

Parcial 2

Paula Alejandra Martínez Huertas

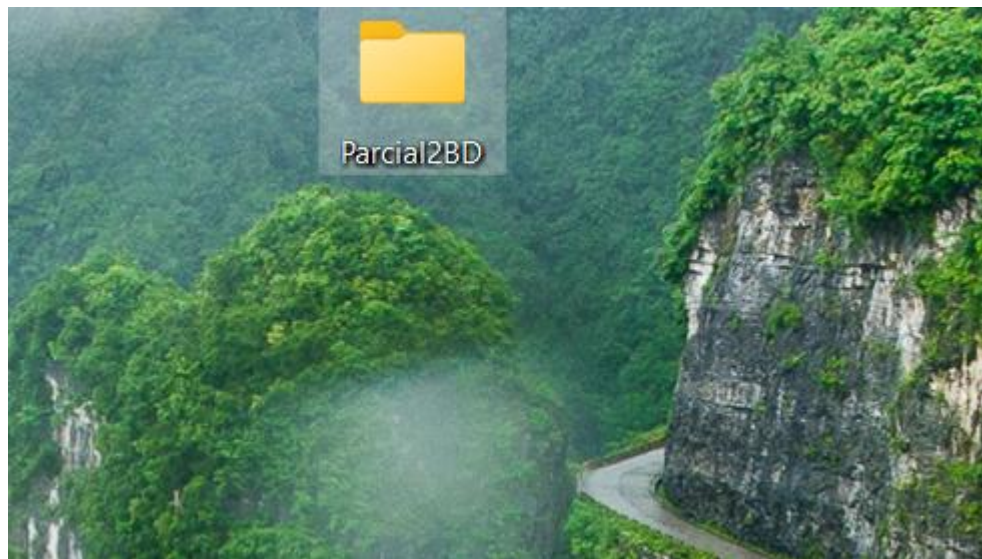
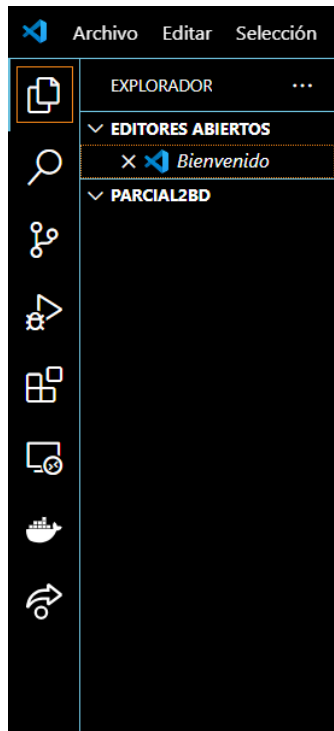
Ingeniería de sistemas, Corporación Universitaria Minuto de Dios

NRC 60747: Base de datos masivas

Ing. William Matallana Porras

Abril 23, 2025

1. Abrimos la carpeta creada en el escritorio en este caso llamada “Parcial2BD” en el espacio de trabajo de nuestro Visual Studio Code.



2. Creamos un nuevo proyecto en supabase.

alejanhuert109 / Gratis / Nuevo proyecto

Comentario

Crear un nuevo proyecto

Tu proyecto tendrá su propia instancia dedicada y una base de datos Postgres completa. Se configurará una API para que puedas interactuar fácilmente con tu nueva base de datos.

Organización: alejanhuert109 / Gratis

Nombre del proyecto: PARCIALbd2

Contraseña de la base de datos: Copiar

Esta es la contraseña de su base de datos de Postgres, por lo que debe ser segura y difícil de adivinar. [Genere una contraseña.](#)

Región: Este de EE. UU. (Virginia del Norte)

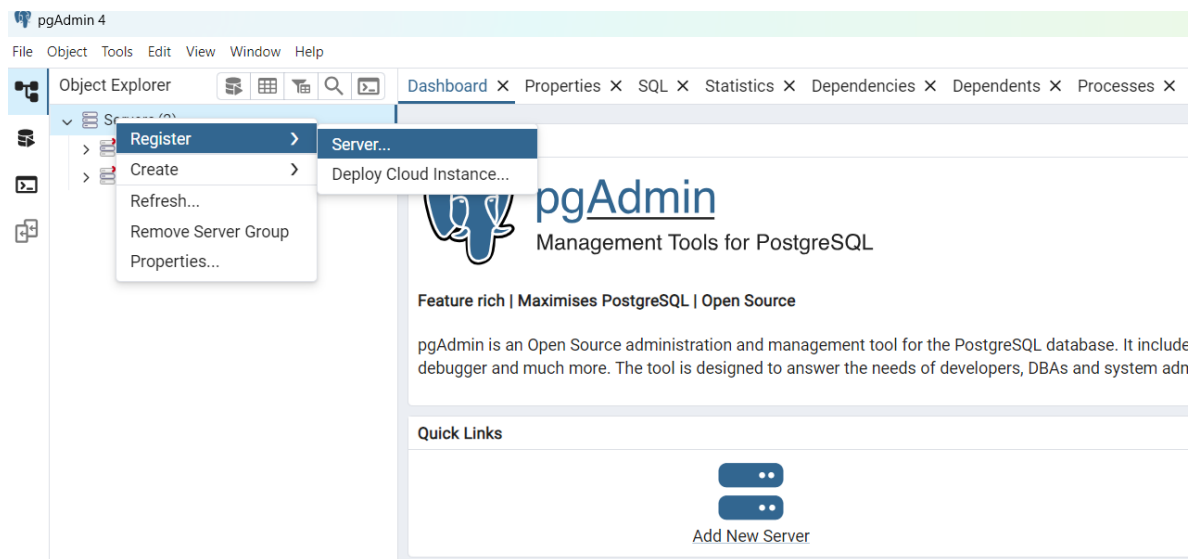
Seleccione la región más cercana a sus usuarios para obtener el mejor rendimiento.

OPCIONES DE SEGURIDAD >

CONFIGURACIÓN AVANZADA >

Cancelar **Crear nuevo proyecto**

3. En pgAdmin se crea un nuevo servidor para establecer la conexión con supabase.



4. Se establece un nombre para el servidor.

Register - Server

General

Connection

Parameters

SSH Tunnel

Advanced

Tags

Name

Parcial2

Server group

Servers

Background

X

Foreground

X

Connect now?

Comments

Either Host name or Service must be specified.

i

?

X Close

Reset

Save

5. Se realiza la respectiva conexión tomando los datos de nuestro proyecto creado en supabase.


Agrupador de sesiones

Pooler compartido

Sólo se recomienda como alternativa a la Conexión directa, cuando se conecta a través de una red IPv4.

```
postgresql://postgres.iuioactaqycnvxnamgmi:[YOUR-PASSWORD]@
```

▼ Ver parámetros

anfitrión : aws-0-us-east-1.pooler.supabase.com 

puerto : 5432

base de datos : postgres

usuario : postgres.iuioactaqycnvxnamgmi

modo piscina : sesión

Por razones de seguridad, su contraseña de base de datos nunca se muestra.

Register - Server

×

General

Connection

Parameters

SSH Tunnel

Advanced

Tags

Host name/address

aws-0-us-east-1.pooler.supabase.com

Port

5432

Maintenance database

postgres

Username

postgres.iuioactaqycnvxnamgmi

Kerberos authentication?

☐

Password

.....

Save password?

☒

Role

Service

i

?

×

Close

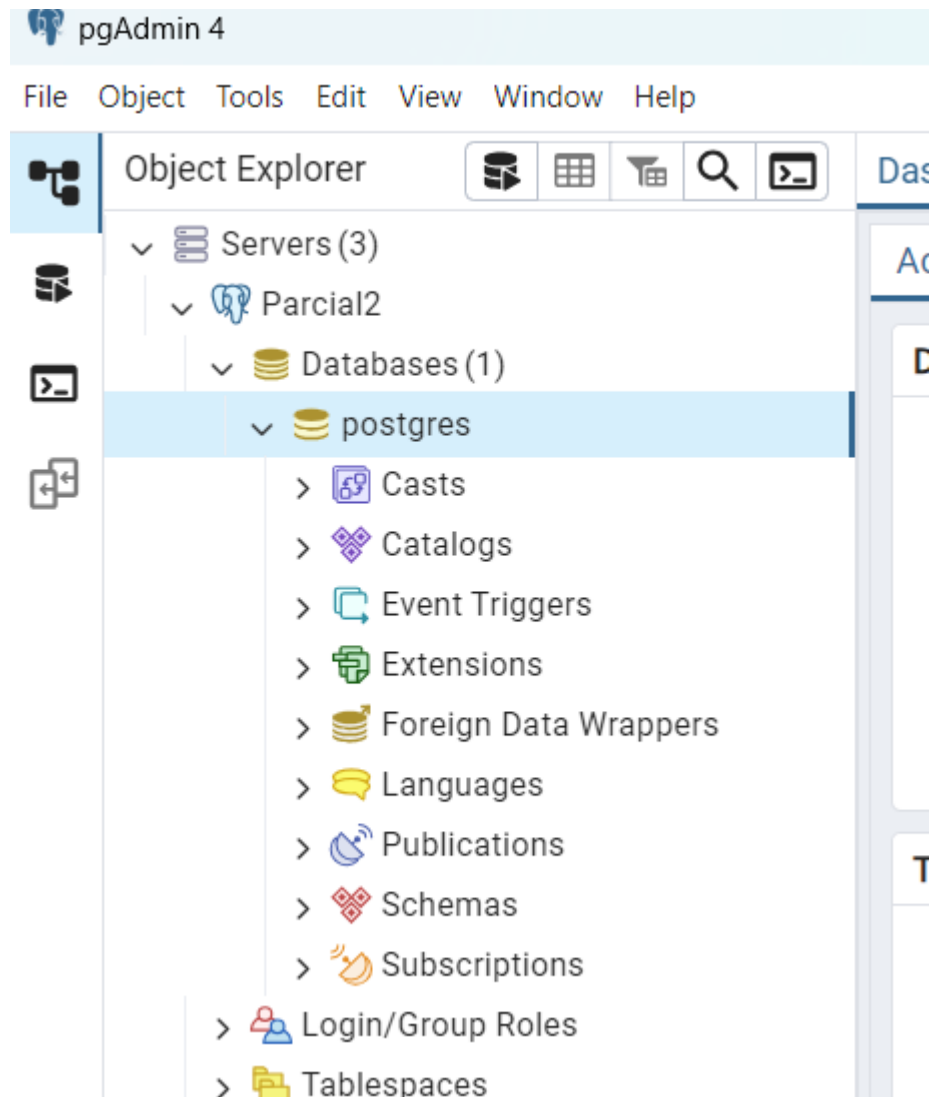
↺

Reset

💾

Save

6. Guardar los cambios de la configuración de la conexión y debe estar creado nuestro servidor “Parcial2”.



7. Creamos las tablas en nuestra base de datos localizada en el servidor “Parcial2” en pgAdmin.

Tabla Restaurante

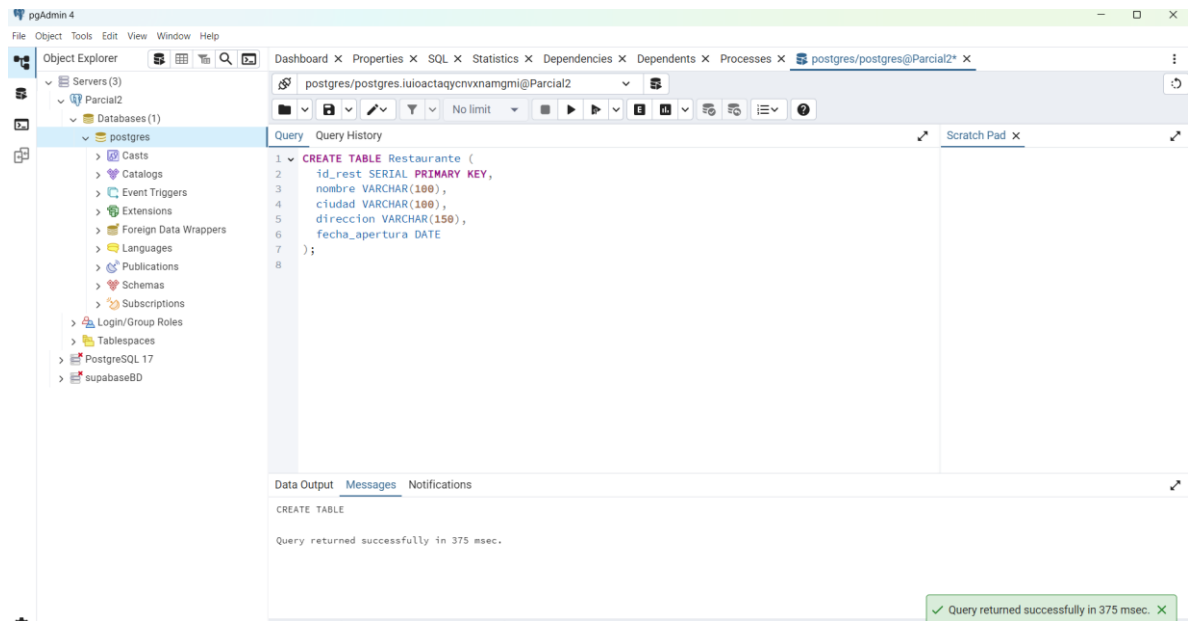


Tabla Empleado

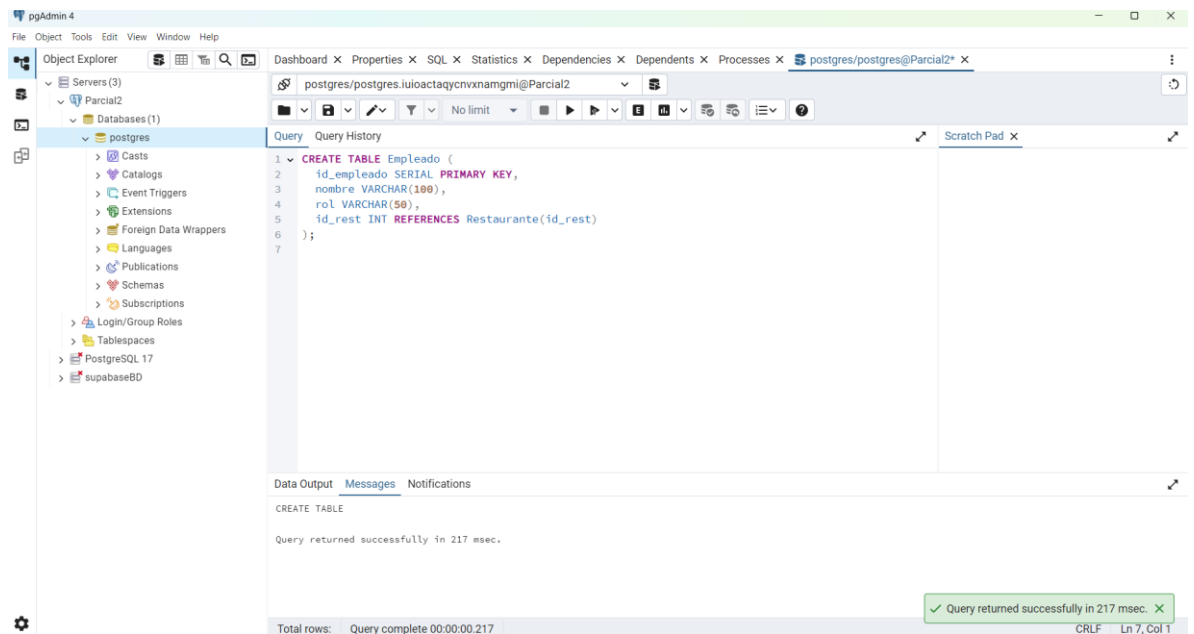


Tabla Producto

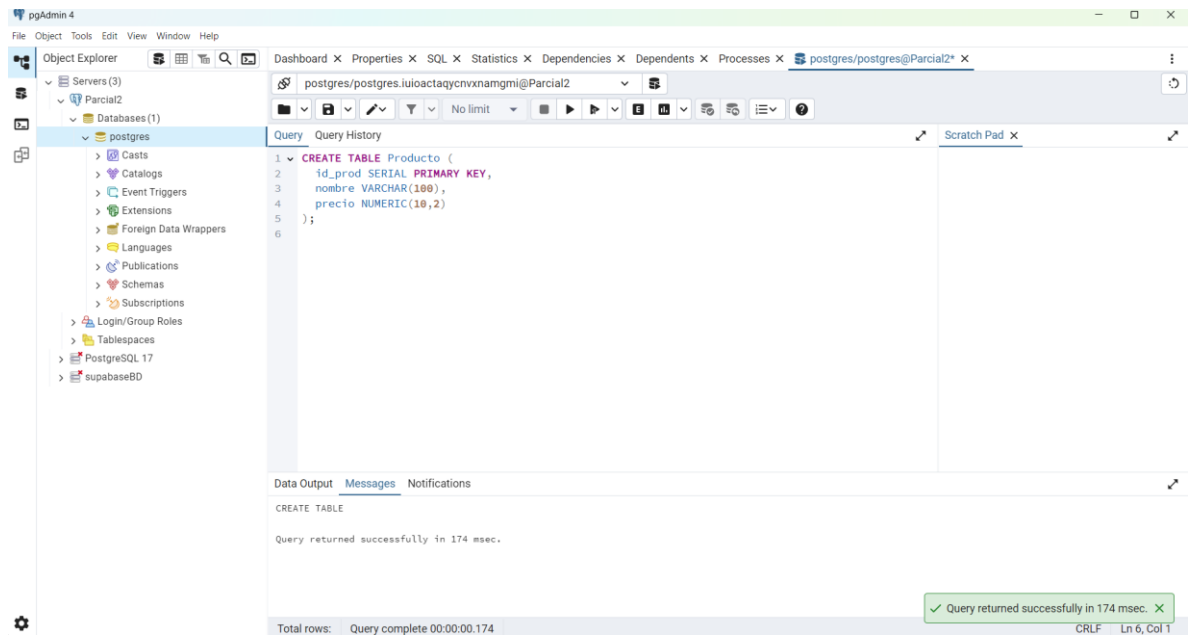


Tabla Pedido

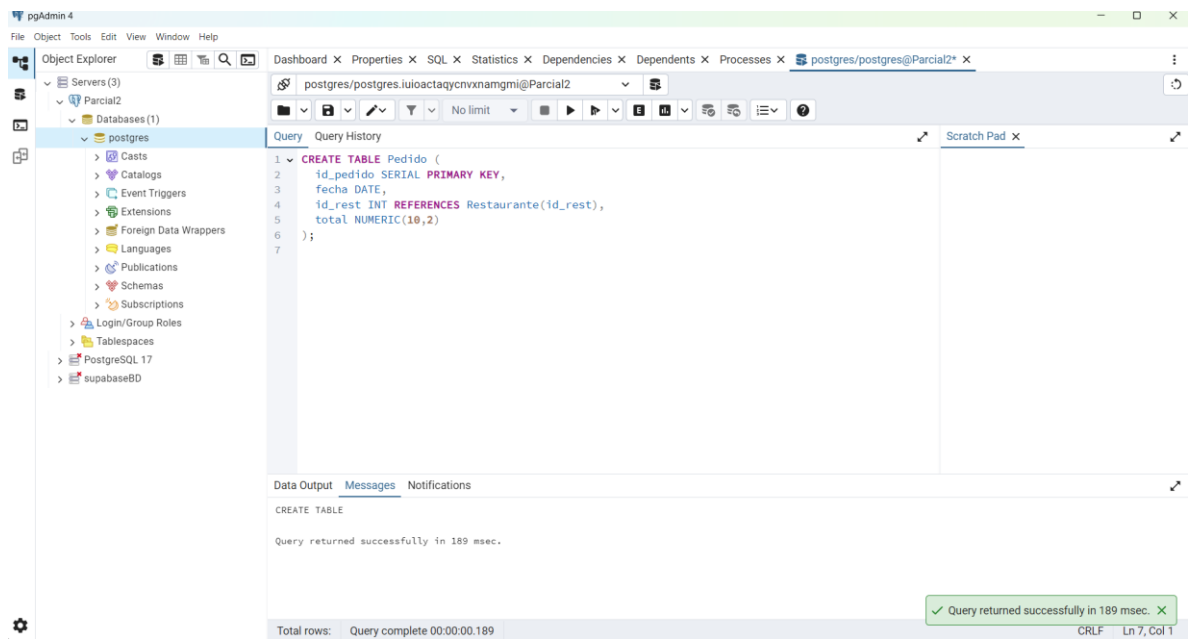
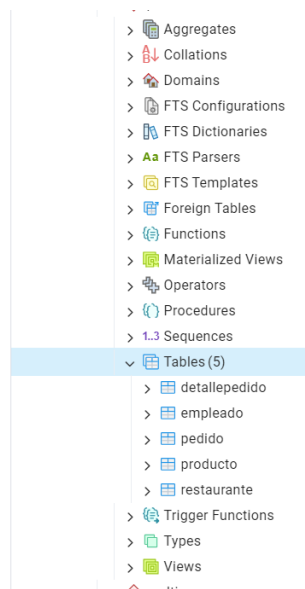
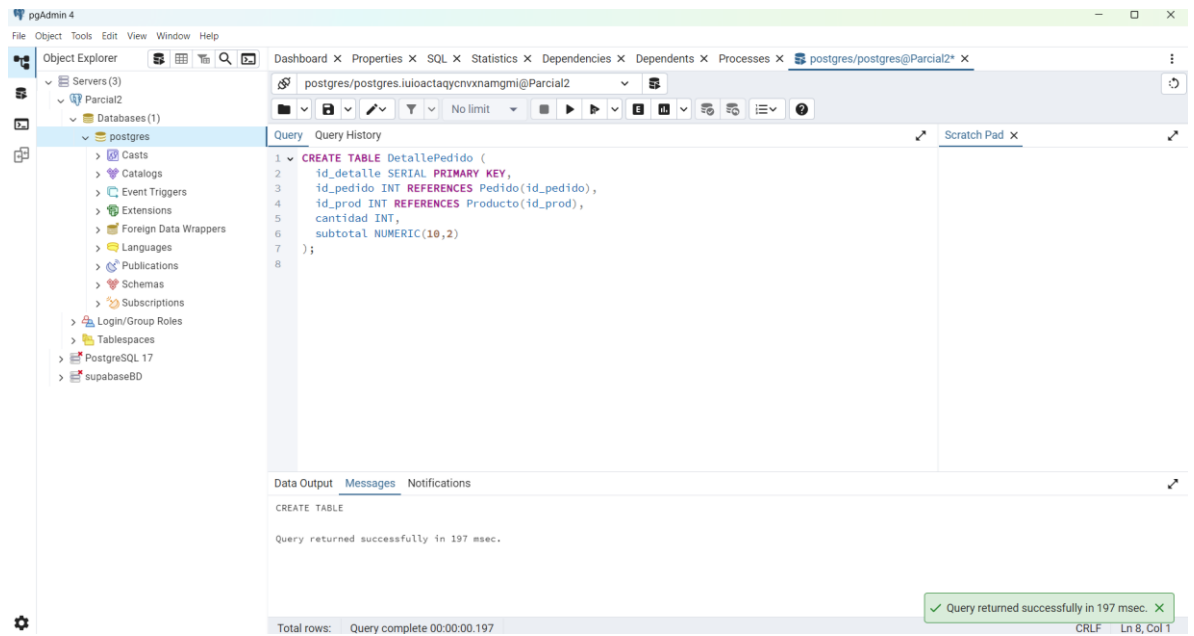
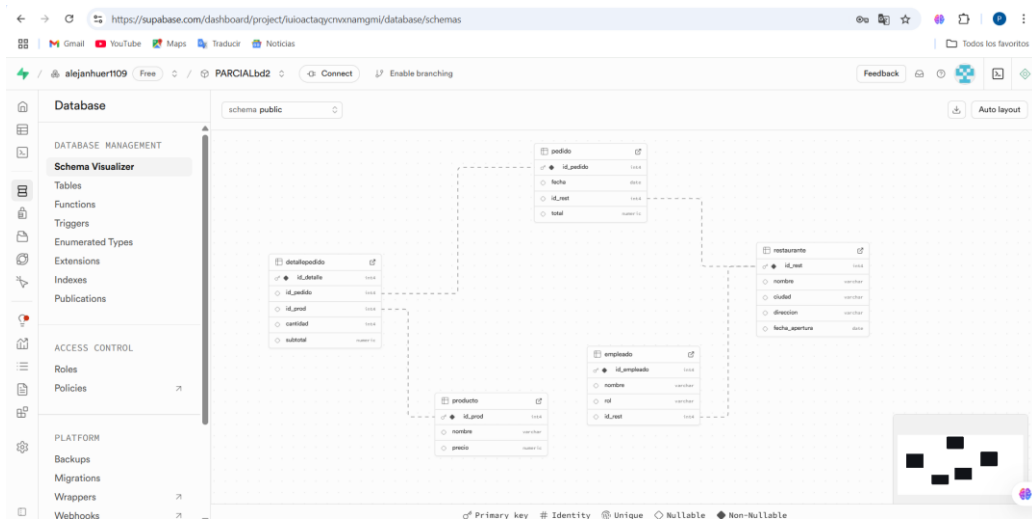


Tabla DetallePedido



8. Modelo relacional obtenido desde supabase.

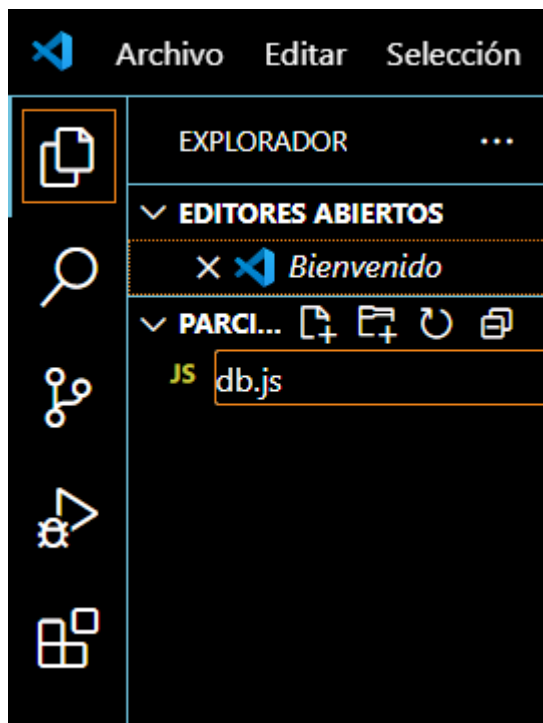


9. Verificamos que las tablas se hallan creado correctamente en el entorno de supabase también.

Database Tables

Name	Description	Rows (Estimated)	Size (Estimated)	Realtime Enabled
detallepedido	No description	0	8192 bytes	X
empleado	No description	0	8192 bytes	X
pedido	No description	0	8192 bytes	X
producto	No description	0	8192 bytes	X
restaurante	No description	0	8192 bytes	X

10. En el entorno de Visual Studio Code dentro de la carpeta a trabajar, en este caso “Parcial2BD” creamos el archivo en lenguaje JavaScript llamado “db.js” para ejercer la conexión con la base de datos con pgAdmin y Supabase.



11. Dentro de este archivo ejecutamos el código para establecer la conexión

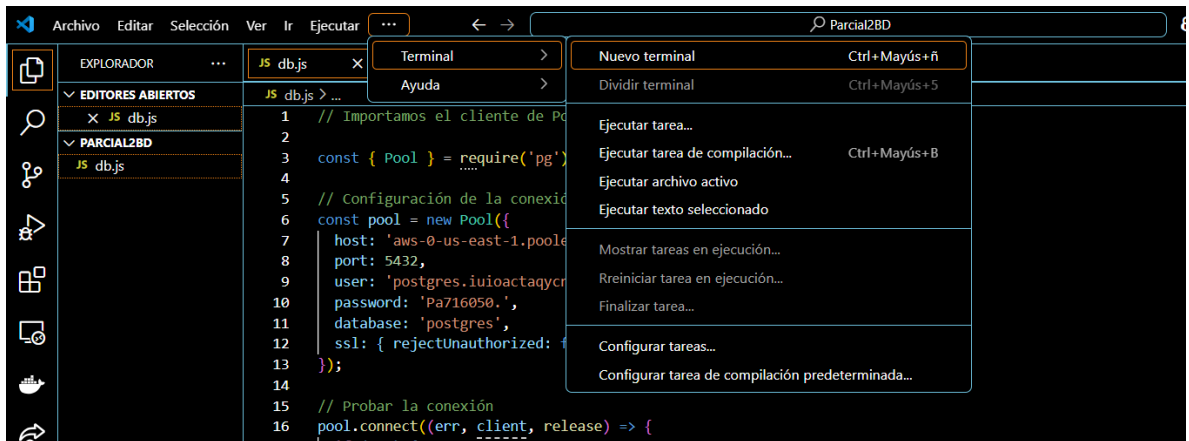
teniendo en cuenta parámetros importantes de nuestra conexión con

Supabase como: **host, port, user, password y database.**



```
1 // Importamos el cliente de PostgreSQL
2
3 const { Pool } = require('pg');
4
5 // Configuración de la conexión a Supabase
6 const pool = new Pool({
7   host: 'aws-0-us-east-1.pooler.supabase.com',
8   port: 5432,
9   user: 'postgres.iuioactaqycrnxnampai',
10  password: 'Pa716050.',
11  database: 'postgres',
12  ssl: { rejectUnauthorized: false } // Supabase requiere SSL
13 });
14
15 // Probar la conexión
16 pool.connect((err, client, release) => {
17   if (err) {
18     console.error('Error conectando con la base de datos', err.stack);
19   } else {
20     console.log('Conectado a la base de datos');
21     release(); // Libera el cliente después de usarlo
22   }
23 });
24
25 module.exports = pool;
26
27
```

12. Abrimos la terminal dentro de nuestro proyecto.



13. Inicializamos un nuevo proyecto con el comando **npm init -y**

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
PS C:\Users\aleja\Desktop\Parcial2BD> npm init -y
```

14. Para instalar los paquetes de PostgreSQL usamos el comando **npm i**

express pg dotenv cors que instala uno o más paquetes de Node.js en el proyecto.

```
PS C:\Users\aleja\Desktop\Parcial2BD> npm i express pg dotenv cors
added 83 packages, and audited 84 packages in 11s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\aleja\Desktop\Parcial2BD>
```

15. Se utiliza **node - - watch index.js** en la terminal para que se detecten los cambios ejercidos en el proyecto automáticamente.

```
PS C:\Users\aleja\Desktop\Parcial2BD> node - - watch index.js
Welcome to Node.js v22.14.0.
Type ".help" for more information.
>
```