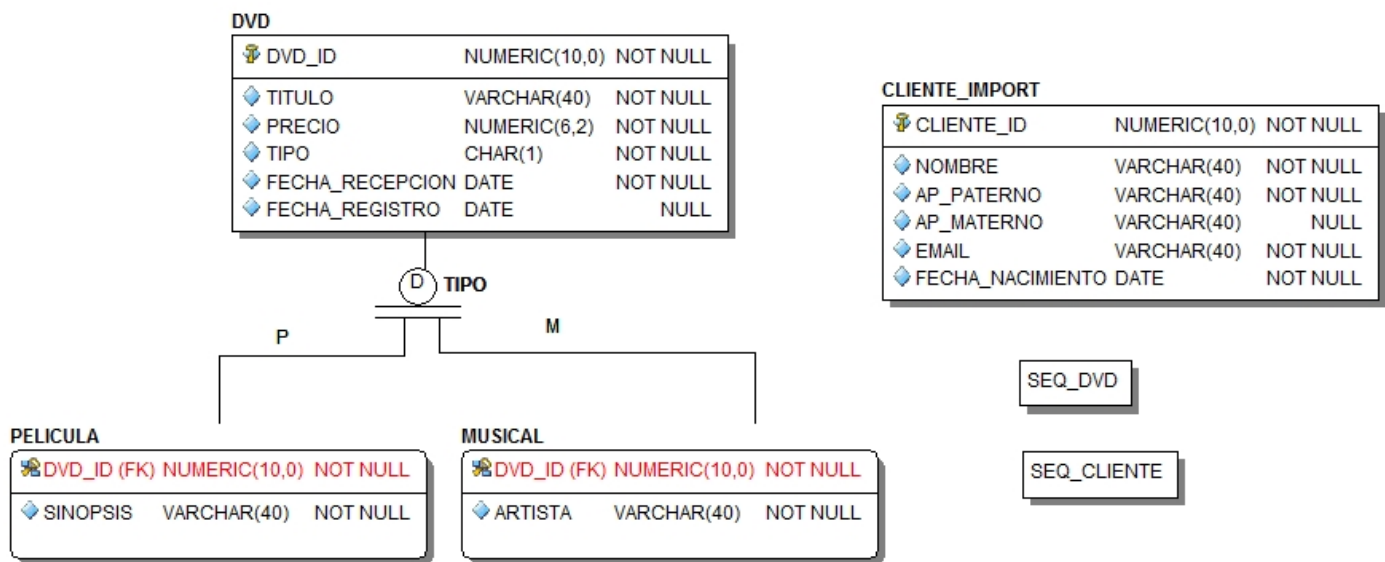


TEMA 8 - EJERCICIOS
LENGUAJE DE MANIPULACION DE DATOS (DML) EN SQL

Antes de realizar los ejercicios de este documento realizar las siguientes acciones:

1. Abrir una terminal, levantar la instancia de la BD.
2. Obtener el archivo `ejercicios-tema8-s01.sql` que se encuentra en la carpeta compartida BD. Abrir el archivo para revisar su contenido.
3. Abrir una nueva terminal y cambiarse al directorio donde se encuentra el archivo.
4. Entrar a `sqlplus`, no es necesario autenticarse. Es decir, ejecutar `sqlplus /nolog`
5. Activar el spool.
6. Ejecutar el archivo. El script solicitará los datos de un nuevo usuario:
`sql> start ejercicios-tema8-s01.sql`

Para los ejercicios de este documento, considerar el siguiente modelo relacional:



1.1. SENTENCIAS DML

Ejercicio 1.

- A. Generar las sentencias necesarias para guardar los datos de una nueva película: Título: Intinity, precio 89.34, fecha de recepción 10/10/2016 a las 5:22 hrs, emplear el valor por default para la fecha de registro (fecha actual). Como valor de sinopsis, proporcionar una descripción. Emplear la secuencia `SEQ_DVD`

Ejercicio 2.

- A. Crear una tabla `CLIENTE` con la misma estructura de la tabla `CLIENTE_IMPORT`. Copiar los datos de `CLIENTE_IMPORT` a `CLIENTE` empleando una sola sentencia SQL.

Se han detectado algunos problemas con los datos importados, realizar las siguientes acciones para corregir (solo en la tabla `CLIENTE`).

- Todos los clientes que tienen correo de gmail tienen el campo `NOMBRE` en minúsculas. Se requiere pasar estos campos a mayúsculas.
- La fecha de nacimiento de los clientes que nacieron en julio es incorrecta. El día se recorrió. Por ejemplo, la fecha 10/07/2010 debe ser 11/07/2010. Si la fecha original es 31/07/2010, la fecha correcta debe ser 01/08/2010. Corregir este problema.
- Se requiere que el valor del apellido materno sea 'N/A' para los empleados que no cuenten con apellido materno.
- Insertar un nuevo registro en `CLIENTE`, proponer los valores.
- Hacer `commit` para confirmar los cambios.

Ejercicio 3.

- A. Se desea que los cambios aplicados a la tabla `CLIENTE` sean reflejados en `CLIENTE_IMPORT`. Realizar las acciones necesarias para aplicar los cambios en `CLIENTE_IMPORT`.

Ejercicio 4.

- A. El DVD creado en el ejercicio 1 tiene datos incorrectos. El DVD es de Música, sus datos correctos son: título: 'Infinity Piano', el artista es Yiruma. Aplicar los cambios necesarios para corregir el error.
- B. Hacer `commit` para confirmar los cambios.

Ejercicio 5.

- A. Crear una tabla `cliente_export` con la misma estructura de la tabla `cliente`, en esta ocasión la nueva tabla debe estar vacía.
- B. Ejecutar el script `ejercicios-tema8-s02.sql` El archivo se encuentra en la carpeta BD.
- C. Hacer `commit` para confirmar los cambios.
- D. Ejecutar la siguiente instrucción para verificar el número de registros insertados:
`select count(*) from cliente_export.`
- E. Eliminar todos los registros de la tabla empleando la instrucción `delete`.
- F. ¿Qué sucede con los datos eliminados anteriormente si se cierra la ventana de SQL plus de forma incorrecta sin hacer `commit`? Realizar el ejercicio para comprobar la respuesta.
- G. Ejecutar nuevamente los incisos B,C,D, E, F La diferencia ahora es emplear la instrucción `truncate`. Explicar las diferencias encontradas.

1.2. CONTROL TRANSACCIONAL.Ejercicio 6:

Ejecutar las instrucciones del ejemplo incluido en los apuntes para comprobar el control transaccional que realiza una BD.

- A. Ejecutar el script `ejercicios-tema8-s03.sql` El archivo se encuentra en la carpeta BD
- B. Ejecutar las instrucciones de la tabla de eventos mostrada en los apuntes, discutir la salida de cada instrucción.

Ejercicio 7:

Niveles de aislamiento – Lecturas Confirmadas

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla `PROD`.

```
delete from prod;
insert into prod(prod_id,cantidad) values(1001,300);
insert into prod(prod_id,cantidad) values(1002,500);
insert into prod(prod_id,cantidad) values(1003,700);
commit;
```

- B. Ejecutar las instrucciones del ejemplo de lecturas confirmadas de los apuntes para validar el comportamiento transaccional.

Ejercicio 8:

Niveles de aislamiento – Lecturas repetibles.

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla `PROD`.

```
delete from prod;
insert into prod(prod_id,cantidad) values(1001,100);
commit;
```

- B. Ejecutar las instrucciones del ejemplo de lecturas repetibles de los apuntes para validar el comportamiento transaccional.

Ejercicio 9.

Niveles de aislamiento – Nivel serializable.

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla `PROD`.

```
delete from prod;  
insert into prod(prod_id,cantidad) values(1001,35);  
commit;
```

- B. Ejecutar las instrucciones del ejemplo para ilustrar el nivel de aislamiento Serializable de los apuntes.

Ejercicio 10

Bloqueos.

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla PROD.

```
delete from prod;  
insert into prod(prod_id,cantidad) values(1001,5);  
commit;
```

- B. Ejecutar las instrucciones del ejemplo incluido en los apuntes para ilustrar los conceptos de bloqueos compartidos y exclusivos en una base de datos.

Ejercicio 11

Reproducción del problema de asignación de asientos.

- A. Ejecutar las instrucciones de la tabla de eventos incluida en los apuntes que ilustran un problema "Lost Update" y la asignación de un mismo asiento 2 clientes diferentes. La tabla y datos iniciales ya fue cargada en el script inicial.

Ejercicio 12

Control de concurrencia pesimista empleando la cláusula `select for update`

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla CONCIERTO.

```
update concierto set ocupado = 0 , nombre = null;
```

- B. Ejecutar las instrucciones incluidas en la tabla de eventos de los apuntes empleada para resolver el problema anterior haciendo uso de la cláusula `select for update` (control de concurrencia pesimista).

Ejercicio 13

Control de concurrencia optimista empleando una sentencia `update` condicionada.

- A. Ejecutar las siguientes instrucciones para establecer las condiciones iniciales de la tabla CONCIERTO.

```
update concierto set ocupado = 0 , nombre = null;
```

- B. Ejecutar las instrucciones incluidas en la tabla de eventos de los apuntes empleada para resolver el problema anterior empleando la técnica de control de concurrencia optimista.