CampusFP

Mario del Rey

EJ\_PY\_18\_PROG

# Ejercicio: CRUD Básico sobre la Tablas Pedidos y Detalles

## Código

## Inicio.py

import funciones

conexion=funciones.conectar("SUPERMERCADO")

while True:

    funciones.menu\_principal()

    opcion=int(input("introduce la tabla: "))

    if opcion==1:

        funciones.menu\_operacion()

        elecion=int(input("introduce la operacion que quierres haces"))

        if elecion==1:

            funciones.crear\_categoria(conexion)

        elif elecion==2:

            funciones.leer\_categoria(conexion)

        elif elecion==3:

            funciones.actualizar\_categoria(conexion)

        elif elecion==4:

            funciones.eliminar\_categoria(conexion)

        elif elecion==5:

            print("Saliendo de la gestion de categorias. Hasta pronto")

            break

    elif opcion==2:

        funciones.menu\_operacion()

        elecion=int(input("introduce la operacion que quierres haces"))

        if elecion==1:

            funciones.crear\_detalle(conexion)

        elif elecion==2:

            funciones.leer\_detalle(conexion)

        elif elecion==3:

            funciones.actualizar\_detalle(conexion)

        elif elecion==4:

            funciones.eliminar\_detalles(conexion)

        elif elecion==5:

            print("Saliendo de la gestion de detalle. Hasta pronto")

            break

    elif opcion==3:

        funciones.menu\_operacion()

        elecion=int(input("introduce la operacion que quierres haces"))

        if elecion==1:

            funciones.crear\_pedidos(conexion)

        elif elecion==2:

            funciones.leer\_pedidos(conexion)

        elif elecion==3:

            funciones.actualizar\_pedido(conexion)

        elif elecion==4:

            funciones.eliminar\_pedido(conexion)

        elif elecion==5:

            print("Saliendo de la gestion de pedido. Hasta pronto")

            break

    elif opcion==4:

        funciones.menu\_operacion()

        elecion=int(input("introduce la operacion que quierres haces"))

        if elecion==1:

            funciones.crear\_cliente(conexion)

        elif elecion==2:

            funciones.leer\_cliente(conexion)

        elif elecion==3:

            funciones.actualizar\_cliente(conexion)

        elif elecion==4:

            funciones.eliminar\_cliente(conexion)

        elif elecion==5:

            print("Saliendo de la gestion de cliente. Hasta pronto")

            break

    elif opcion==5:

        funciones.menu\_operacion()

        elecion=int(input("introduce la operacion que quierres haces"))

        if elecion==1:

            funciones.crear\_producto(conexion)

        elif elecion==2:

            funciones.leer\_producto(conexion)

        elif elecion==3:

            funciones.actualizar\_producto(conexion)

        elif elecion==4:

            funciones.eliminar\_producto(conexion)

        elif elecion==5:

            print("Saliendo de la gestion de producto. Hasta pronto")

            break

    elif opcion==6:

        conexion.close

        break

## funciones.py

import mysql.connector

def conectar(basedatos):

    try:

        conexion=mysql.connector.connect(

            host="localhost",

            user="root",

            password="curso",

            database=basedatos,

        )

        if conexion.is\_connected():

            print(f"conexion a {basedatos} exitosa")

        return conexion

    except mysql.connector.Error as err:

        print(f"error al conectar con {basedatos}: {err}")

        return None

def menu\_principal():

    print("Menu tablas")

    print("1. Categorias")

    print("2. Detalle")

    print("3. Pedidos")

    print("4. Clientes")

    print("5. Producto")

    print("6. Salir")

def menu\_operacion():

    print("Menu de Operaciones")

    print("1. Crear")

    print("2. Listar")

    print("3. Actualizar")

    print("4. Borrar")

    print("5. Salir")

def crear\_categoria(conexion):

    try:

        cursor=conexion.cursor()

        id=int(input("introduce el id de la nueva categoria: "))

        nombre=input("introduce el nombre de la nueva categoria: ")

        consulta="""INSERT INTO categoria VALUES(%s,%s)"""

        cursor.execute(consulta,(id, nombre))

        conexion.commit()

        print(f"la categoria {nombre} se a creado correctamente")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al crear categoría: {err}")

def leer\_categoria(conexion):

    cursor=conexion.cursor()

    consulta="""SELECT \* FROM categoria;""";

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    print("\n--- Categorías ---")

    for fila in resultados:

        print(f"ID: {fila[0]}, Nombre: {fila[1]}")

    cursor.close()

def actualizar\_categoria(conexion):

    try:

        cursor = conexion.cursor()

        idcategoria = input("Introduce el ID de la categoría a actualizar: ")

        nombre = input("Introduce el nuevo nombre de la categoría: ")

        idcategoria = int(idcategoria)

        consulta = """UPDATE categoria SET categoria = %s WHERE idcategoria = %s"""

        cursor.execute(consulta, (nombre, idcategoria))

        conexion.commit()

        print(f"La categoría con ID {idcategoria} ha sido actualizada a {nombre}.")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al actualizar categoría: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def eliminar\_categoria(conexion):

    try:

        cursor = conexion.cursor()

        idcategoria = input("Introduce el ID de la categoría a eliminar:")

        idcategoria = int(idcategoria)

        consulta = """DELETE FROM categoria WHERE idcategoria = %s"""

        cursor.execute(consulta, (idcategoria,))

        conexion.commit()

        if cursor.rowcount > 0:

            print(f"La categoría con ID {idcategoria} ha sido eliminada.")

        else:

            print(f"No se encontró ninguna categoría con ID {idcategoria}.")

        cursor.close()

    except mysql.connector.Error as err:

        print("Error al eliminar categoría: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def crear\_detalle(conexion):

    try:

        cursor=conexion.cursor()

        idpedido=int(input("introduce la id del pedido: "))

        idproducto=int(input("introduce la id del producto: "))

        precio=float(input("introduce el precio: "))

        unidades=int(input("introduce el numero de unidades: "))

        descuento=float(input("introduce el descuento: "))

        cursor.execute("SELECT COUNT(\*) FROM pedido WHERE idpedido = %s", (idpedido,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un pedido con ID {idpedido}.")

            return

        cursor.execute("SELECT COUNT(\*) FROM producto WHERE idproducto = %s", (idproducto,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un producto con ID {idproducto}.")

            return

        consulta="""INSERT INTO detalle VALUES(%s,%s,%s,%s,%s)"""

        cursor.execute(consulta,(idpedido, idproducto, precio, unidades, descuento))

        conexion.commit()

        print(f"los detalles de {idpedido} y {idproducto} se a creado correctamente")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al crear categoría: {err}")

def leer\_detalle(conexion):

    cursor=conexion.cursor()

    consulta="""SELECT \* FROM detalle;""";

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    print("\n--- detalle ---")

    for fila in resultados:

        print(f"idpedido: {fila[0]}, idproducto: {fila[1]}, precio: {fila[2]}, unidades: {fila[3]}, descuento: {fila[4]}")

    cursor.close()

def actualizar\_detalle(conexion):

    try:

        cursor = conexion.cursor()

        idpedido=int(input("introduce la id del pedido: "))

        idproducto=int(input("introduce la id del producto: "))

        precio=float(input("introduce el precio: "))

        unidades=int(input("introduce el numero de unidades: "))

        descuento=float(input("introduce el descuento: "))

        cursor.execute("SELECT COUNT(\*) FROM pedido WHERE idpedido = %s", (idpedido,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un pedido con ID {idpedido}.")

            return

        cursor.execute("SELECT COUNT(\*) FROM producto WHERE idproducto = %s", (idproducto,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un producto con ID {idproducto}.")

            return

        consulta = """UPDATE detalle SET precio = %s, unidades = %s, descuento = %s WHERE idpedido = %s AND idproducto = %s"""

        cursor.execute(consulta, (precio,unidades, descuento, idpedido, idproducto))

        conexion.commit()

        print(f"El detalle con ID Pedido {idpedido} y Producto {idproducto} ha sido actualizado.")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al actualizar detalle: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def eliminar\_detalles(conexion):

    try:

        cursor = conexion.cursor()

        idpedido=int(input("introduce la id del pedido: "))

        idproducto=int(input("introduce la id del producto: "))

        cursor.execute("SELECT COUNT(\*) FROM pedido WHERE idpedido = %s", (idpedido,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un pedido con ID {idpedido}.")

            return

        cursor.execute("SELECT COUNT(\*) FROM producto WHERE idproducto = %s", (idproducto,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un producto con ID {idproducto}.")

            return

        consulta = """DELETE FROM detalle WHERE idpedido = %s and idproducto = %s"""

        cursor.execute(consulta, (idpedido, idproducto))

        conexion.commit()

        print(f"el detalle con idpedido {idpedido} y la idproducto{idproducto} ha sido eliminada.")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al eliminar cdetalle: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def crear\_pedidos(conexion):

    try:

        cursor=conexion.cursor()

        id=int(input("introduce el id de la nueva pedido: "))

        idcliente=input("introduce el id del cliente: ")

        fechapedido=input("introduce la fecha de inicio del pedido: ")

        fechaentrega=input("introduce la fecha de entrega del pedido: ")

        cursor.execute("SELECT COUNT(\*) FROM pedido WHERE idpedido = %s", (idcliente,))

        if cursor.fetchone()[0] == 0:

            print(f"No existe un pedido con ID {idcliente}.")

            return

        consulta="""INSERT INTO categoria VALUES(%s,%s,%s,%s)"""

        cursor.execute(consulta,(id, idcliente, fechapedido, fechaentrega))

        conexion.commit()

        print(f"el pedido con id {id} se a creado correctamente")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al crear categoría: {err}")

def leer\_pedidos(conexion):

    cursor=conexion.cursor()

    consulta="""SELECT \* FROM pedido;""";

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    print("\n--- pedidos ---")

    for fila in resultados:

        print(f"ID: {fila[0]}, idcliente: {fila[1]}, fechapedido: {fila[2]}, fechaentrega: {fila[3]}")

    cursor.close()

def actualizar\_pedido(conexion):

    try:

        cursor = conexion.cursor()

        id=input("Introduce el ID de la pedido a actualizar: ")

        fechaentrega=input("introduce la nueva fecha de entrega")

        consulta = """UPDATE pedido SET pedido = %s WHERE fechaentrega = %s"""

        cursor.execute(consulta, (id, fechaentrega))

        conexion.commit()

        print(f"el pedido con ID {id} ha actualizado sus fecha de entrega {fechaentrega}.")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al actualizar pedido: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def eliminar\_pedido(conexion):

    try:

        cursor = conexion.cursor()

        id=input("Introduce el ID del pedido a eliminar:")

        id = int(id)

        consulta = """DELETE FROM pedido WHERE idpedido = %s"""

        cursor.execute(consulta, (id))

        conexion.commit()

        if cursor.rowcount > 0:

            print(f"El pedido con ID {id} ha sido eliminada.")

        else:

            print(f"No se encontró ningun pedido con ID {id}.")

        cursor.close()

    except mysql.connector.Error as err:

        print(f"Error al eliminar pedido: {err}")

    except ValueError:

        print("El ID debe ser un número entero.")

def crear\_cliente(conexion):

    cursor = conexion.cursor()

    idcliente = input("Introduce el ID del cliente (máximo 5 caracteres):")

    cia = input("Introduce el nombre de la compañía: ")

    contacto = input("Introduce el nombre del contacto: ")

    cargo = input("Introduce el cargo del contacto: ")

    direccion = input("Introduce la dirección: ")

    ciudad = input("Introduce la ciudad: ")

    region = input("Introduce la región: ")

    cp = input("Introduce el código postal: ")

    pais = input("Introduce el país: ")

    tlf = input("Introduce el teléfono: ")

    fax = input("Introduce el fax: ")

    consulta = """INSERT INTO cliente (idcliente, cia, contacto, cargo,direccion, ciudad, region, cp, pais, tlf, fax)VALUES (%s, %s, %s, %s, %s,%s, %s, %s, %s, %s, %s)""";

    cursor.execute(consulta, (idcliente, cia, contacto, cargo, direccion,ciudad, region, cp, pais, tlf, fax))

    conexion.commit()

    print("Cliente &#39;{cia}&#39; creado con éxito.")

    cursor.close()

def leer\_cliente(conexion):

    cursor = conexion.cursor()

    consulta = "SELECT \* FROM cliente"

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    print("\n--- Clientes ---")

    for fila in resultados:

        print("ID: {fila[0]}, Compañía: {fila[1]}, Contacto: {fila[2]},Cargo: {fila[3]}")

    cursor.close()

def actualizar\_cliente(conexion):

    cursor = conexion.cursor()

    idcliente = input("Introduce el ID del cliente a actualizar: ")

    nuevo\_contacto = input("Introduce el nuevo nombre del contacto: ")

    nueva\_cia = input("Introduce el nuevo nombre de la compañía: ")

    consulta = "UPDATE cliente SET contacto = %s, cia = %s WHERE idcliente = "

    cursor.execute(consulta, (nuevo\_contacto, nueva\_cia, idcliente))

    conexion.commit()

    print("Cliente con ID {idcliente} actualizado.")

    cursor.close()

def eliminar\_cliente(conexion):

    cursor = conexion.cursor()

    idcliente = input("Introduce el ID del cliente a eliminar: ")

    consulta = """DELETE FROM cliente WHERE idcliente = %s""";

    cursor.execute(consulta, (idcliente,))

    conexion.commit()

    print("Cliente con ID {idcliente} eliminado.")

    cursor.close()

def crear\_producto(conexion):

    cursor = conexion.cursor()

    idproducto = int(input("Introduce el ID del nuevo producto: "))

    nombre = input("Introduce el nombre del nuevo producto: ")

    precio = float(input("Introduce el precio del producto: "))

    idcategoria = int(input("Introduce el ID de la categoría del producto: "))

    consulta = """INSERT INTO producto (idproducto, nombre, precio, idcategoria) VALUES (%s, %s, %s, %s)"""

    cursor.execute(consulta, (idproducto, nombre, precio, idcategoria))

    conexion.commit()

    print("Producto {nombre} creado con éxito.")

    cursor.close()

def leer\_producto(conexion):

    cursor = conexion.cursor()

    consulta = "SELECT \* FROM producto";

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    print("\n--- Productos ---")

    for fila in resultados:

        print(f"ID: {fila[0]}, Nombre: {fila[1]}, Precio: {fila[2]}, Categoría ID: {fila[3]}")

    cursor.close()

def actualizar\_producto(conexion):

    cursor = conexion.cursor()

    idproducto = int(input("Introduce el ID del producto a actualizar:"))

    nuevo\_nombre = input("Introduce el nuevo nombre del producto: ")

    nuevo\_precio = float(input("Introduce el nuevo precio del producto:"))

    consulta = """UPDATE producto SET nombre = %s, precio = %s WHERE idproducto = %s"""

    cursor.execute(consulta, (nuevo\_nombre, nuevo\_precio, idproducto))

    conexion.commit()

    print("Producto con ID {idproducto} actualizado.")

    cursor.close()

def eliminar\_producto(conexion):

    cursor = conexion.cursor()

    idproducto = int(input("Introduce el ID del producto a eliminar: "))

    consulta = """DELETE FROM producto WHERE idproducto = %s"""

    cursor.execute(consulta, (idproducto,))

    conexion.commit()

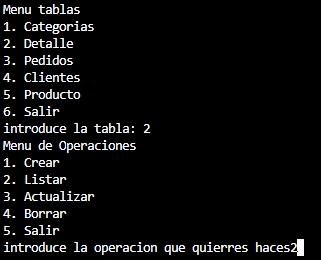
    print("Producto con ID {idproducto} eliminado.")

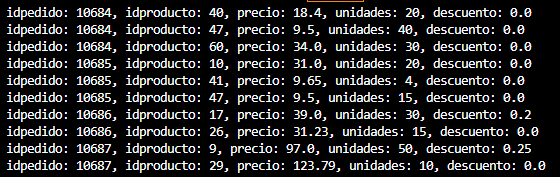
    cursor.close()

## ejemplos imágenes

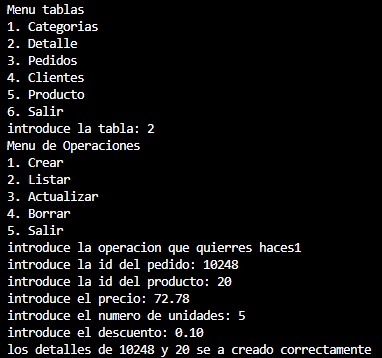
te pongo imágenes separada porque tiene tanta cosa que no carga lo primero metido

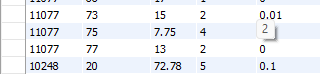
listar:



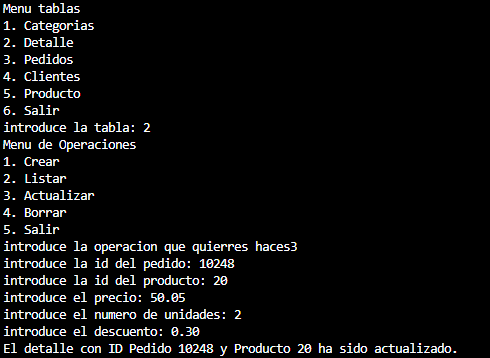


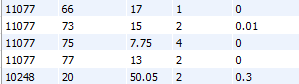
Crear:



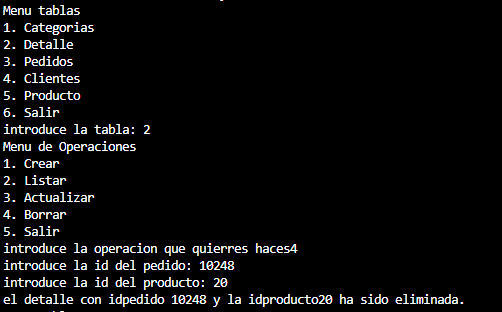


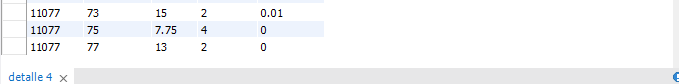
Actualizar:





Eliminar:





## Pedidos

Listar: