



链滴

# 左连接、右连接、内连接的区别

作者: [beejson](#)

原文链接: <https://ld246.com/article/1512974663823>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

假设有A, B两个表。

表A记录如下:

aID	aNum
1	a20050111
2	a20050112
3	a20050113
4	a20050114
5	a20050115

表B记录如下:

bID	bName
1	2006032401
2	2006032402
3	2006032403
4	2006032404
8	2006032408

---

1.left join

sql语句如下:

```
select * from A
```

```
left join B
```

```
on A.aID = B.bID
```

结果如下:

aID	aNum	bID	bName
1	a20050111	1	2006032401
2	a20050112	2	2006032402
3	a20050113	3	2006032403
4	a20050114	4	2006032404
5	a20050115	NULL	NULL

(所影响的行数为 5 行)

结果说明:

left join是以A表的记录为基础的,A可以看成左表,B可以看成右表,left join是以左表为准的.

换句话说,左表(A)的记录将会全部表示出来,而右表(B)只会显示符合搜索条件的记录(例子中为: A.aID = B.bID).

B表记录不足的地方均为NULL.

---

## 2.right join

sql语句如下:

```
select * from A
```

```
right join B
```

```
on A.aID = B.bID
```

结果如下:

aID	aNum	bID	bName
1	a20050111	1	2006032401
2	a20050112	2	2006032402
3	a20050113	3	2006032403
4	a20050114	4	2006032404
NULL	NULL	8	2006032408

(所影响的行数为 5 行)

结果说明:

仔细观察一下,就会发现,和left join的结果刚好相反,这次是以右表(B)为基础的,A表不足的地方用NULL充.

---

## 3.inner join

sql语句如下:

```
select * from A
```

```
innerjoin B
```

```
on A.aID = B.bID
```

结果如下:

aID	aNum	bID	bName
1	a20050111	1	2006032401
2	a20050112	2	2006032402
3	a20050113	3	2006032403
4	a20050114	4	2006032404

结果说明:

很明显,这里只显示出了 A.aID = B.bID的记录.这说明inner join并不以谁为基础,它只显示符合条件的记录.

---

PS:

LEFT JOIN操作用于在任何的 FROM 子句中, 组合来源表的记录。使用 LEFT JOIN 运算来创建一个左外部联接。左边外部联接将包含了从第一个 (左边) 开始的两个表中的所有记录, 即使在第二个 (边) 表中并没有相符值的记录。

语法: FROM table1 LEFT JOIN table2 ON table1.field1 compopr table2.field2

说明: table1, table2参数用于指定要将记录组合的表的名称。

field1, field2参数指定被联接的字段名称。且这些字段必须有相同的数据类型及包含相同类型的数, 但它们不需要有相同的名称。

compopr参数指定关系比较运算符: "=", "<", ">", "<=", ">=" 或 "<>"。

如果在INNER JOIN操作中要联接包含Memo 数据类型或 OLE Object 数据类型数据的字段, 将会发错误

注: left和right是外连接, Inner是内连接。