第06章_多表查询

讲师：尚硅谷-宋红康（江湖人称：康师傅）

官网： http://www.atguigu.com

多表查询-1

```
【题目】
# 1.显示所有员工的姓名，部门号和部门名称。

# 2.查询90号部门员工的job_id和90号部门的location_id

# 3.选择所有有奖金的员工的 last_name , department_name , location_id , city

# 4.选择city在Toronto工作的员工的 last_name , job_id , department_id , department_name

# 5.查询员工所在的部门名称、部门地址、姓名、工作、工资，其中员工所在部门的部门名称为'Executive'

# 6.选择指定员工的姓名，员工号，以及他的管理者的姓名和员工号，结果类似于下面的格式
employees   Emp#    manager Mgr#
kochhar     101 king    100

# 7.查询哪些部门没有员工

# 8. 查询哪个城市没有部门

# 9. 查询部门名为 Sales 或 IT 的员工信息
```

1.显示所有员工的姓名，部门号和部门名称

```
SELECT last_name, e.department_id, department_name
FROM employees e
LEFT OUTER JOIN departments d
ON e.`department_id` = d.`department_id`;
```

2.查询90号部门员工的job_id和90号部门的location_id

```
SELECT job_id, location_id
FROM employees e, departments d
WHERE e.`department_id` = d.`department_id`
AND e.`department_id` = 90;
```

或

```
SELECT job_id, location_id
FROM employees e
JOIN departments d
ON e.`department_id` = d.`department_id`
WHERE e.`department_id` = 90;
```

3.选择所有有奖金的员工的 last_name , department_name , location_id , city

```
SELECT last_name , department_name , d.location_id , city
FROM employees e
LEFT OUTER JOIN departments d
ON e.`department_id` = d.`department_id`
LEFT OUTER JOIN locations l
ON d.`location_id` = l.`location_id`
WHERE commission_pct IS NOT NULL;
```

4.选择city在Toronto工作的员工的 last_name , job_id , department_id , department_name

```
SELECT last_name , job_id , e.department_id , department_name
FROM employees e, departments d, locations l
WHERE e.`department_id` = d.`department_id`
AND d.`location_id` = l.`location_id`
AND city = 'Toronto';
```

或

```
SELECT last_name , job_id , e.department_id , department_name
FROM employees e
JOIN departments d
ON e.`department_id` = d.`department_id`
JOIN locations l
ON l.`location_id` = d.`location_id`
WHERE l.`city` = 'Toronto';
```

5．查询员工所在的部门名称、部门地址、姓名、工作、工资，其中员工所在部门的部门名称为’Executive’

```
SELECT department_name, street_address, last_name, job_id, salary
FROM employees e JOIN departments d
ON e.department_id = d.department_id
JOIN locations l
ON d.`location_id` = l.`location_id`
WHERE department_name = 'Executive'
```

6. 选择指定员工的姓名，员工号，以及他的管理者的姓名和员工号，结果类似于下面的格式

```
employees    Emp#      manager      Mgr#
kochhar      101        king        100
```

```
SELECT emp.last_name employees, emp.employee_id "Emp#", mgr.last_name manager,
mgr.employee_id "Mgr#"
FROM employees emp
LEFT OUTER JOIN employees mgr
ON emp.manager_id = mgr.employee_id;
```

7. 查询哪些部门没有员工

```
#方式1：
SELECT d.department_id
FROM departments d LEFT JOIN employees e
ON e.department_id = d.`department_id`
WHERE e.department_id IS NULL

#方式2：
SELECT department_id
FROM departments d
WHERE NOT EXISTS (
        SELECT *
        FROM employees e
        WHERE e.`department_id` = d.`department_id`
         )
```

8. 查询哪个城市没有部门

```
SELECT l.location_id,l.city
FROM locations l LEFT JOIN departments d
ON l.`location_id` = d.`location_id`
WHERE d.`location_id` IS NULL
```

9. 查询部门名为 Sales 或 IT 的员工信息

```
SELECT employee_id,last_name,department_name
FROM employees e,departments d
WHERE e.department_id = d.`department_id`
AND d.`department_name` IN ('Sales','IT');
```

多表查询-2

```
储备：建表操作：
CREATE TABLE `t_dept` (
 `id` INT(11) NOT NULL AUTO_INCREMENT,
 `deptName` VARCHAR(30) DEFAULT NULL,
 `address` VARCHAR(40) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=INNODB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE `t_emp` (
 `id` INT(11) NOT NULL AUTO_INCREMENT,
 `name` VARCHAR(20) DEFAULT NULL,
  `age` INT(3) DEFAULT NULL,
 `deptId` INT(11) DEFAULT NULL,
empno int  not null,
 PRIMARY KEY (`id`),
 KEY `idx_dept_id` (`deptId`)
 #CONSTRAINT `fk_dept_id` FOREIGN KEY (`deptId`) REFERENCES `t_dept` (`id`)
) ENGINE=INNODB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

```
INSERT INTO t_dept(deptName,address) VALUES('华山','华山');
INSERT INTO t_dept(deptName,address) VALUES('丐帮','洛阳');
INSERT INTO t_dept(deptName,address) VALUES('峨眉','峨眉山');
INSERT INTO t_dept(deptName,address) VALUES('武当','武当山');
INSERT INTO t_dept(deptName,address) VALUES('明教','光明顶');
INSERT INTO t_dept(deptName,address) VALUES('少林','少林寺');
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('风清扬',90,1,100001);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('岳不群',50,1,100002);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('令狐冲',24,1,100003);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('洪七公',70,2,100004);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('乔峰',35,2,100005);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('灭绝师太',70,3,100006);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('周芷若',20,3,100007);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('张三丰',100,4,100008);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('张无忌',25,5,100009);
INSERT INTO t_emp(NAME,age,deptId,empno) VALUES('韦小宝',18,null,100010);

【题目】
#1.所有有门派的人员信息
（ A、B两表共有）

#2.列出所有用户，并显示其机构信息
 （A的全集）

#3.列出所有门派
（B的全集）

#4.所有不入门派的人员
（A的独有）

#5.所有没人入的门派
（B的独有）

#6.列出所有人员和机构的对照关系
(AB全有)
#MySQL Full Join的实现 因为MySQL不支持FULL JOIN,下面是替代方法
#left join + union(可去除重复数据)+ right join

#7.列出所有没入派的人员和没人入的门派
（A的独有+B的独有）
```

1. 所有有门派的人员信息

（ A、B两表共有）

```
select *
from t_emp a inner join t_dept b
on a.deptId = b.id;
```

2. 列出所有用户，并显示其机构信息

(A的全集)

```
select *
from t_emp a left join t_dept b
on a.deptId = b.id;
```

3. 列出所有门派

(B的全集)

```
select *
from  t_dept  b;
```

4. 所有不入门派的人员

(A的独有)

```
select *
from t_emp a left join t_dept b
on a.deptId = b.id
where b.id is null;
```

5. 所有没人入的门派

(B的独有)

```
select *
from t_dept b left join  t_emp a
on a.deptId = b.id
where a.deptId is null;
```

6. 列出所有人员和机构的对照关系

(AB全有)

```
#MySQL Full Join的实现 因为MySQL不支持FULL JOIN,下面是替代方法
#left join + union(可去除重复数据)+ right join

SELECT *
FROM t_emp A LEFT JOIN t_dept B
ON A.deptId = B.id
UNION
SELECT *
FROM t_emp A RIGHT JOIN t_dept B
ON A.deptId = B.id
```

7. 列出所有没入派的人员和没人入的门派

(A的独有+B的独有)

```sql
SELECT *
FROM t_emp A LEFT JOIN t_dept B
ON A.deptId = B.id
WHERE B.`id` IS NULL
UNION
SELECT *
FROM t_emp A RIGHT JOIN t_dept B
ON A.deptId = B.id
WHERE A.`deptId` IS NULL;
```