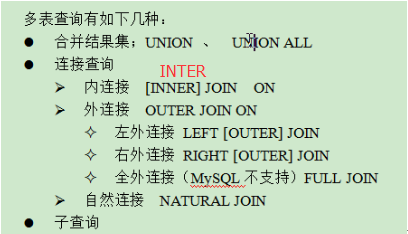
MySQL中的多表查询（合并结果集UNION、连接查询JOIN、子查询）

# 多表查询

## 合并结果集：UNION (去重)、UNION ALL （不去重）

## 连接查询：JOIN ON 内连接、外连接。

## 子查询 :



# 合并结果集（UNION）(简单)

## 作用：合并结果集就是把两个select语句的查询结果合并到一起。

## 要求：被合并的两个结果：列数、列类型必须相同，列名可以不同。

## 合并结果集的两种方式：

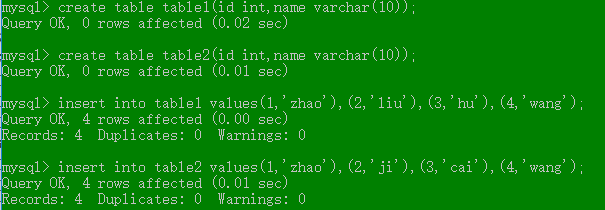
### UNION ：去除重复记录。

select id,name from user1 **UNION** select id,name from user2;

### UNION ALL：列出各自的所有记录，不去除重复记录。

select id,name from user1 **UNION ALL** select id,name from user2;

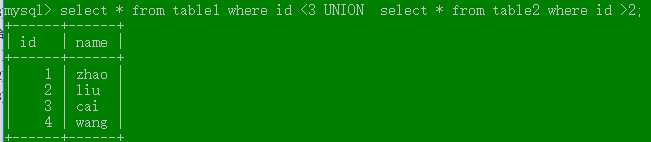
示例：



insert的时候，可以同时插入多条记录。**（批量插入）**





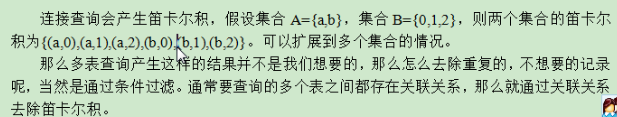


# 连接查询（非常重要）（JOIN）

## 内连接查询 –99查询法（不是标准的SQL语句）



**笛卡尔积：**





字段表示：**表名.field**

解决办法：使用**主外键关系**作为条件**去除无用的信息**。







可以通过**起别名**，简化查询语句：内连接--99查询法





## 内连接查询—新的查询语句 [INTER] JOIN –ON（标准的SQL语句）

语法：

**SELECT \* FROM table1 t1 [INTER] JOIN table2 t2 ON t1.id=t2.id [where ….];**

注意：

### 为了方便，一般需要给表名起简单的别名，如t1、t2。

### ON 只可以用于主外键关联，对于不是主外键的，需要使用WHERE进行选择。



### MySQL默认的就是内连接查询，所以INTER可以省略。



等价于

SELECT s.stuid,s.stuname,c,score FROM student s [**INTER] JOIN** score c **ON** s.stuid =c.stuid where score>70;//将**逗号**用[**INTER] JOIN替换，用ON替换WHERE**

## 外连接查询（重要）

**SELECT \* FROM table1 t1 LEFT/RIGHT [OUTER] JOIN table2 t2 ON t1.id=t2.id [where ….];**

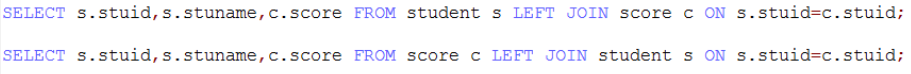
**相对于内连接查询的区别：就是把参照表格中独有的数据也会列出来。**

**本质理解：以左连接查询为例，**首先把**左边的表格**为参照物，然后依次从左边的表格中取出stuid来，去右边的表格中寻找stuid中是否有这个值，如果有，则合并列出来，否则设置为NULL，也会列出来。

**左连接就是以ON左边的表格为参照，右连接就是以ON右边的表格为参照。被参照的表格的所有元素都会被列出来，而未被参照的元素的表格中独有的元素不会被列出来。**

### 左外连接查询

参照左边的表格，**左边表格中满足ON条件的记录和左边表格独有的记录，都会查询出来**，但是右边表格独有的元素不会被列出来。



上面的语句可以理解为：

### 右外连接查询



## 内外连接查询区分

**内连接查询： (inter) join... on... (MySQL默认采用内连接)**

**外连接查询： left (outer) join... on...**

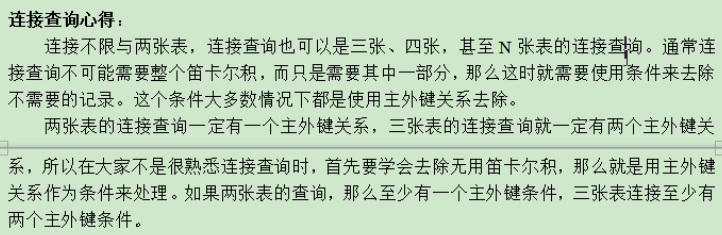
**right (outer) join... on...**

**省略后：**

**join... on... 内连接**

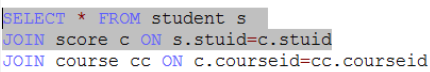
**left join... on... 左外连接**

**right join... on... 右外连接**



**多表查询：**



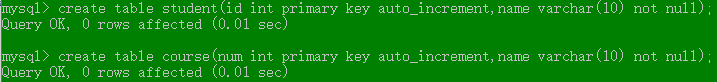


多表查询的过程中：

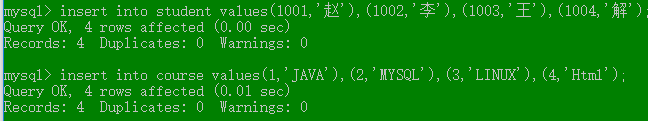
**可以先把两个表查询后的结果当成是一个表，再根据另外一个表进行连接。**

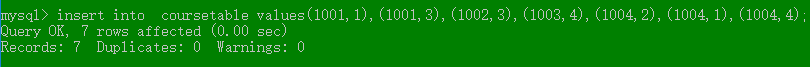
**n个表连接查询，一般需要n-1个条件。**

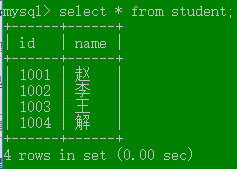
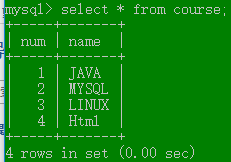
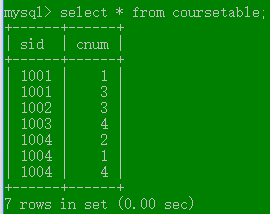
**示例:**







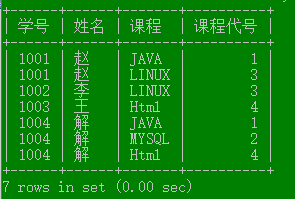


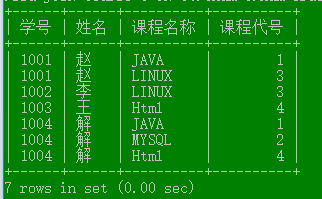
三个表格：1. 学号姓名表，2.科目代号表，3. 学生选课表。

表1与表2表面上 没有直接联系，但是经过表3，可看出表1与表2是多对多的关系。

**select s.id '学号',s.name '姓名',c.name '课程',c.num '课程代号' from student s,course c,coursetable ct where s.id = ct.sid and ct.cnum=c.num order by s.id;**



**select s.id '学号',s.name '姓名',c.name '课程名称',c.num '课程代号' from student s JOIN coursetable ct ON s.id=ct.sid JOIN course c ON ct.cnum=c.num order by s.id;**

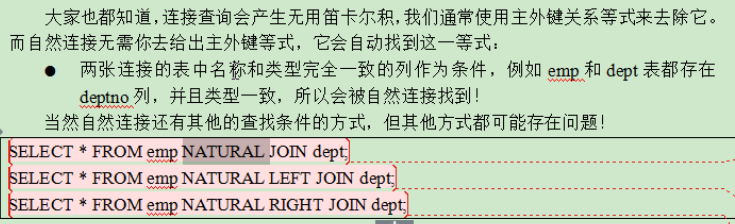


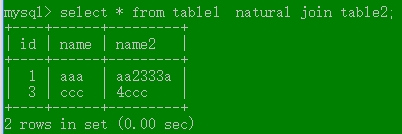
### 全外连接查询

用的不是很多。

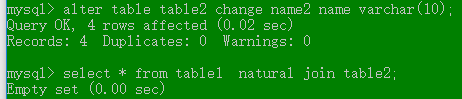
## 自然连接：关键字NATURAL

**自动寻找相同列名，进行筛选**。



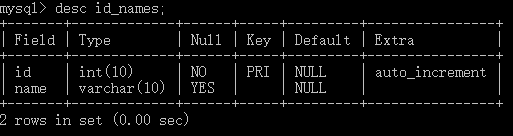


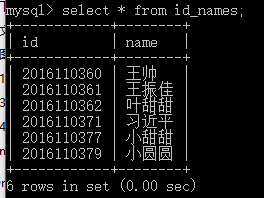
自动以id为选择条件，只看列名和数据类型，但是两个表之间只可以有一个列的列名与数据类型一致，否则，就是以此两类进行选择。

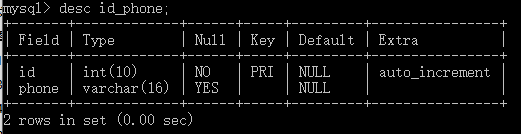


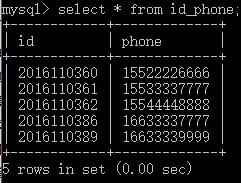
## 连接查询示例：

### 两个表格









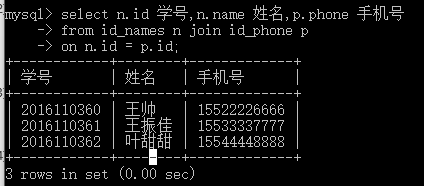
### 99查询法：笛卡尔积

select n.id 学号,n.name 姓名,p.phone 手机号from id\_names n,id\_phone p where n.id = p.id;



### 内连接查询

select n.id 学号,n.name 姓名,p.phone 手机号from id\_names n join id\_phone p on n.id = p.id;

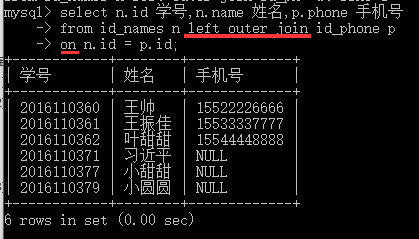


### 外连接查询

左外连接与对换表格的前后顺序是一致的，所以一般利用左连接加对换表格前后位置即可满足需求。

**select n.id 学号,n.name 姓名,p.phone 手机号 from id\_names n left outer join id\_phone p on n.id = p.id;**

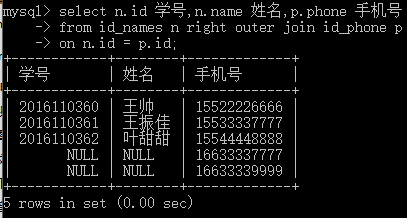
**outer 可以省略**



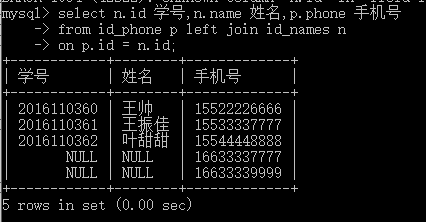
**> select n.id 学号,n.name 姓名,p.phone 手机号from id\_phone p right join id\_names n on p.id = n.id;**



**select n.id 学号,n.name 姓名,p.phone 手机号from id\_names n right outer join id\_phone p on n.id = p.id;**

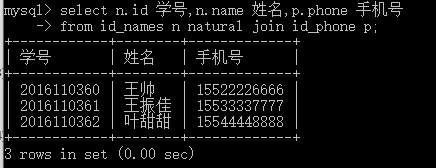


**select n.id 学号,n.name 姓名,p.phone 手机号from id\_phone p left join id\_names n on p.id = n.id;**



### 自然连接查询

select n.id 学号,n.name 姓名,p.phone 手机号from id\_names n **natural join** id\_phone p; 自动查找列名相同的元素



## 内外连接查询的区别

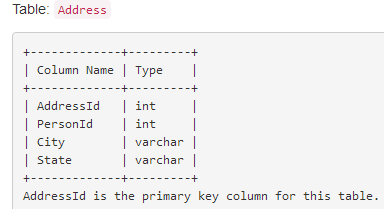
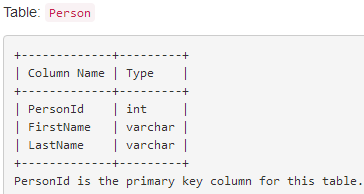
**外连接**：可以将**一个表中的所有元素**列出来，即使可能在另外一个表中没有相应信息，以NULL填充，所以分左连接和右连接；

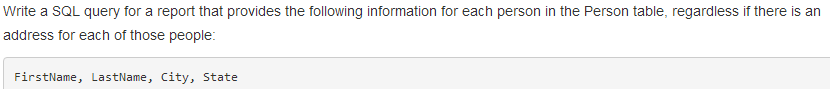
**内连接**：只会将两个表格的共同具有的信息列出来，每个表格独有的信息不会列出来。

LeetCode题目：考察外连接（左连接）

Write a SQL query for a report that provides the following information for each person in the Person table, **regardless if there is an address for each of those people**:

翻译：无论在address表格中是否具有这个人的信息，都要列出来；即必须使用外连接，且以people表格为参照表格。





Answer：

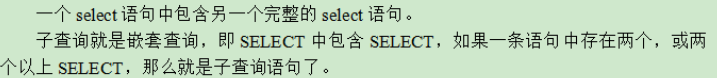
**select p.FirstName,p.LastName,a.City,a.State**

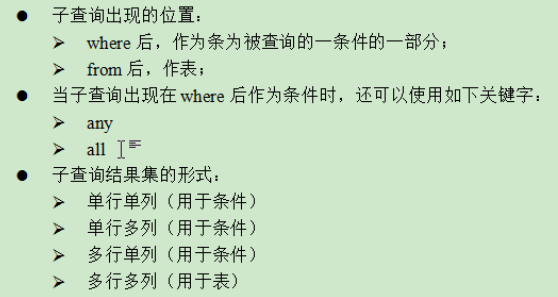
**from Person p left join Address a**

**on p.PersonId = a.PersonId;**

Note: Using where clause to filter the records will fail if there is no address information for a person because it will not display the name information.

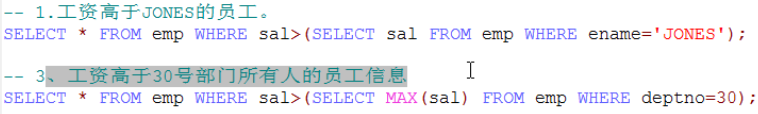
# 子查询（非常重要）（嵌套查询）

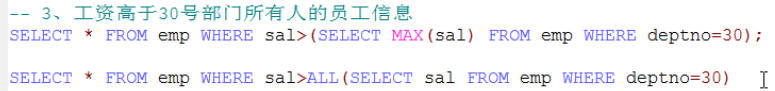




## 单行单列：





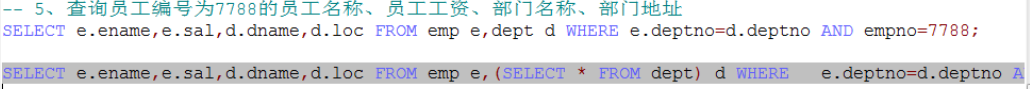


上面两种方式等价。

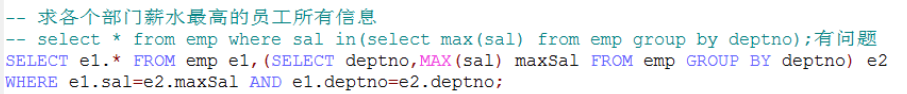
## 单行多列：







示例：



## 自连接查询

**把自己一张表，当成两个或多个表格**。

