mysql

主要参数

/etc/my.cnf MySQL主配置文件

/var/lib/mysql 数据库目录

3306 默认端口号

mysqld MySQL主进程名称

TCP 传输协议

msyql:mysql 进程所有者与所属组

/var/log/mysqld.log 错误日志文件

常用的SQL命令分类:

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 数据事务语言(commit,rollback,savepoint)

首次进入mysql 需要查询自动生成的root密码

grep password /var/log/mysqld.log

修改root密码 若不修改则无法对数据库操作

方法一:

set global validate\_password\_policy=0; #只验证长度

set global validate\_password\_length=6; #修改密码长度,默认值是8个字符

alter user user() identified by "123456"; #修改登陆密码

方法二:

永久设置:

# vim /etc/my.cnf

#修改配置添加如下内容

[mysqld]

validate\_password\_policy=0

validate\_password\_length=6

###############################################################################

连接 MySQL 的方法:命令行、Web 网页、安装图形软件、编写脚本(PHP、java...)

连接MySQL数据库的命令语法格式:

mysql [-h服务器IP或域名 -u用户名 -p密码 数据库名称]

mysql -uroot -pchen -e 'show databases;' #非交互式对数据库进行操作

mysql> create database tts character set utf8mb4; #创建数据库并且定义整个库字符为万国字

mysql> drop database tts;

数据库命名规则

(数字、字母、下划线,不能纯数字;区分大小写;不能使用关键词或特殊符号)

\*\*特别重要不仅仅是数据库 也包括表中字段

创建数据表基本语法格式如下:

create table 数据库名称.数据表名称(

字段名1 数据类型(宽度) 约束条件,

字段名2 数据类型(宽度) 约束条件,

... ...

);

插入数据的语法格式:insert

into 数据库名称.数据表名称 values (值列表);

更新数据语法格式:update 数据库名称.数据表名称 set 字段=值 [where条件]

MySQL 数据类型

1. 字符类型

char(字符数) 固定长度,最大长度255字符,不够指定的字符数时自动在右边填补空格,超

出指定字符数则无法写入。

varchar(字符数) 可变长度,根据实际数据大小分配存储空间,超出指定字符数则无法写入。

text/blob 字符数大于65535时使用。

#超过了指定的 4 个字符,写入失败。

char(4)如果实际写入了 2 个字符,但是占用的空间是 4 个字符.

varchar(30)如果实际写入了 2 个字符,实际仅占用 2 个字符的空间.

2.数值类型

类型 范围(有符号) 范围(无符号) 用途

tinyint -128~127 0~255 微小整数

smallint -32768~32767 0~65535 小整数

mediumint 中整数

\*int 大整数

bigint 极大整数

\*float 单精度浮点数(M,D)

#(M,D),其中 M 为总宽度,D 为小数位数,M 应大于 D

reate table school.num(

id tinyint,

age int(3),

score float(4,2));

3. 日期时间类型

datetime日期时间类型,范围1000-01-01 00:00:00:000000~9999-12-31 23:59:59.999999

timestamp日期时间类型,范围1970-01-01 00:00:00:000000~2038-01-19 03:14:07.999999

date日期类型,范围0001-01-01~9999-12-31

默认使用4位数字表示,当只用2位数字负值时:

01~69自动识别为2001~2069

70~99自动识别为1970~1999

year年份类型,范围1901-2155

time时间类型,范围HH:MM:SS

4. 枚举类型(选择类型)

enum(值1,值2,值3...) #单选项

set(值 1,值 2,值 3...)

#多选项

################################################################

约束条件

1.常用约束条件:

条件约束 功能描述

null 允许为空,默认设置

not null 不允许为空

key 索引类型

default 设置默认值,缺省为NULL

修改表结构

add 添加字段

modify 修改字段类型

change 修改自定名称 \*\* 同时可以对字段类型也修改

drop 删除字段

rename 修改表名称

add 添加字段

alter table 表名 add 字段名 类型(宽度) 约束条件; 默认在最后添加

alter table info add age int(3) not null default 18;

通过after字段名可以将新添加的字段放到某个字段后面,或者first直接放到第一列

alter table info add age int(3) not null default 18 after name;

alter table info add age int(3) not null default 18 first;

---------------------------------------------------------------------------------------------------------------------------

modify 修改字段类型

alter table 表名 modify 字段名 类型(宽度) 约束条件;

alter table info modify age varchar(4) null default 19;

---------------------------------------------------------------------------------------------------------------------------

change 修改自定名称 \*\* 同时可以对字段类型也修改

alter table 表名 change 原字段名 新字段名 类型(宽度) 约束条件;

alter table info change age nianling int(4) nut null default 20;

---------------------------------------------------------------------------------------------------------------------------

drop 删除字段

alter table 表名 drop 字段名 ;

alter table info drop age ;

---------------------------------------------------------------------------------------------------------------------------

rename修改字段名称

alter table 表名 rename 新的数据表名称

##############################################################################

MySQL键值

/var/lib/mysql/mysql 中文件 .frm 结构 .MYD 数据 .MYI 索引

什么是索引:就是对数据表中的若干字段进行排序的方法,类似于对一本书做目录,有了目

录就可以快速定位数据的具体位置。

索引的优点:

➢ 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

➢

可以加快数据的检索速度

索引的缺点:

➢ 当对表中的数据进行增加、删除和修改的时候,索引也要动态的维护,会降低数据库

的写速度

➢

索引需要占额外的物理空间

\*\*\*做索引 可以使查询速度变快 但是会导致写入速度变慢

键值的类型

INDEX 普通索引 #最常用

UNIQUE 唯一索引 #不怎么用 他的功能主键都有

FULLTEXT 全文索引

PRIMARYKEY 主键 #常用

FOREIGNKEY 外键 #常用

---------------------------------------------------------------------------------------------------------------------------

1. 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,切可以赋NULL值

 经常把做查询条件的字段设置为INDEX字段

 INDEX字段的KEY标志是MUL

新建一个表格并创建索引:

create table school.info(

id int(6) not null,

name varchar(5),

sex enum('male','female'),

age int(3) default 1,

index(id),index(name));

在已有的数据表中创建或删除索引。

语法格式:create index 索引名称 on 数据表(字段名称)

create index age on school.info(age);

删除索引: drop index 索引名称 on 数据表(字段名称)

drop index name on school.info;

---------------------------------------------------------------------------------------------------------------------------

2. primary key主键索引

注意事项

 一个表中只能有一个primary key字段

 对应的字段值不允许有重复,且不允许赋NULL值

 如果有多个字段都作为PRIMARY KEY,称为复合主键,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[记录编号字段]

新建一个表格并创建主键:

create table school.student(

stu\_id char(9),

name char(5),

primary key(stu\_id));

或

create table school.student2(

id char(9) primary key,

name char(5));

对已经存在的数据表创建主键索引

ALTER TABLE 数据表名称 add primary key(字段名称);

alter table school.student3 add primary key(id);

删除数据表中的主键索引

alter table school.student3 drop primary key;

---------------------------------------------------------------------------------------------------------------------------

复合主键索引(多个字段做主键)

创建符合主键索引(姓名和单位,不能重复,但是单独的姓名或单位可以重复)

create table school.book(

姓名 char(20),

单位 char(20),

技能 int(100), primary key(姓名,单位));

#提示:单独的姓名重复是可以的

自动添加属性(可以自动将数据自动加1)

create table school.demo(

id int(100) auto\_increment primary key,

name char(10));

#必须为 int 类型

#当同时有自增长和主键时,一定要通过 alter 将自增长删除后才可以删除主键,无法直接删除主键

删除添加属性

alter table 表名 modify 类型(宽度);

alter table 表明 change 原字段名 原字段名 类型(宽度);

---------------------------------------------------------------------------------------------------------------------------

3. foreign key外键

什么是外键?

让当前表字段的值在另一个表中某个字段值的范围内选择。

使用外键的条件:

表存储引擎必须是innodb(默认就是)

字段的类型必须一致

被参照字段必须是索引类型中的一种(primary key)

创建外键的语法:

foreign key(表A的字段名称)

references 表B(字段名称)

on update casecade

#同步更新

on delete casecade

#同步删除