




Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

# Triggers en Oracle

*Grupo de Ingeniería del Software y Bases de Datos  
Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Sevilla  
Diciembre 2016*

© Departamento de Ingeniería Informática, 2016



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Introducción


- **Objetivos de la práctica**
  - Conocer qué son los triggers y para qué se utilizan
  - Aprender a construir triggers
  - Consultar cómo están almacenados y cómo se ven las propiedades de los triggers

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

1

© Departamento de Ingeniería Informática, 2011



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Ejercicios


- **Trigger**, o disparador, es un objeto procedimental asociado a una tabla.
- Está construido por bloques PL/SQL y sentencias SQL.
- Se ejecuta cuando se pretende realizar una determinada instrucción SQL sobre dicha tabla.
- Los Triggers frente a las Restricciones de Integridad, no se aplican a los datos almacenados en la base de datos antes de su definición; sólo se aplican cuando, una vez creados, se ejecutan comandos que manipulan las tablas sobre las que están definidos

1. Introducción  
2. Creación de triggers  
3. Expresiones  
4. Activación  
5. Documentación  
6. Ejercicios

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

2



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Ejercicios


- Un *trigger* tiene muchas aplicaciones entre las que cabe destacar:
  - Forzar reglas de integridad que son difíciles de definir mediante *constraints*.
  - Realizar cambios en la base de datos de forma transparente al usuario.
  - Sincronización entre tablas generando automáticamente valores de columnas derivadas en base a un valor proporcionado por una sentencia INSERT o UPDATE.
  - Derivar valores de columna automáticamente, etc..

1. Introducción  
2. Creación de triggers  
3. Expresiones  
4. Activación  
5. Documentación  
6. Ejercicios

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

3




Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios

## Introducción

- Ejemplos:
  - Campo autocalculado con la duración de un contrato:  
Fecha Fin – Fecha Inicio
  - Campo autocalculado con la cuantía de la Señal:  
PrecioAlquiler\*2
  - Validación de campos: La comisión del empleado debe estar entre el 50% y el 30% del valor del alquiler

noviembre 2011
Introducción a la Ingeniería del Software y a los Sistemas de Información
4



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios

## Creación de Triggers


Para definir los triggers se utiliza el comando SQL:

- **CREATE OR REPLACE TRIGGER**

La definición de un trigger tiene tres partes:

- **Comando:**
  - Se especifica el comando DML de SQL que dispara el Trigger (Delete, Insert o Update), y la tabla asociada.
  - Se puede especificar cuándo se dispara el Trigger:
    - BEFORE, antes de ejecutar el comando,
    - AFTER, después de ejecutar el comando.
- **Restricción:**
  - Se define la condición que tiene que verificar cada fila de la tabla para que se ejecute la acción del trigger
- **Acción:**
  - Se define mediante comandos y sentencias SQL y PL/SQL, la tarea específica que realiza el Trigger.

noviembre 2011
Introducción a la Ingeniería del Software y a los Sistemas de Información
5



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios


### Sintaxis de triggers

- Sintaxis de triggers:

```

CREATE [OR REPLACE] TRIGGER <nombre_trigger>
{BEFORE|AFTER}
    {DELETE|INSERT|UPDATE [OF <col1>, ..., <colN>]
    [OR
    {DELETE|INSERT|UPDATE [OF <col1>, ..., <colN>]]}
ON table <nombre_tabla>
    [FOR EACH ROW [WHEN (<condicion>)]]
          
```

noviembre 2011
Introducción a la Ingeniería del Software y a los Sistemas de Información
6




Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios

### Expresiones

- Para referirse al valor nuevo y al antiguo de una columna de una fila de la tabla, se utilizan los prefijos **:OLD** y **:NEW**
- Cuando estamos haciendo una modificación (UPDATE) de una fila podemos referirnos al valor antes de ser modificado (:OLD) y al valor después de la modificación (:NEW).
- Al introducir valores nuevos (INSERT) podemos referenciar sólo el valor nuevo (:NEW).
- Al borrar (DELETE) podemos referenciar sólo el valor antiguo (:OLD).
- Ejemplo. Compara si el salario ha aumentado en más del 25% de salario antiguo:  
:NEW.salario > (:OLD.salario\*1.25)

noviembre 2011
Introducción a la Ingeniería del Software y a los Sistemas de Información
7



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios


### Activación de triggers

- Un trigger puede estar activado (**ENABLE**) o desactivado (**DISEABLE**).
- Cuando está desactivado no ejecuta ninguna acción.
- Hay dos comandos SQL para activar un trigger :
  - ALTER TRIGGER nombre\_trigger {ENABLE|DISEABLE};
  - ALTER TABLE nombre\_tabla {ENABLE|DISEABLE} ALL TRIGGERS;
- Cuando se crean están activos por defecto.
- Es conveniente desactivar los triggers sobre una tabla cuando se hace una carga masiva de datos.

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

8



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios


### Documentación de triggers

- **Documentación de triggers.**
  - Se pueden consultar las vistas:
    - USER\_TRIGGERS
    - ALL\_TRIGGERS
    - DBA\_TRIGGERS

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

9



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios


- Crear una tabla EMPLEADOS.

```
CREATE TABLE empleados
(dni      char(9) PRIMARY KEY,
nomemp   varchar2(50),
jefe     char(9),
departamento integer,
salario  number(9,2) DEFAULT 1000,
usuario  varchar2(50),
fecha   date,
FOREIGN KEY (jefe) REFERENCES empleados (dni)
);
```

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

10



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios

### Ejercicio 1.

- Crear un trigger sobre la tabla EMPLEADOS para que no se permita que un empleado sea jefe de más de cinco empleados.


```
CREATE OR REPLACE TRIGGER jefes
BEFORE
INSERT ON empleados
FOR EACH ROW
DECLARE
supervisa INTEGER;
BEGIN
SELECT count(*) INTO supervisa
FROM empleados WHERE jefe = :new.jefe;
IF (supervisa > 4)
THEN raise_application_error
(-20600,:new.jefe||'no se puede supervisar más de 5');
END IF;
END;
```

Inserte las tuplas necesarias para comprobar que funcional el trigger

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

11



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios

### Ejercicio 2.


- Crear un trigger para impedir que se aumente el salario de un empleado en más de un 20%.
  - Es necesario comparar los valores *:old.salario* y *:new.salario* cada vez que se modifica el atributo **salario** (BEFORE UPDATE).

IF :NEW.salario > :OLD.salario\*1.20  
THEN raise...

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

12



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios


```
CREATE OR REPLACE TRIGGER aumentoSalario
BEFORE UPDATE OF salario ON empleados
FOR EACH ROW
BEGIN
    IF :NEW.salario > :OLD.salario*1.20
    THEN raise_application_error
        (-20600,'new.Salario||"no se puede aumentar el
        salario más de un 20%');
    END IF;
END;
```

Inserte las tuplas necesarias para comprobar que funcional el trigger

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

13



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios

### Ejercicio 3.

- Crear una tabla empleados\_baja con la siguiente estructura:


```
CREATE TABLE empleados_baja (  
  dni      char(9) PRIMARY KEY,  
  nomemp   varchar2(50),  
  jefe     char(9),  
  departamento integer,  
  salario  number(9,2) DEFAULT 1000,  
  usuario  varchar2(50),  
  fecha    date  
);
```

- Crear un trigger que inserte una fila en la tabla empleados\_baja cuando se borre una fila en la tabla empleados.

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

14



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios

- Los datos que se insertan son los del empleado que se da de baja en la tabla empleados, salvo en las columnas usuario y fecha se grabarán las variables del sistema USER y SYSDATE que almacenan el usuario y fecha actual.
- El comando que dispara el trigger es AFTER DELETE.


```
CREATE OR REPLACE TRIGGER bajas  
  AFTER DELETE ON empleados  
  FOR EACH ROW  
BEGIN  
  
  INSERT INTO empleados_baja VALUES  
    (:old.dni,:old.nomemp,:old.jefe,:old.departamento,  
    :old.salario, USER, SYSDATE);  
  
END;
```

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

15





Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios


### Ejercicio 4.

- Crear un trigger para impedir que, al insertar un empleado, el empleado y su jefe puedan pertenecer a departamentos distintos.

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

16



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

## Ejercicios

```
CREATE OR REPLACE TRIGGER MISMO_DEP
BEFORE INSERT ON EMPLEADOS
FOR EACH ROW
DECLARE
dept_jefe INTEGER;


BEGIN

IF (:NEW.jefe IS NOT NULL) THEN
  SELECT departamento INTO dept_jefe FROM empleados WHERE
dni=:NEW.jefe;
  IF (dept_jefe <> :NEW.departamento) THEN
    raise_application_error
      (-20600,:NEW.departamento||' Un empleado y su jefe no pueden
pertener a departamentos diferentes');
  END IF;
END IF;
END;
```

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

17



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Ejercicios

### Ejercicio 5.


- Crear un trigger para impedir que, al insertar un empleado, la suma de los salarios de los empleados pertenecientes al departamento del empleado insertado supere los 10.000 euros.

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

18



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Ejercicios

```
CREATE OR REPLACE TRIGGER sumaDept
BEFORE
INSERT ON empleados
FOR EACH ROW
DECLARE
    suma INTEGER;
BEGIN
    SELECT SUM(salario) INTO suma FROM empleados
    WHERE departamento=:NEW.departamento;

    suma := suma + :NEW.salario;


    IF (suma > 10000) THEN
        raise_application_error
        (-20600,:NEW.departamento||' La suma de salarios no puede ser
        superior a 10000');
    END IF;
END;
```

1. Introducción
2. Creación de triggers
3. Expresiones
4. Activación
5. Documentación
6. Ejercicios

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

19



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Creación de triggers
- 3. Expresiones
- 4. Activación
- 5. Documentación
- 6. Ejercicios

## Ejercicios

### Ejercicio 6.

- Visualizar los trigger definidos sobre una tabla consultando la **vista ALL- TRIGGERS**.
  - DESC ALL- TRIGGERS
  - SELECT trigger\_name, status FROM ALL- TRIGGERS WHERE table\_name = 'empleados';
- Desactivar (DISABLE) y activar (ENABLE) los trigger definidos sobre una tabla:
  - ALTER TABLE empleados DISABLE ALL TRIGGERS;
- Activar y desactivar un trigger específico:
  - ALTER TRIGGER jefes DISABLE;
- Ver la descripción de un trigger:
  - SELECT description FROM USER- TRIGGERS WHERE trigger\_name = 'JEFES';
- Ver el cuerpo de un trigger:
  - SELECT trigger\_body FROM USER- TRIGGERS WHERE trigger\_name = 'JEFES';

noviembre 2011

Introducción a la Ingeniería del Software y a los Sistemas de Información

20