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: _____

La empresa Industria del Cubito, ya dispone de la base de datos para su ERP, que hemos construido durante el transcurso de la asignatura **Bases de datos para *Data Warehousing***. Quiere implementar una serie de mejoras y, de nuevo, se ha puesto en contacto con nosotros para que implementéis los requisitos que nos han propuesto.

Para la implementación de esta PEC, debéis de crear una base de datos nueva denominada **dbdw_pec4** y ejecutar el *script* adjunto **BBDD_ERP_structure.sql**. El segundo *script* proporcionado (**BBDD_ERP_data.sql**) os dará un conjunto de datos que os permitirá implementar los componentes requeridos en los diferentes apartados de esta PEC. **NOTA**: los datos proporcionados no se pueden modificar y los ejercicios han de realizarse en base a dichos datos.

Consideraciones para la entrega y realización de la PEC:

- Todo lo que se pide en esta PEC está explicado en los bloques didácticos 4 y 5.
- Sed fieles al enunciado. Si algo no está claro, por favor comentadlo en el foro o directamente al correo del profesor colaborador.
- Se recomienda la utilización de **pgAdmin** para la implementación de toda la PEC. Existe otra alternativa que es ***psql*** (línea de comandos), pero es preferible que utilicéis pgAdmin ya que es una interfaz gráfica que os permitirá editar y crear sentencias SQL (así como mostrar los resultados) de forma más sencilla que *psql*.
- Tal y como se indica en el enunciado, la respuesta de los ejercicios que se requiera ha de entregarse en un fichero *.sql* diferente, con el nombre correspondiente. Se evaluará el código entregado en estos ficheros *.sql* y **NO el código que aparezca en el documento o en los pantallazos adjuntos**.
- Las capturas de pantalla de los ejercicios (y explicaciones pertinentes) han de proporcionarse en un documento aparte (se proporciona una plantilla para el caso, **indicad vuestro nombre en el documento**, por favor).

- Se debe de realizar la entrega de todos los ficheros de la PEC (tanto los ficheros *.sql* como el documento con explicaciones y capturas de pantalla) en un fichero comprimido *.zip*.

Consideraciones para la evaluación del ejercicio:

- Se tendrá en cuenta la aplicación de las buenas prácticas de codificación en SQL, de consultas y de programación de procedimientos y disparadores. Es decir: código con sangrado, uso de cláusulas SQL de forma correcta, comentarios, cabeceras en el procedimiento, etc.
- Los *scripts* proporcionados por el estudiante con las soluciones de los ejercicios han de ejecutarse correctamente. El estudiante ha de asegurarse de que lanzando el *script* completo de cada ejercicio no produzca ningún error.
- **Importante**: Las sentencias SQL proporcionadas en los *scripts* han de ser creadas de forma manual y no mediante asistentes que PostgreSQL/pgAdmin puedan proporcionar. Se pretende aprender SQL y no la utilización de asistentes.

Las sentencias SQL proporcionadas en los ejercicios han de ser **solamente** aquellas que pide el enunciado y ninguna otra más. Cualquier sentencia añadida a mayores, si está mal o provoca que el *script* no se ejecute correctamente a la hora de corregirlo, penalizará el ejercicio.

EJERCICIO 1 (30%)

La empresa Industria del Cubito quiere realizar unas mejoras más en la base de datos. Para ello, nos han contactado nuevamente para pedirnos lo siguiente:

1) (10%) Actualmente la tabla *tb_cars* tiene por clave la matrícula del vehículo. Se requiere tener una nueva clave autonumérica. Para resolverlo, nos piden:

- Crear una secuencia llamada *seq_cars_id* que empiece por 100 y que se vaya incrementando en una unidad.
- Añadir el campo *cars_id* de tipo entero, sin nulos. Es muy importante que el campo *cars_id* por defecto utilice la secuencia *seq_cars_id* cuando se añadan nuevos coches.
- Modificar la clave primaria de *cars_registration* a *cars_id*. Para ello sustituir las claves foráneas que apuntan a *cars_registration* por *cars_id*. Esto implica añadir el campo *cars_id* como clave foránea de *tb_cars*, asignar su valor al *cars_id* correspondiente a *cars_registration*, y finalmente eliminar la columna cars registration.
- Para continuar con el resto de ejercicios, volver a ejecutar los *scripts* para restablecer la base de datos original.

2) (10%) Define, **mediante CTE recursivas**, una consulta que devuelva los surtidores y el total de litros por surtidor, sólo de aquellos surtidores que hayan repostado más de 300 litros:

	pm_id integer	parent_list text	liters bigint
1	23	GS01 -> Surtidor Nº1 -> Diesel	371
2	25	GS01 -> Surtidor Nº2 -> Diesel	362
3	33	GS02 -> Surtidor Nº2 -> Diesel	308
4	35	GS02 -> Surtidor Nº3 -> Diesel	384

3) (10%) Define, **mediante funciones analíticas**, la sentencia SQL que obtenga el total de litros repostados de cada coche. En concreto se pide:

- Matrícula y modelo del coche.
- Una columna con el número de repostaje relativo al coche.
- El código de la gasolinera.
- Una columna con el número de repostaje relativo al coche en esta gasolinera.
- La fecha y los litros repostados.
- Una columna con la suma de litros acumulada del coche.
- La información debe estar ordenada por matrícula, gasolinera y fecha.

Las sentencias SQL de estos 3 apartados se tienen que entregar en un fichero llamado **pec4_ej1.sql**.

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

EJERCICIO 2 (35%)

Dadas las siguientes transacciones T1 y T2:

T1
SELECT inv_amount FROM tb_invoice WHERE inv_id=1;
COMMIT

T2
SELECT inv_amount FROM tb_invoice WHERE inv_id=1;
UPDATE tb_invoide SET inv_amout=20 WHERE inv_id=1;
ROLLBACK

a) (10%) Indicad los tipos de interferencias que se pueden producir entre T1 y T2 y el nivel de aislamiento necesario según el SQL estándar para evitarlas.

b) (10%) ¿Que T1 y T2 produzcan interferencias implica que siempre se producirán? Justifica la respuesta.

c) (15%) Detalla todos los posibles horarios de ejecución de T1 y T2. Y para cada uno de ellos indica el tipo de interferencia que genera.

EJERCICIO 3 (20%)

Desde la empresa Industria del Cubito nos piden que realicemos unas tareas de investigación y práctica sobre PostgreSQL y su complemento **postgres_fdw**, como nuevos expertos que sois.

Quieren conocer más **postgres_fdw**, para contestar las siguientes cuestiones que se plantean:

- (5%) ¿Para qué sirve el complemento **postgres_fdw**?
- (15%) Poned un ejemplo mínimo para ver el funcionamiento de este complemento.

EJERCICIO 4 (15%)

a) (5%) Identifica cual es el plan de ejecución que utiliza PostgreSQL para resolver la siguiente consulta. Haz captura de pantalla de dicho plan y descríbelo con tus propias palabras.

```
SELECT * FROM tb_refueling r INNER JOIN tb_cars c
ON c.cars_registration=r.cars_registration
```

b) (10%) PostgreSQL permite habilitar o deshabilitar el algoritmo *SeqScan* utilizado en las *join* mediante las siguientes sentencias:

```
SET enable_seqscan = off;
SET enable_seqscan = on;
```

Completa la siguiente tabla con el plan de ejecución de la *join* del apartado a) en cada caso. Para completar la columna “Sin índice” tendréis que eliminar la clave principal de *tb_cars*. Para la columna “Con Btree” tendréis que crear un índice de tipo *btree*, y eliminarlo antes de completar la siguiente columna. Y para la tercera columna tendréis que crear un índice de tipo hash.

	Sin índice	Con Btree	Con Hash
seq_scan=ON			
seq_scan=OFF			

Finalmente justifica los resultados obtenidos.

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**.