

## SQLite 子查询

子查询或内部查询或嵌套查询是在另一个 SQLite 查询内嵌入在 WHERE 子句中的查询。

使用子查询返回的数据将被用在主查询中作为条件，以进一步限制要检索的数据。

子查询可以与 SELECT、INSERT、UPDATE 和 DELETE 语句一起使用，可伴随着使用运算符如 =、<、>、>=、<=、IN、BETWEEN 等。

以下是子查询必须遵循的几个规则：

- 子查询必须用括号括起来。
- 子查询在 SELECT 子句中只能有一个列，除非在主查询中有多列，与子查询的所选列进行比较。
- ORDER BY 不能用在子查询中，虽然主查询可以使用 ORDER BY。可以在子查询中使用 GROUP BY，功能与 ORDER BY 相同。
- 子查询返回多于一行，只能与多值运算符一起使用，如 IN 运算符。
- BETWEEN 运算符不能与子查询一起使用，但是，BETWEEN 可在子查询内使用。

## SELECT 语句中的子查询使用

子查询通常与 SELECT 语句一起使用。基本语法如下：

```
SELECT column_name [, column_name ]
FROM   table1 [, table2 ]
WHERE  column_name OPERATOR
      (SELECT column_name [, column_name ]
      FROM table1 [, table2 ]
      [WHERE])
```

## 实例

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

现在，让我们检查 SELECT 语句中的子查询使用：

```
sqlite> SELECT *  
      FROM COMPANY  
     WHERE ID IN (SELECT ID  
                  FROM COMPANY  
                 WHERE SALARY > 45000) ;
```

这将产生以下结果:

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

## INSERT 语句中的子查询使用

子查询也可以与 INSERT 语句一起使用。INSERT 语句使用子查询返回的数据插入到另一个表中。在子查询中所选择的数据可以用任何字符、日期或数字函数修改。

基本语法如下：

```
INSERT INTO table_name [ (column1 [, column2 ]) ]  
      SELECT [ *|column1 [, column2 ]  
      FROM table1 [, table2 ]  
      [ WHERE VALUE OPERATOR ]
```

## 实例

假设 COMPANY\_BKP 的结构与 COMPANY 表相似，且可使用相同的 CREATE TABLE 进行创建，只是表名改为 COMPANY\_BKP。现在把整个 COMPANY 表复制到 COMPANY\_BKP，语法如下：

```
sqlite> INSERT INTO COMPANY_BKP  
      SELECT * FROM COMPANY  
     WHERE ID IN (SELECT ID  
                  FROM COMPANY) ;
```

## UPDATE 语句中的子查询使用

子查询可以与 UPDATE 语句结合使用。当通过 UPDATE 语句使用子查询时，表中单个或多个列被更新。

基本语法如下：

```
UPDATE table
SET column_name = new_value
[ WHERE OPERATOR [ VALUE ]
  (SELECT COLUMN_NAME
   FROM TABLE_NAME)
[ WHERE) ]
```

## 实例

假设，我们有 COMPANY\_BKP 表，是 COMPANY 表的备份。

下面的实例把 COMPANY 表中所有 AGE 大于或等于 27 的客户的 SALARY 更新为原来的 0.50 倍：

```
sqlite> UPDATE COMPANY
      SET SALARY = SALARY * 0.50
      WHERE AGE IN (SELECT AGE FROM COMPANY_BKP
                    WHERE AGE >= 27 );
```

这将影响两行，最后 COMPANY 表中的记录如下：

ID	NAME	AGE	ADDRESS	SALARY
-----	-----	-----	-----	-----
1	Paul	32	California	10000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	42500.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

## DELETE 语句中的子查询使用

子查询可以与 DELETE 语句结合使用，就像上面提到的其他语句一样。

基本语法如下：

```
DELETE FROM TABLE_NAME
[ WHERE OPERATOR [ VALUE ]
  (SELECT COLUMN_NAME
   FROM TABLE_NAME)
[ WHERE) ]
```

## 实例

假设，我们有 COMPANY\_BKP 表，是 COMPANY 表的备份。

下面的实例删除 COMPANY 表中所有 AGE 大于或等于 27 的客户记录：

```
sqlite> DELETE FROM COMPANY
        WHERE AGE IN (SELECT AGE FROM COMPANY_BKP
                      WHERE AGE > 27 );
```

这将影响两行，最后 COMPANY 表中的记录如下：

ID	NAME	AGE	ADDRESS	SALARY
-----	-----	-----	-----	-----
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	42500.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

← SQLite 事务

SQLite Autoincrement →

 点我分享笔记