**将.txt文件转换为.ini文件的方法**

有时候需要创建一些后缀比较奇怪的文件,但是又没有办法直接创建,就需要创建成其他类型的文件,在通过一定的方法吧它转化成需要的文件,拿.txt文件转换为.ini文件举例:

打开.txt文件,依次文件->另存为,将保存类型改为"所有类型",将文件名后加上.ini后确定即可

[mysqld]

# 设置3306端口

port=3306

# 设置mysql的安装目录 ----------是你的文件路径-------------

basedir= C:\MySQLInstall\mysql-8.0.26-winx64

# 设置mysql数据库的数据的存放目录 ---------是你的文件路径data文件夹自行创建

datadir= C:\MySQLInstall\data

# 允许最大连接数

max\_connections=200

# 允许连接失败的次数。

max\_connect\_errors=10

# 服务端使用的字符集默认为utf8mb4

character-set-server=utf8mb4

# 创建新表时将使用的默认存储引擎

default-storage-engine=INNODB

# 默认使用“mysql\_native\_password”插件认证

#mysql\_native\_password

default\_authentication\_plugin=mysql\_native\_password

[mysql]

# 设置mysql客户端默认字符集

default-character-set=utf8mb4

[client]

# 设置mysql客户端连接服务端时默认使用的端口

port=3306

default-character-set=utf8mb4

# pymysql连接数据库报错：TypeError: \_\_init\_\_() takes 1 positional argument but 5 were given解决方案

报错原因

两个 pymysql 版本不一样，0.9.2 上可以运行，最新版 1.0.2 上面不能，最新版需要指定参数名，不能够省略。

import pymysql

db = pymysql.connect("127.0.0.1", "root", "password", "test")

解决方法

**将省略的参数添加即可。**

import pymysql

db = pymysql.connect(host="127.0.0.1", user="root", password="password", db="test")

# net start mysql 发生系统错误 5。 拒绝访问。解决方案

发现还报错误，拒绝访问  一般是权限不够，使用管理员身份运行cmd  并启动mysql服务

net start/stop mysql

# 创建和查询MySQL里的数据库

1. 连接 MySQL

输入 mysql -u root -p 命令，回车，然后输入 MySQL 的密码(不要忘记了密码)，再回车，就连接上 MySQL 了。

mysql -u root -p

2. 查看当前的数据库

使用 show databases; 查看当前安装的 MySQL 中有哪些数据库。

show databases;

刚安装 MySQL 时，默认有四个数据库，information\_schema，mysql，perfomance\_schema，sys 。通常情况下，我们不会直接使用这四个数据库，但千万不要把这四个数据库删了，否则会带来很多不必要的麻烦。如果不小心删了，建议是重新安装 MySQL ，在重装之前把自己的数据迁移出来备份好，或者从其他服务器上迁移一个相同的数据库过来。

3. 创建数据库

使用 create database 数据库名; 创建数据库。

create database MyDB\_one;

4. 创建数据库时设置字符编码

使用 create database 数据库名 character set utf8; 创建数据库并设置数据库的字符编码。

create database MyDB\_two character set utf8;

直接创建的数据库，数据库的编码方式是 MySQL 默认的编码方式 latin1 (单字节编码) ，通常我们会在数据库中存放中文数据，所以最好把数据库的编码方式设置成 utf-8 ，这样中文才能正常显示。

5. 查看和显示数据库的编码方式

使用 show create database 数据库名; 显示数据库的创建信息。

1. show create database MyDB\_one;
2. show create database MyDB\_two;

如果不知道一个数据库的编码方式是什么，可以使用 show create database 数据库名 来查看数据库的编码方式。可以看到刚才创建的 MyDB\_one 的编码方式是 MySQL 的默认编码 latin1 ，MyDB\_two 的编码方式是 utf-8 。

6. 使用 alter database 数据库名 character set utf8； 修改数据库编码

alter database MyDB\_one character set utf8;

当然，这种方式不能在创建的同时显示，只能查看一个已经存在的数据库的编码方式。

. 进入或切换数据库

使用 use 数据库名 进入或切换数据库。

1. use MyDB\_one
2. use MyDB\_two;

刚连接上 MySQL 时，没有处于任何一个数据库中，如果要使用某一个数据库，就需要进入到这个数据库中。

use 数据库名 这个命令后面的分号可以省略，这是 SQL 语句中唯一可以省略分号的语句。

8. 显示当前数据库 select database();

select database();

进入数据库中，可以使用 select database(); 来查看当前处于哪个数据库中。长时间操作数据库时，在很多数据库中来回切换后，查看当前的数据库，避免操作错了数据库。

## **三、创建数据表**

1. 查看当前数据库中的表

使用 show tables；查看当前数据库中有哪些表。

show tables;

2. 创建表

使用 create table 表名(字段1 字段类型,字段2 字段类型,字段3 字段类型,…); 来创建一张表。

create table Phone\_table(pid INT, name CHAR(20), price INT);

3. 显示表信息

用 show create table 表名; 来显示已创建的表的信息。

show create table Phone\_table;

使用 show create table 表名;  可以显示表的字段信息， MySQL 的引擎，和默认的字符编码等信息。与显示数据库信息一样，show 只能显示已经创建了的数据表的信息，不能在创建的同时显示信息。

4. 给表增加字段

使用 alter table 表名 add 字段名 数据类型; 为已存在的表添加一个新字段。

alter table Phone\_table add color CHAR(20);

5. 删除表的字段

使用 alter table 表名 drop 字段名; 删除一个表中已存在的字段。

alter table Phone\_table drop price;

6. 修改字段的数据类型

使用 alter table 表名 modify 字段名 数据类型; 修改表中现有字段的数据类型。

alter table Phone\_table modify name VARCHAR(12);

7. 修改字段的数据类型并且改名

使用 alter table 表名 change 原字段名 新字段名 数据类型; 修改表中现有字段的字段名和类型。

alter table Phone\_table change name pname CHAR(18);

## **四、MySQL 常用字段类型**

一个数据表是由若干个字段组成的，一个表十几个字段也很正常，每个字段表示不同的信息，需要使用不同类型的数据。

所以在创建表的时候，要为每个字段指定适合的数据类型。

MySQL 中常用的字段类型有以下这些：

1. 整数类型

2. 字符串类型

3. 小数类型

m 表示浮点数的总长度，n 表示小数点后有效位数。

5. 枚举类型

enum(枚举值1,枚举值2,...)

枚举类型只能在列出的值中选择一个，如性别。

create table student(sno INT, sname CHAR(20), age INT,sexINT );

alter table student modify sno CHAR(20);

# 获取当前时间戳并显示

保留10位

import time

import math

time\_stamp = int(time.time())

print(time\_stamp)#打印时间戳

Datatime=time.strftime("%Y-%m-%d %H:%M:%S",time.localtime(time\_stamp))

print(Datatime)#显示日期和时间

# 列表的显示

mes = [1,'44444','gd']

b =[mes,a]: [[1, '44444', 'gd'], [2, 'ssrt', '深深']]

c = mes + a [1, '44444', 'gd', 2, 'ssrt', '深深']

a =[2,'ssrt','深深']

b =[mes,a]

c=mes + a

print(b,c)

# 元组与列表之间的转换

a=[1,2,3,4]

b=('a',23)

c=list(b)

print(c)

print(tuple(a))