Conectar Python con Mysql

Abrir la consola

Desde Windows cmd

pip install mysql-connector-python

Enter

python -c “import mysql.connector; print(mysql.connector.\_\_version\_\_)”

Enter

Workbench

CREATE DATABASE IF NOT EXISTS perfil\_morbilidad;

USE perfil\_morbilidad;

Click izquierdo en perfil\_morbilidad table data import escoger el archivo csv

Refrescar cuando cargue el archivo

Ingresar a Visual studio

database.py

escribir el código que tengo en notas del celular.

Import. Esto es una biblioteca que conecta Python con sql

Host=

User=root

Pasword= se escribe el pasword de sql

Database= se escribe el nombre de la base de datos como está en mysql

Select \* from morbilidad trae los primeros 10

El resultado lo coloca en una tupla

import mysql.connector

def obtener\_datos():

    try:

        # Establecer conexión con la base de datos

        conn = mysql.connector.connect(

            host="localhost",

            user="root",

            password="0919",

            database="perfil\_morbilidad"

        )

        # Crear cursor para ejecutar consultas

        cursor = conn.cursor()

        # Ejecutar consulta

        consulta = "SELECT \* FROM perfil\_morbilidad.perfil\_de\_morbilidad\_20250423 LIMIT 10;"

        cursor.execute(consulta)

        # Obtener resultados

        resultados = cursor.fetchall()

        for fila in resultados:

            print(fila)

    except mysql.connector.Error as err:

        print(f"Error: {err}")

    finally:

        # Cerrar cursor y conexión si están abiertos

        if 'cursor' in locals() and cursor:

            cursor.close()

        if 'conn' in locals() and conn:

            conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

    obtener\_datos()

**Mejoras aplicadas:**

* Encapsulado el código en una función (obtener\_datos) para mayor modularidad y reutilización.
* Añadido if \_\_name\_\_ == "\_\_main\_\_" para permitir que el script sea importado sin ejecutarse automáticamente.
* Agregado control de existencia de variables (locals()) en el bloque finally para evitar errores si falló la conexión.

LIMPIEZA DE DATOS CON PANDAS

import mysql.connector

import pandas as pd

# Paso 1: Conectar a la base de datos MySQL

conn = mysql.connector.connect(

    host="localhost",  # Dirección del servidor de base de datos (localhost si es local)

    user="root",  # Usuario de MySQL

    password="0919",  # Contraseña del usuario MySQL

    database="perfil\_morbilidad",  # Nombre de la base de datos

)

# Paso 2: Crear un cursor para ejecutar las consultas SQL

cursor = conn.cursor()

# Paso 3: Ejecutar una consulta SQL para obtener los primeros 10 registros de la tabla 'morbilidad'

cursor.execute(

    "SELECT \* FROM perfil\_morbilidad.perfil\_de\_morbilidad\_20250423 LIMIT 10;"

)

# Paso 4: Obtener los resultados de la consulta

resultados = cursor.fetchall()

# Paso 5: Crear un DataFrame de pandas con los resultados

columnas = [

    "CIDIGO\_CIE10",

    "NOMBRE\_DEL\_DIAGNOSTICO",

    "UNIDAD\_FUNCIONAL",

    "DESTINO\_AL\_EGRESO",

    "EDAD\_DE\_ATENCION\_AÑOS",

    "AÑO\_REPORTADO",

]

df = pd.DataFrame(resultados, columns=columnas)

# Eliminar filas duplicadas

df = df.drop\_duplicates()

# Eliminar Filas con valores nulos

df = df.dropna()

# Convertir la columna  de edad a un tipo de datos adecuado (int)

df["EDAD\_ATENCION\_AÑOS"] = df["EDAD\_DE\_ATENCION\_AÑOS"].astype(int)

# Paso 6: Mostrar los primeros 10 registros

print(df.head())

# Mostrar los primeros 10 registros Limpios

print("Primeros 10 registros despues de limpiar los datos")

# Paso 7: Cerrar la conexión a la base de datos

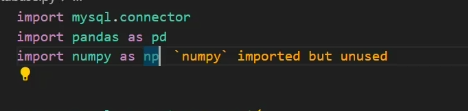
cursor.close()

conn.close()

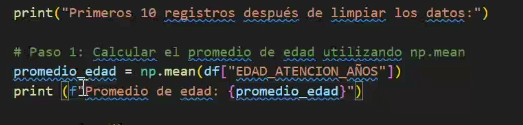
**Alt shift f arregla identación y sintaxis**

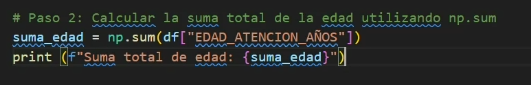
**Instalar numpy**

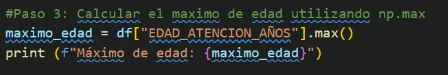
pip install numpy

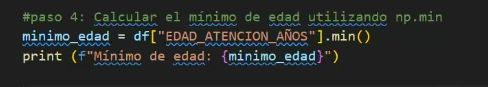


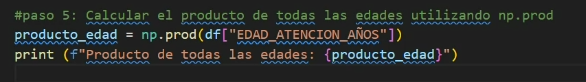
Import numpy as np



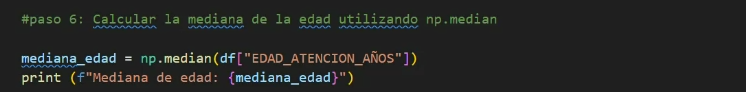




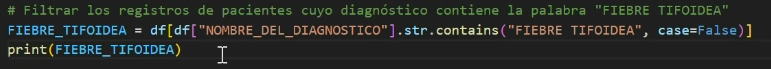




. prod Multiplica todos los valores de la matriz



#paso 6: calcular la mediana de la edad utilizando np.median



Alt shift f

# Paso 1: Calcular el promedio de edad utilizando np.mean

promedio\_edad = np.mean(df["EDAD\_ATENCION\_AÑOS"])

print (f"Promedio de edad: {promedio\_edad}")

# Paso 2: Calcular la suma total de la edad utilizando np.sum

suma\_edad = np.sum(df["EDAD\_ATENCION\_AÑOS"])

print (f"Suma total de edad: {suma\_edad}")

#Paso 3: Calcular el maximo de edad utilizando np.max

maximo\_edad = df["EDAD\_ATENCION\_AÑOS"].max()

print (f"Máximo de edad: {maximo\_edad}")

#paso 4: Calcular el mínimo de edad utilizando np.min

minimo\_edad = df["EDAD\_ATENCION\_AÑOS"].min()

print (f"Mínimo de edad: {minimo\_edad}")

#paso 5: Calcular el producto de todas las edades utilizando np.prod

producto\_edad = np.prod(df["EDAD\_ATENCION\_AÑOS"])

print (f"Producto de todas las edades: {producto\_edad}")

# paso 6: Calcular la mediana de la edad utilizando np.median

mediana\_edad = np.median(df["EDAD\_ATENCION\_AÑOS"])

print(f"Mediana de edad: {mediana\_edad}")

# Filtrar los registros de pacientes cuyo diagnóstico contiene la palabra "FIEBRE TIFOIDEA"

FIEBRE\_TIFOIDEA = df[

    df["NOMBRE\_DEL\_DIAGNOSTICO"].str.contains("FIEBRE TIFOIDEA", case=False)

]

print(FIEBRE\_TIFOIDEA)