

Mysql基础知识

TIPS

- SQL语句都应该以分号结尾";"。

常见关键字

使用数据库

- SHOW

查询数据库操作：

```
SHOW DATABASES;  
SHOW TABLES;  
SHOW COLUMNS FROM TAB_NAME;  
SHOW CREATE DATABASE;  
SHOW CREATE TABLE;
```

查询数据库的状态：

```
SHOW STATUS;  
SHOW ERRORS;  
SHOW WARNINGS;  
SHOW GRANTS;
```

查询

- SELECT
 - 通配符 (*)：查询所有的列。
 - DISTINCT：对查询结果去重操作。使用必须放在所有列的前面，作用于所有列。
 - LIMIT OFFSET, SIZE：限制查询结果的容量，返回第(OFFSET)至(OFFSET+SIZE)条。
 - 完全限定名 (TAB_NAME.COL_NAME)：使用联合查询时候，两张表可能存在一样的字段名。

查询字段信息：

```
SELECT COL_NAME FROM TAB_NAME;
SELECT * FROM TAB_NAME; // * 是通配符，返回数据包括所有字段
SELECT DISTINCT COL_NAME1, COL_NAME2 FROM TAB_NAME; // DISTINCT是去重操作(作用域必须是片
SELECT COL_NAME FROM TAB_NAME LIMIT OFFSET, SIZE; // 返回特定OFFSET区间内的数据
SELECT TAB_NAME.COL_NAME FROM TAB_NAME; //通过字段的完全限定名查询
```

子句

SQL 语句通常由多个字句构成。有些子句是必须的，有些不是必须的。常见的又：FROM 子句、WHERE 子句、ORDER BY 子句。

- ORDER BY：有两种选项，ASC|DESC（默认是ASC）。按照多个字段进行排序时，必须为每个字段都制定特定的排序顺序。

```
SELECT * FROM TAB_NAME ORDER BY COL_NAME1; //查询结果按照COL_NAME1升序进行排列（缺省即默认
SELECT * FROM TAB_NAME ORDER BY COL_NAME ASC; //查询结果按照COL_NAME1升序进行排列。
SELECT * FROM TAB_NAME ORDER BY COL_NAME1 DESC; //查询结果按照COL_NAME1 降序排列。
SELECT * FROM TAB_NAME ORDER BY COL_NAME1 DESC,COL_NAME2 ASC; // 查询结果先按照COL_NAM
SELECT * FROM TAB_NAME ORDER BY COL_NAME1 LIMIT OFFSET, SIZE; // 通过LIMIT 对查询结果进
注：
ORDER BY 对查询结果大小写（A，a）的区分在与数据库的配置。mysql的默认排序方式：字典排序顺序
LIMIT 必须放在 ORDER BY 之后。
```

- WHERE 子句：指定搜索条件，过滤数据（相较于应用层的筛选，引擎层的筛选可以减少传输带宽、减少应用层二次计算）。

操作符	含义
=	等于
<	小于
>	大于
<>	不等于
!=	不等于
<=	小于等于
>=	大于等于

```
SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 = 'BOB';
// 符合 COL_NAME1 等于BOB条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 < 18;
// 符合 COL_NAME2 小于18 条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 > 18;
// 符合 COL_NAME2 大于18 条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 != 18;
// 符合 COL_NAME2 不等于18 条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 <> 18;
// 同上
```

操作符	含义
IS NULL	判断空值
BETWEEN VALUE1 AND VALUE2	位于 VALUE1 和 VALUE2 之间

```
SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 BETWEEN 2 AND 18;
// 符合 COL_NAME2 在[2, 18] 条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME2 IS NULL;
//符合 COL_NAME2 为 NULL 的条件

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 ="BOB" ORDER BY TAB_NAME.COL_NAME1 D
//符合 COL_NAME1 等于 BOB 条件的数据，并且按照 COL_NAME1 降序排列
```

注：

ORDER 必须放在 WHERE 条件语句的之后。

IS NULL： 必须是COL_NAME1 字段为NULL；查询结果为空并不代表 IS NULL。

操作符	含义
AND	联结两个条件，表示两个条件都成立
OR	联结两个条件，表示两个条件成立一个即可
IN	指定条件范围
NOT	否定后续所有的条件

```
SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 = 'BOB' AND TAB_NAME.COL_NAME2 < 18;
//符合 COL_NAME1 等于 BOB 、 COL_NAME2 小于 18 两个条件的数据

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 = 'B0B' AND TAB_NAME.COL_NAME2 < 18;
//符合 COL_NAME1 等于 BOB 或者 COL_NAME2 小于 18 其中之一数据

SELECT * FROM TAB_NAME WHERE (TAB_NAME.COL_NAME1 = 'BOB' OR TAB_NAME.COL_NAME2 < 18)
//符合 COL_NAME1 等于 BOB 且 COL_NAME2 小于 18 条件；或者 COL_NAME > 2 条件，满足之一即可。

SELECT * FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 = 'BOB' OR TAB_NAME.COL_NAME2 < 18
//符合 COL_NAME1 等于 BOB 或者 COL_NAME2 在[2, 18]区间的数据。『AND 和 OR 两者有不同的优先:
//上下两句的语意并不相同

SELECT * FROM TAB_NAME WHERE COL_NAME2 IN (2, 3, 18);
//符合 COL_NAME2 在set(2, 3, 18) 中的数据。
```

注：

- AND 和 OR 两者的优先级不一致，使用时候 AND > OR，避免系统发生错误解析，需要添加括号。
- IN 较 OR 有更高的执行效率、IN 可以包含 SELECT 子句、可读性更好、计算次序更好管理。

- LIKE操作符号
 - 通配符：用来匹配值的一部分的特殊符号
 - 搜索模式：由字面值、通配符或者两者组合构成的搜索条件

操作符	含义
%	代表零个或者多个字符，但是不能匹配NULL
_	代表一个字符

```
SELECT TAB_NAME.COL_NAME1 FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 LIKE '% BOB ' ;
SELECT TAB_NAME.COL_NAME1 FROM TAB_NAME WHERE TAB_NAME.COL_NAME1 LIKE 'BOB_ ' ;
```

注：

- 使用通配符必须用LIKE关键字。
- 不能过度使用通配符，会造成性能下降
- 通配符放在最开始会造成性能下降最严重

- 正则表达式