

Disparadores

Álvaro Herrera

13 de noviembre de 2004

- ▶ Qué es un disparador
- ▶ Componentes
- ▶ La función de disparador
- ▶ Ejemplo PL/pgSQL
- ▶ Ejemplo PL/Tcl

Qué es un disparador?

- ▶ “Trigger”
- ▶ Qué hace: ejecutar una acción
- ▶ Cuándo: un evento ocurre en la base de datos
- ▶ Acciones:
 - ▶ una función de disparador
- ▶ Eventos:
 - ▶ INSERT
 - ▶ UPDATE
 - ▶ DELETE

```
CREATE TRIGGER [ nombre ]  
  
{ BEFORE | AFTER }  
{ evento [ OR ... ] }  
  
[ FOR EACH { ROW | STATEMENT } ]  
  
EXECUTE PROCEDURE funcion(argumentos)
```

```
CREATE FUNCTION nombre ( args )  
RETURNS TRIGGER [ ... ]
```

Elementos del disparador se pasan de forma especial

- ▶ elementos del registro
 - ▶ FOR EACH ROW
 - ▶ NEW / OLD
- ▶ nombre de la tabla, evento, etc
- ▶ argumentos de creación

- ▶ Registros que la función recibe
- ▶ NEW
 - ▶ registro insertado (INSERT)
 - ▶ registro modificado (UPDATE)
- ▶ OLD
 - ▶ registro antes de modificar (UPDATE)
 - ▶ registro eliminado (DELETE)
- ▶ OLD
- ▶ Pueden ser modificados
- ▶ Puede retornar:
 - ▶ NULL (detener la acción)
 - ▶ OLD (evento DELETE)
 - ▶ NEW (eventos INSERT, UPDATE)

Modificaciones sólo con disparadores *BEFORE!*

Ejemplo PL/pgSQL

```
CREATE FUNCTION verifica_empleado() RETURNS trigger
LANGUAGE plpgsql AS '
BEGIN
    IF NEW.sueldo < OLD.sueldo THEN
        RAISE NOTICE ''no bajaremos el sueldo'';
        RETURN NULL;
    END IF;

    IF NEW.sueldo < 0 THEN
        RAISE EXCEPTION ''% debe tener sueldo positivo'';
    END IF;

    NEW.ultima_fecha := 'now';
    NEW.ultimo_usuario := current_user;
    RETURN NEW;
END;
```

Ejemplo PL/pgSQL

```
CREATE FUNCTION actualiza_inventario() RETURNS trig
LANGUAGE plpgsql AS '
    BEGIN

    END
';
```


Ejemplo PL/Tcl

```
CREATE OR REPLACE FUNCTION minusculas()  
RETURNS trigger AS '  
foreach key $args {  
    if {[info exists NEW($key)]} {  
        set NEW($key) [string tolower $NEW($key)]  
    }  
}  
return [array get NEW]  
' LANGUAGE pltcl;
```

```
CREATE TRIGGER minusculizar  
    BEFORE INSERT OR UPDATE ON una_tabla  
    FOR EACH ROW EXECUTE PROCEDURE  
    minusculas('campo_1', 'campo_2');
```

```
CREATE TABLE usuarios
  (usuario_id int,
   nombre text);
```

```
CREATE TABLE usuarios_audit
  (usuario_id int,
   nombre text,
   op text,
   fecha timestamp with time zone);
```

```
CREATE OR REPLACE FUNCTION
  audita_usuarios() RETURNS trigger AS '
  spi_exec "INSERT INTO usuarios_audit
    VALUES ($NEW(usuario_id),
      '[ quote $NEW(nombre) ]',
      '[ quote $TG_op ]',
      now())"
  return [array get NEW]
' LANGUAGE pltcl;
```

```
CREATE TRIGGER audita
  BEFORE UPDATE OR INSERT OR DELETE
  ON usuarios FOR EACH ROW
  EXECUTE PROCEDURE audita_usuarios();
```