

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

NOMBRE Y APELLIDOS:

La empresa Industria del Cubito, ha firmado un acuerdo con una empresa que posee gasolineras para proveer de combustible a los diferentes vehículos de la empresa, y quiere disponer de una base de datos para el control de la factura que esta empresa le proporcionará cada final de mes. Por este motivo, el departamento de sistemas de información de la empresa se ha puesto en contacto con la UOC y nos ha pedido que sean los alumnos de la asignatura **Bases de datos para *Data Warehousing*** los que se encarguen de este trabajo.

Consideraciones para la entrega y realización de la PEC:

- Todo lo que se pide en esta PEC está explicado en el bloque didáctico 2 (salvo que se trate de un ejercicio de investigación, cuyo enunciado lo especificará). No es necesario adelantar el estudio del material de otros bloques didácticos para la realización de esta PEC.
- 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 utilices 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, cada respuesta a los ejercicios 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:

- 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 (25%)

Se pide proporcionar las sentencias SQL para crear los componentes especificados a continuación. El resultado de este ejercicio ha de entregarse en el fichero **pec2_ej1.sql**.

pec2

Nombre de la base de datos donde se deben de crear el esquema y los componentes requeridos.

Nota: el comando de creación de la base de datos ha de proporcionarse comentado en el script.

erp

Nombre del esquema de base de datos donde se deben de crear los componentes especificados a continuación.

tb_cars

Tabla para guardar información acerca de los diferentes vehículos de la empresa, de los campos que disponemos tenemos dos fechas la de entrada en el sistema y la de matriculación. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_cars Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
cars_registration	Cadena de 7 caracteres fija	No	Si		
cars_model	Cadena de 50 caracteres variable	No	No		
cars_function	Cadena de 20 caracteres fija	No	No		
cars_deposit	Númérico Entero	No	No		
cars_fuel	Cadena de 10 caracteres fija	No	No		gasoil
cars_date_input	Fecha	No	No		Fecha sistema
cars_employee	Cadena de 50 caracteres variable	Sí	No		
cars_date_registration	Fecha	No	No		

tb_gas_station

Tabla para guardar información acerca de las gasolineras donde los vehículos de la empresa pueden repostar. Se quiere guardar información según la especificación proporcionada a continuación. **El campo gas_name es único.**

Tabla: tb_gas_station Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
gas_id	Cadena de 5 caracteres fija	No	Si		
gas_name	Cadena de 50 caracteres variable	No	No		



tb_pump

Tabla para guardar información acerca de los diferentes surtidores de cada gasolinera. Se quiere guardar información según la especificación proporcionada a continuación. **Los campos gas_id y pm_id, su conjunto debe ser único.**

Tabla: tb_pump Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
pm_id	Numérico Entero	No	Si		
pm_parent_id	Numérico Entero	Si	No	tb_pump	
gas_id	Cadena de 5 caracteres fija	No	No	tb_gas_station	
pm_descr	Cadena de 20 caracteres variable	No	No		

tb_fuel_price

Tabla para guardar información acerca del precio de los combustibles por fecha. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_fuel_price Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
fp_date	Fecha	No	Si		
fp_fuel	Cadena de 10 caracteres fija	No	Si		
fp_import	Número de 12 dígitos y 3 decimales.	No	No		

tb_refueling

Tabla para guardar información acerca de los repostajes de los vehículos de la empresa. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_refueling Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
gas_id	Cadena de 5 caracteres fija	No	No	tb_gas_station	
cars_registration	Cadena de 7 caracteres fija	No	No	tb_cars	
pm_id	Numérico Entero	No	No	tb_pump	
rf_liters	Numérico Entero	No	No		
rf_date	Fecha	No	No		Fecha Actual
rf_km	Numérico Entero	Si	No		

tb_invoice

Tabla para guardar información de la cabecera de la factura, cada factura tenemos los consumos de un periodo determinado. Se quiere guardar información según la especificación proporcionada a continuación.

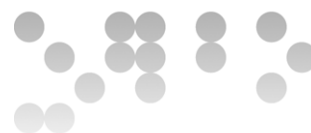


Tabla: tb_invoice Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
inv_id	Numérico Entero	No	Si		
inv_num	Cadena de 5 caracteres fija	No	No		
inv_date_start	Fecha	No	No		
inv_date_end	Fecha	No	No		
inv_amount	Número de 12 dígitos y 2 decimales.	No	No		
inv_liters_total	Numérico Entero	No	No		

tb_lines_invoice

Tabla para guardar información acerca de las líneas de las facturas de cliente. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_lines_invoice Esquema: erp					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
inv_id	Numérico Entero	No	Si	tb_invoice	
linv_id	Numérico Entero	No	Si		
cars_registration	Cadena de 7 caracteres fija	No	No	tb_cars	
gas_id	Cadena de 5 caracteres fija	No	No	tb_gas_station	
linv_liters	Numérico Entero	No	No		
linv_amount	Número de 12 dígitos y 2 decimales.	No	No		

Referente tb_invoice y tb_lines_invoice, es una estructura muy común para almacenar documentos de tipo pedido, albarán, factura. Definimos una tabla cabecera con los datos generales del documento y una tabla líneas con los elementos incluidos en el documento, cada documento tiene 1 registro en cabecera y N registro en líneas. Cada línea dispone de campos numéricos, al agruparse en un documento nos interesa guardar los totales de todas líneas, estos se almacenan en la tabla cabecera, de manera que si hay una modificación en los campos numéricos de las líneas se deben recalculan las cifras de la cabecera.

En nuestra base de datos tenemos repostajes de agosto y septiembre de 2022, pero solo disponemos de la factura del periodo 01/08/2022 al 31/08/2022.

Una vez ejecutadas las sentencias SQL de creación de las estructuras que se requieren, se pide ejecutar el script **BBDD_Erp_data.sql** adjunto. Este fichero NO puede ser modificado para la realización de la PEC y es necesario que todas las sentencias SQL del fichero se ejecuten correctamente. **Si se encuentran indicios de que el fichero se ha modificado o que los datos están incompletos, toda la PEC será considerada como no válida.**



EJERCICIO 2 (25%)

Se pide proporcionar, para cada apartado, **una única consulta en lenguaje SQL**, y el resultado de dicha consulta mediante una captura del resultado en pgAdmin. Para poder realizar este ejercicio, el primer ejercicio ha de estar finalizado de forma correcta.

Las consultas SQL de este ejercicio han de entregarse en un fichero llamado **pec2_ej2.sql** (las capturas de pantalla, y explicaciones si son necesarias, han de entregarse en un documento adjunto).

Los resultados de las consultas no deben contener repeticiones en ningún caso, por lo que la eliminación de resultados puede ser necesaria en alguna de las consultas. Por otro lado, el uso innecesario de la cláusula DISTINCT será penalizado.

Nota: La instalación por defecto de PostgreSQL es *case sensitive*, esto es, la cadena de caracteres “ALUMNO UOC” es diferente de “Alumno UOC” y de “alumno uoc”.

Nota: Es posible que para alguna de las consultas exista más de una solución. En este caso, cualquiera de las soluciones se considerará válida, siempre y cuando cumpla con las aclaraciones proporcionadas en el enunciado.

- a) Proporcionar los vehículos de reparto. Queremos obtener la matrícula, modelo, empleado que conduce y fecha de matriculación. Los datos deben aparecer ordenados por fecha de matriculación en orden descendiente.
- b) Proporcionar los repostajes hechos durante el mes de agosto de 2022, solo debemos mostrar los que han llenado 41 litros de combustible. Queremos obtener el nombre del empleado, modelo de vehículo, nombre de la gasolinera y fecha del repostaje. La consulta debe estar ordenada por nombre empleado ascendentemente y fecha de repostaje descendientemente.
- c) Proporcionar los litros totales consumidos por los vehículos comerciales. Queremos obtener la matrícula del vehículo, el nombre del empleado asignado al vehículo y el total de litros. La consulta debe estar ordenada por el total de litros ascendiente.
- d) Proporcionar la lista de vehículos de reparto que, en el modelo contenga el texto VITO, que la media del importe repostado en el periodo de septiembre de 2022 sea inferior a la media del importe repostado en el mismo periodo de todos los vehículos de reparto. Queremos ver la matrícula del vehículo y la media del importe repostado del vehículo.
- e) Cuando hacen un repostaje, le piden al conductor los kilómetros que tiene actualmente el vehículo, pero a veces no lo entran. Por ello, queremos obtener los repostajes que, al repostar, no hayan proporcionado el kilometraje en las gasolineras, GS04, GS05 y GS02. Queremos ver la matrícula del vehículo, el código de la gasolinera donde ha repostado y la fecha de repostaje. Los resultados deben aparecer ordenados por matrícula descendiente y fecha de repostaje ascendiente.



EJERCICIO 3 (25%)

Se pide proporcionar, para cada apartado las sentencias en lenguaje SQL **y el resultado de estas sentencias mediante una captura del resultado en pgAdmin**. Para poder realizar este ejercicio, el ejercicio 1 ha de estar finalizado.

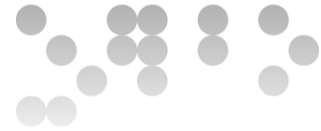
Las sentencias SQL de este ejercicio han de entregarse en un fichero llamado **pec2_ej3.sql** (las capturas de pantalla han de entregarse en un documento adjunto).

- a) Nos dicen que hace falta insertar 3 nuevos vehículos según la especificación dada.

Nombre columna	Vehiculo 1	Vehiculo 2	Vehiculo 3
cars_registration	2233JMN	7542LSN	1974LBN
cars_model	AUDI Q5	AUDI A1	AUDI A3
cars_function	gerencia	gerencia	gerencia
cars_deposit	40	30	35
cars_fuel	gasolina	Valor por defecto	gasoil
cars_date_input	Fecha ejecución inserción	Fecha ejecución inserción	Fecha ejecución inserción
cars_employee	Carmen Sevilla Calvo	Carlos Díaz Sevilla	Javier Díaz Sevilla
cars_date_registration	16/03/2016	01/08/2021	30/09/2019

- b) Nos piden que eliminemos de la tabla `tb_fuel_price` las filas que en esa fecha no tengamos ningún repostaje de nuestros vehículos.
- c) Nos dicen que la columna `fp_import` de la tabla `tb_fuel_price`, tiene el precio del combustible incluyendo el 21% de IVA, nos piden que creamos un campo llamado `fp_import_without_vat` con el mismo formato que `fp_import`, pero sin la restricción de nulos y que lo rellenemos con el precio de la columna `fp_import` sin el IVA, una vez rellenado añadir la restricción que no acepta nulos.
- d) Nos comentan que se debe de alterar, en una única sentencia, la tabla `tb_cars` para controlar que el campo `cars_employee` no sea nulo, siempre debe tener el nombre del empleado que utiliza el vehículo, solo puede estar con valor nulo cuando el automóvil sea de gerencia.
- e) Se ha de crear un usuario de sistema **registerer con acceso al esquema `erp`** y contraseña **1234**. Este usuario puede realizar únicamente lecturas, inserciones, y actualizaciones (nunca borrados ni truncamientos) de la tabla `tb_refueling`, y permisos de lectura en la tabla `tb_cars`. No podrá acceder a las otras tablas del esquema **erp**. Asignar los permisos necesarios a dicho usuario mediante SQL. Asegurad también de que este usuario no pueda asignar permisos a otros usuarios sobre dichas tablas. Referencia a la documentación:

<https://www.postgresql.org/docs/12/sql-grant.html>



Nota: podéis probar los permisos creando una nueva conexión con el usuario creado y mantener así las dos sesiones abiertas. Para ello, debéis crear un nuevo servidor en el Browser de pgAdmin con las credenciales del nuevo usuario.

EJERCICIO 4 (25%)

Desde la empresa nos piden que realicemos unas tareas de investigación sobre PostgreSQL, ya que no disponen de un experto en estos momentos para poder realizarlas.

Contesta brevemente a los siguientes apartados:

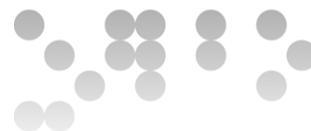
1) Explicar los diferentes tipos de JOIN ofrecidos por PostgreSQL y que son cubiertos en los apuntes de la asignatura, destacando sus diferencias y casos de uso. Además de lo anterior, se requiere explicar el tipo de JOIN llamado CROSS JOIN no abordado en los apuntes. **Se ha de utilizar la documentación en línea de PostgreSQL**, y se han de proporcionar los enlaces a la documentación que se haya utilizado para responder a las preguntas propuestas.

La explicación debe ser complementada además con ejemplos sobre las siguientes tablas.

Tabla: Work_team_a	
Campo: a_id	Campo: employee_a
1	Andrea Mendoza
2	Martin Lara
3	Lucas Rojas
4	Carlos Alvarado

Tabla: Work_team_b	
Campo: b_id	Campo: employee_b
1	Claudia Camacho
2	Carlos Alvarado
3	Andrea Mendoza
4	Pedro Lombardi

2) Explicar las siguientes opciones LIMIT y OFFSET de las consultas SELECT.
La explicación debe ser complementada con ejemplos sobre las tablas del ejercicio anterior.



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.