

## SQLite NULL 值

SQLite 的 **NULL** 是用来表示一个缺失值的项。表中的一个 NULL 值是在字段中显示为空白的一个值。

带有 NULL 值的字段是一个不带有值的字段。NULL 值与零值或包含空格的字段是不同的，理解这点是非常重要的。

### 语法

创建表时使用 **NULL** 的基本语法如下：

```
SQLite> CREATE TABLE COMPANY(  
    ID INT PRIMARY KEY     NOT NULL,  
    NAME           TEXT     NOT NULL,  
    AGE            INT       NOT NULL,  
    ADDRESS        CHAR(50),  
    SALARY         REAL  
);
```

在这里，**NOT NULL** 表示列总是接受给定数据类型的显式值。这里有两个列我们没有使用 NOT NULL，这意味着这两个列可以为 NULL。

带有 NULL 值的字段在记录创建的时候可以保留为空。

### 实例

NULL 值在选择数据时会引起问题，因为当把一个未知的值与另一个值进行比较时，结果总是未知的，且不会包含在最后的結果中。假设有下面的表，COMPANY 的记录如下所示：

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

让我们使用 UPDATE 语句来设置一些允许空值的值为 NULL，如下所示：

```
sqlite> UPDATE COMPANY SET ADDRESS = NULL, SALARY = NULL where ID IN(6,7);
```

现在，COMPANY 表的记录如下所示：

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		
7	James	24		

接下来，让我们看看 **IS NOT NULL** 运算符的用法，它用来列出所有 SALARY 不为 NULL 的记录：

```
sqlite> SELECT ID, NAME, AGE, ADDRESS, SALARY
        FROM COMPANY
        WHERE SALARY IS NOT NULL;
```

上面的 SQLite 语句将产生下面的结果：

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

下面是 **IS NULL** 运算符的用法，将列出所有 SALARY 为 NULL 的记录：

```
sqlite> SELECT ID, NAME, AGE, ADDRESS, SALARY
        FROM COMPANY
        WHERE SALARY IS NULL;
```

上面的 SQLite 语句将产生下面的结果：

ID	NAME	AGE	ADDRESS	SALARY
-----	-----	-----	-----	-----
6	Kim	22		
7	James	24		

[← SQLite Unions 子句](#)

[SQLite 别名 →](#)

 点我分享笔记