**第8章\_联接查询**

**1. 什么是联接查询**

联接查询，就是将多个表连接起来进行查询，相比较子查询（也是可以在多个表里面查询数据），更加简单，容易理解。

**2. 外键**

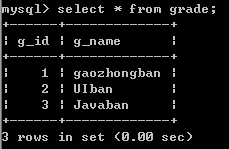
在实际开发项目的时候，一个健壮的数据库一定要有很好的参照性。

例如：有学生表和成绩表两张表，学生表里面有个学生叫张三，成绩表里面有张三的成绩。这个时候，如果你删除学生表里面的张三，那么成绩表中张三的的成绩就会沦为垃圾数据。所以，为了保证数据的完整性，我们就需要在两张表之间建立一定的联系（外键约束）。

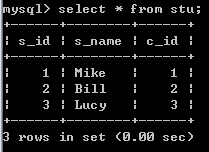
① 什么是外键？

外键是指一个表里面的一个字段去引用另一个表里面的一个字段，另一个表里面的字段往往是主键字段。

班级表（班级编号（主键），班级名称），有如下表数据：



学生表（学生编号（主键），学生姓名，所属班级（外键））



上面两个表中，c\_id这个字段就是外键，因为它是引用的grade表里面的g\_id字段。

被引用的表，我们称之为主表，在这里，grade表就是主表

引用其他表里面主键字段的表，我们称之为从表，在这里，stu表就是从表。

我们通过外键的关系，就让两张表产生了联系。

② 外键约束

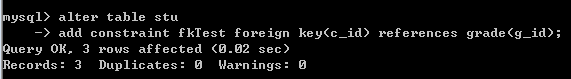
虽然我们两张表有引用关系，但是系统并不知道两张表是有关系，这个时候，我们就需要为其添加外键约束

外键约束创建语法：

alter table 表名

add constraint 外建名 foreign key（外键字段名） references 外表表名（主键字段名）

举例：



上面我们就创建了一个外键约束，外键约束的名字叫做fkTest，它是建立在stu表上面的，它是c\_id参照grade表里面的g\_id

接下来我们插入一条数据，如下：（会报错，因为我们设置了外键约束）

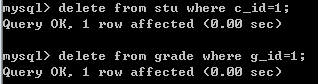


当我们创建了外键约束以后，就说明我们的两张表有了一定的联系，换句话说，绑在了一起，所以在删除数据的时候也要遵循一定规则，先删除从表里面的数据，在删除主表里面的数据，如下：



删除失败，因为在stu表里面还存在引用grade表中g\_id为1的实体

所以，我们要首先删除从表（stu）中c\_id=1的实体，然后才能删除grade表中的g\_id=1的数据。



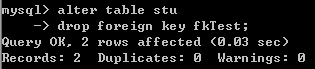
③ 删除外键

在实际开发中，根据业务逻辑要求，有时候需要解除两个表的关系，就需要删除外键。

语法：alter table 表名

drop foreign key 外建名；

举例：



**3. 操作关联表**

我们两个表之间存在着3种关系：一对一，一对多，多对多

一对多：这也是我们生活中比较常见的一种关系。比如员工与部门的关系

建立外键：我们可以在多的一方建立外键保存另一方的主键

多对多：比如选课的时候，一个学生可以选择多门课程，一门课程也可以被多个学生选择。

建立外键：新建一张第三方的表，来保存两张表的主键作为外键

一对一：在实际生活中比较常见，例如一个人只能对应一个身份证号

建立外键：可以在任意一方建立外键。

在以下情况下，我们会用到1对1的关系

1. 分割很多字段的时候

2. 由于安全原因要隔离表的一部分

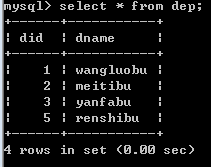
3. 保存临时数据的时候，在不需要的时候毫不费力的就可以删除掉

**4. 联接（连接）查询**

联接查询如果分类的话，可以分为3类：内连接，外连接，自连接，复合条件连接查询

有两张表，部门表和员工表你，表的数据信息如下：

部门表 员工表

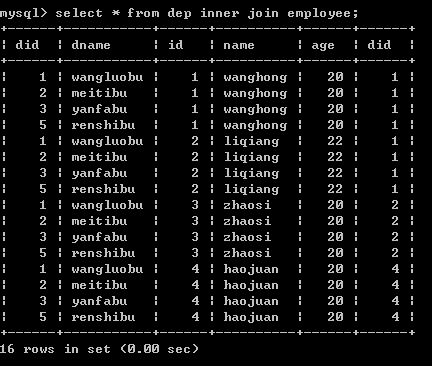
 

① 内连接

内连接使用的关键字为 inner join

语法：select \* from 表1 inner join 表二（将表1和表2连接起来了）

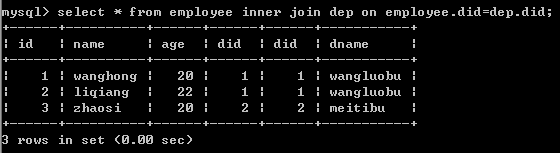
当我们将两个表内连接之后，出现了下面这个毫无用处的东西



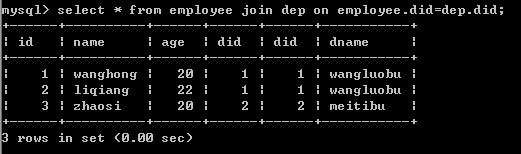
上表中的这个结果叫做笛卡尔乘积（mysql的特殊处理机制，其他没有）

为了得到我们想要的信息，我们必须排除重复的数据，这个时候我们关键字on

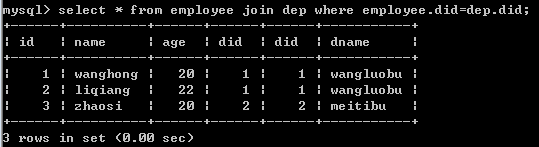
select \* from 表1 inner join 表2 on 表1.关系字段=表2.关系字段



在我们的MySQL里面，默认的就是内连接，所以你可以将inner给省略



on关键字可以用where代替，如下



同样也得到了相同的结果，但是on和where却有着本质上的不同

on：在内联之前就进行了过滤，所以没有产生笛卡尔乘积

where：是过滤条件，它是在两张表内联产生了笛卡尔乘积之后再进行过滤

所以on的效率高于where

补充：cross join（交叉连接）

在我们的MySQL里面，cross join 等价于 inner join，但是也仅仅只限于在My SQL中，如果是其他的RDBMS，两者是不一样，当你要做内连接，，就必须用inner join，要做笛卡尔乘积的话，就必须用cross join

② 外联接

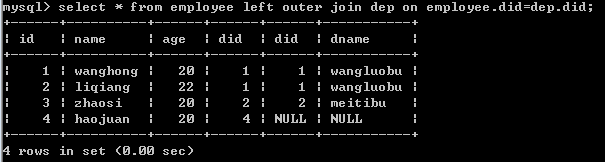
外联接分为三种：左外联接，右外连接，全外联接（在我们的MySQL不支持全外连接）

外联接必须要写关联条件，否则会报错，内联接不写的话默认就是笛卡尔乘积

左外联接：就是左表的所有数据都显示出来，不管有没有和右表匹配上

左外联接使用关键字 left outer join

select \* from 表1 left outer join 表2；



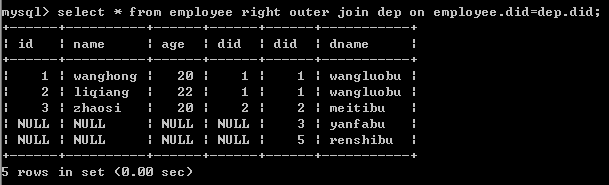
所谓左表右表，是根据在join的位置而定的，在join左边的就是左表，在join后边的右表

右联接：刚好就是和左联接相反

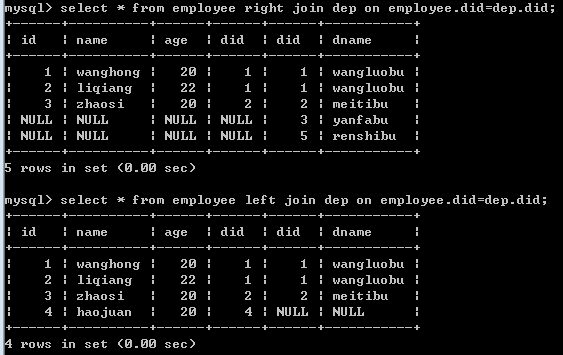
就是右表的数据全部显示出来，不管有没有和左边匹配上

右外联接使用关键字 right outer join

select \* from 表1 right outer join 表2；



补充：我们的outer关键字是可以省略的



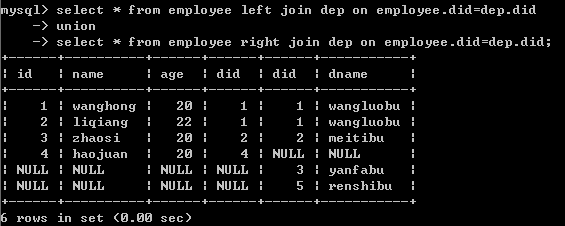
全外连接：就是无论左表或者右表有没有关联上，都显示出来

关键字 full [outer] join

但是，我们的MySQL是不支持全外连接

有一个方法可以模拟全外连接

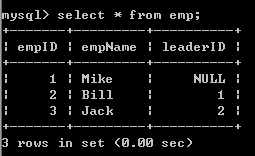
使用关键字union



③ 自连接

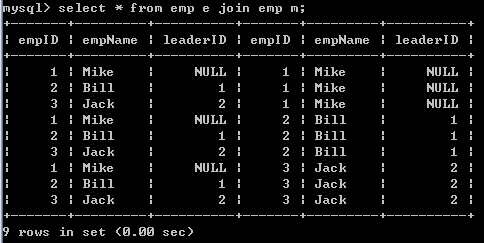
所谓自连接其实是内连接或外联接的一种特例。自连接所连接的表来源于同一张表

现在有如下表数据信息：



查询每位员工编号，员工姓名以及领导姓名

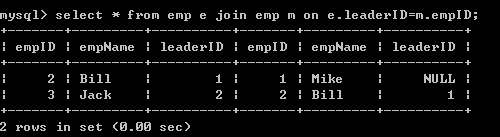
分析：首先需要用到自连接，说白了就是自己连接自己



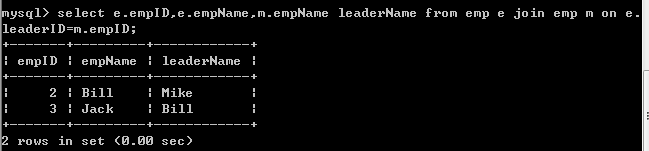
注意点：必须给别名

连接出来之后是一个笛卡尔乘积

接下来我们需要过滤掉不用的信息



只需要我们需要的字段，不要字段就不写了



复合条件连接查询

其实就是指在连接查询的基础上加上一个过滤条件或者一些排序，限制等等

查询所有的员工信息和所在部门，并且按照年龄进行降序排列

select e.id,e.name,e.age,dep.dname d\_name from employee e left join dep on e.did=dep.did order by e.age desc;

