**安装mysql 数据库**

**·必须安装的软件包 server client**

**·首次启动服务 会执行数据初始化**

Systemctl start mysql //启动服务

Systemctl enable mysql //开机运行

Systemctl status mysql //查看状态

Netstat -utnlp | grep :3306 //查看服务信息

Ps -C mysql //查看进程

**·软件安装后自动创建相关目录与文件**

/etc/my.cnf //主配置文件

/var/lib/mysql // 数据库目录

/var/log/mysqld.log //错误日志文件

**修改root密码**

语法： alter user root@”localhost” identified by “新密码”

**密码登录**

语法： mysql -hlocalhost -uroot -p密码 [数据库名] //使用新密码登录

**在mysql 中查看变量**

语法： show variables like “%password%”; //查看变量

**修改变量值**

语法： set global 变量=值；

**永久配置**

把写 变量=值 存在 /etc/my.cnf //永久配置文件里

**·数据库的管理命令（类似与目录）增删改查**

- 在数据库中库名 就是唯一标识 可以创建多个库 通过库名区分

- show databases; //显示已有的库

- select user(); //显示当前连接用户

- use 库名; // 切换位置

- select database(); //显示当前所在的库

- create database 库名； //创建新库

- show tables; //显示已有的表

- drop database 库名； //删库

**·表的管理命令（类似于目录里的文件）增删改查**

- create table 库名.表名（

字段名1 类型（宽度）， 字段名2 类型（宽度）） DEFAULT CHARSET=YUTF8；指定词表支持中文

- desc 库名.表明； //查看表结构

- drop table 库名.表名； //删除表

**·记录管理命令（类似于文件里的行）增删改查**

- select \* from 库名.表名； //查看表记录

- insert into 库名.表名 vvalues (值列表)； //插入表记录

- update 库名.表明 set 字段=值； //修改表记录

- delete from 表名； · ··· ······ //删除表记录

**【·数据库类型】**

【字符类型】

【定长cha】

最大字符个数255 不够指定字符个数时在右边空格补全 字符超出时 无法写入数据

语法： create table db1.t1( name char(5), homedir char(50));

【变长varchar】

按数据实际大小分配空间 字符超出时无法写入数据

语法： create table db1.t2(name char(5), email varchar(30));

【大文本类型】 text/blob

【数值类型】

【整数型】

Smallint 小整数

Int 大整数

Bigint q 极大整型

Unsigned 使用整数范围

语法： create table db1.t3(pay int unsigned, age smallint);

【浮点型】

Float 单精度

Double 双精度

语法：一：字段名 类型

二：字段名（总宽度，小数位）

【日期时间类型】

类型

Datatime

格式： yyyymmddhhmmss (在不输入值时为空)

Timestamp

格式：yyyymmddhhmmss（默认为系统时间）

Date 日期

格式：yyyymmdd

Year 年

格式：yyyy

Time 时间

格式：HH:MM:SS

【时间函数】

Now() 获取当前系统时间和日期

Time( now() ) 获取时间

Year( now() ) 获取年

Day( now() ) 获取日

Date( now() ) 获取日期

【枚举类型】

Enum单选

格式：字段名 enum(“值”,”值”,”值”)

仅能选择一个值 字段必须在列表里选择

【Set多选】

格式：字段名 set(“值”,”值”,”值”,”值”,”值”,”值”)

选择一个或多个值 字段必须在列表里选择

**【约束条件】**

**Desc 库名.表名 //查看约束条件**

**设置约束条件**

Null 允许为空（默认设置）

Not null 不允许为null (空)

Key 键值类型

Default 设置默认值 缺省为null

Extra 额外设置

**【表结构（增删改查）】**

【约束条件】

查看约束条件

语法：desc 库名.表名；

字段名 | 类型 | 空 | 键值 | 默认值 | 额外设置

设置约束条件

Null 允许为空值 not null 不允许为空值 key 键值类型 default 设置默认值 默认为空 extra 额 外设置

【新建表结构】

语法： create table db1.t7(name char(10) not null , age tinyint unsigned default 19);

修改表结构

语法：alter table 库名.表名 执行动作；

执行动作有 add 添加字段 名 类型（宽度） 约束条件 添加位置；

【after 字段名 添加到字段之后】

【first 添加到第一个】

Modify 修改字段类型

语法： modify 字段名 类型（宽度） 约束条件 位置；

修改的字段类型不能拿与已经存储的数据冲突

Change 修改字段名

Drop 删稠字段

Rename 修改表名

**【Mysql 键值】**

**键值类型**

1. index 普通索引
2. Unique 唯一索引
3. Fulltext 全文索引
4. Primary key 主键
5. Foreign key 外键

**【索引介绍】**

1. 类似于书的目录
2. 对表中字段值进行排序
3. 索引类型包括：Btree B+tree hash

**优点：**通过创建唯一性索引 可以保证数据库表中每一行数据的唯一性

可以加快数据的查询速度

**缺点：** 当对表中的数据惊醒增 删 改的时候索引也要动态的调整 降低了数据的维护速度

索引需要占用物理空间

**【普通索引index 使用规则】**

一个表中可以有多个index字段

一个字段的值允许重复 且可以符null(空)值

通常把做为查询条件的字段设置为index字段

Index字段标志是mul

创建索引

在已有的表里创建索引

Create index 索引名 on 表名（字段名）；

删除索引

Drop index 索引名 on 表名；

查看索引信息

Show index from 表名 \G;

【Primary key 主键】

使用规则

字段值不允许重复 且不允许符null 值

一个表中只能有哦一个primary key字段

多个字段都作为主键 称为复合主键 必须一起创建

主键字段的标志是PRL

主键通常与auto\_increment连用

通常把表中唯一标识记录的字设置为主键【记录编号字段】

建表时创建主键（primary key(字段名) ）

语法：create table 库名.表名（name char(3) primary key,id int）；

查看：Desc 库名.表名

在已有的表中添加主键

语法：alter table 表名 add primary key(字段名);

删除主键

语法：alter table 表名 drop primary key;

床键复合主键（primary key(字段名1，字段名2) ）

建表时创建复合主键

Create table 库名.表名（name char(2), class char(7), status enum(“yes”,”no”), primary key(name,class) ）；

与auto\_increament连用（主键每次默认自增1）

Create table 库名.表名（id int primary key auto\_increment, name char(10)）；

【foreign key 外键】

外键功能：插入记录时 字段值在另一个表字段值范围内选择

使用规则：表中储存引擎必须是innodb 两个表中字段类型要一致 被参照字段必须要是索引类型的 一种（primary key）

建表时创建外键

Create table 库名.表名（字段名列表 foreign key(字段名) references 表名（字段名) //指定外键

On update cascade //同步更新

On delete cascade //同步删除

）engine=innodb； //指定存储引擎

删除外键

Alter table 表名 drop foreign key 外键名；

Show create table gz\G //查看外键名

**【数据库导入导出】**

**【管理表记录】**

**【匹配条件】**

**【mysql 管理工具】**

数据导入导出

查看搜索路径

·安装服务时已自动创建

Show variadb like “%file%”; //搜索查看数据库变量 %匹配任意字符

Ls -ld /var/lib/mysql-files/ //查看目录

修改搜索路径

·创建目录并修改所有者 mkdir /myload chown mysql /myload

·修改主配置文件 vim /etc/my.cnf

Secure\_file\_priv=”/myload”

·重启服务 systemctl restart mysqld

数据导入

命令格式

·字段分割符要与文件一致

·表字段类型和字段个数要与文件匹配

·导入数据时指定摁键的绝对路径

导入注意事项

字段分割符要与文件一致

表字段类型和字段个数要与文件匹配

导入数据时指定文件的绝对路径

数据导入

数据导入步骤：

默认只要root用户有数据导入权限

建表

导入数据

语法：load data infile “目录名/文件名”

into table 库名.表名

Fields terminated by “分割符” //指定列的分隔符

Lines terninated by “\n”; //指定行的分割符 默认行的结尾 \n

数据导出

格式1：select命令 into outfile “目录名/文件名”;

格式2：select into outfile “目录名/文件名”

Fields terminated by “分隔符”; //指定列的分割符

格式3：select into outfile “目录名/文件名”

Fieldes terminated by “分割符”

Linex terminated by “\n”;

导出数据注意事项

导出数据行数由sql 查询决定

导出的是表记录 不包括字段名

自动创建储存数据的文件

存储数据文件 具有唯一性

**【管理表记录】**

**增加表记录**

**语法格式**

格式1：添加一条记录给所有字段赋值

Inset into 表名 values (字段值列表);

格式2：添加多条记录 给所有字段赋值

Inser into 表名

Values

(字段值列表),(字段列表),(字段值列表)

格式3：添加1条记录 给指定字段赋值

Inser into 表名（字段名列表） values(字段值列表);

格式4：添加多条记录 给指定字段赋值

Insert into 表名（字段列表）

values

（字段值列表）， //第一条记录

（字段值列表） //第二条记录

。。。；

**注意事项**

字段值要与字段类型相匹配

字段类型的字段 要用“”号括起来

依次给所有字段赋值时 字段名可以省略

只给部分字段赋值时 必须明确写出对应的字段名称

没有赋值的字段使用默认值或自增长赋值

**查询表记录**

语法格式

格式1 查看所有记录

Select 字段1,。。,字段N from 库名.表名；

格式2 条件查询

Select 字段1，。。，字段N from 库名.表名

Where 条件表达式；

**注意事项**

\* 表示所有字段

查看当前库表记录时库名可以省略

字段列表决定显示列个数

条件决定显示行的个数

**更新表记录**

语法格式

格式1 批量更新

Update 库名.表名 set 字段值=值， 字段名=值， 字段名=值，。。。；

格式2 条件匹配更新

Update 库名.表名 set 字段名=值，字段名=值。。。 where 条件表达式；

注意事项

字段值要与字段类型相匹配

对于字符类型的字段 值要用双引号括起来

若不适用where 限定条件 会更新所有记录字段值

限定条件时 只更新匹配条件的记录字段值

**删除表记录**

格式1 条件匹配删除

Delete from 库名.表名 where 条件表达式；

格式2 删除所有记录‘

Delete from 库名.表名；

注意事项

不加条件删除表中的所有行

**【匹配条件】**

**基本匹配条件**

**数值比较**

字段必须是数值类型

= 相等

> 大于

>= 大于/等于

< 小于

<= 小于/等于

！= 不相等

**字符比较/匹配空/非空**

字段必须是字符类型

= 相等

！= 不相等

Is null 空

Is not null 非空

**逻辑匹配**

多个条件判断条件时使用

Or 逻辑或

And 逻辑与

！或 not 逻辑非

**范围匹配/去重显示**

匹配范围内的任意一个值即可

In(值列表) 在。。里。

Not in(值列表) 不在。。里。

Between 数字 ans的数字 在。。之间。。

**高级匹配条件**

**模糊查询**

语法：

Where 字段名 like ‘通配符’

- ： 表示一个字符

%： 表示0~n个字符

**正则表达式**

语法：

Where 字段名 regexp ‘正则表达式’

正则元字符：^ $ . [] \* |

**四则运算**

字段必须时数值类型

+ 加法

- 减法

\* 乘法

/ 除法

% 取余数（求模）

（） 提高优先级

**操作查询结果**

语法： select cont(\*) from userdb.user;

聚集函数

Avg(字段名) //统计字段平均值

Sum(字段名) //统计字段之和

Min(字段名) //统计字段最小值

Max(字段名) //统计字段最大值

Count(字段名) //统计字段值个数

**查询结果排序**

Order by 字段名（通常时数值类型） asc //升序

Desc //降序

Select name , uid from userdb.user order by uid desc

**查询结果分组**

用法sql 查询 group by 字段名（字符类型）

Distinct 字段名 去重显示

**查询结果过滤**

**Having 用法（对查询结果再次处理）**

**Sql查询 having**

**Select name from userdb.user having name=”bob”**

**限制查询结果显示行数**

**Sql查询 limit 数字； //显示查询结果前多少条记录**

**Sql查询 limit 数字1 数字2 //显示指定范围内的查询**

**数字1 起始行 （0表示第一行）**

**数字2 总行数**

**Select name,uid,gid from user limit 3;**

**Select name,uid,gid from user limit 3 , 3;**