**1º DAM/DAW BASES DE DATOS**

1. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y controla las acciones que los empleados realizan sobre dicha tabla almacenando en la tabla "control" el nombre del usuario y la fecha, cada vez que se ingresa un registro en la tabla "libros".**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date

);

1. **Establecemos el formato de fecha para que muestre "DD/MM/YYYY HH24:MI":**

**alter session set NLS\_DATE\_FORMAT = 'DD/MM/YYYY HH24:MI';**

1. **Creamos un disparador que se dispare cada vez que se ingrese un registro en "libros"; el trigger debe ingresar en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó un "insert" sobre "libros".**
2. **Veamos qué nos informa el diccionario "user\_triggers" respecto del trigger anteriormente creado:**
3. **Ingresamos un registro en "libros".**
4. **Verificamos que el trigger se disparó consultando la tabla "control" para ver si tiene un nuevo registro.**
5. **Ingresamos dos registros más en "libros".**
6. **Verificamos que el trigger se disparó consultando la tabla "control" para ver si tiene dos nuevos registros.**
7. **Una librería almacena los datos de sus libros en una tabla denominada “libros” y en una tabla “ofertas”, algunos datos de los libros cuyo preco no supera los 30. Además, controla las inserciones que los empleados realizan sobre “ofertas”, almacenando en la tabla “control” el nombre del usuario, la fecha y hora, cada vez que se ingresa un nuevo registro en la tabla “ofertas”.**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table ofertas(

titulo varchar2(40),

autor varchar2(30),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date

);

1. **Establecer el formato de fecha para que muestre "DD/MM/YYYY HH24:MI".**
2. **Crear un disparador que se dispare cuando se ingrese un nuevo registro en "ofertas"; el trigger debe ingresar en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó un "insert" sobre "ofertas".**
3. **Mostrar la información del diccionario "user\_triggers" respecto del trigger anteriormente creado.**
4. **Insertar algunos registros en "libros".**
5. **Ingrese en "ofertas" los libros de "libros" cuyo precio no superen los 30€.**
6. **Verifique que el trigger se disparó consultando la tabla "control".**
7. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y en una tabla "ofertas", algunos datos de los libros cuyo precio no supera los $30. Además, controla las inserciones que los empleados realizan sobre "ofertas", almacenando en la tabla "control" el nombre del usuario, la fecha y hora, cada vez que se ingresa un nuevo registro en la tabla "ofertas".**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table ofertas(

titulo varchar2(40),

autor varchar2(30),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date

);

1. **Establecemos el formato de fecha para que muestre "DD/MM/YYYY HH24:MI".**
2. **Creamos un disparador que se dispare una vez por cada registro ingresado en "ofertas"; el trigger debe ingresar en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó un "insert" sobre "ofertas".**
3. **Veamos qué nos informa el diccionario "user\_triggers" respecto del trigger anteriormente creado.**
4. **Ingresamos algunos registros en "libros".**
5. **Ingresamos en "ofertas" los libros de "libros" cuyo precio no superen los 30, utilizando la siguiente sentencia.**
6. **Verificamos que el trigger se disparó 2 veces, una por cada fila afectada en la sentencia "insert" anteriormente ejecutada; consultamos la tabla "control".**
7. **Si el trigger hubiese sido creado a nivel de sentencia, no de fila, el "insert" anterior se hubiese activado una sola vez aun cuando se ingresaron 2 registros.**
8. **Una empresa almacena los datos de sus empleados en una tabla denominada "empleados" y en una tabla "control", el nombre del usuario y la fecha, cada vez que se ingresa un nuevo registro en la tabla "empleados".**

create table empleados(

documento char(8),

apellido varchar2(30),

nombre varchar2(30),

seccion varchar2(20)

);

create table control(

usuario varchar2(30),

fecha date

);

1. **Crear un disparador que se dispare una vez por cada registro ingresado en "empleados"; el trigger debe ingresar en la tabla "control", el nombre del usuario y la fecha en la cual se realizó un "insert" sobre "empleados".**
2. **Mostrar la información del diccionario "user\_triggers" respecto del trigger anteriormente creado**
3. **Insertar los siguientes registros en "empleados":**

insert into empleados values('22333444','ACOSTA','Ana','Secretaria');

insert into empleados values('22777888','DOMINGUEZ','Daniel','Secretaria');

insert into empleados values('22999000','FUENTES','Federico','Sistemas');

insert into empleados values('22555666','CASEROS','Carlos','Contaduria');

insert into empleados values('23444555','GOMEZ','Gabriela','Sistemas');

insert into empleados values('23666777','JUAREZ','Juan','Contaduria');

1. **Verificar que el trigger se disparó 6 veces, una por cada fila afectada en la sentencia "insert" anteriormente ejecutada; consultamos la tabla "control":**
2. **Si el trigger hubiese sido creado a nivel de sentencia, no de fila, el "insert" anterior se hubiese activado una sola vez aún cuando se ingresaron 6 registros.**
3. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y controla las acciones que los empleados realizan sobre dicha tabla almacenando en la tabla "control" el nombre del usuario y la fecha, cada vez que se elimina un registro de la tabla "libros".**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date

);

insert into libros values(97,'Uno','Richard Bach','Planeta',25);

insert into libros values(98,'El aleph','Borges','Emece',28);

insert into libros values(99,'Matematica estas ahi','Paenza','Nuevo siglo',12);

insert into libros values(100,'Aprenda PHP','Molina Mario','Nuevo siglo',55);

insert into libros values(101,'Alicia en el pais de las maravillas','Carroll','Planeta',35);

insert into libros values(102,'El experto en laberintos','Gaskin','Planeta',20);

1. **Establecemos el formato de fecha para que muestre "DD/MM/YYYY HH24:MI".**
2. **Creamos un disparador a nivel de fila, que se dispare cada vez que se borre un registro de "libros"; el trigger debe ingresar en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó un "delete" sobre "libros".**
3. **Eliminamos todos los libros cuyo código sea inferior a 100.**
4. **Veamos si el trigger se disparó consultando la tabla "control".**
5. **Se eliminaron 3 registros, como el trigger fue definido a nivel de fila, se disparó 3 veces, una vez por cada registro eliminado. Si el trigger hubiese sido definido a nivel de sentencia, se hubiese disparado una sola vez.**
6. **Reemplazamos el disparador creado anteriormente por otro con igual código pero a nivel de sentencia.**
7. **Eliminamos todos los libros cuya editorial sea "Planeta".**
8. **Se han eliminado 2 registros, pero el trigger se ha disparado una sola vez, consultamos la tabla "control".**
9. **Si el trigger hubiese sido definido a nivel de fila, se hubiese disparado dos veces.**
10. **Una empresa almacena los datos de sus empleados en una tabla denominada "empleados" y en otra llamada "control" guarda un registro por cada empleado que se elimina de la tabla "empleados".**

create table empleados(

documento char(8),

apellido varchar2(20),

nombre varchar2(20),

seccion varchar2(30),

sueldo number(8,2)

);

create table control(

usuario varchar2(30),

fecha date

);

insert into empleados values('22333444','ACOSTA','Ana','Secretaria',500);

insert into empleados values('22777888','DOMINGUEZ','Daniel','Secretaria',560);

insert into empleados values('22999000','FUENTES','Federico','Sistemas',680);

insert into empleados values('22555666','CASEROS','Carlos','Contaduria',900);

insert into empleados values('23444555','GOMEZ','Gabriela','Sistemas',1200);

insert into empleados values('23666777','JUAREZ','Juan','Contaduria',1000);

1. **Crear un disparador a nivel de fila, que se dispare cada vez que se borre un registro de "empleados"; el trigger debe ingresar en la tabla "control", el nombre del usuario y la fecha en la cual se realizó un "delete" sobre "empleados".**
2. **Mostrar la información del diccionario "user\_triggers" respecto del trigger anteriormente creado.**
3. **Eliminar todos los empleados cuyo sueldo supera los $800.**
4. **Verificar si el trigger se disparó consultando la tabla "control".**
5. **Crear un disparador similar al creado anteriormente por otro con igual código pero a nivel de sentencia.**
6. **Eliminar todos los empleados de la sección "Secretaria" y verificar si el trigger se disparó correctamente.**
7. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y controla las acciones que los empleados realizan sobre dicha tabla almacenando en la tabla "control" el nombre del usuario y la fecha, cada vez que se modifica el precio o la editorial de un libro.**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date

);

insert into libros values(100,'Uno','Richard Bach','Planeta',25);

insert into libros values(103,'El aleph','Borges','Emece',28);

insert into libros values(105,'Matematica estas ahi','Paenza','Nuevo siglo',12);

insert into libros values(120,'Aprenda PHP','Molina Mario','Nuevo siglo',55);

insert into libros values(145,'Alicia en el pais de las maravillas','Carroll','Planeta',35);

1. **Establezcer el formato de fecha para que muestre "DD/MM/YYYY HH24:MI".**
2. **Crear un desencadenador a nivel de sentencia que se dispare cada vez que se actualicen los campos "precio" y "editorial"; el trigger debe ingresar en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó un "update" sobre "precio" o "editorial" de "libros".**
3. **Aumentar en un 10% el precio de todos los libros de editorial "Nuevo siglo”.**
4. **Verificar cuántas veces se disparó el trigger consultando la tabla "control".**
5. **Cambiar la editorial, de "Planeta" a "Sudamericana"**
6. **Verificar si el trigger se disparó consultando la tabla "control"  
   Modificar un campo diferente de los que activan el trigger y Verificar que el cambio se realizó y el trigger no se disparó.**
7. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y controla las acciones que los empleados realizan sobre dicha tabla almacenando en la tabla "control" el nombre del usuario, la fecha, y el tipo de modificación que se realizó sobre la tabla "libros".**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date,

operacion varchar2(20)

);

insert into libros values(100,'Uno','Richard Bach','Planeta',25);

insert into libros values(103,'El aleph','Borges','Emece',28);

insert into libros values(105,'Matematica estas ahi','Paenza','Nuevo siglo',12);

insert into libros values(120,'Aprenda PHP','Molina Mario','Nuevo siglo',55);

insert into libros values(145,'Alicia en el pais de las maravillas','Carroll','Planeta',35);

1. **Establecemos el formato de fecha para que muestre "DD/MM/YYYY HH24:MI":**
2. **Creamos un disparador a nivel de sentencia, que se dispare cada vez que se ingrese, actualice o elimine un registro de la tabla "libros". El trigger ingresa en la tabla "control", el nombre del usuario, la fecha y la hora en la cual se realizó la modificación y el tipo de operación que se realizó:**
3. **Veamos qué nos informa el diccionario "user\_triggers" respecto del trigger anteriormente creado:**
4. **Ingresamos un registro en "libros":**
5. **Veamos si el trigger se disparó consultando la tabla "control":**
6. **Vemos que se ingresó un registro que muestra que el usuario "system", el día y hora actual realizó una inserción sobre "libros".**
7. **Actualizamos algunos registros de "libros":**
8. **Veamos cuántas veces el trigger se disparó consultando la tabla "control":**
9. **Vemos que se ingresaron 3 nuevos registros que muestran que el usuario "system", el día y hora actual actualizó tres registros de "libros". Si el trigger se hubiese definido a nivel de sentencia, el "update" anterior se hubiese disparado una sola vez.**
10. **Eliminamos un registro de "libros":**
11. **Veamos si el trigger se disparó consultando la tabla "control":**
12. **Vemos que se eliminó 1 registro.**
13. **Una librería almacena los datos de sus libros en una tabla denominada "libros" y controla las acciones que los empleados realizan sobre dicha tabla almacenando en la tabla "control" el nombre del usuario y la fecha, cada vez que se modifica el precio o la editorial de un libro.**

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

editorial varchar2(20),

precio number(6,2)

);

create table control(

usuario varchar2(30),

fecha date,

operacion varchar2(20)

);

insert into libros values(100,'Uno','Richard Bach','Planeta',25);

insert into libros values(103,'El aleph','Borges','Emece',28);

insert into libros values(105,'Matematica estas ahi','Paenza','Nuevo siglo',12);

insert into libros values(120,'Aprenda PHP','Molina Mario','Nuevo siglo',55);

insert into libros values(145,'Alicia en el pais de las maravillas','Carroll','Planeta',35);

1. **El gerente permite:**
   1. **ingresar o borrar libros de la tabla "libros" unicamente los sábados de 8 a 12 hs.**
   2. **actualizar los precios de los libros de lunes a viernes de 8 a 18 hs. y sábados entre la 8 y 12 hs.**
2. **Crear un disparador para los tres eventos que controle la hora en que se realizan las operaciones sobre "libros". Si se intenta eliminar, ingresar o actualizar registros de "libros" fuera de los días y horarios permitidos, debe aparecer un mensaje de error. Si la operación de ingreso, borrado o actualización de registros se realiza, se debe almacenar en "control", el nombre del usuario, la fecha y el tipo de operación ejecutada**
3. **Cambiar la fecha y hora del sistema a "domingo 19 hs." e ingresar un libro  
   Cambiar la fecha y hora del sistema a "lunes 10 hs." e ingresar un libro.  
   Cambiar la fecha y hora del sistema a "sabado 8 hs."e Ingrese un libro**
4. **Verificar que el trigger ha funcionado correctamente**
5. **Cambiar la fecha y hora del sistema a "domingo 18 hs." y modificar el precio de un libro.  
   Cambiar la fecha y hora del sistema a "sabado 15 hs." y modificar el precio de un libro.  
   Cambiar la fecha y hora del sistema a "sabado 9 hs.". Actualice el precio de un libro**
6. **Realizar un "select" sobre "libros" y sobre "control" para verificar que se han cargado los datos correspondientes.**
7. **Cambiar la fecha y hora del sistema a "martes 11:30 hs.". Actualizar el precio de un libro**
8. **Cambiar la fecha y hora del sistema a "domingo 18:30 hs.". Intente borrar un libro.  
   Cambiar la fecha y hora del sistema a "miercoles 15 hs.". Intente borrar un libro.  
   Cambiar la fecha y hora del sistema a "sábado 11:55 hs.". Borrar un libro**
9. **Realizar un "select" sobre "libros" y sobre "control" para verificar que se han cargado los datos correspondientes.**