2019-3-28

MySQL多表查询之外键、表连接

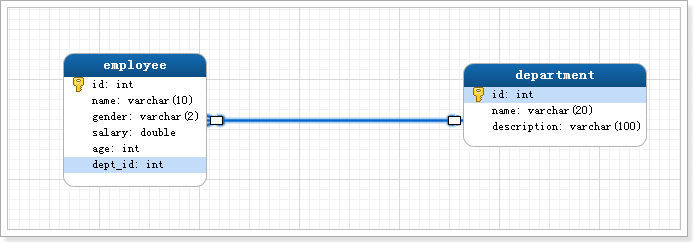
讲师:BruceLiu

# 什么是外键？

* **主键Primary Key**：是唯一标识一条记录，不能有重复的，**不允许为空**，用来保证数据完整性
* **外键**： 是另一表的主键, 外键可以有**重复**的, 可以是空值，用来和其他表建立联系用的。所以说，如果谈到了外键，一定是至少涉及到两张表。

**外键的作用：建立表之间的关系，保证数据库的完整性**

例如下面这两张表：

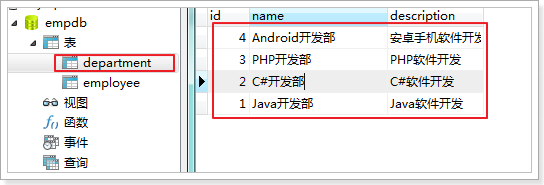


上面有两张表：部门表（dept）、员工表(emp)。Id=Dept\_id，而Dept\_id就是员工表中的外键：因为员工表中的员工需要知道自己属于哪个部门，就可以通过外键Dept\_id找到对应的部门，然后才能找到部门表里的各种字段信息，从而让二者相关联。所以说，外键一定是在从表中创建，从而找到与主表之间的联系；从表负责维护二者之间的关系。



|  |
| --- |
| create table department(  id int primary key,  name varchar(**20**) not null,  description varchar(**100**)  );  create table employee(  id int primary key,  name varchar(**10**) not null,  gender varchar(**2**) not null,  salary float(**10**,**2**),  age int(**2**),  dept\_id int  ); |

* 部门表插入数据



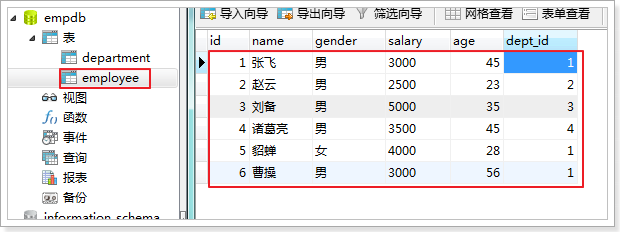
INSERT INTO `department` VALUES ('1', 'Java开发部', 'Java软件开发');

INSERT INTO `department` VALUES ('2', 'C#开发部', 'C#软件开发');

INSERT INTO `department` VALUES ('3', 'PHP开发部', 'PHP软件开发');

INSERT INTO `department` VALUES ('4', 'Android开发部', '安卓手机软件开发');

* 部门表插入数据



INSERT INTO `employee` VALUES ('1', '张飞', '男', '3000', '45', '1');

INSERT INTO `employee` VALUES ('2', '赵云', '男', '2500', '23', '2');

INSERT INTO `employee` VALUES ('3', '刘备', '男', '5000', '35', '3');

INSERT INTO `employee` VALUES ('4', '诸葛亮', '男', '3500', '45', '4');

INSERT INTO `employee` VALUES ('5', '貂蝉', '女', '4000', '28', '1');

INSERT INTO `employee` VALUES ('6', '曹操', '男', '3000', '56', '1');

# 添加外键的方式

我们在第一段中内容中已经将表建好了，数据也填充完了，现在来给从表（员工表）添加外键，让它与主表（部门表）相关联。代码举例如下：

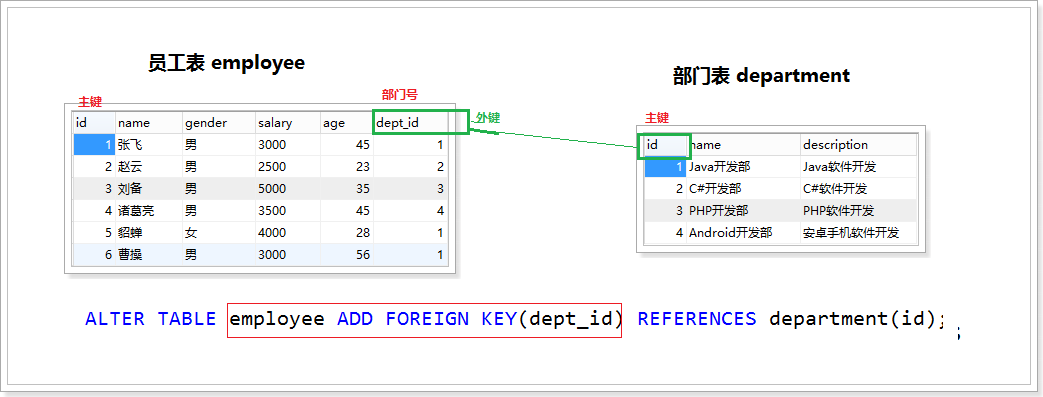
ALTER TABLE employee ADD **FOREIGN KEY**(dept\_id) REFERENCES department(id);

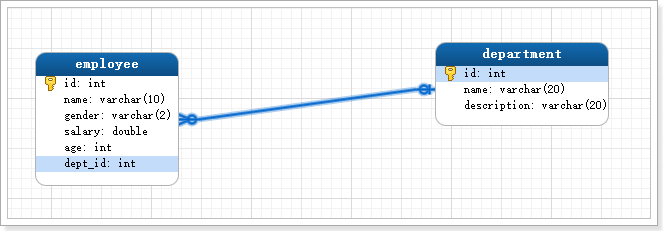
代码解释：

ALTER TABLE employee：在从表employee中进行操作；

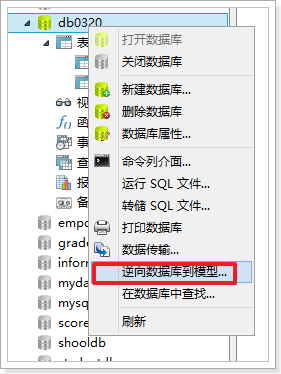
ADD FOREIGN KEY(dept\_id)：将从表的字段dept\_id添加为外键；

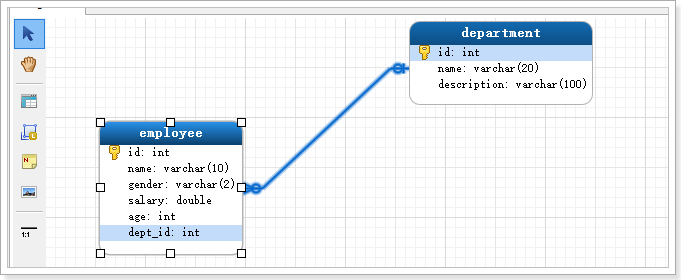
REFERENCES department(id)：映射到主表department当中为id的字段。





# 检测外键是否添加成功

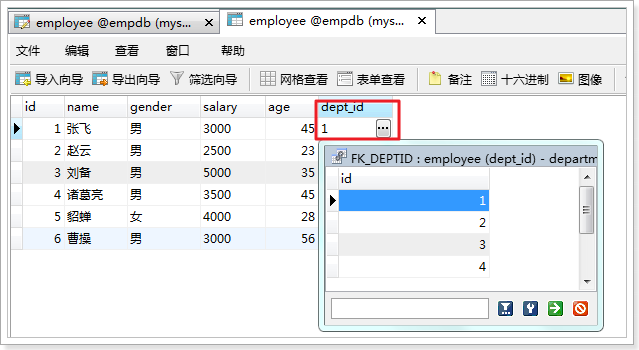




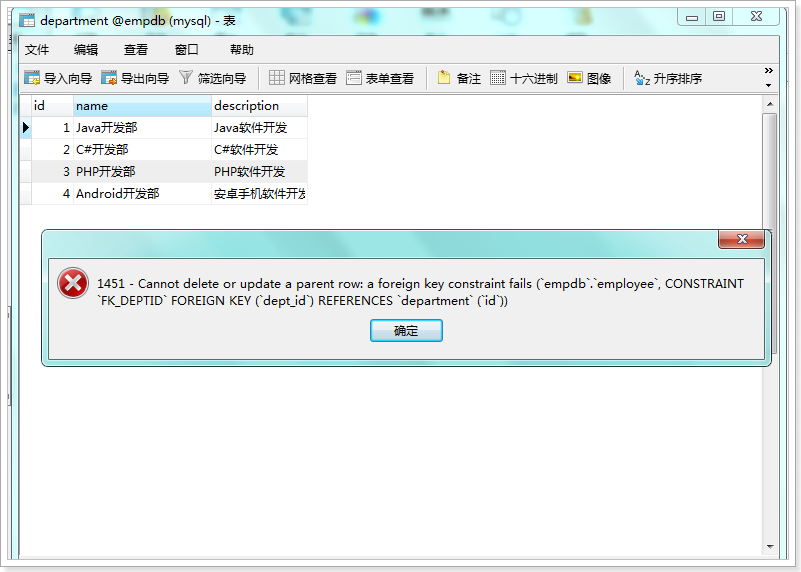
* 运行上方代码后，我们通过navicat来看一下外键有没有添加成功
* 上图中，选中表employee，单击红框部分的“设计表”按钮，界面如下：

RESTRICT(限制)：如果你想删除的那个主表，它的下面有对应从表的记录，此主表将无法删除。

CASCADE（级联）：如果主表的记录删掉，则从表中相关联的记录都将被删掉。



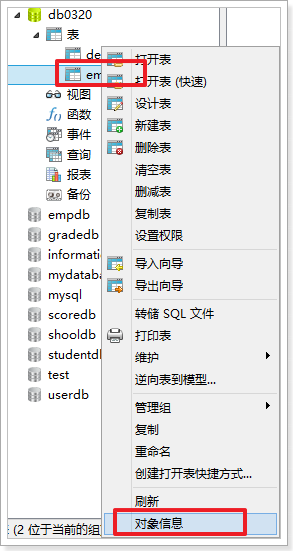
* 然后我们回到主表（部门表），此时如果想删除id为1的宣传部，会弹出如下提示：（因为外键的默认规则为RESTRICT）

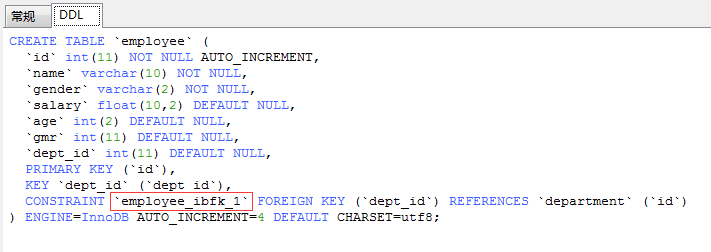


# 删除外键

我们在navicat中可以通过图形界面的方式删除外键，也可以通过sql语句来删除。

(1)获取外键名：

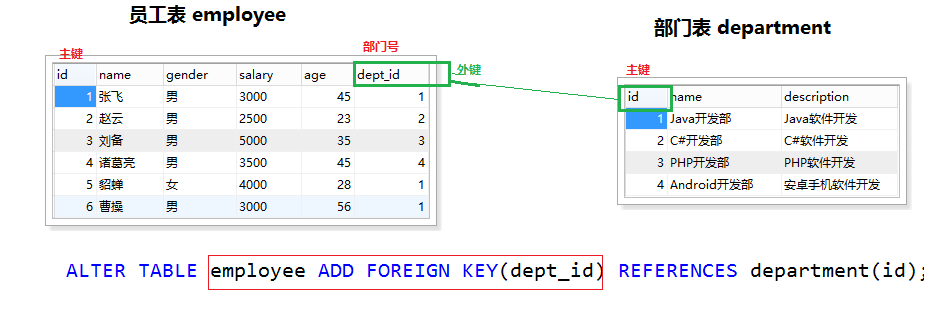




（2）删除外键：

alter table emp drop foreign key 外键名;

# 表连接（join）



## 5.1内连接（重点）

**1、内连接：只列出匹配的记录**

语法：

SELECT … FROM join\_table1

**INNER JOIN** join\_table2

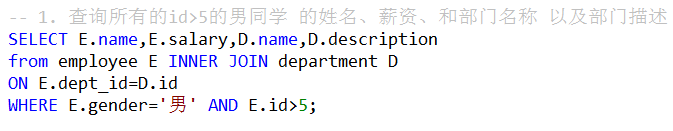
[ON join\_condition]

WHERE where\_definition

解释：只列出这些连接表中与连接条件相匹配的数据行。INNER可以不写，则默认为内连接。[ON join\_condition]里面写的是连接的条件。

举例：

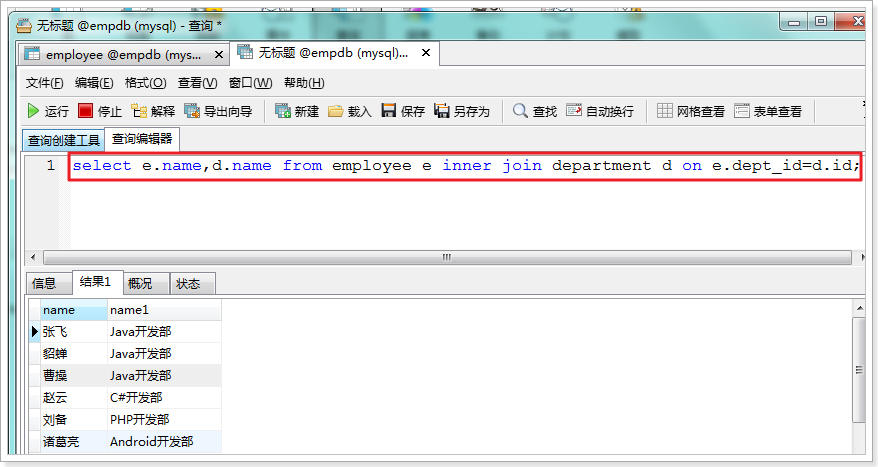
select e.name,d.name from employee e inner join department d on e.dept\_id=d.id;

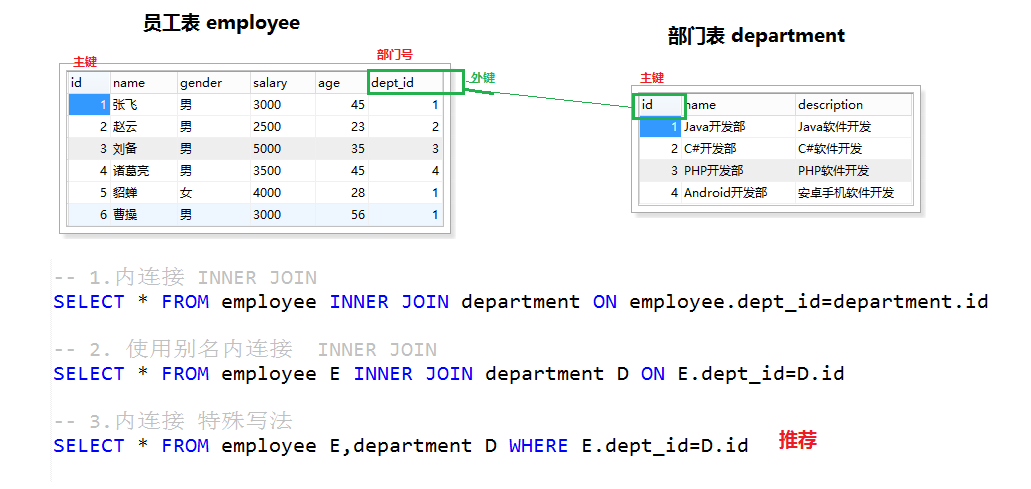


等价于（了解）：

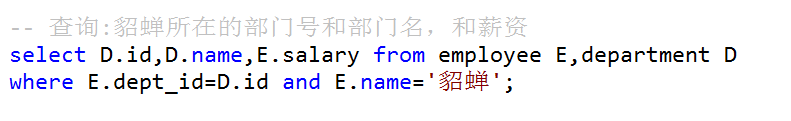
select e.name,d.name from employee e,department d where e.dept\_id=d.id;

运行效果：





备注：在写多表查询的时候，尽量的为每张表取别名，避免字段冲突，使用别名可以区别不同表中的字段



## 外连接（了解）

外连接分类：

* 左外连接(LEFT JOIN) -- LEFT JOIN
* 右外连接(RIGHT JOIN) -- RIGHT LOIN
* 全外连接(FULL JOIN) 注：MySQL5.1的版本暂不支持

语法：

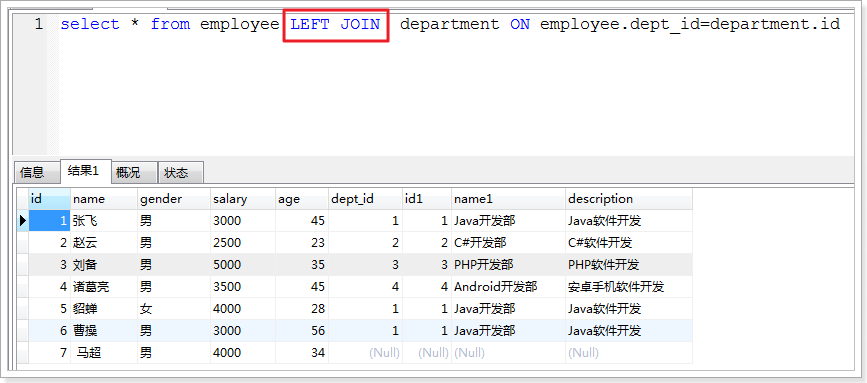
|  |
| --- |
| SELECT … FROM join\_table1  (LEFT | RIGHT ) JOIN join\_table2  ON join\_condition  WHERE where\_definition |

**解释**：

不仅列出与连接条件（on）相匹配的行，还列出左表table1(左外连接)、或右表table2(右外连接)、或两个表(全外连接)中所有符合WHERE过滤条件的数据行。一般都是用左连接或者外连接。

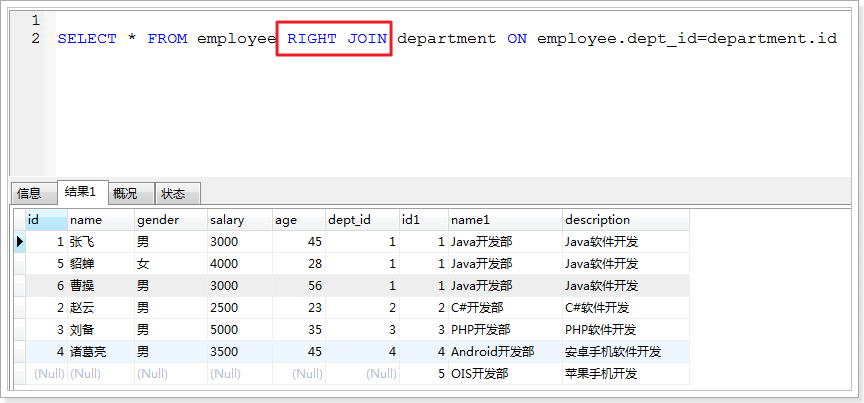
### 左外连接

**左外连接：左表列出全部，右表只列出匹配的记录**



### 右外连接

**右外连接：右表列出全部，左表只列出匹配的记录**



## 交叉连接(笛卡尔积) 了解

语法：

SELECT … FROM join\_table1 **CROSS JOIN** join\_table2;

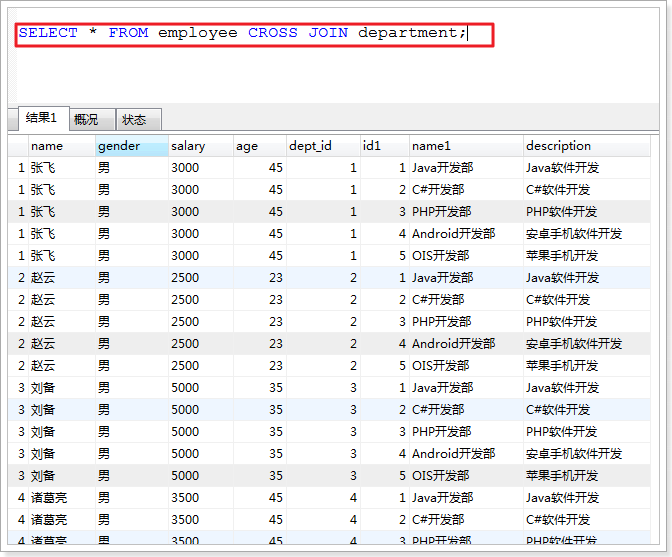
没有ON子句和WHERE子句，它返回的是连接表中所有数据行的**笛卡尔积**。

笛卡尔积举例：

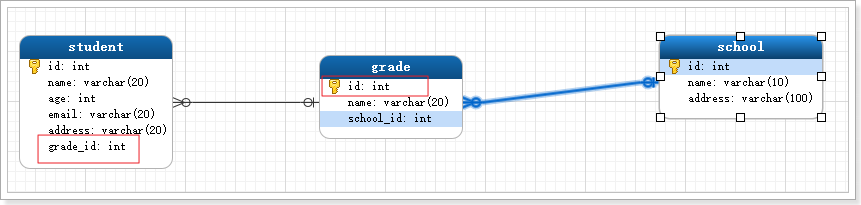
假设集合A={a,b}，集合B={0,1,2}，

则两个集合的笛卡尔积为{(a,0),(a,1),(a,2),(b,0),(b,1), (b,2)}

其结果集合中的数据行数等于第一个表中符合查询条件的数据行数乘以第二个表中符合查询条件的数据行数。



# 多表连接



查询语句：

* + 查询id>3 的学生姓名，邮箱,所属年级名，和所在的学校名

