

SQLite Where 子句

SQLite的 **WHERE** 子句用于指定从一个表或多个表中获取数据的条件。

如果满足给定的条件，即为真（true）时，则从表中返回特定的值。您可以使用 WHERE 子句来过滤记录，只获取需要的记录。

WHERE 子句不仅可用在 SELECT 语句中，它也可用在 UPDATE、DELETE 语句中，等等，这些我们将在随后的章节中学习。

语法

SQLite 的带有 WHERE 子句的 SELECT 语句的基本语法如下：

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition]
```

实例

您还可以使用[比较或逻辑运算符](#)指定条件，比如 >、<、=、LIKE、NOT，等等。假设 COMPANY 表有以下记录：

ID	NAME	AGE	ADDRESS	SALARY
-----	-----	-----	-----	-----
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

下面的实例演示了 SQLite 逻辑运算符的用法。下面的 SELECT 语句列出了 AGE 大于等于 25 **且** 工资大于等于 65000.00 的所有记录：

```
sqlite> SELECT * FROM COMPANY WHERE AGE >= 25 AND SALARY >= 65000;
```

ID	NAME	AGE	ADDRESS	SALARY
-----	-----	-----	-----	-----
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0

下面的 SELECT 语句列出了 AGE 大于等于 25 **或** 工资大于等于 65000.00 的所有记录：

```
sqlite> SELECT * FROM COMPANY WHERE AGE >= 25 OR SALARY >= 65000;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0

下面的 SELECT 语句列出了 AGE 不为 NULL 的所有记录，结果显示所有的记录，意味着没有一个记录的 AGE 等于 NULL：

```
sqlite> SELECT * FROM COMPANY WHERE AGE IS NOT NULL;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

下面的 SELECT 语句列出了 NAME 以 'Ki' 开始的所有记录，'Ki' 之后的字符不做限制：

```
sqlite> SELECT * FROM COMPANY WHERE NAME LIKE 'Ki%';
```

ID	NAME	AGE	ADDRESS	SALARY
6	Kim	22	South-Hall	45000.0

下面的 SELECT 语句列出了 NAME 以 'Ki' 开始的所有记录，'Ki' 之后的字符不做限制：

```
sqlite> SELECT * FROM COMPANY WHERE NAME GLOB 'Ki*';
```

ID	NAME	AGE	ADDRESS	SALARY
6	Kim	22	South-Hall	45000.0

下面的 SELECT 语句列出了 AGE 的值为 25 或 27 的所有记录：

```
sqlite> SELECT * FROM COMPANY WHERE AGE IN ( 25, 27 );
```

ID	NAME	AGE	ADDRESS	SALARY
2	Allen	25	Texas	15000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0

下面的 SELECT 语句列出了 AGE 的值既不是 25 也不是 27 的所有记录：

```
sqlite> SELECT * FROM COMPANY WHERE AGE NOT IN ( 25, 27 );
```

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
3	Teddy	23	Norway	20000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

下面的 SELECT 语句列出了 AGE 的值在 25 与 27 之间的所有记录：

```
sqlite> SELECT * FROM COMPANY WHERE AGE BETWEEN 25 AND 27;
```

ID	NAME	AGE	ADDRESS	SALARY
2	Allen	25	Texas	15000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0

下面的 SELECT 语句使用 SQL 子查询，子查询查找 SALARY > 65000 的带有 AGE 字段的所有记录，后边的 WHERE 子句与 EXISTS 运算符一起使用，列出了外查询中的 AGE 存在于子查询返回的结果中的所有记录：

```
sqlite> SELECT AGE FROM COMPANY
        WHERE EXISTS (SELECT AGE FROM COMPANY WHERE SALARY > 65000);
```

AGE
32
25
23
25
27
22
24

下面的 SELECT 语句使用 SQL 子查询，子查询查找 SALARY > 65000 的带有 AGE 字段的所有记录，后边的 WHERE 子句与 > 运算符一起使用，列出了外查询中的 AGE 大于子查询返回的结果中的年龄的所有记录：

```
sqlite> SELECT * FROM COMPANY
        WHERE AGE > (SELECT AGE FROM COMPANY WHERE SALARY > 65000);
```

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0

[← SQLite 表达式](#)

[SQLite AND/OR 运算符 →](#)

[✎ 点我分享笔记](#)