



Ampliando la caja de herramientas: *common table expression* y funciones analíticas

y

El diablo está en los detalles: optimización de la base de datos en función de su uso.

NOMBRE Y APELLIDOS: GLORIA MARÍA MANRESA SANTAMARÍA

EJERCICIO 1 (30%)

EJERCICIO 1-1

```
-- EJERCICIO 1.1

BEGIN;

    -- CREAR SECUENCIA
CREATE SEQUENCE seq_cars_id INCREMENT BY 1 START WITH 100;

    -- AÑADIR CAMPO cars_id
ALTER TABLE erp.tb_cars
ADD COLUMN cars_id INTEGER NOT NULL DEFAULT nextval('seq_cars_id');

    -- ELIMINAR RESTRICCIÓN DE CLAVE FORÁNEA EN LAS TABLAS DEPENDIENTES
    -- ELIMINAR RESTRICCIÓN CLAVE PRIMARIA EN tb_cars
ALTER TABLE erp.tb_refueling DROP CONSTRAINT fk_cars;
ALTER TABLE erp.tb_lines_invoice DROP CONSTRAINT fk_cars;
ALTER TABLE erp.tb_cars DROP CONSTRAINT pk_cars;

    -- CREAR NUEVA CLAVE PRIMARIA EN tb_cars
ALTER TABLE erp.tb_cars ADD CONSTRAINT pk_cars PRIMARY KEY (cars_id);

    -- AÑADIR CAMPO cars_id EN LAS TABLAS DEPENDIENTES
ALTER TABLE erp.tb_refueling
ADD COLUMN cars_id INTEGER;

ALTER TABLE erp.tb_lines_invoice
ADD COLUMN cars_id INTEGER;

    -- ACTUALIZAR DATOS cars_id
UPDATE erp.tb_refueling T SET cars_id = (SELECT cars_id FROM erp.tb_cars WHERE cars_registration = T.cars_registration);
ALTER TABLE erp.tb_refueling ALTER COLUMN cars_id SET NOT NULL;

UPDATE erp.tb_lines_invoice T SET cars_id = (SELECT cars_id FROM erp.tb_cars WHERE cars_registration = T.cars_registration);
ALTER TABLE erp.tb_lines_invoice ALTER COLUMN cars_id SET NOT NULL;

    -- CREAR NUEVA CLAVE FORÁNEA EN TABLAS DEPENDIENTES
ALTER TABLE erp.tb_refueling ADD CONSTRAINT fk_cars FOREIGN KEY (cars_id) REFERENCES erp.tb_cars(cars_id) ON DELETE CASCADE;
ALTER TABLE erp.tb_lines_invoice ADD CONSTRAINT fk_cars FOREIGN KEY (cars_id) REFERENCES erp.tb_cars(cars_id) ON DELETE CASCADE;

    -- ELIMINAR cars_registration EN TABLAS DEPENDIENTES
ALTER TABLE erp.tb_refueling
DROP COLUMN cars_registration;

ALTER TABLE erp.tb_lines_invoice
DROP COLUMN cars_registration;

ROLLBACK;
```



A continuación se muestran las tablas implicadas antes de realizar el 'rollback' es decir, antes de cancelar todas las modificaciones realizadas en la transacción.

erp.tb_cars:

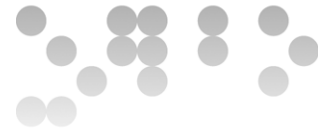
	cars_registration character (7)	cars_model character varying (50)	cars_function character (20)	cars_deposit integer	cars_fuel character (10)	cars_date_input date	cars_employee character varying (50)	cars_date_registration date	cars_id [PK] integer
1	3685HDP	MERCEDES-VITO	reparto	60	gasoil	2022-07-22	Antonio García	2011-05-04	100
2	7045KDM	SKODA-FABIA	comercial	50	gasoil	2022-07-23	Manuel Rodríguez	2017-10-05	101
3	2093GSW	CITROEN-JUMPY	reparto	60	gasoil	2022-07-22	Juan Martínez	2010-01-08	102
4	8806KZN	SEAT-IBIZA	comercial	30	gasolina	2022-07-24	Ana María Sánchez	2019-08-12	103
5	0815GYR	MERCEDES-VITO	reparto	40	gasoil	2022-07-25	José Pérez	2010-09-11	104
6	5649JSN	BMW-X3	gerencia	60	gasolina	2022-07-26	Pablo Díaz	2016-09-08	105
7	0019GVM	MERCEDES-VITO	reparto	60	gasoil	2022-07-24	Rafael Muñoz	2010-04-23	106
8	4273GFK	MERCEDES-CITAN	reparto	60	gasoil	2022-07-24	David Ruiz	2008-06-20	107
9	6392KPT	SKODA-FABIA	comercial	30	gasoil	2022-07-25	Carmen Ruiz	2018-10-03	108
10	9421GMT	MERCEDES-VITO	reparto	60	gasoil	2022-07-26	Jesus Álvarez	2009-06-03	109

erp.tb_refueling:

	gas_id character (5)	pm_id integer	rf_liters integer	rf_date date	rf_km integer	cars_id integer
1	GS01	23	47	2022-08-01	234561	106
2	GS01	23	33	2022-08-26	235411	106
3	GS01	25	55	2022-09-05	236061	106
4	GS01	27	34	2022-09-10	236794	106
5	GS02	31	51	2022-08-06	[null]	106
6	GS02	33	45	2022-08-31	238044	106
7	GS02	33	53	2022-09-15	238794	106
8	GS03	37	39	2022-08-11	239711	106
9	GS03	41	43	2022-09-20	240277	106
10	GS04	43	44	2022-08-16	241161	106
11	GS05	51	42	2022-08-21	241877	106
12	GS01	25	32	2022-08-08	125300	104

erp.tb_lines_invoice

	inv_id [PK] integer	linv_id [PK] integer	gas_id character (5)	linv_liters integer	linv_amount numeric (12,2)	cars_id integer
1	1	1	GS01	80	159.11	106
2	1	2	GS02	96	193.88	106
3	1	3	GS03	39	73.28	106
4	1	4	GS04	44	85.32	106
5	1	5	GS05	42	89.84	106
6	1	6	GS01	95	185.09	104
7	1	7	GS02	60	120.54	104
8	1	8	GS03	28	58.80	104
9	1	9	GS04	60	119.96	104
10	1	10	GS05	65	131.03	104
11	1	11	GS01	139	279.88	102
12	1	12	GS02	144	282.36	102



EJERCICIO 1-2

```

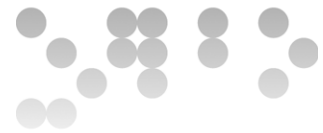
49 BEGIN;
50 WITH RECURSIVE parent_id AS(
51     SELECT pm_id,
52            CAST(pm_descr as text) as parent_list ,
53            SUM(rf_liters) AS liters,
54            pm_parent_id
55     FROM erp.tb_refueling NATURAL JOIN erp.tb_pump
56     GROUP BY pm_id,gas_id,pm_descr,pm_parent_id
57     HAVING SUM(rf_liters) > 300
58 UNION ALL
59     SELECT r.pm_id,
60            CAST( p.pm_descr || '<-' || r.parent_list as text) as parent_list ,
61            r.liters,
62            p.pm_parent_id
63     FROM parent_id r INNER JOIN erp.tb_pump p ON (r.pm_parent_id=p.pm_id)
64 )
65 SELECT pm_id,parent_list,liters|
66 FROM parent_id
67 WHERE pm_parent_id IS NULL
68 COMMIT;

```

Data Output Messages Notifications



	pm_id integer	parent_list text	liters bigint
1	23	GS01<-Surtidor N°1<-Diesel	371
2	25	GS01<-Surtidor N°2<-Diesel	362
3	33	GS02<-Surtidor N°2<-Diesel	308
4	35	GS02<-Surtidor N°3<-Diesel	384



EJERCICIO 1-3

-- EJERCICIO 1.3

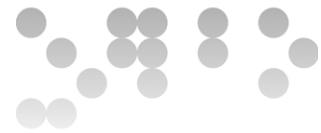
```
BEGIN;
SELECT cars_registration,
       cars_model,
       ROW_NUMBER() OVER (PARTITION BY cars_registration ORDER BY gas_id) AS position_cars,
       gas_id,
       ROW_NUMBER() OVER (PARTITION BY cars_registration,gas_id) AS position_gas,
       rf_date,
       rf_liters,
       SUM(rf_liters) OVER (PARTITION BY cars_registration ORDER BY gas_id,rf_date)
FROM erp.tb_refueling NATURAL JOIN erp.tb_cars
ORDER BY cars_registration,gas_id,rf_date
COMMIT;
```

70 -- EJERCICIO 1.3

71

Data Output Messages Notifications

	cars_registration character (7)	cars_model character varying (50)	position_cars bigint	gas_id character (5)	position_gas bigint	rf_date date	rf_liters integer	sum bigint
1	0019GVM	MERCEDES-VITO	1	GS01	1	2022-08-01	47	47
2	0019GVM	MERCEDES-VITO	2	GS01	2	2022-08-26	33	80
3	0019GVM	MERCEDES-VITO	3	GS01	3	2022-09-05	55	135
4	0019GVM	MERCEDES-VITO	4	GS01	4	2022-09-10	34	169
5	0019GVM	MERCEDES-VITO	5	GS02	1	2022-08-06	51	220
6	0019GVM	MERCEDES-VITO	6	GS02	2	2022-08-31	45	265
7	0019GVM	MERCEDES-VITO	7	GS02	3	2022-09-15	53	318
8	0019GVM	MERCEDES-VITO	8	GS03	1	2022-08-11	39	357
9	0019GVM	MERCEDES-VITO	9	GS03	2	2022-09-20	43	400
10	0019GVM	MERCEDES-VITO	10	GS04	1	2022-08-16	44	444
11	0019GVM	MERCEDES-VITO	11	GS05	1	2022-08-21	42	486
12	0815GYR	MERCEDES-VITO	1	GS01	1	2022-08-08	32	32
13	0815GYR	MERCEDES-VITO	2	GS01	2	2022-08-14	32	64
14	0815GYR	MERCEDES-VITO	3	GS01	3	2022-08-18	31	95
15	0815GYR	MERCEDES-VITO	4	GS01	4	2022-09-02	35	130



EJERCICIO 2 (35%)

EJERCICIO 2-A

Dadas las transacciones T1 y T2:

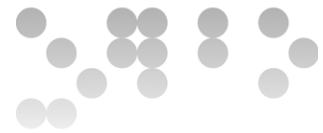
TRANSACCIÓN 1
SELECT inv_amount FROM tb_invoice WHERE inv_id=1
COMMIT

TRANSACCIÓN 2
SELECT inv_amount FROM tb_invoice WHERE inv_id=1
UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
ROLLBACK

Se puede dar la interferencia de lectura no confirmada. Por ejemplo, si las transacciones se ejecutan en el siguiente orden:

TRANSACCIÓN 1	TRANSACCIÓN 2
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
COMMIT	
	ROLLBACK

La lectura de la transacción 1 es un dato pendiente de confirmar, y en este caso finalmente no se confirma (rollback) por lo que el dato leído en realidad nunca estuvo realmente en la base de datos. Esta interferencia toma el nombre de lectura no confirmada y se puede evitar con los siguientes niveles de aislamiento: READ COMMITTED, REPEATABLE READ y SERIALIZABLE.



EJERCICIO 2-B

Esta interferencia no tiene porque ocurrir siempre. Por ejemplo, si las transacciones se ejecutan de la siguiente manera no se produciría ninguna interferencia.

TRANSACCIÓN 1	TRANSACCIÓN 2
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
	ROLLBACK
SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
COMMIT	

En este caso el update se ha cancelado antes de realizar la lectura de la transacción 1 por lo que la interferencia no ocurre. Todo depende del orden de ejecución de las transacciones.

EJERCICIO 2-C

Omitiremos el uso de COMMIT de la transacción 1 ya que al tratarse únicamente de una consulta y esta transacción no modifica ningún dato no es necesario aceptar o rechazar la transacción.

Los posibles horarios de las transacciones son los siguientes:

TRANSACCIÓN 1	TRANSACCIÓN 2
SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
	ROLLBACK
No hay interferencias.	

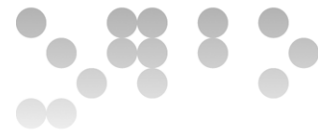
TRANSACCIÓN 1	TRANSACCIÓN 2
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1



SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
	ROLLBACK
No hay interferencias.	

TRANSACCIÓN 1	TRANSACCIÓN 2
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
	ROLLBACK
Interferencia: Lectura no confirmada	

TRANSACCIÓN 1	TRANSACCIÓN 2
	SELECT inv_amount FROM tb_invoice WHERE inv_id=1
	UPDATE tb_invoice SET inv_amount = 20 WHERE inv_id = 1
	ROLLBACK
SELECT inv_amount FROM tb_invoice WHERE inv_id=1	
No hay interferencias.	



EJERCICIO 3 (20%)

EJERCICIO 3-A

El complemento postgres_fdw viene de “Foreign Data Wrapper”, este complemento permite acceder a datos en servidores externos como si fueran locales.

EJERCICIO 3-B

-- EJERCICIO 3

-- Habilitamos la extensión fdw y creamos un nuevo esquema.

```
CREATE EXTENSION postgres_fdw;
CREATE SCHEMA esquema_destino;
```

-- Configuramos el acceso al servidor remoto.

-- En este caso a la base de datos pública RNACentral.

```
CREATE SERVER servidor_bd_remota
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'hh-pgsql-public.ebi.ac.uk', port '5432', dbname 'pfmegrnargs');
```

-- Asignamos un usuario local de nuestro servidor a un usuario del servidor remoto.

```
CREATE USER MAPPING FOR CURRENT_USER SERVER servidor_bd_remota
OPTIONS (USER 'reader', password 'NWDME5xdipIjRrp');
```

-- Importamos el esquema remoto al esquema creado anteriormente.

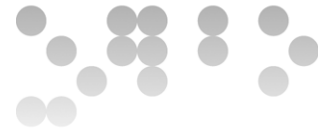
-- En este caso importamos únicamente una tabla 'rnc_taxonomy'.

```
IMPORT FOREIGN SCHEMA rnacen
LIMIT TO (rnc_taxonomy)
FROM SERVER servidor_bd_remota
INTO esquema_destino;
```

-- Consultamos la tabla externa.

```
SELECT *
FROM esquema_destino.rnc_taxonomy
LIMIT(50);
```

	id integer	name text	lineage text
1	1776368	Osedax sp. 1336_61_2	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
2	721752	Osedax sp. 16	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
3	417742	Osedax sp. 2 SBJ-2006	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
4	417743	Osedax sp. 3 SBJ-2006	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
5	417744	Osedax sp. 4 SBJ-2006	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
6	1588042	Osedax sp. Nude-palp-G	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
7	596124	Osedax sp. sagami-3	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
8	596127	Osedax sp. sagami-4	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
9	596125	Osedax sp. sagami-5	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
10	596126	Osedax sp. sagami-6	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
11	596128	Osedax sp. sagami-7	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar
12	596123	Osedax sp. sagami-8	cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Protostomia; Spiralia; Lophotrochozoa; Annelida; Polychaeta; Sedentar



EJERCICIO 4 (15%)

EJERCICIO 4-A


```

1  -- EJERCICIO 4-A
2  EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
3  ON c.cars_registration=r.cars_registration
4

```

Data Output Messages Notifications



	QUERY PLAN	
	text	
1	Hash Join (cost=1.23..3.77 rows=113 width=438)	
2	Hash Cond: (r.cars_registration = c.cars_registration)	
3	-> Seq Scan on tb_refueling r (cost=0.00..2.13 rows=113 width=30)	
4	-> Hash (cost=1.10..1.10 rows=10 width=408)	
5	-> Seq Scan on tb_cars c (cost=0.00..1.10 rows=10 width=408)	

El primer hash join que aparece se utiliza para realizar la equicombinación entre las tablas para ello se crea una tabla temporal con los datos de tb_cars y crea un índice de dispersión (hash) sobre su clave primaria, para ello realiza un acceso secuencia (seq scan) sobre dicha tabla. Una vez hecho esto realiza una lectura secuencial (seq scan) de las filas tb_refueling y para cada fila accederá a su equivalente en tb_cars utilizando el índice de dispersión.

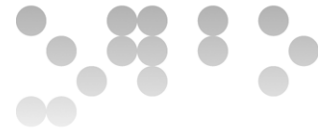
EJERCICIO 4-B

En los planes de ejecución que encontramos en la tabla siguiente observamos que mientras se ha mantenido habilitada la opción seqscan ésta ha sido la opción utilizada independientemente del índice que hayamos creado.

Al deshabilitar la opción de seqscan se han utilizado los índices creados.



SEQ SCAN	SIN INDICE	CON BTREE	CON HASH
ON	<div><div>QUERY PLAN</div><div>text</div><div>Hash Join (cost=1.23..4.91 rows=113 width=438)</div><div>Hash Cond: (r.cars_registration = c.cars_registration)</div><div>-> Seq Scan on tb_refueling r (cost=0.00..2.13 rows=113 width=30)</div><div>-> Hash (cost=1.10..1.10 rows=10 width=408)</div><div>-> Seq Scan on tb_cars c (cost=0.00..1.10 rows=10 width=408)</div></div>	<div><div>QUERY PLAN</div><div>text</div><div>Hash Join (cost=1.23..3.77 rows=113 width=438)</div><div>Hash Cond: (r.cars_registration = c.cars_registration)</div><div>-> Seq Scan on tb_refueling r (cost=0.00..2.13 rows=113 width=30)</div><div>-> Hash (cost=1.10..1.10 rows=10 width=408)</div><div>-> Seq Scan on tb_cars c (cost=0.00..1.10 rows=10 width=408)</div></div>	<div><div>QUERY PLAN</div><div>text</div><div>Hash Join (cost=1.23..3.77 rows=113 width=438)</div><div>Hash Cond: (r.cars_registration = c.cars_registration)</div><div>-> Seq Scan on tb_refueling r (cost=0.00..2.13 rows=113 width=30)</div><div>-> Hash (cost=1.10..1.10 rows=10 width=408)</div><div>-> Seq Scan on tb_cars c (cost=0.00..1.10 rows=10 width=408)</div></div>
OFF	<div><div>QUERY PLAN</div><div>text</div><div>Hash Join (cost=20000000001.22..20000000004.91 rows=113 width=438)</div><div>Hash Cond: (r.cars_registration = c.cars_registration)</div><div>-> Seq Scan on tb_refueling r (cost=10000000000.00..10000000002.13 rows=113 width=30)</div><div>-> Hash (cost=10000000001.10..10000000001.10 rows=10 width=408)</div><div>-> Seq Scan on tb_cars c (cost=10000000000.00..10000000001.10 rows=10 width=408)</div></div>	<div><div><div></div><div>QUERY PLAN</div><div>text</div><div>1 Hash Join (cost=10000000012.41..10000000014.96 rows=113 width=438)</div><div>2 Hash Cond: (r.cars_registration = c.cars_registration)</div><div>3 -> Seq Scan on tb_refueling r (cost=10000000000.00..10000000002.13 rows=113 wid.</div><div>4 -> Hash (cost=12.29..12.29 rows=10 width=408)</div><div>5 -> Index Scan using indexbtree on tb_cars c (cost=0.14..12.29 rows=10 width=408)</div></div></div>	<div><div>QUERY PLAN</div><div>text</div><div>Hash Join (cost=10000000012.41..10000000014.96 rows=113 width=438)</div><div>Hash Cond: (r.cars_registration = c.cars_registration)</div><div>-> Seq Scan on tb_refueling r (cost=10000000000.00..10000000002.13 rows=113 wid..</div><div>-> Hash (cost=12.29..12.29 rows=10 width=408)</div><div>-> Index Scan using indexhash on tb_cars c (cost=0.14..12.29 rows=10 width=408)</div></div>



```

1  -- EJERCICIO 4-A
2  EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
3  ON c.cars_registration=r.cars_registration
4
5  -- EJERCICIO 4-B
6  -- SIN INDICE - Borramos clave principal tb_cars
7  BEGIN;
8  ALTER TABLE ERP.TB_CARS DROP CONSTRAINT pk_cars CASCADE;
9      -- PLAN EJECUCIÓN SEQSCAN OFF SIN INDICE
10 SET enable_seqscan = off;
11 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
12 ON c.cars_registration=r.cars_registration
13
14      -- PLAN EJECUCIÓN SEQSCAN ON SIN INDICE
15 SET enable_seqscan = on;
16 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
17 ON c.cars_registration=r.cars_registration
18
19 ROLLBACK;
20 -- CON BTREE
21 BEGIN;
22      -- CREAR INDICE CON BTREE
23 CREATE INDEX indexbtree
24 ON erp.tb_cars USING btree (cars_registration)
25      -- PLAN EJECUCIÓN SEQSCAN OFF CON BTREE
26 SET enable_seqscan = off;
27 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
28 ON c.cars_registration=r.cars_registration
29      -- PLAN EJECUCIÓN SEQSCAN ON CON BTREE
30 SET enable_seqscan = on;
31 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
32 ON c.cars_registration=r.cars_registration
33 ROLLBACK;
34
35 -- CON HASH
36 BEGIN;
37      -- CREAR INDICE CON HASH
38 CREATE UNIQUE INDEX indexhash
39 ON erp.tb_cars USING btree (cars_registration)
40      -- PLAN EJECUCIÓN SEQSCAN OFF CON HASH
41 SET enable_seqscan = off;
42 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
43 ON c.cars_registration=r.cars_registration
44
45      -- PLAN EJECUCIÓN SEQSCAN ON CON HASH
46 SET enable_seqscan = on;
47 EXPLAIN SELECT * FROM erp.tb_refueling r INNER JOIN erp.tb_cars c
48 ON c.cars_registration=r.cars_registration
49 ROLLBACK;

```



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. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad (PEC4).

La fecha límite para entregar la PEC es el **23/01/2023**.

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.