

Puesta en servicio de Base de Datos de Sistema de Gestión

Paso a paso

Durante la clase del martes 15/10 se siguieron los pasos necesarios para disponer la base de datos de Sistema de Gestión, tanto en *PostgreSQL* como en *SQLServer*. La idea es tener una base con los mismos datos en ambos servidores, para tener la posibilidad de ejercitar y probar con los dos entornos. Los pasos seguidos durante la clase son los mismos que se deberían realizar si se necesitara migrar desde una base de datos PostgreSQL hacia una base de datos SQLServer.

En este apunte, intentamos repetir paso a paso las tareas.


Creación de la base en PostgreSQL

1. Descargar de la plataforma e-fich “BD Gestion.zip” de la publicación que se muestra en este gráfico:

Semana 8: Lunes 7/10 a Jueves 10/10

Martes 08/10/2023

El martes 8/10 se trabajó sobre el comando SELECT, utilizando una guía con ejemplos de consultas sobre la base de datos de "Gestión Comercial".

 **Sistema de Gestión Comercial**

Se adjunta en formato comprimido (.zip) los elementos necesarios para crear la base de datos del Sistema de Gestión Comercial, en PostgreSQL.

En el .zip se encuentran:

1. Modelo PowerDesigner (.pdm), con los 3 submodelos de trabajo: Persona, Producto y Venta. De este mismo modelo, se incluye también una versión en pdf.
2. Archivo "1-gestion_crea_tablas.sql": Scripts de creación de esquemas y tablas. Se recomienda previamente crear una base de datos con Pgadmin o con DBeaver. Una vez creada la base, conectarse y ejecutar el script completo de creación.
3. Archivo "2-gestion_carga_datos.sql". Scripts de carga de datos ejemplo. Se recomienda, una vez completado el paso anterior de creación de tablas, ejecutar el script en forma completa.

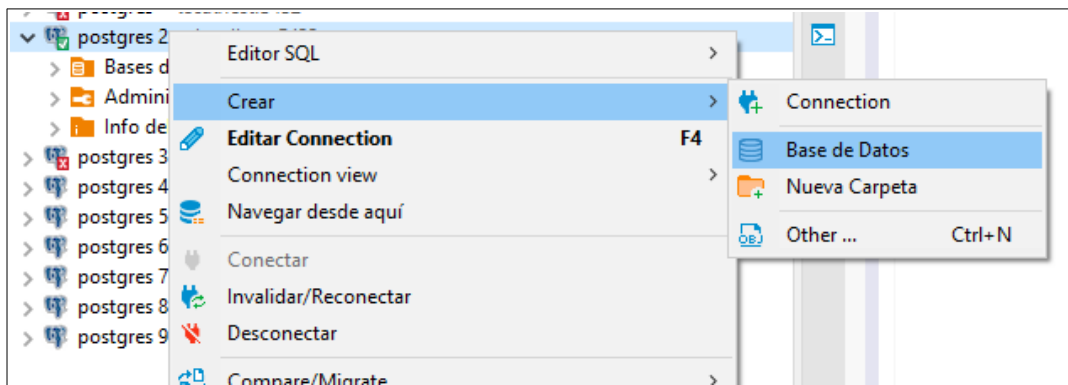
Una vez descargado el archivo en la PC tendrán:

1. Diagramas del modelo físico de la base de datos (tanto en PowerDesigner como en PDF). Archivos:
 - “BD 2024 - Modelo Físico Sistema de Gestión.pdf”
 - “Gestion.pdm”
2. Archivo de creación de esquemas y tablas: “1-gestion_crea_tablas.sql”
3. Archivo de carga de datos: “2-gestion_carga_datos.sql”.

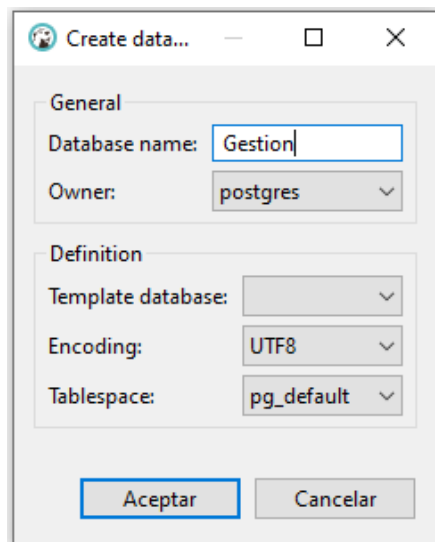
Previamente deben tener instalado en la PC:

1. PostgreSQL (en mi caso tengo la versión 12)
2. SQL Server (en mi caso tengo la versión 2008)
3. Dbeaver, con conexión a ambos servidores (1 y 2).

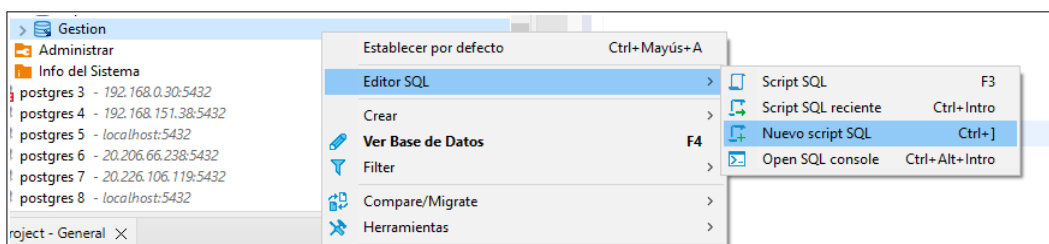
2. Desde Dbeaver crear en PostgreSQL una base de datos nueva:



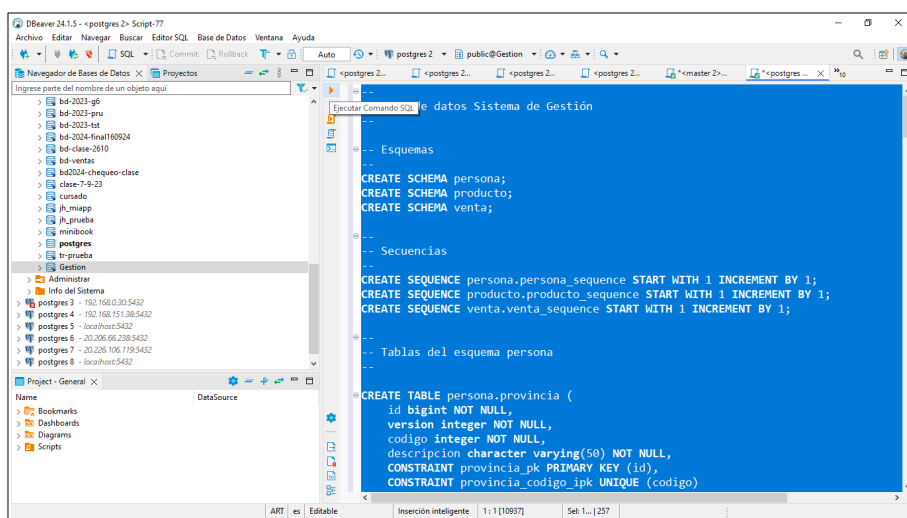
La interfaz les solicitará un nombre de base de datos. Poner “Gestion”, dejar el resto de valores por defecto y confirmar:



3. Crear esquemas y tablas: una vez generada la base de datos, ubicarla y realizar una conexión para ejecutar un “nuevo script”:

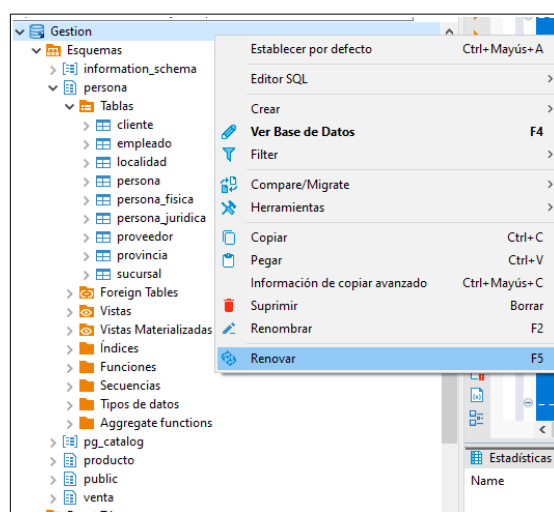


Una vez conectado a la base de datos, abrimos desde un editor el archivo “1-gestion_crea_tablas.sql”, seleccionamos todo y copiamos dentro de la consola de Dbeaver:



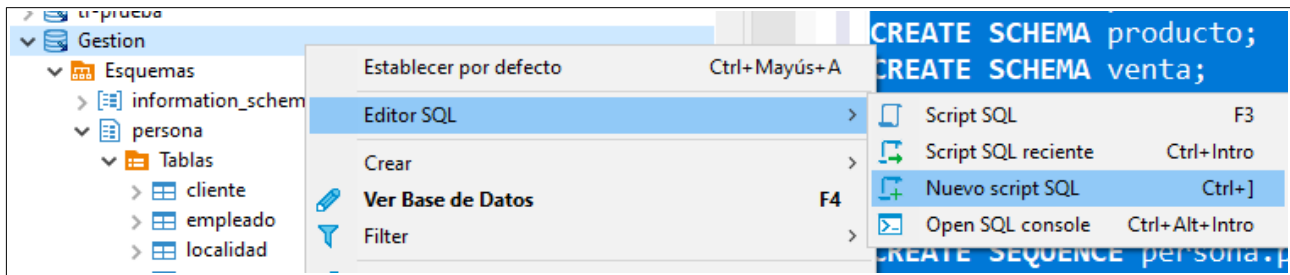
Marcamos todo el contenido del script y le damos “Ejecutar”.

El script nos creará los esquemas y sus tablas. Ubicandonos sobre “Gestion” e indicando “Renovar”(F5), nos mostrará los esquemas persona, producto y venta (y demás esquemas propios de Postgres):

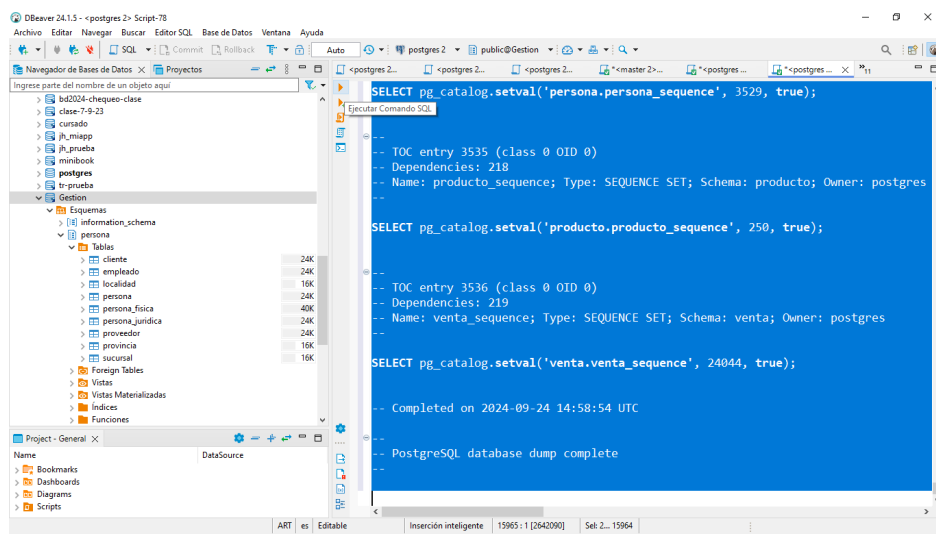


4. Cargar datos:

Conectado a la BD Gestion, generar una nueva conexión para ejecutar scripts sql:



Cargar en la nueva conexión el contenido del archivo: “2-gestion_carga_datos.sql”. Seleccionar todo y ejecutar:



Luego de unos instantes, si todo fue bien, deberían estar cargados los datos en las tablas.

Para verificar ejecutamos el siguiente script:

```
select count(*) from venta.factura_detalle fd
```

El mismo nos informará que tabla factura_detalle tiene 10.003 filas cargadas.

Si llegamos a este punto, buen trabajo! Tenemos la base de datos Gestion dispuesta en un servidor PostgreSQL. Podemos ejercitar trabajando con las guías de consultas Select en este entorno.

Migración a SQLServer


1. Descargar de la plataforma e-fich “BD Gestion SQL Server.zip” de la publicación que se muestra en el siguiente gráfico:

Temas 4 y 5

Semana 9: Lunes 14/10 a Viernes 18/10

Martes 15/10/2023

Esta martes continuaremos con la ejercitación de SELECT. De ser posible, trataremos de abordar los temas: Procedimientos Almacenados y Triggers.

 **Comando Select - Guía Nro. 2**

Guía de trabajo en clase Nro. 2 sobre uso del comando Select.

Sistema de Gestión Comercial - SQL Server

Se adjunta en formato comprimido (.zip) los elementos necesarios para migrar la base de datos del Sistema de Gestión Comercial, desde PostgreSQL hacia SQLServer.

En el .zip se encuentra:

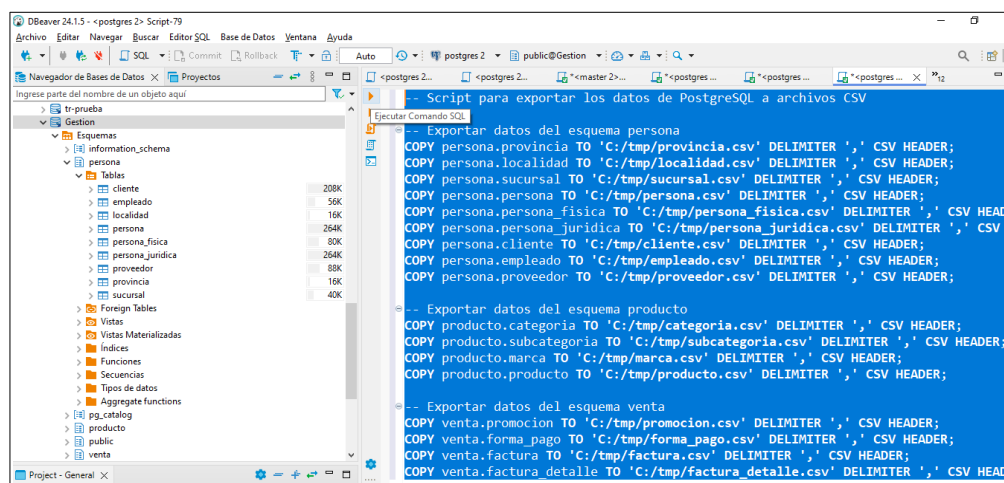
1. Script de creación de tablas (Create-table-sqlserver.sql). Permite crear las tablas en una base de datos previamente creada en SQL Server. Tiene adaptados los tipos de datos a SQL Server.
2. Script para exportar datos desde PostgreSQL (exportar-postgresql.sql). Permite ejecutar comandos necesarios para exportar los datos desde la base de datos PostgreSQL, generando en una carpeta temporal, archivos ".csv" que luego se importarán en SQL Server.
3. Script para importar datos en SQL Server: usando el comando "BULK INSERT ...FROM.." se toman los datos desde el sistema operativo y se suben a SQL Server.

Una vez descargado el archivo en la PC tendrán:

- Script para crear tablas en SQL Server: “Create-table-sqlserver.sql”
- Script para exportar datos de PostgreSQL: “exportar-postgresql.sql”
- Importar los datos en SQLServer: “Importar-datos-en-sqlserver.sql”.

2. Exportar datos de PostgreSQL:

Conectado a PostgreSQL ejecutamos el script “exportar-postgresql.sql”:



```
-- Script para exportar los datos de PostgreSQL a archivos CSV

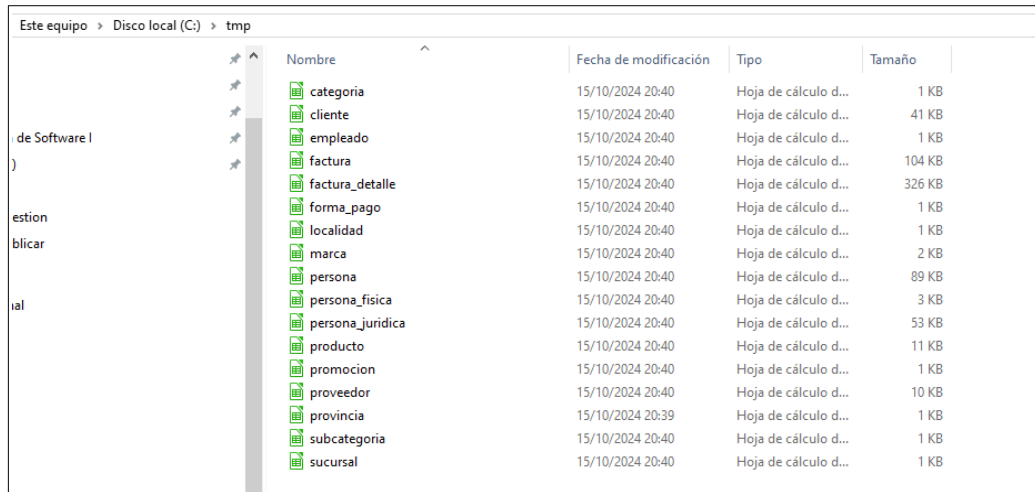
-- Exportar datos del esquema persona
COPY persona.provincia TO 'C:/tmp/provincia.csv' DELIMITER ',' CSV HEADER;
COPY persona.localidad TO 'C:/tmp/localidad.csv' DELIMITER ',' CSV HEADER;
COPY persona.sucursal TO 'C:/tmp/sucursal.csv' DELIMITER ',' CSV HEADER;
COPY persona.persona TO 'C:/tmp/persona.csv' DELIMITER ',' CSV HEADER;
COPY persona.persona_fisica TO 'C:/tmp/persona_fisica.csv' DELIMITER ',' CSV HEADER;
COPY persona.persona_juridica TO 'C:/tmp/persona_juridica.csv' DELIMITER ',' CSV HEADER;
COPY persona.cliente TO 'C:/tmp/cliente.csv' DELIMITER ',' CSV HEADER;
COPY persona.empleado TO 'C:/tmp/empleado.csv' DELIMITER ',' CSV HEADER;
COPY persona.proveedor TO 'C:/tmp/proveedor.csv' DELIMITER ',' CSV HEADER;

-- Exportar datos del esquema producto
COPY producto.categoria TO 'C:/tmp/categoria.csv' DELIMITER ',' CSV HEADER;
COPY producto.subcategoria TO 'C:/tmp/subcategoria.csv' DELIMITER ',' CSV HEADER;
COPY producto.marca TO 'C:/tmp/marca.csv' DELIMITER ',' CSV HEADER;
COPY producto.producto TO 'C:/tmp/producto.csv' DELIMITER ',' CSV HEADER;

-- Exportar datos del esquema venta
COPY venta.promocion TO 'C:/tmp/promocion.csv' DELIMITER ',' CSV HEADER;
COPY venta.forma_pago TO 'C:/tmp/forma_pago.csv' DELIMITER ',' CSV HEADER;
COPY venta.factura TO 'C:/tmp/factura.csv' DELIMITER ',' CSV HEADER;
COPY venta.factura_detalle TO 'C:/tmp/factura_detalle.csv' DELIMITER ',' CSV HEADER;
```

Lo que hace este script es utilizando la sentencia “COPY” de postgresql descargar las filas de cada una de las tablas a archivos de texto (.csv) que son dispuestos en la carpeta “C:/tmp” (asegurarse de crear esta carpeta y que esté vacía).

Si la ejecución es correcta, se encontrarán con un archivo por cada tabla en la carpeta mencionada. Son 17 archivos:

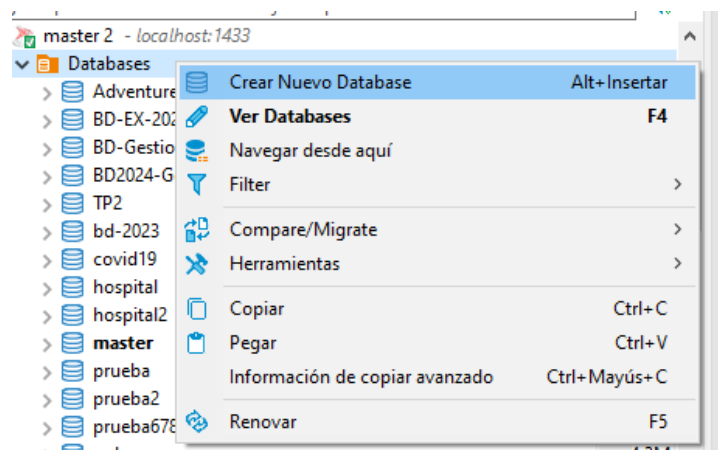


| Nombre | Fecha de modificación | Tipo | Tamaño |
|------------------|-----------------------|----------------------|--------|
| categoria | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| cliente | 15/10/2024 20:40 | Hoja de cálculo d... | 41 KB |
| empleado | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| factura | 15/10/2024 20:40 | Hoja de cálculo d... | 104 KB |
| factura_detalle | 15/10/2024 20:40 | Hoja de cálculo d... | 326 KB |
| forma_pago | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| localidad | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| marca | 15/10/2024 20:40 | Hoja de cálculo d... | 2 KB |
| persona | 15/10/2024 20:40 | Hoja de cálculo d... | 89 KB |
| persona_fisica | 15/10/2024 20:40 | Hoja de cálculo d... | 3 KB |
| persona_juridica | 15/10/2024 20:40 | Hoja de cálculo d... | 53 KB |
| producto | 15/10/2024 20:40 | Hoja de cálculo d... | 11 KB |
| promocion | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| proveedor | 15/10/2024 20:40 | Hoja de cálculo d... | 10 KB |
| provincia | 15/10/2024 20:39 | Hoja de cálculo d... | 1 KB |
| subcategoria | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |
| sucursal | 15/10/2024 20:40 | Hoja de cálculo d... | 1 KB |

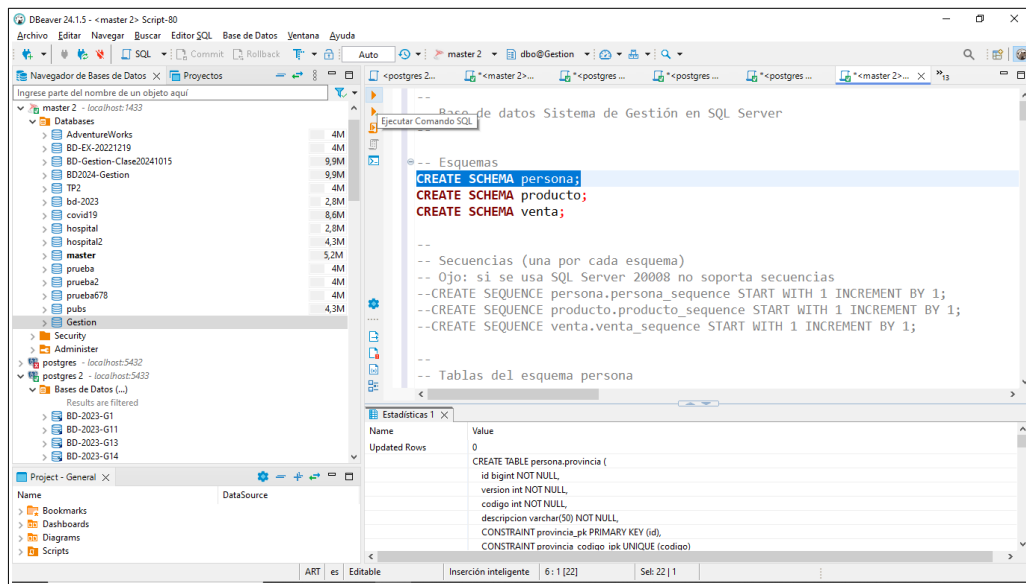
Estos archivos luego, serán importados en SQL Server.

3. Crear base de datos en SQLServer

Conectado con Dbeaver a SQLServer, crear una nueva base de datos “Gestion”:



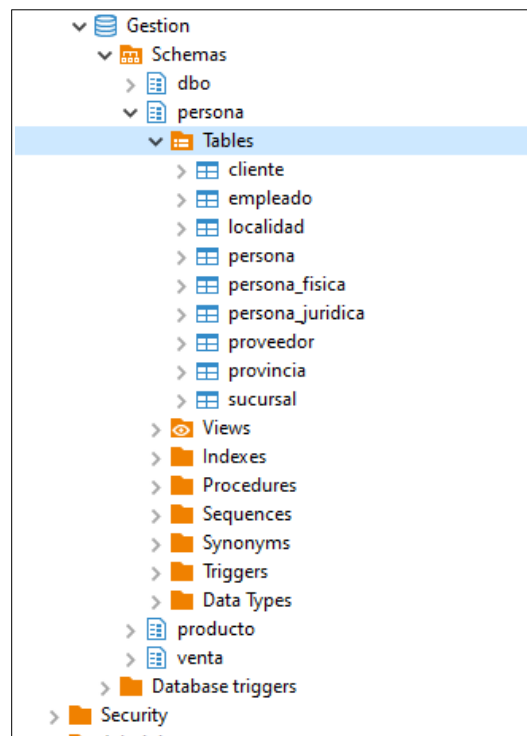
4. Una vez creada la base de datos, conectarse a la misma y ejecutar el contenido de: “Create-table-sqlserver.sql”:



Se sugiere primero ejecutar la creación de cada uno de los esquemas. Y luego, el resto de las tablas de una sola vez.

Aclaración: Comenté en el script el uso de secuencias, porque en mi caso, tengo una versión “viejita” de SQLServer. Si tienen versiones mas nuevas, no tendrán inconvenientes con las secuencias.

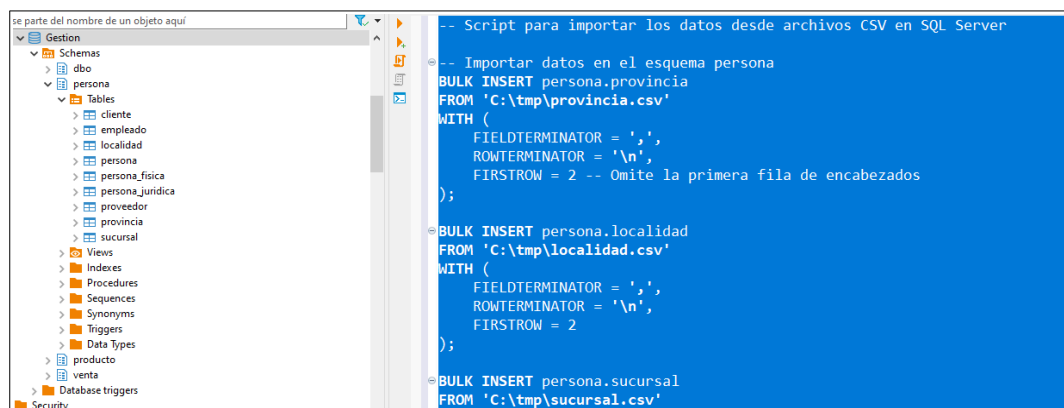
Si los scripts se ejecutan sin inconvenientes, se crearán los esquemas y sus tablas. Ubicandonos sobre “Gestion” e indicando “Renovar”(F5), nos mostrará los esquemas persona, producto y venta (y dbo propio de SQLServer):



5. Cargar datos en SQLServer

Una vez creada la base de datos y creadas las tablas, lo que falta es ejecutar el script de importación de datos en SQLServer.

Estando conectados a la BD Gestion de SQLServer, se debe ejecutar el contenido del archivo: “Importar-datos-en-sqlserver.sql”:



Este script lo que hace es ejecutar un “BULK INSERT” que toma los datos de los archivos de texto dispuestos en la carpeta: [C:\tmp](#) y los inserta en las tablas correspondientes.

Si todo se ejecuta sin inconvenientes, los datos estarán disponibles en SQLServer para ser usados. Para verificar ejecutamos el siguiente script:

```
select count(*) from venta.factura_detalle fd
```

El mismo nos informará que tabla factura_detalle tiene 10.003 filas cargadas.

Si llegamos a este punto, FELICITACIONES! Ahora podemos trabajar con las guías de Select en tanto en PostgreSQL como en SQLServer.