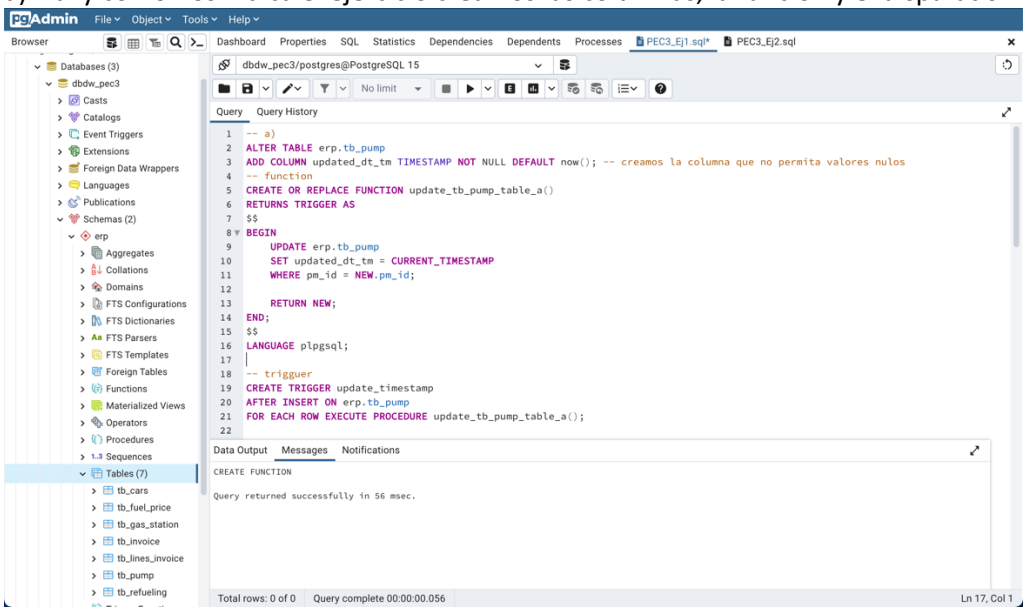
De la creación a la manipulación de una base de datos relacional

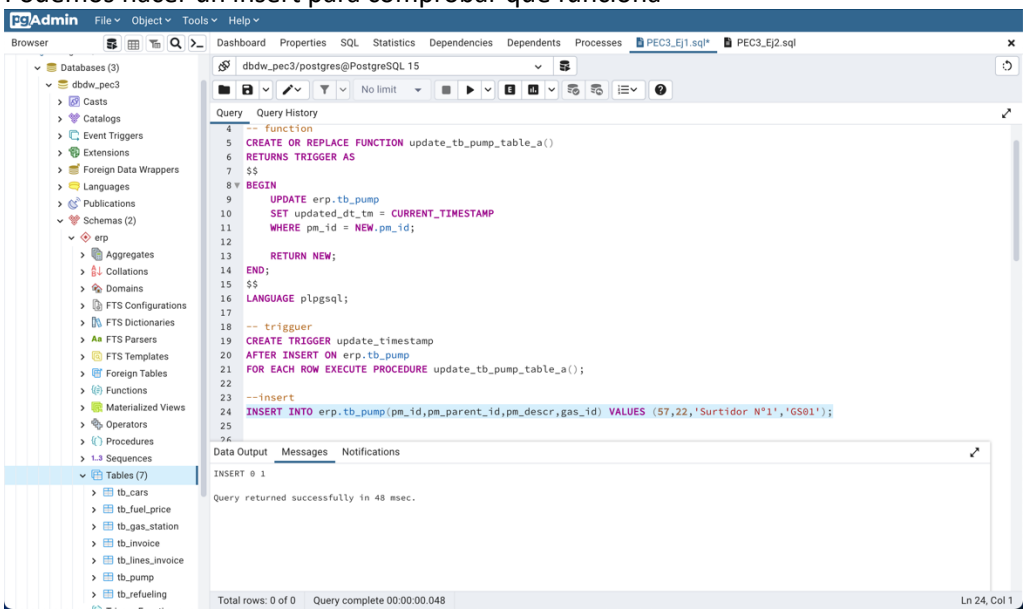
NOMBRE Y APELLIDOS: Paula Corbatón Álvarez

EJERCICIO 1 (35%)

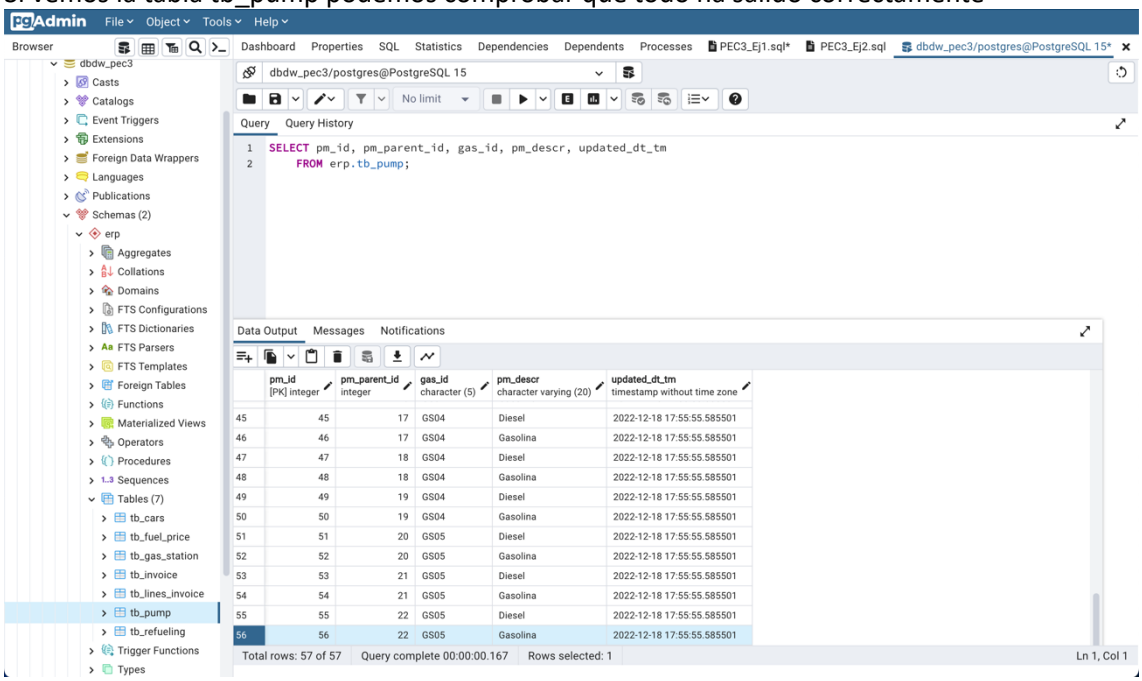
a) Tal y como nos indica el ejercicio creamos las columnas, la función y el disparador



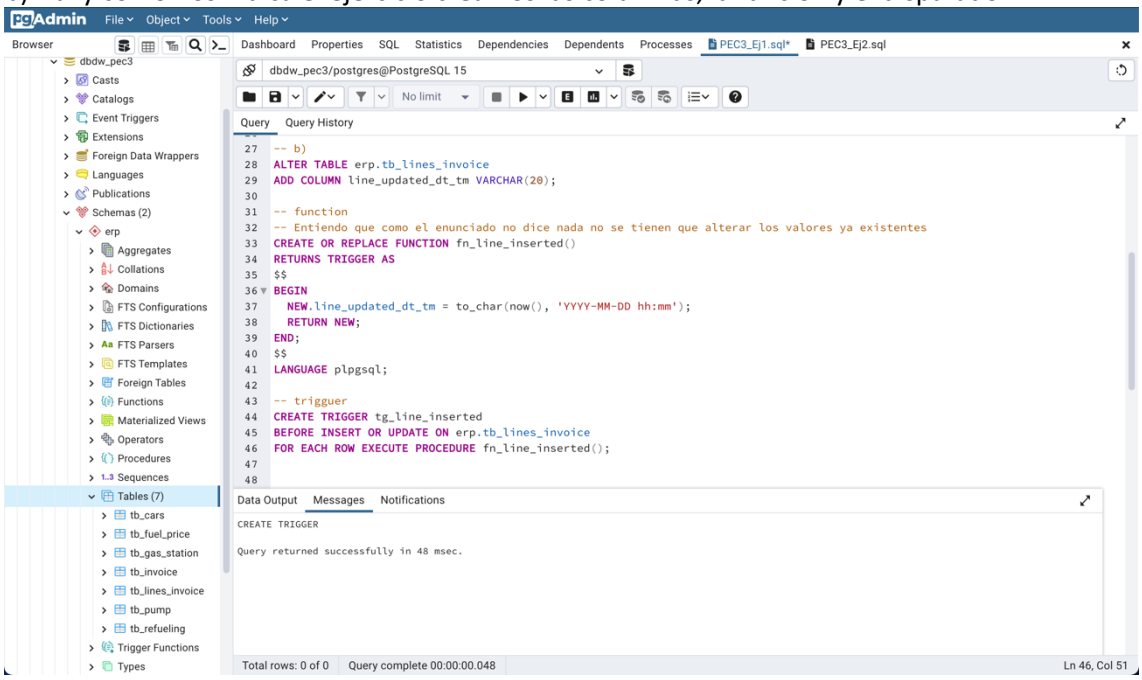
Podemos hacer un insert para comprobar que funciona



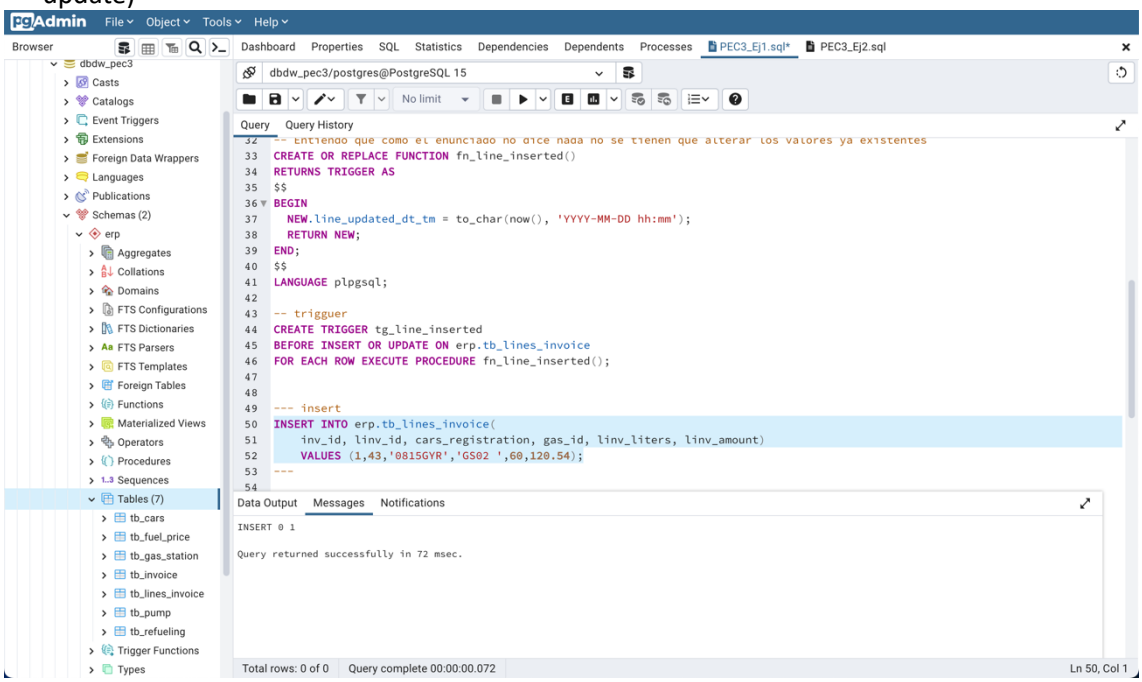
Si vemos la tabla tb_pump podemos comprobar que todo ha salido correctamente



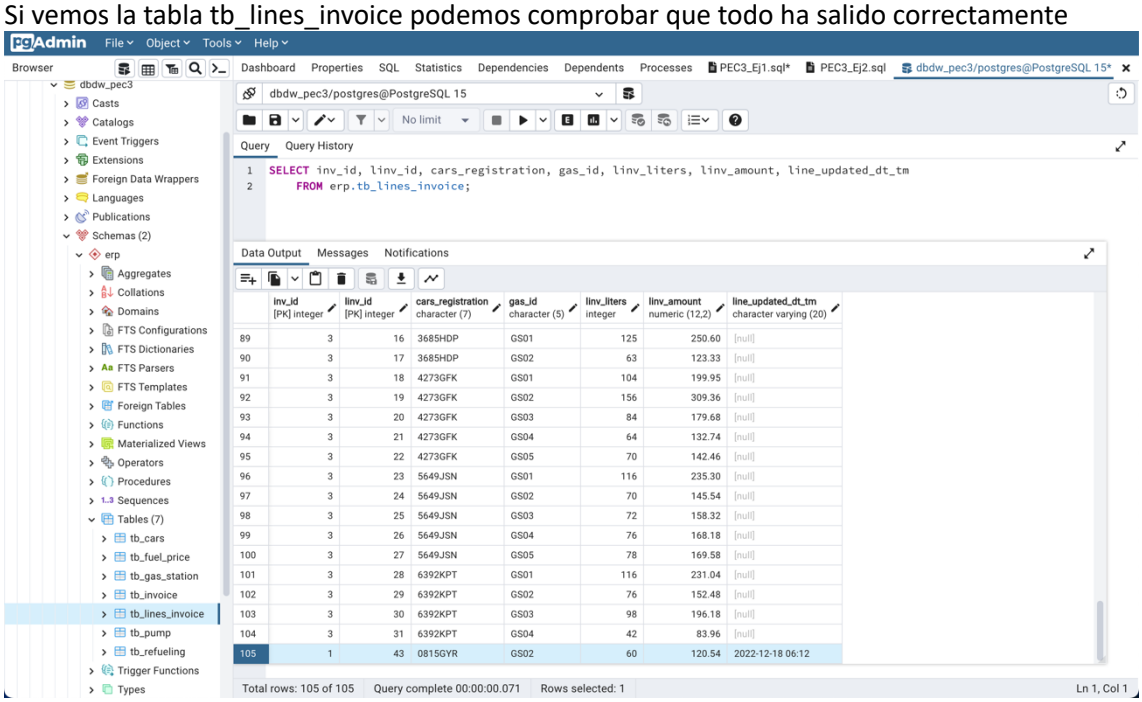
b) Tal y como nos indica el ejercicio creamos las columnas, la función y el disparador



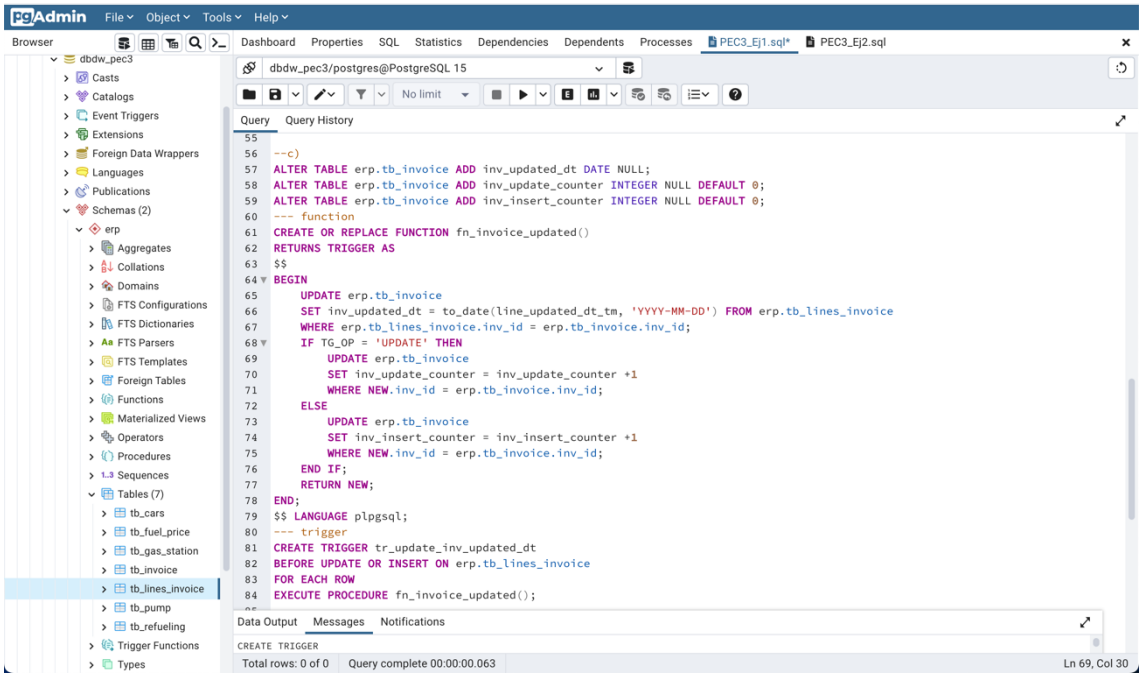
Podemos hacer un insert para comprobar que funciona (también podríamos haber hecho un update)



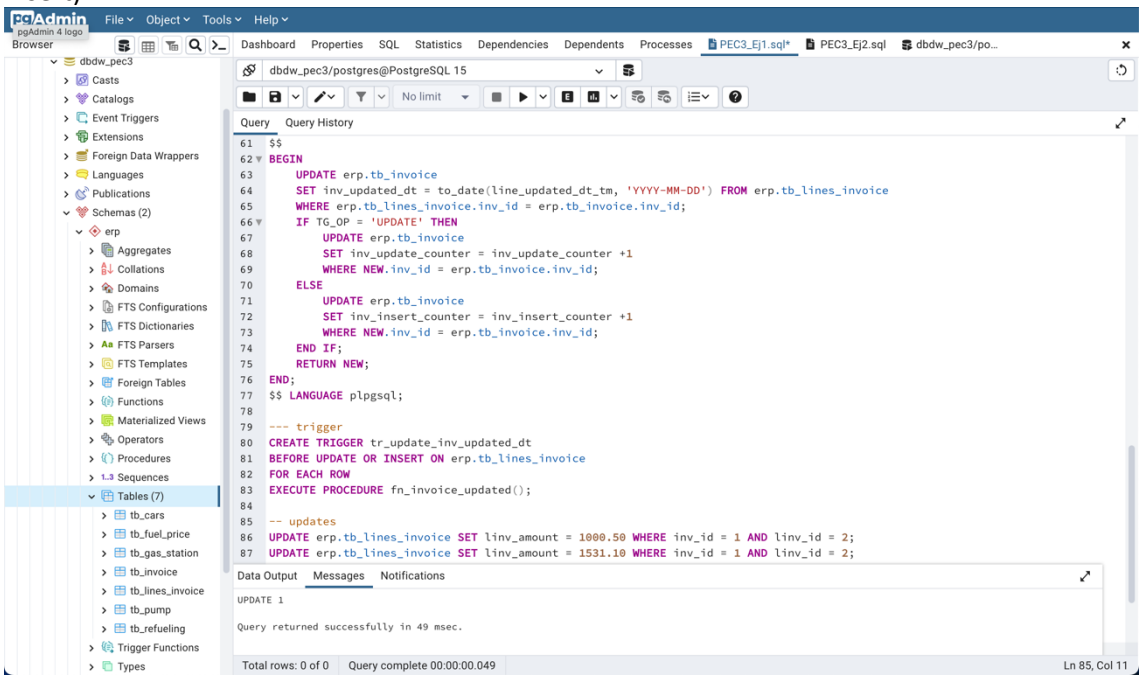
Si vemos la tabla tb_lines_invoice podemos comprobar que todo ha salido correctamente



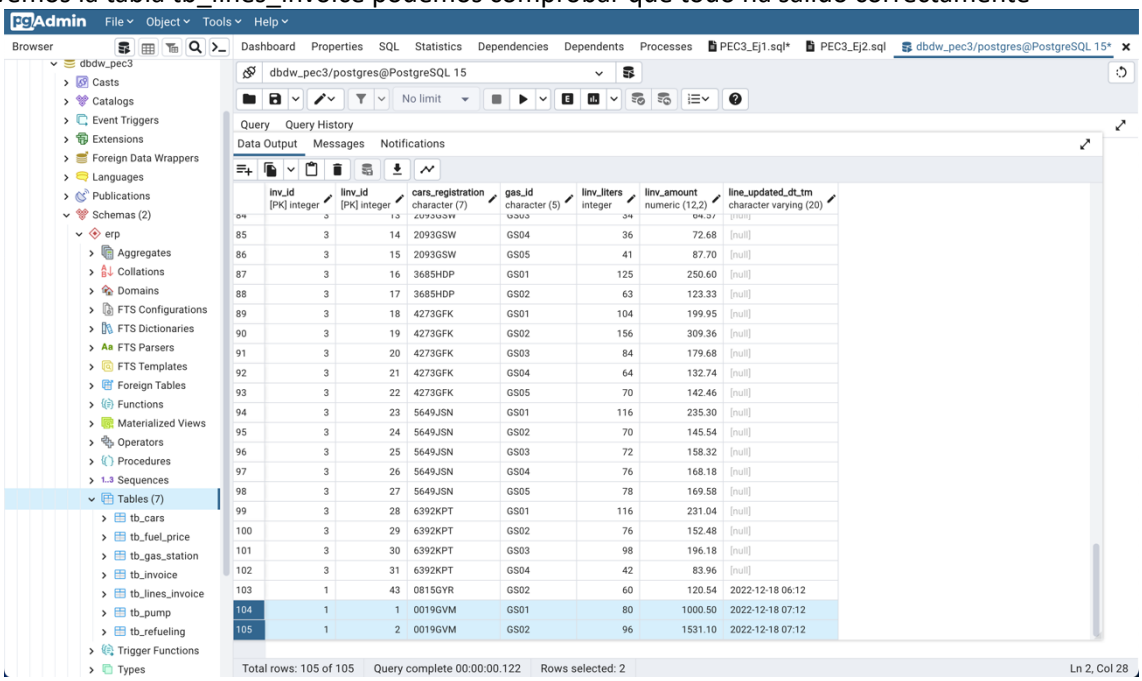
c) Tal y como nos indica el ejercicio creamos las columnas, la función y el disparador



Podemos hacer un update para comprobar que funciona (también podríamos haber hecho un insert)

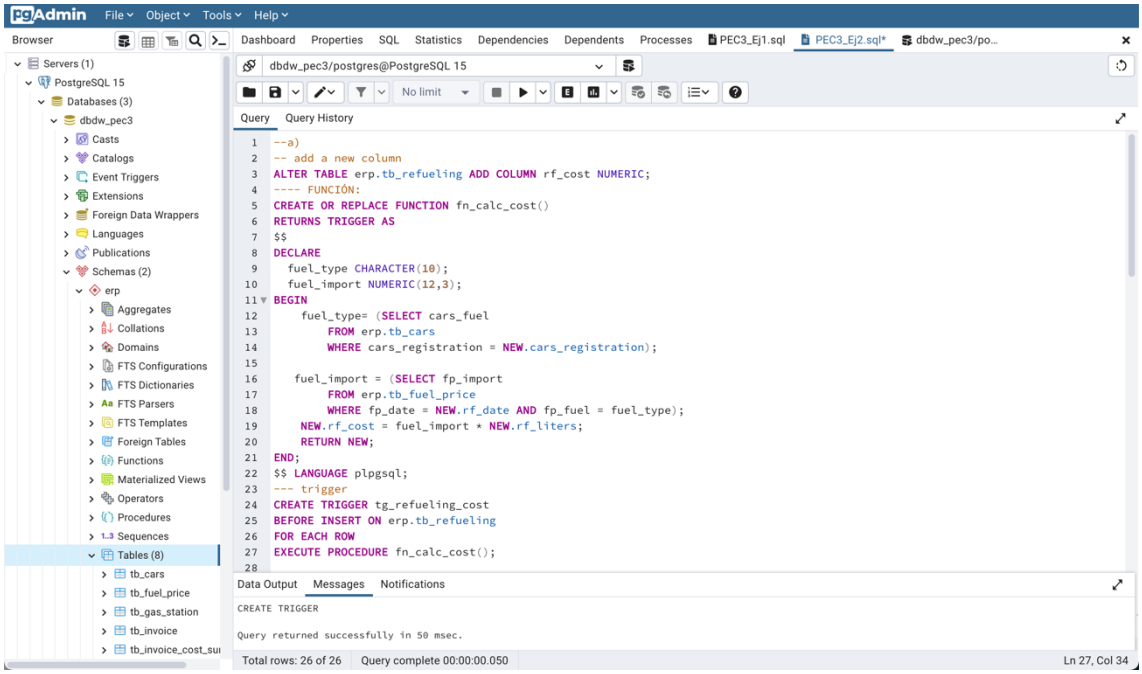


Si vemos la tabla tb_lines_invoice podemos comprobar que todo ha salido correctamente

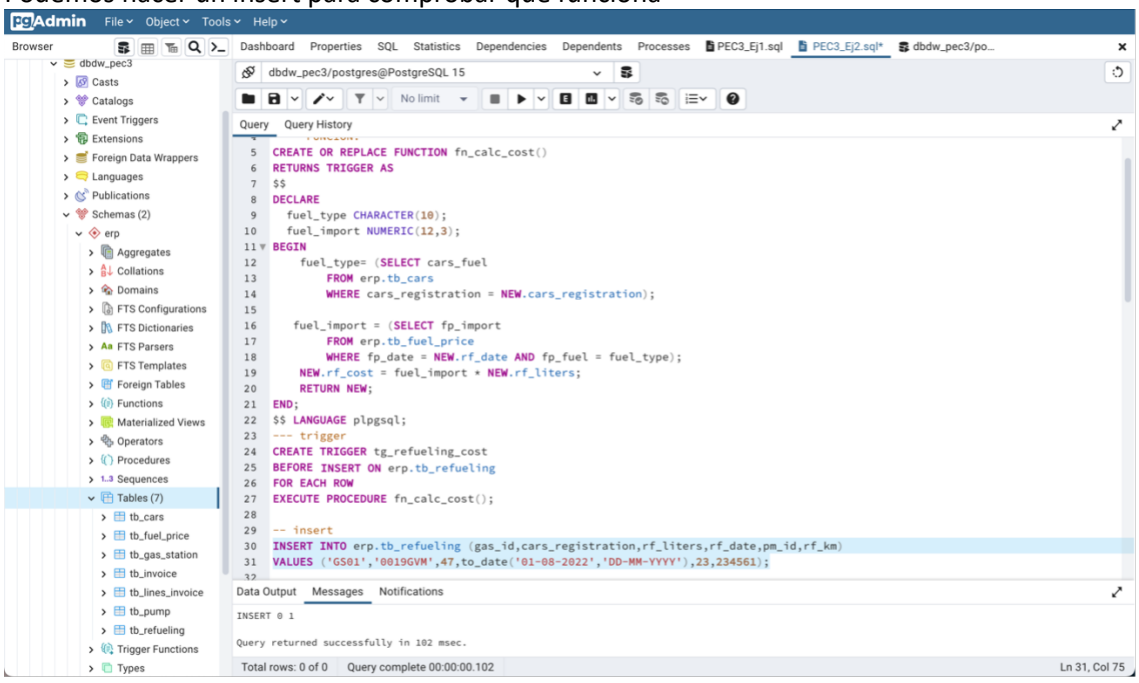


EJERCICIO 2 (55%)

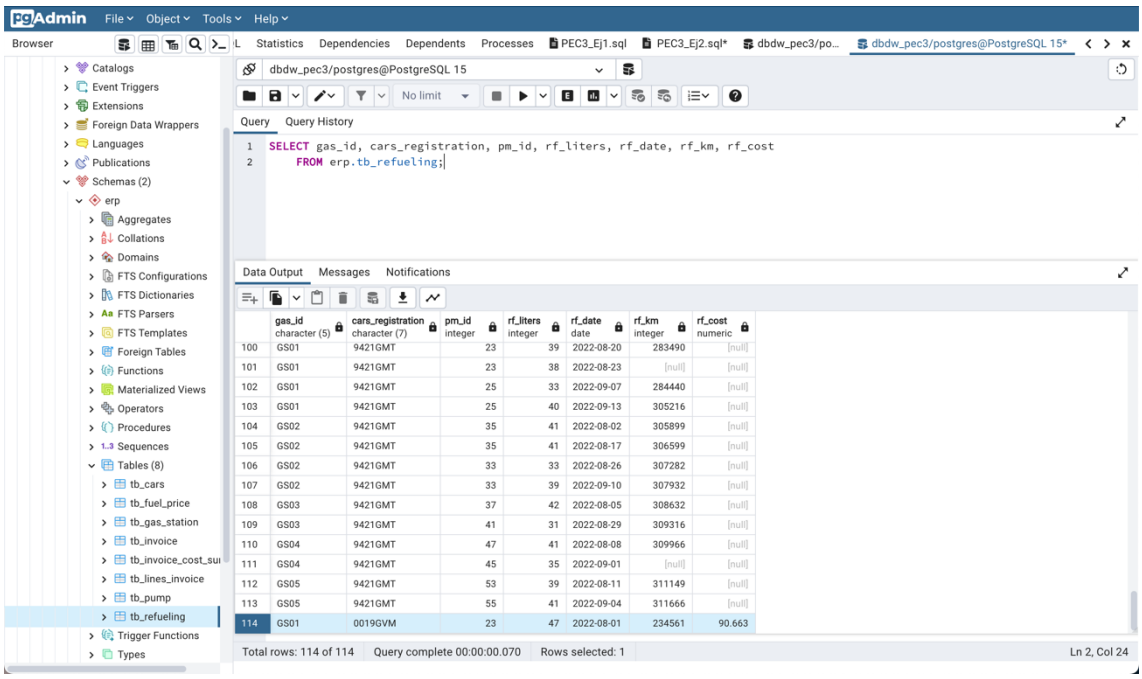
a) Tal y como nos indica el ejercicio creamos la columna, la función y el disparador



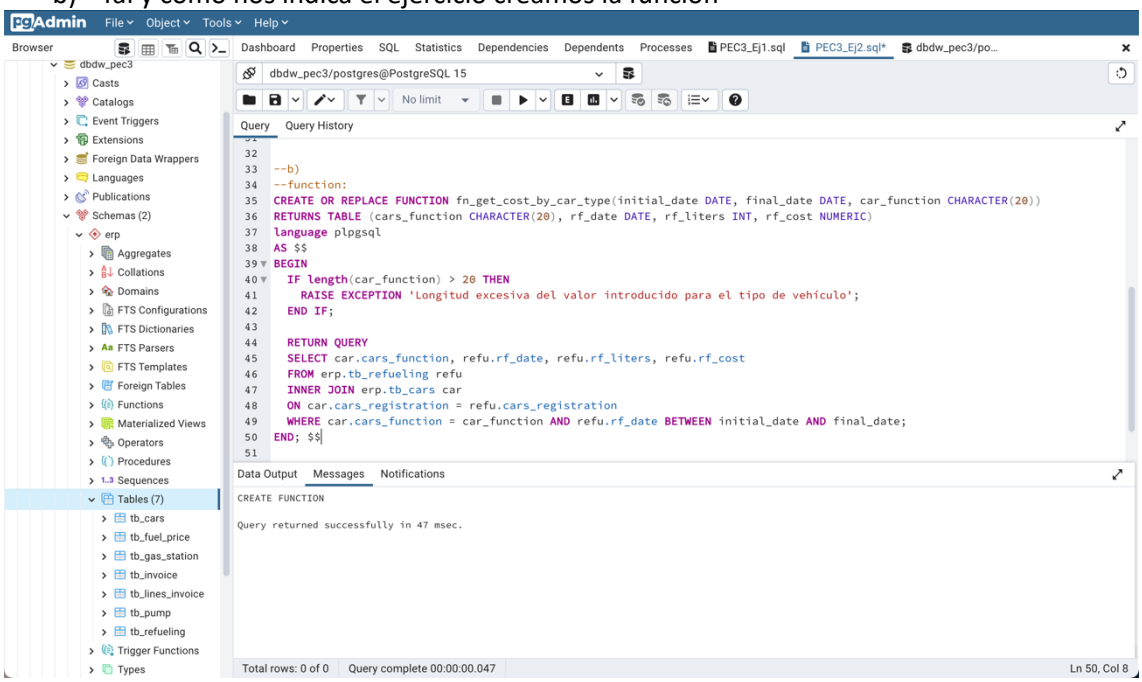
Podemos hacer un insert para comprobar que funciona



Si vemos la tabla tb_refueling podemos comprobar que todo ha salido correctamente

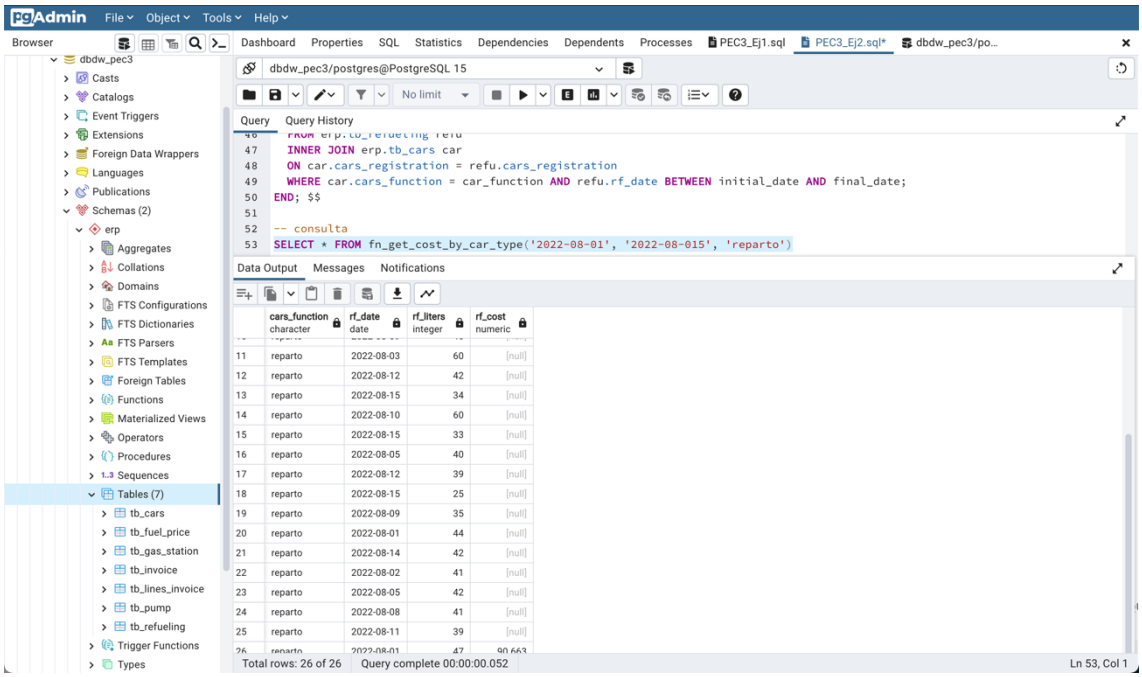


b) Tal y como nos indica el ejercicio creamos la función

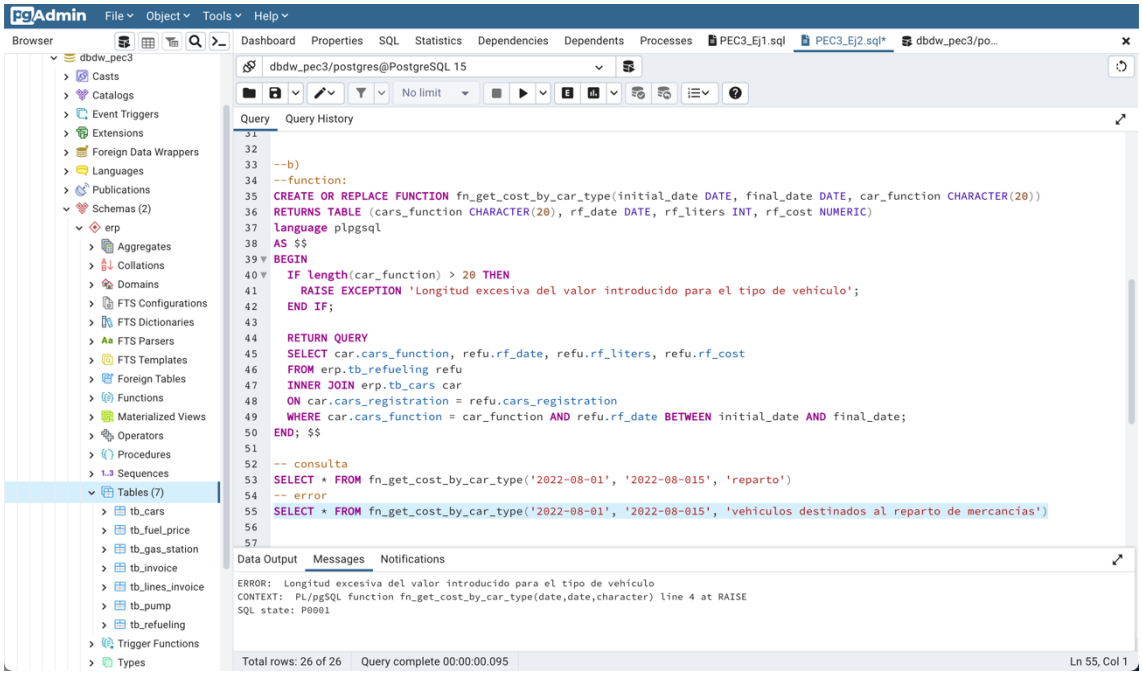


Podemos hacer una consulta buena y otra que debería dar error para comprobar que funciona:

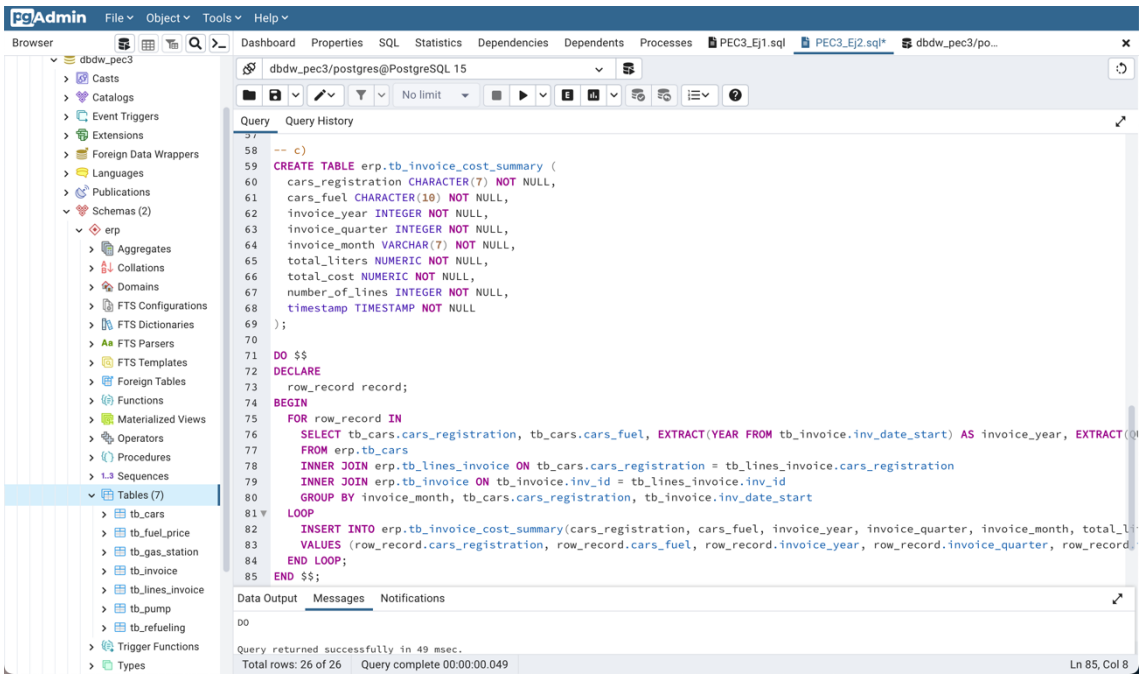
- Consulta buena:



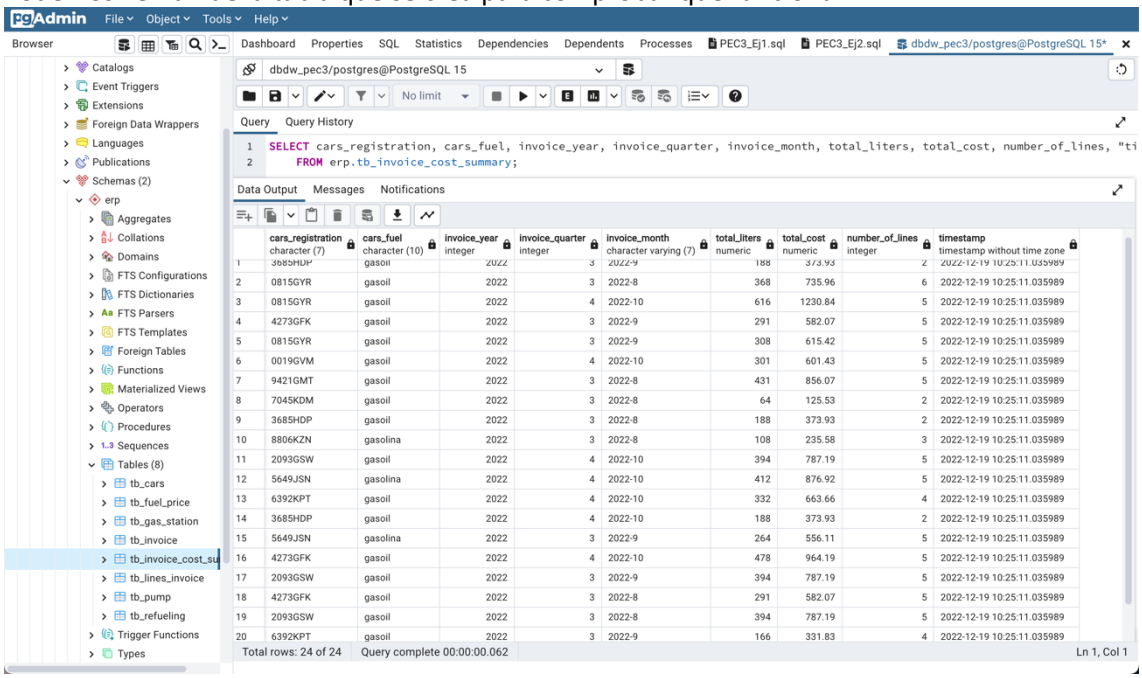
- Consulta con error:



c) Tal y como nos indica el ejercicio creamos el procedimiento y la tabla



Podemos ver la nueva tabla que se crea para comprobar que funciona

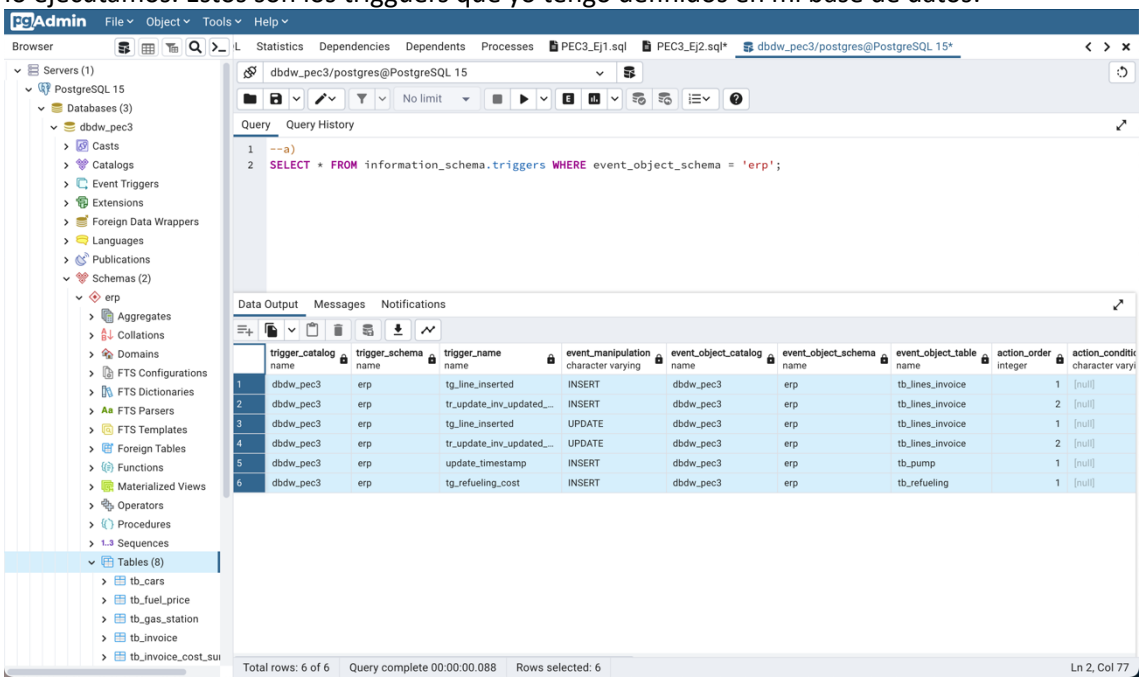


EJERCICIO 3 (10%)

a) Para saber cuántos trigguers tenemos definidos podemos utilizar la siguiente función:

```
SELECT * FROM information_schema.triggers WHERE event_object_schema = 'erp';
```

Este Código nos muestra los trigguers que tenemos definidos en el momento en el que lo ejecutamos. Estos son los trigguers que yo tengo definidos en mi base de datos:



b)

En PostgreSQL, una tabla interna es un tipo especial de tabla que no se almacena en el disco y no es accesible para los usuarios. Son utilizados internamente por el servidor PostgreSQL para almacenar datos temporales o resultados intermedios.

Una vista, por otro lado, es una tabla virtual que está definida por una instrucción SELECT. Una vista no almacena datos en sí misma, sino que recupera datos de una o más tablas o vistas subyacentes cuando se consulta. Las vistas se utilizan a menudo para simplificar consultas complejas dividiéndolas en partes más pequeñas y manejables, o para proporcionar una vista personalizada de los datos para diferentes usuarios.

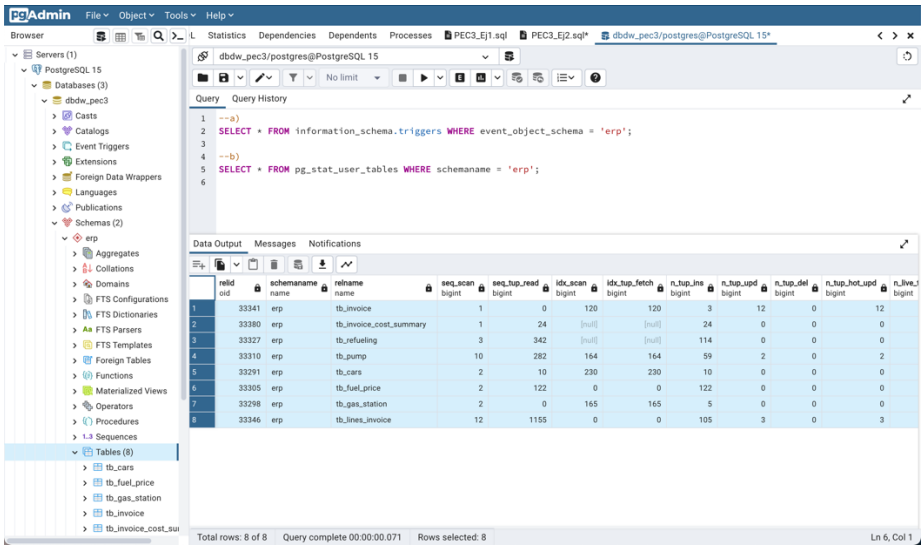
Las tablas internas se utilizan para el almacenamiento temporal y el procesamiento de datos, mientras que las vistas se utilizan para presentar los datos a los usuarios de forma personalizada o simplificada.

A continuación detallo un par de ejemplos de tablas y consultas que pueden ser útiles cuando se trabaja con el “Statistics Collector”:

1. Para ver información estadística sobre las tablas en una base de datos específica, incluida la cantidad de filas en cada tabla y la cantidad de espacio en disco utilizado por cada tabla, se puede usar la siguiente consulta:

```
SELECT * FROM pg_stat_user_tables WHERE schemaname = 'nombre_esquema';
```

En mi base de datos se vería así:



2. Para ver la actividad actual en el sistema de la base de datos, incluido el PID (ID de proceso) de cada proceso backend activo, el usuario con el que está conectado y la consulta actual que ejecuta cada proceso, se puede usar la siguiente consulta:

```
SELECT * FROM pg_stat_activity;
```

En mi base de datos se vería así: