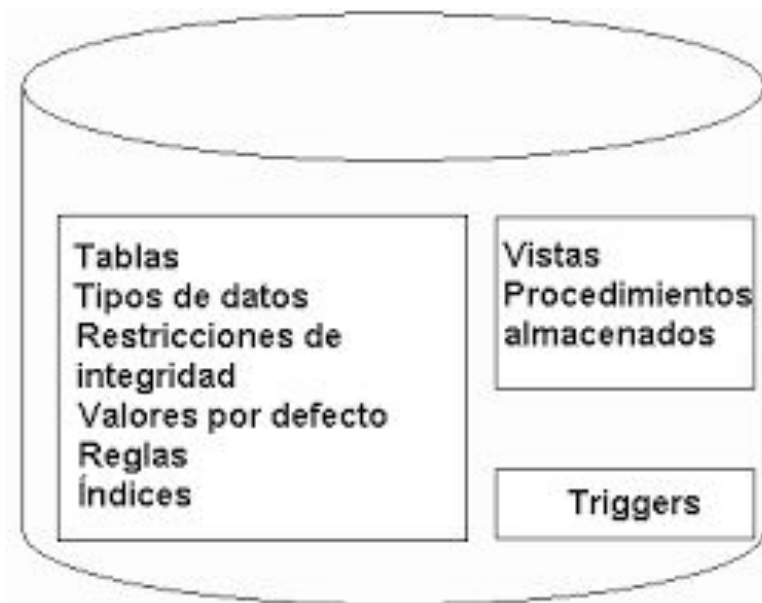
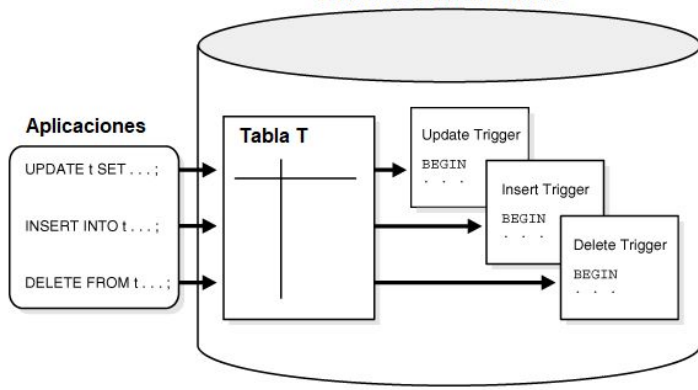


Trigger (Disparador). Objetivo

Construir **triggers** para la manipulación de datos.





Trigger (Disparador)

Muchas veces es importante **proteger** nuestras tablas y registros de acciones accidentales o procesos que como diseñadores no queremos que se realicen, una forma de lograr este objetivo es el **uso de Triggers**.

Es importante mencionar que se deben tener los permisos necesarios para hacer algunos procesos, y algunos de los procesos que signifiquen transacciones muy peligrosas deben de ser realizadas con mucho cuidado.

Trigger (Disparador). Usan para...

- Mantener la integridad referencial.



Pueden controlar ciertas acciones, por ejemplo: pueden rechazar o modificar ciertos valores que no cumplan con determinadas reglas, para prevenir que datos inválidos sean insertados en la base.

- Asegurar reglas de negocio complejas.



Cuando ciertas reglas de negocio son muy elaboradas y necesitan ser expresadas a nivel base de datos, es necesario las restricciones.

- Auditar cambios en los datos.



Los triggers se usan para llenar tablas de auditoría, en donde se registren ciertos tipos de transacciones o accesos hacia tablas.

-- ALGORITMO DE UN DESENCADENADOR

USA LA BASE DE DATOS SISA

CREA DESENCADENADOR EstableceEstadoAlumno

SOBRE PromedioAsignatura

DESPUES DE ACTUALIZAR

INICIO

ESTABLECE AlumnoActualizado = SELECCIONA Matricula

DE PromedioAsignaturaTemporalConDatosAnteriores

SELECCIONA NumeroAsignaturasReprobadas = CuentaCalificacionesReprobadas

DE PromedioAsignatura

DONDE Matricula = AlumnoActualizado Y Calificacion = 'NA'

SI NumeroAsignaturasReprobadas > 2 ENTONCES

ACTUALIZA Alumno ESTABLECE estado = 'Reprobado'

DONDE Matricula = AlumnoActualizado

FIN

Trigger (Disparador)

Un "trigger" (disparador o desencadenador) es un bloque de código que se ejecuta automáticamente cuando ocurre algún evento como **insert**, **update** o **delete** sobre una determinada tabla.



- ✓ No pueden ser invocados directamente.
- ✓ No reciben y retornan parámetros.
- ✓ Son apropiados para mantener la integridad de los datos.

Se agrega "or replace" al momento de crearlo, sí existe un trigger con el mismo nombre, el trigger será borrado y vuelto a crear.

CREATE [OR REPLACE] TRIGGER NOMBRETRIGGER

BEGIN

MOMENTO (BEFORE, AFTER o INSTEAD OF) , EVENTO (INSERT, UPDATE o DELETE)

SENTENCIAS A EJECUTAR

ON NO

Momento (indica

disparará en

NIVEL

event

WHEN CONDIC

Before (Antes de): se activará antes que el evento (insert, update o delete) sobre la tabla del mismo.

After (Despues de): se activará después de la operación que causó la activación.

Instead of (En vez de): sólo puede definirse sobre vistas, anula la sentencia disparadora, se ejecuta en lugar de tal sentencia (ni antes ni después).

Trigger (Disparador). Sintaxis

Indica la tabla asociada al trigger.

For each row: (Por cada fila), es decir se activa una vez por cada registro afectado por la operación sobre la tabla.

Statement: (Declaración) Se activa una sola vez; antes o después de ejecutar la operación sobre la tabla.

Evento: Especifica la operación que causa que el trigger se active, puede ser "insert", "update" o "delete".

A nivel de sentencia o de fila.

Trigger (Disparador). Sintaxis

```
CREATE [OR REPLACE] TRIGGER nombre  
{BEFORE | AFTER | INSTEAD OF}  
{INSERT | DELETE | UPDATE [OF ]} ON  
[REFERENCING {NEW AS ..., OLD AS ...}+]  
[FOR EACH ROW] [WHEN condición]  
BEGIN  
    cuerpo del disparador (bloque PL/SQL)  
END;
```


Trigger (Disparador). Considera que...

- Cuando se trabaja con un trigger a **nivel de fila**, Oracle provee dos tablas temporales a las cuales se puede acceder.
- Estas tablas contienen los **antiguos** y **nuevos** valores de los campos del registro **afectado por la sentencia que activó el trigger**.
- Para especificar el campo se utiliza la siguiente sintaxis:

:new.CAMPO

:old.CAMPO

Trigger (Disparador) Considera que...

Tipo de evento	Campo accedido
Insert	:new
update	:new y :old
delete	:old

Los valores de "old" y "new" están disponibles en triggers after y before.

Trigger (Disparador) Considera que...

- El valor de ":new" puede modificarse en un **trigger before**, es decir, se puede acceder a los nuevos valores antes que se ingresen en la tabla y cambiar los valores asignando a ":new.CAMPO" otro valor.
- El valor de ":new" **NO** puede modificarse en un **trigger after**, esto es porque el trigger se activa luego que los valores de "new" se almacenaron en la tabla.
- El campo ":old" **nunca** se modifica, sólo puede leerse.

Trigger (Disparador)

- En los triggers a nivel de fila, se **puede incluir una restricción adicional**, agregando la cláusula "**when**" con una condición que se evalúa para cada fila que afecte el trigger; si resulta **cierta**, se **ejecutan** las sentencias del trigger para ese registro; si resulta **falsa**, el trigger **no se activa** para ese registro.

Trigger (Disparador)

Un trigger puede estar en dos estados: habilitado o deshabilitado.

- Sintaxis para deshabilitar un trigger:

```
alter trigger nombretrigger disable;
```

- Sintaxis para habilitar un trigger que está deshabilitado:

```
alter trigger nombretrigger enable;
```

Trigger (Disparador)

- Para eliminar un trigger se emplea la siguiente sintaxis:

```
drop trigger nombretrigger;
```

- Si se elimina una tabla, se eliminan todos los triggers establecidos sobre ella.

Trigger (Disparador). Ejemplo

El dueño de la base de datos EMPRESA, ha solicitado que todos los empleados que se den de alta en la tabla EMPLEADO, pertenecerán al proyecto Reorganización, esto para efectos de cumplir con la restricción de que todo empleado debe pertenecer al menos a un proyecto.

Podríamos utilizar un desencadenador **AFTER** asociado a la tabla EMPLEADO, el cual, al detectar una operación de inserción en la misma asocie este nuevo empleado al proyecto Reorganización.

Trigger (Disparador). Ejemplo

```
CREATE TRIGGER tr_empleado_inserta AFTER INSERT
ON EMPLEADO
BEGIN
    DECLARE @nuevo_empleado_dni CHAR(9);
    -- Seleccionar el nuevo DNI del empleado que se inserto
    SELECT @nuevo_empleado_dni = (SELECT Dni FROM inserted);
    PRINT 'Antes de ejecutar la asociación de ' + @nuevo_empleado_dni + ' con proyecto por defecto'
    -- Asociar el nuevo empleado con el proyecto 20
    INSERT INTO TRABAJA_EN (DniEmpleado, NumProy, Horas)
    VALUES (@nuevo_empleado_dni, 20, NULL);
    PRINT 'Después de ejecutar la asociación de ' + @nuevo_empleado_dni + ' con proyecto por defecto'
END
```


Trigger (Disparador). Ejemplo

Una vez creado el desencadenador, ejecutaremos una inserción de un empleado en la tabla EMPLEADO, de la siguiente manera:

```
INSERT INTO EMPLEADO(Nombre, Apellido1, Apellido2, Dni, FechaNac, Direccion, Sexo, Sueldo, SuperDni, Dno)
VALUES('Tomas', 'Arteaga', 'Lavín', '136903690', '05-10-1980', 'El Vajio, 17', 'H', 29000, '888665555', 1);
```

Justo después de ejecutar la instrucción anterior, el desencadenador `tr_empleado_inserta` se ejecutará automáticamente, permitiendo así asociar el empleado Tomás Arteaga con el proyecto Reorganización.