**Ejercicios Python - 15 - CRUD Básico SUPERMERCADO**

## **Ejercicios de Python**

### **Ejercicio: CRUD Básico sobre la Tabla categoria**

El objetivo de este ejercicio es desarrollar una aplicación en Python que permita realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la tabla categoria de la base de datos SUPERMERCADO. La aplicación debe funcionar a través de un **menú de opciones**, donde el usuario puede seleccionar la operación deseada. Este menú estará en la parte superior y siempre indicará que estamos trabajando con la tabla categoria.

#### **Funcionalidades del Menú**

1. **Crear**: Insertar un nuevo registro en la tabla categoria.
   * Solicita al usuario el idcategoria y el nombre de la categoría.
   * Inserta esta información en la base de datos.
2. **Leer**: Listar todos los registros de la tabla categoria.
   * Muestra un listado con todas las categorías disponibles en la tabla.
3. **Actualizar**: Modificar un registro existente en la tabla categoria.
   * Solicita al usuario el idcategoria del registro que desea modificar.
   * Permite cambiar el nombre de la categoría.
4. **Eliminar**: Eliminar un registro de la tabla categoria.
   * Solicita el idcategoria del registro que desea eliminar.
5. **Salir**: Termina la ejecución del programa.

#### **Ejemplo de Menú**

El menú de la aplicación debe aparecer en pantalla de la siguiente manera:

=== Gestión de Categorías ===

Seleccione una opción:

1. Crear nueva categoría

2. Leer categorías existentes

3. Actualizar una categoría

4. Eliminar una categoría

5. Salir

from Funciones import DOW\_categoria as dw

from colorama import Fore as fr

def Menu(cursor,conexion):

while True:

try:

print(fr.BLUE+"=== Gestión de Categorías ===\nSeleccione una opción:\n1. Crear nueva categoría\n2. Leer categorías existentes\n3. Actualizar una categoría\n4. Eliminar una categoría\n5. Salir"+fr.RESET)

opcion = int(input("> "))

match opcion:

case 1:

dw.Create(cursor,conexion)

case 2:

dw.List(cursor,conexion)

case 3:

dw.Actu(cursor,conexion)

case 4:

dw.Delete(cursor,conexion)

case 5:

print(fr.LIGHTRED\_EX+"Cerrando base de datos y saliendo del programa"+fr.RESET)

cursor.close()

conexion.close()

break

case \_:

print("Opción no valida")

except Exception as e:

print(fr.RED+f"Ocurrió un error inesperado: {e} " + fr.RESET)

except ValueError as e:

print(fr.RED+f"No has ingresado una opción valida: {e}"+ fr.RESET)

Cada opción llevará al usuario a una operación específica:

* Si elige **1**: Se pedirá el idcategoria y el nombre de la categoría para crear un nuevo registro.
* Si elige **2**: El programa listará todas las categorías en el formato ID - Nombre.
* Si elige **3**: Se solicitará el idcategoria para identificar el registro a actualizar y el nuevo nombre de la categoría.
* Si elige **4**: Se solicitará el idcategoria del registro a eliminar.
* Si elige **5**: El programa finalizará.

### **Ejemplos de Ejecución**

**Crear Nueva Categoría**  
Ingrese el ID de la nueva categoría: 9

Ingrese el nombre de la nueva categoría: Cereales

[Mensaje de confirmación] La categoría 'Cereales' ha sido creada con éxito.

**Leer Categorías Existentes**python  
Copiar código  
Listado de Categorías:

1 - Bebidas

2 - Condimentos

3 - Repostería

4 - Lácteos

...

**Actualizar una Categoría**  
Ingrese el ID de la categoría a actualizar: 3

Ingrese el nuevo nombre de la categoría: Postres

[Mensaje de confirmación] La categoría con ID 3 ha sido actualizada a 'Postres'.

**Eliminar una Categoría**  
Ingrese el ID de la categoría a eliminar: 8

[Mensaje de confirmación] La categoría con ID 8 ha sido eliminada.

**Salir**  
[Mensaje] Saliendo de la gestión de categorías. ¡Hasta pronto!

### **Puntos Clave del Ejercicio**

* **Entrada y Validación de Datos**: Asegúrate de que el idcategoria sea único al crear y que exista cuando se actualice o elimine.
* **Control de Errores**: Maneja posibles errores, como idcategoria no existente o intentos de duplicación.
* **Interacción con la Base de Datos**: Usa sentencias SQL de INSERT, SELECT, UPDATE y DELETE según cada operación.
* **Confirmación de Acciones**: Proporciona mensajes claros tras cada operación para que el usuario sepa si fue exitosa.

from colorama import Fore as fr

def Create(cursor, conexion):

try:

idcategoria = int(input(fr.LIGHTCYAN\_EX + "Ingrese el ID de la nueva categoría: " + fr.RESET))

categoria = input(fr.LIGHTCYAN\_EX + "Ingrese el nombre de la nueva categoría: " + fr.RESET).strip()

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

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

conexion.commit()

print(fr.GREEN + f"La categoría '{categoria}' ha sido creada con éxito." + fr.RESET)

print(fr.CYAN + "Listado de Categorías:" + fr.RESET)

consulta = "SELECT \* FROM categoria ORDER BY idcategoria;"

cursor.execute(consulta)

resultados = cursor.fetchall()

for idcategoria, categoria in resultados:

print(fr.YELLOW + f"ID: {idcategoria}, Categoria: {categoria}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al crear una nueva categoría: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def List(cursor, conexion):

try:

consulta = "SELECT \* FROM categoria;"

cursor.execute(consulta)

print(fr.CYAN + "Listado de Categorías:" + fr.RESET)

resultados = cursor.fetchall()

for idcategoria, categoria in resultados:

print(fr.YELLOW + f"ID: {idcategoria}, Categoria: {categoria}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al listar categorías: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def Actu(cursor, conexion):

try:

idcategoria = int(input(fr.LIGHTCYAN\_EX + "Ingrese el ID de la categoría que desea actualizar: " + fr.RESET))

categoria = input(fr.LIGHTCYAN\_EX + "Ingrese el nuevo nombre de la categoría: " + fr.RESET)

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

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

conexion.commit()

print(fr.GREEN + f"La categoría '{categoria}' ha sido actualizada con éxito." + fr.RESET)

print(fr.CYAN + "Listado de Categorías Actualizado:" + fr.RESET)

consulta = "SELECT \* FROM categoria ORDER BY idcategoria;"

cursor.execute(consulta)

resultados = cursor.fetchall()

for idcategoria, categoria in resultados:

print(fr.YELLOW + f"ID: {idcategoria}, Categoria: {categoria}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al actualizar una categoría: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def Delete(cursor, conexion):

try:

idcategoria = int(input(fr.LIGHTCYAN\_EX + "Ingrese el ID de la categoría que desea eliminar: " + fr.RESET))

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

cursor.execute(consulta, (idcategoria,))

conexion.commit()

print(fr.GREEN + f"La categoría con ID '{idcategoria}' ha sido eliminada con éxito." + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al eliminar una categoría: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)