Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Iniciación con Python

Clase 15 - “SQLite”

Buenos Aires Ciudad

¡Les damos la bienvenida!

Vamos a comenzar a grabar la clase

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Buenos Aires Ciudad

Talento Tech

Clase 14.

Bases de Datos

1. Instalación y uso del módulo SQLite
2. Consultas SQL básicas desde Python con SQLite
3. Consultas SQL básicas: SELECT, INSERT, UPDATE, DELETE

Clase 15.

SQLite

1. Definimos y creamos la base de datos.
2. Agregamos a las funciones creadas antes de las consultas de SQL.
3. Revisión del enunciado del TFI.

Clase 16.

Proyecto Final

1. Módulo Colorama.
2. Retroalimentación general del curso y de los proyectos.
3. Conclusiones y cierre final.

Pero antes...

¡Resolvamos los “**Ejercicios prácticos**” de la clase anterior!

SQLite

SQLite y TFI

Llegamos a la última clase antes de la entrega del TFI. Hoy nos proponemos repasar conceptos vistos y delinear las funciones necesarias para el Trabajo Final Integrador.

Utilizaremos una base de datos de ejemplo, llamada **Personas,** para ilustrar conceptos similares a los que aplicarán en el TFI y que servirán de ejemplos prácticos para que adaptes estas funciones a tu proyecto.

Creación de la base de datos

Antes de comenzar, debemos crear la base de datos y la tabla que usaremos para almacenar la información. Esto debe hacerse por única vez:

```
import sqlite3
def crear_tabla_personas():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS Personas (
            nombre TEXT, edad INTEGER, ciudad TEXT) ''')
    conexion.commit()
    conexion.close()
crear_tabla_personas()
```

Función registrar_persona()

Esta función permite agregar una persona a la base de datos. Solicitamos los datos al usuario y los guardamos con una consulta INSERT:

```
def registrar_persona():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    nombre = input("Nombre: ")
    edad = int(input("Edad: "))
    ciudad = input("Ciudad: ")
    cursor.execute("INSERT INTO Personas
        (nombre, edad, ciudad) VALUES (?, ?, ?)",
        (nombre, edad, ciudad)
        )
    conexion.commit()
    conexion.close()
```

Función mostrar_personas()

```python
def mostrar_personas():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    cursor.execute("SELECT * FROM Personas")
    resultados = cursor.fetchall()
    for registro in resultados:
        print("Nombre:",
              registro[0], "Edad:",
              registro[1], "Ciudad:",
              registro[2])
    conexion.close()
```

Con esta función, podemos visualizar todos los registros de la tabla.
Esto es similar a la función para listar productos que necesitamos implementar en el TFI.

Función actualizar_persona()

La función actualizar_persona() permite modificar la información de una persona. Pedimos el nombre y la nueva edad, y actualizamos el registro con UPDATE.

```
def actualizar_persona():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    nombre = input("Nombre de la persona: ")
    nueva_edad = int(input("Nueva edad: "))
    cursor.execute("UPDATE Personas SET edad = ? WHERE nombre = ?",
                   (nueva_edad, nombre)
                  )
    conexion.commit()
    conexion.close()
```

Función eliminar_persona()

```
def eliminar_persona():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    nombre = input("Nombre de la persona a eliminar: ")

    cursor.execute("DELETE FROM Personas WHERE nombre = ?",
                   (nombre,)
                   )
    conexion.commit()
    conexion.close()
```

Con esta función, eliminamos un registro específico, indispensable para dar de baja aquellos datos que ya no necesitamos. Solicitamos el nombre de la persona a borrar y ejecutamos una consulta DELETE.

Función buscar_persona()

Con esta función podremos consultar los datos almacenados en un registro específico. En nuestro ejemplo, recuperamos los datos de una persona usando su nombre.

```
def buscar_persona():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    nombre = input("Nombre de la persona: ")
    cursor.execute("SELECT * FROM Personas WHERE nombre = ?",
                   (nombre,))
    resultado = cursor.fetchone()
    print("Nombre:", resultado[0],
          "Edad:", resultado[1],
          "Ciudad:", resultado[2])
    conexion.close()
```

Función reporte_menores_edad()

```
def reporte_menores_edad():
    conexion = sqlite3.connect("base_datos.db")
    cursor = conexion.cursor()
    cursor.execute("SELECT * FROM Personas WHERE edad < 18")
    resultados = cursor.fetchall()
    for registro in resultados:
        print("Nombre:", registro[0],
              "Edad:", registro[1],
              "Ciudad:", registro[2])
    conexion.close()
```

Ahora, aplicaremos la cláusula WHERE para “filtrar” los registros utilizando un criterio que permite seleccionar solamente aquellos en los que el valor del campo “edad” sea menor a 18.

Menú principal

Es necesario implementar un menú que nos permita utilizar todas las funciones desarrolladas. Puede tener una estructura como la siguiente:

```
def mostrar_menu():
    print("\nMenú:")
    print("1. Registrar persona")
    print("2. Mostrar personas")

    # Continúa con las demás opciones
```

Llamada al menú principal

Al final del archivo, deberás incluir la llamada a la función que muestra el menú (por ejemplo, *mostrar_menu()*) , para lanzar el menú interactivo del programa.
Recordá que las funciones deben estar definidas antes de ser invocadas, por lo que esta llamada probablemente estará al final de tu script Python.

```
# Iniciar el programa llamando al menú principal
mostrar_menu()
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Proyecto Final Integrador

Clase N° 15

Buenos Aires Ciudad

Proyecto Final Integrador

Obligatorio | Entregable

Estamos en la etapa final del curso y es hora de poner en práctica todo lo que hemos aprendido en el **Proyecto Final Integrador (PFI)**.

El objetivo es **crear una aplicación en Python capaz de gestionar el inventario de una pequeña tienda**. Esta aplicación funcionará desde la terminal **y hará uso de una base de datos**, permitiendo que los datos se guarden de forma permanente.

Formato de Entrega:

Compartir un link al drive (público) que contenga los archivos y carpetas que conforman tu proyecto. Los links deberán ser entregados en el apartado de “Entrega de Proyecto” en el Campus Virtual.

Funcionalidad básica de la aplicación

Tu aplicación le debe permitir a la persona que lo utiliza efectuar:

- **Registro de productos:** Ingresar nuevos productos al inventario, solicitando nombre, descripción, cantidad disponible, precio y categoría.
- **Consulta de productos:** Consultar el inventario y ver la información detallada de cada producto, como stock disponible y precio.
- **Actualización de productos:** Modificar la cantidad disponible de un producto específico.
- **Eliminación de productos:** Permitir eliminar productos del inventario.
- **Listado Completo:** Generar un listado completo del inventario.
- **Reporte de Bajo Stock:** Mostrar un reporte de productos con bajo stock.

Entrega de proyecto

Recuerda:

Esta instancia de entrega es obligatoria.

Recuerda seguir las buenas prácticas de codificación que hemos discutido en clase y utilizar bucles, listas y condicionales de manera eficiente.

No dudes en consultar a tu instructor o instructora sobre cualquier aspecto del trabajo que te genere dudas.

Tendrás 7 días de corrido para realizar la entrega