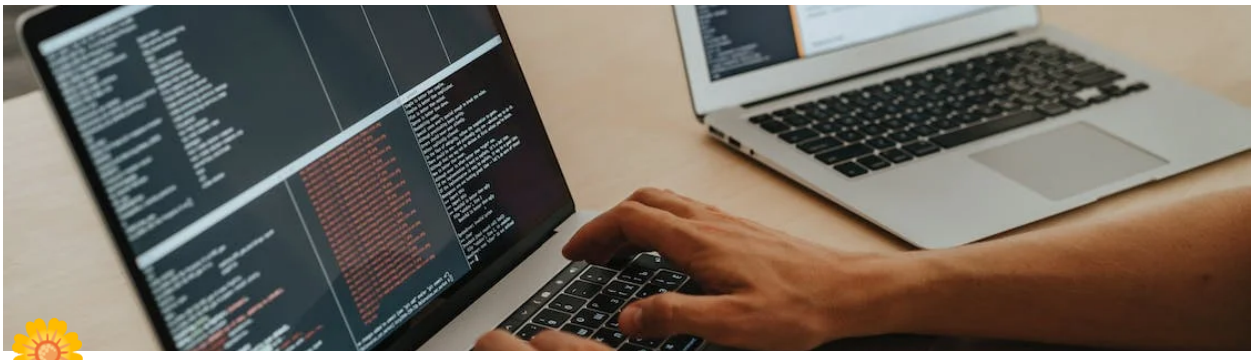


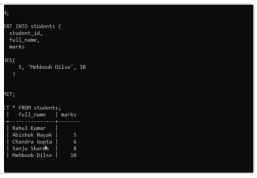
9 Triggers plpgSQL

Links de interés:

PostgreSQL - BEGIN - GeeksforGeeks

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and

https://geeksforgeeks.org/postgresql-begin/



Capítulo 8 Modelo relacional | Base de Datos: Una aproximación a la inf

Manual de referencia en bases de datos, desde un aspecto introductorio y forma p una serie de libros sobre datos e información.

https://bookdown.org/paranedagarcia/database/modelo-relacional.html

MOVE

MOVE MOVE — position a cursor Synopsis MOVE [direction] [FROM | IN] cursor_name where direction can …

https://www.postgresql.org/docs/current/sql-move.html

PostgreSQL CHECK Constraint

In this tutorial, we will introduce you to PostgreSQL CHECK constraint to constrain the value of columns in a table based on a Boolean expression.

https://postgresqltutorial.com/postgresql-tutorial/postgresql-check-constraint/

CREATE TRIGGER

CREATE TRIGGER CREATE TRIGGER — define a new trigger Synopsis CREATE [OR REPLACE] [CONSTRAINT]

https://www.postgresql.org/docs/current/sql-createtrigger.html

Understanding PostgreSQL Triggers: A Comprehensive 101 Guide - Lea

This article provides a comprehensive guide on PostgreSQL Triggers, different ope associated with them and the example queries to implement them.

https://hevodata.com/learn/postgresql-triggers/

Ampliación de conceptos:

-Mirar lo que es una transacción en bases de datos (no lo ha explicado bien).

-Mirar lo que es una variable tipo cursor en bases de datos

-Vamos a dar hasta triggers

-Mirar como blindar la base de datos con triggers

-Si se elimina y se vuelve a crear una función, esta nueva función no es el mismo objeto que la vieja, por tanto, tendrán que eliminarse las reglas, vistas y disparadores existentes que hacían referencia a la antigua función.

FUNCIONES PLPGSQL

CREACIÓN DE TRIGGERS:

Se usan cuando hay un evento asociado a una tabla

Primero hay que crearlo y luego utilizarlo sobre la tabla

Se invocan automáticamente cuando el suceso va a pasar una vez se crean

Existen dos tipos de triggers:

1. BEFORE Triggers: se activan antes de que se ejecute la acción que los activa. Por lo tanto, pueden usarse para realizar acciones de validación o cambios previos a la ejecución de la operación.
2. AFTER Triggers: se activan después de que se ha ejecutado la acción que los activa. Por lo tanto, se pueden usar para realizar acciones posteriores a la ejecución de la operación. Por ejemplo, se puede usar un trigger AFTER INSERT para realizar una operación adicional que dependa del valor de la fila que se ha insertado.

Para un trigger hay dos partes:
CREACIÓN y FUNCIÓN ASOCIADA AL TRIGGER

CREACIÓN:

```
CREATE TRIGGER nombretrigger
{ BEFORE | AFTER | INSTEAD OF } { evento [OR…] }
ON tabla
[FOR EACH { ROW | STATEMENT } ]
[WHEN ( condición ) ]
EXECUTE PROCEDURE función_disparadora()
```

evento = INSERT, UPDATE, DELETE…

FUNCIÓN ASOCIADA AL TRIGGER:

```
CREATE or REPLACE FUNCTION nombreFuncion() returns trigger AS $$
[DECLARE]
BEGIN
  INSERT INTO/UPDATE/DELETE
  RETURN NEW/OLD;
END;
$$LANGUAGE plpgsql;
```

Tened en cuenta que `NEW` es NULL en los triggers `ON DELETE` y `OLD` es NULL en los triggers `ON INSERT`.

LLamada al trigger	`NEW` tiene valores	`OLD` tiene valores
`ON INSERT`	✓	
`ON UPDATE`	✓	✓
`ON DELETE`		✓

Primero creamos la tabla de logging(donde ponemos la info del trigger)

```
CREATE TABLE employees_ AS SELECT * FROM employees WHERE employeenumber is null;
```

Le añadimos un campo que actúe como sello temporal

```
ALTER TABLE employees_ add selloTemporal timestamp;
```

Esta es la declaración del trigger (se hace posteriormente a la declaración de la función que dispara)

```
CREATE TRIGGER trigger_update_empleado
AFTER UPDATE
ON employees
FOR EACH ROW
EXECUTE PROCEDURE updateEmployee();
```

Y por último declaramos la función asociada al trigger

```
CREATE OR REPLACE FUNCTION updateEmployee() returns trigger AS
$$
DECLARE
BEGIN

INSERT INTO employees_
values(OLD.employeenumber,OLD.lastname,OLD.firstname,OLD.extension,OLD.email,OLD.officecode,OLD.reportsto,OLD.jobtitle,now());
RETURN null;

END;
$$LANGUAGE plpgsql;
```

NOTA: Cuidado con la longitud del nombre del trigger porque da problemas si es demasiado largo

TG_OP

TG_OP es una variable especial en PostgreSQL que se utiliza dentro de un trigger para determinar la operación que ha activado el trigger. El valor de la variable TG_OP depende del tipo de trigger (BEFORE o AFTER) y de la operación (INSERT, UPDATE o DELETE) que lo ha activado.

Por ejemplo, si tenemos un trigger AFTER INSERT en una tabla llamada "usuarios", la variable TG_OP tomará el valor 'INSERT' cuando el trigger se active por una operación de inserción en la tabla "usuarios".

La variable TG_OP se puede utilizar dentro del cuerpo del trigger para realizar acciones específicas según la operación que ha activado el trigger. Por ejemplo, se puede usar en una cláusula condicional para ejecutar cierto código solo en caso de una operación de inserción y no en caso de una operación de actualización o eliminación.

TG_TABLE_NAME

TG_TABLE_NAME es otra variable especial en PostgreSQL que se utiliza dentro de un trigger para hacer referencia al nombre de la tabla sobre la que se ha activado el trigger. Esta variable es útil cuando se trabaja con múltiples tablas y se desea realizar diferentes acciones según la tabla que ha desencadenado el trigger.

Por ejemplo, supongamos que tenemos un trigger AFTER INSERT en dos tablas diferentes, "clientes" y "ventas". Si queremos realizar una acción específica solo en la tabla "ventas", podemos utilizar la variable TG_TABLE_NAME para identificar cuál de las tablas ha activado el trigger.

La variable TG_TABLE_NAME se puede utilizar dentro del cuerpo del trigger para hacer referencia al nombre de la tabla y ejecutar cierto código solo en caso de que se active en una tabla específica. Por ejemplo, se puede usar en una cláusula condicional para ejecutar cierto código solo en caso de que el trigger se active en la tabla "ventas" y no en caso de que se active en la tabla "clientes".