

```

import pymysql
import os
import pandas as pd

#pd.set_option()就是pycharm输出控制显示的设置
pd.set_option('expand_frame_repr', False)#True就是可以换行显示。设置成False的时候
不允许换行
pd.set_option('display.max_columns', None)# 显示所有列
#pd.set_option('display.max_rows', None)# 显示所有行
pd.set_option('colheader_justify', 'centre')# 显示居中

try:
    conn = pymysql.connect(host='localhost', user='root', password='你自己数据库的
密码', db='你自己数据库的名称', charset='utf8')
    cur = conn.cursor()
    print('数据库连接成功! ')
    print(' ')
except:
    print('数据库连接失败! ')

os.chdir('D:/stock_data/') #将路径设置成你csv文件放的地方
path = os.getcwd()
files = os.listdir(path)

i = 0 #计数器, 后面可以用来统计一共导入了多少个文件
for file in files:
    if file.split('.')[-1] in ['csv']: #判断文件是不是csv文件, file.split('.')[-1]获取 '.' 后的字
    符串
        i += 1
        filename = file.split('.')[0] #获取剔除后缀的名称
        filename = 'data_' + filename
        f = pd.read_csv(file, encoding='gbk') #用pandas读取文件, 得到pandas框架格式
        的数据
        columns = f.columns.tolist() #获取表格数据内的列标题文字数据

        types = f.dtypes #获取文件内数据格式
        field = [] #设置列表用来接收文件转换后的数据, 为写入mysql做准备
        table = []
        char = []
        for item in range(len(columns)): #开始循环获取文件格式类型并将其转换成mysql
        文件格式类型
            if 'object' == str(types[item]):
                char = '`' + columns[item] + '`' + ' VARCHAR(255)' #必须加上`这个点, 否则
                在写入mysql是会报错
            elif 'int64' == str(types[item]):
                char = '`' + columns[item] + '`' + ' INT'

```

```

elif 'float64' == str(types[item]):
    char = '`' + columns[item] + '`' + ' FLOAT'
elif 'datetime64[ns]' == str(types[item]):
    char = '`' + columns[item] + '`' + ' DATETIME'
else:
    char = '`' + columns[item] + '`' + ' VARCHAR(255)'
table.append(char)
field.append('`' + columns[item] + '`')

tables = ','.join(table) #将table中的元素用, 连接起来为后面写入mysql做准备
fields = ','.join(field)

cur.execute('drop table if exists {};'.format(filename))
conn.commit()

#创建表格并设置表格的列文字跟累数据格式类型
table_sql = 'CREATE TABLE IF NOT EXISTS ' + filename + '(' + 'id INT PRIMARY
KEY NOT NULL AUTO_INCREMENT,' + tables + ');'
print('表:' + filename + ',开始创建数据表...')
cur.execute(table_sql)
conn.commit()
print('表:' + filename + ',创建成功!')

print('表:' + filename + ',正在写入数据当中...')
f_sql = f.astype(object).where(pd.notnull(f), None) #将原来从csv文件获取到的
空值数据设置成None, 不设置将会报错
values = f_sql.values.tolist() #获取数值
s = ','.join(['%s' for _ in range(len(f.columns))]) #获得文件数据有多少列, 每个列
用一个 %s 替代
insert_sql = 'insert into {}({}) values({})'.format(filename,fields,s)
cur.executemany(insert_sql, values)
conn.commit()
print('表:' + filename + ',数据写入完成! ')
print(' ')
cur.close()
conn.close()
print('文件导入数据库完成! 一共导入了 {} 个CSV文件。'.format(i))

```