SOURCE CODE

# Setup.py

from os import name

import mysql.connector

import os.path as os

import json

import pandas as pd

from types import SimpleNamespace

import sys

ul\_msg = '''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

warning\_msg = '''

██     ██  █████  ██████  ███    ██ ██ ███    ██  ██████

██     ██ ██   ██ ██   ██ ████   ██ ██ ████   ██ ██

██  █  ██ ███████ ██████  ██ ██  ██ ██ ██ ██  ██ ██   ███

██ ███ ██ ██   ██ ██   ██ ██  ██ ██ ██ ██  ██ ██ ██    ██

 ███ ███  ██   ██ ██   ██ ██   ████ ██ ██   ████  ██████

DO NOT RUN THIS FILE IF YOU ARE NOT A DEVELOPER.

UNAUTHORIZED USE OF THIS FILE MAY RESULT IN PERMANENT WIPE OF DATA.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

exit\_msg = '''

████████ ██   ██  █████  ███    ██ ██   ██     ██    ██  ██████  ██    ██

   ██    ██   ██ ██   ██ ████   ██ ██  ██       ██  ██  ██    ██ ██    ██

   ██    ███████ ███████ ██ ██  ██ █████         ████   ██    ██ ██    ██

   ██    ██   ██ ██   ██ ██  ██ ██ ██  ██         ██    ██    ██ ██    ██

   ██    ██   ██ ██   ██ ██   ████ ██   ██        ██     ██████   ██████

'''

update\_msg =  '''

██    ██ ██████  ██████   █████  ████████ ███████ ██████

██    ██ ██   ██ ██   ██ ██   ██    ██    ██      ██   ██

██    ██ ██████  ██   ██ ███████    ██    █████   ██   ██

██    ██ ██      ██   ██ ██   ██    ██    ██      ██   ██

 ██████  ██      ██████  ██   ██    ██    ███████ ██████

'''

fail\_msg = '''

███████  █████  ██ ██      ███████ ██████

██      ██   ██ ██ ██      ██      ██   ██

█████   ███████ ██ ██      █████   ██   ██

██      ██   ██ ██ ██      ██      ██   ██

██      ██   ██ ██ ███████ ███████ ██████

'''

finish\_msg = '''

                ███████ ███████ ████████ ██    ██ ██████

                ██      ██         ██    ██    ██ ██   ██

                ███████ █████      ██    ██    ██ ██████

                     ██ ██         ██    ██    ██ ██

                ███████ ███████    ██     ██████  ██

 ██████  ██████  ███    ███ ██████  ██      ███████ ████████ ███████ ██████

██      ██    ██ ████  ████ ██   ██ ██      ██         ██    ██      ██   ██

██      ██    ██ ██ ████ ██ ██████  ██      █████      ██    █████   ██   ██

██      ██    ██ ██  ██  ██ ██      ██      ██         ██    ██      ██   ██

 ██████  ██████  ██      ██ ██      ███████ ███████    ██    ███████ ██████

'''

#startup

print(warning\_msg)

cont = input("Do you want to continue?(y/n) \n")

if cont != "y":

    sys.exit(exit\_msg)

#constant setup

def prettifier(value):

    value = value.replace(" ", "\_").lower()

    return(value)

change\_constants = False

print(ul\_msg)

if os.isfile("./constants.json"):

    with open('constants.json') as datafile:

        constants = pd.Series(json.load(datafile))

    print(constants)

    makechanges = input("Do you want to make any changes to these constants?(y/n)\n")

    if makechanges == "y":

        change\_constants = True

else:

    change\_constants = True

if change\_constants:

    print(ul\_msg)

    constants = {}

    constants['host'] = input("Enter MySQL host:\n")

    constants['user'] = input("Enter MySQL User:\n")

    constants['password'] = input("Enter MySQL Password:\n")

    constants['database'] = prettifier(input("Enter the name of the database to be used/created (Spaces will be replaced with an underscore):\n"))

    constants['tablename'] = prettifier(input("Enter the name of the table to be used/created:\n"))

    with open("./constants.json", "w") as constantfile:

        json.dump(constants, constantfile)

    print(update\_msg)

    print(f'{constantfile.name} has been updated successfully')

    print(f'New constants: \n{constants}')

    print(ul\_msg)

#mysql setup

with open('constants.json') as datafile:

    constants = SimpleNamespace(\*\*json.load(datafile))

tablename = constants.tablename

host = constants.host

password = constants.password

user = constants.user

database = constants.database

try:

    mydb = mysql.connector.connect(host=host,

                                   user=user,

                                   password=password)

    mycursor = mydb.cursor(buffered=True)

    mycursor.execute("SHOW DATABASES")

    for (x,) in mycursor:

        if x == database:

            break

    else:

        mycursor.execute(f"CREATE DATABASE {database}")

        print(f"\n\n\nNew database {database} created successfully")

    mycursor.execute(f"use {database}")

    mycursor.execute("SHOW TABLES")

    tablelist = []

    create = True

    for (x,) in mycursor:

        tablelist.append(x)

    if tablename in tablelist:

        val = input(f"\n\n\nData Backup Found."

                    f"The table {tablename} already exists in the database."

                    f"Do you want to use the backup or delete the table and start fresh.?"

                    f"WARNING: starting fresh will remove all existing data in the table permanently."

                    f"(y/n):")

        if val == 'y':

            mycursor.execute(f"DROP TABLE {tablename}")

        else:

            create = False

    if create:

        query = f"CREATE TABLE {tablename} (Id int(10) PRIMARY KEY AUTO\_INCREMENT, Name varchar(100) NOT NULL, Phone int(10), Purchased\_Date Date NOT NULL, Item varchar(50) NOT NULL, Last\_Modified char(30) NOT NULL)"

        mycursor.execute(f"CREATE TABLE {tablename} ("

                         "Id int(10) PRIMARY KEY AUTO\_INCREMENT, "

                         "Name varchar(100) NOT NULL, "

                         "Phone varchar(10), "

                         "Purchased\_Date Date NOT NULL, "

                         "Item varchar(50) NOT NULL, "

                         "Price int(10),"

                         "Last\_Modified char(30) NOT NULL"

                         ")"

                         )

        print(f"\n\n\nNew Table {tablename} created successfully")

    print(ul\_msg)

    finish = True

except mysql.connector.Error as error:

    print(fail\_msg)

    print(f"ERROR: {error}")

    print(ul\_msg)

    finish = False

#completed

if finish:

    print(finish\_msg)

else:

    print("Please restart the file to fix the errors and setup properly.")

    print(ul\_msg)

# Menu.py

from datetime import datetime

import json

from types import SimpleNamespace

keyMenu = '''

+-----+---------+--------------------------------------+

| Key | Command | Description                          |

+-----+---------+--------------------------------------+

| 1   | ID      | Search using Customer's ID           |

| 2   | Ph. No  | Search using Customer's Phone Number |

+-----+---------+--------------------------------------+

Press a key to continue. (1,2)

'''

viewMenu = '''

+-----+--------------+---------------------------+

| Key | Command      | Description               |

+-----+--------------+---------------------------+

| 1   | Full Table   | Display the full table    |

| 2   | Custom Range | Display a specific ranges |

+-----+--------------+---------------------------+

Press a key to continue. (1,2)

'''

customMenu = '''

+-----+----------------+-----------------------------------------------+

| Key | Command        | Description                                   |

+-----+----------------+-----------------------------------------------+

| 1   | ID             | Display records using Bill ID                 |

| 2   | Name           | Display records using Customer's Name         |

| 3   | Ph. No.        | Display records using Customer's Phone Number |

| 4   | Purchased Date | Display records using Purchased date          |

| 5   | Item           | Display records using Purchased Item          |

| 6   | Price          | Display records using Price                   |

+-----+----------------+-----------------------------------------------+

Press a key to continue. (1,2,3,4,5,6)

'''

#constants

with open('constants.json') as datafile:

    constants = SimpleNamespace(\*\*json.load(datafile))

tablename = constants.tablename

host = constants.host

password = constants.password

user = constants.user

database = constants.database

#functions

def new\_menu():

    name = input("Enter name:\n")

    ph\_no = int(input("Enter phone number:\n"))

    up\_date = (datetime.now()).strftime("%d/%m/%Y %H:%M:%S")

    pr\_date = (datetime.now()).strftime("%Y-%m-%d")

    item = input("Enter Item:\n")

    price = int(input("Enter Price of the item:\n"))

    dict = {"Name": f"'{name}'",

            "Phone": f"'{ph\_no}'",

            "Purchased\_Date": f"'{pr\_date}'",

            "Last\_Modified": f"'{up\_date}'",

            "Item": f"'{item}'",

            "Price": f"{price}"

            }

    columns = ', '.join(str(x) for x in dict.keys())

    values = ', '.join(str(x) for x in dict.values())

    query = f"INSERT INTO {tablename} ({columns}) VALUES ({values})"

    return query

def update\_menu():

    value = int(input("Enter Customer's ID:\n"))

    name = input("Enter Updated Name \n(Enter $ if you dont want to update it):\n")

    ph\_no = input("Enter Updated Phone Number \n(Enter $ if you dont want to update it):\n")

    up\_date = (datetime.now()).strftime("%d/%m/%Y %H:%M:%S")

    pr\_date = input("Enter Updated Purchased date (Format: YYYY-MM-DD) \n(Enter $ if you dont want to update it):\n")

    item = input("Enter Updated Item \n(Enter $ if you dont want to update it):\n")

    price = input("Enter Updated Price \n(Enter $ if you dont want to update it):\nZ")

    dict = {"Name": f"'{name}'",

            "Phone": f"'{ph\_no}'",

            "Purchased\_Date": f"'{pr\_date}'",

            "Last\_Modified": f"'{up\_date}'",

            "Item": f"'{item}'",

            "Price": price

            }

    query = ""

    for i in dict:

        if dict[i] not in ["'$'", "$"] :

            query += f"{i} = {dict[i]}, "

    if query != "":

        query = f"UPDATE {tablename} SET " + query

        query = query[:-2] + f" WHERE Id = {value}"

    return query

def delete\_menu():

    id = int(input("Enter Customer's ID:\n"))

    query = f"DELETE FROM {tablename} WHERE Id = {id}"

    return query

def view\_menu():

    selection = int(input(viewMenu))

    if selection == 1:

        query = f"SELECT \* FROM {tablename}"

    elif selection == 2:

        selection = int(input(customMenu))

        criteria, value = "", ""

        if selection == 1:

            criteria = "Id"

            value = int(input("Enter Customer's ID:\n"))

        elif selection == 2:

            criteria = "Name"

            value = f"\'{input('Enter Customers Name: ')}\'"

        elif selection == 3:

            criteria = "Phone"

            value = f"\'{input('Enter Customers Phone Number: ')}\'"

        elif selection == 4:

            criteria = "Purchased\_Date"

            value = f"\'{input('Enter Purchased Date (Format: YYYY-MM-DD): ')}\'"

        elif selection == 5:

            criteria = "Item"

            value = f"\'{input('Enter the Item Purchased: ')}\'"

        elif selection == 6:

            criteria = "Price"

            value = int(input("Enter Customer's Phone Number:\n"))

        if criteria == "":

            query = "select \'Please\'"

            print("Please enter a correct key")

        elif value == "":

            query = "select \'Please\'"

            print("Please enter a value")

        else:

            query = f"SELECT \* FROM {tablename} WHERE {criteria} = {value}"

    else:

        query = "select \'Please\'"

        print("Please enter a correct key")

    return query

# Program.py

import mysql.connector

import json

from types import SimpleNamespace

import pandas as pd

import matplotlib.pyplot as plt

from tabulate import tabulate

import menu

import sys

welcome\_msg = '''

██████  ██ ██      ██      ██ ███    ██  ██████

██   ██ ██ ██      ██      ██ ████   ██ ██

██████  ██ ██      ██      ██ ██ ██  ██ ██   ███

██   ██ ██ ██      ██      ██ ██  ██ ██ ██    ██

██████  ██ ███████ ███████ ██ ██   ████  ██████

███████ ██    ██ ███████ ████████ ███████ ███    ███

██       ██  ██  ██         ██    ██      ████  ████

███████   ████   ███████    ██    █████   ██ ████ ██

     ██    ██         ██    ██    ██      ██  ██  ██

███████    ██    ███████    ██    ███████ ██      ██

𝐏𝐫𝐨𝐣𝐞𝐜𝐭 𝐛𝐲

𝐀𝐡𝐚𝐦𝐦𝐞𝐝 𝐘𝐚𝐬𝐢𝐧,

𝐌𝐨𝐡𝐚𝐦𝐦𝐞𝐝 𝐒𝐡𝐨𝐮𝐤𝐚𝐭,

𝐉𝐨𝐬𝐡𝐮𝐚 𝐕𝐚𝐫𝐠𝐡𝐞𝐬𝐞 𝐒𝐚𝐧𝐭𝐡𝐨𝐬𝐡

𝟏𝟐. 𝐁

𝐎𝐮𝐫 𝐎𝐰𝐧 𝐄𝐧𝐠𝐥𝐢𝐬𝐡 𝐇𝐢𝐠𝐡 𝐒𝐜𝐡𝐨𝐨𝐥, 𝐅𝐮𝐣𝐚𝐢𝐫𝐚𝐡, 𝐔𝐀𝐄

'''

exit\_msg = '''

████████ ██   ██  █████  ███    ██ ██   ██     ██    ██  ██████  ██    ██

   ██    ██   ██ ██   ██ ████   ██ ██  ██       ██  ██  ██    ██ ██    ██

   ██    ███████ ███████ ██ ██  ██ █████         ████   ██    ██ ██    ██

   ██    ██   ██ ██   ██ ██  ██ ██ ██  ██         ██    ██    ██ ██    ██

   ██    ██   ██ ██   ██ ██   ████ ██   ██        ██     ██████   ██████

'''

ul\_msg = '''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

main\_menu = '''

+-----+---------+------------------------------+

| Key | Command | Description                  |

+-----+---------+------------------------------+

| 1   | NEW     | Create a New Entry           |

| 2   | UPDATE  | Update an Old Entry          |

| 3   | DELETE  | Delete an Entry              |

| 4   | VIEW    | Return Entry/Range of Entries|

| 5   | GRAPHS  | Graph Options                |

| 6   | EXIT    | Closes the server connection |

|     |         | and exits the program.       |

+-----+---------+------------------------------+

Press a key to continue. (1,2,3,4,5,6)

'''

graphMenu = '''

+-----+---------+-------------------------+

| Key | Command |       Description       |

+-----+---------+-------------------------+

|   1 | Item    | Plots a graph based on  |

|     |         | items purchased         |

|   2 | Date    | Plots a graph based on  |

|     |         | monthly collection      |

+-----+---------+-------------------------+

Press a key to continue. (1,2)

'''

#constants

try:

    with open('constants.json') as datafile:

        constants = SimpleNamespace(\*\*json.load(datafile))

    tablename = constants.tablename

    host = constants.host

    password = constants.password

    user = constants.user

    database = constants.database

except:

    sys.exit("Database not setup properly. Please contact the developer.")

#mysql

run = False

try:

    mydb = mysql.connector.connect(host=host,

                                   database=database,

                                   user=user,

                                   password=password)

    if mydb.is\_connected():

        mycursor = mydb.cursor(buffered=True)

        mycursor.execute("SELECT DATABASE()")

        db = mycursor.fetchone()

        print("You're connected to: ", db)

        mycursor.execute("SET AUTOCOMMIT=1")

        run = True

except mysql.connector.Error as error:

    print("Error while connecting to MySQL", error)

#functions

def execute(query):

    try:

        mycursor.execute(query)

        print(f"{tablename} has been modified with the query: {query}")

    except mysql.connector.Error as error:

        print("Error while trying to modify, ", error)

def select(query):

    try:

        records = pd.read\_sql\_query(query, mydb)

        print(tabulate(records, tablefmt="pretty", headers="keys"))

    except mysql.connector.Error as error:

        print("ERROR: ", error)

def sql\_to\_df(query):

    try:

        df = pd.read\_sql\_query(query, mydb)

        return df

    except mysql.connector.Error as error:

        print("ERROR: ", error)

def graph\_menu():

    selection = int(input(graphMenu))

    if selection == 1 :

        query = f"SELECT Item, COUNT(\*) from {tablename} GROUP BY Item"

        records = sql\_to\_df(query)

        records.plot(kind= 'bar', x='Item', y= 'COUNT(\*)')

        plt.show()

    if selection == 2:

        query = f'SELECT EXTRACT(MONTH FROM Purchased\_Date) "Month", \

SUM(Price) FROM {tablename} GROUP BY EXTRACT(MONTH FROM Purchased\_Date)'

        records = sql\_to\_df(query)

        records.plot(kind='bar', x='Month', y='SUM(Price)')

        plt.show()

def close():

    print("CLOSING. PLEASE WAIT...")

    if (mydb.is\_connected()):

        mycursor.close()

        mydb.close()

        print("MySQL connection is closed")

    print(exit\_msg)

#program

print(welcome\_msg)

while run == True:

    print(ul\_msg)

    selection = int(input(main\_menu))

    if selection == 1 :

        query = menu.new\_menu()

        execute(query)

    elif selection == 2 :

        query = menu.update\_menu()

        execute(query)

    elif selection == 3 :

        query = menu.delete\_menu()

        execute(query)

    elif selection == 4 :

        query = menu.view\_menu()

        select(query)

    elif selection == 5 :

        query = graph\_menu()

    elif selection == 6:

        close()

        run = False

    else:

        print("Please enter a correct key.")