



链滴

# 多表查询

作者: [chenyuan995](#)

原文链接: <https://ld246.com/article/1587214096855>

来源网站: [链滴](#)

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



- 查询语法:

```
select  
    列名列表  
from  
    表名列表  
where....
```

- 准备sql

创建部门表

```
CREATE TABLE dept(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    NAME VARCHAR(20)  
);
```

```
INSERT INTO dept (NAME) VALUES ('开发部'),('市场部'),('财务部');
```

创建员工表

```
CREATE TABLE emp (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    NAME VARCHAR(10),  
    gender CHAR(1), -- 性别  
    salary DOUBLE, -- 工资  
    join_date DATE, -- 入职日期  
    dept_id INT,  
    FOREIGN KEY (dept_id) REFERENCES dept(id) -- 外键, 关联部门表(部门表的主键)  
);  
INSERT INTO emp(NAME,gender,salary,join_date,dept_id) VALUES('孙悟空','男',7200,'2013-02-4',1);
```

```
INSERT INTO emp(NAME,gender,salary,join_date,dept_id) VALUES('猪八戒','男',3600,'2010-12-
```

```
2',2);
```

```
INSERT INTO emp(NAME,gender,salary,join_date,dept_id) VALUES('唐僧','男',9000,'2008-08-08'  
2);
```

```
INSERT INTO emp(NAME,gender,salary,join_date,dept_id) VALUES('白骨精','女',5000,'2015-10-  
7',3);
```

```
INSERT INTO emp(NAME,gender,salary,join_date,dept_id) VALUES('蜘蛛精','女',4500,'2011-03-  
4',1);
```

- 笛卡尔积：
  - 有两个集合A,B .取这两个集合的所有组成情况。
  - 要完成多表查询，需要消除无用的数据
- 多表查询的分类：
  - 内连接查询：
    - 隐式内连接：使用where条件消除无用数据
    - 例子：

```
-- 查询所有员工信息和对应的部门信息  
SELECT * FROM emp,dept WHERE emp.`dept_id` = dept.`id`;
```

```
-- 查询员工表的名称，性别。部门表的名称  
SELECT emp.name,emp.gender,dept.name FROM emp,dept WHERE emp.`dept_id` = dept.`id`;
```

```
SELECT  
  t1.name, -- 员工表的姓名  
  t1.gender,-- 员工表的性别  
  t2.name -- 部门表的名称  
FROM  
  emp t1,  
  dept t2  
WHERE  
  t1.`dept_id` = t2.`id`;
```

- 显式内连接：
  - 语法：  
select 字段列表 from 表名1 [inner] join 表名2 on 条件
  - 例如：

```
* SELECT * FROM emp INNER JOIN dept ON emp.`dept_id` = dept.`id`;
```

```
* SELECT * FROM emp JOIN dept ON emp.`dept_id` = dept.`id`;
```

-内连接查询：

- 从哪些表中查询数据

- 条件是什么
- 查询哪些字段
- 外链接查询:

- 左外连接:

- 语法:

select 字段列表 from 表1 left [outer] join 表2 on 条件;

- 查询的是左表所有数据以及其交集部分。
- 例子:

-- 查询所有员工信息, 如果员工有部门, 则查询部门名称, 没有部门, 则不显示部门名称  
SELECT t1.\*,t2.`name` FROM emp t1 LEFT JOIN dept t2 ON t1.`dept\_id` = t2.`id`;

- 右外连接:

- 语法:

select 字段列表 from 表1 right [outer] join 表2 on 条件;

- 查询的是右表所有数据以及其交集部分。
- 例子:

SELECT \* FROM dept t2 RIGHT JOIN emp t1 ON t1.`dept\_id` = t2.`id`;

- 子查询:

- 概念: 查询中嵌套查询, 称嵌套查询为子查询。
- 例子

-- 查询工资最高的员工信息

-- 1 查询最高的工资是多少 9000

SELECT MAX(salary) FROM emp;

-- 2 查询员工信息, 并且工资等于9000的

SELECT \* FROM emp WHERE emp.`salary` = 9000;

-- 一条sql就完成这个操作。子查询

SELECT \* FROM emp WHERE emp.`salary` = (SELECT MAX(salary) FROM emp);

- 子查询不同情况

- 子查询的结果是单行单列的:

- 子查询可以作为条件, 使用运算符去判断。运算符: > >= < <= =

-- 查询员工工资小于平均工资的人

SELECT \* FROM emp WHERE emp.salary < (SELECT AVG(salary) FROM emp);

- 子查询的结果是多行单列的:

- 子查询可以作为条件, 使用运算符in来判断

-- 查询'财务部'和'市场部'所有的员工信息

```
SELECT id FROM dept WHERE NAME = '财务部' OR NAME = '市场部';
```

```
SELECT * FROM emp WHERE dept_id = 3 OR dept_id = 2;
```

-- 子查询

```
SELECT * FROM emp WHERE dept_id IN (SELECT id FROM dept WHERE NAME = '财务部' OR  
AME = '市场部');
```

- 子查询的结果是多行多列的:

- 子查询可以作为一张虚拟表参与查询

-- 查询员工入职日期是2011-11-11日之后的员工信息和部门信息

-- 子查询

```
SELECT * FROM dept t1 ,(SELECT * FROM emp WHERE emp.`join_date` > '2011-11-11') t2 WH  
RE t1.id = t2.dept_id;
```

-- 普通内连接

```
SELECT * FROM emp t1,dept t2 WHERE t1.`dept_id` = t2.`id` AND t1.`join_date` > '2011-11-11'
```