

TAREA 2 : Introducción a PostgreSQL y su Conexión con Octave*

Pablo Andres Montufar Perez, 201902235¹

¹Facultad de Ingeniería, Escuela de Mecánica Eléctrica, Universidad de San Carlos, Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Utiliza el lenguaje SQL (Structured Query Language) para gestionar bases de datos y es conocido por su robustez, escalabilidad y cumplimiento de los estándares. PostgreSQL es ampliamente utilizado en aplicaciones que requieren una alta integridad de datos, tales como aplicaciones web, sistemas empresariales, y análisis de datos complejos.

I. CREACIÓN DE UNA TABLA EN POSTGRESQL

Para comenzar a trabajar con bases de datos en PostgreSQL, uno de los primeros pasos es crear una tabla que sirva como contenedor para los datos. En PostgreSQL, las tablas se definen usando el comando `CREATE TABLE`, donde se especifican los nombres de las columnas, sus tipos de datos y las restricciones que pueden aplicarse (como `NOT NULL`, claves primarias, etc.).

A continuación, se presenta un ejemplo básico de cómo crear una tabla en PostgreSQL para almacenar datos de una red de usuarios:

```
CREATE TABLE redes (  
    Nombre VARCHAR(20),  
    Carnet INT NOT NULL  
);
```

En este ejemplo:

- **Nombre** es una columna de tipo `VARCHAR`, que puede almacenar cadenas de texto de hasta 20 caracteres.
- **Carnet** es una columna de tipo `INT`, que almacenará números enteros, y la restricción `NOT NULL` garantiza que no se puedan insertar registros sin un valor para **Carnet**.

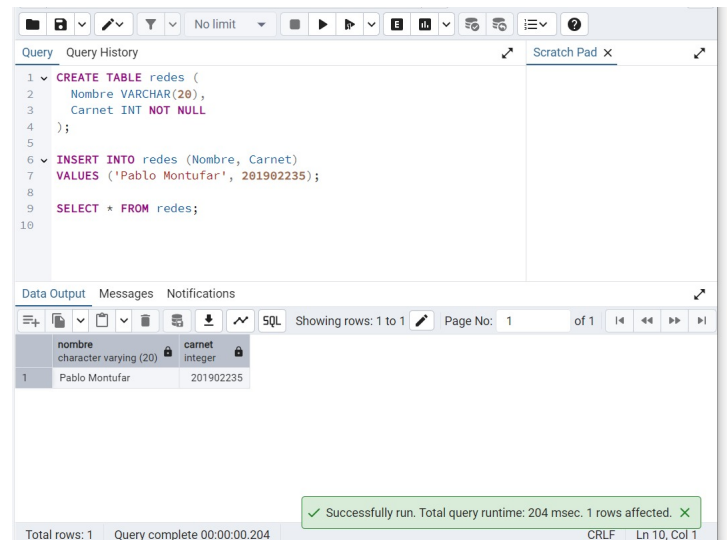


Figura 1: Código para crear la tabla en PostgreSQL

II. CONEXIÓN DE LA BASE DE DATOS CON OCTAVE

Para poder interactuar con PostgreSQL desde Octave, se utiliza el paquete `database` y la función `pq_connect`, la cual establece una conexión con la base de datos. Es necesario especificar los parámetros de conexión como el nombre de la base de datos, el host (generalmente `localhost` para bases de datos locales), el puerto, el usuario y la contraseña.

El siguiente código muestra cómo conectar Octave con PostgreSQL:

```
>> conn = pq_connect(setdbopts('dbname','test','host','localhost',  
'port','5432','user','postgres','password','2008000070*a'))
```

Figura 2: Código para establecer la conexión de Octave con PostgreSQL

Aquí se utilizan los siguientes parámetros:

- **dbname**: El nombre de la base de datos a la que se desea conectar.
- **host**: El servidor donde está alojada la base de datos (en este caso, `localhost`).

* PROYECTOS DE COMPUTACION APLICADA A I.E. Sección A

- **port**: El puerto de la base de datos, que por defecto es 5432 para PostgreSQL.
- **user**: El nombre de usuario para autenticarse en PostgreSQL.
- **password**: La contraseña asociada al usuario de PostgreSQL.

Una vez ejecutado este comando, Octave establecerá una conexión con la base de datos, y se podrán realizar operaciones como consultas, inserciones y modificaciones de datos.

```
>> conn = pq_connect(setdbopts('dbname','taread2','host','localhost', 'port','5432','user','postgres','password','2008000070*a'));
conn = <Pgconn object>
>> |
```

Figura 3: Conexión de Octave a la base de datos PostgreSQL

III. MODIFICACIÓN DE LA TABLA DESDE OCTAVE

Con la conexión establecida, ahora es posible interactuar con la base de datos y modificar los datos dentro de las tablas usando comandos SQL. Octave proporciona la función `pq_exec_params` que permite ejecutar consultas SQL, como `INSERT`, `UPDATE`, o `DELETE`, para manipular los datos.

A continuación, se muestra cómo insertar un nuevo registro en la tabla `redes`:

```
error: __pq_connect__ failed
error: called from
pq_connect at line 112 column 8
>> conn = pq_connect(setdbopts('dbname','taread2','host','localhost', 'port','5432','user','postgres','password','2008000070*a'));
conn = <Pgconn object>
>> % Insertar datos en la tabla
>> N = pq_exec_params(conn, "insert into redes values ('Maria','201900000');")
N =
>> N=pq_exec_params(conn, 'select * from redes;') %ver datos en la tabla
N =
scalar structure containing the fields:
data =
columns =
types =
```

Figura 4: Código para modificar y agregar un nombre

Este código inserta un nuevo registro en la tabla `redes`, con el nombre "Maria" el carnet "201900000".

```
Ventana de comandos
>> N=pq_exec_params(conn, 'select * from redes;') %ver datos en la tabla
N =
scalar structure containing the fields:
data =
{
[1,1] = Pablo Montufar
[2,1] = Maria
[1,2] = 201902235
[2,2] = 201900000
}
columns =
{
[1,1] = nombre
[1,2] = carnet
}
types =
1x2 struct array containing the fields:
name
is_array
is_composite
is_enum
elements
>> |
```

Figura 5: Inserción del nombre en la tabla desde Octave

```
Ventana de comandos
data =
{
[1,1] = Pablo Montufar
[2,1] = Carlos
[3,1] = Maria
[4,1] = Juan
[5,1] = David
[1,2] = 201902235
[2,2] = 201400524
[3,2] = 201900001
[4,2] = 201900002
[5,2] = 201900003
}
columns =
{
[1,1] = nombre
[1,2] = carnet
}
types =
1x2 struct array containing the fields:
name
is_array
is_composite
is_enum
elements
>>
```

Figura 6: Inserción de datos en la tabla desde Octave

A. ¿Cómo Funciona Este Proceso?

El proceso de conexión de Octave con PostgreSQL implica varios pasos:

1. Primero, debes crear una base de datos y tablas en PostgreSQL usando SQL.
2. Luego, estableces la conexión desde Octave a PostgreSQL usando la función `pq_connect`, proporcionando los parámetros de conexión correctos.
3. Una vez que la conexión está establecida, puedes ejecutar comandos SQL para modificar o consultar los datos en la base de datos directamente desde Octave.
4. Los resultados de las consultas, como las inserciones de registros, pueden ser manejados dentro de Octave para análisis o procesamiento adicional.

B. ¿Para Qué Sirve Este Proceso?

Este proceso es útil para integrar PostgreSQL con Octave, lo que permite:

- Almacenar y gestionar grandes volúmenes de datos de manera eficiente en una base de datos.
- Realizar análisis de datos de forma programática desde Octave, aprovechando su capacidad de procesamiento y visualización.
- Automatizar tareas de inserción, actualización y eliminación de datos desde Octave, integrando las bases de datos dentro de un flujo de trabajo analítico o de investigación.

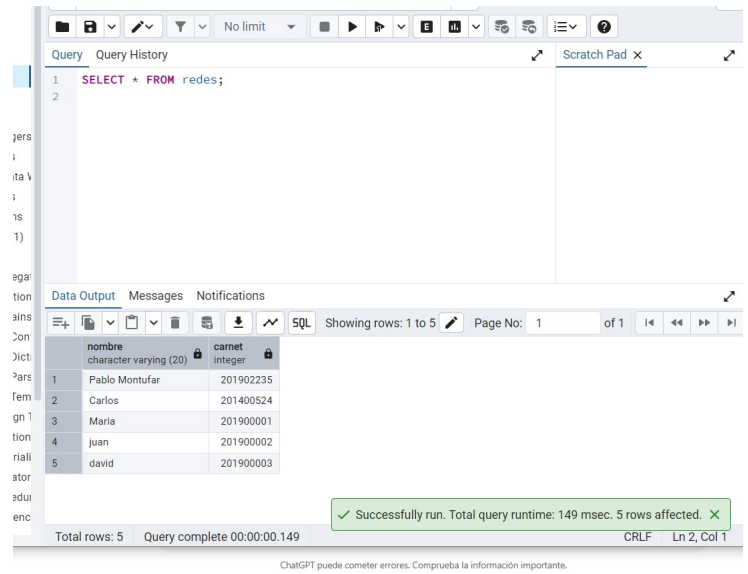


Figura 7: Inserción de datos en la tabla desde Octave

1. Finalizacion

Finalmente la tabla queda de la siguiente forma en PostgreSQL y llamamos a nuestra tabla modificada usando el código:

```
SELECT * FROM redes;
```

IV. REPOSITORIO DEL PROYECTO

El código fuente de este proyecto, junto con ejemplos y otros recursos, está disponible en el siguiente enlace:

Repositorio en GitHub

Este repositorio contiene el código original, las mejoras realizadas y las gráficas generadas por los programas descritos en este documento.