

# Bases de Datos

Clase 10: Triggers

# Triggers

Acciones que realiza el sistema al chequearse una condición

Siguen la estructura Evento - Condición - Acción

Mucho más procedurales que otras restricciones

# Triggers

## Ejemplo

No podemos cambiar el largo de una película

```
CREATE TRIGGER NoCambioLargo
AFTER UPDATE OF largo ON Películas
REFERENCING
    OLD ROW AS TuplaAntigua
    NEW ROW AS TuplaNueva
FOR EACH ROW
WHEN (TuplaAntigua.largo <> TuplaNueva.largo)
SET largo = TuplaAntigua.largo
WHERE titulo = TuplaNueva.titulo
```

# Triggers

## Ejemplo

Este es el evento:

**AFTER UPDATE OF** largo ON Películas

Corresponde a modificar el atributo largo de una tupla en la tabla Películas

# Triggers

Ejemplo

Declaramos como nos referimos a la tupla antes y después del cambio

## REFERENCING

OLD ROW AS TuplaAntigua

NEW ROW AS TuplaNueva

# Triggers

## Ejemplo

Trigger se ejecuta una vez por fila, la condición es que el largo haya cambiado su valor

```
FOR EACH ROW
```

```
WHEN (TuplaAntigua.largo <> TuplaNueva.largo)
```

# Triggers

## Ejemplo

Ejecutamos la acción, que corresponde a restaurar el largo antiguo

```
SET largo = TuplaAntigua.largo  
WHERE titulo = TuplaNueva.titulo
```

# Triggers

## Ejemplo

Suponga la tabla Empleados(RUT, rango, sueldos):  
queremos que el sueldo promedio de los gerentes no exceda los 3 millones de pesos

Una operación que cambie varias tuplas puede elevar el sueldo de un gerente, para después bajarlo

No podemos usar **FOR EACH ROW**



# Triggers

## Ejemplo

Solución:

```
CREATE TRIGGER SueldoPromedio
AFTER UPDATE OF sueldo ON Empleados
REFERENCING
    OLD TABLE AS TuplasAntiguas
    NEW TABLE AS TuplasNuevas
FOR EACH STATEMENT
WHEN (3000000 > (SELECT AVG(sueldo)
                FROM Empleados WHERE rango = 'gerente'))
BEGIN
    DELETE FROM Empleados
    WHERE (run,rango,salario) in TuplasNuevas;
    INSERT INTO Empleados (SELECT * FROM TuplasAntiguas)
END;
```

# Triggers

Trigger malvado

¿Por qué el siguiente Trigger no termina?

```
CREATE TRIGGER Maldad
AFTER UPDATE OF precio ON Productos
REFERENCING
    OLD TUPLE AS TuplaAntigua
    NEW TUPLE AS TuplaNueva
FOR EACH ROW
WHEN (TuplaNueva.precio > 50)
UPDATE Productos
SET precio = TuplaNueva.precio * 2
WHERE Producto.id = TuplaNueva.id
```

# Triggers

## Sintaxis

```
CREATE TRIGGER SueldoPromedio  
AFTER UPDATE OF sueldo ON Empleados
```

- **AFTER** puede ser **INSERT**, **DELETE**, **UPDATE**
- También existe **BEFORE** o **INSTEAD OF**

# Triggers

Sintaxis

REFERENCING

OLD TABLE AS TuplasAntiguas

NEW TABLE AS TuplasNuevas

- Podemos referenciar **OLD** y **NEW TABLE** o **ROW**

# Triggers

## Sintaxis

### FOR EACH STATEMENT

- **FOR EACH STATEMENT:** ejecuta una vez por instrucción, cuando esta terminó
- **FOR EACH ROW:** ejecuta una vez por tupla modificada, cuando está terminó (pero la instrucción quizás no)

# Triggers

## Sintaxis

```
WHEN (3000000 < (SELECT AVG(suelo)
                FROM Empleados WHERE rango = 'gerente'))
BEGIN
...
END;
```

- **WHEN** es una condición como en SQL
- Entre **BEGIN** y **END** pueden haber varias instrucciones SQL