

[online](#)

Título de la tarea: Controlar la seguridad de datos mediante Triggers y procedimientos

Unidad 6: Programación de BD

Ciclo formativo y módulo: Ciclo superior de DAW/DAM

Módulo de BDDA

Curso académico: 2024/2025

¿Qué contenidos o resultados de aprendizaje trabajaremos?

Contenidos

1. Conceptos básicos.
 1. Unidades léxicas (I).
 2. Tipos de datos simples, variables y constantes.
 3. El bloque PL/SQL.
 4. Estructuras de control (I).
 5. Manejo de errores (I).
2. Tipos de datos compuestos.
 1. Registros.
 2. Colecciones. Arrays de longitud variable.
 3. Cursores.
3. Abstracción en PL/SQL.
 1. Subprogramas.
 2. Paquetes.
 3. Objetos.
4. Disparadores.
 1. Definición de disparadores.
 2. Ejemplos de disparadores.
5. Interfaces de programación de aplicaciones para lenguajes externos.

Resultados de Aprendizaje

- **RA**

1.- Descripción de la tarea

¿Qué te pedimos que hagas?

Utilizando la BD de Películas que hemos utilizado en tareas anteriores vamos a seguir añadiendo restricciones y elementos que garanticen la seguridad de la misma. En este caso mediante triggers y procedimientos.

1º parte, ejercicios básicos. 1 punto cada uno

- **Ejercicio 1: Procedimiento - Insertar una nueva película**

Crear un **procedimiento almacenado** que permita insertar una nueva película en la base de datos asegurando que el título no esté vacío y que la duración sea mayor a 0.

PROCEDURE InsertarPelicula(p_titulo VARCHAR2, p_duracion NUMBER, p_calificacion NUMBER)

Requisitos:

- Si el título está vacío o solo tiene espacios en blanco, mostrar un error.
- Si la duración es menor o igual a 0, mostrar un error.
- Insertar la película en la tabla Peliculas con los datos proporcionados.

- **Ejercicio 2: Procedimiento - Contar películas alquiladas por un usuario**

Crear un **procedimiento almacenado** que reciba el **ID de un usuario** y devuelva la cantidad de películas alquiladas por ese usuario.

PROCEDURE ContarPeliculasAlquiladas(p_id_usuario NUMBER, p_total OUT NUMBER)

Requisitos:

- Consultar cuántas películas tiene alquiladas el usuario en la tabla Alquilada.
- Devolver el número total en un **parámetro de salida**.
- Si el usuario no tiene alquileres, devolverá **0**.

- **Ejercicio 3: Trigger - Evitar precios negativos en alquileres de películas**

Crear un **trigger** que impida insertar o actualizar un alquiler de película con un **precio menor a 0**.

Requisitos:

- Si el precio es menor a 0, lanzar un error con `RAISE_APPLICATION_ERROR`.
- Aplicar la restricción en las operaciones `INSERT` y `UPDATE` sobre la tabla Alquiladas.

- **Ejercicio 4: Trigger - Registrar fecha de alquiler automáticamente**

Crear un **trigger** que, al insertar un alquiler en la tabla Alquilada, asigne **automáticamente la fecha actual** si no se especifica.

Requisitos:

- Si `fecha_alquiler` es `NULL`, asignar la fecha actual (`SYSDATE`).
- El trigger debe ejecutarse **antes de insertar** un nuevo registro en la tabla Alquilada.

2º parte, ejercicios complejos. 2 puntos cada uno.

- **Ejercicio 5: Procedimiento para cambiar las películas de un usuario a otro**

Crear un **procedimiento** que permita cambiar todas las películas alquiladas por un usuario determinado (usuario origen) a otro usuario (usuario destino).

PROCEDURE CambiarPelículasUsuario(id_UsuarioOrigen NUMBER, id_UsuarioDestino NUMBER)

El procedimiento deberá:

- Comprobar que ambos usuarios existen y que no son iguales.
- En caso de que uno de los usuarios no exista, lanzar un error con `RAISE_APPLICATION_ERROR`.
- Si los usuarios son iguales, mostrará un mensaje de error.
- Actualizar todas las películas alquiladas por el usuario origen para asignarlas al usuario destino.
- Mostrar un mensaje con la cantidad de películas cambiadas y los nombres de los usuarios afectados.

- **Ejercicio 6: Disparador para asegurar restricciones en la tabla Películas**

Crear un **TRIGGER** que valide las siguientes restricciones antes de insertar o actualizar una película:

- El título de la película no puede estar vacío.
- La duración de la película debe ser mayor a 0 minutos.
- La calificación debe estar entre 0 y 10.

Si alguna de estas restricciones no se cumple, se debe lanzar un error con un mensaje adecuado indicando la razón.

- **Ejercicio 7: Disparador para validar alquileres**

Crear un TRIGGER que valide lo siguiente al insertar o actualizar en la tabla Alquilerada:

- La fecha de devolución debe ser posterior a la fecha de alquiler.
- El precio del alquiler no puede ser menor a 0.
- Un usuario no puede alquilar la misma película dos veces sin haberla devuelto antes.

Si alguna de estas condiciones no se cumple, se debe lanzar un error con un mensaje adecuado.

2.- Información de interés

Recursos necesarios y recomendaciones

Se tiene que entregar únicamente un fichero .sql con las sentencias precedidas de las líneas oportunas de comentario, con minimamente el nº correspondiente de la pregunta.

El archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_BDDA06_Tarea_1.sql

3.- Requisitos formato de la tarea

Requisitos formato tarea

Enlace al documento con los requisitos del formato que debe cumplir la tarea. [ENLACE](#)

4.- Evaluación de la tarea

Criterios de evaluación implicados

- 1º parte 1 punto cada ejercicio
- 2º parte 2 puntos cada ejercicio

ESTA TAREA CORRESPONDE AL 100% DE LA NOTA DE EJERCICIOS DE LA UT 6