

# 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：

```
CREATE DATABASE joindb;
USE joindb;
```

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

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

```
INSERT INTO a_table (a_id, a_name, a_part)
VALUES('1', '老潘', '总裁部'),
      ('2', '老王', '秘书部'),
      ('3', '老张', '设计部'),
      ('4', '老李', '运营部');
INSERT INTO b_table (b_id, b_name, b_part)
VALUES('2', '老王', '秘书部'),
      ('3', '老张', '设计部'),
      ('5', '老刘', '人事部'),
      ('6', '老黄', '生产部');
```

### 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

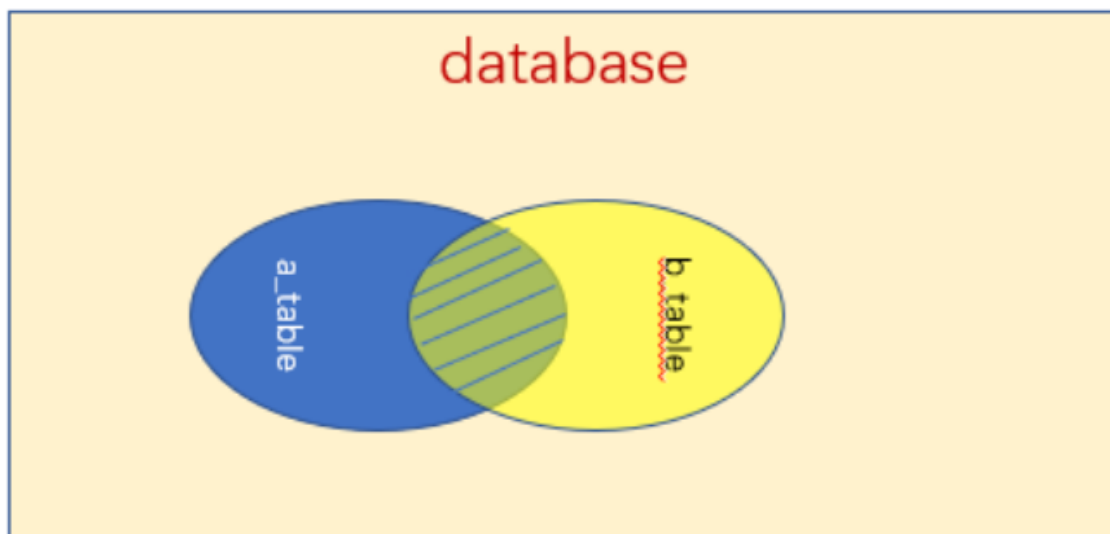
语句：

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

执行结果：

	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。









## 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;
```

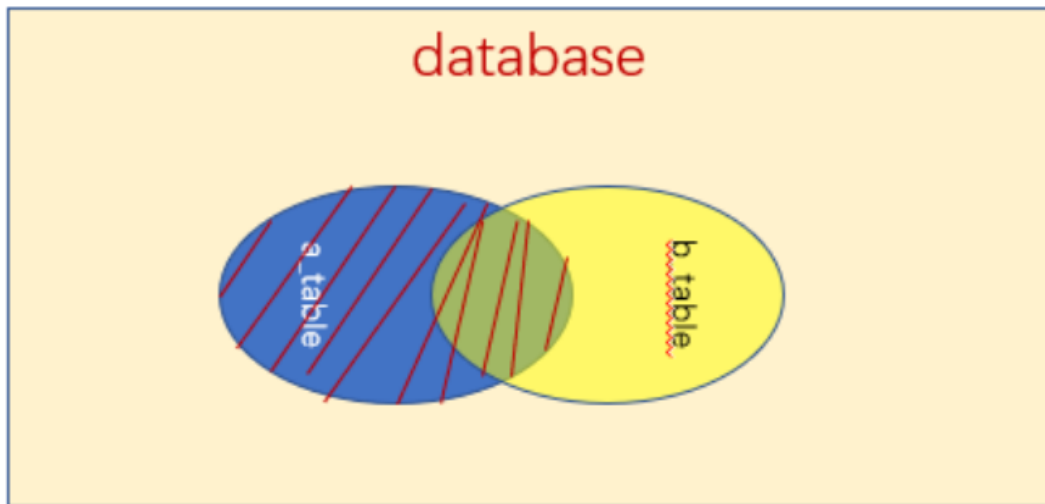
执行结果：

	 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。









### 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;
```

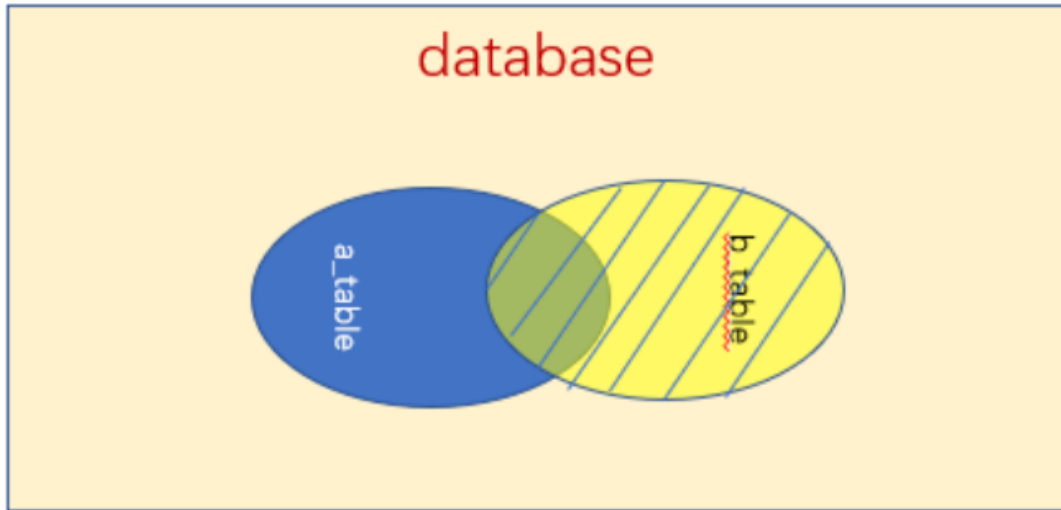
执行结果：

	 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

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NULL。



## 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。

```
select * from a_table a left join b_table b on a.a_id = b.b_id UNION select *  
from a_table a right outer 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>
5	<null>	<null>	<null>	5	老刘	人事部
6	<null>	<null>	<null>	6	老黄	生产部