昨日回顾：

高级查询：

select 【all|distinct】 字段列表|\*|字段名 【as】别名 from 数据源 【as】表别名 【where子句】【group by子句】【having子句】【order by子句】【limit 子句】

【all|distinct】 select选项：

all 默认的，显示时包含重复的记录

distinct 去重

字段别名： as 别名

数据源： 数据的来源。

单表数据源 就是一个表

多表数据源 多个表

迪卡尔积 在形式上是字段数相加，记录数相乘。

子查询数据源 from后面是一个select语句，这样的数据源就是子查询数据源。

任何一条select语句的结果，都是在内存中的产生一个临时表，完全可以作为一个表来使用，但这个表需要定义一个别名。

【where子句】

作用是在将数据从硬盘中读取到内存中时进行第1次筛选。

【group by子句】

作用是对where子句得到的结果进行分组统计。

统计函数(聚合函数)：

count(\*|字段名)

count(\*) 统计每一组的记录数

count(字段名) 统计每一组中指定的字段不为null的记录，如果被统计的字段为Null,那么这条记录并不会参与统计。

max()

min()

avg()

sum()

注意：

在进行分组统计时，一般字段列表中只会出现分组字段，以及统计函数。

多字段分组统计:

多字段分组统计的统计函数默认是对最小组进行统计。如果想对包含最小组的大组进行同样的统计，这就需要使用回溯统计——with rollup

【having子句】

作用：

对group by得到结果再进行第2次筛选，也就是对统计的结果进行筛选。

【order by子句】

作用：

对where子句、group by子句、having子句，得到的结果进行一个显示顺序上的控制。

语法：

order by 字段名 【asc|desc】【,字段名【asc|desc】】

【limit 子句】

作用：

对where子句、group by子句、having子句、order by子句，得到的结果进行一个显示行数的控制。

语法：

limit 【offset,】rows

offset是一个偏移量，第1条记录的偏移是为0，第2条记录的偏移量为1…..，如果省略默认为0

数据分页：

与分页相关的因素：

每一页显示的记录数 rowsPerPage 人为设定

当前页 curPage 用户点击的页码数 1234567

limit (curPage-1)\*rowsPerPage,rowsPerPage

联合查询：

为了加快查找一条记录，所以使用分表存储数据，这样是解决了查找一条记录的效率。

但是还会有将各个分表中的数据显示在一个结果集中的情况，这就可以使用联合查询。

语法：

(select语句A)

union 【all|distinct】

(select语句B)

说明：

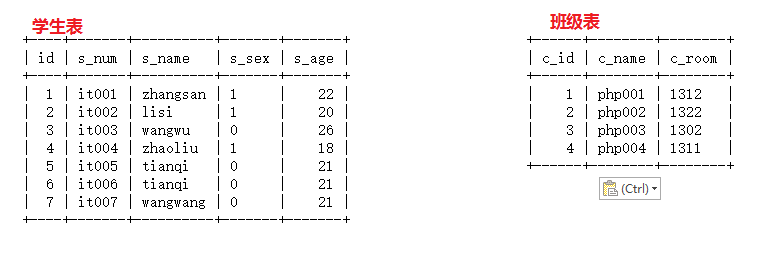
如果出现order by子句，那么必须匹配limit子句。

一、外键

1、业务场景

表关系：就是告诉大家在为某个项目创建数据表时，一个数据表就存储一个实体的信息。

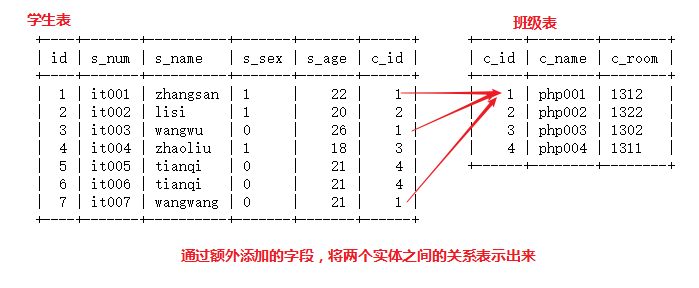
根据表关系的学习，我们将两个实体分别存在两个表中

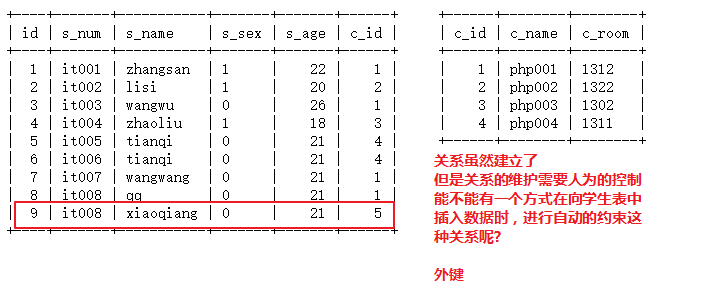


现实中学生与班级之间具有关系，如何在数据表中之间也将这种关系表示出来呢？

解决方案：

在学生表中添加一个字段，用于存储这个学生所属的班级的id，这样的话就建立了表与表之间关系，也就是学生与班级之间的关系。





2、外键

表B的某个字段的取值，是表A中主键字段值中的1个。我们就称这个字段就是外键字段。

表B中被表A中约束的字段就是外键字段。

表B称之为从表

表A称之为主表

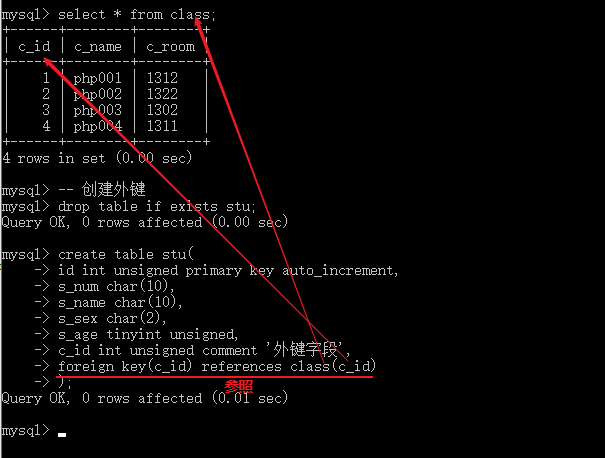
3、创建外键

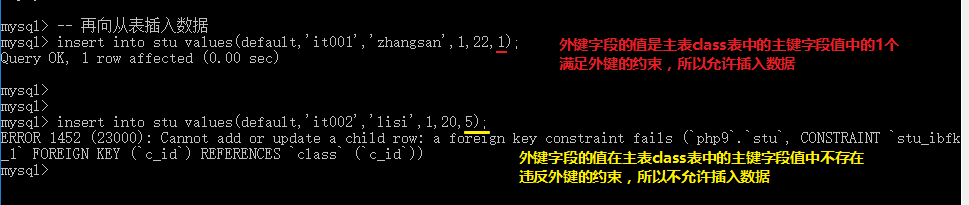
要创建外键，必须创建主表

语法：

foreign key(外键字段) references 主表(主键字段)

示例：





提示：

外键的主要的作用就是建立一种强制的约束。

4、外键的约束

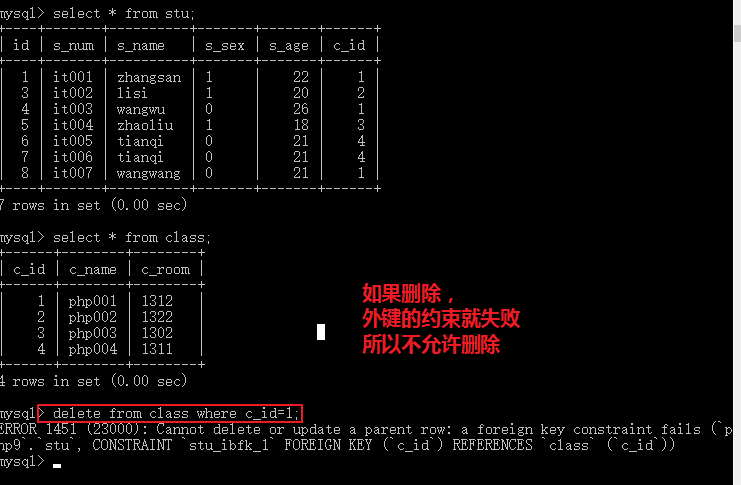
①、从表被主表约束

外键字段的类型必须与主表中的主键字段的类型一致。

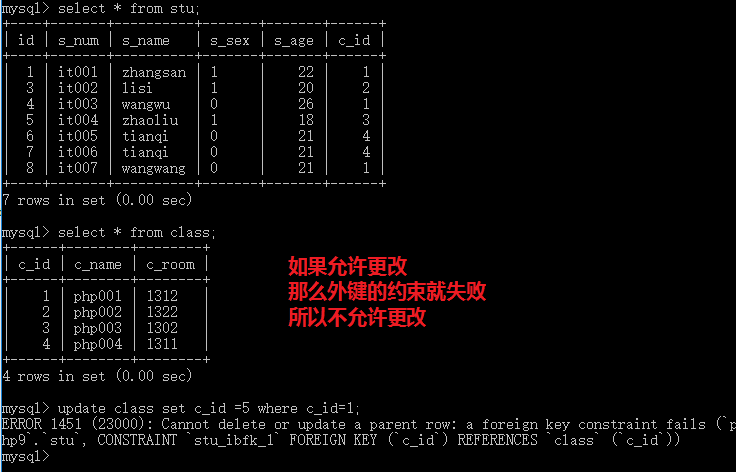
从表中外键字段的值必须是主表中主键字段值中的1个。

②、主表被从表约束

不能随意的删除主表中的记录



主表中主键字段的值也不能随意的更改



③、级联约束与置空约束

默认的主表中的记录不能随意删除，主表中的主键字段的值也不能随意修改，但实际开发中确实有这种需求。那么如何解决？级联约束与置空约束就是解决这种需求。

on update cascade

当主表的主键字段的值被修改后，从表中与之相关的外键字段的值也同时修改。

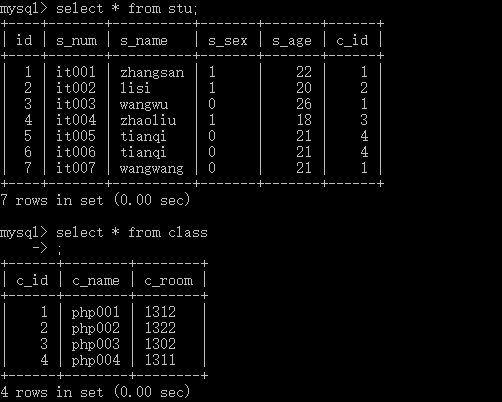
on delete set null

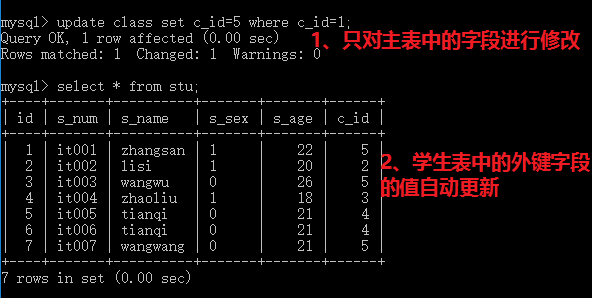
当主表中的记录被删除时，从表中的与之相关的外键字段的值设置为null

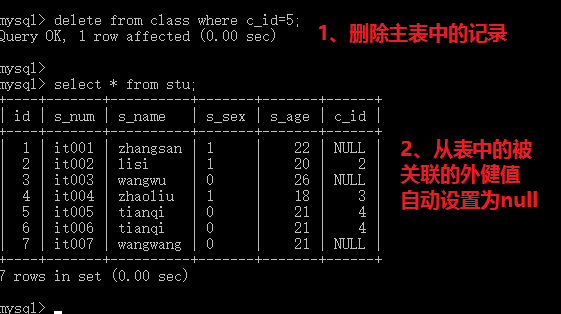
示例：设置级联与置空约束



测试：

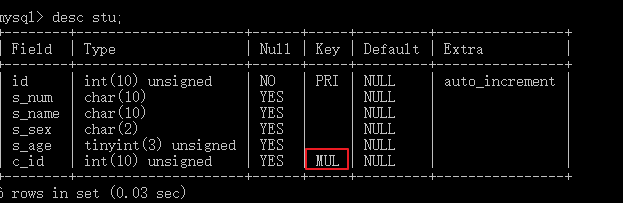






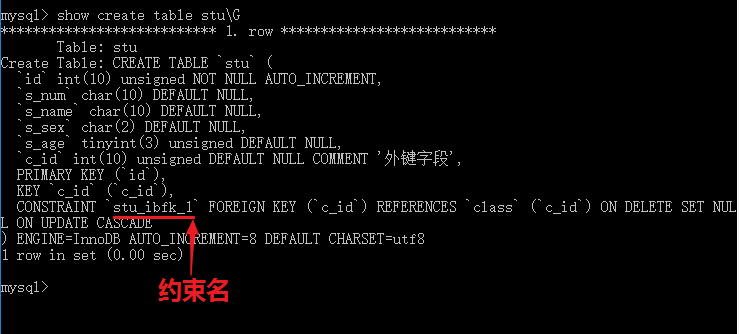
5、删除外键

查看外键：

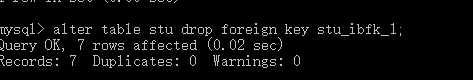


语法：  
 alter table 表名 drop foreign key 外键的约束名。

查看外键的约束名：



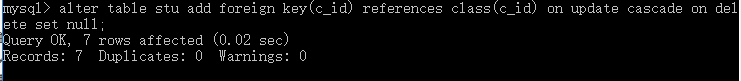
删除：



6、添加外键

语法：

alter table 表名 add foreign key(外键字段) references 主表(主键字段) 【on update cascade】【on delete set null】



二、子查询

1、定义

在一个select语句中包含另一个select语句，被包含的select语句就是子查询。

2、子查询的分类

有两种分类方式：

根据子查询的出现的位置

where子查询

from子查询

根据子查询查询出来的数据

标量子查询

列子查询

行子查询

以上三个子查询都是where子查询

表子查询

表子查询是from子查询

3、子查询详解

①、理解where子句的表达式

where表达式的本质，就是字段名与值的等值匹配情况，有以下几种：

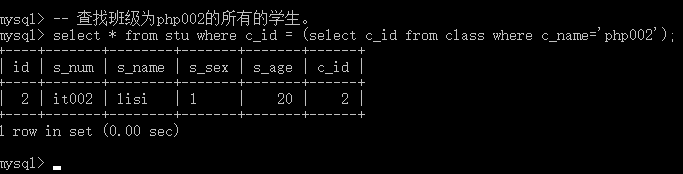
|  |  |  |
| --- | --- | --- |
| 字段个数 | 值的个数 | 需要使用的运算符 |
| 1 | 1 | = |
| 1 | 1 | In |
| 1 | n | in |
| n | n | (字段列表)**=**(值列表) |

②、标量子查询

子查询的书写方式：先写最终想得到的那个select，另一个子查询其实就是where表达式中的一个值。

查询出来的结果是1行1列。

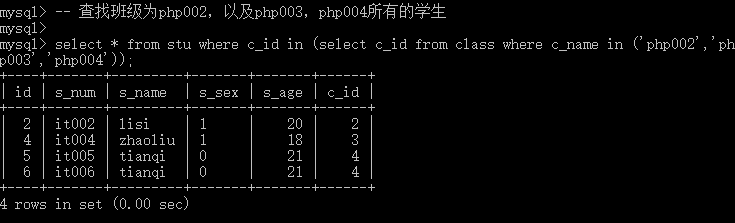
示例：



③、列子查询

子查询得到的结果是1列多行

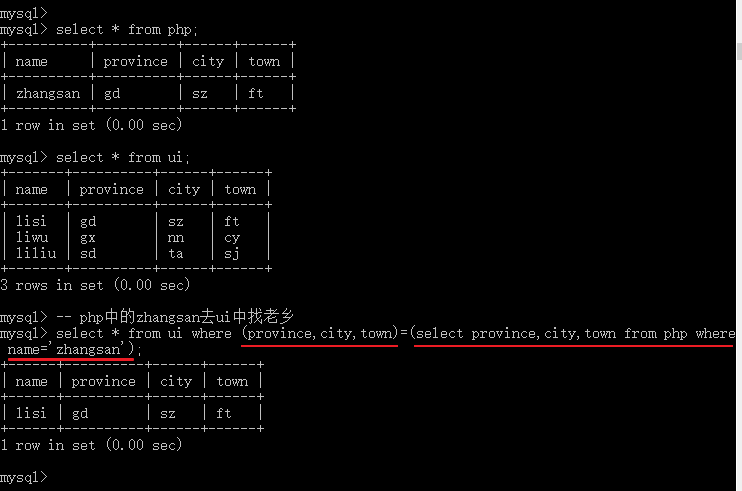
示例：



④、行子查询

子查询得到的结果1行多列。

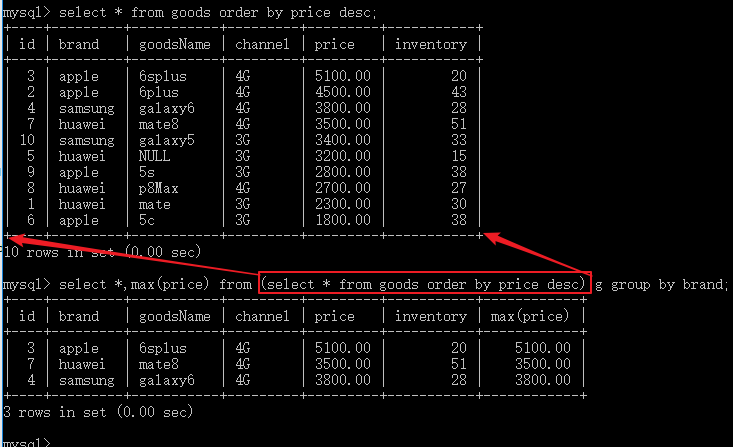
找老乡



提示：  
 以上是where子查询

⑤、表子查询

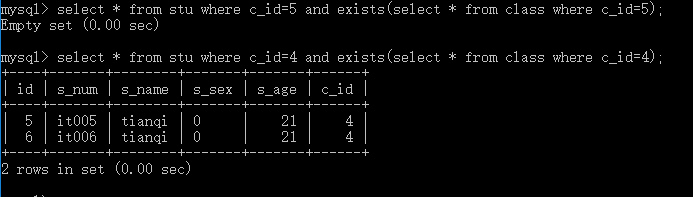
子查询得到的数据是多行多列，主要用在from后作为一个数据源。



**提示：**

**所有的子查询都必须使用括号括起来，但是只有表子查询必须使用别名。**

⑥、exists子查询



4、all、any、some

主要用于where表达式中的不等值比较。

all与any、some主要用于一个字段与多个值的不等值比较。

all 所有

any 任何一个 some(是any的一个同义词)

字段 >all(多个值) >最大的 一个字段多个值的and max()

字段 <all(多个值) <最小的 一个字段多个值的and min()

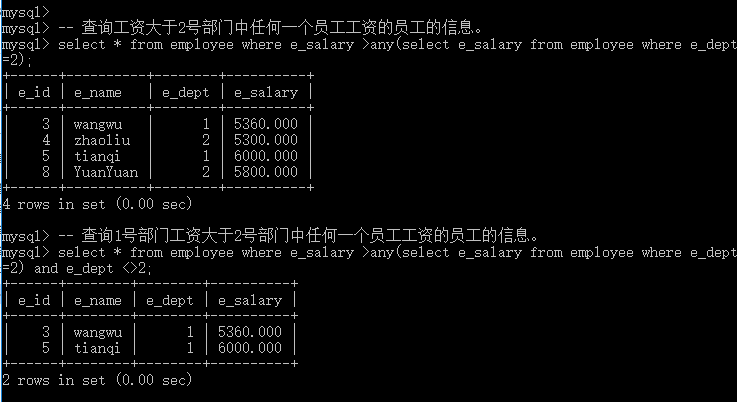
字段 <>all(多个值) not in

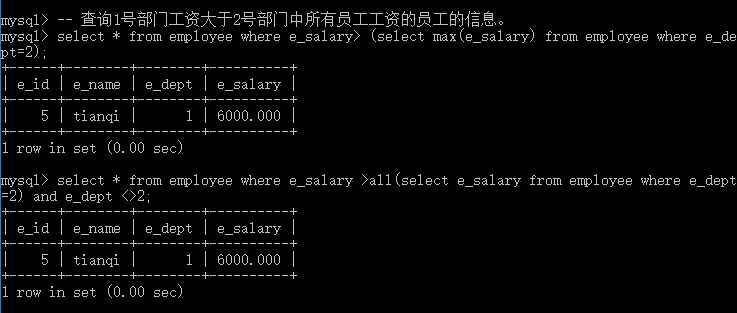
字段 >any(多个值) >最小值 一个字段多个值的or min()

字段 <any(多个值 ) <最大值 一个字段多个值的or max()

字段 =any(多个值) in

示例：





三、连接查询

子查询的特点：虽然涉及到多个表的查询，但最终得到的结果数据只来源于一个表的数据。

连接查询的特点：最终数据是来源于多个表。

1、内连接

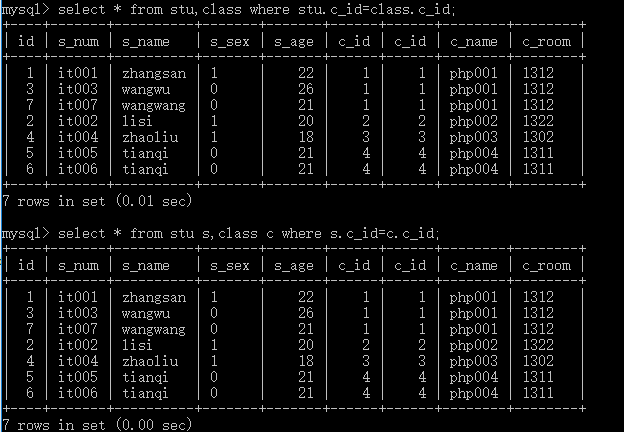
①、隐式内连接

语法：

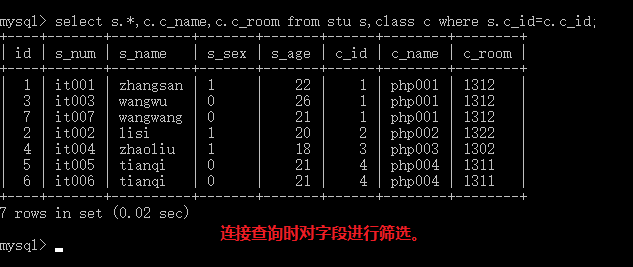
select \* from 表A,表B 连接条件;

说明：

默认多表数据源产生的结果是迪卡尔积，但如果指定连接条件，那么就可以将有意义的记录筛选出来。



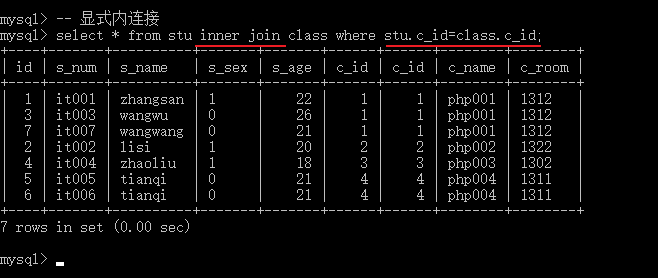
示例：



②、显式内连接

语法：  
 select \* from 表A 【inner 】 join 表B 连接条件;

示例：



提示：

内连接的特点，得到结果就是两个连接的表中匹配成功的记录，对于匹配不成功的丢弃。

内连接也称之为等值连接，根据连接条件相等的记录才保留。

2、外连接

①、左外连接

语法：

select \* from 表L **left join** 表R **on** 连接条件

②、右外连接

语法：

select \* from 表L **right join** 表R **on** 连接条件

说明：

无论是左外连接还是右外连接，匹配成功的记录都会被保留。

左外连接与右外连接的匹配是在匹配不成功时的处理方式。

匹配不成功有两种情况：

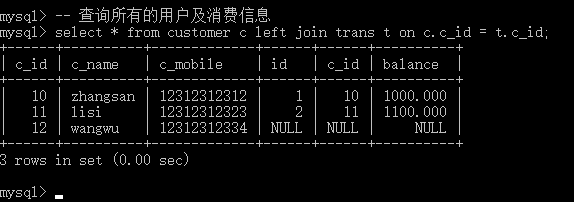
左表中的记录在右表中找不到匹配的记录。

右表中的记录在左表中找不到匹配的记录

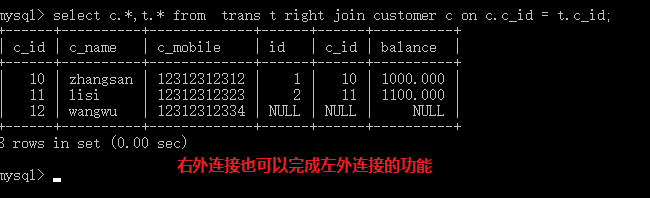
对于左外连接，会保留左表中匹配不成功的记录。

对于右外连接，会保留右表中匹配不成功的记录。

示例：



示例：



小节：

内连接，只保留匹配成功的

外连接，不但保留匹配成功的，还会保留匹配不成功。

在左外连接与右外连接，主要体现在匹配不成功的记录上。

3、自然连接(扩展了解）

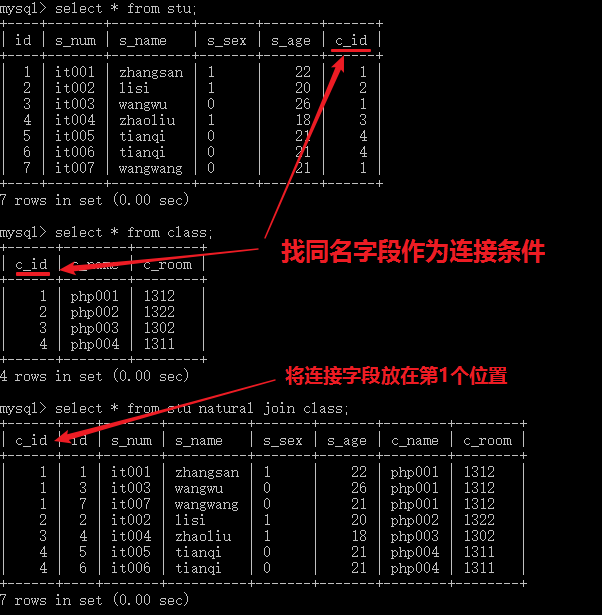
自然连接就是自动匹配，**不需要人为的指定连接字段**。自然连接会自动找进行连接的两个表中**同名的字段进行连接**。

自然连接就是对内连接与外连接的一种简化。

①、自然内连接。

语法：  
 select \* from 表A natural join 表B;

示例：



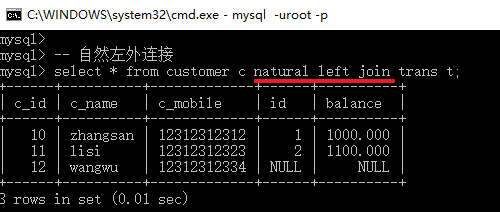
②、自然左外连接

自然左外连接就是对左外连接的简化。

语法：

select \* from 表L natural left join 表R;

示例：



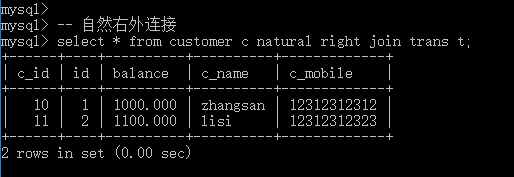
②、自然右外连接

自然右外连接就是对右外连接的简化。

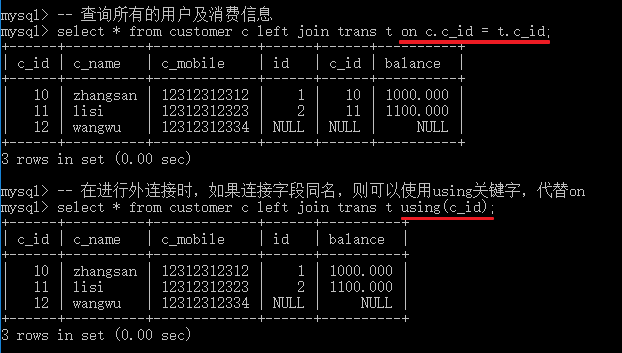
语法：

select \* from 表L natural right join 表R;

示例：



using关键字



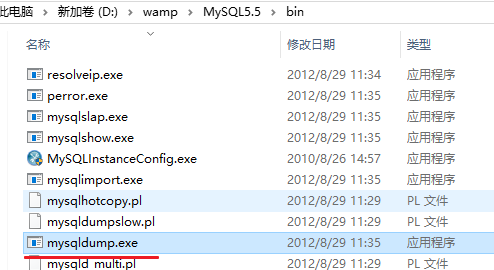
四、数据备份

1、sql备份与还原

①、sql备份

我们是想对数据进行备份，任何一个时刻，数据库的数据表中数据都可以使用一条insert语句来表示。sql备份备份的是sql语句。

备份工具是mysqldump.exes

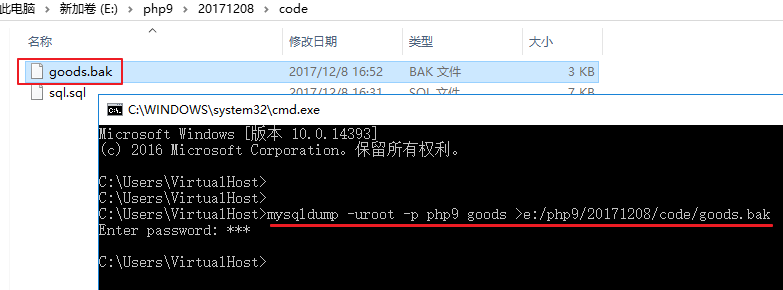


运行环境在cmd窗口

语法：

mysqldump –uroot –p 数据库名 【数据表】> 备份的文件名

示例：

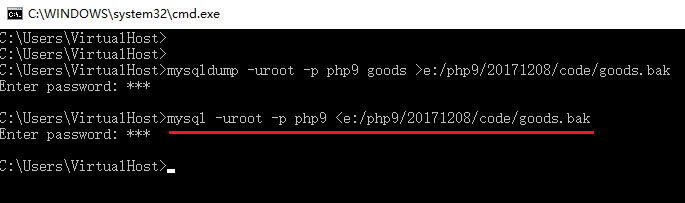


②、还原

工具mysql.exe

语法：

mysql –uroot –p 数据库 < 还原的文件(备份文件)



2、数据备份

①、备份

语法：

select 字段名 from 表名 where 条件 into outfile ‘备份文件’ 【fields选项】【lines选项】

说明：

fields用于设置字段的相关信息

以fields开头

terminated by 设置字段与字段之间的分隔符 默认为tab

enclosed by 设置字段的包裹 默认是空字符’’

escaped by 设置null值使用什么代替 默认\N

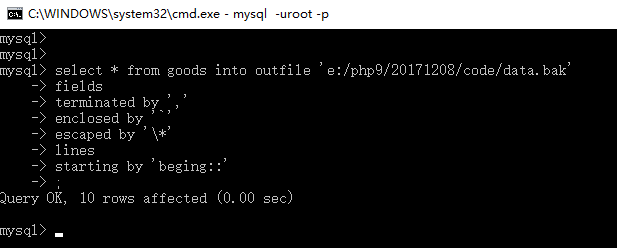
lines用于设置记录的相关信息

以lines开头

starting by 设置一行的开始符号

terminated by 设置一行的结束符号 默认\r\n

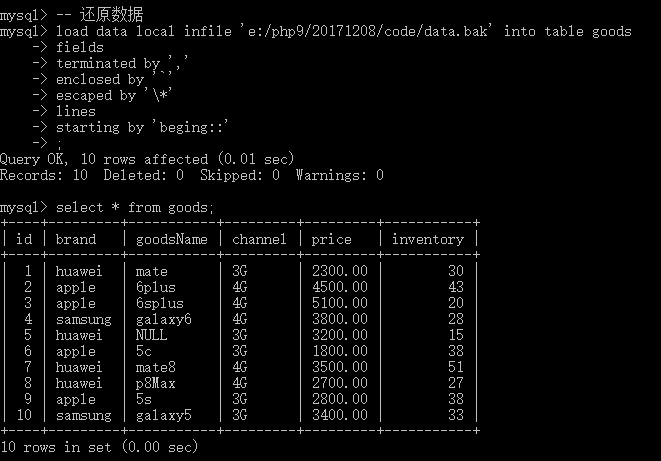
示例：



②、还原

语法：

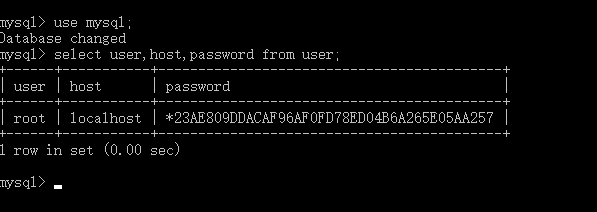
load data local infile ‘还原的文件名’ into table 表名 【fields选项】【lines选项】



五、用户管理

1、查看用户

数据库中的用户数据是保存在mysql数据库中的user数据表中。



2、创建用户

语法：

create user ‘登陆名’@’标识’ identified by ‘密码’;

说明：

登陆名 用户名

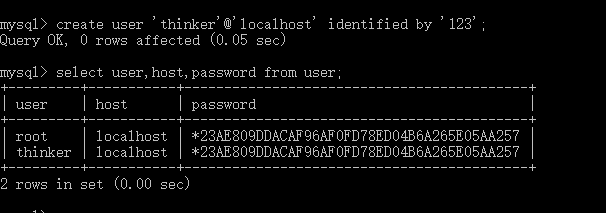
标识 作用是用于标记这个用户所能登陆位置，取值：

localhost 只能在本机登陆

ip地址 220.231.114.115

% 不限制登陆位置

示例：



3、授予权限

语法：

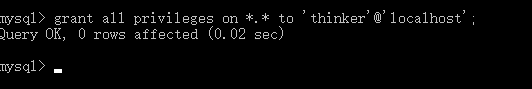
grant 权限或all privileges on 数据库名【.数据表名】 to ‘登陆名’@’标识’

说明：

all privileges 所有的权限

权限 create、drop、alter、select、update、insert……

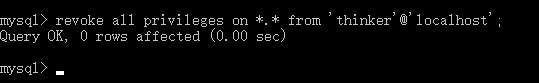
示例：



4、收回权限

语法：

revoke 权限或all privileges on 数据库名【.数据表名】 from ‘登陆名’@’标识’



5、删除用户

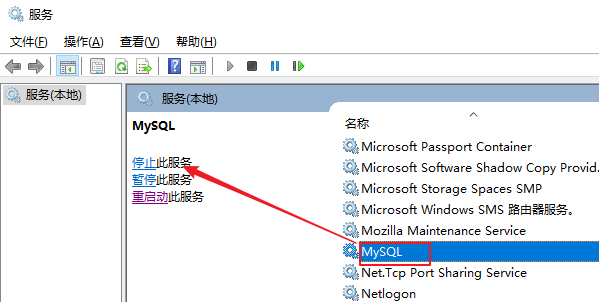
语法：

drop user ‘登陆名‘@’标识’



6、重置root密码

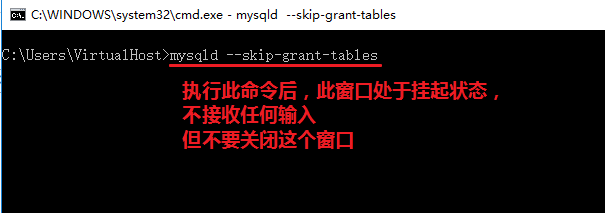
①、停止mysql服务



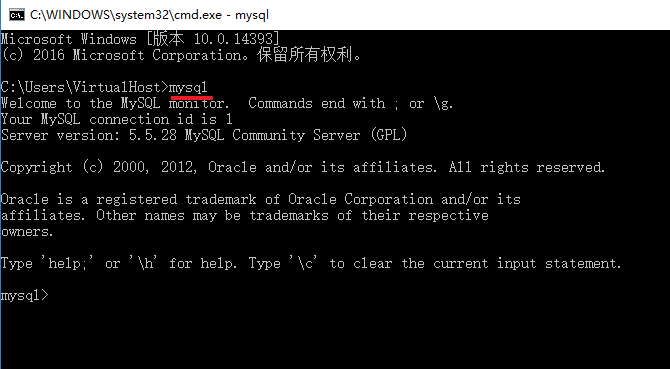
②、启动

使用mysqladmin.exe工具来启动mysql服务，mysqladmin.exe工具在启动msyql时可以跳过授权表。

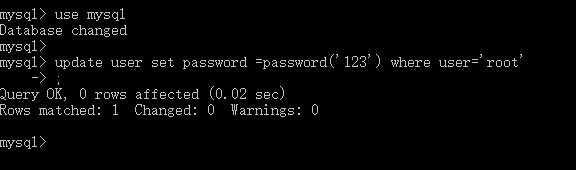
语法  
 mysqld --skip-grant-tables



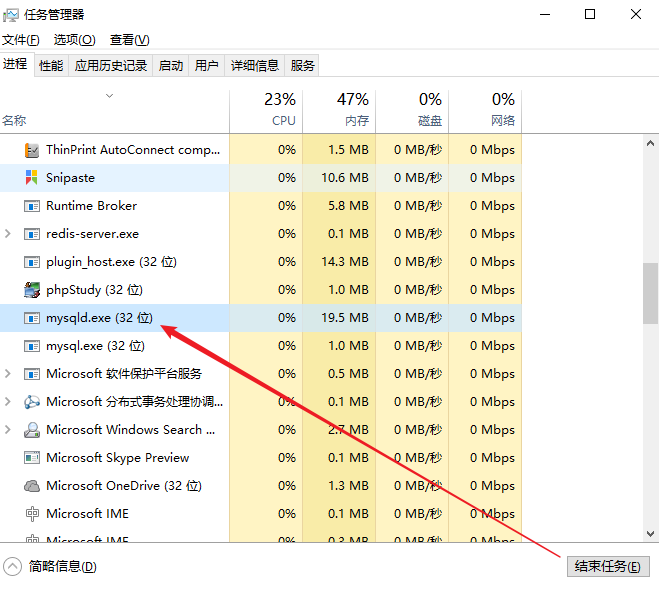
③、登陆mysql



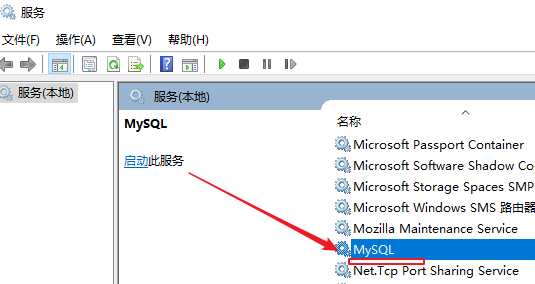
④、修改密码



⑤、关闭第2步中处于挂起的窗口，并结束mysql服务。



⑥、重启mysql服务



⑦、使用新密码进行登陆