进入mysql管理：

方法1：

(1) 打开cmd命令窗

(2) 进入mysql所在的路径，比如D:\wamp\bin\mysql\mysql5.6.12\bin

(3) 键入 mysql –uroot –pjack 即可进入mysql世界….

**或者使用套件提供的MYSQL命令窗口**

方法二：

（1）使用图形化软件，比如phpmyadmin、Navicate for mysql

（2）登陆后，即可进入mysql世界….

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**root用户管理篇**

1. **设置root 用户登陆密码**

mysql> set password for ‘root’@’localhost’ = password(‘xxxxx’);

mysql> flush privileges;

/\*\*\*或者\*\*\*/

用UPDATE直接编辑user表

　　mysql> use mysql;

　　mysql> UPDATE user SET Password = PASSWORD('newpass') WHERE user = 'root';

mysql> FLUSH PRIVILEGES; // 这样不必“重启mysql服务器”，让设置生效。

1. **删除root密码**

mysql> set password for ‘root’@’localhost’ = password(‘’);

mysql> flush privileges;

1. **修改root密码**

mysql> set password for ‘root’@’localhost’ = password(‘xxxxx’);

mysql> flush privileges;

1. **忘记root密码**

4.1 net stop mysql; //停掉mysql服务进程

如果使用wamp,则停止wampmysqld服务进程 net stop wampmysqld

如果使用phpstudy 则停止mysql.exe 和 mysqld.exe进程

* 1. 进入mysql.exe 目录，比如D:\wamp\bin\mysql\mysql5.6.12\bin

4.3 mysqld --skip-grant-tables; // 跳过权限检查

* 1. 不关闭当前窗口，新打开cmd窗口
  2. 进入mysql.exe 所在目录
  3. mysql –u root
  4. mysql update mysql.user set password= password(‘xxxx’) where user = ‘root’;

如果不想修改，select host,user,password from mysql.user; //即可查看到用户和密码

* 1. flush privileges; **更新权限！！！**
  2. net start mysql; （或者net start wampmysqld）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**数据库操作篇**

1、查看数据库：

　　mysql>SHOW DATABASES;

2、创建数据库：

//db\_name为数据库名

2.1 mysql>CREATE DATABASE db\_name;

2.2 mysql>CREATE DATABASE IF NOT EXISTS db\_name DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

2.3 先删除数据库 DROP DATABASE IF EXISTS db\_name;

然后，再创建 CREATE DATABASE db\_name DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

3、使用数据库：

　mysql>USE db\_name;

4、删除数据库：

　mysql>DROP DATABASE db\_name;

5、导出数据库

数据库的导出与导入须使用mysqldump 命令（而不是mysql命令），所有要通过cmd命令进入mysql的bin 目录，使用mysqldump来导出导入

导出数据库

mysqldump –uroot –proot 数据库名称 >[输出位置比如D:\,不写默认为bin 目录]新的数据库名称.sql （语句没有结束的分号）

导入数据库

/\*Mysqldump –uroot –proot source 数据库名称.sql \*/

1. 创建新的数据库 比如 temp
2. 使用新创建的数据库 use temp
3. source d:\xxxx.sql

注意：先创建好数据库，再导入进去。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**表操作篇**

**1.创建新表**

　　　　>CREATE TABLE table\_name(

　　　　>id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT,　　　　//id值，无符号、非空、递增——唯一性，可做主键。

　　　　>name VARCHAR(60) NOT NULL,

　　　　>score TINYINT UNSIGNED NOT NULL DEFAULT 0,　　　　//设置默认列值

　　　　>PRIMARY KEY(id) //创建主键

>key name(name) //创建索引

　　　　>)ENGINE=MyISAM charset=utf8;　　　//设置表的存储引擎，一般常用InnoDB和MyISAM；InnoDB可靠，支持事务；MyISAM高效 支持全文索引

　　　　//设置默认的编码，防止数据库中文乱码

　　　　如果有条件的创建数据表还可以使用 >CREATE TABLE IF NOT EXISTS tb\_name(........

**注意事项：（1）字段创建顺序：字段名+类型+（无符号）+（not null）+ （auto\_increment）/（default）（2）字符类型需加默认长度;**

创建表的其他方式1：

Create table if not exists tb\_ame (

….

)engine=MyISAM auto\_increment=5 charset=utf8;

或者先删除现有表

Drop table if exists tb\_name;

Create table tb\_ame (

….

)engine=MyISAM auto\_increment=5 charset=utf8;

创建表的其他方式2：

首先，创建.sql文件，文件中包含创建表的语句

其次，use 库名称

最后，source sql文件的完整物理路径（注意，后面不加封号）

1.1 新增加字段

alter table table1 add id int unsigned not Null auto\_increment primary key

1.2 修改某个表的字段类型及指定为空或非空

alter table 表名称 change 老字段名称 新字段名称 字段类型 [是否允许非空];

alter table 表名称 modify 字段名称 字段类型 [是否允许非空];

1.2 修改某个表的字段名称及指定为空或非空

alter table 表名称 change 字段原名称 字段新名称 字段类型 [是否允许非空

1.3 删除字段

ALTER TABLE mytable DROP 字段 名;

1.4 字段排序

Alter table tablename newcolumn int not null after id

Alter table tablename newcolumn int not null first

**2. 复制表**

　　　　>CREATE TABLE tb\_name2 SELECT \* FROM tb\_name;

　　　　或者部分复制：

　　　　>CREATE TABLE tb\_name2 SELECT id,name FROM tb\_name;

**3. 查看数据库中有哪些表**

> use 数据库名称;

> show tables;

**4. 查看数据表中的数据**

>use 数据库名称

>SELECT \* FROM 表名称

**5. 查看表结构**

　　>DESCRIBE tb\_name;

　　　　也可以使用：

　　　　>SHOW COLUMNS in tb\_name; 　　　　//from也可以

**6. 删除表**

　　　　>DROP [ TEMPORARY ] TABLE [ IF EXISTS ] tb\_name[ ,tb\_name2.......];

　　　　实例：

　　　　>DROP TABLE IF EXISTS tb\_name;

**7. 表重命名**

　　　　>RENAME TABLE name\_old TO name\_new;

　　　　还可以使用：

　　　　>ALTER TABLE name\_old RENAME name\_new;

**8. 修改表结构**

增加字段

>ALTER TABLE tb\_name ADD COLUMN address varchar(100) NOT NULL;

删除字段

>ALTER TABLE tb\_name DROP address;

修改字段

>ALTER TABLE tb\_name CHANGE phone tel SMALLINT(4) NOT NULL;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**数据操作篇**

**1. 插入数据**

* 1. 语法格式：

INSERT [INTO] table [(column1,column2,….)] VALUES (value1,value2,...);

* 1. 字符串类型的数据应该包含在单引号或双引号中，数字和日期类型则不需要引号。
  2. 可以选择某些字段插入值，字段要与值一一对应。Insert into tb (column1,column2) values (value1,value2);
  3. 也可以采用set方式插入。Insert into tb set column1=value1,column2=value2;
  4. 自动增量列须为主键字段，且不需要加入到语句中。

1.6 一次插入多条记录

>INSERT INTO tb\_name(name,tel)VALUES('张三','13309899989'),('张四','18088888898'),('张五','17099982323');

这里的插入多条数据直接在后边加上逗号，直接写入插入的数据即可；主键id是自增的列，可以不用写。

**2. 从数据表中获取数据**

2.1 使用select 语句从匹配的表中获取数据。基本语法格式：

SELECT [OPTIONS] items

FROM tables

[ WHERE conditions ]

[ GROUP BY grouptype ]

[ HAVING where\_definetion ]

[ ORDER BY order\_type ]

[ LIMIT limit\_criteria ]

注意条件顺序，LIMIT关键词必须在WHERE、ORDERBY 、ON、GROUP BY、HAVING关键词的后面。

2.2 一个简单的查询例子

2.2.1 查询特定字段的数据

SELECT column1,column2,… FROM tb

2.2.2 查询所有字段数据

SELECT \* FROM tb

2.3 获取满足特定条件的数据

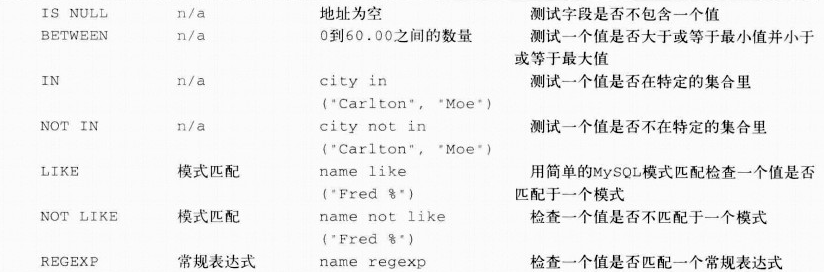
它的基本格式如下：

SELECT columns FROM tb WHERE column 比较运算符 值 [ AND 或者OR 其他字段]

比如，SELECT \* FROM orders WHERE customerid>5 AND pay<20000;

WHERE 字句常用的比较运算符有：





2.4 从多个表中获取数据

多个表之间一般有一定的关系，否则联表查询没有意义。

2.4.1 简单关联两个或更多的表查询，通常叫做内联或完全关联

SELECT tb1.column1,tb2.column5,..

FROM tb1,tb2

WHERE tb1.id = tb2.uid

[ AND ..]

[OR …]

2.4.2 左关联与右关联

左关联是在两个表之间指定的关联条件下匹配的数据行。如果右边的表中没有匹配行，则匹配的值为NULL。

左关联意味着首先要满足左边表的匹配，是主表。反之右关联则右边表为主表，要首先得到满足。

关联条件写在ON字句中。

SELECT \* FROM tb1.column2,tb1.column4,tb2.column1,…

FROM tb1 LEFT JOIN tb2

ON tb1.column1=tb2.column3

[ WHERE …]

2.4.3 使用表的别名

即使用as 新名称 来替代表名。

SELECT a.id , c.name,…

FROM tb1 as a RIGHT JOIN tb2 as c

[ ON …]

[ WHERE …]

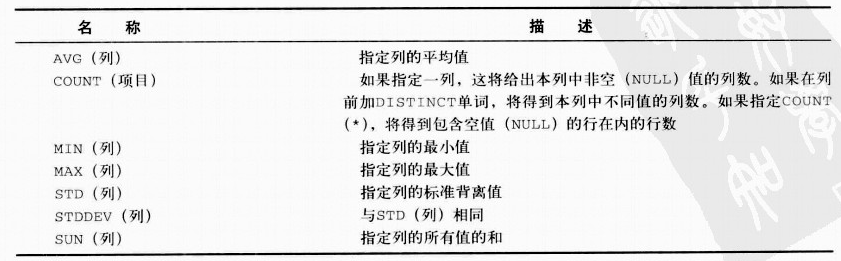
2.4.4 排序数据

使用ORDER BY 字段名 ASC 或者DESC，进行数据排序。默认为ASC即顺序。

SELECT \* FROM tb [ WHERE ] ORDER BY column DESC

2.4.5 分组与合计函数

对符合条件的数据，按照特定的分组集合进行合计的操作。



例子1： SELECT avg(column) from tb //计算改字段的平均值，该字段为数字类型。

例子2： SELECT column1,avg(column2) FROM tb GROUP BY column1

需要注意的是，如果使用了一个合计函数或GROUP BY 字句，出现在SELECT 字句中的必须是合计函数名称和GROUP BY 字句的字段名称。同样，如果希望在一个GROUP BY 字句中使用一列，该列名称必须在SELECT 字句中列出。

使用HAVING 字句测试一个合计结果。它要放在GROUP BY 字句的后面。类似于WHERE字句。

SELECT column1, avg(column2)

FROM tb

GROUP BY column1

HAVING avg(column2)>20;

2.4.6 限制返回的数据

使用LIMIT关键字来限定返回的行数，这个分页的限制条件。

SELECT \* FROM tb [ WHERE ] [ ORDER BY ] LIMIT 0,4

注意0表示从第一条数据开始，第二个参数表示从第一个参数位置开始返回的行数。

2.4.7 子查询

子查询是一个嵌套在另一个查询内部的查询。

SELECT column FROM tb WHERE column3 IN ( SELECT column FROM tb2 WHERE …)

**3. 修改数据**

3.1 UPDATE 语句的常用格式：

UPDATE table

SET colum1=express1,column2=express2,…

[ WHERE ]

如果没有限制语句，则将修改所有行数据。

3.2 字段可以使用表达式

UPDATE tb

SET column1 = column1+5

[ WHERE ]

[ ORDER BY ]

[ LIMIT ]

**4. 删除数据**

>DELETE FROM tb\_name WHERE id=3;

**5.清空数据**

>truncate tablename

还原表，记录从1开始