

IBV
100

escuela
británica de
artes creativas
y tecnología

Buenas prácticas



Clase 01

Buenas prácticas

Consultas entre tablas pt I

A continuación te compartimos algunas buenas prácticas con python:

Importar las librerías necesarias para trabajar con archivos CSV y bases de datos SQLite

- import os
- import csv
- import sqlite3

Establecer una conexión con la base de datos SQLite

- conn = sqlite3.connect('clientes_varios.db')
- cursor = conn.cursor()

Crear una función para cargar datos desde un archivo CSV a una tabla en la base de datos

- def cargar_datos_desde_csv(archivo_csv, tabla, campos):
- with open(archivo_csv, 'r') as file:
- dict_reader = csv.DictReader(file)
- to_db = [(i[campo] for campo in campos) for i in dict_reader]

Clase 01

Buenas prácticas

Consultas entre tablas pt I

- `cursor.execute(f'CREATE TABLE {tabla} ({", ".join(campos)});')`
- `cursor.executemany(f'INSERT INTO {tabla} ({", ".join(campos)}) VALUES ({", ".join(["?" for _ in campos])});', to_db)`
- `conn.commit()`

Ejemplo de cómo cargar datos desde un archivo CSV a una tabla

```
campos_customers1 = ['customer_name', 'address', 'city', 'country']
cargar_datos_desde_csv('Customers1.csv', 'Customers1', campos_customers1)
```

Realizar una consulta utilizando INNER JOIN para combinar datos de dos tablas

```
cursor.execute("""
SELECT P1.customer_name, P1.address, P2.contact_name
FROM Customers1 AS P1
INNER JOIN Customers2 AS P2
ON P1.customer_name = P2.customer_name;
""")
```

Clase 01

Buenas prácticas

Consultas entre tablas pt I

```
# Obtener y procesar los resultados de la consulta
resultados = cursor.fetchall()
for resultado in resultados:
    print(resultado)
```

```
# Cerrar la conexión con la base de datos
conn.close()
'''
```

Explicación del código:

- Se importan las librerías ``os``, ``csv`` y ``sqlite3`` necesarias para trabajar con archivos y bases de datos.
- Se establece una conexión con la base de datos SQLite y se crea un cursor para ejecutar consultas.
- Se define una función ``cargar_datos_desde_csv`` que toma como argumentos el nombre del archivo CSV, el nombre de la tabla y los campos de la tabla. Esta función lee el archivo CSV, crea la tabla en la base de datos y carga los datos.
- Se llama a la función ``cargar_datos_desde_csv`` para cargar los datos del archivo ``Customers1.csv`` en la tabla ``Customers1``.
- Se ejecuta una consulta SQL utilizando ``INNER JOIN`` para combinar datos de las tablas ``Customers1`` y ``Customers2`` basándose en el nombre del cliente.
- Se obtienen los resultados de la consulta y se imprimen.

Clase 01

Buenas prácticas

Consultas entre tablas pt I

```
# Obtener y procesar los resultados de la consulta
resultados = cursor.fetchall()
for resultado in resultados:
    print(resultado)
```

```
# Cerrar la conexión con la base de datos
conn.close()
'''
```

Explicación del código:

- Se importan las librerías ``os``, ``csv`` y ``sqlite3`` necesarias para trabajar con archivos y bases de datos.
- Se establece una conexión con la base de datos SQLite y se crea un cursor para ejecutar consultas.
- Se define una función ``cargar_datos_desde_csv`` que toma como argumentos el nombre del archivo CSV, el nombre de la tabla y los campos de la tabla. Esta función lee el archivo CSV, crea la tabla en la base de datos y carga los datos.

Clase 01

Buenas prácticas

Consultas entre tablas pt I

- Se llama a la función ``cargar_datos_desde_csv`` para cargar los datos del archivo ``Customers1.csv`` en la tabla ``Customers1``.
- Se ejecuta una consulta SQL utilizando ``INNER JOIN`` para combinar datos de las tablas ``Customers1`` y ``Customers2`` basándose en el nombre del cliente.
- Se obtienen los resultados de la consulta y se imprimen.
- Finalmente, se cierra la conexión con la base de datos.

Clase 01

Buenas prácticas

Consultas entre tablas pt I

- Utilizar funciones para modularizar el código y hacerlo más legible y reutilizable.
- Cerrar siempre la conexión con la base de datos para liberar recursos.
- Utilizar ``INNER JOIN`` para combinar registros que tienen valores coincidentes en ambas tablas, lo cual es útil para relacionar datos de diferentes fuentes.
- Emplear alias en las tablas para simplificar las consultas y mejorar la legibilidad del código SQL.

Ejemplos de utilización en el mercado laboral:

- En el sector de ventas, se pueden combinar datos de clientes y pedidos para analizar el comportamiento de compra y mejorar las estrategias de marketing.
- En el ámbito financiero, se pueden unir datos de cuentas y transacciones para detectar patrones de fraude o para realizar análisis de riesgo crediticio.
- En la gestión de recursos humanos, se pueden integrar datos de empleados y proyectos para optimizar la asignación de recursos y la planificación de la fuerza laboral.

Clase 01

Buenas prácticas

Consultas entre tablas pt I

- Utilizar funciones para modularizar el código y hacerlo más legible y reutilizable.
- Cerrar siempre la conexión con la base de datos para liberar recursos.
- Utilizar ``INNER JOIN`` para combinar registros que tienen valores coincidentes en ambas tablas, lo cual es útil para relacionar datos de diferentes fuentes.
- Emplear alias en las tablas para simplificar las consultas y mejorar la legibilidad del código SQL.

Ejemplos de utilización en el mercado laboral:

- En el sector de ventas, se pueden combinar datos de clientes y pedidos para analizar el comportamiento de compra y mejorar las estrategias de marketing.
- En el ámbito financiero, se pueden unir datos de cuentas y transacciones para detectar patrones de fraude o para realizar análisis de riesgo crediticio.
- En la gestión de recursos humanos, se pueden integrar datos de empleados y proyectos para optimizar la asignación de recursos y la planificación de la fuerza laboral.

Clase 02

Buenas prácticas

Consultas entre tablas pt II

A continuación te compartimos algunas buenas prácticas, que deben ser aplicadas en el contexto de las habilidades ya adquiridas en módulos anteriores, como la programación en Python, el manejo de estructuras de datos, y la visualización de datos, entre otros:

****Entender la lógica de las operaciones de unión**:** Es crucial comprender cómo funcionan las operaciones ``UNION``, ``INTERSECT``, ``EXCEPT``, ``Semi Join`` y ``Anti Join`` para poder aplicarlas correctamente y obtener los resultados deseados. Esto implica saber cuándo se incluyen o excluyen duplicados y cómo se relacionan las tablas entre sí.

2. ****Uso de subconsultas para operaciones avanzadas**:** Las subconsultas son una herramienta poderosa que permite realizar operaciones más complejas como filtrar registros basados en estadísticas de otra tabla (por ejemplo, seleccionar clientes con montos de compra menores al promedio general).

3. ****Aplicación en el mercado laboral**:** Estas habilidades son aplicables en múltiples contextos del mercado laboral, como en la gestión de bases de datos de clientes, análisis financiero, o cualquier área donde se requiera la manipulación y análisis de grandes conjuntos de datos relacionales.

Clase 02

Buenas prácticas

Consultas entre tablas pt II

```
# Conectar a la base de datos SQLite
conn = sqlite3.connect('mi_base_de_datos.db')
cursor = conn.cursor()

# Ejecutar una consulta SELECT con CASE para clasificar ventas
query = """
SELECT customer_id,
       CASE
         WHEN total_purchase < 100 THEN 'baja'
         WHEN total_purchase BETWEEN 100 AND 500 THEN 'media'
         ELSE 'alta'
       END as purchase_category
FROM customers2;
"""

cursor.execute(query)
resultados = cursor.fetchall()
for row in resultados:
    print(row)

# Cerrar la conexión
conn.close()
"""
```

Clase 02

Buenas prácticas

Consultas entre tablas pt II

Este código conecta a una base de datos SQLite, ejecuta una consulta que clasifica las ventas en categorías según el monto de la compra y luego cierra la conexión. Cada línea del código tiene un propósito específico:

- Importar el módulo ``sqlite3`` para trabajar con bases de datos SQLite.
- Conectar a la base de datos existente `'mi_base_de_datos.db'`.
- Crear un objeto cursor para ejecutar consultas.
- Definir y ejecutar una consulta SQL que utiliza ``CASE`` para clasificar las ventas.
- Recuperar y mostrar los resultados de la consulta.
- Cerrar la conexión con la base de datos para liberar recursos.

¡Mucho éxito en tus estudios!