

## Tarea 02: Creación de tablas en Octave, PostgreSQL y Python

Alejandra Magalí López Miranda, 201600085<sup>1,\*</sup>

<sup>1</sup>*Escuela de Ingeniería Mecánica Eléctrica, Facultad de ingeniería, Universidad de San Carlos, Guatemala.*  
(Dated: 30 de julio de 2025)

### I. INTRODUCCIÓN

En el presente trabajo se explorará el proceso de creación de tablas y el uso de bases de datos en PostgreSQL, Octave y Python. Se utilizarán ambos entornos para comparar cómo se insertan, visualizan y gestionan los datos, así como las diferencias en la forma en que cada lenguaje de programación aborda estas tareas. Además, se analizarán las ventajas y limitaciones de cada uno, con el fin de comprender cuál resulta más adecuado según el tipo de aplicación o necesidad.

### II. OBJETIVOS

#### A. General

- Implementar la creación de tablas e inserción de datos en PostgreSQL, Octave y Python, utilizando ejemplos prácticos.

#### B. Específicos

- \* Comparar el uso de bases de datos en PostgreSQL, Python y Octave, mediante la creación de tablas, inserción y visualización de datos.
- \* Analizar la forma en que cada lenguaje permite visualizar y gestionar los datos almacenados.
- \* Identificar las ventajas, desventajas y posibles usos de PostgreSQL, Octave y Python en el manejo de bases de datos según distintos tipos de necesidades.

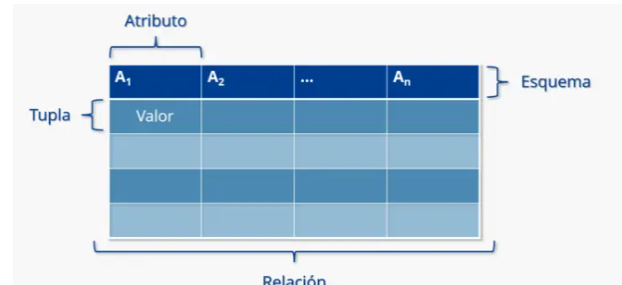
### III. MARCO TEÓRICO

#### A. Base de datos relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla

contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

Figura 1: Base de datos relacional



#### B. SQL

SQL es la abreviatura de Structured Query Language (lenguaje de consulta estructurado), un lenguaje diseñado para interactuar con bases de datos relacionales. SQL permite realizar tareas como actualizar y recuperar datos de bases de datos.

Una base de datos SQL, también conocida como base de datos relacional, es una colección estructurada de datos organizados en tablas, a las que se puede tener acceso mediante un lenguaje de consulta estructurado.

Cada fila de una tabla representa una entidad de datos y cada columna define un campo de información específico. A través de SQL se puede tener acceso a los datos de estas tablas, así como administrarlos, modificarlos, actualizarlos, controlarlos y organizarlos.

A continuación se muestra un subconjunto de sentencias SQL y para qué se utilizan:

- CREATE DATABASE - crea una instancia de base de datos para almacenar objetos como tablas e índices.
- CREATE TABLE - se utiliza para crear una tabla de datos
- INSERT - se utiliza para insertar nuevas filas en una tabla.
- UPDATE - se utiliza para realizar cambios en los datos contenidos en una tabla de base de datos.

\* 2816733570108@ingenieria.usac.edu.gt

- DELETE - se utiliza para eliminar filas de datos.
- GRANT - se utiliza para dar a los usuarios permiso para ver o modificar objetos de la base de datos como tablas.
- REVOKE - se utiliza para eliminar permisos de usuarios o grupos de usuarios.
- ALTER - permite al usuario añadir o eliminar columnas de un objeto para actualizar los permisos.
- DROP TABLE - elimina una tabla.
- CREATE INDEX - creará un INDEX para permitir un acceso eficiente a los datos utilizando una clave-valor.
- TRUNCATE TABLE - permite al usuario vaciar el contenido de una tabla.

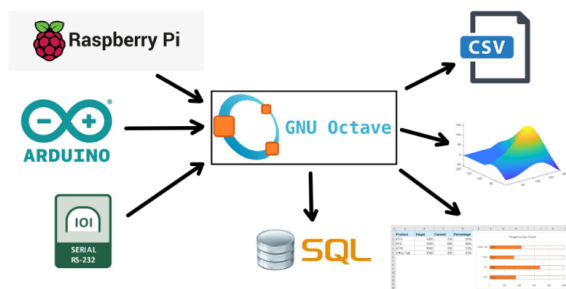
### C. Octave

Octave o GNU Octave es un programa y lenguaje de programación para realizar cálculos numéricos. Como su nombre indica, Octave es parte del proyecto GNU. Es considerado el equivalente libre de MATLAB. Entre varias características que comparten, se puede destacar que ambos ofrecen un intérprete, permitiendo ejecutar órdenes en modo interactivo.

### D. Octave Forge

Octave no tiene una función integrada para trabajar directamente con bases de datos SQL. Sin embargo, Octave Forge es una colección de paquetes que proporcionan funcionalidad adicional para GNU Octave.

Figura 2: Octave Forge

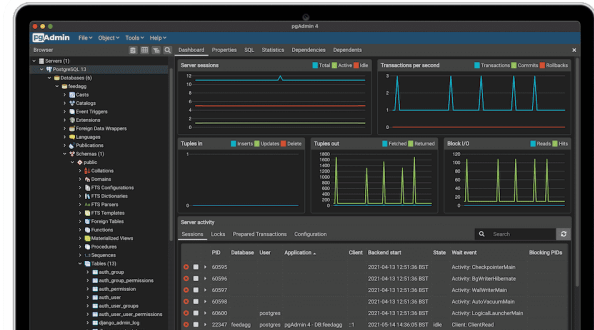


### E. PostgreSQL

Es una base de datos de código abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y apoyo a los estándares técnicos abiertos.

A diferencia de otros RDBMS (sistemas de gestión de bases de datos relacionales), PostgreSQL admite tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles en la actualidad.

Figura 3: PostgreSQL



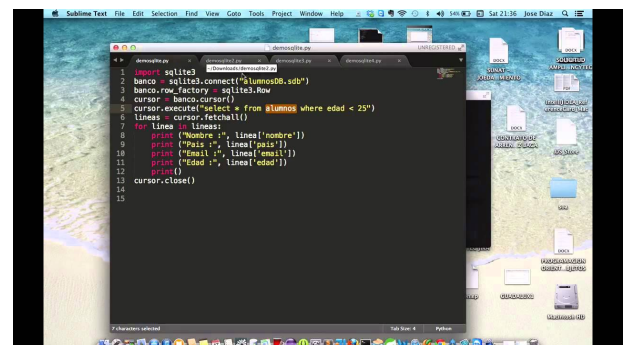
### F. Python

Python es un lenguaje de programación versátil y poderoso que se utiliza en una amplia gama de aplicaciones, incluyendo el manejo de bases de datos.

Python ofrece una gran cantidad de módulos y bibliotecas especializadas para el manejo de bases de datos, lo que lo hace fácil de usar para una amplia gama de tareas, desde la conexión y consulta de bases de datos hasta la manipulación y análisis de datos.

Una biblioteca popular para el manejo de bases de datos en Python es psycopg2. Psycopg2 es un adaptador de base de datos PostgreSQL para Python. Con psycopg2, puede conectarse a una base de datos PostgreSQL y realizar tareas como consultar y modificar datos, ejecutar sentencias SQL y controlar transacciones.

Figura 4: Python



## IV. RESULTADOS

### A. Tabla en Octave

Primero se descarga y se carga el paquete para trabajar con bases de datos en Octave.

Despues se establece conexión a la base de datos PostgreSQL local usando las credenciales y parámetros necesarios.

Se inserta un nuevo dato en la tabla 'redes'.

Por último se consulta y se visualizan los datos.

En data contiene los datos. En columns indica los nombres de las columnas. En types contiene los detalles técnicos del tipo de datos.

Figura 5: Tabla en Octave

```
GNU Octave (CLI)
octave:6> pkg load database
octave:7> conn = pg_connect(setdbopts('dbname','postgres','host','localhost',
',port','5432','user','postgres','password','koala'));
octave:8> N=pg_exec_params(conn, "INSERT INTO redes VALUES ('Carlos', 20140
0524);");
octave:9> N=pg_exec_params(conn, "SELECT * FROM redes;");
octave:10> disp(N)
data =
    {
    [1,1] = Alejandra
    [2,1] = Carlos
    [1,2] = 201600085
    [2,2] = 201400524
    }

columns =
    {
    [1,1] = nombre
    [1,2] = carnet
    }

types =

1x2 struct array containing the fields:

    name
    is_array
    is_composite
    is_enum
    elements

octave:11> |
```

### B. Tabla en PostgreSQL

Se creó una tabla llamada 'redes'. Con dos columnas:

- Nombre: almacena una cadena de hasta 20 caracteres.
- Carnet: número entero

Luego se insertó una fila con los valores 'Alejandra' '201600085'.

Por último se indica que se muestren todos los datos de la tabla 'redes'.

Figura 6: Tabla en PostgreSQL

Query	Query History
1 CREATE TABLE redes (	
2 Nombre VARCHAR(20),	
3 Carnet INT NOT NULL	
4 );	
5	
6 INSERT INTO redes VALUES ('Alejandra', 201600085);	
7	
8 SELECT * FROM redes;	

Data Output	Messages	Notifications
<div> <div>nombre</div> <div>character varying (20)</div> </div> <div> <div>carnet</div> <div>integer</div> </div>		
1 Alejandra 201600085		

### C. Tabla en Python

Se importa la librería que permite contactar y trabajar con bases de datos PostgreSQL.

Se establece la conexión a la base de datos.

Se crea un cursor para ejecutar comandos SQL.

Se agregan los datos, se guardan los cambios.

Se ejecuta una consulta que tiene todos los registros de la tabla y por último se muestra la tabla.

Figura 7: Tabla en Python

```
1 import psycopg2
2
3 conn = psycopg2.connect("dbname=postgres user=postgres password=koala host=localhost port=5432")
4 cursor = conn.cursor()
5
6 cursor.execute("INSERT INTO redes VALUES ('Daniela', 20154650);")
7 conn.commit()
8
9 cursor.execute("SELECT * FROM redes;")
10 for nombre, carnet in cursor.fetchall():
11     print(f"Nombre: {nombre}, Carnet: {carnet}")
12
13 cursor.close()
14 conn.close()
15 |
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Alejandra\AppData\Local\Programs\Microsoft VS Code\ & C:\Users\Alejandra\AppData\Local\Programs\Python\Python39\python.exe
alejpython.py
Nombre: Alejandra, Carnet: 201600085
Nombre: Carlos, Carnet: 201400524
Nombre: Daniela, Carnet: 20154650
PS C:\Users\Alejandra\AppData\Local\Programs\Microsoft VS Code> |
```

## V. CONCLUSIONES

- \* PostgreSQL es más intuitivo y directo para la gestión de bases de datos, ya que está diseñado específicamente para ese propósito, permitiendo insertar, consultar y visualizar los datos de forma clara y ordenada sin pasos adicionales.
- \* Octave permite conectarse y trabajar con bases de datos como PostgreSQL, pero con más complejidad ya que requiere de carga de paquetes, conexión manual y muestra una tabla que debe interpretarse.
- \* Aunque al igual que Octave, Python requiere establecer una conexión manual con la base de datos y utilizar una librería externa, su enfoque es mucho

más completo. Además permite mostrar resultados en formatos mucho más legibles.

- \* Tanto PostgreSQL como Octave y Python hacen posible trabajar con tablas y bases de datos, aunque cada uno lo hace desde una perspectiva diferente. PostgreSQL está diseñado específicamente para gestionar bases de datos de forma estructurada y eficiente, mientras que Octave y Python, ofrece herramientas para conectarse a bases de datos y manipular esa información desde su entorno.

## VI. ANEXOS

Link de repositorio: <https://github.com/AlejandraMLM/980-Proyectos.git>

- 
- [1] PostgreSQL. Disponible en: <https://www.ibm.com/es-es/think/topics/postgresql>
- [2] Octave Forge. Disponible en: <https://funprojects.blog/tag/sql/>

- [3] SQL. Disponible en: <https://www.actian.com/es/sql-all-you-need-to-know/>
- [4] Python. Disponible en: <https://conovatec.dsdeirl.com/python-para-manejo-de-bases-de-datos-una-introduccion/>