**常见的sql操作指令**

**DDL数据定义语言（create alter drop）**

**DML数据操作语言（insert update delete）**

**DCL数据控制语言（grant revoke）**

**DTL数据事物语言（commit rollback savepoint）**

**mysql初始配置**

修改root口令，默认的数据库管理帐号root，仅允许localhost访问，首次登陆密码随机生成，存在错误文件里（/var/log/mysqld.log）

mysql -uroot -p随机密码

mysql>set globel vaildate\_password=policy=0;//只验证长度（0只验证长度，1验证长度、数字、大/小写、特殊字符，2验证长度、数字、大/小写、特殊字符和字典文件）

mysql>set global vaildate\_password\_length=6;//修改密码长度，默认值是8个字符

mysql>alter user root() identified by “123456”;//修改登录密码

mysql -uroot -p123456//使用新密码登录

**mysql数据类型**

常见的信息种类：数值型、字符型、枚举型、日期时间型

数值型（整型、浮点型）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 大小 | 范围（有符号） | 范围（无符号） | 用途 |
| TINYINT | 1字节 | -128~127 | 0~255 | 微小整数 |
| SMALLINT | 2字节 | -32768~32767 | 0~65535 | 小整数 |
| MEDIUMINT | 3字节 | -231~231-1 | 0~224-1 | 中整数 |
| INT | 4字节 | -231~231-1 | 0~223-1 | 大整数 |
| BIGINT | 8字节 | -263~263-1 | 0~264-1 | 极大整数 |
| FLOAT | 4字节 |  |  | 单精度浮点数 |
| DOUBLE | 8个字节 |  |  | 双精度浮点数 |
| DECIMAL | 对DEVIMAL(M,D)，其中M为有效位数、D为小数位数、M赢大于D，占用M+2字节 | | | |

整数型：

定义格式：类型 int

使用unsigned修饰时，对应的字段只保存正数，也就是无符号数

如果存小数，会对小数位四舍五入

数值不够指定宽度时，在左边填空格补位

宽度仅是显示宽度，存数值的大小由类型决定

使用关键字zerofill时，填0代替空格补位

浮点型

定义格式：float(总宽度,小数位) float(m,n) double(m,n)

当字段值与类型不匹配时，字段作为0处理

数值超出范围时，仅保存最大/最小值

字符型

默认宽度为1，varchar必须指定宽度，否则为语法错误，char可以不指定宽度

char定长：格式：char(字符数) char(6)

最大长度255字符

不够指定字符数时在右边用空格补齐

字符判断超出时，无法写入数据

varchar变长：格式：varvhar(字符数) varchar(6)

0-65532个字符

按数据实际分配存储空间

字符数段超出时，无法写入数据

大文本类型：text/blob（二进制文件，音频文件，不需要指定宽度）

格式：text/blob

字符数大于65535存储时使用

枚举类型，最多大于64个

从给定集合中选择单个值，enum

定义格式：enum(值1,值2,值N) enum(“girl”,”boy”,”in”)

从给定集合中选择一个或多个值,set

定义格式：set(值1,值2，值N) set(“aa”,”bb”,”cc”)

日期时间类型

日期时间，datetime

占用8个字节

范围：1000-01-01 00：00：00.000000~9999-12-31 23.59.59.999999

赋值：20180202111100，表示2018年2月2日11点11分11秒

日期时间，timestamp

占用4个字节

范围：1970-01-01 00:00:00.000000-2038-01-19 03:14:07:999999

赋值：20180202111100，表示2018年2月2日11点11分11秒

日期，date

占用4个字节

范围：0001-01-01~9999-12-31

赋值：20181232，表示2018年12月31日

年份，year

占用1个字节

范围：1901~2155

赋值：2018，表示2018年

时间，time

占用3个字节

格式：HH:MM:SS

赋值：111111，表示11点11分11秒

关于日期时间字段

当未给timestamp字段赋值时，自动以当前系统时间赋值，而datetime字段默认值为空

year年份的处理

默认用4位数字表示

当只用2位数字表示时，01~69视为2000~2069，而70~99视为1970~1999

时间函数

|  |  |
| --- | --- |
| 类型 | 用途 |
| now() | 获取系统当前日期和时间 select now()； |
| year() | 获取指定日期时间的年 select year(20190101)； |
| sleep(N) | 休眠N秒 select sleep(2)； |
| curdate() | 获取当前的系统日期select curdate(); |
| curtime() | 获取当前的系统时刻select curime(); |
| month() | 获取指定时间中的月份select month(now()); |
| date() | 获取指定时间中的日期select date(20190101); |
| time() | 获取指定时间中的时刻select time(now()); |

无需库、表，可直接调用，使用select指令输出函数结果

mysql>select now(),sysdate(),curldate();

**数据库基本管理**

数据存储流程：连接数据库服务器->建库->建表->插入记录->断开连接

连接mysql服务器：mysql [-h 服务器 -u用户名 -p密码 数据库]

quit或exit退出

注意事项：操作指令不区分大小写（密码、变量值除外）

每条sql指令以分号结束或分割

不支持tab补齐

\c可废弃当前编写错的操作指令

**库管理命令**

show databases;//显示已有的库

use 库名;//切换库

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;//删除库

数据库的命名规则

可以使用数字/字母/下划线，但不能纯数字

区分大小写，具有唯一性

不可使用指令关键字、特殊字符

**表管理命令**

新建指定名称的表

create table 库名.表名(

字段1 字段类型（宽度） 约束条件,

字段2 字段类型（宽度） 约束条件,

......

字段N 字段类型（宽度） 约束条件

)default charset=utf8;(设置格式为utf8)

desc 表名;//查看表结构（表的字段，有字段名|字段类型|是否为空|是否为主键|默认值|描述信息的相关信息）

select \* from 表名;//查看表记录（\*字段）

drop table 表名;//删除表

show create table 表名;//查看创建表的命令

**记录管理命令**

select \* from 库名.表名;//查看全部表记录

insert into 表名 values(值列表1),(值列表2);//插入表记录

update 表名 set 字段=值;//修改全部表记录

delete from 表名;//删除全部表记录

**约束条件**

NULL允许为空，默认为空，not null不允许为空

key索引类型，默认没有，需要单独指定

default，默认为null

extra额外设置，是否配置自动增长

**修改表结构**

基本用法：alter table 表名 执行动作;

常用动作：add添加字段

modify修改字段

change修改字段名

drop删除字段

rename重命名表

添加新字段

基本用法：alter table 表名 add 字段名 类型（宽度） 约束条件 after/first;

修改字段类型（可以修改类型和约束条件）

基本用法：alter table 表名 modify 字段名 类型（宽度） 约束条件 after/first;

注意：新类型如果与已存数据冲突，不允许修改；修改过程中，修改的写出来，不变的原样抄下来，否则还原为系统默认

修改字段名

基本用法：alter table 表名 change 字段名 新字段名 类型（宽度） 约束条件：

当跟新类型和约束条件时，也可修改字段类型

删除字段

基本用法：alter table 表名 drop 字段名;

表中有多条记录时，此时列的此字段的值都会被删除

修改表名

基本用法：alter table 表名 rename 新表名;

表对应的文件名也会改变

**mysql键值，约束如何给字段赋值**

mysql索引（类似于书的目录），常用类型包括：btree二叉树算法、b+tree、hash

优缺点：

优点：通过创建唯一性索引，可以保证数据库表中的每一行数据的唯一性

可以加快数据的查询速度

缺点：当对标中的数据进行增加、删除和修改的时候，索引也要动态维护，降低了数据的维护速度

索引需要占用物理空间

减慢写入数据的速度

键值类型

index：普通索引

primary key：主键

foreign key：外键

index普通索引

使用说明

一个表中可以有多个Index字段

字段的值允许有重复，可以赋NULL值

经常把做查询的字段设置为Index字段

index字段的key标志是MUL

建表的时候指定索引字段

create table aa(

name char(4),

class char(15),

sex enum(“boy”,”girl”) not null default “boy”,

index(name),index(class)

);

在已有的表中设置index字段

create index 索引名 on 表名(字段名)；

删除指定表的索引字段

drop index 索引名 on 表名;

查看表的索引信息

show index from 表名;

primary key主键

使用说明

一个表中只能有一个primary key主键

对应的字段值不允许有重复，切不允许为NULL值

如果有多个字段都作为主键，称为符合主键，必须一起创建

主键字段的key标志是PRI

通常与auto\_increment连用

经常把表中能够唯一标识记录的字段设置为主键字段

建表时指定主键字段

create table aa(

name char(4) primary key,

class char(15),

);

在已有表中设置primary key(字段名)

alter table 表名 add primary key(字段名);

移除表中的primary key字段

alter table 表名 drop primary key;

移除主键前，如果有自增属性，必须先去掉

复合主键（一个以上的主键，这两个字段的值不同时，就不算重复）

在新表中添加复合主键

create table aa(

name char(4),

class char(15),

sex enum(“boy”,”girl”) not null default “boy”,

primary key(name,class)

);

删除复合主键

alter table aa drop primary key;

删除已有表中添加复合主键

alter table aa add primary key(cip,port);

自增长

auto\_increment字段值自增长，每次自增加一（必须是主键且是数值类型）

create table aa(

name char(4) primary key auto\_increment,

class char(15),

);

foreign key外键

概念：让当前表字段的值在另一个表中字段值的范围内选择

外键的条件：

表的存储引擎必须是innodb

字段类型要一致

被参照组字段必须要是索引类型的一种（主键）

基本用法

foreign key（表A的字段名）references 表B（字段名） on update cascade on delete cascade

删除外键字段

alter table 表名 drop foreign key 外键名;

如果有外键，不能删除被参考的主键的表和字段

**数据导入导出**

设置搜索路径

查看默认使用目录及目录是否存在：mysql>show variables like “secure\_file\_priv”;

修改目录及查看修改结果：

mkdir /email;chown mysql /email

vim /etc/my.cnf

[mysqld]

secure\_file\_priv=”/email”

systemctl restart mysqld

数据导入（把系统的文件内容存储到数据库里面）

基本用法

load data infile “目录名/文件名” into table 表名

fields terminated by “分隔符” //列以什么分隔

lines terminated by “\n”; //行以什么分隔

注意事项

字段分隔符要与文件一致

指定导入文件的绝对路径

导入数据的表字段类型要与文件字段匹配

禁用selinux保护机制

数据导出（表记录存储到系统文件里）

基本用法

sql的查询结果 into outfile “目录名/文件名”

fields terminated by “分隔符”

lines terminated by “\n”;

注意事项

导出的内容由sql查询语句决定

导出的是表中的记录，不包括字段名

禁用selinux

**管理表记录**

增加表记录

语法格式

格式1：添加1条记录，给所有字段赋值

insert into 表名 values(字段值列表);

格式2：添加N条记录，给所有字段赋值

insert into 表名 vlues(字段列表1),(字段列表2)...(字段列表N);

格式3：添加一条记录，给指定字段赋值

insert into 表名(字段名列表) values(字段值列表);

格式4：添加N条记录，给指定字段赋值

insert into 表名(字段名列表，中间用逗号隔开) vlues(字段列表1),(字段列表2)...(字段列表N);

注意事项

字段值与字段类型相匹配

对于字符类型的字段，要用双或单引号括起来

一次给所有字段赋值时，字段名可以省略

只给一部分字段赋值时，必须明确写出对应的字段名称

更新表记录

语法格式

格式1，更新表内所有记录

update 表名 set 字段1=字段1值,字段2=字段2值...;

格式2，只更新符合条件的部分记录

upodate 表名 set 字段1=字段1值，字段2=字段2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，要用单或双引号括起来

若不使用where限定条件，会更新所有记录

限定条件时，只更新匹配条件的记录

删除表记录

语法格式

格式1，仅删除符合条件的记录

delete from 表名 where 条件表达式;

格式2，删除所有表记录

delete from 表面;

查看表记录

语法格式

格式1：

select 字段1,...,字段N from 库.表 where条件表达式;

格式2

select 字段1,...,字段N from 表名;

注意事项

使用\*可匹配所有字段

指定表名时，可采用 库名.表名 的形式

**mysql存储引擎**

8个结构构成：管理工具、连接池、sql接口、分析器、优化器、缓存、存储引擎、硬盘

常见的存储引擎：innodb myisam

列出可用的存储引擎：show enignes或show engines\G

设置默认存储引擎

vim /etc/my.cnf

[mysqld]

default-storage-engine=存储引擎名称

systemctl resatrt mysqld

创建表时指定存储引擎

create table aa(aa int)engine=innodb;

修改表的存储引擎

alter table bb engine=myisam;

innodb存储引擎主要特点

支持行级锁定

支持事务、事务回滚、外键（ibdata1 ib\_logfile0 ib\_logfile1三个是事务日志）

相关表文件：.frm：存放表结构

.ibd：存放数据+索引

myisam存储引擎特点

支持页级锁定

不支持事务、事务回滚、外键

相关表文件：.frm：存放表结构

.MYI：存放索引信息

.MYD：存放数据

锁机制

表级锁：直接对整张进行加锁

行级锁：只锁定某一行

页级锁：（给内存里的东西加锁，1M就是一页）：对整个页面（Mysql管理数据的基本存储单位）进行加锁

读锁（共享锁）：支持并发读操作（进行select操作是读锁）

写锁（互斥锁、排它锁）：是独占锁，上锁期间其他线程不能读表或写表

事务特性

原子性：事务的整个操作是一个整体，不可分割，要么全部成功，要不全部失败

一致性：事务操作的前后，表中的记录没有变化

隔离性：事务操作是相关隔离不受影响的

持久性：数据一旦提交，不可更改，永久改变表数据

**匹配条件（where后面）**

数值比较（字段类型必须是数据数值类型）where 字段名 符号 数字

= 等于 >、>= 大于、大于或等于 <、<= 小于、小于或等于 != 不等于

字符比较/匹配空/非空（字符比较时，字段类型必须是字符类型，字符必须用双引号括起来）where 字段名 符号 “值”

= 相等 != 不相等 is null 匹配空 is not null 匹配非空

逻辑匹配（多个判断时使用）where 条件A 符号 条件B

or 逻辑或，多个条件满足一个即可 and 逻辑与，多个条件都得成立 ！非 () 提高优先级

范围内匹配/去重显示（匹配范围内的任意一个数值）where 字段名 in (值列表)

in(值列表) 在...里 not in(值列表) 不在...里 betweent 数字1 and 数字2 在..之间.. distinct字段名 去重显示，不显示重复值，只显示第一次出现的

模糊查询

where 字段名 like ‘通配符’（\_匹配单个字符、%匹配0-N个字符）

正则匹配

where 字段名 regexp ‘正则表达式’

正则元字符^ $ . [] \* |

四则运算（运算操作 字段必须是数值类型）select 字段名 运算操作 from 表

+ - \* / %

**操作查询结果**

聚集函数（字段类型必须是数值型的）select 函数 from 表名 where 匹配条件

avg（字段名） //统计字段的平均值

sum（字段名） //统计字段之和

min（字段名） //统计字段的最小值

max（字段名） //统计字段的最大值

count（字段名） //统计字段值个数

查询结果排序（默认升序）

sql的查询结果 order by （通常是数值类型字段） [asc|desc];

查询结果分组

sql的查询结果 group by 字段名（通常是字符类型字段）

查询结果过滤

sql的查询结果 having 表达式;

sql的查询结果 where 条件 having 条件表达式;

sql的查询结果 group by 字段名 having 条件表达式;

限制查询结果显示行数

sql的查询结果 limit N; //显示查询结果前N条记录

sql的查询结果 limit N,M; //显示指定范围内的查询记录（从0开始）

sql的查询结果 where 条件查询 limit 3; //显示查询结果前3条记录

sql的查询结果 where 条件查询 limit 3,3; //从第4条开始，共显示3条

**使用技巧：**

**在mysql命令行下面执行系统命令：加上system即可，也就是system+系统命令**