面我们通过比较运算符对王者荣耀的英雄属性进行条件筛选。

WHERE 子句的基本格式是：SELECT ..... (列名) FROM .....  
(表名) WHERE ..... (子句条件)

比如我们想要查询所有最大生命值大于 6000 的英雄：

 复制代码


```
1 SQL: SELECT name, hp_max FROM heros WHERE hp_max > 6000
```



运行结果 (41 条记录)：

name	hp_max
夏侯惇	7350
钟无艳	7000
张飞	8341
.....	.....
凯	6700

想要查询所有最大生命值在 5399 到 6811 之间的英雄：

 复制代码

```
1 SQL: SELECT name, hp_max FROM heros WHERE hp_max BETWEEN
```




运行结果：（41 条记录）

name	hp_max
半月	6164
雅典娜	6264
后裔	5986
.....	.....
百里守约	5611

需要注意的是hp\_max可以取值到最小值和最大值，即 5399 和 6811。

我们也可以对 heros 表中的hp\_max字段进行空值检查。

 复制代码

```
1 SQL: SELECT name, hp_max FROM heros WHERE hp_max IS NUL
```



运行结果为空，说明 heros 表中的hp\_max字段没有存在空值的数据行。


# 逻辑运算符

我刚才介绍了比较运算符，如果我们存在多个 WHERE 条件子句，可以使用逻辑运算符：

含义	逻辑运算符
并且	AND
或者	OR
在指定条件范围内	IN
非（否定）	NOT

我们还是通过例子来看下这些逻辑运算符的使用，同样采用 heros 这张表的数据查询。

假设想要筛选最大生命值大于 6000，最大法力大于 1700 的英雄，然后按照最大生命值和最大法力值之和从高到低进行排序。

 复制代码


```
1 SQL: SELECT name, hp_max, mp_max FROM heros WHERE hp_ma
```



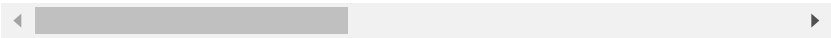
运行结果：（23 条记录）

name	hp_max	mp_max
廉颇	9328	1708
牛魔	8476	1926
刘邦	8073	1940
.....	.....	.....
孙尚香	6014	1756

如果 AND 和 OR 同时存在 WHERE 子句中会是怎样的呢？  
假设我们想要查询最大生命值加最大法力值大于 8000 的英雄，或者最大生命值大于 6000 并且最大法力值大于 1700 的英雄。

 复制代码

```
1 SQL: SELECT name, hp_max, mp_max FROM heros WHERE (hp_m
```




运行结果：（33 条记录）

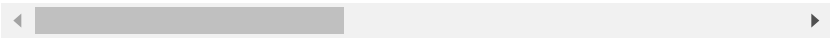
name	hp_max	mp_max
廉颇	9328	1708
牛魔	8476	1926
白起	8638	1666
.....	.....	.....
孙尚香	6014	1756

你能看出来相比于上一个条件查询，这次的条件查询多出来了 10 个英雄，这是因为我们放宽了条件，允许最大生命值 + 最大法力值大于 8000 的英雄显示出来。另外你需要注意到，当 WHERE 子句中同时存在 OR 和 AND 的时候，AND 执行的优先级会更高，也就是说 SQL 会优先处理 AND 操作符，然后再处理 OR 操作符。

如果我们对这条查询语句 OR 两边的条件增加一个括号，结果会是怎样的呢？

 复制代码

```
1 SQL: SELECT name, hp_max, mp_max FROM heros WHERE ((hp_
```




运行结果：

name	hp_max	mp_max
廉颇	9328	1708
牛魔	8476	1926
刘邦	8073	1940
.....	.....	.....
孙尚香	6014	1756



所以当 WHERE 子句中同时出现 AND 和 OR 操作符的时候，你需要考虑到执行的先后顺序，也就是两个操作符执行的优先级。一般来说 () 优先级最高，其次优先级是 AND，然后是 OR。

如果我想要查询主要定位或者次要定位是法师或是射手的英雄，同时英雄的上线时间不在 2016-01-01 到 2017-01-01 之间。

 复制代码

```
1 SQL:
2 SELECT name, role_main, role_assist, hp_max, mp_max, bi
3 FROM heros
4 WHERE (role_main IN ('法师', '射手') OR role_assist IN ('法师', '射手'))
5 AND DATE(birthdate) NOT BETWEEN '2016-01-01' AND '2017-01-01'
6 ORDER BY (hp_max + mp_max) DESC
```

你能看到我把 WHERE 子句分成了两个部分。第一部分是关于主要定位和次要定位的条件过滤，使用的是 `role_main in ('法师', '射手') OR role_assist in ('法师', '射手')`。这里用到了 IN 逻辑运算符，同时 `role_main` 和 `role_assist` 是 OR（或）的关系。

第二部分是关于上线时间的条件过滤。NOT 代表否，因为我们要找到不在 2016-01-01 到 2017-01-01 之间的日期，因此用到了NOT BETWEEN '2016-01-01' AND '2017-01-01'。同时我们是在对日期类型数据进行检索，所以使用到了 DATE 函数，将字段 birthdate 转化为日期类型再进行比较。关于日期的操作，我会在下一篇文章中再作具体介绍。


这是运行结果（6 条记录）：

name	role_main	role_assist	hp_max	mp_max	birthdate
张良	法师		5799	1988	2015-10-26
貂蝉	法师	刺客	5611	1960	2015-12-15
干将莫邪	法师		5583	1946	2017-05-22
周瑜	法师		5513	1974	2015-11-10
百里守约	射手	刺客	5611	1784	2017-08-08
芈月	法师	坦克	6164	100	2015-12-08

## 使用通配符进行过滤

刚才讲解的条件过滤都是对已知值进行的过滤，还有一种情况是我们要检索文本中包含某个词的所有数据，这里就需要使用通配符。通配符就是我们用来匹配值的一部分的特殊字符。这里我们需要使用到 LIKE 操作符。

如果我们想要匹配任意字符串出现的任意次数，需要使用 (%) 通配符。比如我们想要查找英雄名中包含“太”字的英雄都有哪些：

 复制代码


```
1 SQL: SELECT name FROM heros WHERE name LIKE '% 太 %'
```

运行结果：（2 条记录）

name
东皇太一
太乙真人

需要说明的是不同 DBMS 对通配符的定义不同，在 Access 中使用的是 (\*) 而不是 (%)。另外关于字符串的搜索可能是需要区分大小写的，比如 'liu%' 就不能匹配上 'LIU BEI'。具体是否区分大小写还需要考虑不同的 DBMS 以及它们的配置。

如果我们想要匹配单个字符，就需要使用下划线 ( `_` ) 通配符。 ( `%` ) 和 ( `_` ) 的区别在于， ( `%` ) 代表一个或多个字符，而 ( `_` ) 只代表一个字符。比如我们想要查找英雄名除了第一个字以外，包含“太”字的英雄有哪些。

 复制代码

```
1 SQL: SELECT name FROM heros WHERE name LIKE '_% 太 %'
```

运行结果 (1 条记录)：

name
东皇太一

因为太乙真人的太是第一个字符，而 `_% 太 %` 中的太不是在第一个字符，所以匹配不到“太乙真人”，只可以匹配上“东皇太一”。

同样需要说明的是，在 Access 中使用 ( `?` ) 来代替 ( `_` ) ，而且在 DB2 中是不支持通配符 ( `_` ) 的，因此你需要在使用的时候查阅相关的 DBMS 文档。

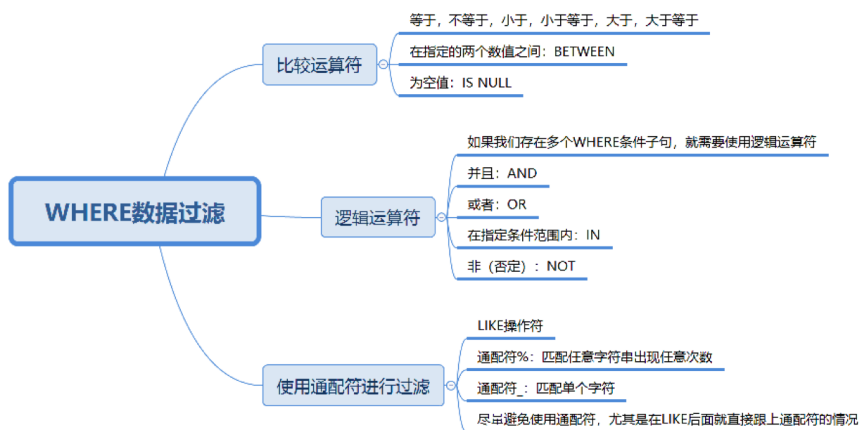
你能看出来通配符还是很有用的，尤其是在进行字符串匹配的时候。不过在实操过程中，我还是建议你尽量少用通配符，因为它需要消耗数据库更长的时间来进行匹配。即使你对 LIKE 检索的字段进行了索引，索引的价值也可能会失效。如果要想让索引生效，那么 LIKE 后面就不能以 (%) 开头，比如使用 `LIKE '%太%'` 或 `LIKE '%太'` 的时候就会对全表进行扫描。如果使用 `LIKE '太%'`，同时检索的字段进行了索引的时候，则不会进行全表扫描。

## 总结

今天我对 SQL 语句中的 WHERE 子句进行了讲解，你可以使用比较运算符、逻辑运算符和通配符这三种方式对检索条件进行过滤。

比较运算符是对数值进行比较，不同的 DBMS 支持的比较运算符可能不同，你需要事先查阅相应的 DBMS 文档。逻辑运算符可以让我们同时使用多个 WHERE 子句，你需要注意的是 AND 和 OR 运算符的执行顺序。通配符可以让我们对文本类型的字段进行模糊查询，不过检索的代价也是很高的，通常都需要用到全表扫描，所以效率很低。只有当 LIKE 语句后面不用通配符，并且对字段进行索引的时候才不会对全表进行扫描。

你可能认为学习 SQL 并不难，掌握这些语法就可以对数据进行筛选查询。但实际工作中不同人写的 SQL 语句的查询效率差别很大，保持高效率的一个很重要的原因，就是要避免全表扫描，所以我们会考虑在 WHERE 及 ORDER BY 涉及到的列上增加索引。



你能说一下 WHERE 子句中比较运算符、逻辑运算符和通配符这三者各自的作用吗？以 heros 数据表为例，请你编写 SQL 语句，对英雄名称、主要定位、次要定位、最大生命和最大法力进行查询，筛选条件为：主要定位是坦克或者战士，并且次要定位不为空，同时满足最大生命值大于 8000 或者最大法力小于 1500 的英雄，并且按照最大生命和最大法力之和从高到底的顺序进行排序。

欢迎你在评论区写下你的思考，也欢迎点击请朋友读，把这篇文章分享给你的朋友或者同事。



# SQL 必知必会

## 从入门到数据实战

陈 旻  
清华大学计算机博士



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 05 | 检索数据：你还在SELECT \* 么？

## 精选留言 (21)

 写留言



一步

2019-06-24

所以使用到了 DATE 函数，将字段 birthdate 转化为日期类型再进行比较

-----

对于日期的比较，不是可以直接进行比较吗？对于上面的例子，直接可以使用 birthdate 字段进行时间比较，为什么还要使用DATE函数转换一下呢？

💬 1

👍 6



## 一步

2019-06-24

就是要避免全表扫描，所以我们会考虑在 WHERE 及 ORDER BY 涉及到的列上增加索引

-----

where 条件字段上加索引是可以明白的，但是为什么 order by 字段上还要加索引呢？这个时候已经通过...

作者回复: 这是一个很好的问题，关于ORDER BY字段是否增加索引：

在MySQL中，支持两种排序方式：FileSort和Index排序。

Index排序的效率更高，

Index排序：索引可以保证数据的有序性，因此不需要再进行排序。

FileSort排序：一般在内存中进行排序，占用CPU较多。如果待排结果较大，会产生临时文件I/O到磁盘进行排序，效率较低。

所以使用ORDER BY子句时，应该尽量使用Index排序，避



免使用FileSort排序。

当然具体优化器是否采用索引进行排序，你可以使用explain来进行执行计划的查看。

优化建议：

1、SQL中，可以在WHERE子句和ORDER BY子句中使用索引，目的是在WHERE子句中避免全表扫描，ORDER BY子句避免使用FileSort排序。

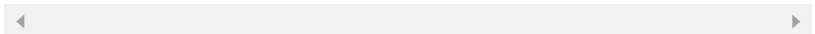
当然，某些情况下全表扫描，或者FileSort排序不一定比索引慢。但总的来说，我们还是要避免，以提高查询效率。

一般情况下，优化器会帮我们进行更好的选择，当然我们也需要建立合理的索引。

2、尽量Using Index完成ORDER BY排序。

如果WHERE和ORDER BY相同列就使用单索引列；如果不同使用联合索引。

3、无法Using Index时，对FileSort方式进行调优。



1



1

这个需求  
做不了

啦啦啦

2019-06-24

打卡打卡



1



leol

2019-06-24

```
select name, hp_max, mp_max, role_main, role_assist
from heros
where (role_main in ('坦克' or '战士') and role_assist
is not null)
and (hp_max > 8000 or hp_max < 1500)...
```



**许童童**

2019-06-24

```
SELECT name, role_main, role_assist, hp_max, mp_max
from heros WHERE role_main IN ('战士', '坦克') AND
role_assist IS NOT NULL AND (hp_max > 8000 OR
mp_max < 1500) ORDER BY (hp_max + mp_max)
DESC
```



**mickey**

2019-06-24

**MYSQL:**

```
SELECT name, role_main, role_assist, hp_max,
mp_max
FROM heros
WHERE ( role_main = '坦克' or role_main = '战士' )...
```





**Destroy,**

2019-06-24

```
select name, role_main, role_assist from heros where  
(role_main in ('坦克', '战士') and role_assist is not  
null) and (hp_max > 8000 or mp_max < 1500) order  
by (hp_max + mp_max) desc;
```



**太乙鯉**

2019-06-24

```
select name,role_main,role_assist,mp_max,hp_max  
from heros where role_main in('战士','坦克') and  
role_assist is not null and (hp_max > 8000 or  
mp_max<1500) order by (hp_max+mp_max) desc;
```



**Samson**

2019-06-24

老师，似乎有个问题：“(%)”和“( )”的区别在于，  
(%) 代表一个或多个字符，而( ) 只代表一个字符。”

%似乎是代表0个或任意正整数个字符，而不是一个或...





**hedeqiang**

2019-06-24

```
SELECT name,role_main,role_assist,hp_max,mp_max  
FROM heros WHERE ((role_main in ('战士','坦克'))  
AND (role_assist IS NOT NULL))  
AND (hp_max > 8000 OR mp_max < 1500) ORDER  
BY hp_max+mp_max DESC;
```

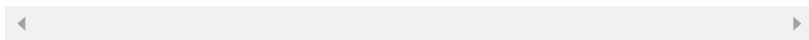


**flow**

2019-06-24

关于通配符匹配里的 % 相当于正则表达式里的 .\* 表示匹配大于等于0个任意字符，  
所以 % 太 % 匹配的是 [大于等于0个任意字符]太[大于等于0个任意字符]，[东皇]太[一] 和 []太[乙真人]都符合；  
而 \_% 相当于正则表达式里的 .+ 表示匹配至少一个，...

作者回复: 对的 解释正确



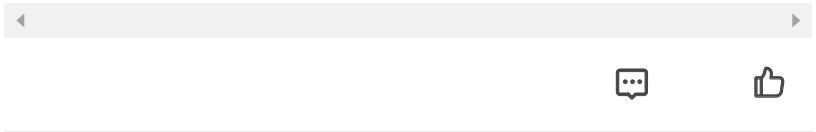
**郡鸿**

2019-06-24

```
select name,role_main,role_assist,hp_max,mp_max  
from heros where
```

```
(role_main in ('坦克','战士')
and role_assist is not null)
and (hp_max > 8000 or mp_max < 1500) order by...
```

作者回复: 正确!

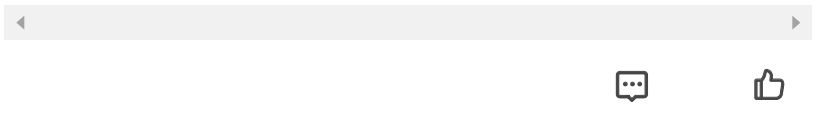


**flow**

2019-06-24

```
select name, role_main, role_assist, hp_max, mp_max
from heros
where role_main in ('坦克', '战士')
and role_assist is not null
and (hp_max > 8000 or mp_max < 1500)...
```

作者回复: 正确!



**悟空**

2019-06-24

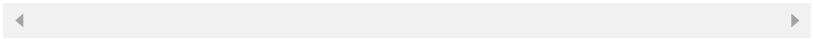
老师关于通配符给的解释, 不够清晰!  
说明如下:

```
SQL: SELECT name FROM heros WHERE name LIKE
```

'\_% 太 %'

...

作者回复: 哈哈 太乙真人太太 这个解释的好。也就是在第一个字符之后能匹配上“太”字。



**葛亮**

2019-06-24

通配符%不应该是任意字符吗（也就是含0个字符），至少SqlServer下是这样的



**野马**

2019-06-24

SELECT 英雄名称,主要定位,次要定位,最大生命,最大法力  
FROM heros WHERE （主要定位 IN('坦克','战士')OR次要定位 IS NOT NULL） AND 最大生命>8000 AND 最大法力<1500 ORDER BY (最大生命+最大法力) DESC



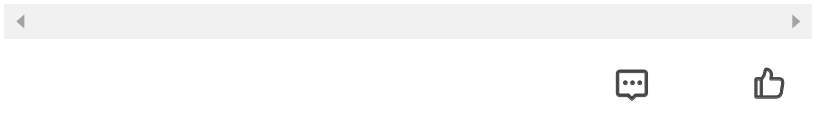
**hlz-123**

2019-06-24

where子句WHERE 子句中比较运算符、逻辑运算符和通配符这三者各自作用？

- 1、比较运算符，比较数值的大小，数值类型可以是整数，浮点数，字符串，布尔类型等等。
- 2、逻辑运算符，定义where子句中多个条件之间的关...

作者回复: 解释的很好，最后一个SQL查询也正确



小豹子

2019-06-24

比较运算符一般是单字段与值或两个字段之间进行比较(使用><=!), 逻辑运算符是一个或多个逻辑语句的拼接(使用not, and, or), 通配符一般只用于字符串里面(使用like)



演

2019-06-24

- 打卡
- SQL: `SELECT name FROM heros WHERE name LIKE '%太 %'`

可以筛选出包含'太'且不会出现在首位置的英雄名字



**莱茵金属**

2019-06-24

请问老师的课程针对那些层面的程序员？

