**MySQL数据库技术**

目录

[二、 数据库操作 4](#_Toc495419846)

[1. 查看数据库 4](#_Toc495419847)

[1) 查看数据库服务器中的全部数据库 4](#_Toc495419848)

[2) 查看数据库的创建语句 5](#_Toc495419849)

[2. 创建数据库 5](#_Toc495419850)

[1) 创建语法 5](#_Toc495419851)

[2) 数据库名的命名规范 6](#_Toc495419852)

[3) 注意事项 6](#_Toc495419853)

[3. 删除数据库 6](#_Toc495419854)

[4. 修改数据库 7](#_Toc495419855)

[5. 选择数据库 7](#_Toc495419856)

[三、 数据表操作 8](#_Toc495419857)

[1. 查看数据表 8](#_Toc495419858)

[1) 查看当前数据库中的全部数据表 8](#_Toc495419859)

[2) 查看表结构 9](#_Toc495419860)

[2. 字段类型初步介绍 10](#_Toc495419861)

[1) 整型 10](#_Toc495419862)

[2) 小数类型 10](#_Toc495419863)

[3) 字符串型 10](#_Toc495419864)

[4) 日期时间型 11](#_Toc495419865)

[3. 创建数据表 11](#_Toc495419866)

[1) 语法 11](#_Toc495419867)

[2) 需求：创建学生就业信息表，存储：学号,姓名，性别，公司名称，就业薪资，毕业时间 12](#_Toc495419868)

[3) 注意事项 12](#_Toc495419869)

[4. 案例：创建学生信息表 12](#_Toc495419870)

[5. 列属性介绍 13](#_Toc495419871)

[1) null,not null 13](#_Toc495419872)

[2) default 13](#_Toc495419873)

[3) primary key 13](#_Toc495419874)

[4) auto\_increment 14](#_Toc495419875)

[5) unique key 14](#_Toc495419876)

[6) comment 14](#_Toc495419877)

[7) 需求：创建学生就业信息表，存储：学号,姓名，性别，公司名称，就业薪资，毕业时间 14](#_Toc495419878)

[6. 查看数据表的创建语句 15](#_Toc495419879)

[7. 删除数据表 15](#_Toc495419880)

[8. 案例：创建商品表 16](#_Toc495419881)

[9. 案例：订单表 16](#_Toc495419882)

[四、 数据操作初步 17](#_Toc495419883)

[1. 插入数据 17](#_Toc495419884)

[1) 注意事项： 17](#_Toc495419885)

[2. 查询数据 18](#_Toc495419886)

[3. 修改数据 19](#_Toc495419887)

[4. 删除数据 20](#_Toc495419888)

[1) drop 与delete区别： 20](#_Toc495419889)

[2) where条件 20](#_Toc495419890)

[五、 查询高级操作 20](#_Toc495419891)

[1. where子句 20](#_Toc495419892)

[2. group by 子句 21](#_Toc495419893)

[3. limit 子句 22](#_Toc495419894)

[4. 注意事项 24](#_Toc495419895)

[六、 连接查询与子查询 24](#_Toc495419896)

[1. 连接查询 24](#_Toc495419897)

[1) 查询所有苹果产品分类以及对应的商品信息 24](#_Toc495419898)

[2) 查询所有的学生基本信息以及院系信息 25](#_Toc495419899)

[3) 查询高端设备的基本信息以及分类信息 25](#_Toc495419900)

[4) 查询江西学生的基本信息以及院系信息 25](#_Toc495419901)

[2. 子查询 26](#_Toc495419902)

[1) 查询价格高于ipad pro的商品信息 26](#_Toc495419903)

[2) 查询杨洋的老乡信息 26](#_Toc495419904)

[3) 子查询返回一列多行的情况 27](#_Toc495419905)

[七、 navicat for mysql的使用 27](#_Toc495419906)

[1. 建立连接 27](#_Toc495419907)

[2. 创建数据库 28](#_Toc495419908)

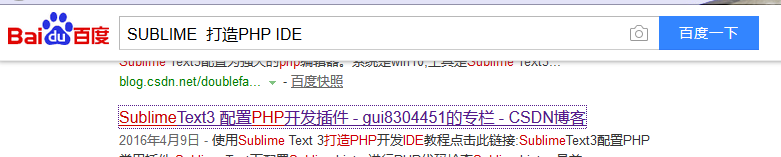
[3. 创建数据表 29](#_Toc495419909)

[4. 插入数据 30](#_Toc495419910)

[5. 查看数据表的结构 30](#_Toc495419911)

[6. 执行SQL语句 30](#_Toc495419912)

**sublime配置CTRL + SHIFT + P（**emmet，SublimeLinter，DocBlockr，SublimeTmpl**），**

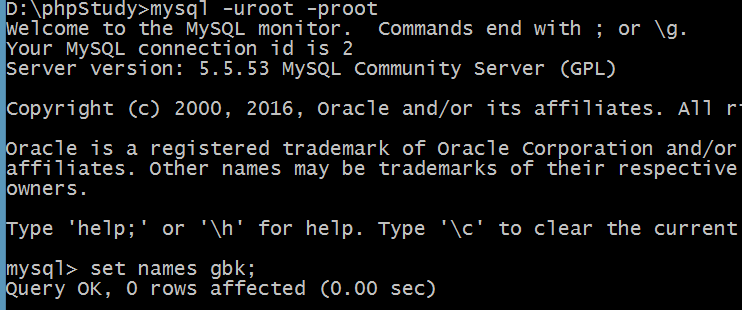


**思考的问题：**

1. **如何对数据库进行增删改查？**
2. **如何对数据表进行增删改查？**
3. **如何对数据进行增删改查？**
4. **MySQL的数据类型有哪些？**
5. **MySQL的字段属性有哪些？**
6. **查询语句有哪些子句可以使用？**
7. **子查询和连接查询的使用场景以及语法？**

# 数据库操作

先连接数据库：

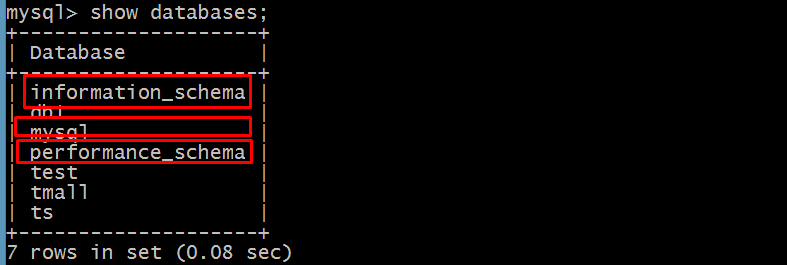


数据库服务器中一般存储多个数据库，每个数据库一般对应单个项目。

## 查看数据库

### 查看数据库服务器中的全部数据库

**show** databases;



注意事项：

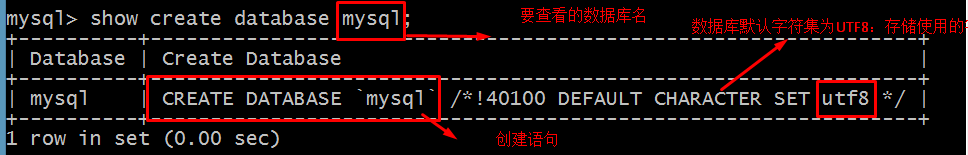
* SQL语句：使用英文状态的分号，作为语句结束标志。
* 系统数据库（information\_schema,performance\_schema,mysql）不能删除。

### 查看数据库的创建语句

show create database 数据库名;

显示信息包括：数据库的创建语句及数据库选项（包括字符集）。

默认字符集：数据库数据存储默认使用UTF8(对中文进行编码)该字符集。

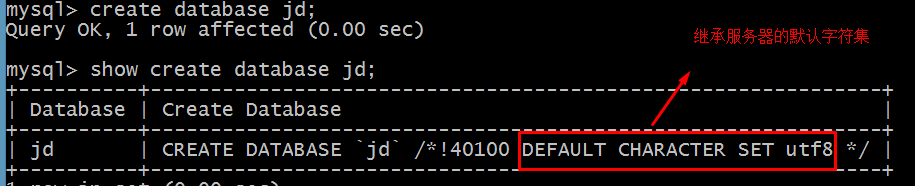


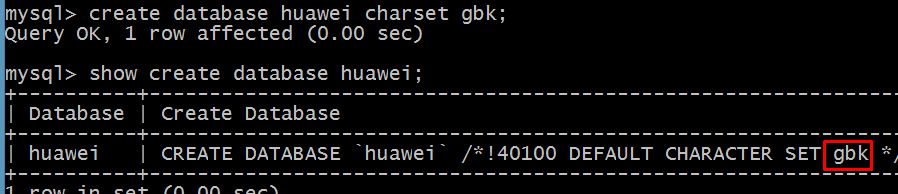
## 创建数据库

创建数据库，用来保存项目数据。

### 创建语法

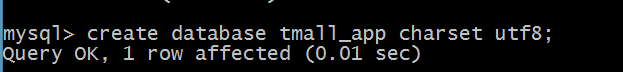
create database 数据库名 [charset 字符集];





### 数据库名的命名规范

* 字母、数字、下划线组成。
* 命名见名知意。数据表表名或者数据库的库名使用下划线命名法。可以添加项目前缀。
* 长度不超过32个字符，便于使用。

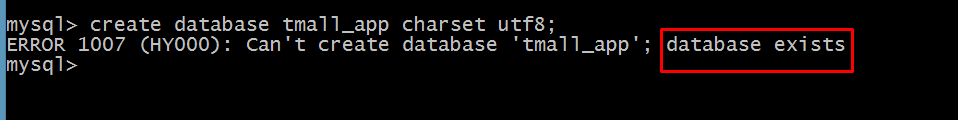


### 注意事项

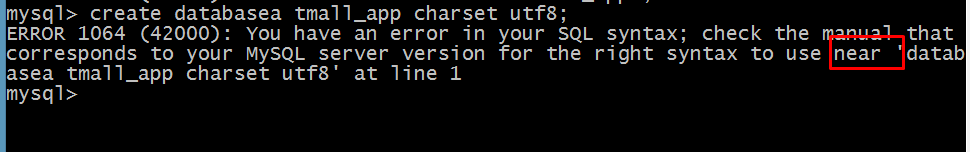
数据库选项包括字符集及校对规则。不指定时，使用默认的字符集及校对规则。

常见字符集：ASCII（英文字母，数字），GBK（国标扩展中文简体字符集，big5,繁体中文），UTF8（三码合一，数据库、PHP脚本、浏览器显示为UTF8）。

* 已经存在的数据库，无法再次创建。



* 查找错误提示：从near开始



## 删除数据库

drop database 数据库名;

删除数据库（删除结构），全部数据丢失。

提示：不能删除系统数据库（information\_schema, mysql, performance\_schema）,

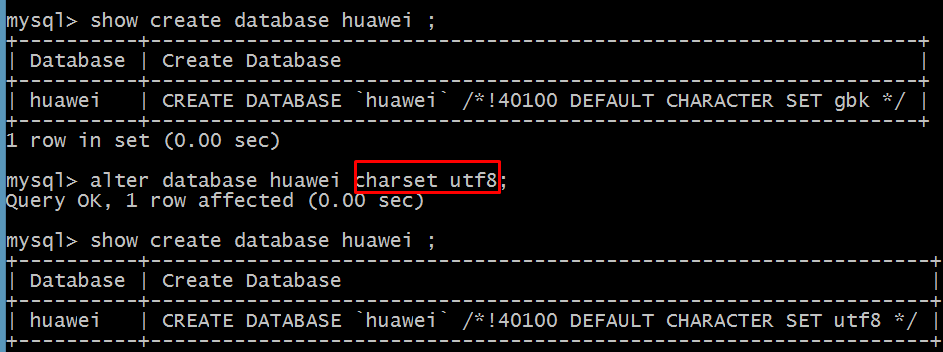
* 删除数据库：数据库中的所有数据表被删除(存储的数据丢失)，需要先备份。



## 修改数据库

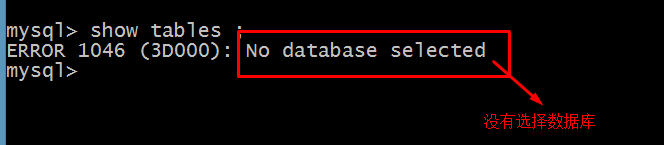
alter database 数据库名 charset 新字符集;

数据库名称不能修改。修改数据库的主要操作为修改字符集(修改数据源存储的字符集)。

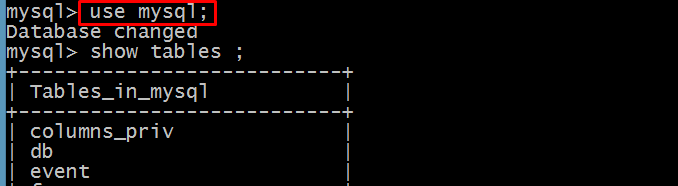


## 选择数据库

数据真正存储的地方：数据表。一般先选择数据库(数据表所在的数据库)，然后才能对其中的数据表进行操作。



use 数据库名;



# 数据表操作

电商tmall数据库：商品表goods，用户表user，收货地址表address，商铺表shop，订单表order，评论表comments

真正存储数据的地方：数据表，table。项目开发一大重要的技能：将项目的数据,使用合理的数据表存储在数据库中。

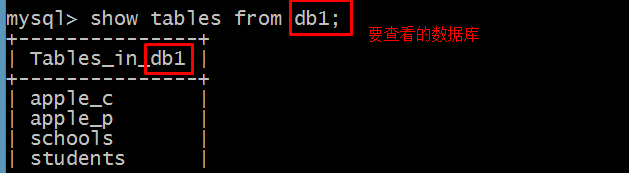
## 查看数据表

### 查看当前数据库中的全部数据表

show tables;



* 查看前，需要首先指定要查看的数据表所在的数据库。否则：**no database selected**
* 未选择数据库，可以通过from指定数据库名称



### 查看表结构

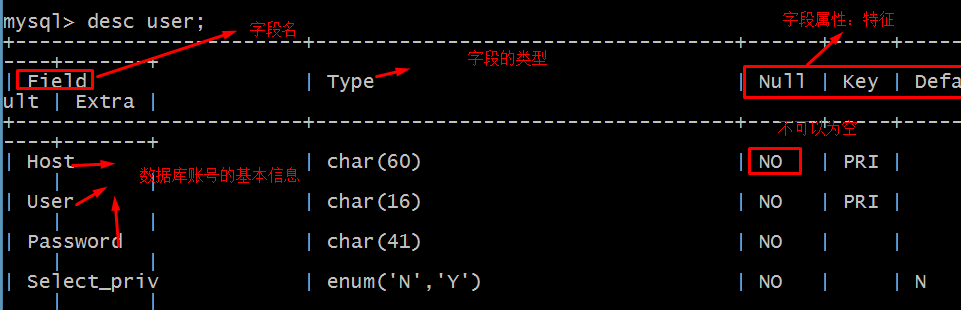
关系型数据库的数据表是一个二维结构。由行和列组成，需要根据存储数据的情况，限定数据表中，

* 存储实体的哪些信息（字段，年龄，姓名，身高），
* 存储什么类型的数据（字段类型），
* 数据有哪些特征（字段属性，字段是否可以为空,是否可以重复（如订单编号））。

**订单信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **订单编号order\_id** | **商品名称**  **goods\_name** | **价格**  **price** | **数量**  **goods\_num** | **配送地址**  **post\_address** |
| 100123 | 华为mate10(字符串类型) | 4999 | 2（整型） | 北京市京顺路99号 |
| 100124 | 花裙子 | 499 | 1 | 北京市清华大学1号楼 |

desc 数据表名；



## 字段类型初步介绍

### 整型

存储如年龄，产品数量，编号等。

int（21亿） 占据4个字节的空间，存储有符号数（可存储负数）

tinyint(-128,127) 占据1个字节的空间。

### 小数类型

float（单精度浮点数） ,decimal（定点类型） eg.重量，工资，奖金，价格等

float存储小数类型存在精度的问题。 4字节。

对应精度要求比较高的，使用decimal类型，实现小数的精确存储。其原理为可以存储每位数字。一般用来存储与钱有关的数字。

### 字符串型

varchar(M)，char(M)

* M为该字段可以存储的最多字符数，如varchar(10)。M参数不可以省略，此外不能超出指定的长度。
* varchar一般用来存储长度变化比较大的字符串，如文章标题，商品名称，速度慢。
* char存储长度比较固定的字符串，如手机号，身份证号，序列号，邮编，速度快
* 可以使用text类型，存储较长的字符串，无需指定字符串的具体长度。

### 日期时间型

PHP时间：time()---时间戳，10位整型数字

* datetime年月日时分秒，
* date(年月日)，
* time（时分秒）。

也可以使用PHP获取时间戳，按照整型数字存储（date函数进行格式化）。

## 创建数据表

### 语法

create table 数据表名(

字段1 类型 [属性] ,

字段2 类型 [属性],

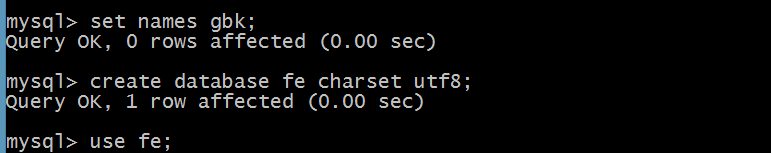
…

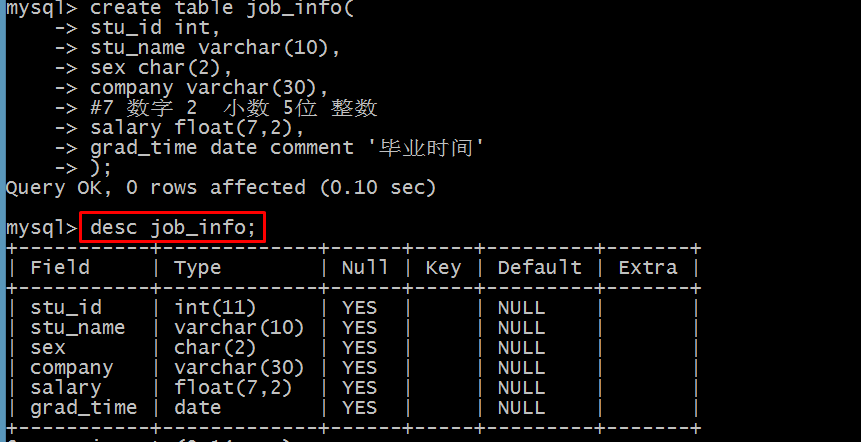
字段n 类型 [属性]

);

创建数据表的核心是定义数据存储的结构：字段名称 类型 [属性]

### 需求：创建学生就业信息表，存储：学号,姓名，性别，公司名称，就业薪资，毕业时间





### 注意事项

* 创建数据表之前,需要先分析数据表字段的个数，类型及属性。数据表的创建与字段的定义同时进行。
* 字段的定义包括：字段名称 字段类型 字段属性等
* 不同的字段定义之间，使用英文的逗号隔开
* 最后一个字段定义与括号之间不能有逗号

## 案例：创建学生信息表



## 列属性介绍

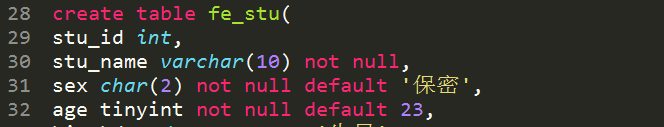
字段数据的属性（特征），限定其满足某些特征。

### null,not null

限制字段值不能为空

### default

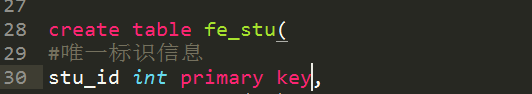
设置字段的默认值，在没有录入时自动使用默认值填充。



### primary key

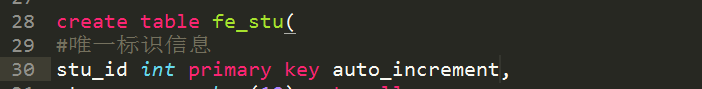
设置字段为主键，主键字段的值不能重复，不能为空。

一个数据表中只能设置一个字段为主键，作为每行记录的唯一身份信息（索引），通过索引可以快速找到数据在磁盘的位置，加快数据的读取速度。



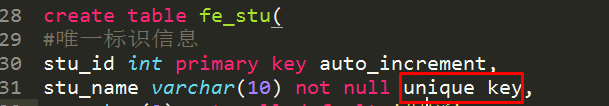
### auto\_increment

* + 设置字段为自动增长，默认从1开始自动分配编号。类型必须为整型
* 自增长字段必须为一个key（索引，如primary key，主键索引，是一种特殊的数据结构，便于快速查找数据，相当于书的目录）。



### unique key

* + 唯一键，设置字段的值为唯一的，可以同时设置多个字段为唯一键。
  + 唯一键字段的值可以为空。

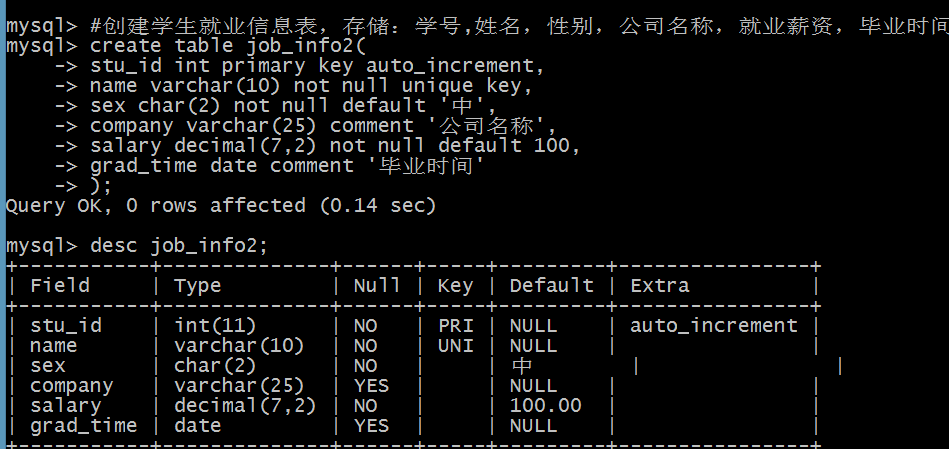


如商品的名称，文章的名称，都可以通过唯一键来进行限制。

### comment

字段注释

### 需求：创建学生就业信息表，存储：学号,姓名，性别，公司名称，就业薪资，毕业时间



## 查看数据表的创建语句

查看数据库的创建语句：

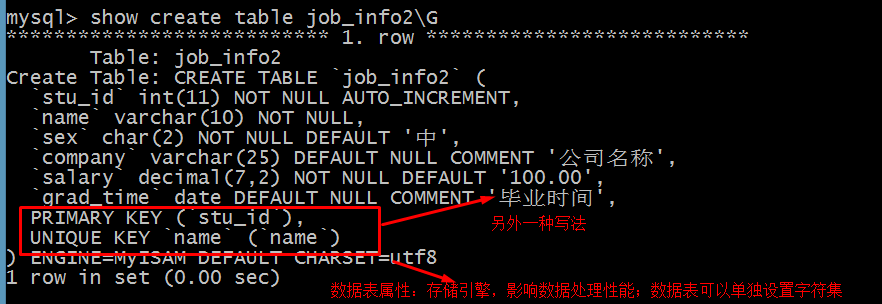
show create database 数据库名；

查看数据表的创建语句：

show create table 数据表名;

//；可以使用\G替换，在命令中显示更清晰

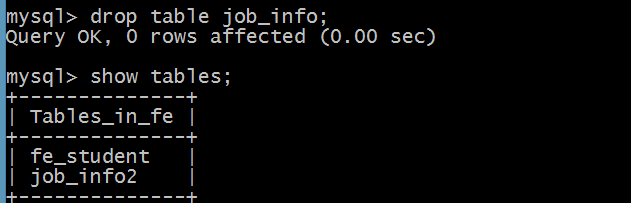
可以查看数据字段定义以及字符集设置等信息。



## 删除数据表

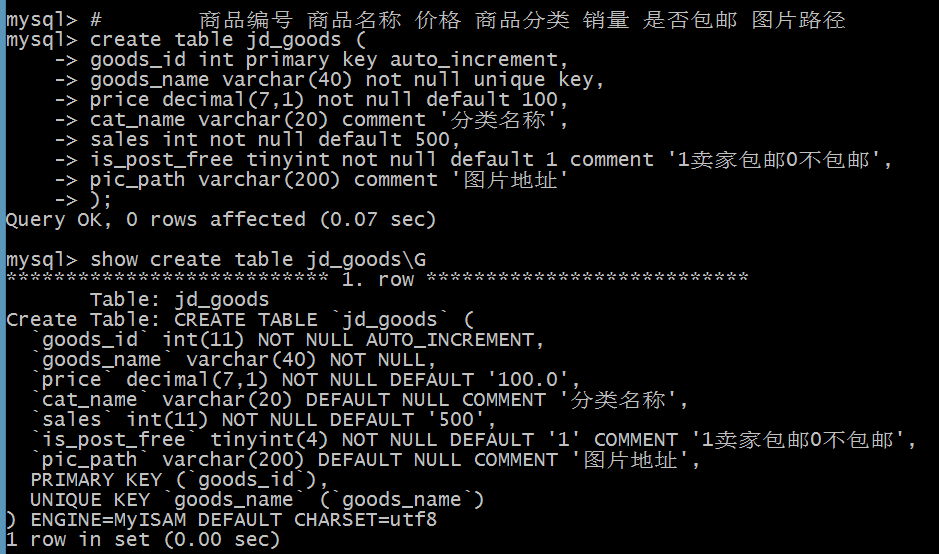
drop table 数据表表名；

删除数据表的结构定义，会导致数据表结构文件丢失，同时数据表中的全部数据丢失。删除之前，做好数据的备份工作。

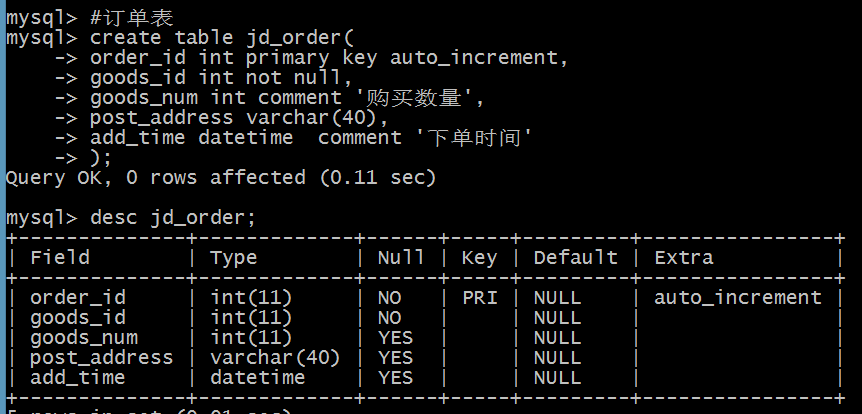


## 案例：创建商品表

商品编号 商品名称 价格 商品分类 销量 是否包邮 图片路径



## 案例：订单表



# 数据操作初步

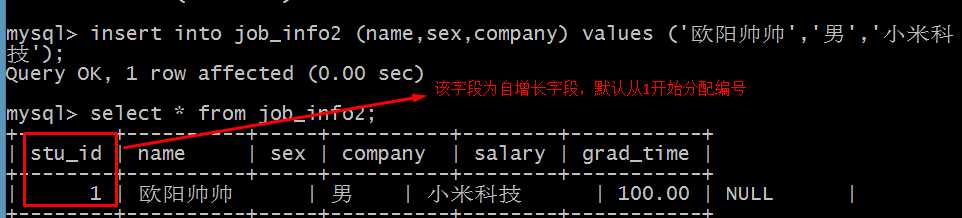
基本操作：CRUD，

* **增**（C,create,insert）,
* **删**(D,delete),
* **改**(U,update)，
* **查**（Retrieve，select）。

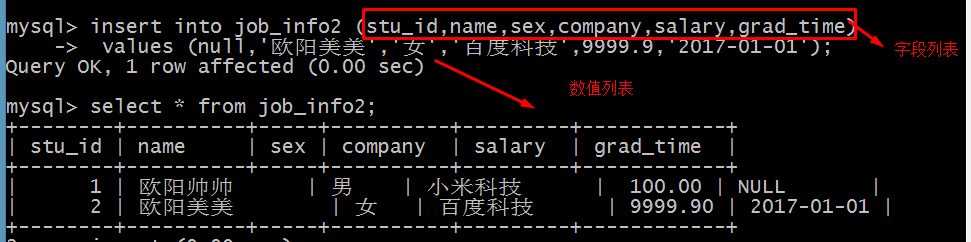
## 插入数据

**insert** into 数据表 （字段1，字段2）values (值1，值2)；

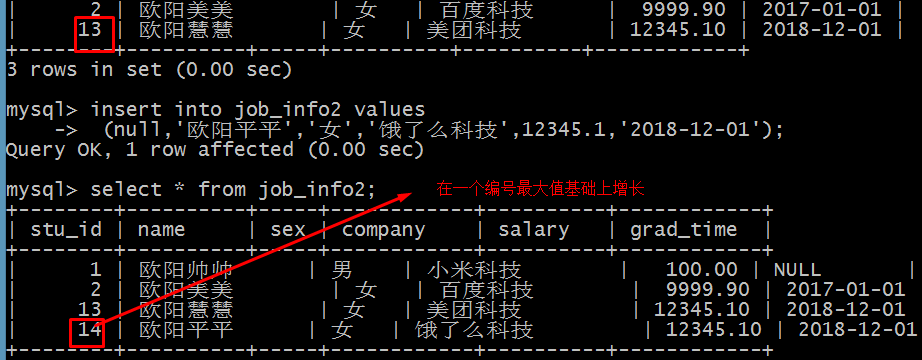
### 注意事项：



* 插入数据时，字段列表与数据列表要一一对应。
* 插入数据的类型，需要与字段设置的类型一致。数据应该不超出字段的表示范围。
* 字符串类型必须使用单引号或者双引号包括。日期时间型也需要使用单双引号，日期类型应该是合法的数据。



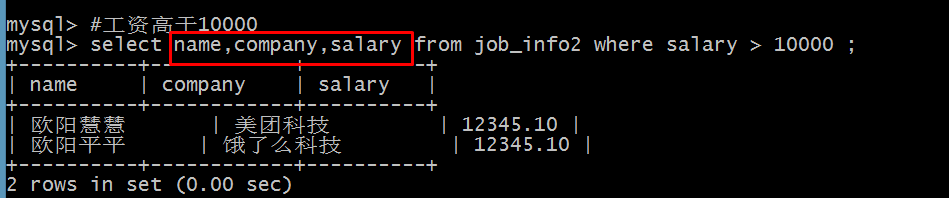
* 字段列表中，字段出现的顺序不需要与数据表中字段的顺序一致。
* 插入全部字段信息时，可以省略字段列表。此时，必须按照数据表中字段出现的顺序录入。



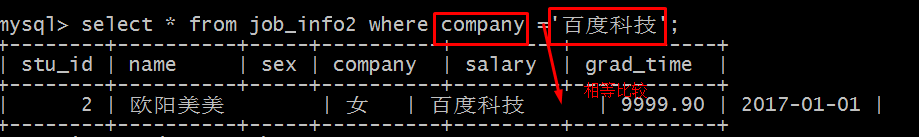
## 查询数据

select 字段1，字段2 | \* from 数据表名 [where 条件]；

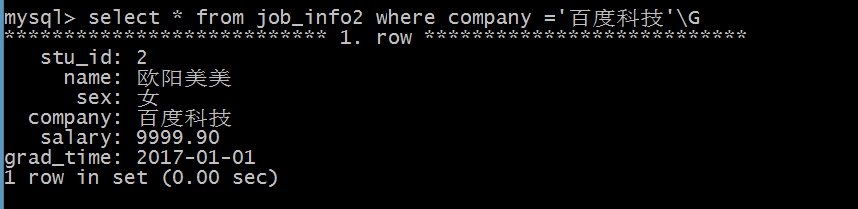
* \*代表查看全部字段的信息，返回数据表的全部字段的信息
* 所得即所需。可以指定查询的具体信息（字段列表，或者通过where条件限定），实现仅查询需要的数据，减少数据的请求量。



* where执行过程:逐一扫描磁盘上数据表中的所有行，满足限定条件，该行在结果集中显示。

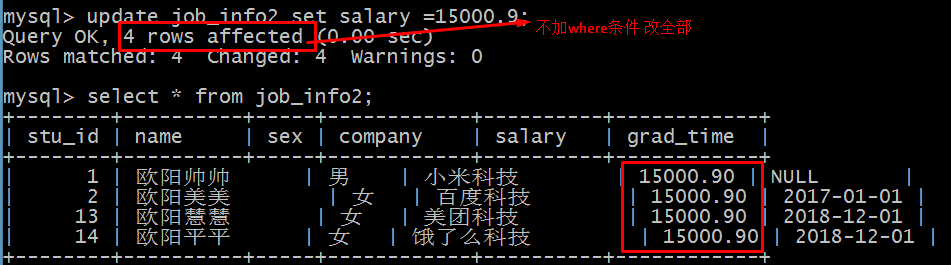


* 对于字段显示的信息较多时，分号可以使用\G替换，显示更清晰。

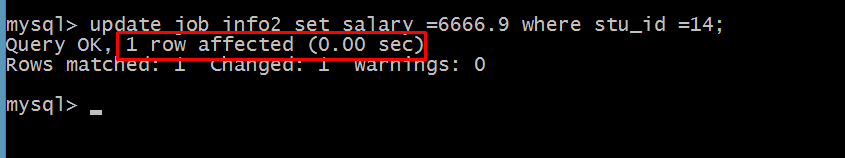


## 修改数据

**update** 数据表 set 字段1 =值1,字段2 =值2 where 条件；



* 不添加where条件，修改数据表的全部行对应字段的信息。
* 更新语句，通常需要添加where限定条件。

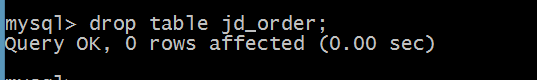


## 删除数据

删除数据表中的指定行，仅对数据操作，不对表结构进行操作。

**delete** from 数据表 [where 条件]；

### drop 与delete区别：

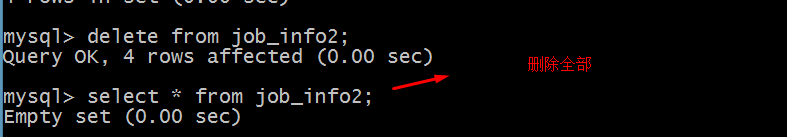


**drop**:SQL的结构定义语句（DDL），整个数据表消失。

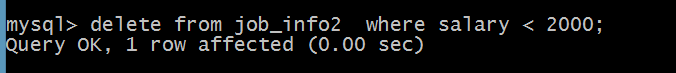
**delete**:SQL的数据操作语言（DML）,一般删除指定行。

### where条件

* 不添加where条件，删除数据表中的所有行。数据表的结构依然存在。



* 添加where条件，删除满足条件的所有行。

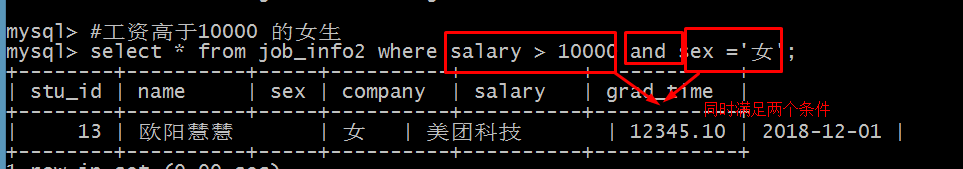


# 查询高级操作

## where子句

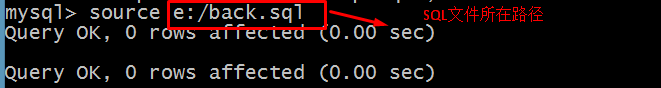
查询时，不添加where条件返回数据表所有行。需要添加限定条件，只返回需要的行。

select 字段列表 from table where子句；



## group by 子句

导入测试数据：



**导入之后的情况：**





select 字段列表 from table [where 子句] group by 分组字段；

按照字段分组，再统计各个分组的某些方面数据，如年龄的最大值，分数的最小值等。

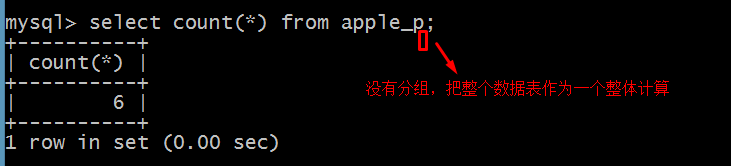
一般与聚合函数共同使用。

count(字段):返回该分组的行数

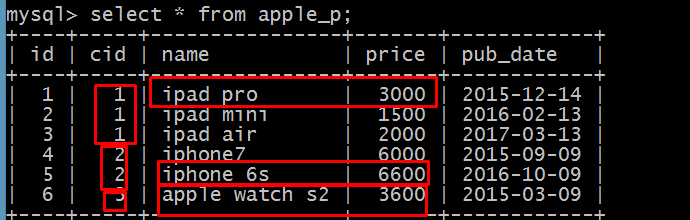
max(字段)：返回分组字段的最大值

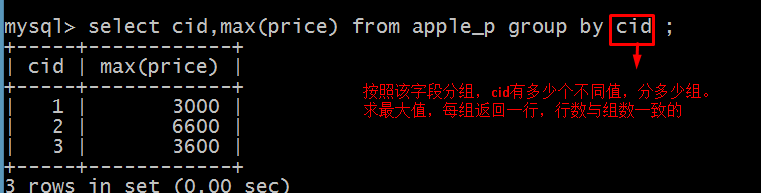
min(字段)：返回分组字段的最小值

**统计各个分类的商品数目：**



**统计各个分类价格最高的商品价格：**





## limit 子句

满足条件的数据可能很多，通过limit子句，限定返回的是哪些行的数据。减少数据的传输，便于用户查看数据。



select 字段列表 from table limit 偏移量offset，每页显示的行数pagesize

对于一个页面，每页显示的行pagesize一般为固定的。

偏移量：偏移指定行之后，再开始显示。

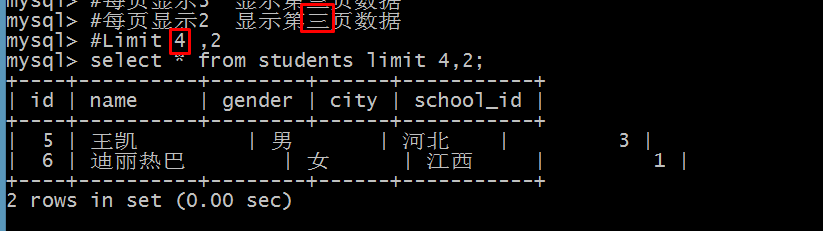
假设每页显示10行，当前页码为page

显示第二页 **偏移**2-1页数据（2-1） \*10 =10

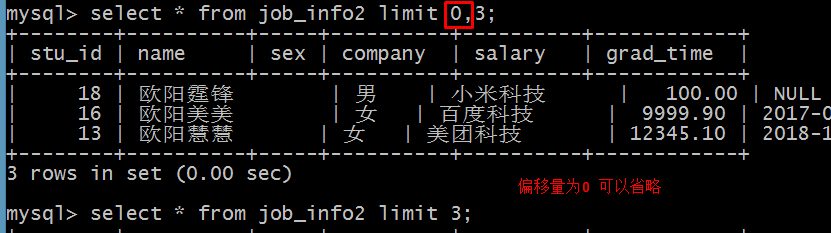
显示第三页 **偏移**3-1页数据（3-1） \*10 =20

第n页，偏移前n-1页已经显示的内容。

offset =(page -1 ) \* pagesize







## 注意事项

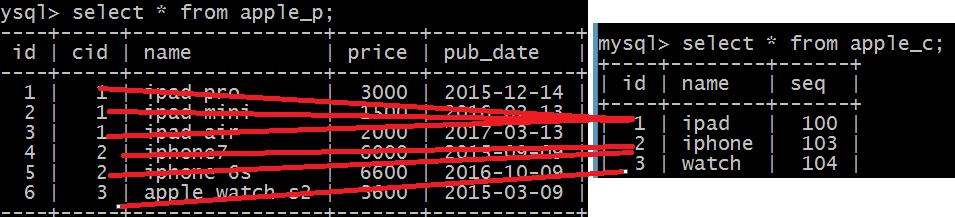
where子句、group by子句、limit子句可以同时使用。需要注意各个子句之间的顺序。

select 字段列表 from table\_name [where 子句] [group by 子句] [limit 子句]

# 连接查询与子查询

## 连接查询

关系型数据库，数据分布在不同的数据表中。每个数据表中只存储某一方面的信息，需要获取比较完整的信息，需要将不同的数据表连接起来查询。



select 字段1 as 别名1，字段2 from A join B

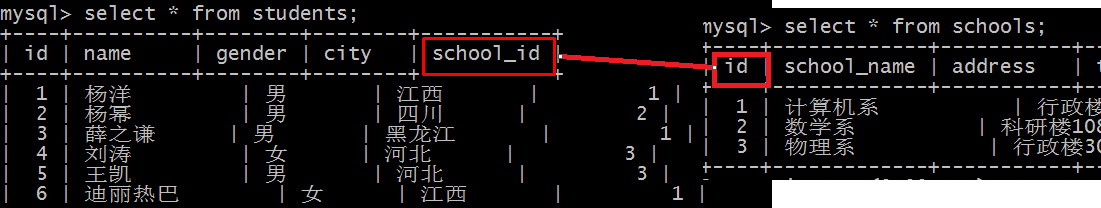
on A.字段 = B.字段 [where子句];

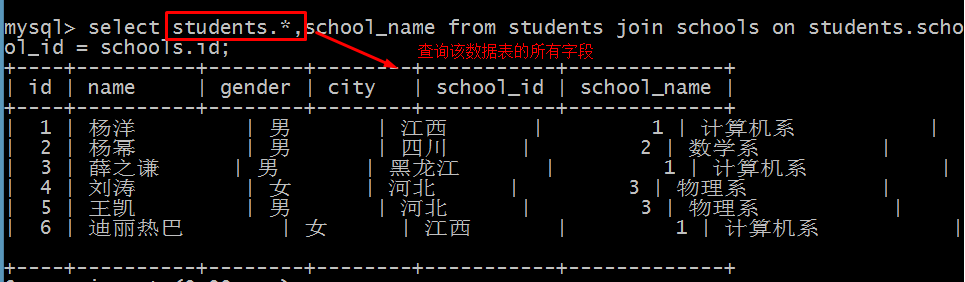
### 查询所有苹果产品分类以及对应的商品信息





### 查询所有的学生基本信息以及院系信息



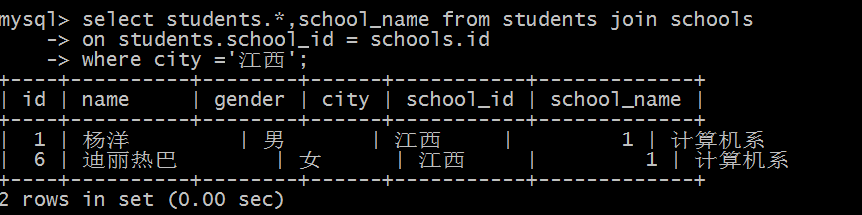


### 查询高端设备的基本信息以及分类信息

价格大于3000



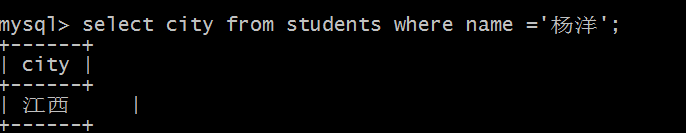
### 查询江西学生的基本信息以及院系信息



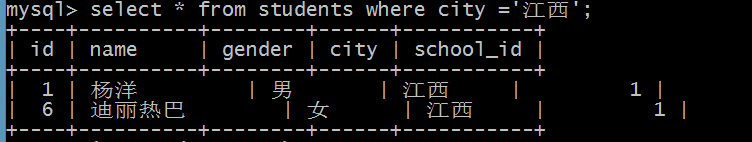
## 子查询

查询杨洋的老乡信息

1. 查询杨洋的籍贯



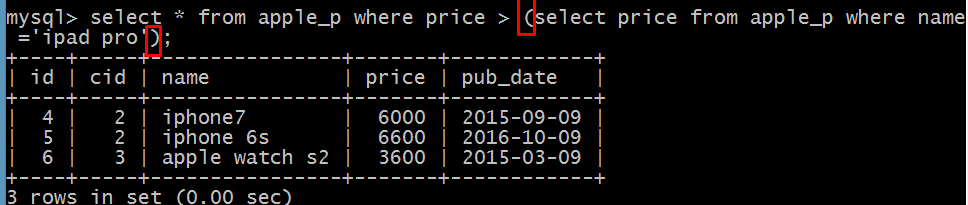
1. 查询籍贯与杨洋籍贯一致的学生信息



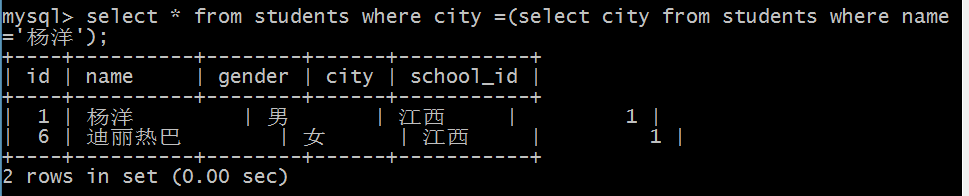
**子查询**就是一个查询的结果作为另外一个查询的条件使用。外层的查询称为父查询，作为条件的查询称为子查询（subquery）。

select 字段列表 from table where … （子查询）；

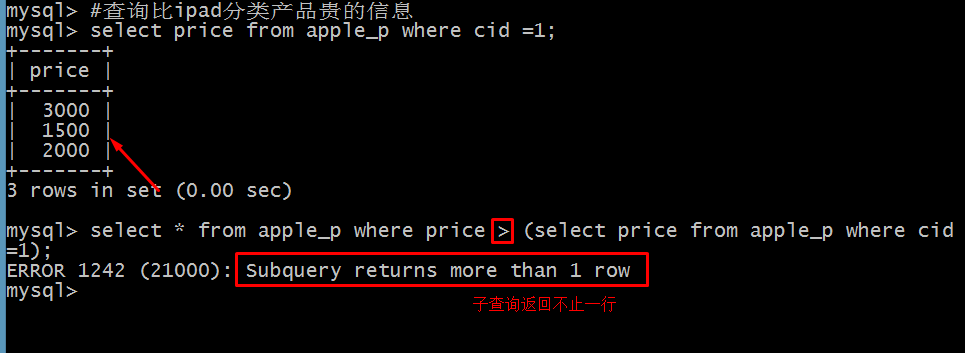
### 查询价格高于ipad pro的商品信息

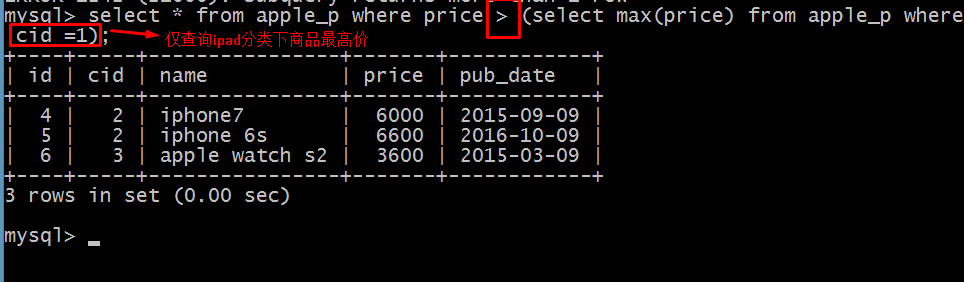


### 查询杨洋的老乡信息



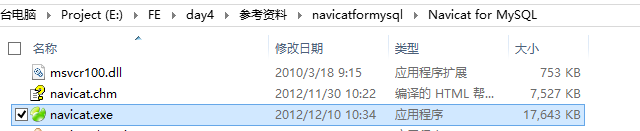
### 子查询返回一列多行的情况



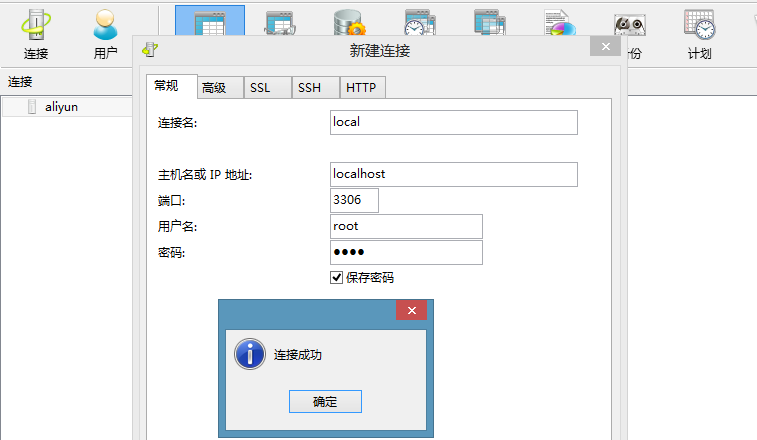


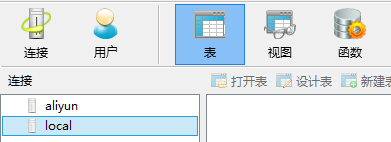
# navicat for mysql的使用

navicat for mysql是一个专业的MySQL数据库图形化界面工具。绿色软件。

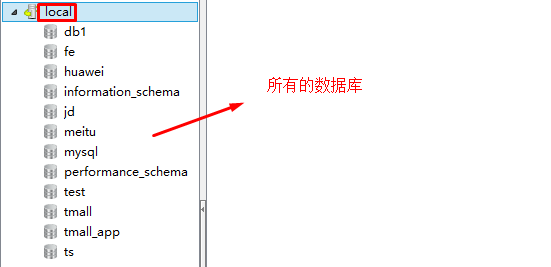


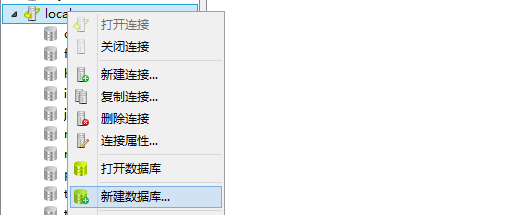
## 建立连接

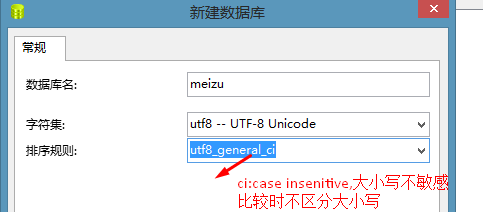


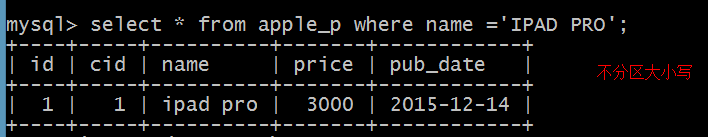


## 创建数据库







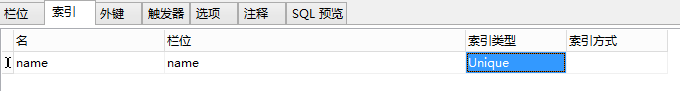


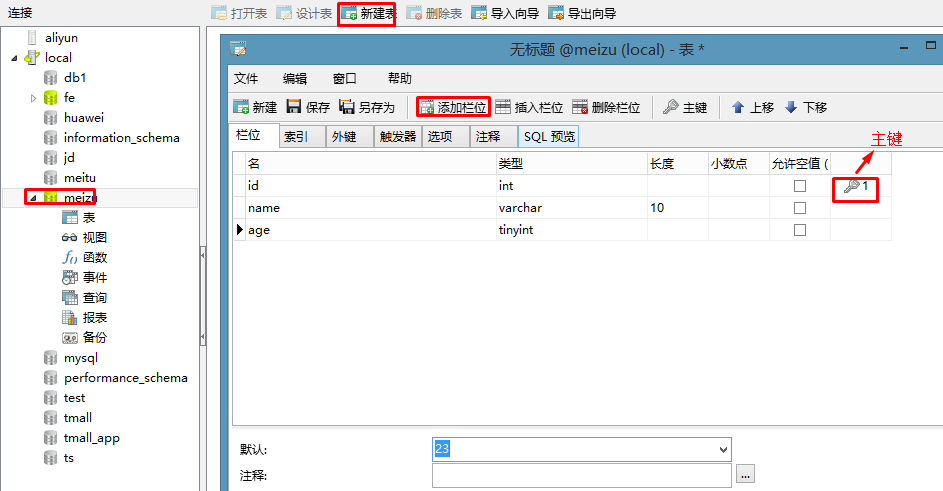
## 创建数据表

新建表—新建---设置字段名、类型及长度---保存—设置表名

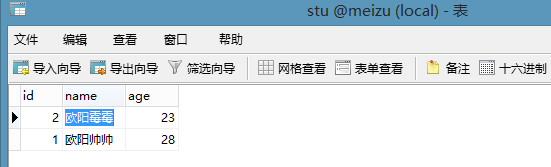
设置唯一键：





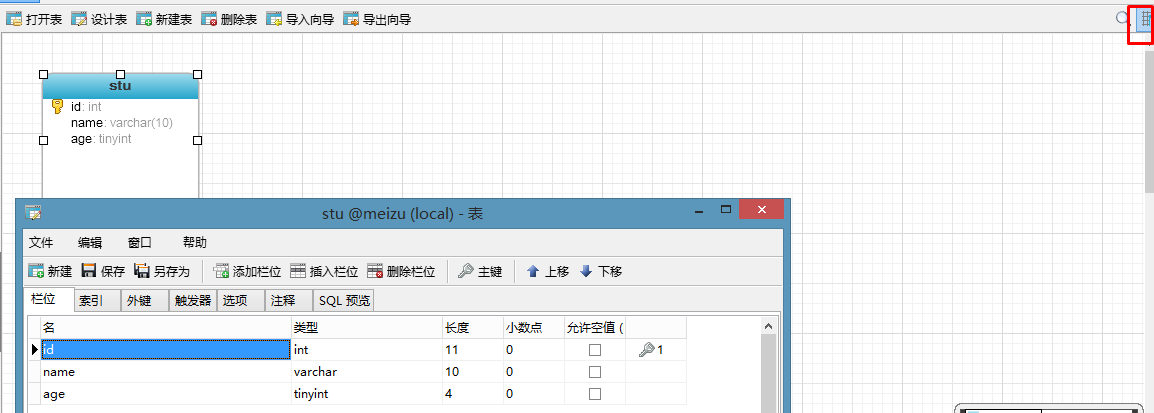


## 插入数据



## 查看数据表的结构

主界面右上角的E-R图标 — 双击



## 执行SQL语句

