# 数据库介绍

## 数据库概述

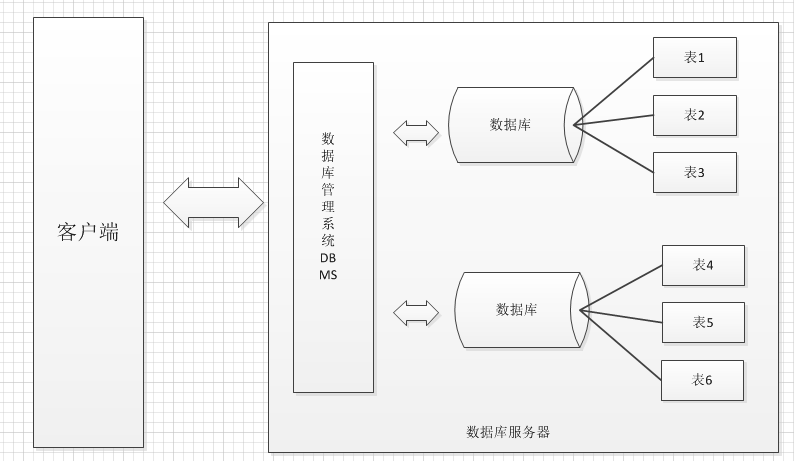
* 什么是数据库(DB:DataBase)

数据库就是存储数据的**仓库**，其本质是一个**文件系统**，数据按照特定的格式将数据存储起来，用户可以对数据库中的数据进行增加，修改，删除及查询操作。

* 什么是数据库管理系统

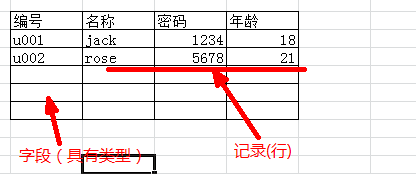
数据库管理系统（DataBase Management System，DBMS）：指一种操作和管理数据库的 大型**软件**，用于建立、使用和维护数据库，对数据库进行统一管理和控制，以保证数据库的安全性和完整性。用户通过数据库管理系统访问数据库中表内的数据。

* 数据库与数据库管理系统的关系

n

## 数据库表

数据库中以表为组织单位存储数据。表中有行和列,我们叫做字段和记录



## 表数据

表类似我们的Java类，每个字段都有对应的数据类型。

那么用我们熟悉的java程序来与关系型数据对比，就会发现以下对应关系。

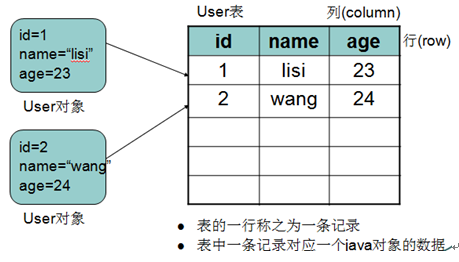
类----------表

类中属性----------表中字段

对象----------记录

根据表字段所规定的数据类型，我们可以向其中填入一条条的数据，而表中的每条数据类似类的实例对象。表中的一行一行的信息我们称之为记录。

* 表记录与java类对象的对应关系



## 常见数据库

* 常见的数据库管理系统

MYSQL ：开源免费的数据库，小型的数据库.已经被Oracle收购了.MySQL6.x版本也开始收费。

Oracle ：收费的大型数据库，Oracle公司的产品。Oracle收购SUN公司，收购MYSQL。

DB2 ：IBM公司的数据库产品,收费的。常应用在银行系统中.

SQLServer：MicroSoft 公司收费的中型的数据库。C#、.net等语言常使用。

SyBase ：已经淡出历史舞台。提供了一个非常专业数据建模的工具PowerDesigner。

SQLite : 嵌入式的小型数据库，应用在手机端。

常用数据库：**MYSQL** Oracle

这里使用MySQL数据库。MySQL中可以有多个数据库，数据库是真正存储数据的地方。

# MySql数据库

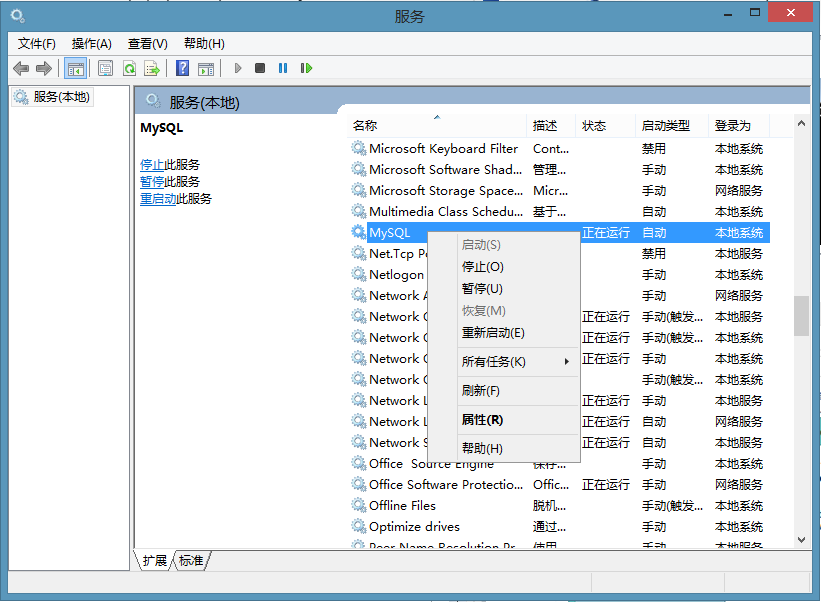
## MySql安装

* 安装

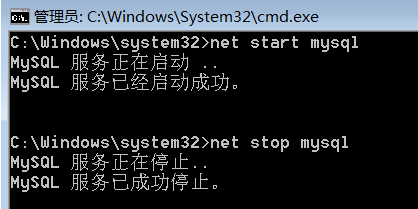
参考MySQL安装图解.doc



安装后，MySQL会以windows服务的方式为我们提供数据存储功能。开启和关闭服务的操作：右键点击我的电脑→管理→服务→可以找到MySQL服务开启或停止。



也可以在DOS窗口，通过命令完成MySQL服务的启动和停止（必须以管理运行cmd命令窗口）

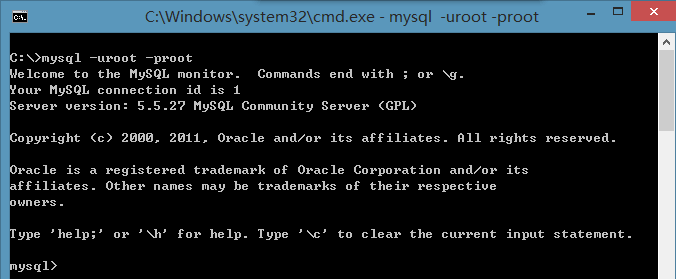


## 登录MySQL数据库

MySQL是一个需要账户名密码登录的数据库，登陆后使用，它提供了一个默认的root账号，使用安装时设置的密码即可登录。

格式1：cmd> mysql –u用户名–p密码

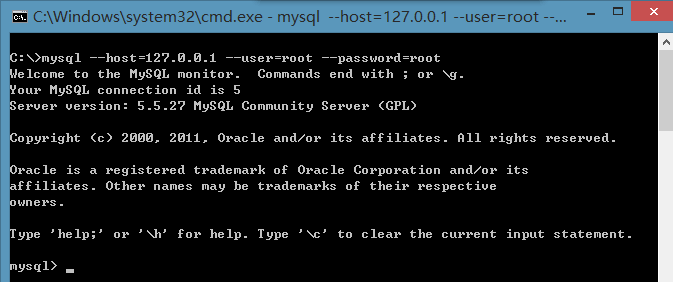
例如：mysql -uroot –proot



格式2：cmd> mysql --host=ip地址 --user=用户名 --password=密码

例如：mysql --host=127.0.0.1 --user=root --password=root

Mysql –h127.0.0.1 –uroot -proot



## SQLyog（MySQL图形化开发工具）

* 安装：

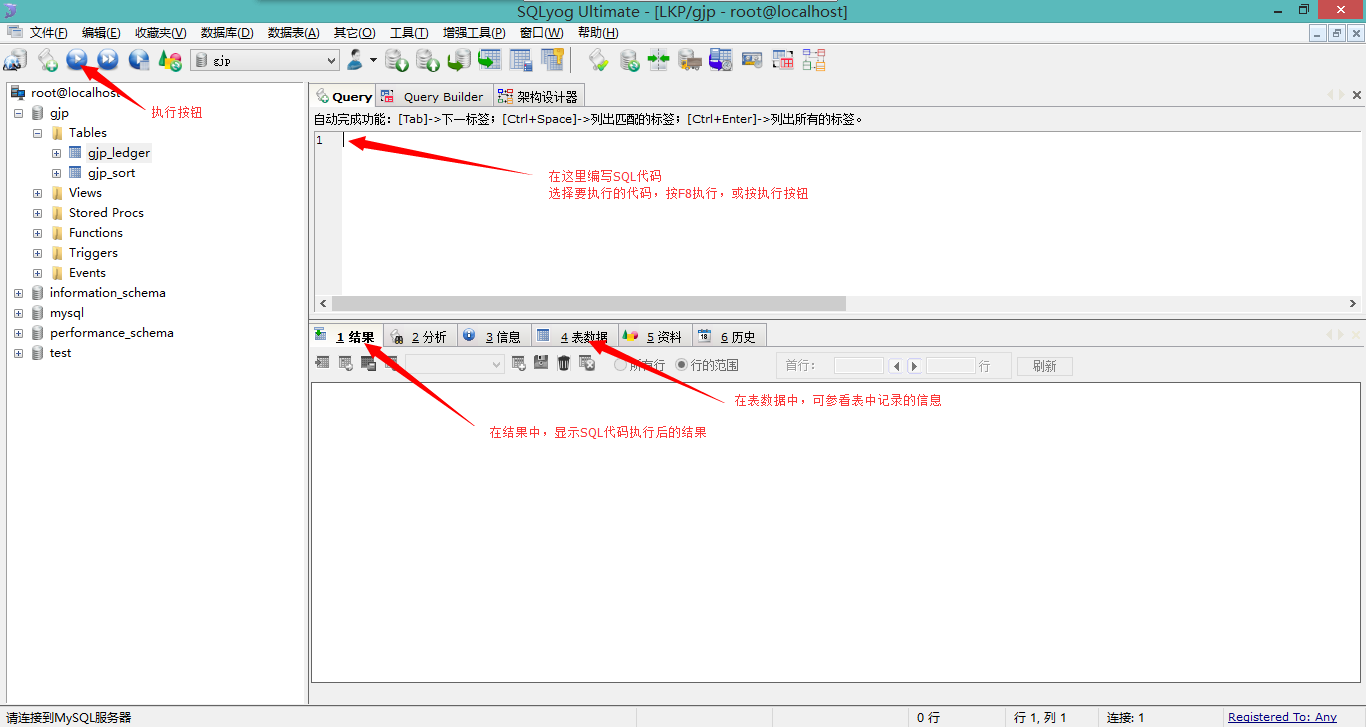
提供的SQLyog软件为免安装版，可直接使用

* 使用：

输入用户名、密码，点击连接按钮，进行访问MySQL数据库进行操作



在Query窗口中，输入SQL代码，选中要执行的SQL代码，按F8键运行，或按执行按钮运行。



# SQL语句

## SQL概述

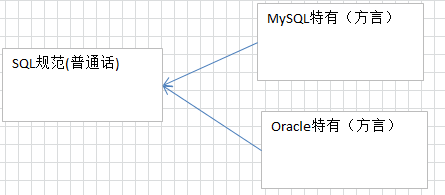
### SQL语句介绍

数据库是不认识JAVA语言的，但是我们同样要与数据库交互，这时需要使用到数据库认识的语言SQL语句，它是数据库的代码。

结构化查询语言(Structured Query Language)简称SQL，是关系型数据库管理系统都需要遵循的规范。不同的数据库生产厂商都支持SQL语句，但都有特有内容。

**普通话：各数据库厂商都遵循的ISO标准。**

**方言：数据库特有的关键字,语法。**

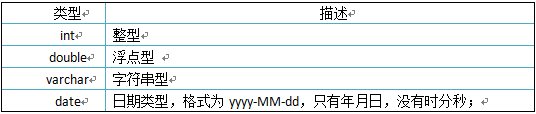


### SQL语句分类

* SQL分类：
  + 数据定义语言：简称DDL(Data Definition Language)，用来定义数据库中的对象：数据库，表，列等。关键字：create，alter，drop, show等
  + 数据操作语言：简称DML(Data Manipulation Language)，用来对数据库中表的记录进行更新。关键字：insert，delete，update等
  + 数据控制语言：简称DCL(Data Control Language)，用来定义数据库的访问权限和安全级别，及创建用户。
  + 数据查询语言：简称DQL(Data Query Language)，用来查询数据库中表的记录。关键字：select，from，where等

### SQL通用语法

* SQL语句可以单行或多行书写，以分号结尾
* 可使用空格和缩进来增强语句的可读性
* MySQL数据库的SQL语句不区分大小写，关键字建议使用大写
  + 例如：SELECT \* FROM user。
* 同样可以使用/\*\*/的方式完成注释 在Sqlyog中也可以是 #或者- -注释一行
* MySQL中的我们常使用的数据类型如下



详细的数据类型如下(不建议详细阅读！)

|  |  |  |
| --- | --- | --- |
| **分类** | **类型名称** | **说明** |
| **整数类型** | tinyInt | 很小的整数 |
| smallint | 小的整数 |
| mediumint | 中等大小的整数 |
| int(integer) | 普通大小的整数 |
| **小数类型** | float | 单精度浮点数 |
| double | 双精度浮点数 |
| decimal（m,d） | 压缩严格的定点数 decimal(10,2) 一共10位,小数2位 |
| **日期类型** | year | YYYY 1901~2155 |
| time | HH:MM:SS -838:59:59~838:59:59 |
| date | YYYY-MM-DD 1000-01-01~9999-12-3 |
| datetime | YYYY-MM-DD HH:MM:SS 1000-01-01 00:00:00~ 9999-12-31 23:59:59 |
| timestamp | YYYY-MM-DD HH:MM:SS 1970~01~01 00:00:01 UTC~2038-01-19 03:14:07UTC |
| **文本、二进制类型** | CHAR(M) | M为0~255之间的整数**定长 char(10) tom** |
| VARCHAR(M) | M为0~65535之间的整数**变长 varchar(10) tom** |
| TINYBLOB | 允许长度0~255字节 |
| BLOB | 允许长度0~65535字节 |
| MEDIUMBLOB | 允许长度0~167772150字节 |
| LONGBLOB | 允许长度0~4294967295字节 |
| TINYTEXT | 允许长度0~255字节 |
| TEXT | 允许长度0~65535字节 |
| MEDIUMTEXT | 允许长度0~167772150字节 |
| LONGTEXT | 允许长度0~4294967295字节 |
| VARBINARY(M) | 允许长度0~M个字节的变长字节字符串 |
| BINARY(M) | 允许长度0~M个字节的定长字节字符串 |

## DDL之数据库操作：database

### 创建数据库(增)

格式:

\* create database 数据库名;

\* create database 数据库名 character set 字符集;

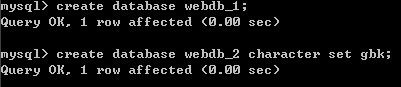
例如：

#创建数据库数据库中数据的编码采用的是安装数据库时指定的默认编码 utf8

CREATE DATABASE webdb\_1;

#创建数据库并指定数据库中数据的编码

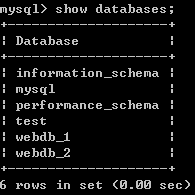
CREATE DATABASE webdb\_2 CHARACTER SET utf8;



### 查看数据库(查)

查看数据库MySQL服务器中的所有的数据库:

show databases;

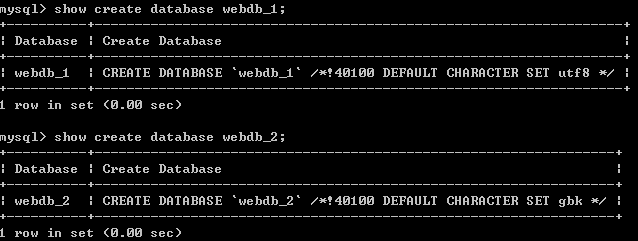


查看某个数据库的定义的信息:

show create database 数据库名;

例如：

Show create database webdb\_1;



### 删除数据库(删)

drop database 数据库名称;

例如：

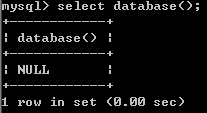
drop database webdb\_2;



### 修改正在使用的数据库(切换数据库)

* 查看正在使用的数据库:

select database();

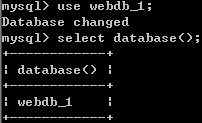


* 切换数据库:

use 数据库名;

例如：

use webdb\_1;



## DDL之表操作：table

### 创建表

* 格式：

create table 表名(

字段名 类型(长度) [约束],

字段名 类型(长度) [约束],

...

);

类型：

varchar(n) 字符串

int 整形

double 浮点

date 时间

timestamp 时间戳

约束(后期讲解,现在先知道)：

primary key 主键，被主键修饰字段中的数据，不能重复、不能为null。

例如：

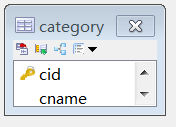
###创建分类表

CREATE TABLE category (

cid INT primary key, #分类ID

cname VARCHAR(100) #分类名称

);



### 查看表

* 查看数据库中的所有表：

格式：show tables;

* 查看表结构：

格式：desc 表名;

例如：desc sort;

### 删除表

* 格式：drop table 表名;

例如：drop table category;

### 修改表(只改名字)

* rename table 表名 to 新表名;

作用：修改表名

例如：

#5, 为分类表category改名成 category2

RENAME TABLE category TO category2;

## DDL之表结构操作：列

### 对表的结构进行操作(主要是操作表中的列)：

* alter table 表名 add 列名 类型(长度) [约束];

作用：修改表--添加列.

例如：

#1，为分类表添加一个新的字段为分类描述 varchar(20)

ALTER TABLE category ADD `desc` VARCHAR(20);

* alter table 表名 drop 列名;

作用：修改表--删除列.

例如：

#4, 删除分类表中snamename这列

ALTER TABLE category DROP description;

* alter table 表名 change 旧列名 新列名 类型(长度) 约束;

作用：修改表--修改列名.

例如：

#3, 为分类表的分类名称字段进行更换更换为 snamesname varchar(30)

ALTER TABLE category CHANGE `desc`description VARCHAR(30);

* alter table 表名 modify 列名 类型(长度) 约束;

作用：修改表--修改列的类型长度及约束.

例如：

#2, 为分类表的描述字段进行修改，类型varchar(50) 添加约束 not null

ALTER TABLE category MODIFY `desc` VARCHAR(50) NOT NULL;

* alter table 表名 character set 字符集;(一般不修改)

作用：修改表的字符集

例如：

#6, 为分类表 category 的编码表进行修改，修改成 gbk

ALTER TABLE category CHARACTER SET gbk;

* 查看表结构中的列(查看表时已经学过)：

格式：desc 表名;

例如：desc sort;

## DML数据操作语言

介绍两个约束

1. primary key 主键,值唯一,并且不能为空
2. auto\_increment 自动增长,数据可以由MySql自己维护

### 插入表记录：insert(增)

* 语法：

-- 向表中插入某些字段

insert into 表 (字段1,字段2,字段3..) values (值1,值2,值3..);

--向表中插入所有字段,字段的顺序为创建表时的顺序

insert into 表 values (值1,值2,值3..);

* 注意：
  + 值与字段必须对应，个数相同，类型相同
  + 值的数据大小必须在字段的长度范围内
  + 除了数值类型外，其它的字段类型的值必须使用引号引起。（建议单引号）
  + 如果要插入空值，可以不写字段，或者插入 null。
* 例如：

INSERT INTO category(cid,cname) VALUES('c001','电器');

INSERT INTO category(cid,cname) VALUES('c002','服饰');

INSERT INTO category(cid,cname) VALUES('c003','化妆品');

INSERT INTO category(cid,cname) VALUES('c004','书籍');

INSERT INTO category(cid) VALUES('c002');

INSERT INTO category(cname) VALUES('耗材');

### 更新表记录：update(改)

用来修改指定条件的数据，将满足条件的记录指定列修改为指定值

* 语法：

--更新所有记录的指定字段

update 表名 set 字段名=值,字段名=值,...;

--更新符号条件记录的指定字段

update 表名 set 字段名=值,字段名=值,... where 条件;

* 注意：
  + - 列名的类型与修改的值要一致.
    - 修改值得时候不能超过最大长度.
    - 除了数值类型外，其它的字段类型的值必须使用引号引起

### 删除记录：delete(删)

* 语法：

delete from 表名 [where 条件];

或者

truncate table 表名;

* 面试题：

删除表中所有记录使用delete from 表名; 还是用truncate table 表名;ctrl+z

删除方式：delete 一条一条删除，不清空auto\_increment记录数。

truncate 直接将表删除，重新建表，auto\_increment将置为零，从新开始。

## DOS操作数据乱码解决

我们在dos命令行操作中文时，会报错

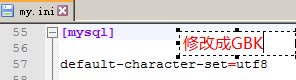
insert into category(cid,cname) values(‘c010’,’中文’);

ERROR 1366 (HY000): Incorrect string value: '\xB7\xFE\xD7\xB0' for column 'cname' at row 1

解决方案1：在cmd命令窗口中输入命令，此操作当前窗口有效，为临时方案。

set names gbk;

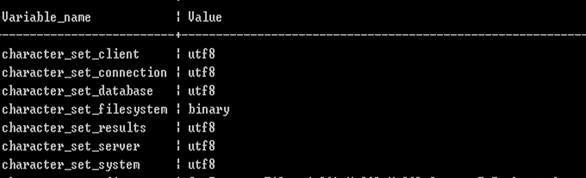
解决方案2：安装目录下修改my.ini文件，重启服务所有地方生效。[不建议]



错误原因:因为mysql的客户端设置编码是utf8,而系统的cmd窗口编码是gbk

1）查看MySQL内部设置的编码

show variables like 'character%'; 查看所有mysql的编码



2）修改client、connection、results的编码一致（GBK编码）

client connetion result 和客户端保持一致，都为GBK

* 将客户端编码修改为gbk.

#方式1：单独设置

set character\_set\_client=gbk;

set character\_set\_connection=gbk;

set character\_set\_results=gbk;

#方式2：快捷设置

set names gbk;

## DQL数据查询语言

### 准备工作

#创建商品表：

create table product(

pid int primary key,

pname varchar(20),

price double,

category\_id varchar(32)

);

INSERT INTO product(pid,pname,price,category\_id) VALUES(1,'联想',5000,'c001');

INSERT INTO product(pid,pname,price,category\_id) VALUES(2,'海尔',3000,'c001');

INSERT INTO product(pid,pname,price,category\_id) VALUES(3,'雷神',5000,'c001');

INSERT INTO product(pid,pname,price,category\_id) VALUES(4,'JACK JONES',800,'c002');

INSERT INTO product(pid,pname,price,category\_id) VALUES(5,'真维斯',200,'c002');

INSERT INTO product(pid,pname,price,category\_id) VALUES(6,'花花公子',440,'c002');

INSERT INTO product(pid,pname,price,category\_id) VALUES(7,'劲霸',2000,'c002');

INSERT INTO product(pid,pname,price,category\_id) VALUES(8,'香奈儿',800,'c003');

INSERT INTO product(pid,pname,price,category\_id) VALUES(9,'相宜本草',200,'c003');

INSERT INTO product(pid,pname,price,category\_id) VALUES(10,'面霸',5,'c003');

INSERT INTO product(pid,pname,price,category\_id) VALUES(11,'好想你枣',56,'c004');

INSERT INTO product(pid,pname,price,category\_id) VALUES(12,'香飘飘奶茶',1,'c005');

INSERT INTO product(pid,pname,price,category\_id) VALUES(13,'果9',1,NULL);

### 语法：

select [distinct] \* | 列名1,列名2 from 表 [where 条件]

### 简单查询

# 1.查询所有的商品. select \* from product;

# 2.查询商品名和商品价格. select pname,price from product;

# 3.去掉重复值. select distinct price from product;

# 4.查询结果是表达式（运算查询）：将所有商品的价格+10元进行显示.

select pname,price+10 from product;

# 5.别名查询.使用的关键字是as（as可以省略的）.

5.1表别名: select \* from product as p;

5.2列别名：select pname as pn from product;

### 条件查询

|  |  |  |
| --- | --- | --- |
| 比较运算符 | > < <= >= = <>  != | 大于、小于、大于(小于)等于、不等于 |
| BETWEEN ...AND... | 显示在某一区间的值(含头含尾)(也可以是日期) |
| IN(set) | 显示在in列表中的值，例：in(100,200) |
| LIKE ‘张%’ | 模糊查询，Like语句中，  %代表零个或多个任意字符，  \_代表一个字符，  例如：first\_name like ‘\_a%’; |
| IS NULL /IS NOT NULL | 判断为空/不为空 where name IS NULL  和name = ‘’ |
| 逻辑运算符 | and | 多个条件同时成立 |
| or | 多个条件任一成立 |
| not | 不成立，例：where not(salary>100); |

#查询商品名称为“花花公子”的商品所有信息：

SELECT \* FROM product WHERE pname = '花花公子'

#查询价格为800商品

SELECT \* FROM product WHERE price = 800

#查询价格不是800的所有商品

SELECT \* FROM product WHERE price != 800

SELECT \* FROM product WHERE price <> 800

SELECT \* FROM product WHERE NOT(price = 800)

#查询商品价格大于60元的所有商品信息

SELECT \* FROM product WHERE price > 60;

#查询商品价格在200到1000之间所有商品

SELECT \* FROM product WHERE price >= 200 AND price <=1000;

SELECT \* FROM product WHERE price BETWEEN 200 AND 1000;

#查询商品价格是200或800的所有商品

SELECT \* FROM product WHERE price = 200 OR price = 800;

SELECT \* FROM product WHERE price IN (200,800);

#查询含有'霸'字的所有商品

SELECT \* FROM product WHERE pname LIKE '%霸%';

#查询以'香'开头的所有商品

SELECT \* FROM product WHERE pname LIKE '香%';

#查询第二个字为'想'的所有商品

SELECT \* FROM product WHERE pname LIKE '\_想%';

#查询没有分类的商品

SELECT \* FROM product WHERE category\_id IS NULL

#查询有分类的商品

SELECT \* FROM product WHERE category\_id IS NOT NULL

#查询所有价格大于2000的电脑商品(catetory\_id是c001)

或者价格大于2000的服装商品(catetory\_id是c002)\*\*