**目 录**

[MySQL数据库 3](#_Toc5969)

[MySQL数据库管理系统 3](#_Toc736)

[Web开发与数据库的关系 3](#_Toc12922)

[MySQL数据库的优点 3](#_Toc2489)

[MySQL数据库管理系统的结构 4](#_Toc11848)

[SQL语句四大类 4](#_Toc27752)

[MySQL操作筹备 4](#_Toc28299)

[配置环境变量 4](#_Toc13003)

[数据库字符集 4](#_Toc28853)

[链接MySQL数据库(Cmd命令) 5](#_Toc16)

[MySQL了解事项 5](#_Toc11840)

[对库(DATABASE)的操作 6](#_Toc22464)

[创建数据库 6](#_Toc11111)

[删除数据库 6](#_Toc6665)

[修改数据库 6](#_Toc26319)

[查看数据库 6](#_Toc30248)

[选择数据库 6](#_Toc26635)

[对表(TABLES)的操作 6](#_Toc28620)

[数据类型 6](#_Toc18626)

[创建数据表 8](#_Toc5088)

[删除数据表 9](#_Toc31856)

[修改数据表 9](#_Toc16694)

[备份数据表 9](#_Toc8952)

[对字段的操作 9](#_Toc11045)

[添加字段 9](#_Toc4398)

[删除字段 9](#_Toc3729)

[修改字段 9](#_Toc18018)

[查看字段 10](#_Toc5044)

[数据库中的运算符 10](#_Toc32610)

[算数运算符 10](#_Toc27545)

[比较运算符 10](#_Toc23200)

[操作符 11](#_Toc28217)

[按位运算符 11](#_Toc24145)

[数据的操作 11](#_Toc26251)

[添加数据 11](#_Toc30918)

[删除数据 11](#_Toc25610)

[修改数据 12](#_Toc5199)

[查看数据 12](#_Toc25984)

[排序查询 12](#_Toc25441)

[多表联合查询 13](#_Toc3861)

[数据库的导出和导入 13](#_Toc28384)

[导出 13](#_Toc28153)

[导入 13](#_Toc32693)

[导入导出注意事项 14](#_Toc7704)

[创建用户并且授权 14](#_Toc1564)

[PHP链接MySQL数据库 14](#_Toc2747)

[链接数据库 14](#_Toc29769)

[选择数据库 14](#_Toc2320)

[设置字符集 14](#_Toc28504)

[准备SQL语句 15](#_Toc17373)

[发送SQL语句 15](#_Toc4262)

[判断并处理结果 15](#_Toc27123)

[关闭数据库 15](#_Toc770)

# MySQL数据库

## MySQL数据库管理系统

### Web开发与数据库的关系

* 动态网站就是对数据库进行操作。
* 平时浏览网页时，会发送网页内容的变化，而页面的主题结构并不会发生改变。
* Web系统的开发基本是离不开数据库的。
* 只要是基于数据库的Web开发，都是属于动态网站的范畴。

### MySQL数据库的优点

* php+mysql可以说是黄金搭档，两者都是：开源、免费、配置简单、稳定性好、性能优良。
* mysql数据库在开源数据库中占据了很大的份额，也是开源数据库的代表。
* SQL语句结构简洁、功能强大、简单易学。关系型数据库都可以使用

### MySQL数据库管理系统的结构

* MySQL数据库管理系统是C/S(客户端/服务器)结构
* Web服务是B/S(浏览器/服务器)的开发

### SQL语句四大类

* 数据库定义语言(DDL)
* 数据库操作语言(DML)
* 数据库查询语言(DQL)
* 数据库控制语言(DCL)

## MySQL操作筹备

### 配置环境变量

* 复制路径(自己数据库的路径)：C:\wamp64\bin\mysql\mysql5.7.11\bin\
* 右击计算机->属性->高级系统设置->环境变量->在用户环境->新建->变量名：PATH->变量值:C:\wamp64\bin\mysql\mysql5.7.11\bin\. 注意：.（点）代表匹配该目录下的所有内容
* 打开运行cmd
* 输入mysqldump 看到如下效果代表环境配置成功

C:\Users\何大大>mysqldump

Usage: mysqldump [OPTIONS] database [tables]

OR mysqldump [OPTIONS] --databases [OPTIONS] DB1 [DB2 DB3...]

OR mysqldump [OPTIONS] --all-databases [OPTIONS]

For more options, use mysqldump --help

### 数据库字符集

* GB2312双字节编码 早期的标准 不推荐使用。
* GBK 双字节编码 中期的标准 支持的系统很多，并且在GB2312的基础上增加了很多偏僻的生字。
* UTF-8 1~4字节的编码 互联网广泛使用。亚洲通用字符集。国际标准化，支持任何语言,但是要注意:MySQL数据库也使用utf-8字符集，只是在mysql数据库中不能写成utf-8，要写成utf8。
* UTF-8和GBK的区别
* 存储数据的长度不一样，存储汉字在GB系统2个字节，UTF8是三个字节
* 在不同的操作系统(语言的不同)中看起不一样
* 推荐使用UFT-8标准
* UTF-16 2字节或者4字节编码，Java内部使用的。
* 解决乱码问题最好的方法就是字符集的统一，页面字符集utf8、声明字符集uft8、数据库字符集utf8

### 链接MySQL数据库(Cmd命令)

* (cmd)输入 mysql -h localhost(数据库服务器地址) -u root(用户名) -p
* 回车输入密码(没有密码直接回车)

C:\Users\何大大>mysql -h localhost -u root -p

Enter password: \*\*\*\*\*\*

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

* 清楚cmd中等待命令
* -> 表示当前命令没有命名执行符或者说等待命令执行符来确定
* \c 用来清楚->效果
* ‘>表示前面SQL命令中缺少单引号
* 补上’然后用\c来清楚该效果
* “>表示前面SQL命令中缺少双引号
* 补上”然后用\c来清楚该效果
* 退出mysql客户端的cmd命令：Exit;(大写E) \q \quit

### MySQL了解事项

* 在MySQL版本小于5.7之前允许游客模式的出现，如果用户名不对，会以游客的身份进入到MySQL数据库，这是查看MySQL数据库时，只会显示两个数据库。
* 一个是text 测试数据库
* 一个是information\_schema 数据字典
* 系统中自带的库(不能去删除和修改)

mysql> SHOW DATABASES;

+---------------------------+

| Database |

+---------------------------+

| information\_schema | 数据字典

| mysql | 默认的库

| performance\_schema | mysql5.5版本新增的一个库 性能字典

| sys | 系统数据库

+---------------------------+

5 rows in set (0.00 sec)

## 对库(DATABASE)的操作

对数据库进行增、删、改、查操作属于数据库的定义语言DDL。

同时需要注意：所有的SQL命令推荐使用大写

### 创建数据库

* CREATE DATABASE 库名;
* 如果库不存在则创建：CREATE DATABASE IF NOT EXISTS 库名;

### 删除数据库

* DROP DATABASE 库名;
* DROP DATABASE IF EXISTS 库名;

### 修改数据库

没有修改数据库名的命令，但要注意：是可以通过进入mysql文件夹中/data/数据库目录，在里面将其通过修改文件名的方式修改数据库名。同时需要知道，凡是可以修改的数据库名，只能修改myisam引擎的数据库。

* Myisam引擎 可实现快速存储
* Innodb 引擎 带有事物回滚机制

### 查看数据库

* SHOW DATABASES;

### 选择数据库

* USE 库名;

## 对表(TABLES)的操作

### 数据类型

* 整数型：



* TINYINT 最小的数值整型
* SMALLINT 较小的整型
* MEDIUMINT 小整型
* INT 标准的整型
* BIGINT 大整型
* 浮点型：



* FLOAT 单进度类型 (例如：3.14)
* DOUBLE(m,d) 双进度浮点型 (例如：3.1415) m表示:有多少个数(总长度) d表示:小数点后面保留多少位。
* Decimal(m,d) 字符串类型的浮点数(例如:’3.1415’) 金融数据一般使用该类型。他不是一个数，而是属于字符串。
* 字符串型：



* CHAR(M) 定义字符串类型。 M所指长度，在CHAR类型中0~255之间的数。
* VARCHAR(M) 变长字符串类型。 M的值是0~65536之间的数。
* CHAR()和VARCHAR()区别：
* 存储的长度不一样。
* CHAR()如果存储的值不够括号中定义的长度，会站位到该括号中定义的长度
* VARCHAR()如果存入的值不够括号中定义的长度，会以实际存储值的长度为准。
* 按照效率来分配：CHAR()占用空间较大，但是查询效率高。 VARCHAR()占用空间较小，查询效率低。
* TEXT类型 属于文本类型。一把在网页中用于存储大文本、文章等信息
* BLOB 类型是以二进制方式存储

### 创建数据表

在创建数据表的时候，需要注意的是在创建数据表的时候必须连同字段一起创建

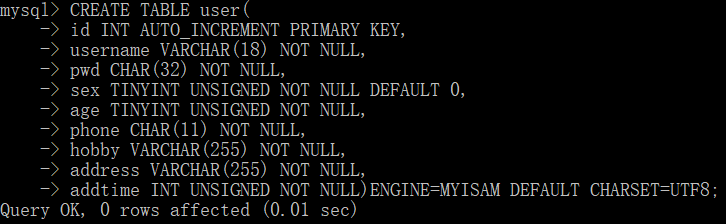
* CREATE TABLE [IF NOT EXISTS] 表名(

字段名 1 字段类型 [列的完整性约束],

字段名 2 字段类型 [列的完整性约束],

····)ENGINE=MYISAM DEFAULT CHARSET=UTF8;

* ENGINE 指定表引擎
* DEFAULT CHARSET 指定字符集



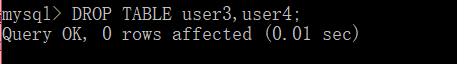
* 常用的列的完整性约束
* PRIMARY KEY 主键
* AUTO\_INCREMENT 自增
* DEFAULT ‘值’ 默认值
* NOT NULL 不为空
* NULL 为空
* UNSIGNED 无符号

### 删除数据表

* DROP TABLE 表名;

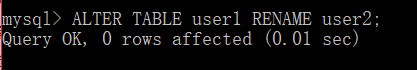


* DROP TABLE IF EXISTS 表名; (如果存在则删除)
* DROP TABLE 表名1,表名2··; (同时删除多张表)



### 修改数据表

* ALTER TABLE 原表名 RENAME 新表名;



### 备份数据表

* CREATE TABLE 新表名 SELECT \* FROM原表名;

注意：该备份能备份所有的表结构以及数据，但是唯一不能备份过来的就是主键和自增以及字符集。

## 对字段的操作

### 添加字段

* ALTER TABLE 表名 ADD 新字段名 数据类型 [列的完整性约束] [位置参数]
* 位置参数：
* AFTER 字段名 (表示在这个字段名之后添加)
* FIRST 字段名 (表示在第一个字段之后加入)

### 删除字段

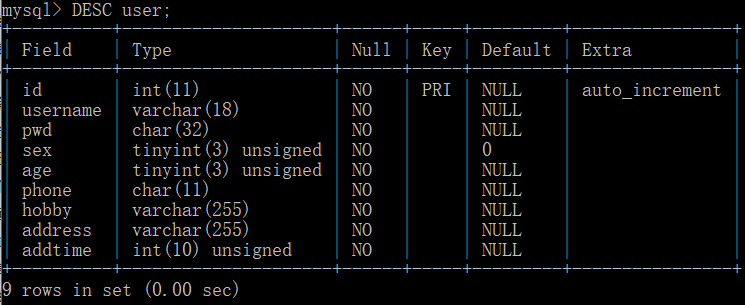
* ALTER TABLE 表名 DROP 字段名;
* ALTER TABLE 表名 DROP 字段名1, DROP 字段名2···;

### 修改字段

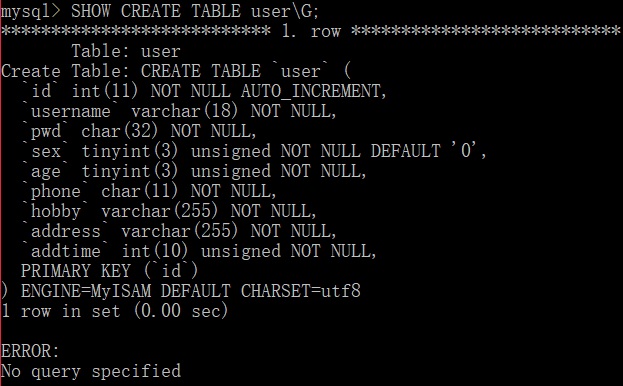
* ALTER TABLE 表名 MODIFY 字段名 新字段类型 [新列的完整性约束];
* ALTER TABLE 表名 CHANGE 原字段名 新字段名 新字段类型 [新列的完整性约束];
* MODIFY 和CHANGE 区别在于：
* MODIFY 只能更改原字段的字段类型以及列的完整性约束。
* CHANGE 可以更改字段名以及字段类型和列的完整性约束。

### 查看字段

* DESC 表名; (查看表中所有的字段和表结构)



* SHOW CREATE TABLE 表名\G;(查看建表语句)



## 数据库中的运算符

### 算数运算符

+ - \* /或者DIV %或者MOD

### 比较运算符

= < > != >= <=

### 操作符

* BETWEEN 值1 AND 值2 存在于指定范围内 (表示值1到值2 之间，包含值1 和值2)
* IN() 存在指定集合
* NOT IN() 不存在指定集合中
* IS NULL 为NULL
* IS NOT NULL 不为NULL
* REGEXP 正则匹配(SELECT ‘abcdef’ REGEXP ‘ab’ 匹配结果返回1 不匹配返回0)
* LIKE 通配符匹配

### 按位运算符

& 按位与 | 按位或 ^ 按位异或 ~按位异或 <<左移 >> 右移

## 数据的操作

### 添加数据

* SQL语句中相同的用逗号分隔，不同的用空格分隔
* INSERT INTO 表名(`字段名1`,`字段名2`...)VALUES(‘值1’,’值2’....)
* 字段名中出现反引号，作用是为了防止关键字报错。
* 如果该值是字符串类型，那么必须加引号。整数型可以省略引号
* INSERT INTO 表名 VALUES(‘值1’,’值2’....)
* 所有的字段必须依照顺序一一对应赋值，不写值的可以写成null。
* INSERT INTO 表名(`字段名1`,`字段名2`...)VALUES(‘值1’,’值2’....),(‘值1’,’值2’....)
* INSERT INTO 表名VALUES(‘值1’,’值2’....),(‘值1’,’值2’....)

### 删除数据

* DELETE FROM 表名 [删除条件];
* 只要是关于删除操作的都要在删除前备份该表数据
* 删除条件WHERE
* 删除id=3 和id=10的数据
* DELETE FROM user WHERE id=3 OR id=10；
* 删除id>3 和id<7的数据
* DELETE FROM user WHERE id>3 AND id<7;
* 删除 id<7 和id>9的数据
* DELETE FROM user WHERE id<7 OR id>9;
* 删除指定的集合(in())
* DELETE FROM user WHERE id IN(1,3,5,7,8,10,15);
* 删除指定的范围(between ... and)
* DELETE FROM user WHERE id BETWEEN 4 and 9;
* 删除表中id字段为偶数的数据
* DELETE FROM user1 WHERE id MOD 2=0;

### 修改数据

* UPDATE 表名 SET 字段名=值,字段名=值......[修改条件];
* Is null 字段为null(修改sex=2条件age字段中为null)
* UPDATE user SET sex=2 WHERE age IS NULL;
* Is not null字段不为null(修改user表中 sex字段等于3 条件是age字段不为空的)
* UPDATE user SET sex=3 WHERE age IS NOT NULL;

### 查看数据

* SELECT \* FROM 表名 [查询条件];
* SELECT 使用逗号分隔写入所有字段名称 FRMO 表名[查询条件];
* LIKE 模糊匹配 格式：字段LIKE ‘%内容%’
* 查询user字段中有a出现所有数据
* SELECT \* FROM user WHERE user LIKE ‘%a%’;
* 查询user字段中以l开头后面是任意内容
* SELECT \* FROM user WHERE user LIKE ‘l%’;
* 查询user字段中以i结尾前面是任意内容
* SELECT \* FROM user WHERE user LIKE ‘%i’；
* 查询user字段中第二个字母是i的所有内容
* SELECT \* FROM user WHERE user LIKE ‘\_i%’;
* 查询user字段中倒数第二个字母是n的所有内容
* SELECT \* FROM user WHERE user LIKE ‘%n\_’;
* %表示通配符，如果在字母前，表示该字母前有任意内容，如果在字母后，表示该字母后有任意内容。
* \_表示匹配一位字符，如果在字母前出现一个\_,表示该字母前必须有一位。
* 查询数据中有多少条数据
* SELECT COUNT(\*) FROM user;
* SELECT COUNT(id) FROM user;

### 排序查询

* 一个是正序排序(正序) ORDER BY 字段 ASC (按照字段正序排序)
* 一个是倒序排序(倒序) ORDER BY 字段 DESC (按照字段倒序排序)
* 按照age字段正序排序
* SELECT \* FROM user ORDER BY age ASC;
* 按照id 字段倒序培训
* SELECT \* FROM user ORDER BY id DESC;
* 查询sex=0的所有数据 按照id倒序排序
* SELECT \* FROM user WHERE sex=0 ORDER BY id DESC;
* 按照sex 正序排序，当sex相同时按照age倒序排序
* SELECT \* FROM user ORDER BY sex ASC,age DESC；
* 限制查询的数量 LIMIT
* SELECT 字段FROM 表名 LIMIT 查询数量;
* SELECT 字段 FROM 表名 LIMIT 跳过几条,查询几条；
* 查询sex=1 的数据取出3条
* SELECT \* FROM user WHERE sex=1 LIMIT 0,3;
* SELECT \* FROM user WHERE sex=1 LIMIT 3;
* 分组查询：GROUP BY 字段名
* 查询每个部门(bumen)的总工资(gz 工资表)？
* SELECT bumen,sum(gz) FROM gz GROUP BY bumen;
* 查询每个部门有多少人？
* SELECT bumen,count(id) FROM gz GROUP BY bumen;
* 查询部门编号大于等于2的所有部门的总工资
* SELECT bumen,sum(gz) FROM gz WHERE bumen>=2 GROUP BY bumen;
* 查询部门编号大于等于2的所有部门的总工资大于70000的
* SELECT bumen,SUM(gz) FROM gz WHERE bumen>=2 GROUP BY bumen HAVING SUM(gz)>70000;
* HAVING ：效率极低
* 一般使用在分组里面做二次过滤，使用方法和where方法一样。
* 查询条件使用的顺序
* SELECT \* FROM 表名 [WHERE条件][GROUP BY条件][HAVING 条件][ORDER BY 条件][LIMIT 条件];

### 多表联合查询

* 查询工资表(gz)中所有的内容条件是将gz表中的bumen编号替换成bm表中的文字name字段
* SELECT \* FROM gz,bm WHERE gz.bumen=bm.id;
* SELECT gz.id,gz.name,gz.gz,bm.name FROM gz,bm WHERE gz.bumen=bm.id;
* 多表联合查询一定要有字段是关联的才能查询

## 数据库的导出和导入

### 导出

* MYSQLDUMP [-h localhost] -u root -p 库名[表名] >导出的文件
* C:\>MYSQLDUMP -h localhost -u root -p lianxi > sql.sql

### 导入

* MYSQL [-h localhost] -u root -p 要导入的库名 < sql文件路径
* C:\>MYSQL -h localhost -u root -p lianxi < sql.sql

### 导入导出注意事项

* 在cmd下导出和导入时需要以管理员的身份进入cmd然后进行操作，否则会显示拒绝访问。以管理员进入cmd在Systme 32 下找到cmd.exe 右键。
* 导入和导出的命令后面不要给命令执行符，否知会报错，对于导入和导出箭头(< >) 后面给的文件名。

### 创建用户并且授权

* GRANT 权限 ON 数据库.[数据表] TO 用户名@登录主机地址 IDENTIFIED BY ‘密码’
* 创建一个wangyun用户 让他对jkxy中gz表有查询权限
* GRANT SELECT on jkxy.gz TO wangyun@localhost IDENTIFIED BY ‘123’;
* 创建用户 要求对所有的库以及所有的表有增、删、改、查权限
* GRANT SELECT,INSERT,UPDATE,DELETE ON \*.\* TO 用户名@登录主机 IDENTIFIED BY ‘密码’
* \*.\* 表示 所有库中的所有表

## PHP链接MySQL数据库

### 链接数据库

* Mysqli\_connect();

@param1: 链接数据库的地址

@param2: 数据库的用户名

@param3: 数据库的密码

@param4: 可选参数，要操作的数据库

返回值: 成功返回对象，失败返回false

### 选择数据库

* Mysqli\_select\_db();

@param1: 由mysqli\_connect()链接数据库成功返回的对象标识。

@param2: 要操作或者更改的数据库

返回值: 选择数据库成功返回true，失败返回false

### 设置字符集

* Mysqli\_set\_charset();

@param1: 由mysqli\_connect()链接数据库成功返回的对象标识

@param2: 要设置的字符集，数据库中UTF-8设置为UTF8

返回值: 成功返回true 失败返回false

### 准备SQL语句

* 准备增、删、改、查的各项SQL语句

### 发送SQL语句

* Mysqli\_query();

@param1: 有mysqli\_connect()链接数据库成功返回的对象标识

@param2: 要发送的SQL语句

返回值: 做添加、删除、修改时，成功返回true，失败返回false。 做查询时，成功返回一个对象，失败返回false

### 判断并处理结果

* Mysqli\_affected\_rows() 返回上一步mysql操作的影响行

@param1: 由mysqli\_connect()链接数据库成功返回的对象标识

返回值: 如果有影响行则返回影响行，如果没有返回0，操作返回-1

注意：在增、删、改时会有影响行产生

* mysqli\_num\_rows() 返回查询数据中的行数

@param1: 有mysqli\_query() 执行查询语句成功返回的对象。

返回值: 返回该对象中查询到数据的行数

注意: 在查询操作时返回的是查询到数据的行数

* mysqli\_fetch\_assoc() 从查询成功结果对象中获取一行作为关联数组返回

@param1: 由mysqli\_query() 执行查询语句成功返回的对象

返回值: 从该对象中返回一行数据作为关联数组

* mysqli\_insert\_id() 返回上一步添加操作所产生的唯一ID

@param1: 由mysqli\_connect()链接数据库成功返回的对象标识。

返回值: 返回当前插入数据的id

### 关闭数据库

* mysqli\_close() 关闭数据库

@Param1: 有mysqli\_connect()链接数据库成功返回的对象标识。

返回值: 成功返回true，失败返回false