IES Augusto González de Linares.

Tarea 6:

Programación de bases de datos.

BASES DE DATOS.

(DAM_BD)

Ciclo formativo

Desarrollo De Aplicaciones Multiplataforma

(DAM)

CURSO:

2021/2022

Francisco M. Colls Gutiérrez.

Actividad 1.

Crear un procedimiento que permita cambiar a todos los agentes de una familia determinada (familia origen) a otra familia (familia destino).

El procedimiento tendrá la siguiente cabecera CambiarAgentesFamilia (id_FamiliaOrigen, id_FamiliaDestino), donde cada uno de los argumentos corresponde a un identificador de Familia. Cambiará la columna Identificador de Familia de todos los agentes, de la tabla AGENTES, que pertenecen a la Familia con código id_FamiliaOrigen por el código id_FamiliaDestino

Previamente comprobará que ambas familias existen y que no son iguales.

Para la comprobación de la existencia de las familias se puede utilizar un cursor variable, o contar el número de filas y en caso de que no exista, se visualizará el mensaje correspondiente mediante una excepción del tipo RAISE_APPLICATION_ERROR. También se mostrará un mensaje en caso de que ambos argumentos tengan el mismo valor.

El procedimiento visualizará el mensaje "Se han trasladado XXX agentes de la familia XXXXXX a la familia ZZZZZZ" donde XXX es el número de agentes que se han cambiado de familia, XXXXXX es el nombre de la familia origen y ZZZZZZ es el nombre de la familia destino.

Código PL/SQL.

```
set SERVEROUTPUT on; --Activamos la salida de texto.

/*Creamos el procedimiento llamando CambiarAgentesFamilia

que recibe 2 parametros id_FamiliaOrigen y id_FamiliaDestino*/

create or replace PROCEDURE CambiarAgentesFamilia(id_FamiliaOrigen in familias.identificador%TYPE, id_FamiliaDestino in familias.identificador%TYPE)

is

   contador_agentes integer; --Declaramos una variable para almacenar los agentes modificados.

   Begin

   if (id_FamiliaOrigen=Id_FamiliaDestino) then --Comprobamos si ambas familias son iguales

      raise_application_error(-20000,'Las familias no pueden ser iguales'); --Si son iguales lanza un error.

   end if;

      -->>>Bloque SQL anidado.

      --Declaramos un cursor de tipo variable llamado cursor_Familia.

      Declare

      TYPE cursor_Familia is REF CURSOR RETURN familias%ROWTYPE;

      cFamilia cursor_Familia;

      familia cFamilia%ROWTYPE;
```

```
BEGIN
-- Iniciamos el cursor para comprobar que existe el id de Familia de origen.
OPEN cFamilia for select * from familias where identificador=id_FamiliaOrigen;
   FETCH cFamilia into familia;
   if (cFamilia%NOTFOUND) then --Si no existe, lanza un error
      raise_application_error(-20001,'Familia origen no existe');
   end if;
close cFamilia;
 -- Iniciamos el cursor para comprobar que existe el id de Familia de destino.
OPEN cFamilia for select * from familias where identificador=id_FamiliaDestino;
   FETCH cFamilia into familia;
   if (cFamilia%NOTFOUND) then --Si no existe, lanza un error
      raise_application_error(-20002,'Familia destino no existe');
   end if;
close cFamilia;
end;
  --Si ambas familias existen procedemos a actualizar el id.
update agentes SET familia=id_familiadestino where familia=id_familiaorigen;
 --Obtenemos el número de filas que fueron modificadas con ROWCOUNT.
contador_agentes:=sql%ROWCOUNT;
-- Si el número de filas modificadas es 0, eso quiere decir que no hay agentes en la familia de origen.
if (contador_agentes=0) then
   raise_application_error(-20003,'No hay agentes en la famila de origen');
end if;


dbms_output.put_line('Se han trasladado ' ||contador_agentes||' agentes de la familia ' || id_familiaorigen  || ' a la familia '||id_familiadestino);
end CambiarAgentesFamilia;
```

Para comprobar el procedimiento ejecutamos la siguiente sentencia:

```
execute  CambiarAgentesFamilia(&Familia_Origen,&Familia_Destino);
```

Francisco M. Colls Gutiérrez.

Actividad 2.

Queremos controlar algunas restricciones a la hora de trabajar con agentes:

La longitud de la clave de un agente no puede ser inferior a 6.

La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).

La categoría de un agente sólo puede ser igual a 0, 1 o 2.

Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.

Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.

Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.

Código PL/SQL.

```
--Creamos un trigger llamado t_agentes.

create or replace trigger t_agentes

before insert or UPDATE on agentes --El trigger se activa antes de insertar o actualizar la tabla agentes.

for each row

begin

--  La longitud de la clave de un agente no puede ser inferior a 6.

   if (length(:new.clave)<6) then --Si la longitud es menor a 6 lanza un error

      raise_application_error(-20004,'La longitud de la clave de un agente no puede ser inferior a 6.');

   end if;

-- La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).

   if (:new.habilidad <0 OR :new.habilidad >9) then

      raise_application_error(-20005,'La habilidad de un agente debe estar comprendida entre 0 y 9');

   end if;

--La categoría de un agente sólo puede ser igual a 0, 1 o 2.

   if (:new.categoria <0 OR :new.categoria >2) then

      raise_application_error(-20006,'La categoría de un agente sólo puede ser igual a 0, 1 o 2.');

   end if;
```

Francisco M. Colls Gutiérrez.

```
-- Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
 if (:new.categoria=2) then
        if :new.familia is not null then
            raise_application_error(-20007,'No puede pertenecer a ninguna familia ');
        elsif :new.oficina is null then
            raise_application_error(-20008, 'Debe pertenecer a una oficina.');
        end if;
    end if;
    -- Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.
    if (:new.categoria=1) then
        if (:new.oficina is not null) then
            raise_application_error(-20009,'No puede pertenecer a ninguna oficina ');
        elsif (:new.familia is null) then
            raise_application_error(-20010, 'Debe pertenecer a una familia');
        end if;
    end if;
    -- Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.
    if (:new.oficina is null and :new.familia is null) then
        raise_application_error(-20011,'Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez');
    elsif (:new.oficina is not null and :new.familia is not null) then
        raise_application_error(-20012,'Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez');
    end if;

end t_agentes;
```

Francisco M. Colls Gutiérrez.

Algunas de las restricciones implementadas con el disparador se pueden incorporar a la definición del esquema de la tabla utilizando el Lenguaje de Definición de Datos (Check,Unique,..).Identifica cuáles son y con qué tipo de restricciones las implementarías.

-- La longitud de la clave de un agente no puede ser inferior a 6.

alter table agentes add constraint chk_clave check(LENGTH(clave) > 6);

-- La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).

alter table agentes add constraint chk_habilidad check (habilidad BETWEEN 0 and 9);

--La categoría de un agente sólo puede ser igual a 0, 1 o 2.

alter table agentes add constraint chk_categoria check (categoria BETWEEN 0 and 2);

--Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.

--Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.

--Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.

*/ Intente realizar 3 restricciones para los casos anteriores, pero generaba un error por lo cual agrupe las restricciones en una sola, y de esa forma si pude ejecutar la restricción */

alter table agentes add constraint chk_categoria2 check ((categoria=2 and familia is null and oficina is not null) OR (categoria=1 and familia is not null and oficina is null) or ((familia is null and oficina is not null) OR (familia is not null and oficina is null)));

Francisco M. Colls Gutiérrez.