

SQLite Join

SQLite 的 **Join** 子句用于结合两个或多个数据库中表的记录。JOIN 是一种通过共同值来结合两个表中字段的手段。

SQL 定义了三种主要类型的连接：

- 交叉连接 - CROSS JOIN
- 内连接 - INNER JOIN
- 外连接 - OUTER JOIN

在我们继续之前，让我们假设有两个表 COMPANY 和 DEPARTMENT。我们已经看到了用来填充 COMPANY 表的 INSERT 语句。现在让我们假设 COMPANY 表的记录列表如下：

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

另一个表是 DEPARTMENT，定义如下：

```
CREATE TABLE DEPARTMENT(  
  ID INT PRIMARY KEY      NOT NULL,  
  DEPT      CHAR(50) NOT NULL,  
  EMP_ID    INT      NOT NULL  
);
```

下面是填充 DEPARTMENT 表的 INSERT 语句：

```
INSERT INTO DEPARTMENT (ID, DEPT, EMP_ID)  
VALUES (1, 'IT Billing', 1 );  
  
INSERT INTO DEPARTMENT (ID, DEPT, EMP_ID)  
VALUES (2, 'Engineering', 2 );  
  
INSERT INTO DEPARTMENT (ID, DEPT, EMP_ID)  
VALUES (3, 'Finance', 7 );
```

最后，我们在 DEPARTMENT 表中有下列的记录列表：

ID	DEPT	EMP_ID
-----	-----	-----
1	IT Billing	1
2	Engineerin	2
3	Finance	7

交叉连接 - CROSS JOIN

交叉连接 (CROSS JOIN) 把第一个表的每一行与第二个表的每一行进行匹配。如果两个输入表分别有 x 和 y 行，则结果表有 x*y 行。由于交叉连接 (CROSS JOIN) 有可能产生非常大的表，使用时必须谨慎，只在适当的时候使用它们。

交叉连接的操作，它们都返回被连接的两个表所有数据行的笛卡尔积，返回到的数据行数等于第一个表中符合查询条件的数据行数乘以第二个表中符合查询条件的数据行数。

下面是交叉连接 (CROSS JOIN) 的语法：

```
SELECT ... FROM table1 CROSS JOIN table2 ...
```

基于上面的表，我们可以写一个交叉连接 (CROSS JOIN)，如下所示：

```
sqlite> SELECT EMP_ID, NAME, DEPT FROM COMPANY CROSS JOIN DEPARTMENT;
```

上面的查询会产生以下结果：

EMP_ID	NAME	DEPT
-----	-----	-----
1	Paul	IT Billing
2	Paul	Engineerin
7	Paul	Finance
1	Allen	IT Billing
2	Allen	Engineerin
7	Allen	Finance
1	Teddy	IT Billing
2	Teddy	Engineerin
7	Teddy	Finance
1	Mark	IT Billing
2	Mark	Engineerin
7	Mark	Finance
1	David	IT Billing
2	David	Engineerin
7	David	Finance
1	Kim	IT Billing

2	Kim	Engineerin
7	Kim	Finance
1	James	IT Billing
2	James	Engineerin
7	James	Finance

内连接 - INNER JOIN

内连接 (INNER JOIN) 根据连接谓词结合两个表 (table1 和 table2) 的列值来创建一个新的结果表。查询会把 table1 中的每一行与 table2 中的每一行进行比较，找到所有满足连接谓词的行的匹配对。当满足连接谓词时，A 和 B 行的每个匹配对的列值会合并成一个结果行。

内连接 (INNER JOIN) 是最常见的连接类型，是默认的连接类型。INNER 关键字是可选的。

下面是内连接 (INNER JOIN) 的语法：

```
SELECT ... FROM table1 [INNER] JOIN table2 ON conditional_expression ...
```

为了避免冗余，并保持较短的措辞，可以使用 **USING** 表达式声明内连接 (INNER JOIN) 条件。这个表达式指定一个或多个列的列表：

```
SELECT ... FROM table1 JOIN table2 USING ( column1 ,... ) ...
```

自然连接 (NATURAL JOIN) 类似于 **JOIN...USING**，只是它会自动测试存在两个表中的每一列的值之间相等值：

```
SELECT ... FROM table1 NATURAL JOIN table2...
```

基于上面的表，我们可以写一个内连接 (INNER JOIN)，如下所示：

```
sqlite> SELECT EMP_ID, NAME, DEPT FROM COMPANY INNER JOIN DEPARTMENT
        ON COMPANY.ID = DEPARTMENT.EMP_ID;
```

上面的查询会产生以下结果：

EMP_ID	NAME	DEPT
-----	-----	-----
1	Paul	IT Billing
2	Allen	Engineerin
7	James	Finance

外连接 - OUTER JOIN

外连接 (OUTER JOIN) 是内连接 (INNER JOIN) 的扩展。虽然 SQL 标准定义了三种类型的外连接：LEFT、RIGHT、FULL，但 SQLite 只支持 **左外连接 (LEFT OUTER JOIN)**。

外连接 (OUTER JOIN) 声明条件的方法与内连接 (INNER JOIN) 是相同的, 使用 ON、USING 或 NATURAL 关键字来表达。最初的结果表以相同的方式进行计算。一旦主连接计算完成, 外连接 (OUTER JOIN) 将从一个或两个表中任何未连接的行合并进来, 外连接的列使用 NULL 值, 将它们附加到结果表中。

下面是左外连接 (LEFT OUTER JOIN) 的语法:

```
SELECT ... FROM table1 LEFT OUTER JOIN table2 ON conditional_expression ...
```

为了避免冗余, 并保持较短的措辞, 可以使用 **USING** 表达式声明外连接 (OUTER JOIN) 条件。这个表达式指定一个或多个列的列表:

```
SELECT ... FROM table1 LEFT OUTER JOIN table2 USING ( column1 ,... ) ...
```

基于上面的表, 我们可以写一个外连接 (OUTER JOIN), 如下所示:

```
sqlite> SELECT EMP_ID, NAME, DEPT FROM COMPANY LEFT OUTER JOIN DEPARTMENT  
ON COMPANY.ID = DEPARTMENT.EMP_ID;
```

上面的查询会产生以下结果:

EMP_ID	NAME	DEPT
-----	-----	-----
1	Paul	IT Billing
2	Allen	Engineerin
	Teddy	
	Mark	
	David	
	Kim	
7	James	Finance

[← SQLite 约束](#)[SQLite Unions 子句 →](#)[✎ 点我分享笔记](#)