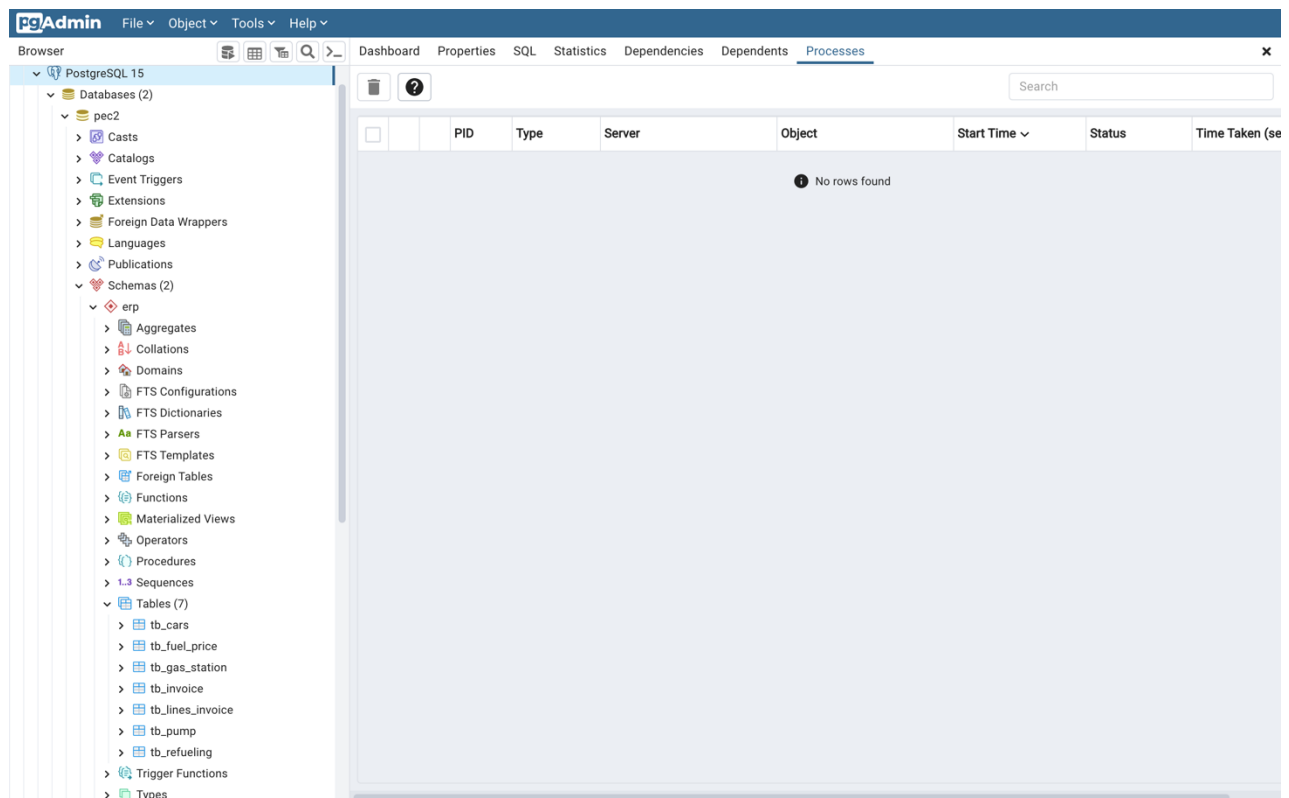


De la creación a la manipulación de una base de datos relacional

NOMBRE Y APELLIDOS: Paula Corbatón Álvarez

EJERCICIO 1 (25%)





EJERCICIO 2 (25%)

a)

PGAdmin interface showing a query execution result for a PostgreSQL database.

Query:

```
1 SELECT cars.cars_registration, cars.cars_model, cars.cars_employee, cars.cars_date_registration
2 FROM erp.tb_cars AS cars
3 WHERE cars.cars_function = 'reparto';
```

Data Output:

	cars_registration [PK] character (7)	cars_model character varying (50)	cars_employee character varying (50)	cars_date_registration date
1	3685HDP	MERCEDES-VITO	Antonio Garcia	2011-05-04
2	2093GSW	CITROEN-JUMPY	Juan Martinez	2010-01-08
3	0815GYR	MERCEDES-VITO	José Pérez	2010-09-11
4	0019GVM	MERCEDES-VITO	Rafael Muñoz	2010-04-23
5	4273GFK	MERCEDES-CITAN	David Ruiz	2008-06-20
6	9421GMT	MERCEDES-VITO	Jesus Alvarez	2009-06-03

Total rows: 6 of 6 Query complete 00:00:00.068 Ln 3, Col 38

b)

PGAdmin interface showing a query execution result for a PostgreSQL database.

Query:

```
1 SELECT car.cars_employee, car.cars_model, gas.gas_name, refi.rf_date
2 FROM erp.tb_refueling AS refi
3 JOIN erp.tb_gas_station AS gas
4   ON gas.gas_id = refi.gas_id
5 JOIN erp.tb_cars AS car
6   ON car.cars_registration = refi.cars_registration
7 WHERE refi.rf_date between '2022-08-01' and '2022-08-31' AND refi.rf_liters = 41
8 ORDER BY car.cars_employee ASC, refi.rf_date DESC;
```

Data Output:

	cars_employee character varying (50)	cars_model character varying (50)	gas_name character varying (50)	rf_date date
1	Jesus Alvarez	MERCEDES-VITO	Gasolinera Norte	2022-08-17
2	Jesus Alvarez	MERCEDES-VITO	Gasolinera Centro	2022-08-08
3	Jesus Alvarez	MERCEDES-VITO	Gasolinera Norte	2022-08-02
4	Juan Martinez	CITROEN-JUMPY	Gasolinera Estación tr...	2022-08-21
5	Pablo Diaz	BMW-X3	Gasolinera Poligono	2022-08-29

Total rows: 5 of 5 Query complete 00:00:00.106 Ln 8, Col 51



c)

pgAdmin interface showing a query execution result for PostgreSQL 15.

Query:

```
1 SELECT car.cars_registration, car.cars_employee, SUM(refu.rf_liters)
2 FROM erp.tb_cars AS car
3 JOIN erp.tb_refueling AS refu -- entendido que por el total de litros se refiere a la suma de l
4 ON refu.cars_registration = car.cars_registration
5 WHERE car.cars_function = 'comercial'
6 GROUP BY car.cars_registration, car.cars_employee
7 ORDER BY SUM(refu.rf_liters) ASC;
```

Data Output:

cars_registration (PK) character (7)	cars_employee character varying (50)	sum bigint
7045KDM	Manuel Rodríguez	174
6392KPT	Carmen Ruiz	224
8806KZN	Ana Maria Sánchez	315

Total rows: 3 of 3 Query complete 00:00:00.077 Ln 7, Col 34

d)

pgAdmin interface showing a query execution result for PostgreSQL 15.

Query:

```
1 SELECT rep.linv_amount, rep.cars_registration
2 FROM ( SELECT inv.linv_amount, car.cars_registration
3 FROM erp.tb_lines_invoice AS inv
4 JOIN erp.tb_cars AS car
5 ON inv.cars_registration = car.cars_registration
6 WHERE car.cars_function = 'reparto') as rep
7 GROUP BY rep.linv_amount, rep.cars_registration
8 HAVING AVG(rep.linv_amount) >
9 (SELECT AVG(inv.linv_amount)
10 FROM erp.tb_lines_invoice AS inv
11 JOIN erp.tb_cars AS car
12 ON inv.cars_registration = car.cars_registration
13 JOIN erp.tb_refueling AS refu
14 ON refu.cars_registration = inv.cars_registration
15 WHERE car.cars_function = 'reparto'
16 AND car.cars_model LIKE '%VITON' AND refu.rf_date between '2022-09-01' and '2022-09-30');;
```

Data Output:

linv_amount numeric (12,2)	cars_registration character (7)
154.68	4273GFK
159.11	0019GVM
185.09	0815GYR
193.88	0019GVM
199.95	4273GFK
229.49	9421GMT
250.60	3685HDP
279.88	2093GSW
282.36	2093GSW
325.11	9421GMT

Total rows: 10 of 10 Query complete 00:00:00.179 Ln 16, Col 91



e)

pgAdmin

Browser

- 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

Query

```

1 SELECT car.cars_registration, refu.gas_id, refu.rf_date
2 FROM erp.tb_refueling AS refu
3 JOIN erp.tb_cars AS car
4   ON refu.cars_registration = car.cars_registration
5 WHERE refu.rf_km IS null AND (refu.gas_id = 'GS04' OR refu.gas_id = 'GS05' OR refu.gas_id = 'GS02')
6 ORDER BY car.cars_registration DESC, refu.rf_date ASC;

```

Data Output

	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)

The screenshot shows the PgAdmin interface with the 'erp' schema selected. The 'tb_cars' table is highlighted in the left pane. The SQL query editor contains the following code:

```
1 INSERT INTO erp.tb_cars(
2   cars_registration, cars_model, cars_function, cars_deposit, cars_fuel, cars_date_input, cars_employee, cars_date_r
3   VALUES ('2233JMN', 'AUDI Q5', 'gerencia', 40, 'gasolina', now(), 'Carmen Sevilla Calvo', to_date('16-03-2016', 'DD-MM-YYYY
4   ('7542LSN', 'AUDI A1', 'gerencia', 30, DEFAULT, now(), 'Carlos Díaz Sevilla', to_date('01-08-2021', 'DD-MM-YYYY')),
5   ('1974LBN', 'AUDI A3', 'gerencia', 35, 'gasol', now(), 'Javier Díaz Sevilla', to_date('30-09-2019', 'DD-MM-YYYY'));
6
7 SELECT * FROM erp.tb_cars
```

The 'Data Output' tab shows the results of the SELECT query. The table has 13 rows and 9 columns. The status bar indicates 'Total rows: 13 of 13' and 'Query complete 00:00:00.064'.

	cars_registration	cars_model	cars_function	cars_deposit	cars_fuel	cars_date_input	cars_employee	cars_date_registration
1	3685HDP	MERCEDES-VITO	reparto	60	gasol	2022-07-22	Antonio García	2011-05-04
2	7045KDM	SKODA-FABIA	comercial	50	gasol	2022-07-23	Manuel Rodríguez	2017-10-05
3	2093GSW	CITROEN-JUMPY	reparto	60	gasol	2022-07-22	Juan Martínez	2010-01-08
4	8806KZN	SEAT-IBIZA	comercial	30	gasolina	2022-07-24	Ana María Sánchez	2019-08-12
5	0815GYR	MERCEDES-VITO	reparto	40	gasol	2022-07-25	José Pérez	2010-09-11
6	5649JSN	BMW-X3	gerencia	60	gasolina	2022-07-26	Pablo Díaz	2016-09-08
7	0019QVM	MERCEDES-VITO	reparto	60	gasol	2022-07-24	Rafael Muñoz	2010-04-23
8	4273GFK	MERCEDES-CITAN	reparto	60	gasol	2022-07-24	David Ruiz	2008-06-20
9	6392KPT	SKODA-FABIA	comercial	30	gasol	2022-07-25	Carmen Ruiz	2018-10-03
10	9421GMT	MERCEDES-VITO	reparto	60	gasol	2022-07-26	Jesus Álvarez	2009-06-03
11	2233JMN	AUDI Q5	gerencia	40	gasolina	2022-11-17	Carmen Sevilla Calvo	2016-03-16
12	7542LSN	AUDI A1	gerencia	30	gasol	2022-11-17	Carlos Díaz Sevilla	2021-08-01
13	1974LBN	AUDI A3	gerencia	35	gasol	2022-11-17	Javier Díaz Sevilla	2019-09-30

b)

The screenshot shows the PgAdmin interface with the 'erp' schema selected. The 'tb_refueling' table is highlighted in the left pane. The SQL query editor contains the following code:

```
1 DELETE FROM erp.tb_fuel_price
2 WHERE fp_date NOT IN (SELECT rf_date
3   FROM erp.tb_refueling)
```

The 'Data Output' tab shows the results of the DELETE query. The status bar indicates 'Total rows: 26 of 26' and 'Query complete 00:00:00.076'. A message box displays: 'DELETE 26' and 'Query returned successfully in 76 msec.'



c)

pgAdmin interface showing a query execution for the PostgreSQL database. The query modifies the `erp.tb_fuel_price` table by adding a new column `fp_import_without_vat` and updating its values.

```

1 ALTER TABLE erp.tb_fuel_price ADD COLUMN fp_import_without_vat NUMERIC(12, 3);
2 UPDATE erp.tb_fuel_price SET fp_import_without_vat = fp_import - (fp_import * 0.21);
3 ALTER TABLE erp.tb_fuel_price ALTER COLUMN fp_import_without_vat SET NOT NULL;
4
5 SELECT * FROM erp.tb_fuel_price;

```

The Data Output tab displays the results of the query, showing 96 rows of data. The columns are `fp_date`, `fp_fuel`, `fp_import`, and `fp_import_without_vat`.

	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 interface showing a query execution for the PostgreSQL database. The query adds a constraint to the `erp.tb_cars` table.

```

1 ALTER TABLE erp.tb_cars ADD CONSTRAINT cars_employee
2 CHECK (cars_function != 'gerencia' OR cars_employee IS NOT NULL);
3
4

```

The Messages tab displays the query execution status: "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*

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

pec2/postgres@PostgreSQL 15

Query Query History

```

1 CREATE USER registerer WITH PASSWORD '1234';
2 GRANT USAGE ON SCHEMA erp TO registerer;
3 GRANT SELECT, INSERT, UPDATE ON erp.tb_refueling TO registerer;
4 GRANT SELECT ON erp.tb_cars TO registerer;
5
6 SELECT username AS role_name,
7 CASE
8 WHEN usesuper AND usecreatedb THEN
9 CAST('superuser, create database' AS pg_catalog.text)
10 WHEN usesuper THEN
11 CAST('superuser' AS pg_catalog.text)
12 WHEN usecreatedb THEN
13 CAST('create database' AS pg_catalog.text)
14 ELSE
15 CAST('' AS pg_catalog.text)
16 END role_attributes
17 FROM pg_catalog.pg_user
18 ORDER BY role_name desc;
19

```

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.

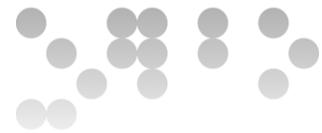
SELECT *

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 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



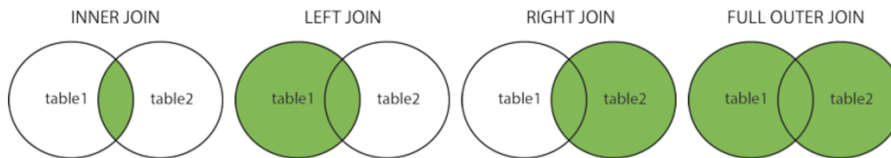
e) Cross Join: devuelve el producto cartesiano de filas

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

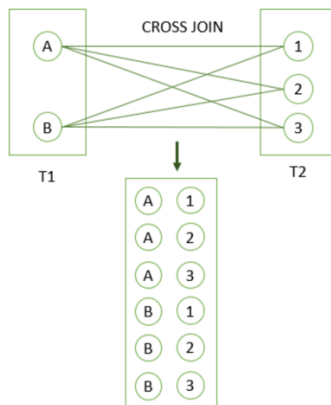
Cross table of 2 tables			
a_id	employee_a	b_id	employee_b
1	Andrea Mendoza	1	Claudia Camacho
1	Andrea Mendoza	2	Carlos Alvarado
1	Andrea Mendoza	3	Andrea Mendoza
1	Andrea Mendoza	4	Pedro Lombardi
2	Martin Lara	1	Claudia Camacho
2	Martin Lara	2	Carlos Alvarado
2	Martin Lara	3	Andrea Mendoza
2	Martin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

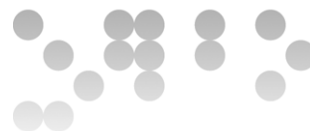
OFFSET en la tabla Work_team_a	
a_id	employee_a
3	Lucas Rojas
4	Carlos Alvarado



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.