

## *Trabajando con Neon*

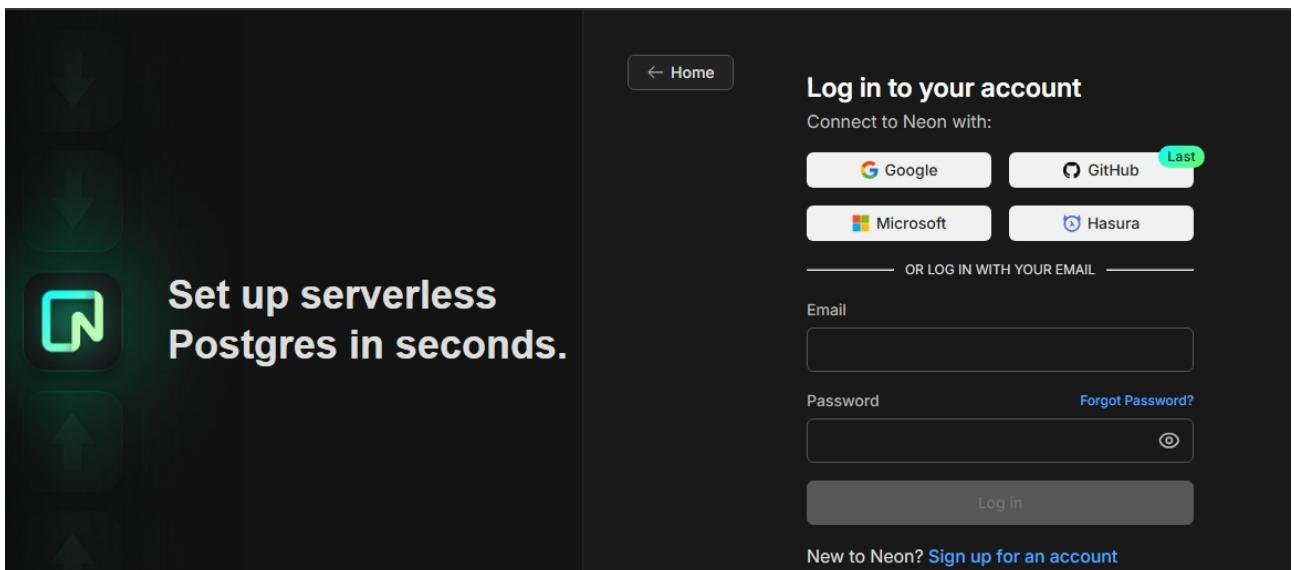
Neon tiene la siguiente misión:

***Our mission: deliver Postgres as a cloud service designed to help teams build scalable, dependable applications faster than ever.***

Como se ve, es un servicio de Postgres en la nube. Básicamente, podremos tener nuestra base de datos Postgres en la nube y conectarnos a ella desde diferentes aplicaciones.

Neon está construido sobre una arquitectura distribuida que separa el almacenamiento del computo. La idea es que coloquemos la base de datos en Neon y, desde la aplicación de Reflex por ejemplo, consumir los datos para mostrarlos posteriormente.

Podemos crear una cuenta gratuita en <https://neon.com/>



*Imagen 1. Página principal de Neon*

Podemos crear nuestra cuenta o conectarnos con Google o Github. Una vez creada nuestra cuenta e iniciada la sesión, veremos algo similar a esto

The screenshot shows the Metabase dashboard for the user 'espinoza.enrique.87@gmail.com'. The left sidebar has sections for Organization (Projects, People, Billing, Integrations, Settings), and the main area displays 'espinoza.enrique.87@gmail.com's projects'. It shows metrics: Compute (0.1 CU-hrs), Storage (0.12 GB), History (0 GB), and Network transfer (0 GB). A search bar is at the bottom. A table lists one project: 'metabase-bootcamp' (Region: AWS US East 1 (N. Virginia), Created at: Nov 19, 2025 10:44 am, Storage: 0, Compute last active at: Dec 8, 2025 8:08 pm, Branches: 2, Integrations: Add).

*Imagen 2. Menú principal al iniciar sesión.*

Si hacemos click en **New Project** nos aparecerá la siguiente ventana:

The screenshot shows the 'Create project' form. It has fields for 'Project name' (placeholder: e.g., app name or customer name) and 'Postgres version' (selected: 17). Below that is a section for 'Cloud service provider' with radio buttons for 'AWS' (selected) and 'Azure'. There is also a 'Region' dropdown set to 'AWS US East 2 (Ohio)' with a note below it: 'Select the region closest to your application.'

*Imagen 3. Ventana de creación de un nuevo proyecto.*

Le damos un nombre y elegimos la zona en donde vamos a establecer nuestro servidor para iniciar el proyecto. Cuando el proyecto se inicie, nos enviará a un dashboard principal como se mostraba en la imagen 2.

Name	Region	Created at	Storage	Compute last active at	Branches	Integrations	
metabase-bootcamp	AWS US East 1 (N. Virginia)	Nov 19, 2025 10:44 am	0	Dec 8, 2025 8:53 pm	2	Add	

*Imagen 4. Lista de proyectos.*

Si seleccionamos el proyecto indicado en la imagen 4, nos mostrará la información relevante de este.

PROJECT

- Dashboard
- Branches
- Integrations
- Auth
- Settings

BRANCH

- development
- Overview
- Monitoring
- SQL Editor
- Tables
- Others

Project dashboard

Get started with your new database

Connect to your database  
Get connection details your app needs to connect to your database

Import your data  
Automatically import your data directly into your Neon database

Connect from your IDE  
Use the Neon Local connect IDE extension to connect to your database.

Branches 2 / 10 Compute 0.1 / 100 CU-hrs Storage 0.12 / 0.5 GB Network transfer 0 / 5 GB

*Imagen 5. Opciones del dashboard del proyecto.*

Azul: Nos indica el nombre del usuario logueado y el nombre del proyecto.

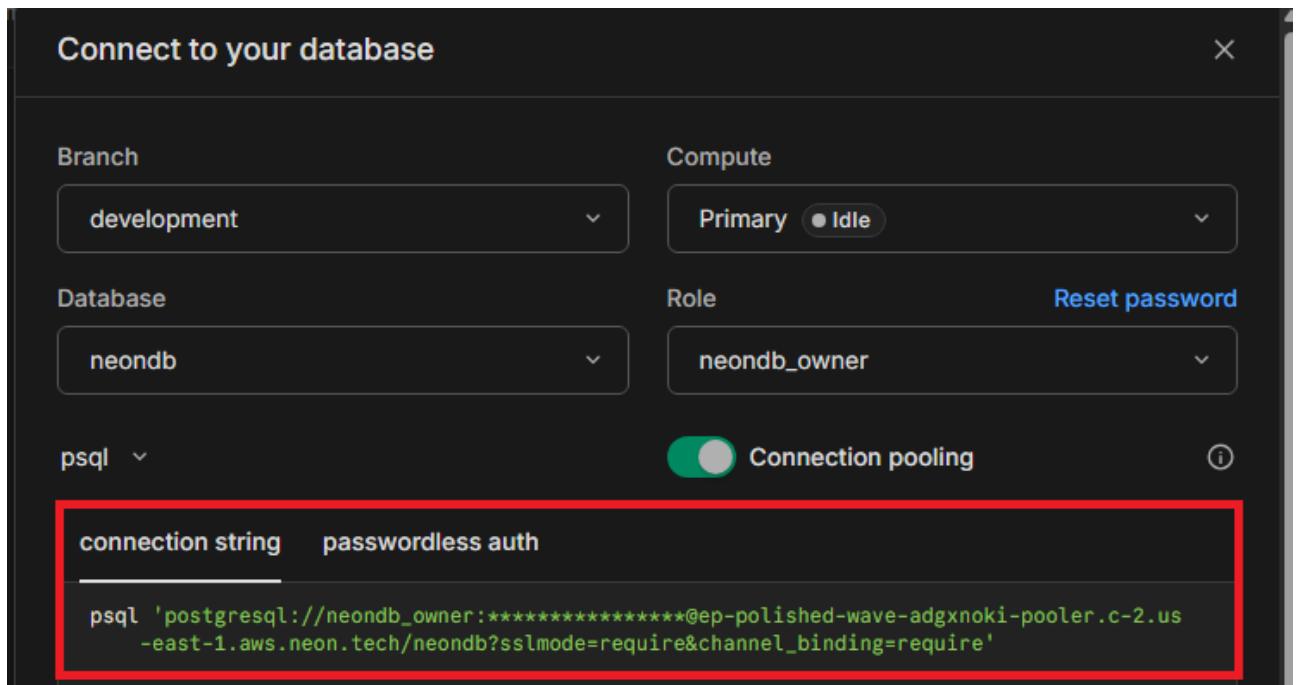
Verde: Nos indica la rama del proyecto en la que estamos trabajando. Podemos tener hasta 10 ramas distintas.

Naranja: Nos muestra las estadísticas de uso de nuestro proyecto. Segundo tengo entendido, algunas se resetean a fin de mes como el caso de **Compute**. Pero **Branches**, **Storage** tienen un límite mensual.

Rosa: Es el editor SQL que puede usarse como se hace en pgadmin.

Rojo: Una de las primeras partes importantes. Esta nos permite obtener el string de conexión.

Si hacemos click en la opción roja, podremos observar lo siguiente:



*Imagen 6. Opciones del string de conexión.*

En la imagen 6 vemos la información relevante para el string de conexión. Estas son:

-Branch: La rama del proyecto donde estamos trabajando.

-Compute: Es la máquina que se encargará de computar nuestro trabajo. Como tenemos la versión gratuita, constamos sólo con una.

-Database: El nombre de la base de datos. Por defecto, nos brinda una que es **neondb**. Pero nosotros podemos crear las nuestras.

-Role: El rol del usuario con esa conexión.

La parte en rojo es nuestro string de conexión. Podemos desglosarlo de la siguiente manera:

Parte	Significado
psql	Cliente para conectarte
postgresql://	Protocolo
neondb_owner	Usuario
*****	Contraseña. La opción <b>Show Password</b> mostrará este valor en la cadena de conexión
ep-polished-wave-adgxnoki-pooler...	Servidor Neon
neondb	Nombre de la base

Parte	Significado
sslmode=require	Requiere conexión encriptada
channel_binding=require	Seguridad adicional

Esta información es útil para poder conectarnos mediante pgadmin. Y ahora, vamos a ver cómo lo hacemos. Para ello, vamos a pgadmin y hacemos clic en **Servers → Register → Server**

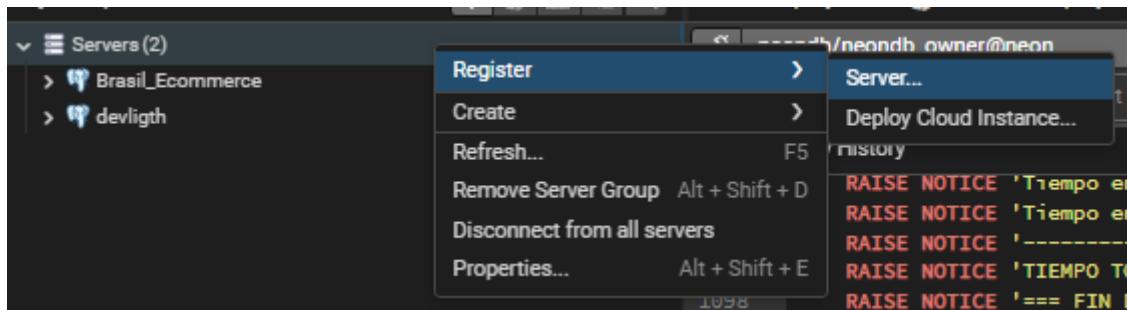
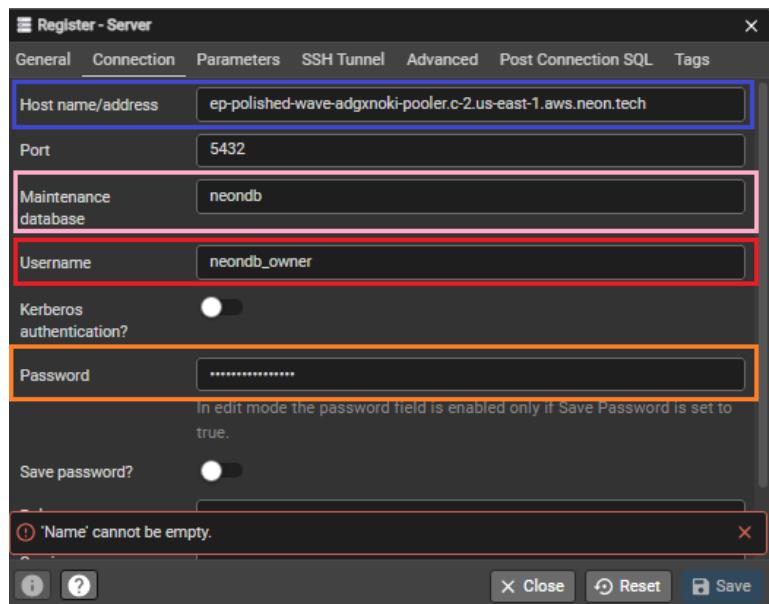


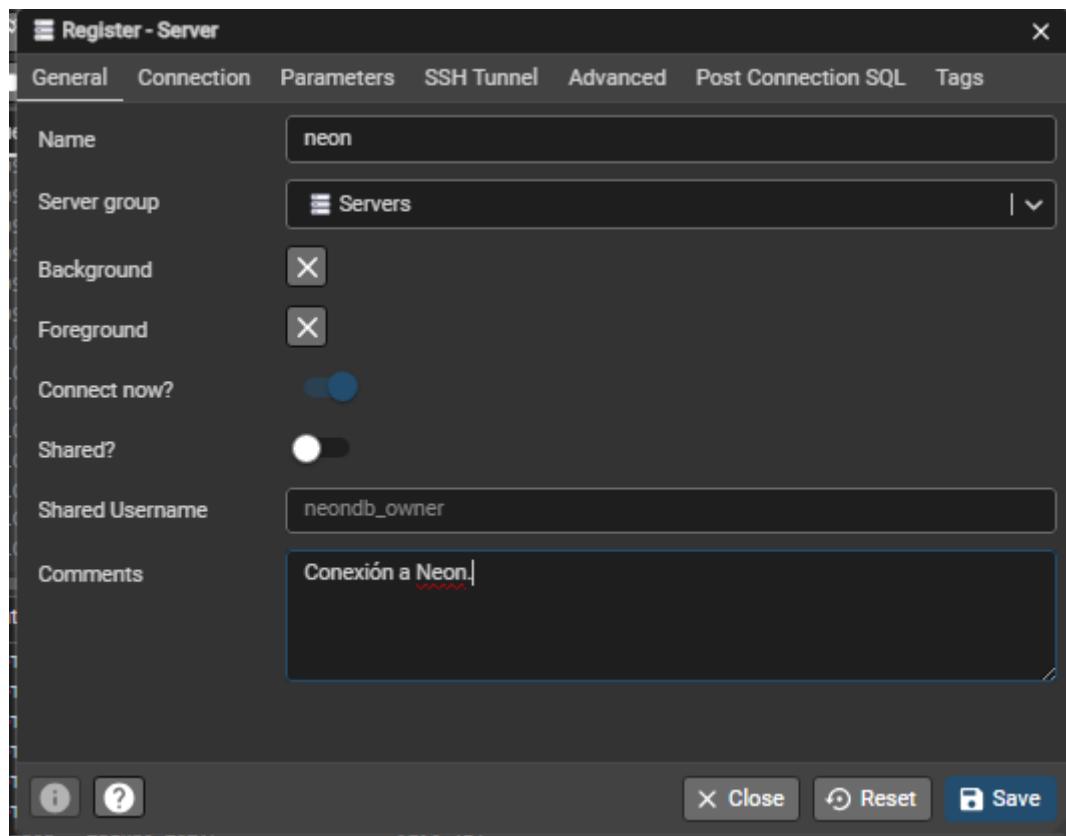
Imagen 7. Crear un nuevo servidor.



*psql*

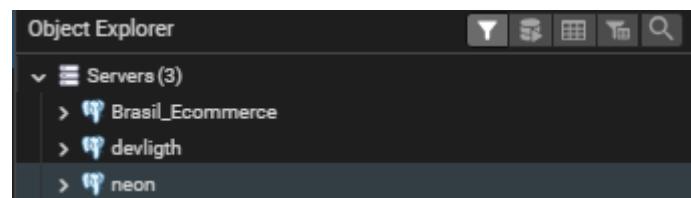
```
'postgresql://neondb_owner:
*****@ep-polished-
wave-adgnoki-pooler.c-2.us-
east-1.aws.neon.tech/neondb?
sslmode=require&channel_bind-
ing=require'
```

Imagen 8. Datos para la conexión en la opción Connection.



*Imagen 9. Datos generales para la conexión.*

Damos a **Save** y notaremos que se creo la nueva conexión.



*Imagen 10. Nueva conexión creada.*

Podemos empezar a ejecutar código SQL. Vamos a cargar una nueva tabla en Neon. Si vamos a la opción **SQL Editor** de Neon, realicemos la siguiente sentencia:

The screenshot shows a PostgreSQL query tool interface. At the top, there is a code editor with the following content:

```
12 | select * from games;
```

Below the code editor, there is a status bar indicating "Connected (1 query)". Underneath the status bar are three buttons: "Run", "Explain", and "Analyze".

In the main area, a red box highlights an error message:

**⚠ ERROR: relation "games" does not exist (SQLSTATE 42P01)**

The error message is preceded by the query:

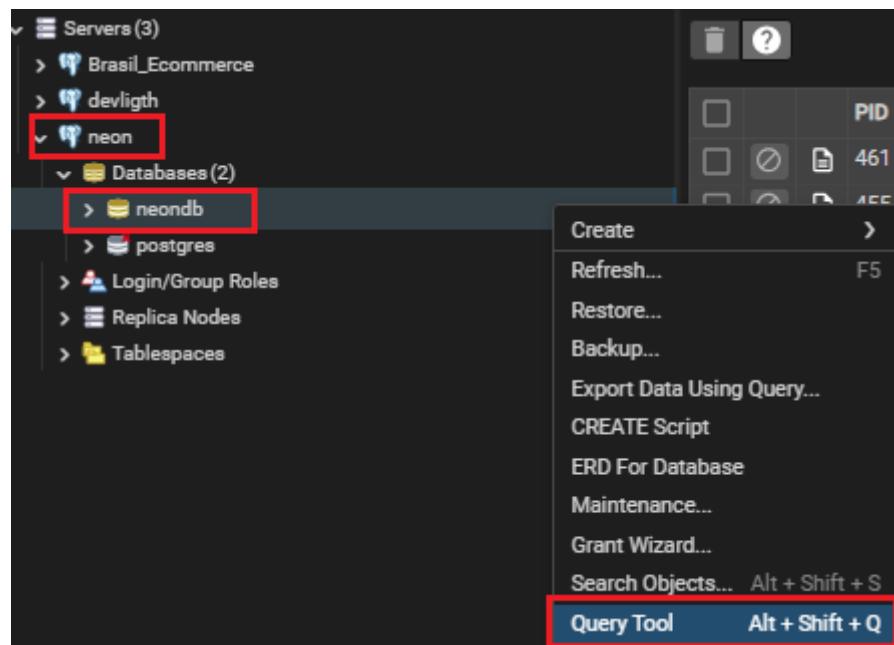
```
select * from games
```

To the right of the error message is a button labeled "Fix with AI".

*Imagen 11. Error en la consulta.*

En la imagen 11 nos da claramente un error, porque no tenemos una tabla llamada ***games***. Vamos a crear una de la siguiente manera:

1- Creemos una ***Query Tool*** con la conexión a Neon



*Imagen 12. Abrir una query tool para la conexión con Neon.*

2-En esa query tool, ejecutar el siguiente código:

```
CREATE TABLE video_game_sales (
    rank          INTEGER,
    name          VARCHAR(255),
    platform      VARCHAR(100),
    year          INTEGER,
    genre          VARCHAR(100),
    publisher     VARCHAR(255),
    na_sales      NUMERIC(10,2),
    eu_sales      NUMERIC(10,2),
    jp_sales      NUMERIC(10,2),
    other_sales   NUMERIC(10,2),
    global_sales  NUMERIC(10,2)
);
```

*Imagen 13. Código para crear la tabla de juegos.*

Como vemos la tabla se llamada **video\_game\_sales**. Si ejecutamos ese código y volvemos a la opción de **SQL Editor** en Neon, notaremos esto:

The screenshot shows the Neon SQL Editor interface. In the main text area, there is a single line of SQL code: "12 | select \* from video\_game\_sales;". Below the text area, a status bar displays "Connected (1 query)". At the bottom of the interface, there are three buttons: "Run", "Explain", and "Analyze". A message at the bottom of the screen states "Statement executed successfully".

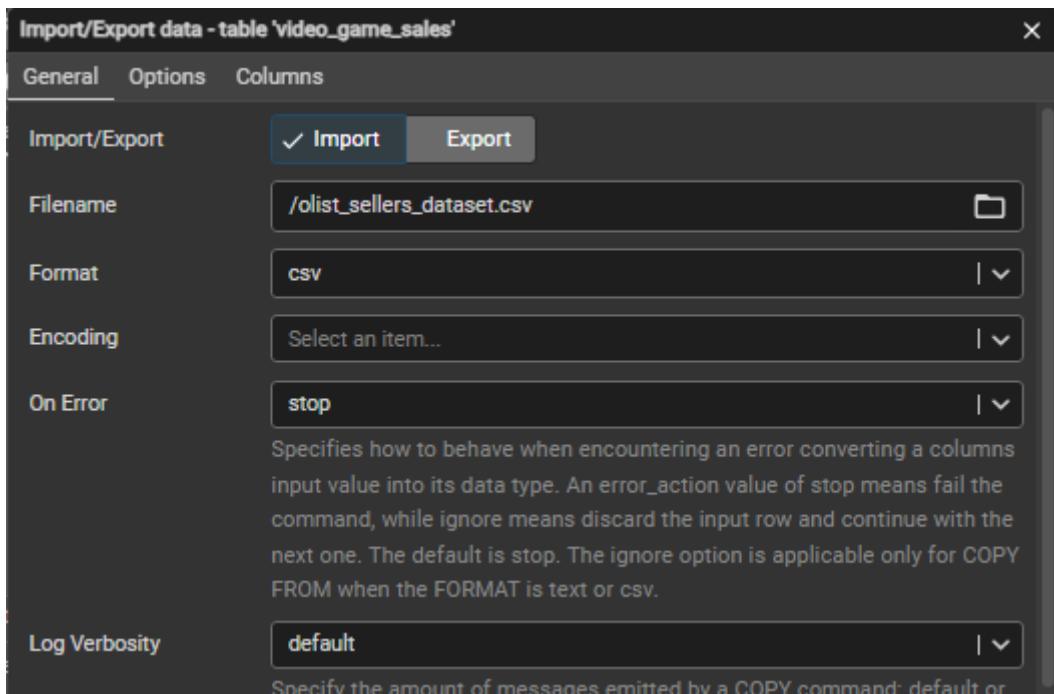
*Imagen 14. Ejecución de código SQL nuevamente en Neon.*

Notamos que no nos da un error porque ahora existe esa tabla. Pero, no nos muestra nada porque está vacía. Para poblarla, descarguemos este archivo:

<https://www.kaggle.com/datasets/zahidmughal2343/video-games-sale>

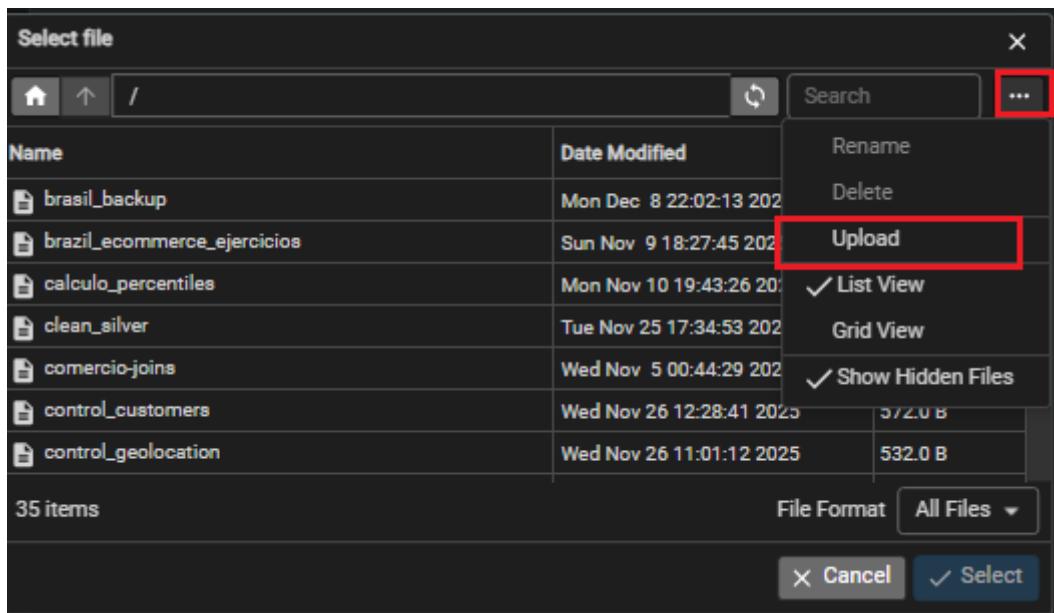
En pgadmin, nos vamos a **neon → databases → Schemas → public → Tables**

Debemos ver la tabla **video\_games\_sales**. Hacemos click derecho sobre ella y elegimos **Import/Export Data**. Nos aparecerá lo siguiente:



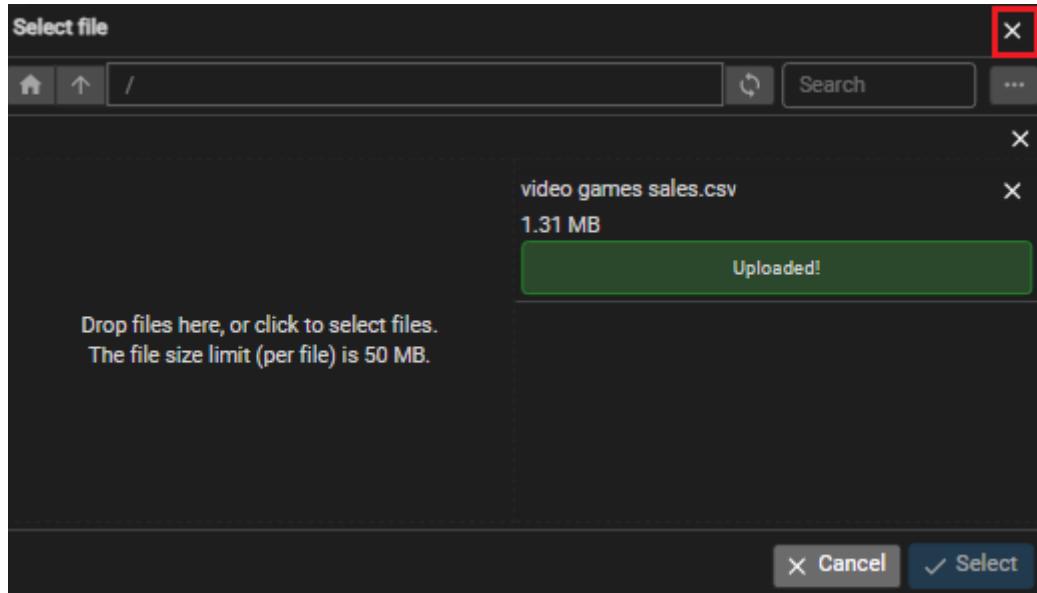
**Imagen 15.** Ventana para la importación de datos.

En la opción **Filename**, hacemos click sobre el ícono de la carpeta. En la nueva ventana que se nos muestra en la siguiente imagen, elegimos los tres puntos y luego la opción **Upload**



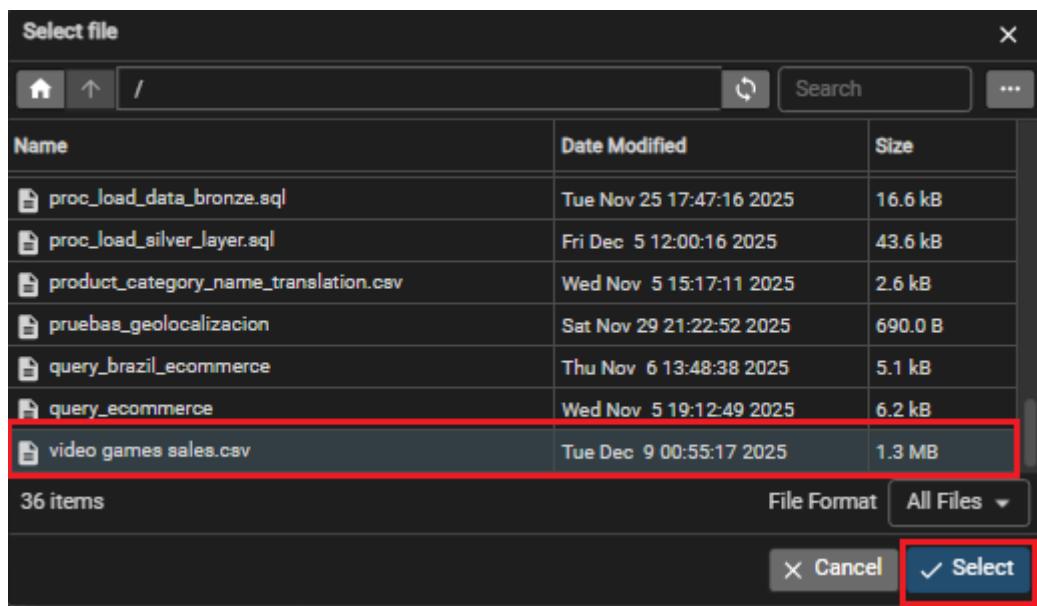
**Imagen 16.** Opción para cargar los datos a importar.

Así se abrirá una nueva ventana en donde debemos arrastrar el archivo con los datos de la venta de juegos.

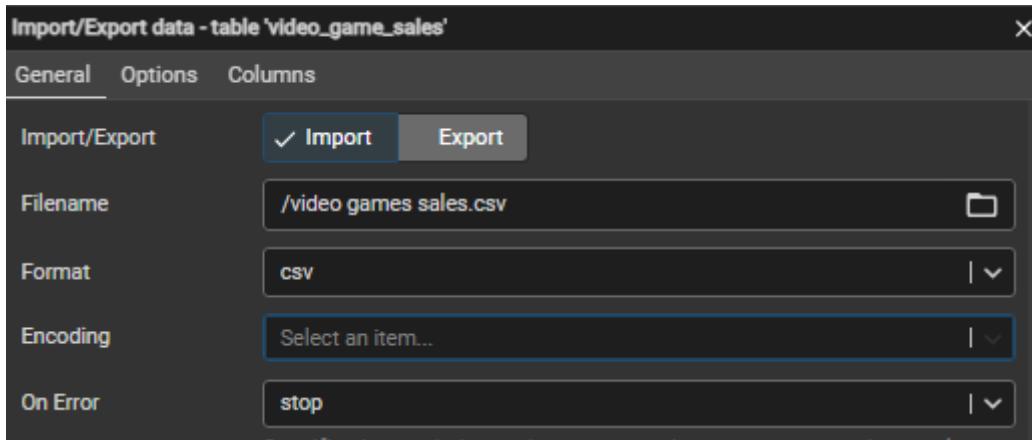


**Imagen 17. Ventana para arrastrar los datos para la importación.**

Cuando el archivo termine de cargarse hacemos click en la cruz indicada en la imagen 17. Eso nos retornará a la ventana de la imagen 16. Simplemente buscamos el archivo que acabamos de cargar y elegimos **Select**.

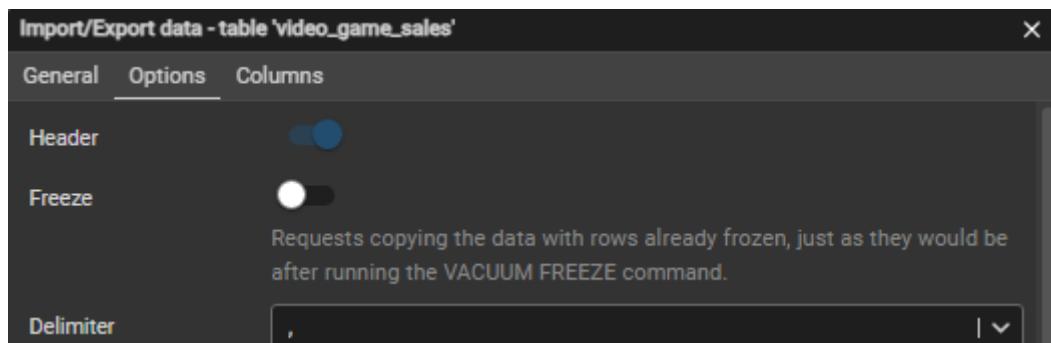


**Imagen 18. Selección del nuevo archivo.**



**Imagen 19. Archivo preparado para ser importado.**

Antes de importar el archivo, seleccionamos la opción **Options** de la imagen 19.



**Imagen 20. Opciones de importación.**

En esas opciones, si nuestro archivo tiene cabecera, debemos dejar activa la opción **Header**. Luego, en **Delimiter** indicamos el delimitador de nuestro archivo.

**Nota:** El archivo contiene valores 'N/A' en campos que supuestamente deben ser numéricos.

Podemos evitar errores si en la imagen 15, en la opción **On Error** elegimos **Ignore**. Esto lo hacemos así porque el propósito de esta sección es mostrar cómo llenar las tablas en Neon.

Si volvemos a Neon:

```
12 select * from video_game_sales limit 10;
```

Connected (1 query)

Run Explain Analyze 199ms 10 rows

#	rank	name	platform	year	genre	publisher
1	1	Wii Sports	Wii	2006	Sports	Nintendo
2	2	Super Mario Bros.	NES	1985	Platform	Nintendo
3	3	Mario Kart Wii	Wii	2008	Racing	Nintendo

*Imagen 21. Nuevo resultado en Neon.*

## Referencias

<https://www.youtube.com/watch?v=YFTYxhgJcHA>  
<https://www.kaggle.com/datasets/zahidmughal2343/video-games-sale>