

Trigger



Universidad Autónoma
de Entre Ríos

Definición

- Un TRIGGER, es un 'disparador' que se ejecuta cuando queremos hacer una operación en una tabla, por ejemplo (inserción, borrado, modificación).
- Es un bloque de código, inserto en una función que es utilizado para comprobar la integridad de los datos y que la base de datos contenga al final de una transacciones un estado correcto, conforme lo previamente establecido.

Ventajas

Veamos algunas ventajas de utilizar Triggers por parte de los DBMS.

- Mantener la integridad de los datos.
- Garantizar que se ejecute para cada operación que se realiza en una tabla.
- Facilidad de implementar.
- Facilidad para modificar cuando cambian las reglas de negocio.
- Mantener la independencia de los datos.

Veamos un ejemplo

Tenemos una tabla *emplea* con los siguientes datos

```
CREATE TABLE emplea (  
dni int NOT NULL,  
apellido character varying(32),  
nombre CHARACTER VARYING(25),  
fecNac date,  
Constraint emplea_pkey Primary Key (dni)  
);
```

Restricciones

- Que el atributo apellido no sea nulo
- Que el empleado sea mayor de 18 años.

Vamos a generar una función para comprobar estas restricciones

AFTER - BEFORE

- Los triggers disparadores que llamana a una función que se ejecutan al realizar alguna operación sobre las tablas de la B.D.
- Se puede indicar que se ejecute **antes (before)** de producirse alguna opreación sobre la tabla (INSERT, DELETE o UPDATE) ó bien **después (after)** de realizada la operación.
- ¿cuál sería la diferencia?
- ¿Cuándo utilizar AFTER y cuándo BEFORE?

Funciones para TRIGGERS

- A continuación hay un ejemplo con la semántica de las funciones para *triggers*.
- Los triggers nos permiten contar con dos **TUPLAS VIRTUALES**
- **NEW** contiene los valores de los atributos para la tupla que se quiere insertar o modificar
- **OLD** contiene los valores que están actualmente almacenados en la tabla.
- **NEW.atributo** u **OLD.atributo** obtenemos el valor del atributo para esa tupla.

Función

```
CREATE OR REPLACE FUNCTION func_e() RETURNS TRIGGER AS $funcemp$  
DECLARE  
edad smallint ;  
BEGIN  
edad := date_part('year',age(NEW.fecNac));  
IF NEW.apellido IS NULL THEN  
    RAISE EXCEPTION 'no puede tener apellido vacío';  
END IF;  
IF edad <= '18' THEN  
    RAISE EXCEPTION 'no puede ser menor de 18 años';  
END IF;  
    RETURN NEW;  
END; $funcemp$ LANGUAGE plpgsql;
```


Creamos el Trigger

```
CREATE TRIGGER trigger_e BEFORE  
INSERT OR UPDATE ON emplea  
FOR EACH ROW EXECUTE PROCEDURE  
func_e();
```

Probamos insertar tuplas

- `INSERT INTO emplea VALUES('11111111', 'GONZALEZ', 'JUAN', '1998-11-10');`

ERROR: no puede ser menor de 18 años

- `INSERT INTO emplea VALUES('11111111',null, 'JUAN', '1950-11-10');`

ERROR: no puede tener apellido vacío

Probamos insertar tuplas

- `INSERT INTO emplea VALUES('111111111', 'GONZALEZ', 'JUAN', 1978-11-10');`

Query returned successfully: one row affected, 1028 ms execution time.

- `select dni,apellido,nombre, (date_part('year',age(fecNac))) as edad from emplea;`

DNI	APELLIDO	NOMBRE	EDAD
11111111;	"GONZALEZ";	"JUAN";	34