

## 目录

1. 基本匹配条件: .....	1
1.1 范围匹配.....	1
1.2 逻辑匹配.....	1
2. 高级匹配条件: .....	2
2.1 模糊查询.....	2
2.2 正则表达式.....	3
2.3 数学计算.....	4

## 1. 基本匹配条件:

### 1.1 范围匹配

- 匹配范围内的任意一个值即可

类 型	比 较
in (值列表)	在...里...
not in (值列表)	不在...里...
between 数字 and 数字	在...之间...

查询user表中name, uid字段的值, name字段必须在"mysql","bin","sync","null" 范围内

```
mysql> select name,uid from user where name in("mysql","bin","sync","null");
```

查询user表中name, uid字段的值, 其中uid字段的值必须在3,6,9,15范围内

```
mysql> select name,uid from user where uid in(3,6,9,15);
```

查询user表中name, shell字段的值, shell字段的值不在"/bin/bash","/sbin/nologin"范围内

```
mysql> select name,shell from user where shell not in("/bin/bash","/sbin/nologin");
```

查询user表中name, uid字段的值,其中uid字段的值必须在15和30之间

```
mysql> select name,uid from user where uid between 15 and 30;
```

查询user表中的所有字段的记录,其中id字段的值必须是在10和13之间

```
mysql> select * from user where id between 10 and 13;
```

### 1.2 逻辑匹配

- 多个判断条件时使用

类 型	用 途	格 式
or	逻辑或	条件1 or 条件2 or 条件3
and	逻辑与	条件1 and 条件2 and 条件3
! 或 not	逻辑非	

查询user表中name, uid字段的值,name字段的值是"root" 且shell字段的值为  
"/sbin/nologin"

```
mysql> select name,uid from user where name="root" and shell="/sbin/nologin";
```

查询user表中name, uid,shell字段的值,name字段的值是"root"或者 "/sbin/nologin"

```
mysql> select name,uid,shell from user where name="root" or shell="/sbin/nologin";
```

查询user表中name, uid字段的值,name字段的值是"root" 并且shell字段的值为"/bin/bash"

```
mysql> select name,uid from user where name="root" and shell="/bin/bash";
```

查询user表中name,shell字段的值,其中shell字段的值必须是"/bin/bash"

```
mysql> select name,shell from user where shell = "/bin/bash";
```

查询user表中name,shell字段的值,其中shell字段的值不是"/bin/bash"

```
mysql> select name,shell from user where shell != "/bin/bash";
```

查询user表中name,shell字段的值,shell字段的值在("/bin/bash","/sbin/nologin") 范围内

```
mysql> select name,shell from user where shell in ("/bin/bash","/sbin/nologin");
```

查询user表中name,shell字段的值,shell字段的值不在("/bin/bash","/sbin/nologin") 范围内

```
mysql> select name,shell from user where shell not in ("/bin/bash","/sbin/nologin");
```

## 2. 高级匹配条件:

### 2.1 模糊查询

用法: where 字段名 like '通配符'

\_ 表示 1 个字符

% 表示 0~n 个字符

查询user表中name字段的记录, 其中name字段的记录必须是由任意4个字符组成的

```
mysql> select name from user where name like '____';
```

查询user表中name字段的记录, 其中name字段的记录中必须包含a

```
mysql> select name from user where name like '%a%';
```

查询user表中name字段的记录, 其中name字段的记录是由a开头的

```
mysql> select name from user where name like 'a%';
```

模糊匹配 % 和 \_ 混合使用

查询user表中name字段，name字段的记录必须最少由4个字符组成

```
mysql> select name from user where name like '__%__';
```

## 2.2 正则表达式

用法: where 字段 regexp '正则表达式'

正则元字符 : ^ (以...开头) \$ (以...结尾) . (任意一个字符) [] (范围内匹配)

\* (表达式出现的次数) | (或)

查询user表中name字段的记录，其中name字段的记录必须是以 r 开头的

```
mysql> select name from user where name regexp '^r';
```

查询user表中name字段的记录，其中name字段的记录必须是以 a开头的

```
mysql> select name from user where name regexp '^a';
```

查询user表中name字段的记录，其中name字段的记录必须是以 a开头或以t结尾的

```
mysql> select name from user where name regexp '^a|t$';
```

查询user表中name字段的记录，其中name字段的记录必须是以 r 开头并且以t结尾的

```
mysql> select name from user where name regexp '^r.*t$';
```

#.\*任意字符

查找用户名包含数字的name字段

```
mysql> select name from user where name regexp '[0-9]';
```

#查询为空

```
mysql> insert into user(name) values ("yaya9"),("6yaya"),("ya8ya"),("y7aya");
```

```
mysql> select name from user where name regexp '[0-9]';
```

#查询成功

查询以数字开头的name字段

```
mysql> select name from user where name regexp '^[0-9]';
```

## 2.3 数学计算

- 字段必须是数值类型

符号	用途	例子
+	加法	uid + gid
-	减法	uid - gid
*	乘法	uid * gid
/	除法	uid / gid
%	取余数 (求模)	uid % gid
()	提高优先级	(uid + gid) / 2

查询user表中id,name,uid 字段的记录，其中id字段的记录小于等于5

```
mysql> select id,name,uid from user where id <=5;
```

修改user表中id字段小于等于5的值，uid字段的值都自加1

```
mysql> update user set uid=uid+1 where id <= 5;
```

```
mysql> select id,name,uid from user where id <=5;
```

修改user表中id字段小于等于5的值，uid字段的值都自减1

```
mysql> update user set uid=uid-1 where id <= 5;
```

```
mysql> select id,name,uid from user where id <=5;
```

查询user表中name,uid 字段的记录，其中uid字段的值必须为偶数

```
mysql> select name,uid from user where uid % 2 = 0;
```

查询user表中name,uid 字段的记录，其中uid字段的值必须为奇数

```
mysql> select name,uid from user where uid % 2 != 0;
```

计算halt用户uid和gid字段的平均值

```
mysql> select name,uid,gid from user where name="halt";
```

```
mysql> select name,uid,gid,(uid+gid)/2 from user where name="halt"; #显示uid和gid
```

字段的平均值 (uid+gid)/2，默认以算法作为临时字段名

设置临时字段名

```
mysql> select name,uid,gid,(uid+gid)/2 pjz from user where name="halt";
```

计算出出生年月(2023 - age 当前年份减去年龄)

在user表中的name字段的后面插入一条新的age字段

```
mysql> alter table user add age tinyint unsigned default 20 after name;
```

```
mysql> select name,age,2023 - age start_y from user where name="root"
```

#start\_y临时字段名