登陆：

mysql -h主机地址 -P端口号 -u用户名 –p密码

数据库操作

增 create database 数据库名 【库选项】

删 drop database 数据库名;

改

查

查看有多少个数据库存在 show database**s**;

查看创建语句 show create database 数据库名;

对表进行操作

增 create table 表名(

字段1 列类型 【列属性】,

字段2 列类型 【列属性】,

….

) 【表选项】

【表选项】 charset 存储字符集，collate 校验集，engine存储引擎

删 drop table 表名;

改

查

查看有多少个数据表存在 show table**s**;

查看表的创建语句 show create table 表名;

查看表的结构 desc 表名;

对数据进行操作

增 insert into 表名【(字段列表)】 values(值列表)

查 select 字段列表|\* from 表名 【where子句】【group by子句】【having子句】【orderby子句】【limit子句】

>、<、>=、<=、<>、=、and、or、not、between m and n、in(值列表)、is null、like模拟查询

like需要两个占位符

% 代表当前位置及其后的0个或多个字符

\_ 代表当前位置1个字符

where 字段名 运算符 值

改 update 表名 set 字段名=值,字段名=值…. 【where子句】

删 delete from 表名 【where 子句】

列类型：

数值型

整数型

tinyint 【unsigned】

smallint 【unsigned】

mediumint 【unsigned】

int 【unsigned】

bigint 【unsigned】

小数型

浮点型

float(m,d)

double(m,d)

定点型

decimal(m,d)

字符串型

定长 char(len)

len 最大取值255

变长 varchar(max)

max 理论取值是65535。(如一个字段想达到65535，有很多条件限制:表的存储字符集为lartin1 && 表中只有1个字段 && 不允许为Null)

mysql一条最大长度是65535个字节。

text文本类型

tinytext 255字节

text

mediumtext

longtext

blob

tinyblob

blob

mediumblob

logonblob

enum与set都是用于规范数据。从预定的多个数据项中，选择1个或多个。

enum(值列表)

set(值列表)

日期时间型

year(2|4)

date ‘2017-12-05’

time ‘9:56:30’

datetime ‘2017-12-05 9:56:30’

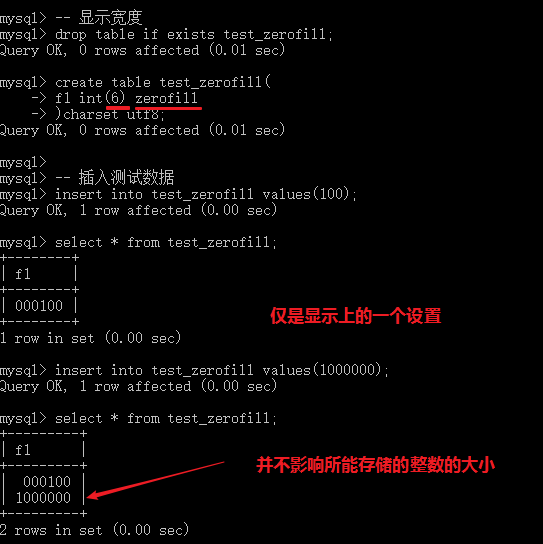
timestamp ‘2017-12-05 9:56:30’

一、显示宽度与0填充

在我们为某个字段设置整数类型时，可以为其指定显示宽度。必须与zerofill联合使用

语法：

整型(显示宽度)



二、列属性

create table 表名(

字段名 列类型 【列属性】,

字段名 列类型 【列属性】,

…..

)【表选项】

1、列类型 VS 列属性

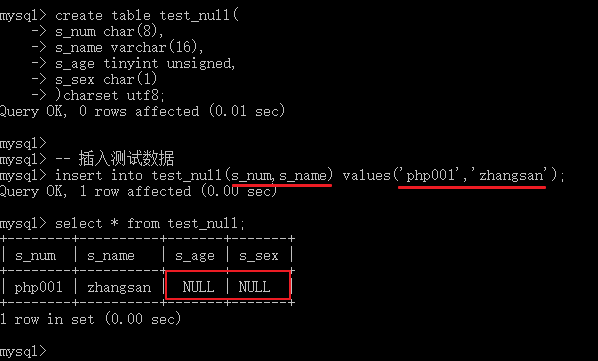
列类型主要用于约束单独的一个字段在整行中的取值限制。

列属性主查用于约束单独的一个字段在整列中的取值限制。

2、null

MySQL系统的默认值，

默认，在插入记录时当我们没有为某个字段提供数据时，那么mysql会使用null值为填充。

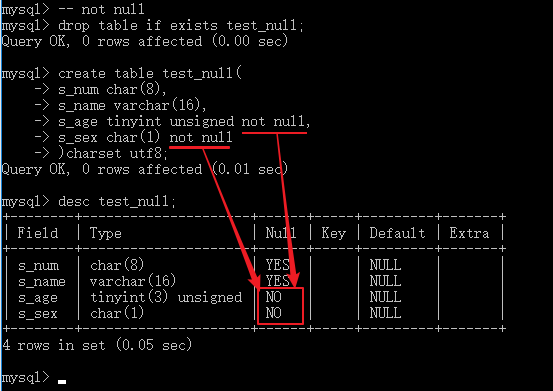


提示：

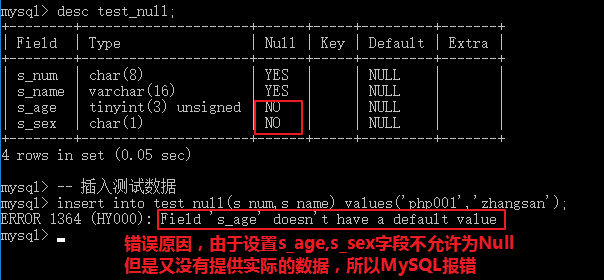
一般我们在创建表时，不允许字段的值为null。

3、not null

主要用于约束当前的字段不允许为null值(不允许MySQL使用null值进行填充)



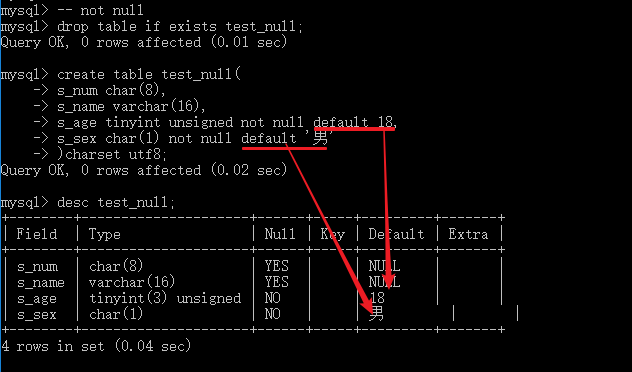
插入测试数据：



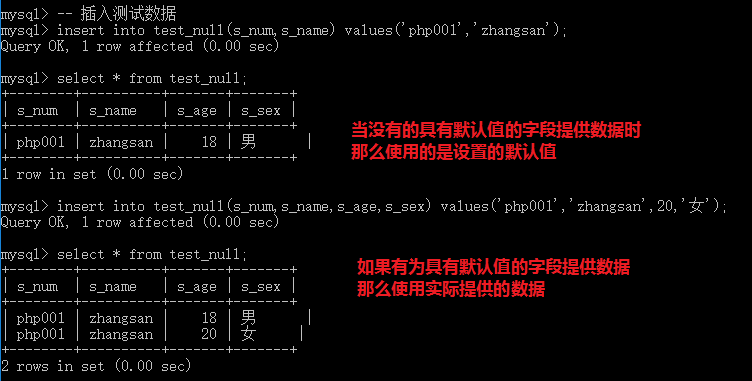
当为某个字段设置了not null约束后，要么在插入数据时必须提供数据，要么为not null字段设置一个默认值。

4、default

自定义默认值。

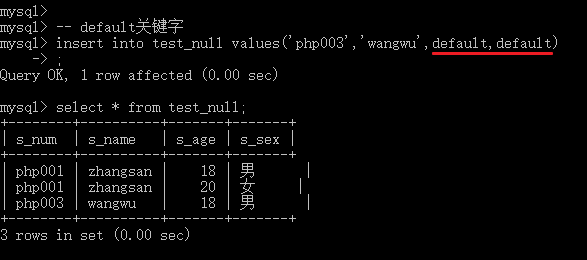


插入测试数据：



default关键字

当我们插入记录时，也可以使用default关键字。来代替默认值。



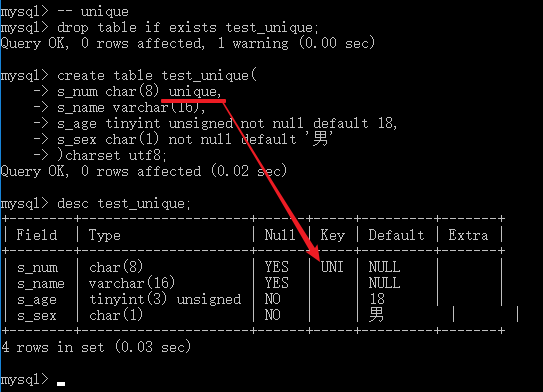
提示：

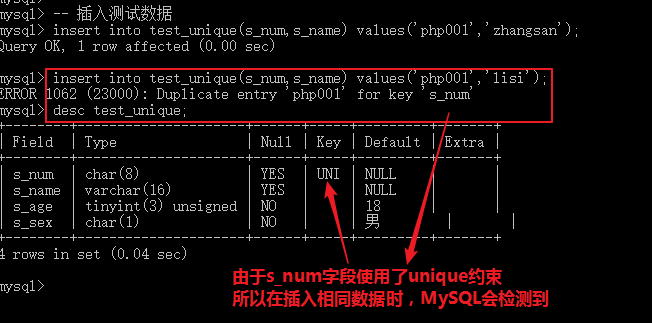
在设置默认值时，如果有很大的机率能确定这个值就使用确定的值。

如果没有办法确定，对于字段型的字段使用default ‘’，对于数值型使用default 0

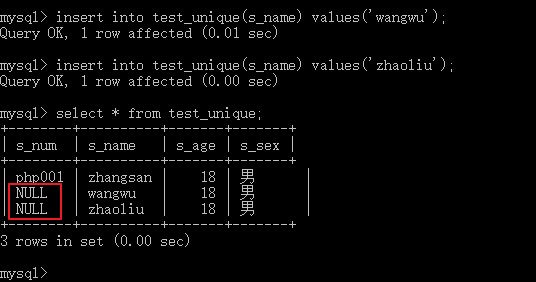
5、unique

用于约束当前的字段在整列中都是唯一的。





示例：unique的特点



提示：

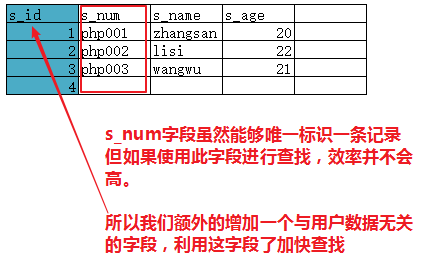
unique约束，不允许字段的值重复，但是允许为null，原因是null值根本不等于null

6、primary key

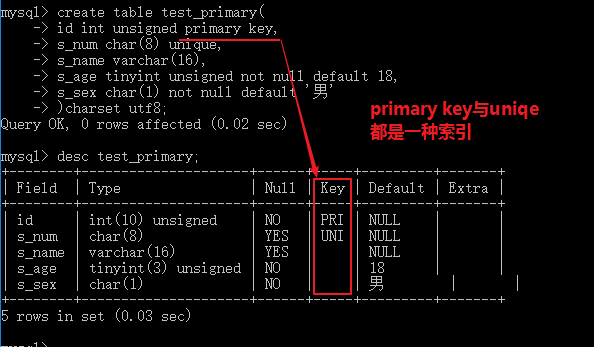
primary 主要的意思

key 键，字段

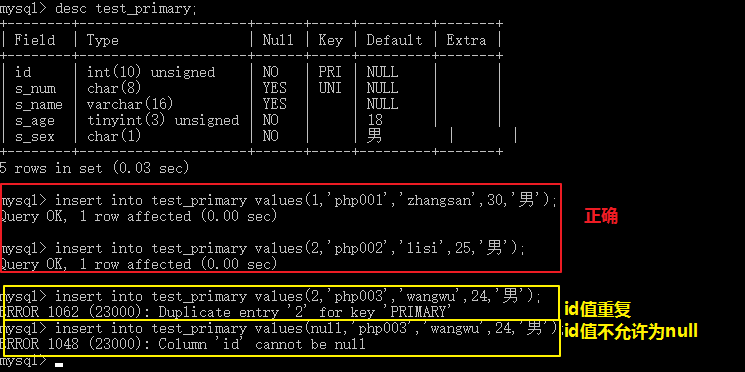
primary key 重要的主要的字段。此处的重要并不说对用户重要，而是对MySQL来说重要。主要体现在对记录的查找上。



示例：

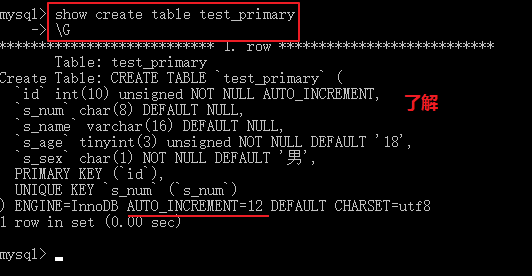


示例：插入测试



**提示：  
 主键的特点，不允许重复，且不允许为null值**

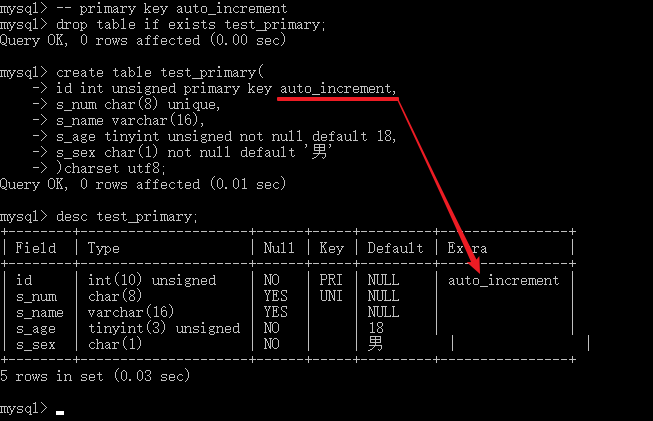
查看下一个自增长的主键的值。



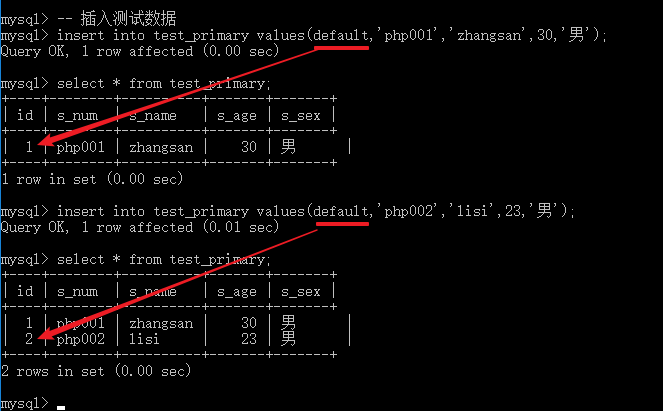
7、auto\_increment

对于主键的值，我们选择了整数型，就是想达到一种自动增长，这样就不会重复。而且能唯一标识一条记录。

auto\_increment，是自增长，就是用于解决主键值的问题。



示例：

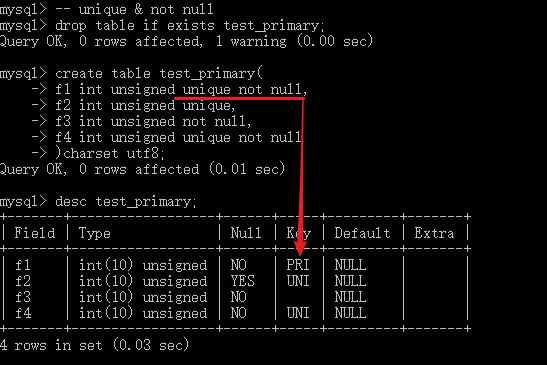


提示：

对于auto\_increment属性，必须应用在整型字段并且是primary key 或者unique。但是绝大多数与primary key联合使用。

8、模拟主键(了解)

当一个表中如果没有设置primary key，那么MySQL会在表中寻找第1个具有unique与not null约束的字段，作为主键字段。



提示：

一个表中能有1个主键

一个表中能有多个唯一键

一个表中能有1个自增长

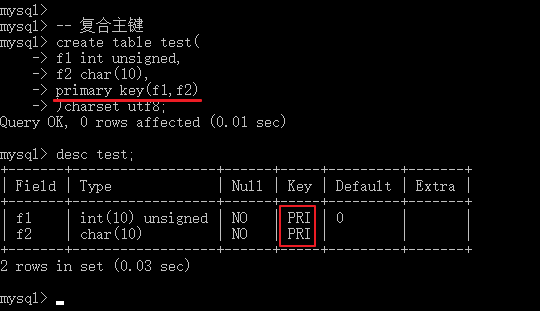
9、复合主键(了解)

MySQL中可以使用多个字段，共同组成主键字段。

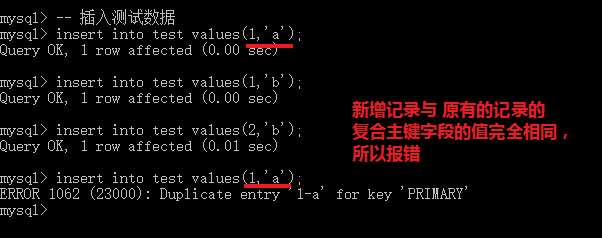
语法：

primary key(字段1,字段2)

示例：



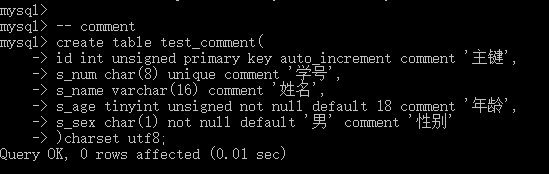
对于复合主键的值，只要当组成复合主键的字段的值完全相同才算是重复。

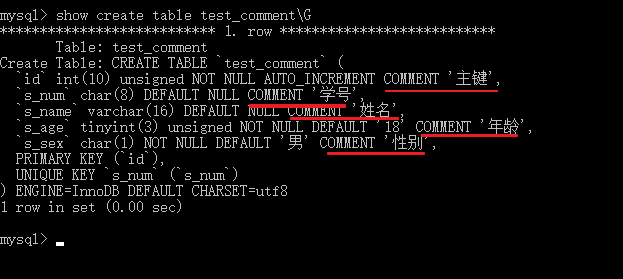


10、备注

语法：

comment ‘备注’





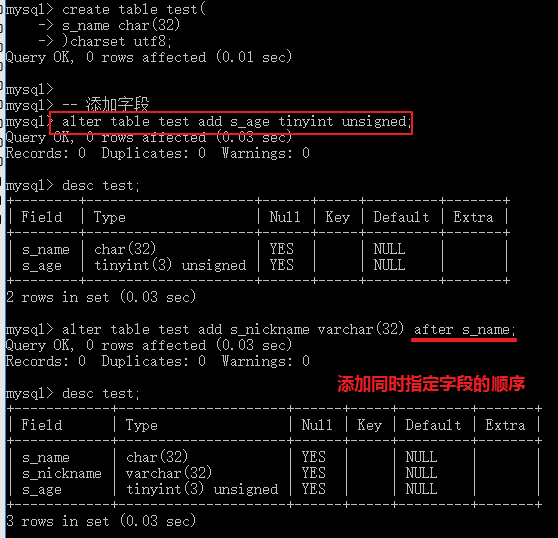
三、修改表结构

**总的语法：** alter table 表名 add column|add primary key|drop|drop primary key|drop key|modify|change|rename|

1、增加与删除字段

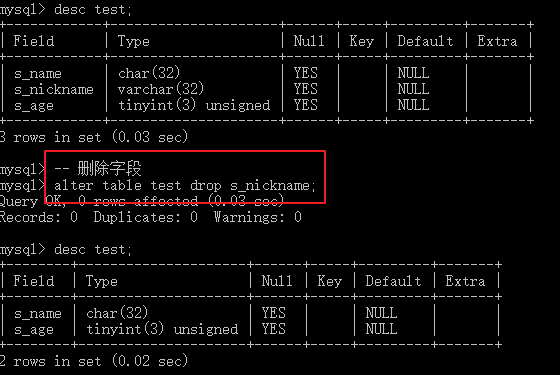
①、添加

alter table 表名 add 【column】 字段名 列类型 列属性 【first| afterp字段名】



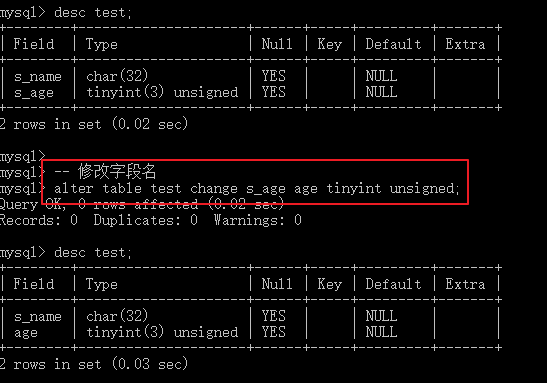
②、删除

alter table 表名 drop 【column】 字段名;



2、修改字段名

alter table 表名 change 原字段名 新字段名 列类型 列属性。



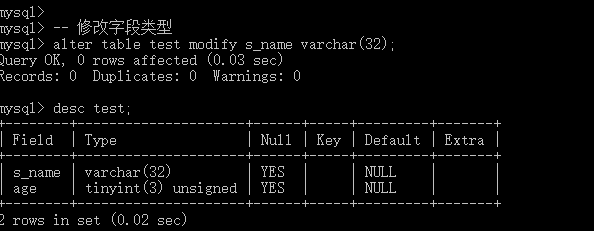
提示：

就算是仅修改字段名，那么字段的原类型，原属性也要重新书写，不然就变为删除。

3、修改列类型

alter table 表名 modify 字段名 列类型 列属性 【first|after】

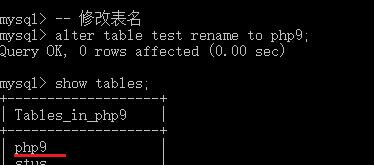
示例：



4、修改表名

alter table 表名 rename to 新名

示例：



5、修改表选项

create table 表名(

)【表选项】

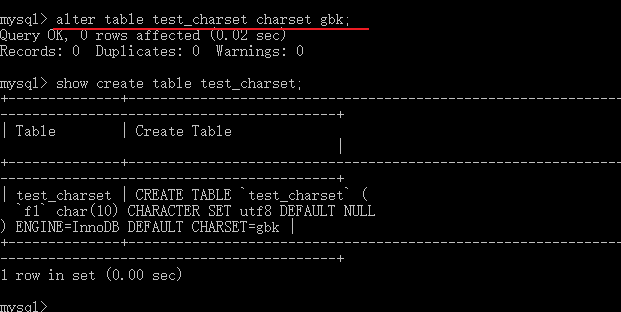
engine 存储引擎

charset 存储字符集

collate 校验集

语法：

alter table 表名 engine 存储引擎|charset 存储字符集|collate校验集



提示：

虽然MySQL提供了修改存储字符集的命令，但是如果一个表中已有数据，那么不要执行修改存储字符集的命令。

6、修改列属性

列属性包含not null、default、unique、auto\_increment、primary key。

当一个表创建之后，对于列属性的操作，我们可以使用alter table 表名 modify来操作，在操作时，如果书写了列属性就是添加列属性，如果没有书写就是删除原有的列属性。

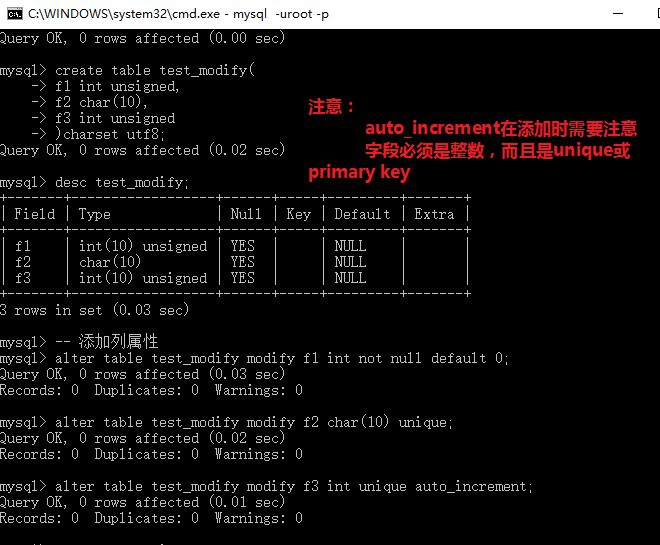
但是有两个特殊的列属性，primary key 与unique。

①、增加列属性

**普通的列属性**

alter table 表名 modify 字段名 not null defult 值 unique、auto\_increment

示例：



提示：

注意unique与default不能联合使用。

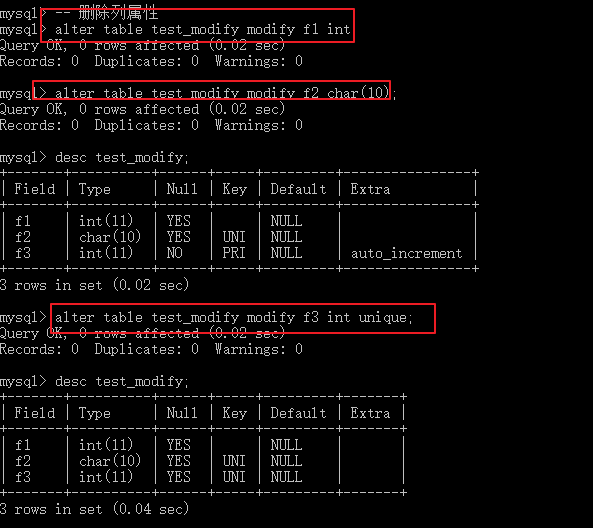
**主键字段的添加**

语法：

alter table 表名 add primary key(字段名)

②、删除属性

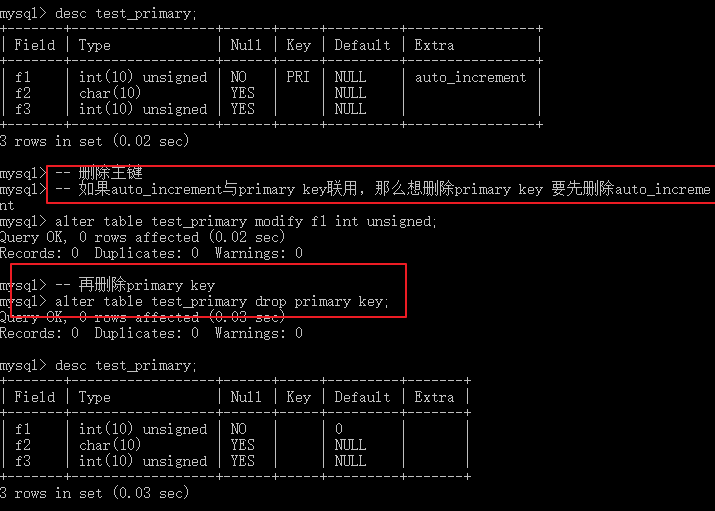
**普通的列属性**



**对于not null、default、auto\_increment不书写就是删除。**

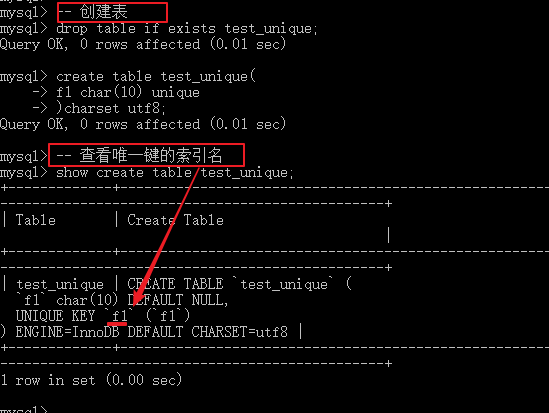
primary key与unique的删除

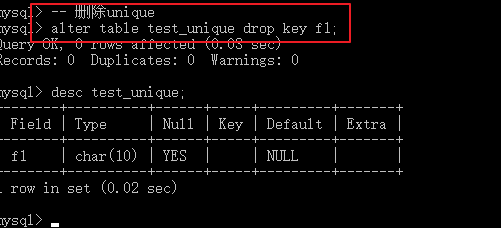
alter table 表名 drop primary key;



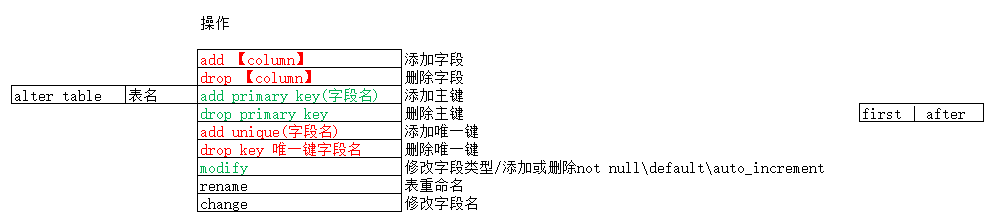
alter table表名 drop key 唯一键的索引名;

默认唯一键的索引名与字段名相同。





小节：



四、实体与实体关系(了解)

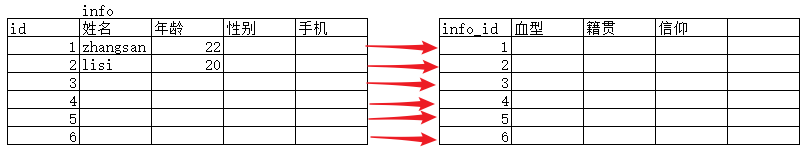
实体：

现实世界中的任何一个事物都是一个实体。在使用MySQL保存现实中的事物的信息时，那么一个事物(一个实体)的信息就保存在一个表中。

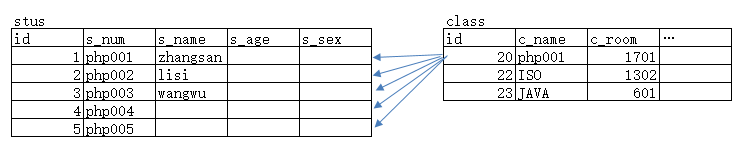
现实中的实体与实体之间具有关系，那么MySQL数据库中，保存实体的表与表之间也会反应出现实中实体与实体之间的关系。

这样的关系可以分为以下三类：

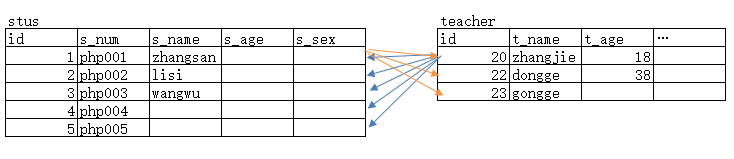
一对一



一对多



多对多



五、高级操作

1、复制表结构

语法：

create table 表B like 表A;

说明：  
 使用表A的结构重新创建一个表B

只是复制结构，不包含表的数据。



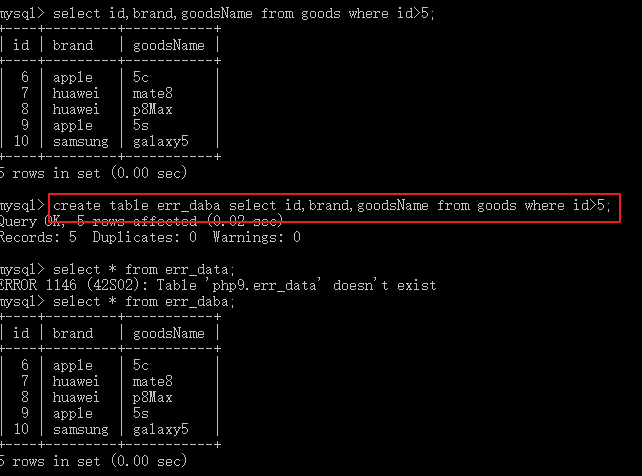
2、备份SQL执行结果

语法：

create table 表名 select 语句;

说明：

将一条select语句得到的结果保存到一个新创建的表中。



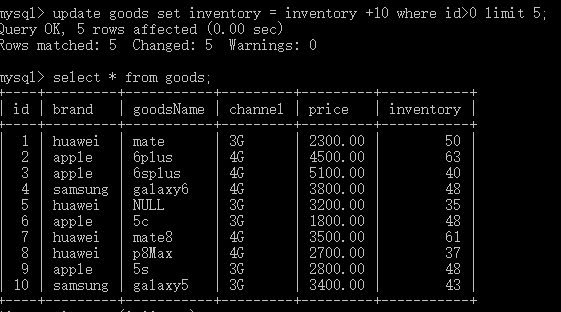
3、限制更新

语法：

update 表名 set 字段=值 【where子句】【limit子句】

说明：

用于将where子句匹配到记录，仅更改limit子句限制的条数。



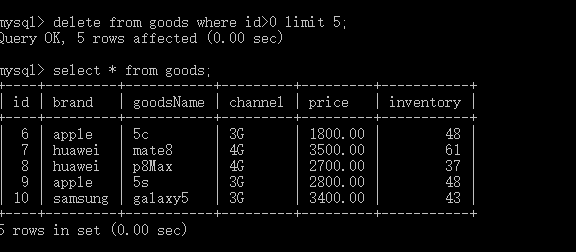
4、限制删除

语法：

delete from 表名 【where子句】【limit子句】

说明：

用于将where子句匹配到记录，仅删除limit子句限制的条数。



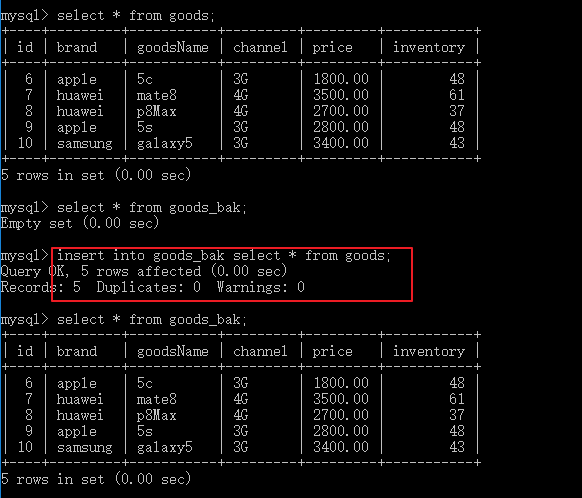
5、批量插入

语法：

insert into 表A【(字段列表)】 select 字段列表 from 表B;

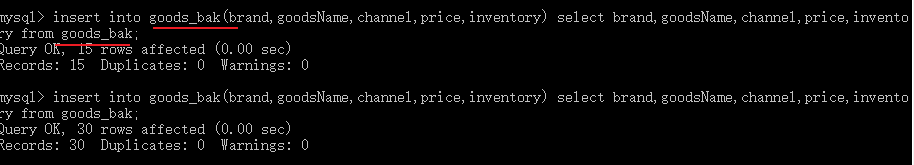
说明：

将select语句得到的数据，插入到表A中。



6、蠕虫复制

蠕虫复制是批量插入的一个特殊情况，在使用批量插入语句时，如果插入的表与数据查询的表是同一个表，那么这种情况，术语就叫蠕虫复制。



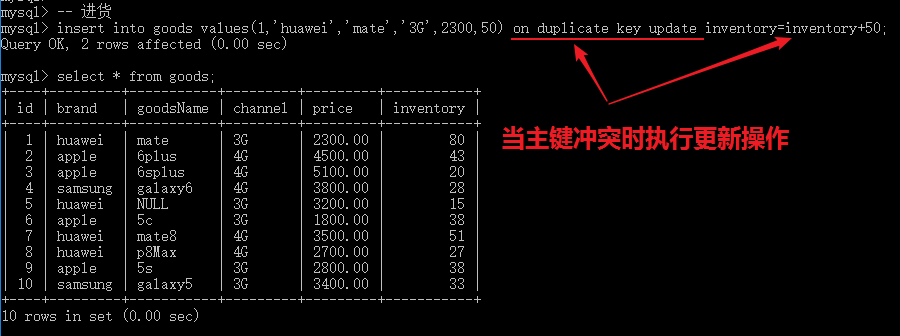
7、主键冲突

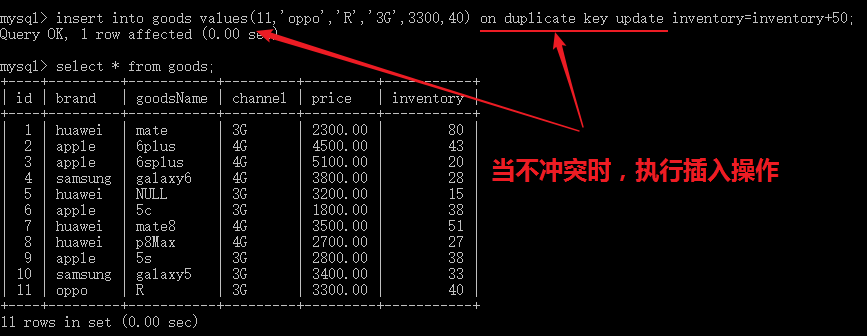
主要讲述的就是主键冲突的解决方案》

①、冲突更新。

语法：

insert into 表名【(字段列表)】 values(值列表) **on duplicate key update** 字段=值,字段=值…;





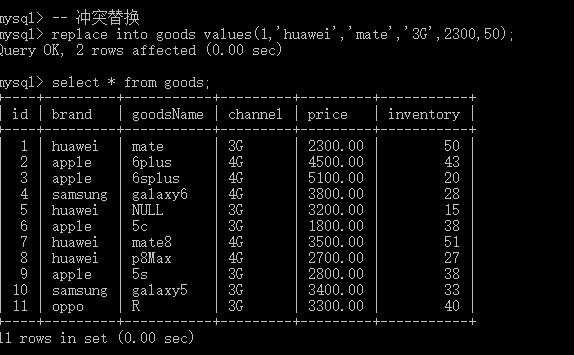
②、冲突替换

语法：

replace into 表名【(字段列表)】 values(值列表)

说明：

如果主键冲突，会将所有的字段替换掉



8、清空表

语法：

delete from 表名;

说明：

只会将表中的数据删除，并不会重建索引。

语法：

truncate 表名;

说明：  
 不但会删除表中的数据，而且还会重建索引。

