

## 图解MySQL各种连接

笔记本: 大数据

创建时间: 2021/1/5 15:40

URL: <https://blog.csdn.net/plg17/article/details/78758593>

更新时间: 2021/1/5 15:41

## 图解MySQL各种连接

用两个表 (a\_table、b\_table) , 关联字段a\_table.a\_id和b\_table.b\_id来演示一下MySQL的内连接、外连接 (左(外)连接、右(外)连接、全(外)连接)

MySQL版本: Server version: 5.6.31 MySQL Community Server (GPL)

数据库表: a\_table、b\_table

主题: 内连接、左连接 (左外连接) 、右连接 (右外连接) 、全连接 (全外连接)

## 前提

### 建表语句:

```
1 CREATE TABLE `a_table` (  
2   `a_id` int(11) DEFAULT NULL,  
3   `a_name` varchar(10) DEFAULT NULL,  
4   `a_part` varchar(10) DEFAULT NULL  
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

```
1 CREATE TABLE `b_table` (  
2   `b_id` int(11) DEFAULT NULL,  
3   `b_name` varchar(10) DEFAULT NULL,  
4   `b_part` varchar(10) DEFAULT NULL  
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

### 表测试数据:

	a_id	a_name	a_part
1	1	老潘	总裁部
2	2	老王	秘书部
3	3	老张	设计部
4	4	老李	运营部

	b_id	b_name	b_part
1	2	老王	秘书部
2	3	老张	设计部
3	5	老刘	人事部
4	6	老黄	生产部

## 一、内连接

关键字: inner join on

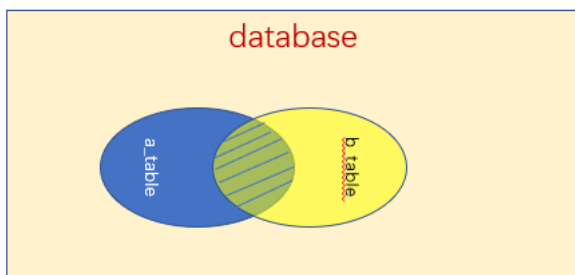
语句: select \* from a\_table a inner join b\_table b on a.a\_id = b.b\_id;

执行结果:

```
select * from a_table a inner join b_table b on a.a_id = b.b_id;
```

	a_id	a_name	a_part	b_id	b_name	b_part
1	2	老王	秘书部	2	老王	秘书部
2	3	老张	设计部	3	老张	设计部

说明：组合两个表中的记录，返回关联字段相符的记录，也就是返回两个表的交集（阴影）部分。



<http://blog.csdn.net/plg17>

## 二、左连接（左外连接）

关键字：left join on / left outer join on

语句：select \* from a\_table a left join b\_table b on a.a\_id = b.b\_id;

执行结果：

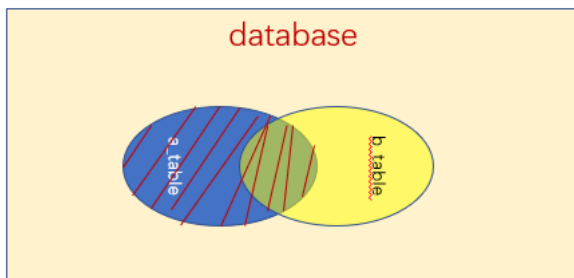
```
select * from a_table a left join b_table b on a.a_id = b.b_id;
```

	a_id	a_name	a_part	b_id	b_name	b_part
1	2	老王	秘书部	2	老王	秘书部
2	3	老张	设计部	3	老张	设计部
3	1	老潘	总裁部	[NULL]	[NULL]	[NULL]
4	4	老李	运营部	[NULL]	[NULL]	[NULL]

说明：

left join 是left outer join的简写，它的全称是左外连接，是外连接中的一种。

左(外)连接，左表(a\_table)的记录将会全部表示出来，而右表(b\_table)只会显示符合搜索条件的记录。右表记录不足的地方均为NULL。



<http://blog.csdn.net/plg17>

## 三、右连接（右外连接）

关键字：right join on / right outer join on

语句：select \* from a\_table a right outer join b\_table b on a.a\_id = b.b\_id;

执行结果：

```
select * from a_table a right outer join b_table b on a.a_id = b.b_id;
```

a\_table(+)

select \* from a\_table a right outer join b\_table b on a.a\_id = b | Enter a SQL expression to filter results (L

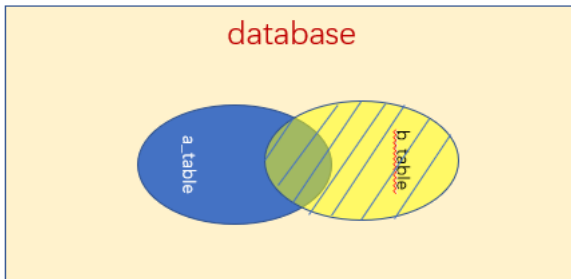
	a_id	a_name	a_part	b_id	b_name	b_part
1	2	老王	秘书部	2	老王	秘书部
2	3	老张	设计部	3	老张	设计部
3	[NULL]	[NULL]	[NULL]	5	老刘	人事部
4	[NULL]	[NULL]	[NULL]	6	老黄	生产部

<http://blog.csdn.net/plgl7>

说明：

right join是right outer join的简写，它的全称是右外连接，是外连接中的一种。

与左(外)连接相反，右(外)连接，左表(a\_table)只会显示符合搜索条件的记录，而右表(b\_table)的记录将会全部表示出来。左表记录不足的地方均为



<http://blog.csdn.net/plgl7>

## 四、全连接（全外连接）

MySQL目前不支持此种方式，可以用其他方式替代解决。

## 五、补充，MySQL如何执行关联查询

MySQL认为任何一个查询都是一次“关联”，并不仅仅是一个查询需要到两个表匹配才叫关联，所以在MySQL中，每一个查询，每一个片段（包括子查询）都可以是一次关联。

当前MySQL关联执行的策略很简单：MySQL对任何关联都执行嵌套循环关联操作，即MySQL先在一个表中循环取出单条数据，然后在嵌套循环到行，依次下去，直到找到所有表中匹配的行为止。然后根据各个表匹配的行，返回查询中需要的各个列。请看下面的例子中的简单的查询：

查询语句：select tbl1.col1, tbl2.col2 from tbl1 inner join tbl2 using(col3) where tbl1.col1 in (5, 6);

假设MySQL按照查询中的表顺序进行关联操作，我们则可以用下面的伪代码表示MySQL将如何完成这个查询：

```

1  outer_iter = iterator over tbl1 where col1 in (5, 6)
2  outer_row = outer_iter.next
3  while outer_row
4    inner_iter = iterator over tbl2 where col3 = outer_row.col3
5    inner_row = inner_iter.next
6    while inner_row
7      output [ outer_row.col1, inner_row.col2]
8    inner_row = inner_iter.next
9  end
10 outer_row = outer_iter.next
11 end

```

上面的执行计划对于单表查询和多表关联查询都适用，如果是一个单表查询，那么只需要上面外层的基本操作。对于外连接，上面的执行过程仍然的查询语句修改如下：

select tbl1.col1, tbl2.col2 from tbl1 left outer join tbl2 using(col3) where tbl1.col1 in (5, 6);

那么，对应的伪代码如下：

```

1  outer_iter = iterator over tbl1 where col1 in (5, 6)
2  outer_row = outer_iter.next
3  while outer_row
4    inner_iter = iterator over tbl2 where col3 = outer_row.col3
5    inner_row = inner_iter.next
6    if inner_row

```

```
7 | while inner_row      8 | output [ outer_row.col1, inner_row.col2]
9 | inner_row = inner_iter.next
10 | end
11 | else
12 | output [ outer_row.col1, null]
13 | end
14 | outer_row = outer_iter.next
15 | end
```

**说明：**第五部分摘自《高性能MySQL 第三版》