import sqlite3

import time

import os

conexion = sqlite3.connect("ejemplo.db")

cursor = conexion.cursor()

resp = ""

def creartabla():

try:

cursor.execute("CREATE TABLE usuarios (nombre TEXT, edad NUMERIC, email TEXT , dni TEXT)")

conexion.commit()

cursor.close()

conexion.close()

except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def insertar():

cursor.execute("INSERT INTO usuarios VALUES ('Charlie', 25, 'charlie@ejemplo.com','10210902')")

conexion.commit()

def devolveregistro():

cursor.execute("SELECT \* FROM usuarios")

usuario = cursor.fetchone()

print usuario

def insertargrupo():

usuarios = [('Maria', 23, 'maria@ejemplo.com','10210903'),

('Mercedes', 38, 'mercedes@ejemplo.com','10210904'),

('Juanita', 19, 'juanita@ejemplo.com','10210905')

]

cursor.executemany("INSERT INTO usuarios VALUES (?,?,?,?)", usuarios)

conexion.commit()

def listartodos():

cursor.execute("SELECT \* FROM usuarios")

usuarios = cursor.fetchall()

for usuario in usuarios:

print('Nombre: ' + usuario[0] + ' Edad:' + str(usuario[1]) + ' Correo:' + usuario[2] + ' DNI:' + usuario[3] )

def modificarregistro():

cursor.execute("UPDATE usuarios SET nombre ='Charlie Sanchez', edad=26 WHERE dni='10210902'")

conexion.commit()

def eliminar():

cursor.execute("DELETE FROM usuarios WHERE dni='10210902'")

conexion.commit()

def menu():

print ("Menu Opciones")

print ("1.-Crear Tabla")

print ("2.-Insertar")

print ("3.-Devolver Registro")

print ("4.-Insertar Varios")

print ("5.-Listar")

print ("6.-Modificar")

print ("7.-Eliminar")

opc =raw\_input("Ingrese Opcion==>")

if opc=="1":

creartabla()

elif opc=="2":

insertar()

elif opc=="3":

devolveregistro()

elif opc=="4":

insertargrupo()

elif opc=="5":

listartodos()

elif opc=="6":

modificarregistro()

elif opc=="7":

eliminar()

while resp!='N':

menu()

resp=raw\_input("desea continuar:").upper()

os.system('cls')

time.sleep(3)

FIN

import sqlite3

import os

import time

bd = sqlite3.connect("tpv.db")

cursor = bd.cursor()

resp=""

def cargainventario():

    try:

        productos = [

        """

        INSERT INTO productos(nombre, precio, estado, codcat,barra)

        values ("Pizza Pollo",28.7,"1",1,"10210902") ,

        ("Pizza Vegetariana",18.3,"1",1,"10210903") ,

        ("Pizza Quezo",16.2,"1",1,"10210904") ,

        ("Pizza Carne",21.3,"1",2,"10210905") ,

        ("Pizza Fruta",30.1,"1",2,"10210906")

        """

    ]

        for sentencia in productos:

            cursor.execute(sentencia)

            bd.commit() #Guardamos los cambios al terminar el ciclo

        print("Productos insertados correctamente")

    except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def listar():

    try:

        sentencia = "SELECT nombre,precio,estado,codcat FROM productos;"

        cursor.execute(sentencia)

        productos = cursor.fetchall()

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

        print("|{:^50}|{:^20}|{:^10}|{:^10}|".format("Nombre", "Precio", "Estado", "Categoria"))

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

        for nombre, precio, estado, codcat in productos:

            print("|{:^50}|{:^20}|{:^10}|{:^10}|".format(nombre, precio, estado, codcat))

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

    except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def buscar():

    try:

        busqueda = int(raw\_input("Ingrese o pistole el codigo de barra: "))

        sentencia = "SELECT nombre,precio,estado,codcat FROM productos WHERE barra = "+ str(busqueda)

        cursor.execute(sentencia)

        productos = cursor.fetchall()

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

        print("|{:^50}|{:^20}|{:^10}|{:^10}|".format("Nombre", "Precio", "Estado", "Categoria"))

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

        for nombre, precio, estado, codcat in productos:

            print("|{:^50}|{:^20}|{:^10}|{:^10}|".format(nombre, precio, estado, codcat))

        print("+{:-<50}+{:-<20}+{:-<10}+{:-<10}+".format("", "", "", ""))

    except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def eliminar():

    try:

        busqueda = int(raw\_input("Ingrese el codigo del producto: "))

        sentencia = "delete FROM productos WHERE codigo = "+ str(busqueda)

        cursor.execute(sentencia)

        bd.commit()

        listar()

    except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def actualizar():

    try:

        busqueda = int(raw\_input("Ingrese el codigo del producto: "))

        nomprod = raw\_input("Ingrese el nombre del producto: ")

        preprod = int(raw\_input("Ingrese el precio del producto: "))

        sentencia =" UPDATE productos SET nombre = " + "'" + nomprod +"'," + " precio = " + str(preprod) + " WHERE codigo =" + str(busqueda)

        cursor.execute(sentencia)

        bd.commit()

        listar()

    except sqlite3.OperationalError as error:

        print("Error al abrir:", error)

def menuprincipal():

opc="0"

print ("menu principal")

print ("1.-Carga de Inventario")

print ("2.-Listado")

print ("3.-Buscar")

print ("4.-Eliminar")

print ("5.-Actualizar")

opc=raw\_input("Elija Opcion:")

if opc=="1":

cargainventario()

elif opc=="2":

listar()

elif opc=="3":

buscar()

elif opc=="4":

eliminar()

elif opc=="5":

actualizar()

else:

print ("Opcion No Existe")

while resp!='N':

menuprincipal()

resp=raw\_input("desea continuar:").upper()

os.system('cls')