Práctica Tema 2 y 3

Resultados de aprendizaje: RA3 y RA4

Todos los comandos de SQL y conexiones a la base de datos se deben hacer desde una terminal del sistema, no se puede usar el pgAdmin.

1. Diseña un diagrama Entidad–Relación (E/R) para una base de datos de la temática que vosotros queráis (por ejemplo, una biblioteca, una red social, un gimnasio, un hospital, una tienda online, un videojuego, etc.). Dos grupos no pueden tener bases de datos similares. Podéis aprovechar y hacerlo para la base de datos del proyecto intermodular del ciclo.

El modelo debe cumplir obligatoriamente con los siguientes requisitos:

- Incluir al menos una entidad fuerte y una entidad débil.
- Contener relaciones de tipo:
    - 1:1
    - 1:N
    - N:M
- Incluir una relación reflexiva (una entidad que se relacione consigo misma).
- Tener al menos una relación con un atributo propio.
- Contar con entidades que presenten los siguientes tipos de atributos:
    - Un atributo único (no cuenta la clave primaria).
    - Un atributo opcional.
    - Un atributo multivaluado.
    - Un atributo compuesto.
    - Un atributo derivado.

Puedes añadir más entidades, relaciones y atributos si lo consideráis necesario, siempre que se cumplan los puntos anteriores.

2. A partir del diagrama Entidad–Relación que has diseñado en el ejercicio anterior, realiza su transformación al modelo relacional.

3. A partir del modelo relacional obtenido en el ejercicio anterior, elabora dos scripts SQL que permitan implementar y poblar tu base de datos.

- Script 1 – Creación de tablas y relaciones
    - Crea una base de datos cuyo nombre esté formado por la palabra Proyecto_ seguida de la concatenación de las dos primeras letras del nombre de cada miembro del grupo. De ahora en adelante, en este documento, la expresión “Código de grupo” se utilizará para referirse a dicha concatenación.
      Ejemplo: si el grupo está formado por Laura, Miguel y Andrea, el nombre de la base de datos sería: Proyecto_LaMiAn
    - Crea todas las tablas y relaciones correspondientes a tu modelo relacional.
    - Guarda este archivo con el nombre: creacion.sql
- Script 2 – Inserción de datos de prueba
    - Inserta datos en todas las tablas creadas.
    - Cada tabla debe contener al menos 10 filas de datos significativos y coherentes con el modelo.
    - Guarda este archivo con el nombre: insercion.sql

4. Una vez creada la base de datos, documenta los siguientes procesos:

- Vistas simples

    1. Crea una vista simple con el nombre vista_simple_<Código de grupo> que devuelva el resultado de una consulta que incluya, como mínimo:
        - Una unión entre tres o más tablas.
        - Una subconsulta.
        - Agrupamiento mediante GROUP BY.
        - Ordenación mediante ORDER BY.

       Si por el modelo de negocio no es posible incluir todos estos elementos en una única vista, puedes crear varias. En ese caso, debes nombrarlas como: vista_simple_<Código de grupo>_1, vista_simple_<Código de grupo>_2, etc.
       Ejecuta la vista o vistas creadas para comprobar que se hayan creado correctamente.

    2. Selecciona una de las vistas creadas y renómbrala añadiendo el prefijo “mi_” al inicio. Por ejemplo: vista_simple_<Código de grupo> → mi_vista_simple_<Código de grupo>.
       Ejecuta la vista modificada.

3. Una vez renombrada la vista, procede a cambiar el nombre de una de sus columnas. Ejecuta la vista modificada.

4. Crea una nueva vista llamada vista_muy_simple_<Código de grupo> que muestre todas las filas y todas las columnas de una tabla, excepto una columna que elijas. Ejecuta la vista creada.
   A continuación, consigue que la vista con el mismo nombre (vista_muy_simple_<Código de grupo>) pero que muestre todas las columnas de esa tabla. Ejecuta la vista modificada.
   Finalmente, elimina dicha vista. Intenta ejecutar la vista eliminada para verificar que ya no existe.

- Vistas materializadas

  1. Crea una vista materializada con el nombre vista_materializada_<Código de grupo> que devuelva el resultado de una consulta que incluya, como mínimo:

     - Una unión entre dos o más tablas. Usa un JOIN distinto al elegido en el apartado anterior.

     Identifica cuántas filas se han almacenado en caché.

  2. Ejecuta la vista creada.

  3. Añade una fila o modifica los datos de alguna tabla involucrada en la vista. Posteriormente, actualiza la vista materializada y ejecútalo para comprobar que se ha actualizado correctamente.

- Vistas actualizables

  1. Crea una vista actualizable con el nombre vista_actualizable_<Código de grupo> que devuelva todos las columnas de una tabla.

  2. Utiliza la vista creada para:

     - Añadir una nueva fila a la tabla.
     - Modificar una fila existente.
     - Eliminar una fila.

5. Ahora, documenta el proceso completo para crear y administrar un rol cuyo nombre coincida con el Código de grupo. El rol debe poder iniciar sesión. En la documentación hay que incluir:

1. Otorga al rol los permisos necesarios para realizar las tareas que se detallan a continuación. Para cada tarea, debes documentar los tres momentos del proceso:

    a. Comprobación inicial, mostrando que el rol no dispone del permiso y que la acción falla.

    b. Concesión del permiso correspondiente.

    c. Verificación posterior, realizando la acción con el rol y demostrando que ahora sí puede ejecutarla.

    Las tareas que deben documentarse son:

    - Conectarse a una base de datos.
    - Crear una tabla en el esquema público.
    - Hacer una consulta en una tabla que no haya sido creada con ese rol.
    - Insertar y actualizar una tabla que no haya sido creada con ese rol (dar ambos permisos en una misma sentencia).
    - Hacer una consulta a una de las vistas creadas anteriormente que no utilice la tabla donde ya tiene permiso de consulta.

2. Modificación del rol, documentando cada paso:

    - Cambiar la contraseña del rol.
    - Quitar el permiso de crear tablas.

3. Eliminar el rol.

6. Finalmente, documenta el proceso completo para crear y comprobar el correcto funcionamiento de los siguientes triggers, asegurándose de que tanto los triggers como sus funciones asociadas incluyen el Código de grupo en su nombre:

1. Trigger para mantener un atributo derivado

- Añade en la tabla correspondiente un campo destinado a almacenar el atributo derivado.
- Implementa un trigger que actualice dicho campo cada vez que se inserte una nueva fila o se modifique una existente, asegurando así que el valor derivado se mantiene siempre coherente.

2. Trigger para refrescar una vista materializada

- Crea un trigger que se ejecute cuando se modifique cualquiera de las tablas que forman parte de la vista materializada.

- El trigger deberá ejecutar una instrucción de refresco sobre la vista para mantenerla actualizada.

3. Trigger para validar datos antes de insertar o actualizar

- Implementa un trigger que compruebe que un determinado campo cumple unas reglas de validación antes de que se inserte o actualice una fila.
- Por ejemplo, validar que un campo de correo electrónico tenga un formato correcto.
- Si la validación no se cumple, el trigger debe generar un error y evitar que la operación continúe.

4. Trigger para registrar las modificaciones realizadas por un usuario

- Crea una tabla de auditoría con una clave primaria numérica autoincremental que almacene:
    - El nombre del usuario que realizó la modificación.
    - La fecha en la que se produjo la operación.
    - El tipo de operación realizada (INSERT, UPDATE o DELETE).
- Implementa un trigger que inserte un registro en esta tabla cada vez que un usuario modifique la tabla objetivo.