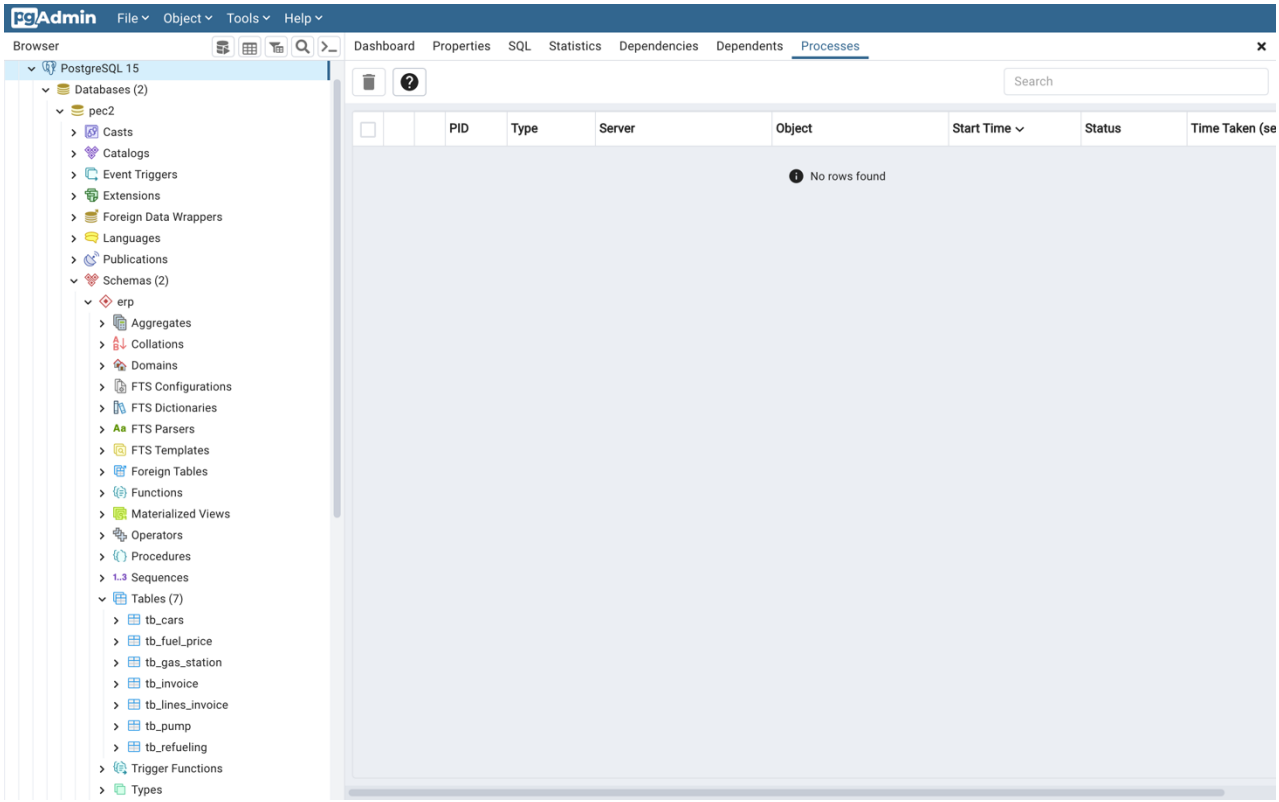
De la creación a la manipulación de una base de datos relacional

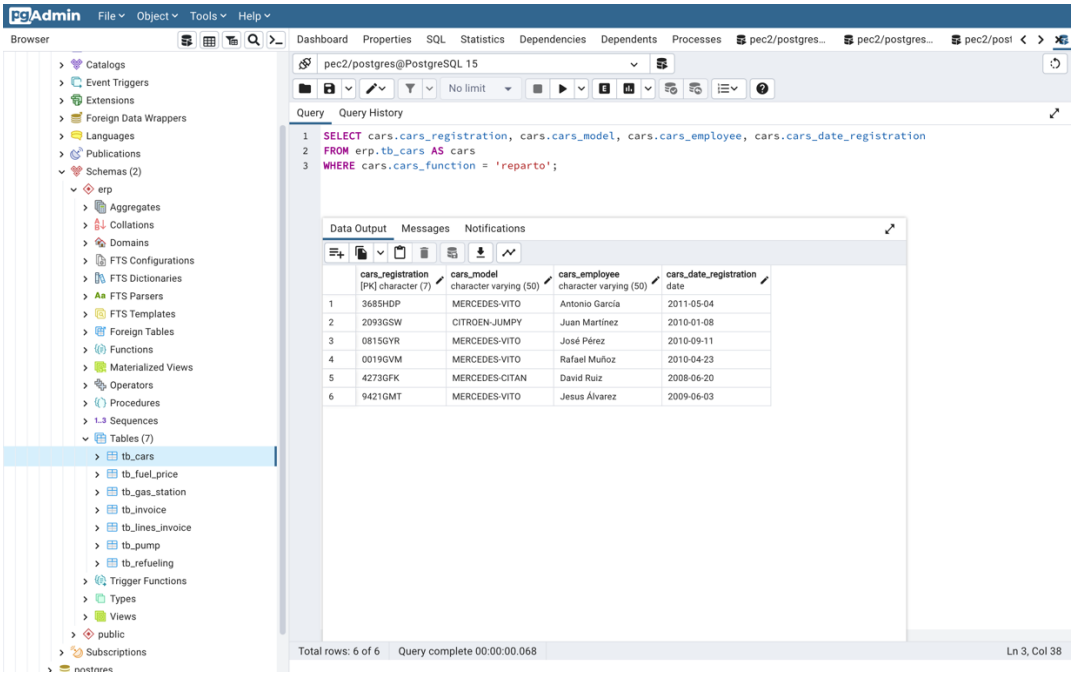
NOMBRE Y APELLIDOS: Paula Corbatón Álvarez

EJERCICIO 1 (25%)

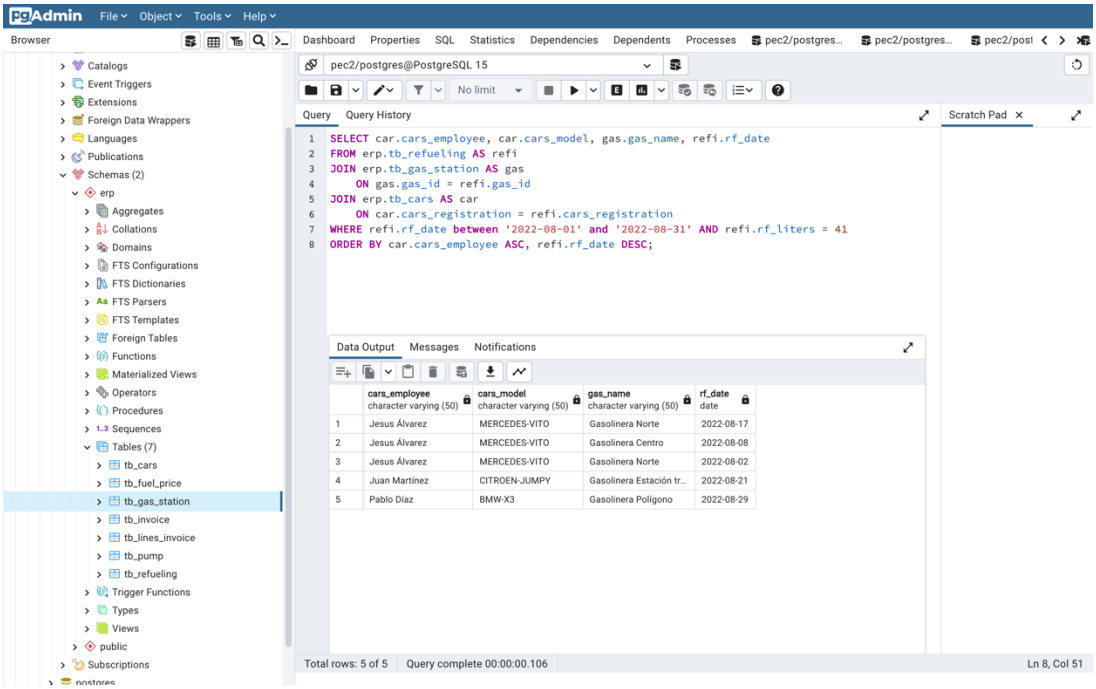


EJERCICIO 2 (25%)

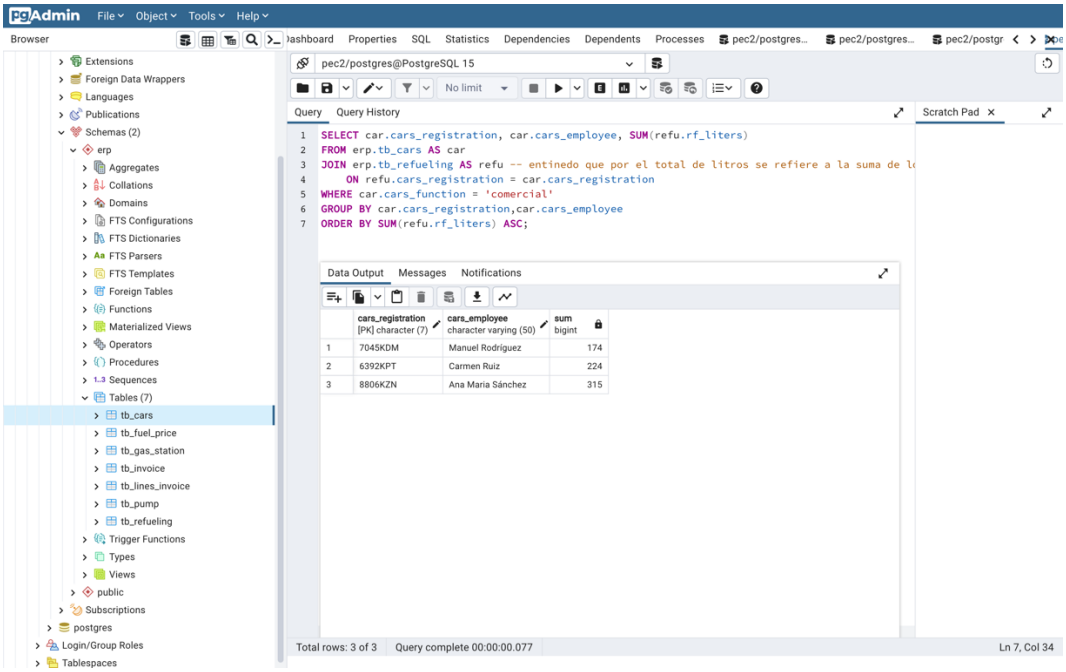
a)



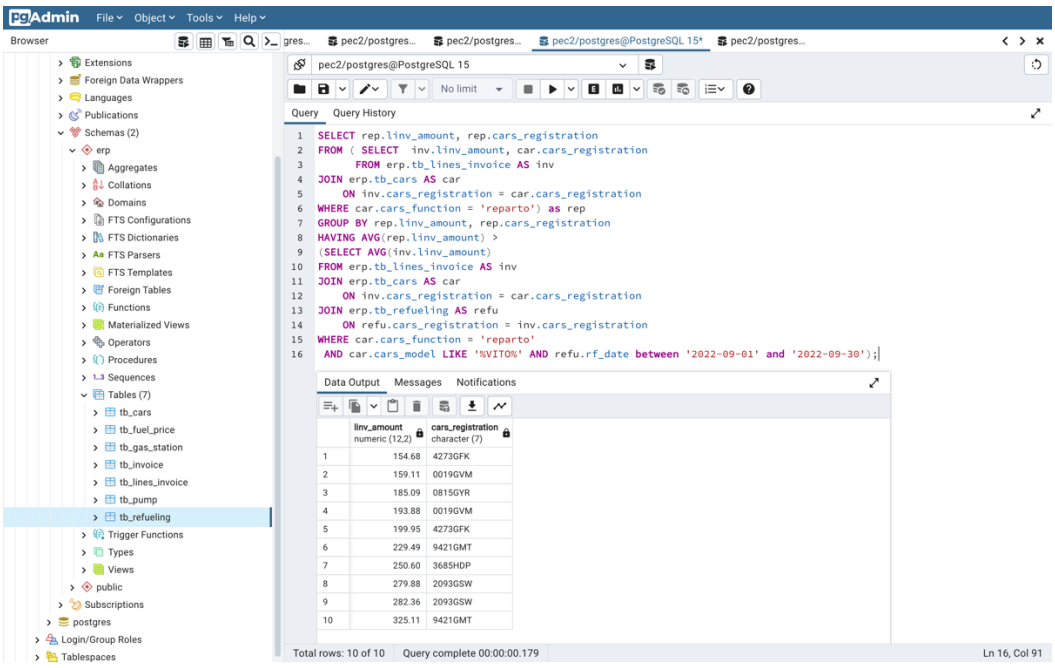
b)



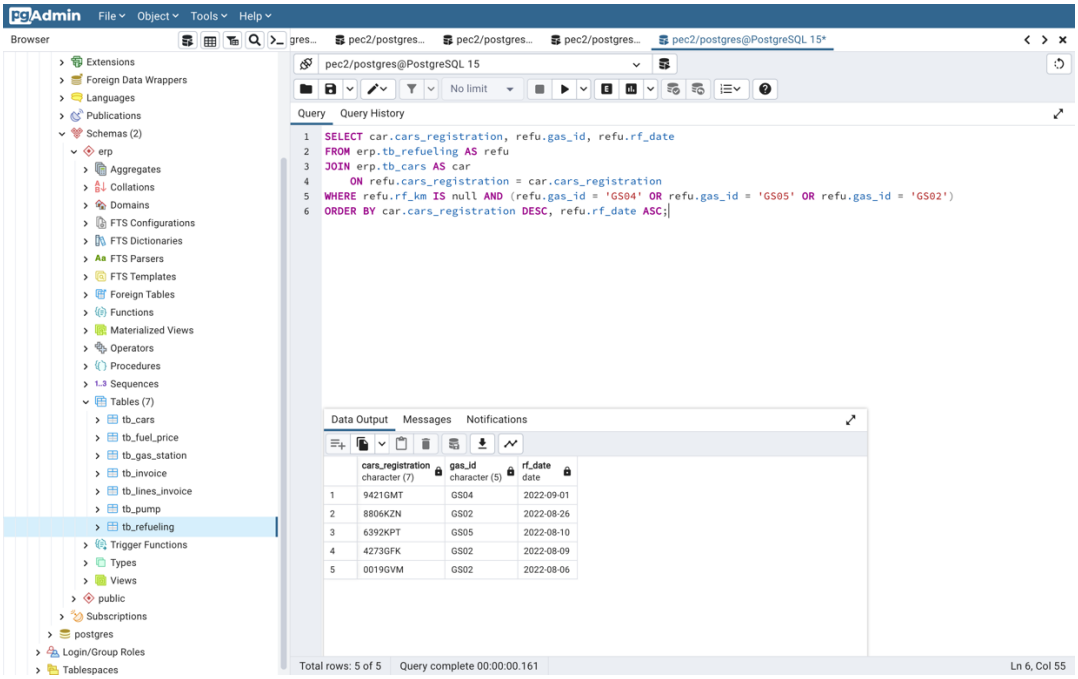
c)


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents Processes
pec2/postgres@PostgreSQL 15
No limit
Query Query History
Scratch Pad
SELECT car.cars_registration, car.cars_employee, SUM(refu.rf_liters)
FROM erp.tb_cars AS car
JOIN erp.tb_refueling AS refu -- entinedo que por el total de litros se refiere a la suma de l
ON refu.cars_registration = car.cars_registration
WHERE car.cars_function = 'comercial'
GROUP BY car.cars_registration,car.cars_employee
ORDER BY SUM(refu.rf_liters) ASC;
Data Output Messages Notifications
cars_registration [PK] character (7)
cars_employee character varying (50)
sum bigint
1 7045KDM Manuel Rodríguez 174
2 6392KPT Carmen Ruiz 224
3 8806KZN Ana María Sánchez 315
Total rows: 3 of 3
Query complete 00:00:00.077
Ln 7, Col 34

d)

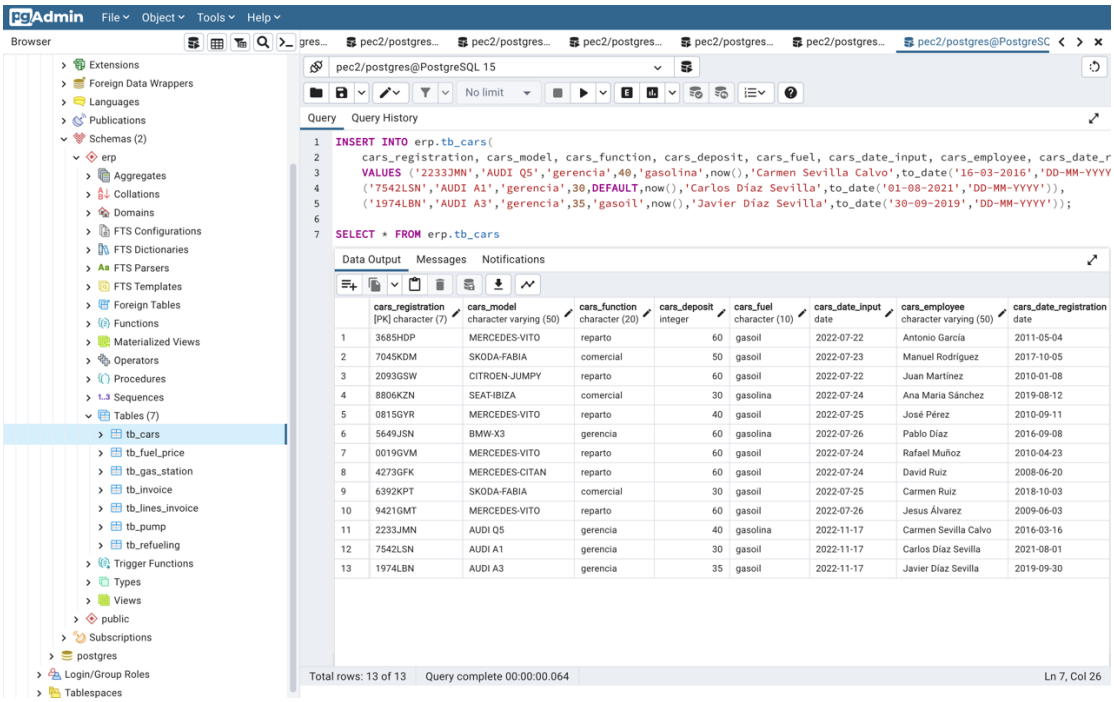

pgAdmin
File Object Tools Help
Browser
pec2/postgres@PostgreSQL 15
No limit
Query Query History
SELECT rep.linv_amount, rep.cars_registration
FROM (SELECT inv.linv_amount, car.cars_registration
FROM erp.tb_lines_invoice AS inv
JOIN erp.tb_cars AS car
ON inv.cars_registration = car.cars_registration
WHERE car.cars_function = 'reparto') as rep
GROUP BY rep.linv_amount, rep.cars_registration
HAVING AVG(rep.linv_amount) >
(SELECT AVG(inv.linv_amount)
FROM erp.tb_lines_invoice AS inv
JOIN erp.tb_cars AS car
ON inv.cars_registration = car.cars_registration
JOIN erp.tb_refueling AS refu
ON refu.cars_registration = inv.cars_registration
WHERE car.cars_function = 'reparto'
AND car.cars_model LIKE '%VITO%' AND refu.rf_date between '2022-09-01' and '2022-09-30');
Data Output Messages Notifications
linv_amount numeric (12,2)
cars_registration character (7)
1 154.68 4273GFK
2 159.11 0019GVM
3 185.09 0815GYR
4 193.88 0019GVM
5 199.95 4273GFK
6 229.49 9421GMT
7 250.60 3685HDP
8 279.88 2093GSW
9 282.36 2093GSW
10 325.11 9421GMT
Total rows: 10 of 10
Query complete 00:00:00.179
Ln 16, Col 91

e)

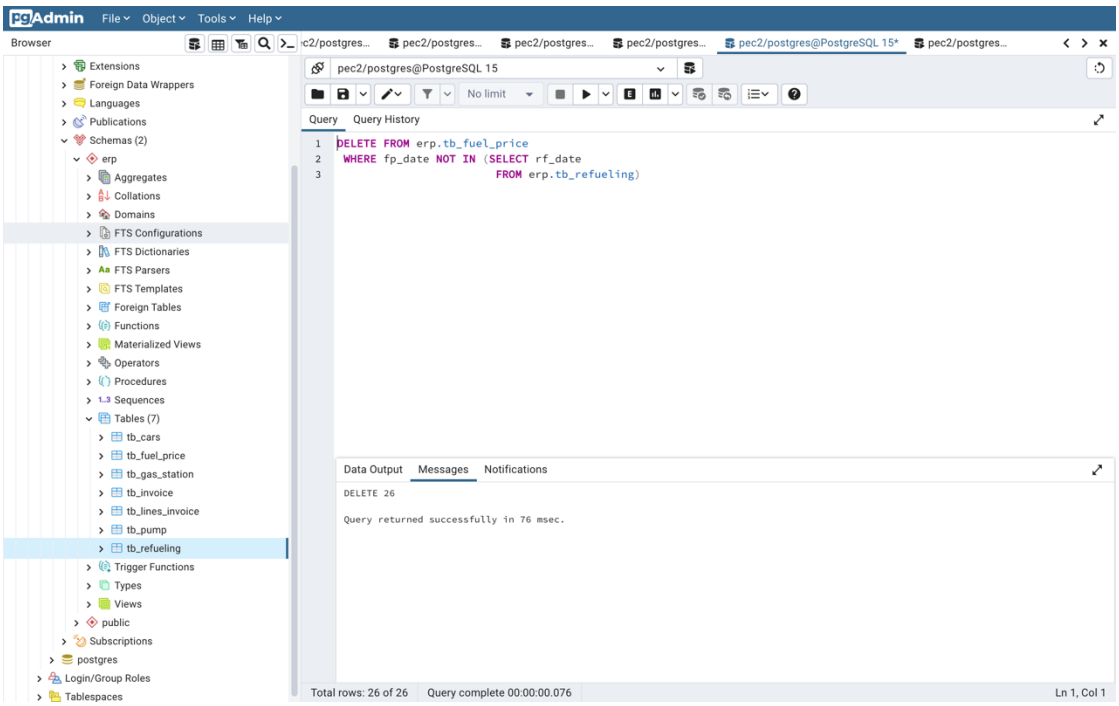

pgAdmin
File
Object
Tools
Help
Browser
pec2/postgres@PostgreSQL 15*
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (2)
erp
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (7)
tb_cars
tb_fuel_price
tb_gas_station
tb_invoice
tb_lines_invoice
tb_pump
tb_refueling
Trigger Functions
Types
Views
public
Subscriptions
postgres
Login/Group Roles
Tablespaces
pec2/postgres@PostgreSQL 15
No limit
Query
Query History
SELECT car.cars_registration, refu.gas_id, refu.rf_date
FROM erp.tb_refueling AS refu
JOIN erp.tb_cars AS car
ON refu.cars_registration = car.cars_registration
WHERE refu.rf_km IS null AND (refu.gas_id = 'GS04' OR refu.gas_id = 'GS05' OR refu.gas_id = 'GS02')
ORDER BY car.cars_registration DESC, refu.rf_date ASC;
Data Output
Messages
Notifications
cars_registration character (7)
gas_id character (5)
rf_date date
1 9421GMT GS04 2022-09-01
2 8806KZN GS02 2022-08-26
3 6392KPT GS05 2022-08-10
4 4273GFK GS02 2022-08-09
5 0019GVM GS02 2022-08-06
Total rows: 5 of 5
Query complete 00:00:00.161
Ln 6, Col 55

EJERCICIO 3 (25%)

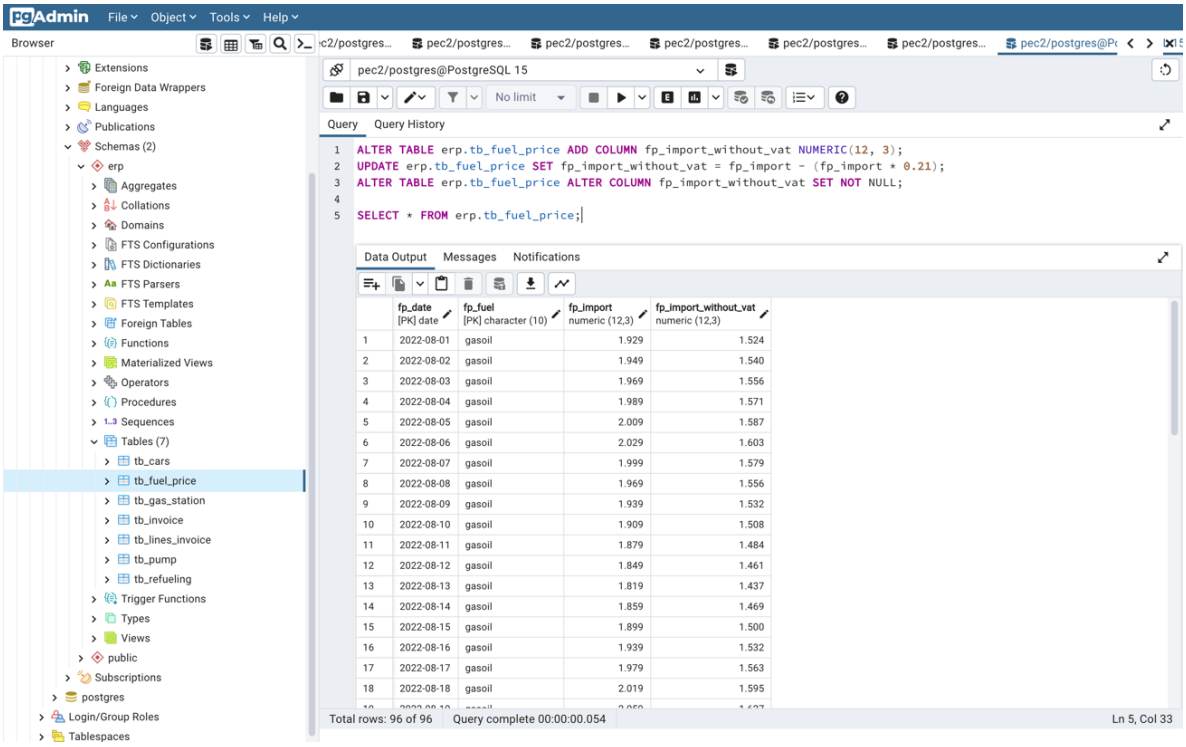
a)



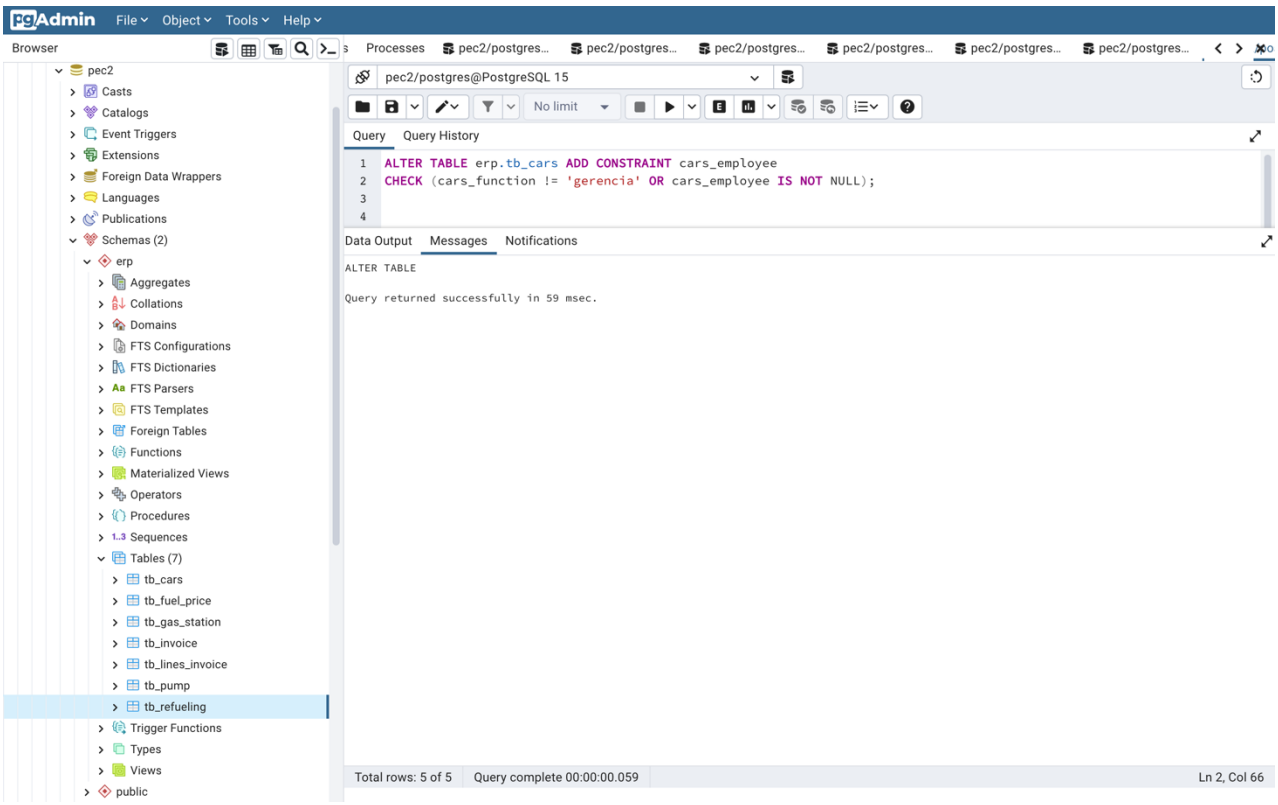
b)



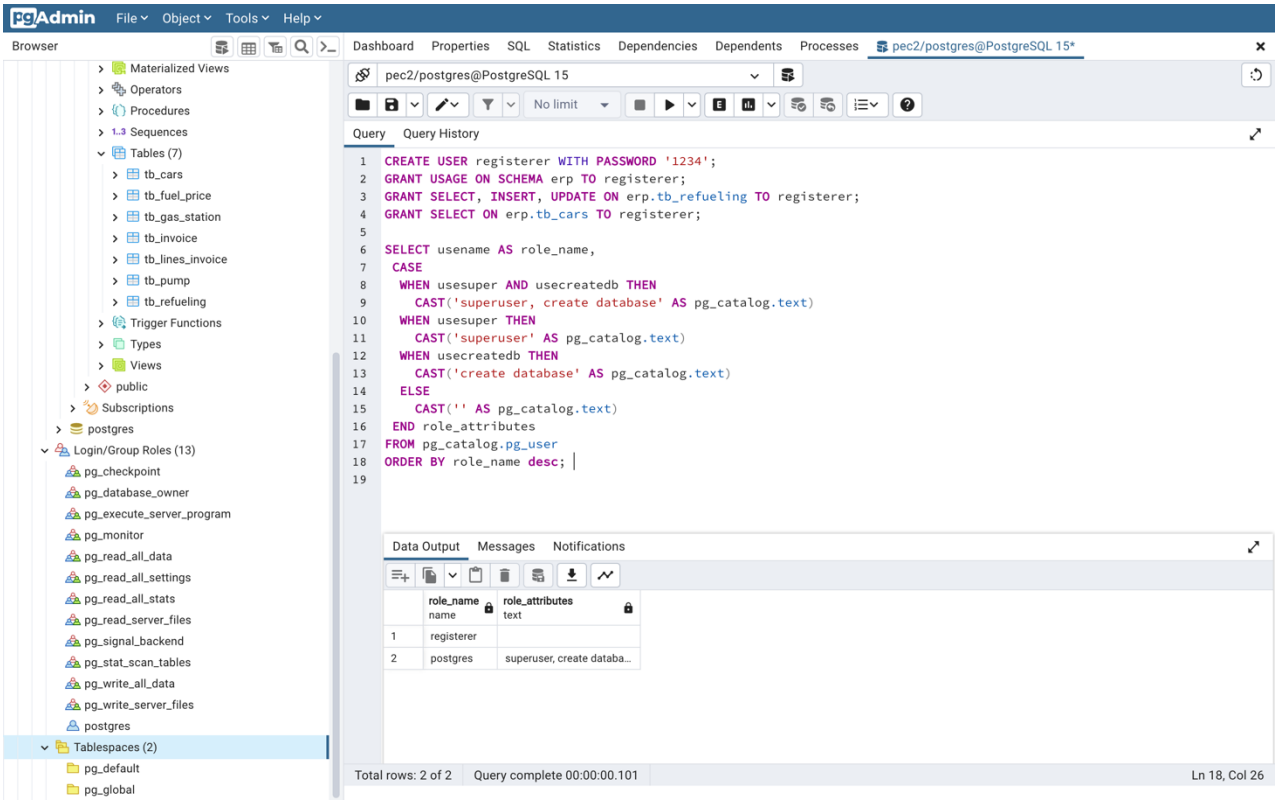
c)


pgAdmin
File Object Tools Help
Browser
pec2/postgres@PostgreSQL 15
No limit
Query Query History
ALTER TABLE erp.tb_fuel_price ADD COLUMN fp_import_without_vat NUMERIC(12, 3);
UPDATE erp.tb_fuel_price SET fp_import_without_vat = fp_import - (fp_import * 0.21);
ALTER TABLE erp.tb_fuel_price ALTER COLUMN fp_import_without_vat SET NOT NULL;
SELECT * FROM erp.tb_fuel_price;
Data Output Messages Notifications
fp_date [PK] date
fp_fuel [PK] character (10)
fp_import numeric (12,3)
fp_import_without_vat numeric (12,3)
1 2022-08-01 gasoil 1.929 1.524
2 2022-08-02 gasoil 1.949 1.540
3 2022-08-03 gasoil 1.969 1.556
4 2022-08-04 gasoil 1.989 1.571
5 2022-08-05 gasoil 2.009 1.587
6 2022-08-06 gasoil 2.029 1.603
7 2022-08-07 gasoil 1.999 1.579
8 2022-08-08 gasoil 1.969 1.556
9 2022-08-09 gasoil 1.939 1.532
10 2022-08-10 gasoil 1.909 1.508
11 2022-08-11 gasoil 1.879 1.484
12 2022-08-12 gasoil 1.849 1.461
13 2022-08-13 gasoil 1.819 1.437
14 2022-08-14 gasoil 1.859 1.469
15 2022-08-15 gasoil 1.899 1.500
16 2022-08-16 gasoil 1.939 1.532
17 2022-08-17 gasoil 1.979 1.563
18 2022-08-18 gasoil 2.019 1.595
Total rows: 96 of 96
Query complete 00:00:00.054
Ln 5, Col 33

d)


pgAdmin
File Object Tools Help
Browser
pec2/postgres@PostgreSQL 15
No limit
Query Query History
ALTER TABLE erp.tb_cars ADD CONSTRAINT cars_employee
CHECK (cars_function != 'gerencia' OR cars_employee IS NOT NULL);
Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 59 msec.
Total rows: 5 of 5
Query complete 00:00:00.059
Ln 2, Col 66

e)


pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
pec2/postgres@PostgreSQL 15*
pec2/postgres@PostgreSQL 15
No limit
Query
Query History
Materialized Views
Operators
Procedures
Sequences
Tables (7)
tb_cars
tb_fuel_price
tb_gas_station
tb_invoice
tb_lines_invoice
tb_pump
tb_refueling
Trigger Functions
Types
Views
public
Subscriptions
postgres
Login/Group Roles (13)
pg_checkpoint
pg_database_owner
pg_execute_server_program
pg_monitor
pg_read_all_data
pg_read_all_settings
pg_read_all_stats
pg_read_server_files
pg_signal_backend
pg_stat_scan_tables
pg_write_all_data
pg_write_server_files
postgres
Tablespaces (2)
pg_default
pg_global
CREATE USER registerer WITH PASSWORD '1234';
GRANT USAGE ON SCHEMA erp TO registerer;
GRANT SELECT, INSERT, UPDATE ON erp.tb_refueling TO registerer;
GRANT SELECT ON erp.tb_cars TO registerer;
SELECT usename AS role_name,
CASE
WHEN usesuper AND usecreatedb THEN
CAST('superuser, create database' AS pg_catalog.text)
WHEN usesuper THEN
CAST('superuser' AS pg_catalog.text)
WHEN usecreatedb THEN
CAST('create database' AS pg_catalog.text)
ELSE
CAST('' AS pg_catalog.text)
END role_attributes
FROM pg_catalog.pg_user
ORDER BY role_name desc;
Data Output
Messages
Notifications
role_name
name
role_attributes
text
1
registerer
2
postgres
superuser, create databa...
Total rows: 2 of 2
Query complete 00:00:00.101
Ln 18, Col 26

EJERCICIO 4 (25%)

1)

a) Inner Join: devuelve únicamente las filas en las que el valor del campo de la primera tabla coincide con el valor del campo correspondiente en la segunda tabla.

```
SELECT *
FROM Work_team_a AS a
INNER JOIN Work_team_b AS b
ON a.employee _a = b.employee_b ;
```

Inner join of 2 tables			
a_id	employee_a	b_id	employee_b
1	Andrea Mendoza	3	Andrea Mendoza
4	Carlos Alvarado	2	Carlos Alvarado

b) Left outer Join: devuelve todas las filas de la tabla de la izquierda (hagan o no match) y todas las filas de la tabla de la derecha que hagan match.

```
SELECT *
FROM Work_team_a AS a
LEFT OUTER JOIN Work_team_b AS b
ON a.employee _a = b.employee_b ;
```

Left outer join of 2 tables			
a_id	employee_a	b_id	employee_b
1	Andrea Mendoza	3	Andrea Mendoza
2	Martin Lara	NULL	NULL
3	Lucas Rojas	NULL	NULL
4	Carlos Alvarado	2	Carlos Alvarado

c) Right outer Join: devuelve todas las filas todas las filas de la tabla de la derecha que hagan match y todas las filas de la tabla de la izquierda (hagan o no match)

```
SELECT *
FROM Work_team_a AS a
RIGHT OUTER JOIN Work_team_b AS b
ON a.employee _a = b.employee_b;
```

Right outer join of 2 tables			
a_id	employee_a	b_id	employee_b
NULL	NULL	1	Claudia Camacho
4	Carlos Alvarado	2	Carlos Alvarado
1	Andrea Mendoza	3	Andrea Mendoza
NULL	NULL	4	Pedro Lombardi

d) Full outer join: devuelve todas las filas, tanto las de la tabla de la izquierda como las de la tabla de la derecha (hagan o no match)

```
SELECT *
FROM Work_team_a AS a
FULL OUTER JOIN Work_team_b AS b
ON a.employee _a = b.employee_b;
```

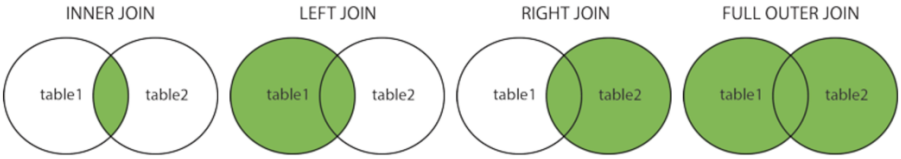
Full outer join of 2 tables			
a_id	employee_a	b_id	employee_b
1	Andrea Mendoza	3	Andrea Mendoza
2	Martin Lara	NULL	NULL
3	Lucas Rojas	NULL	NULL
4	Carlos Alvarado	2	Carlos Alvarado
NULL	NULL	1	Claudia Camacho
NULL	NULL	4	Pedro Lombardi
1	Andrea Mendoza	3	Andrea Mendoza
2	Martin Lara	NULL	NULL

e) Cross Join: devuelve el producto cartesiano de filas

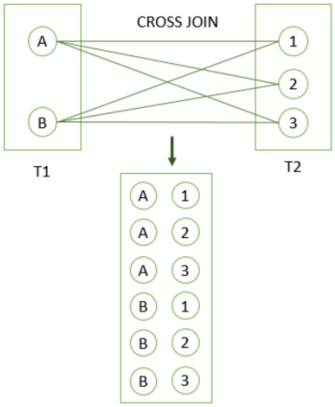
```
SELECT *
FROM Work_team_a
CROSS JOIN Work_team_b;
```

Cross table of 2 tables			
a_id	employee_a	b_id	employee_b
1	Andrea Mendoza	1	Claudia Camacho
1	Andrea Mendoza	2	Carlos Alvarado
1	Andrea Mendoza	3	Andrea Mendoza
1	Andrea Mendoza	4	Pedro Lombardi
2	Martin Lara	1	Claudia Camacho
2	Martin Lara	2	Carlos Alvarado
2	Martin Lara	3	Andrea Mendoza
2	Martin Lara	4	Pedro Lombardi
3	Lucas Rojas	1	Claudia Camacho
3	Lucas Rojas	2	Carlos Alvarado
3	Lucas Rojas	3	Andrea Mendoza
3	Lucas Rojas	4	Pedro Lombardi
4	Carlos Alvarado	1	Claudia Camacho
4	Carlos Alvarado	2	Carlos Alvarado
4	Carlos Alvarado	3	Andrea Mendoza
4	Carlos Alvarado	4	Pedro Lombardi

Un ejemplo que ilustra muy bien cada uno de los casos es este:



Fuente: w3schools



Fuente: PostgreSQL tutorial

2)

a) SELECT LIMIT: se usa para limitar el número de filas que devuelve una cláusula select:

SELECT * FROM work_team_a LIMIT 2;

LIMIT en la tabla Work_team_a	
a_id	employee_a
1	Andrea Mendoza
2	Martin Lara

b) SELECT OFFSET: establece el número de filas que se saltan antes de empezar a devolver filas de la consulta :

SELECT * FROM work_team_a OFFSET 2;

OFFSET en la tabla Work_team_a	
a_id	employee_a
3	Lucas Rojas
4	Carlos Alvarado

DOCUMENTACIÓN:

Para responder a las preguntas, además de los apuntes de la asignatura, he utilizado los siguientes enlaces:

https://ingenieriadesoftware.es/tipos-sql-join-guia-referencia/
https://www.w3schools.com/sql/sql_join.asp
https://www.sqlshack.com/sql-cross-join-with-examples/
https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-cross-join/
https://www.sqlservertutorial.net/sql-server-basics/sql-server-offset-fetch/
https://www.techonthenet.com/sql/select_limit.php

Criterios de valoración

En el enunciado se indica el peso/valoración de cada ejercicio.

Para conseguir la puntuación máxima en los ejercicios, es necesario explicar con claridad la solución que se propone.

Formato y fecha de entrega

Tenéis que enviar la PEC al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación). El formato del archivo que contiene vuestra solución puede ser **.pdf, .doc y .docx**. **Para otras opciones, por favor, contactar previamente con vuestro consultor**.

La fecha límite para entregar la PEC2 es el **21/11/2022**.

Nota: **Propiedad intelectual**

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente, si así corresponde.