**第4章 使用DML语句更改数据**

**1. 添加数据**

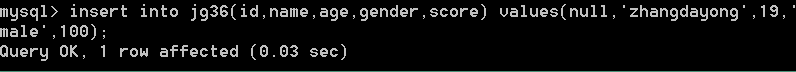
通过前面的学习，我们已经学会了如何建库和建表，但是如果表里面没有数据的话，这张表就没有意义，所以我们需要往表里面添加数据

添加数据语法：

1. 指定所有字段名 添加的关键字为insert into …. Values

Insert into 表名（字段名，字段名，字段名）values（数据，数据，数据）；

举例：



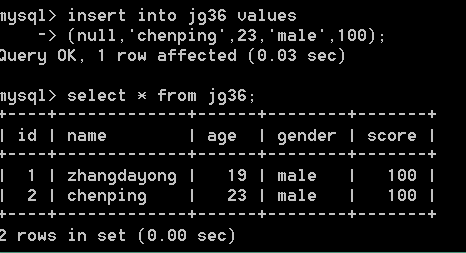
注意点：字段的顺序必须要和values后面值的顺序一一对应

②不指定字段名 △△△

Insert into 表名values（数据，数据，数据）；

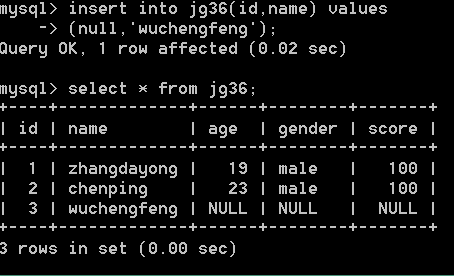
备注：因为没有指定字段名，所以这里录入的数据应该是对应所有字段

举例：



③ 指定部分字段

Insert into 表名（字段名，字段名，字段名）values（数据，数据，数据）；

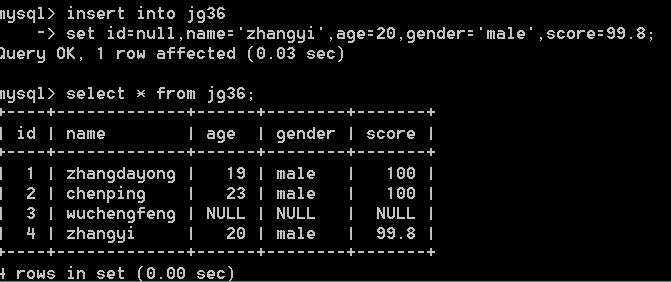


④ 添加数据还有一种方法

语法：

Insert into 表名 set 字段名=值1，字段名=值2….

举例：



⑤ 一次性插入多条数据

Insert into 表名（字段名，字段名，字段名）values

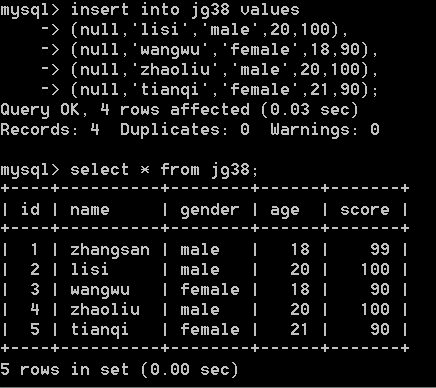
（数据，数据，数据），

（数据，数据，数据），

（数据，数据，数据），

（数据，数据，数据）；

备注：最后一条数据以分号结束



**2. 更新数据**

有些时候，我们需要对已经有的数据，进行修改。这也是一种很常见的操作。关键字为update….set（一般后面添加上过滤条件，过滤条件的关键字为where）

备注：所谓过滤条件，就是只要满足的都会被修改

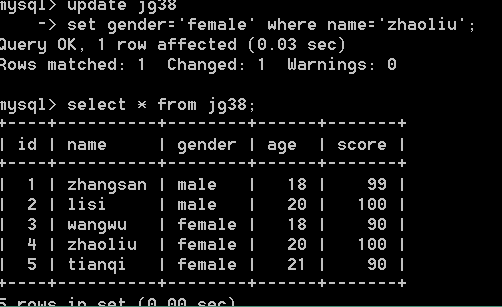
1. 修改单个字段的数据信息的时候，一般和关键字where（哪里，哪儿）

语法为：

Update 表名

Set 字段名=值；

举例：



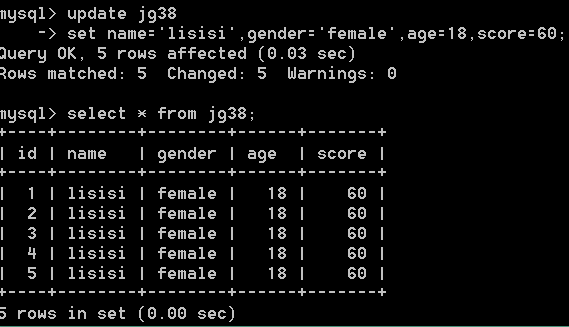
1. 修改多个字段的信息

语法：

Update 表名

Set 字段名=值，字段名=值，字段名=值；

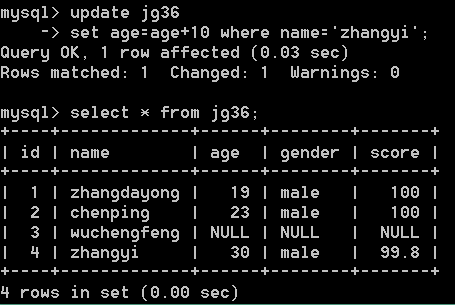
举例：



（忘记写过滤条件了，失败案例）

1. 我们的更新语句也是支持表达式的

举例：



**3. 删除数据**

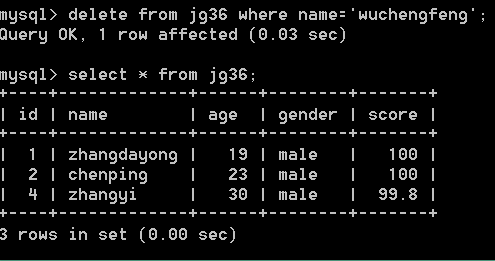
删除数据也是我们DML语句中所常用的操作，删除数据使用关键字delete….from

1. delete 删除部分数据

语法：

Delete from 表名 where 过滤条件

举例：

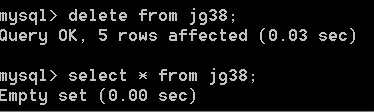


注意点：删除某条数据后，下面一条数据的编号并不会因为上面的数据被删除而改变

1. delete删除全部数据

Delete from 表名

举例：

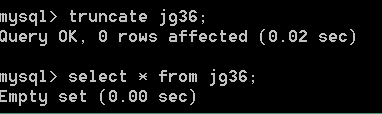


1. 用关键字truncate来删除表中的数据

语法：

Truncate 表名；

举例：

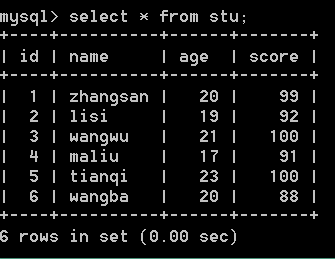


Delete和truncate的区别：

1. truncate是直接销毁表，然后建一个和所销毁的表一模一样的表，delete则是一条一条的对数据进行删除。所以truncate的效率要高于delete
2. delete可以使用过滤条件，指定删除某一部分的数据，然而truncate不行，truncate删除表的话就是删除所有数据
3. truncate语句一般是被认为是DDL语句，delete语句是DML语句
4. 使用truncate删除所有语句后再添加数据的话，自动增长字段的初始值从1开始，然而delete语句是从删除前最大值+1开始
5. **表的复制**

有些时候，我们需要对已有的表进行复制。这个我们称之为表的复制或者表的拷贝。表的复制大致下面几种方法：

现在有如下表数据：



1. 拷贝表结构到新表中

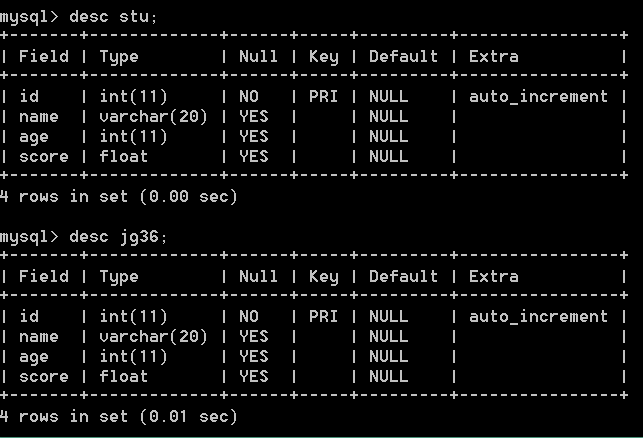
语法：

Create table 新表名 like 旧表名

举例：



我们会发现，虽然表建立成功了，但是里面没有数据，因为上面的语句只是对表结构进行拷贝



1. 拷贝数据到新表中

语法：

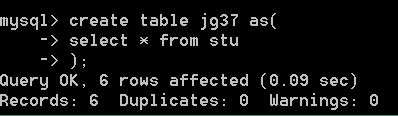
Create table 新表名 as

（

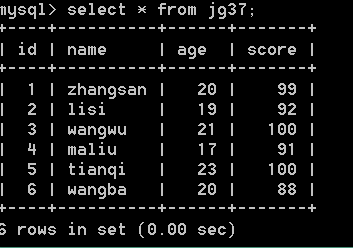
Select \* from 旧表名

）；

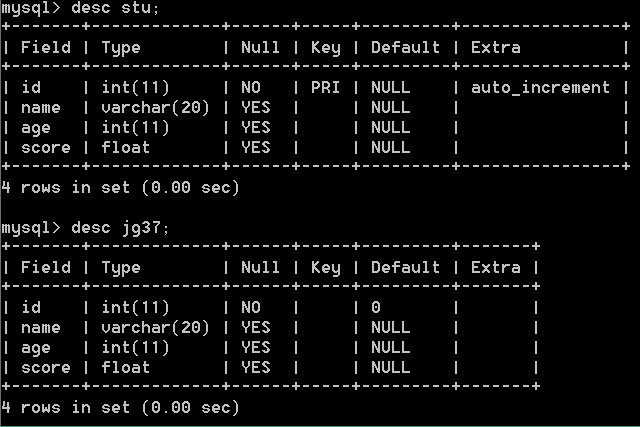
举例：



查询表以后，我们发现，stu表里面的所有数据已经被拷贝到jg37表里面了



但是，上面的语句仅仅是将stu里面的数据进行了拷贝，然后建立了一张新表将数据装进去，并没有对表的结构进行拷贝

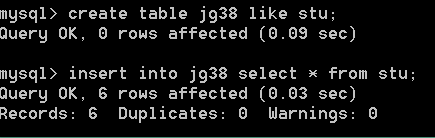


1. 如果你要拷贝一张表，既拷贝数据，同时也要拷贝表结构，可以这么写↓

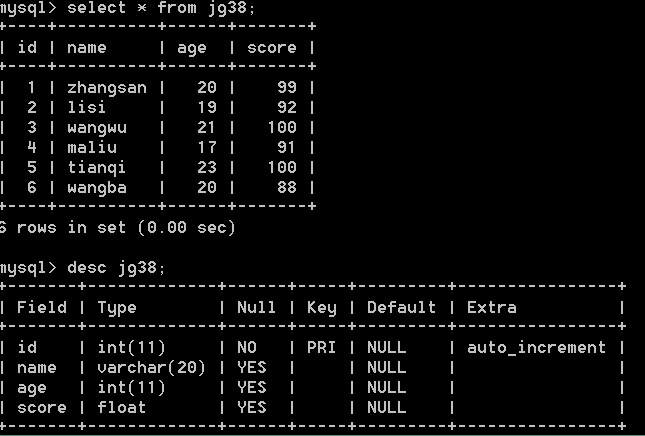
Create table 新表名 like 旧表名；

Insert into 新表名 select \* from 旧表名；

举例：



接下来我们再来查看表数据和表结构，发现表结构和表数据都进行拷贝



1. 我们还可以在不同的数据库中对表进行拷贝

举例：

当前数据库为school，在school里面拷贝jg36数据库当中的一张表stu



1. 我们可以拷贝一个表中的某些字段到新表里面

语法：

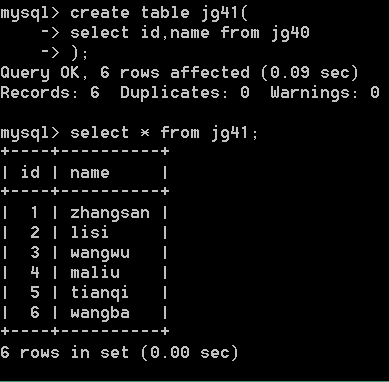
Create table 新表名 as

（

Select 字段名，字段名 from 旧表名

）；

举例：



1. 我们可以给新表的字段修改名称

Create table 新表名 as

（

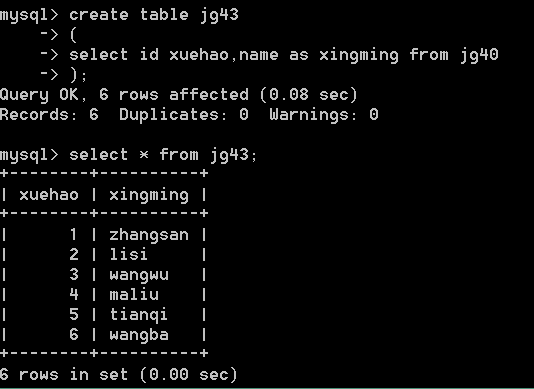
Select 旧字段名 as 新字段名 ，旧字段名 as 新字段名 from 旧表名

）；

举例：



备注：as这个关键字是可以省略的



1. 我们在拷贝表的时候同样是可以对旧表的字段进行过滤的

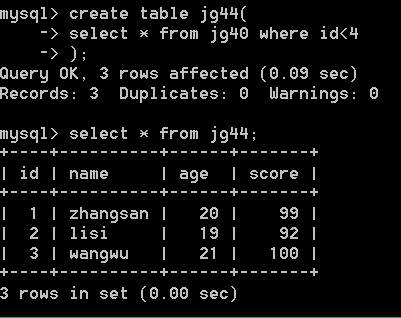
Create table 新表名 as

（

Select 字段名，字段名 from 旧表名 where 过滤条件

）；

举例：



1. 我们也可以在创建表的同时定义新表中的字段

例如：

Create table 新表名（

Id int primary key auto\_increment

)

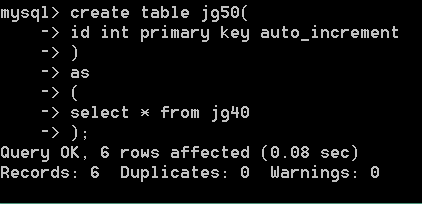
As

(

Select \* from 旧表名

）；

举例：



接下来我们来查看表的信息和结构，发现id这一字段就是我们自己定义的

