



UNIVERSIDAD DE SONORA
DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES
MAESTRÍA EN CIENCIA DE DATOS



Proyecto Final de Introducción a la Ciencia de Datos y sus Metodologías:
MySQL y Python.

Matemáticas para Ciencia de Datos

Profesor: Doctor Juan Pablo Soto Barrera

Alumno: Jordan Joel Urias Paramo

Hermosillo, Sonora a 25 de noviembre del 2022

Contenido

1. Descripción.....	3
2. Objetivos	3
3. Descripción de fuente de datos.....	3
4. Manejo de datos por medio de MySQL	3
4.1 Creación de base de datos	3
4.2 Extracción de información de la base de datos desde Python	6
4.3 Creación de una vista desde Python	9
4.4 Creación de una función desde la consola.....	10
4.5 Creación de un procedimiento desde la consola.....	11
5. Exportando la base de datos.	12
6. Referencias	13

1. Descripción

Hoy en día vivimos en un mundo donde obtener datos no es un problema, pero su manejo para el análisis lo es; por ello es necesario comprender y saber usar los manejadores de bases de datos, estos funcionan a base de algoritmos optimizados para recorrer y realizar estadística descriptiva. Además, las herramientas que usamos como científicos de datos tienen la facilidad de incluir conectores que nos permiten interactuar directamente con los agentes que manejan las bases de datos

2. Objetivos

- Crear una base de datos por medio de la línea de comandos.
- Establecer una conexión de la base de datos por medio de un conector para Python.
- Efectuar una consulta de la base de datos a través de Jupyter Notebook.
- Crear una vista, un procedimiento almacenado y una función en una base de datos.

3. Descripción de fuente de datos

En esta ocasión no quiero que los datos sean la estrella del proyecto, por ello usaremos datos usados por el Profesor Julio Waissman y el usuario de GitHub DogoMoreno. Estos datos son datos genéricos usados para la creación de mapas relacionados con aspectos de salud, en este caso guardaremos en la base datos dos tablas, una con los municipios de Sonora que incluyen las coordenadas y otra que incluye la población, jurisdicción sanitaria y coordinación médica.

4. Manejo de datos por medio de MySQL

4.1 Creación de base de datos

Primeramente, creamos un conector y mediante este llamamos a un curso que ejecutara el comando para crear una base de datos llamada sonora

```
try:
    conn = mysql.connect(host='localhost', user='root', password= 'root' ) #Establecer conexion
    if conn.is_connected():
        cursor = conn.cursor()#Crear cursor
        cursor.execute("CREATE DATABASE sonora") #Ejecutar instruccion
        print("La base de datos sonora fue creada")
        #Normalmente aqui se cierra el conector pero lo usaremos mas adelante en la libreta
except Error as e:
    print("Error al conectarse: ", e)
```

Jalamos los datos de la primera tabla, con las coordenadas de los municipios

```
df = pd.read_csv('https://raw.githubusercontent.com/mcd-unison/ing-caract/main/ejemplos/mapas/python/municipios_latlon.csv')
df.head()
```

	MUNICIPIO	LATITUD	LONGITUD
0	ACONCHI	29.825420	-110.20840
1	AGUA PRIETA	31.330048	-109.54904
2	ALAMOS	27.026820	-108.92560
3	ALTAR	30.715070	-111.82135
4	ARIVECHI	29.078530	-109.19583

Una vez que la base de datos esta creada y tenemos los datos podemos crear las tablas e insertarle los datos

```
try:
    conn = mysql.connect(host='localhost',
                        database='sonora', user='root',
                        password='root')
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Estas coenctado a: ", record)
        cursor.execute('DROP TABLE IF EXISTS coordenada;')#Si existe una tabla con este nombre la borramos
        print('Creado tabla coordenada...')#Creacion de tabla con tipos y restricciones de datos
        cursor.execute(f"""CREATE TABLE coordenada (municipio CHAR(100)
                                NOT NULL, latitud FLOAT(9,6) NOT NULL,
                                longitud FLOAT(9,6) NOT NULL)""")
        print("La tabla coordenada fue creada...")
        for i,row in df.iterrows():#Insercion de valores
            sql = "INSERT INTO sonora.coordenada VALUES (%s,%s,%s)"
            cursor.execute(sql, tuple(row))
            #La conexion no confirma los cambios en automatico por default, por lo que debemos correr el siguiente comando
            conn.commit()
            print('Operacion exitosa')
except Error as e:
    print("Error al conectarse: ", e)
```

Verificamos la que se realizó la transacción de información.

```
# Corremos un query
sql = "SELECT * FROM coordenada"
cursor.execute(sql)
# Jalar todos los resultados
result = cursor.fetchall()
for i in result:
    print(i)

('ACONCHI', 29.82542, -110.208397)
('AGUA PRIETA', 31.330048, -109.549042)
('ALAMOS', 27.026819, -108.925598)
('ALTAR', 30.715071, -111.82135)
('ARIVECHI', 29.078529, -109.195831)
('ARIZPE', 30.40764, -110.164749)
('ATIL', 30.884211, -111.582779)
('BACADEHUACHI', 29.81189, -109.140793)
('BACANORA', 28.98597, -109.400398)
('BACERAC', 30.50449, -108.924011)
('BACOACHI', 30.633801, -109.968391)
('BACUM', 27.55286, -110.0821)
('BANAMICHI', 30.00947, -110.213371)
('BAVIACORA', 29.71299, -110.161568)
```

Replicamos el proceso para la base de datos con información sobre los municipios.

```
df = pd.read_csv('https://raw.githubusercontent.com/mcd-unison/ing-caract/main/ejemplos/mapas/python/POBLACION_SONORA2020.csv')
df.head()
```

	JURISDICCION SANITARIA	COORDINACION MEDICA LOCAL	MUNICIPIO	POBLACION TOTAL
0	JURISDICCION SANITARIA I - HERMOSILLO	CML 01 HERMOSILLO	CARBO	6261
1	JURISDICCION SANITARIA I - HERMOSILLO	CML 01 HERMOSILLO	LA COLORADA	1675
2	JURISDICCION SANITARIA I - HERMOSILLO	CML 01 HERMOSILLO	HERMOSILLO	914163
3	JURISDICCION SANITARIA I - HERMOSILLO	CML 01 HERMOSILLO	MAZATAN	1403
4	JURISDICCION SANITARIA I - HERMOSILLO	CML 01 HERMOSILLO	ONAVAS	421

```
try:#Mismo proceso para creacion de tablas que se realizo antes
    conn = mysql.connect(host='localhost',
                        database='sonora', user='root',
                        password='root')
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Estas conectado a: ", record)
        cursor.execute('DROP TABLE IF EXISTS poblacion;')
        print('Creado tabla poblacion...')
        cursor.execute(f"""CREATE TABLE poblacion (
                        jurisdiccion_sanitaria CHAR(100) NOT NULL,
                        coord_medica_local CHAR(100) NOT NULL,
                        municipio CHAR(100) NOT NULL,
                        poblacion INT NOT NULL)""")
        print("La tabla coordenada fue creada...")
        for i,row in df.iterrows():
            sql = "INSERT INTO sonora.poblacion VALUES (%s,%s,%s,%s)"
            cursor.execute(sql, tuple(row))
            # the connection is not autocommited by default, so we must commit to save our change
            conn.commit()
            print('Operacion exitosa')
except Error as e:
    print("Error al conectarse: ", e)
```

```
# Execute query
sql = "SELECT * FROM poblacion"
cursor.execute(sql)
# Fetch all the records
result = cursor.fetchall()
for i in result:
    print(i)
```

```
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'CARBO', 6261)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'LA COLORADA', 1675)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'HERMOSILLO', 914163)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'MAZATAN', 1403)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'ONAVAS', 421)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'SAN JAVIER', 745)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'SAN MIGUEL DE HORCASITAS', 10939)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'SAN PEDRO DE LA CUEVA', 1815)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'SOYOPA', 1373)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'SUAQUI GRANDE', 1171)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 01 HERMOSILLO', 'VILLA PESQUEIRA', 1242)
('JURISDICCION SANITARIA I - HERMOSILLO', 'CML 02 URES', 'ACONCHI', 2928)
```

Por último, como se mencionó en el código se cierra la conexión al terminar la operación de creación en inserción de datos

```
# Close the connection
if (conn.is_connected()):
    cursor.close()
    conn.close()
print("MySQL cerrada")
```

4.2 Extracción de información de la base de datos desde Python

Para trabajar con los datos desde Python es necesario extraerlos y convertir los datos en un formato que facilite su manejo, es este caso usaremos la herramienta que se usa en manejar tablas de información en Python pandas.

```
try:
    mydb = mysql.connect(host="localhost", database = 'sonora', user="root", passwd="root", use_pure=True)
    query_pob = "Select * from poblacion;"
    municipios_poblacion = pd.read_sql(query_pob, mydb)
    query_mun = "Select * from coordenada;"
    municipios = pd.read_sql(query_mun, mydb)
    mydb.close() #close the connection
except Exception as e:
    mydb.close()
    print(str(e))
```

Y de una vez creamos la tabla tidy que podemos usar para crear la tabla, para ello realizamos un merge, así se le llama al JOIN en pandas.

```
municipios = municipios.merge(municipios_poblacion, on="municipio")

municipios.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 72 entries, 0 to 71
Data columns (total 6 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   municipio                             72 non-null     object
1   latitud                               72 non-null     float64
2   longitud                              72 non-null     float64
3   jurisdiccion_sanitaria               72 non-null     object
4   coord_medica_local                   72 non-null     object
5   poblacion                            72 non-null     int64
dtypes: float64(2), int64(1), object(3)
memory usage: 3.9+ KB
```

De igual manera realizamos la instrucción JOIN mediante el cursor desde Python, puesto que es más optimo realizar las transformaciones a los datos desde el manejador

```
try:
    mydb = mysql.connect(host="localhost", database = 'sonora',user="root", passwd="root",use_pure=True)
    query = f"""SELECT c.municipio, c.latitud,c.longitud,
                    p.jurisdiccion_sanitaria,p.coord_medica_local, p.poblacion
                    FROM coordenada c INNER JOIN poblacion p ON p.municipio = c.municipio;"""
    municipios_join = pd.read_sql(query,mydb)
    mydb.close() #close the connection
except Exception as e:
    mydb.close()
    print(str(e))
```

```
municipios_join.info()
```

```
municipios_join = municipios_join.sort_values(by=['municipio']).reset_index(drop=True)
```

Despues de una ordenada, podemos observar que los resultados son equivalentes

```
municipios_join.head()
```

	municipio	latitud	longitud	jurisdiccion_sanitaria	coord_medica_local	poblacion
0	ACONCHI	29.825420	-110.208397	JURISDICCION SANITARIA I - HERMOSILLO	CML 02 URES	2928
1	AGUA PRIETA	31.330048	-109.549042	JURISDICCION SANITARIA III - SANTA ANA	CML 12 AGUA PRIETA	92364
2	ALAMOS	27.026819	-108.925598	JURISDICCION SANITARIA V - NAVOJOA	CML 18 ALAMOS	28461
3	ALTAR	30.715071	-111.821350	JURISDICCION SANITARIA II - CABORCA	CML 07 CABORCA	10232
4	ARIVECHI	29.078529	-109.195831	JURISDICCION SANITARIA I - HERMOSILLO	CML 03 SAHUARIPA	1304

```
municipios.head()
```

	municipio	latitud	longitud	jurisdiccion_sanitaria	coord_medica_local	poblacion
0	ACONCHI	29.825420	-110.208397	JURISDICCION SANITARIA I - HERMOSILLO	CML 02 URES	2928
1	AGUA PRIETA	31.330048	-109.549042	JURISDICCION SANITARIA III - SANTA ANA	CML 12 AGUA PRIETA	92364
2	ALAMOS	27.026819	-108.925598	JURISDICCION SANITARIA V - NAVOJOA	CML 18 ALAMOS	28461
3	ALTAR	30.715071	-111.821350	JURISDICCION SANITARIA II - CABORCA	CML 07 CABORCA	10232
4	ARIVECHI	29.078529	-109.195831	JURISDICCION SANITARIA I - HERMOSILLO	CML 03 SAHUARIPA	1304

Por último, usamos la información guardada en las base de datos para crear mapas interactivo


```

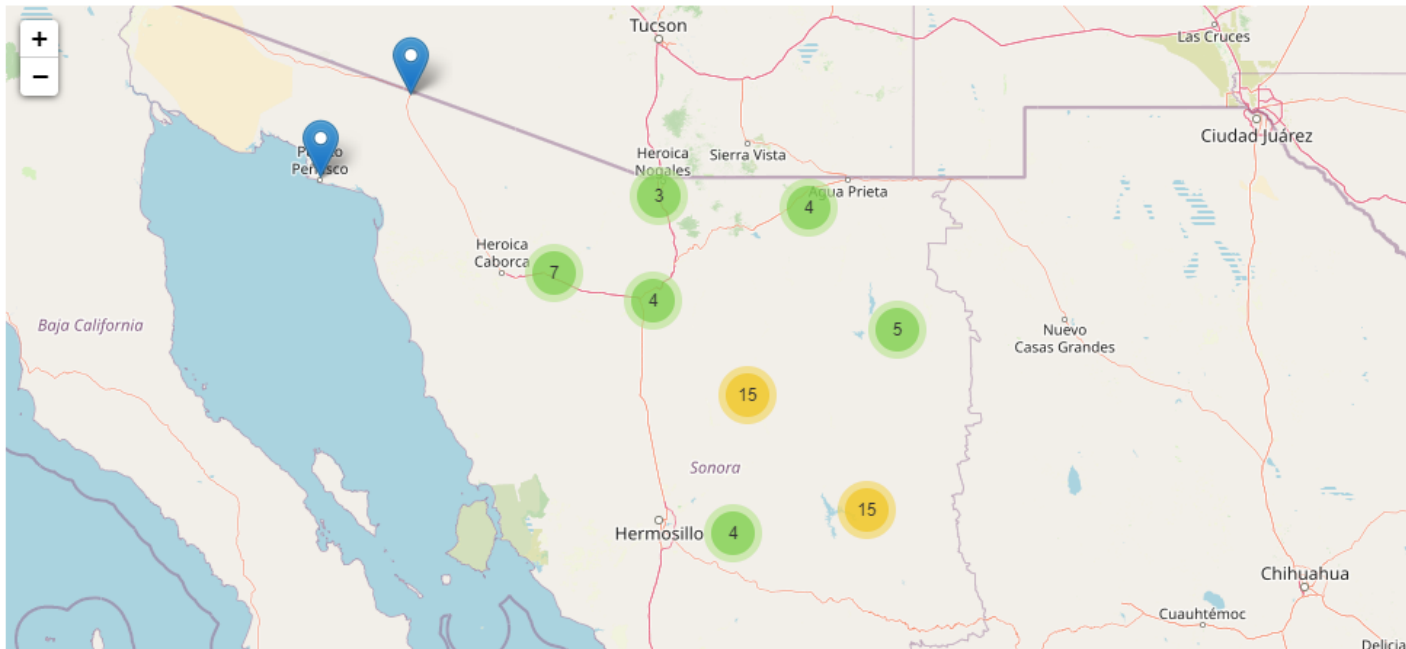
from folium.plugins import MarkerCluster

m = folium.Map(
    location=(29.66667, -110.5),
    tiles='OpenStreetMap',
    zoom_start=7,
    min_zoom=4,
    max_zoom=12,
)

grupo = MarkerCluster().add_to(m)

for row in municipios.itertuples():
    folium.Marker(
        location=[row.latitud, row.longitud],
        tooltip=f"{row.municipio} Población: {row.poblacion} habitantes",
    ).add_to(grupo)
m

```



4.3 Creación de una vista desde Python

Para la creación de una vista es necesario primero crear una consulta y después mediante el comando `CREATE VIEW [vista] AS [consulta]`, para ejemplificar esta funcionalidad usaremos la consulta que usamos anteriormente que fusiona dos tablas.

```

try:
    conn = mysql.connect(host='localhost',
                        database='sonora', user='root',
                        password='root')
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Estas conectado a: ", record)
        cursor.execute('DROP VIEW IF EXISTS vw_mapa;')
        print('Creado tabla poblacion...')
        cursor.execute(f"""CREATE VIEW vw_mapa AS
                        SELECT c.municipio, c.latitud,c.longitud,
                           p.jurisdiccion_sanitaria,p.coord_medica_local, p.poblacion
                        FROM coordenada c INNER JOIN poblacion p ON p.municipio = c.municipio
                        ORDER BY municipio
                        ;

                        """)
        print("La vista vw_mapa fue creada...")
        print('Operacion exitosa')
except Error as e:
    print("Error al conectarse: ", e)

```

Como se puede observar, la creación de la vista no es muy distinta a la de la creación de la tabla que se realizó anteriormente, se ejecutó como un comando.

```

try:
    query_pob = "Select * from vw_mapa;"
    municipios = pd.read_sql(query_pob,conn)
except Exception as e:
    print(str(e))

```

```
municipios.head()
```

	municipio	latitud	longitud	jurisdiccion_sanitaria	coord_medica_local	poblacion
0	ACONCHI	29.825420	-110.208397	JURISDICCION SANITARIA I - HERMOSILLO	CML 02 URES	2928
1	AGUA PRIETA	31.330048	-109.549042	JURISDICCION SANITARIA III - SANTA ANA	CML 12 AGUA PRIETA	92364
2	ALAMOS	27.026819	-108.925598	JURISDICCION SANITARIA V - NAVOJOA	CML 18 ALAMOS	28461
3	ALTAR	30.715071	-111.821350	JURISDICCION SANITARIA II - CABORCA	CML 07 CABORCA	10232
4	ARIVECHI	29.078529	-109.195831	JURISDICCION SANITARIA I - HERMOSILLO	CML 03 SAHUARIPA	1304

4.4 Creación de una función desde la consola.

Por fines didácticos, para mostrar rango de habilidades, nos movemos a la consola donde introducimos las siguientes líneas:

```

DROP FUNCTION IF EXISTS fn_get_poblacion_jurisdiccion
DELIMITER //
CREATE FUNCTION sonora.fn_get_poblacion_jurisdiccion(jurisdiccion CHAR(200))
RETURNS INT
DETERMINISTIC
BEGIN
    declare jur_poblacion INT;
    SELECT SUM(poblacion) INTO jur_poblacion
    FROM vw_mapa WHERE jurisdiccion_sanitaria = jurisdiccion;
    RETURN jur_poblacion;
END //

```

DELIMITER ;

La instrucción anterior es una función que nos permite obtener la población total por cada jurisdicción sanitaria. Dicha función puede ser llamada desde Python de la siguiente manera

```
try:
    conn = msql.connect(host='localhost',
                        database='sonora', user='root',
                        password='root')
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Estas coenctado a: ", record)
        query_pob_hermosillo = "Select fn_get_poblacion_jurisdiccion('JURISDICCION SANITARIA I - HERMOSILLO');"
        poblacion_hermosillo = pd.read_sql(query_pob_hermosillo,conn)
        query_pob_santaana = "Select fn_get_poblacion_jurisdiccion('JURISDICCION SANITARIA III - SANTA ANA');"
        poblacion_santaana = pd.read_sql(query_pob_santaana,conn)
        print('Operacion exitosa')
except Error as e:
    print("Error al conectarse: ", e)
```

Estas coenctado a: ('sonora',)
Operacion exitosa

poblacion_santaana

fn_get_poblacion_jurisdiccion('JURISDICCION SANITARIA III - SANTA ANA')	
0	492356

poblacion_hermosillo

fn_get_poblacion_jurisdiccion('JURISDICCION SANITARIA I - HERMOSILLO')	
0	1012683

4.5 Creación de un procedimiento desde la consola.

Por último, creamos un procedimiento mediante la consola ingresando el siguiente código

```
DELIMITER $$
CREATE PROCEDURE fn_get_poblacion_coord_medica(IN coord CHAR(200))
BEGIN
SELECT
    SUM(poblacion) AS poblacion_coord
    FROM vw_mapa WHERE coord_medica_local = coord;
END$$
DELIMITER ;
```

Para poder apreciar las diferencias entre la creación de una función y un procedimiento la operación que realiza nuestro procedimiento es análogo a la función que recupera la población en base da la jurisdicción, pero con la coordinación médica.

```

try:
    conn = mysql.connect(host='localhost',
                        database='sonora', user='root',
                        password='root')
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Estas coenctado a: ", record)
        query_pob_caborca = "CALL fn_get_poblacion_coord_medica('CML 07 CABORCA');"
        poblacion_caborca = pd.read_sql(query_pob_caborca,conn)
        print('Operacion exitosa')
except Error as e:
    print("Error al conectarse: ", e)

```

Estas coenctado a: ('sonora',)
Operacion exitosa

poblacion_caborca

SUM(poblacion)

0	122597.0
---	----------

5. Exportando la base de datos.

Por último, importamos la base de datos creada usando Python, pero más específicamente la librería del sistema operativo.

```

import os
os.system('mysql -u root -p%s sonora > sonora.sql' % 'root')
mysql: [Warning] Using a password on the command line interface can be insecure.

```

6. Referencias

- I. MySQL 8.0 Reference Manual. CREATE PROCEDURE and CREATE FUNCTION Statements
<https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html>
- II. MySQL Connector/Python Developer Guide. Web Manual.
<https://dev.mysql.com/doc/connector-python/en/>
- III. Waissman, Julio. (21 de septiembre del 2021). Repositorio Covid-19 en Sonora.
<https://github.com/juliowaissman/Covid19-Sonora-Municipios>