Procedimientos almacenados y disparadores, ¿para qué son necesarios?

NOMBRE Y APELLIDOS: _____

El Comité Olímpico de la UOC, encargado de organizar los *Juegos Atléticos Olímpicos UOC*, ya dispone de la base de datos que hemos construido durante el transcurso de la asignatura **Bases de datos para *Data Warehousing***. Por fin ha obtenido las subvenciones necesarias para 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 propuesta de solución de esta PEC, debéis de crear una base de datos nueva denominada **dbdw_pec3.** Primero ejecutar el script adjunto **DB_olympic_structure.sql**. A continuación ejecutar el script **DB_olympic_data.sql.** Estos dos scripts SQL crearán la estructura y el conjunto de datos necesarios para el desarrollo de esta PEC.

Consideraciones para la entrega y realización de la PEC:

- Todo lo que se pide en esta PEC está explicado en los bloques didácticos 2 y 3 (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.
- En esta PEC trabajaremos procedimientos/funciones y disparadores. Al tratarse de objetos más complejos, debéis de asegurar una correcta ejecución de estos. Se recomienda que probéis de forma exhaustiva los componentes programados y os creéis vuestros casos de prueba utilizando la base de datos proporcionada.
- 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, 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:

- 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%)

El Comité Olímpico UOC nos ha pedido que implementemos una serie de modificaciones en el esquema *olympic* de su base de datos que se enumeran en los siguientes puntos:

a) (10%) En la tabla REGISTRO (*tb_register*) tenemos el atributo *register_date* de tipo *date.* Es necesario almacenar el valor de la fecha y hora del registro, así pues el tipo del atributo se convertirá en *timestamp.* También debe actualizarse automáticamente al insertar una nueva fila de información, no permitir almacenar valores nulos y modificar el nombre actual *register_date* por *register_ts*.

b) (10%) Se pide crear el nuevo atributo register_updated de tipo *timestamp* en la tabla REGISTRO (*tb_register*). Este atributo permite almacenar valores nulos.
El valor inicial tiene que ser el mismo valor que el introducido en el atributo *register_ts*. Crea la función *fn_register_inserted* para rellenar el atributo register_updated.
Esta función será ejecutada por el disparador *tg_register_inserted* siempre que se introduzca un nuevo registro.

c) (10%) Se pide crear la función *fn_register_updated* con el objetivo de actualizar el valor del atributo *register_updated,* de la tabla REGISTRO (*tb_register*), automáticamente cuando haya una modificación de cualquier otro valor de los atributos de la tupla de información.
Es necesario que la función utilice una sentencia *update* para actualizar el atributo *register_updated.*
El nuevo valor del atributo *register_updated* será la fecha y tiempo actual.
Esta función será ejecutada por el disparador *tg_register_updated*.

Nota: utilizad las denominadas variables especiales de PostgreSQL, que están disponibles para los *trigger procedures*.

El código de este ejercicio se tiene que entregar en un fichero llamado **PEC3_E1.sql.**

EJERCICIO 2 (50%)

El Comité Organizador UOC nos ha pedido que implementemos una serie de novedades en su base de datos para poder gestionar una serie de requisitos que les piden los colaboradores y patrocinadores de los atletas.
Básicamente, los colaboradores y patrocinadores quieren que se les notifique, vía correo electrónico, de forma diaria los resultados de sus atletas.

a) (10%) Crear un dominio *email_type*, en el esquema *olympic,* que debe chequear si se le introduce el formato correcto de correo electrónico. Crear una nueva columna *email* de tipo *email_type* en cada una de las tablas siguientes: PATROCINADORES (*tb_sponsor*) y COLABORADORES (*tb_collaborator*).

b) (10%) Crea la tabla *tb_athletes_info_log*, en el esquema *olympic, que* servirá para almacenar la información de clasificación de los atletas después de que se actualice en la tabla *tb_register*, para poder ser enviada. El formato de esta tabla tiene que ser el siguiente:

- ID atleta
- ID disciplina
- Número de tanda
- Nombre atleta
- Nombre disciplina
- Marca del atleta
- Posición en la tanda
- Fecha de la información

Tabla: tb_athletes_info_log **Esquema:** olympic				
Nombre columna	**Tipo de datos**	**Acepta Nulos**	**Clave primaria**	**Clave foránea**
athlete_id	Cadena de 7 caracteres	No	Si	tb_register
discipline_id	Numérico entero	No	Si	tb_register
round_number	Numérico entero	No	Si	tb_register
athlete_name	Cadena de 50 caracteres variable	No	No	
discipline_name	Cadena de 50 caracteres variable	No	No	
mark	Cadena de 12 caracteres variable	No	No	
rating	Numérico entero	No	No	
info_log_dt	Fecha	Si	No	

c) (15%) Crea un disparador *tg_athletes_info* que ejecute la función *fn_athletes_info* cuando se inserte, elimine o actualice cada fila de la tabla REGISTRO *(tb_register)*. La nueva función, *fn_athletes_info*, tiene que insertar o actualizar la información que necesita la tabla *tb_athletes_info_log,* o eliminarla en caso de que se haya eliminado de la tabla REGISTRO (*tb_register)*.
Para poder hacer cualquiera acción sobre la tabla *tb_athletes_info_log,* la función obtendrá toda la información necesaria de la tablas: REGISTRO *(tb_register)*, ATLETA (*tb_athlete*) y DISCIPLINA (*tb_discipline*).

El procedimiento tiene las siguientes características:
- No tiene parámetros.
- Tiene que devolver los nuevos valores introducidos.

El atributo Marca del atleta (mark de la tabla tb_athletes_info_log) almacenará el valor no nulo de uno de los dos atributos siguientes de la tabla *tb_register*:
- El valor del atributo *register_time*
- El valor del atributo *register_measure*

Recordad que el atributo Marca del atleta (*mark* de la tabla *tb_athletes_info_log*) debe contener una cadena de 12 caracteres variables.
Así pues, es necesario transformar el tipo del valor del atributo no nulo (*register_time* o *register_measure*) a cadena de caracteres.

d) (15%) Crea una nueva función, *fn_get_info_by_sponsor,* donde se obtenga la información requerida a continuación según la fecha y el nombre del patrocinador que se le pase.

El procedimiento tiene las siguientes características:
- Tiene como parámetros:
  - Fecha de entrada (nombre de la variable: *select_date*)
  - Nombre del patrocinador (nombre de la variable: *sponsor*)
- Tiene que devolver una tabla con los atributos siguientes:
  - Correo electrónico patrocinador (tipo del dominio *email_type*)
  - Nombre patrocinador (cadena de 100 caracteres)
  - Nombre atleta (cadena de 50 caracteres)
  - Nombre disciplina (cadena de 50 caracteres)
  - Número de tanda (número entero)
  - Marca del atleta (cadena de 12 caracteres)
  - Posición en la tanda (número entero)
  - Fecha de la información (fecha)

El código de este ejercicio se tiene que entregar en un fichero llamado **PEC3_E2.sql.**

EJERCICIO 3 (20%)

Desde el Comité Olímpico UOC nos piden que realicemos unas tareas de investigación y práctica sobre PostgreSQL, como nuevos expertos que sois.

Quieren conocer más sobre cómo estructurar los datos en formato *JSON* y cómo agrupar datos en *arrays*, para ello responde los siguientes apartados:

a) (10%) Desarrolla con tus propias palabras las siguientes preguntas, dentro del contexto de los SGBD de PostgreSQL:

- ¿ Qué son los *JSON*?
- ¿ Qué son los *arrays*?
- ¿ Qué diferencia hay entre *arrays* y *JSON*?

Nota: En caso de que sea necesario, se puede acompañar la explicación con esquemas, imágenes o ejemplos simples que ayuden a facilitar la comprensión de la respuesta.

(máximo 1 página)

b) (10%) Transforma la propuesta de solución del apartado **d)** del ejercicio **2**, para incorporar cada tupla de información en formato JSON y que se almacenen todas ellas en un array.

El código de este apartado se tiene que entregar en un fichero llamado **PEC3_E3.sql.**

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 (PEC3).

La fecha límite para entregar la PEC3 es el **09/12/2021**.

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.