

PROCEDIMIENTOS en PL/SQL

Valeria Beratto Ulloa

Qué son?

- Un procedimiento es un **conjunto de sentencias SQL , PL/SQL y de control de flujo**, que se almacena en la BD.
- Existen dos tipos:
 - Trigger
 - Procedimientos Almacenados (Procedure Store)

TRIGGER

Un trigger es un **procedimiento invocado automáticamente** por el SGBD en respuesta a una cambio específico en la BD

Una BD que tiene un conjunto de triggers asociados se denomina **Base de Datos Activa (BDA)**

TRIGGER

Un trigger está compuesto por tres partes:

- **Evento:** Un cambio en la BD que activa el trigger
- **Condición:** Una consulta o test que se ejecuta cuando el trigger es activado
- **Acción:** Un procedimiento que es ejecutado cuando el trigger es activado y la condición es verdadera

- Las operaciones que pueden activar un trigger son: **UPDATE, DELETE, INSERT**
- En un trigger se hace referencia a las nuevas tuplas con la variable *new*, y a las antiguas con la variable *old*
- Una acción en un trigger puede involucrar la modificación de la BD

Sintaxis

```
CREATE [OR REPLACE] TRIGGER <nombre_trigger>
{BEFORE|AFTER}      {DELETE|INSERT|UPDATE [OF col1, col2,
..., colN]
[OR {DELETE|INSERT|UPDATE [OF col1, col2, ...,
colN]...}]
ON <nombre_tabla>

[FOR EACH ROW [WHEN (<condicion>)]]
DECLARE
    -- variables locales
BEGIN
    -- Sentencias
[EXCEPTION]
    -- Sentencias control de excepción
END <nombre_trigger>;
```

Manejo de Variables :OLD y :NEW

ACCION SQL	OLD	NEW
INSERT	No definido; todos los campos toman valor NULL.	Valores que serán insertados cuando se complete la orden.
UPDATE	Valores originales de la fila, antes de la actualización.	Nuevos valores que serán escritos cuando se complete la orden.
DELETE	Valores, antes del borrado de la fila.	No definidos; todos los campos toman el valor NULL.

Ejemplo

- Dado el esquema de los navegantes
 - *Navegantes* (*idN*, *nombreN*, *categoría*, *edad*, *ciudad*)
 - *Reservas* (*idN*, *idB*, *fecha*)
 - *Botes* (*idB*, *nombreB*, *color*)
-
- Genere un trigger que controle que todos los botes que sean reservados, existan en la base de datos, de lo contrario, se debe ingresar el bote en la tabla correspondiente.

Resolviendo

1. Crear trigger
2. Qué evento gatilla el trigger?
3. Qué debo condición se debe cumplir?
4. Qué Acción debo realizar?
5. Variables a declarar

```
CREATE OR REPLACE TRIGGER TR_RESERVASINBOTE
AFTER INSERT ON RESERVA FOR EACH ROW

DECLARE
contador number;

BEGIN
SELECT COUNT(*) INTO contador
FROM botes
WHERE idb = :new.idb;

IF contador = 0 THEN

    INSERT INTO BOTES (IDB, NOMBREB, COLOR)
VALUES (:NEW.IDB, null, null);

END IF;
END;
```

Habilitar/Deshabilitar

- **ALTER TRIGGER Nombre_trigger DISABLE**
- **ALTER TRIGGER Nombre_trigger ENABLE**

Orden de ejecución de los triggers

- Una misma tabla puede tener varios triggers. En tal caso es necesario conocer el orden en el que se van a ejecutar.
- Los disparadores se activan al ejecutarse la sentencia SQL.
- Si existe, se ejecuta el disparador de tipo BEFORE (disparador previo) con nivel de orden.
- Para cada fila a la que afecte el orden:
 - Se ejecuta si existe, el disparador de tipo BEFORE con nivel de fila.
 - Se ejecuta la propia orden.
 - Se ejecuta si existe, el disparador de tipo AFTER (disparador posterior) con nivel de fila.
- Se ejecuta, si existe, el disparador de tipo AFTER con nivel de orden.

Restricciones

- No son RECURSIVOS.
- No se pueden hacer tareas que afecten a la misma tabla que está gestionando el trigger en curso.
- No se puede definir más de un trigger para el MISMO evento en la misma tabla.

EJEMPLO

Qué hace?

```
CREATE OR REPLACE TRIGGER TR_RESERVA AFTER INSERT ON Reserva
FOR EACH ROW
BEGIN
    :new.fecha := sysdate;
    INSERT INTO Reserva (IDN,IDB,FECHA)
    VALUES (:NEW.IDN,:NEW.IDB,SYSDATE) ;

END ;
```

Ejercicio:

Genere un trigger que controle que no se puede eliminar un bote, en caso que tenga reservas asociadas

```
CREATE OR REPLACE TRIGGER BorrarBOTES
BEFORE DELETE ON BOTES FOR EACH ROW

DECLARE

contador number;

BEGIN

SELECT COUNT(*) INTO CONTADOR FROM RESERVAS WHERE IDB = :OLD.IDB;

    IF contador >0 THEN

        Raise_Application_Error(-20120, 'NO SE PUEDE ELIMINAR');

    END IF;

END;
```

- Un trigger nos ayuda a mantener la BD consistente, así como también las restricciones en SQL
- Las restricciones no son definidas de una manera procedural, sino que declaradas al momento de creación de una relación
- Un trigger es un procedimiento que nos permite ejecutar acciones en el caso de que una restricción es violada
- Un trigger por lo tanto, nos permite implementar **ciertas opciones** ante el cambio en la BD, que dejan la BD consistente
- En cierto grado, los triggers son más flexibles que las restricciones

Ejercicios

- (1) Genere un trigger que no permita reservar a un cliente el mismo bote durante el mismo mes.

```
CREATE OR REPLACE TRIGGER TR_CLIE_RESV BEFORE INSERT ON Reservas
FOR EACH ROW
DECLARE

fecha_res DATE;
cursor c_fecha IS SELECT fecha FROM RESERVAS
WHERE IDN = :new.IDN
AND IDB = :new.IDB ;

BEGIN
OPEN c_fecha;
FETCH c_fecha INTO fecha_res;
WHILE c_fecha%FOUND
LOOP
    IF TO_CHAR(fecha_res, 'MM') = TO_CHAR(:new.fecha, 'MM') THEN
        raise_application_error (-20300, 'TIENE RESERVA ESTE MES');
    END IF;
    FETCH c_fecha INTO fecha_res;
END LOOP;
CLOSE c_fecha;
END;
```


Ejercicio 2

Genere un trigger que controle que un navegante no puede tener más de 5 reservas en el mismo mes

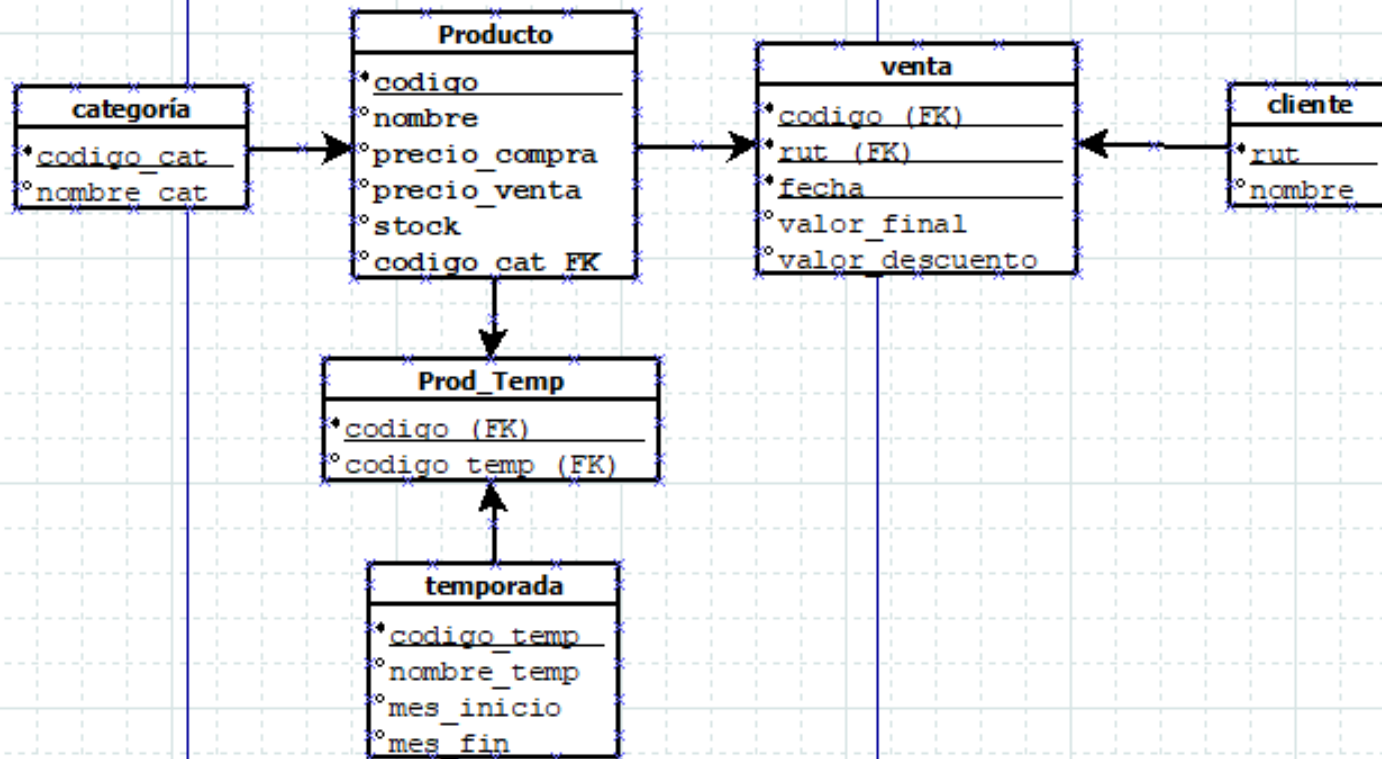
```
CREATE OR REPLACE TRIGGER TR_CANT_RESV BEFORE INSERT ON Reservas  
FOR EACH ROW
```

```
DECLARE  
cantidad number;
```

```
BEGIN  
SELECT count(*) into cantidad FROM RESERVAS  
WHERE IDN = :new.idn  
AND TO_CHAR(fecha, 'MM') = TO_CHAR(:new.fecha, 'MM');  
  
IF cantidad >=5 THEN  
    raise_application_error (-20300, 'TIENE MUCHAS RESERVAS ESTE MES');  
END IF;  
  
END;
```

Ejercicios a resolver

Enunciado Zapatería



Ejercicios a resolver

Enunciado Zapatería

- Ejercicio 1: Es importante notar que los productos que se venden en temporada tienen un recargo en el costo de 20% sobre el precio de venta, considerando como valor final el valor de venta del producto más el sobrecargo de temporada. Cree un trigger que permita calcular el precio final y registrarlo en la venta.
- Ejercicio 2: Si el cliente ha adquirido más de cinco productos en el mes anterior se le debe realizar un 5% de descuento que podrá utilizar en las compras del próximo mes. El valor de descuento se debe comparar con el valor de descuento ingresado, en caso de existir diferencia se debe reemplazar el valor de descuento ingresado.