

PostgreSQL Módulo 1 - Triggers

Rodrigo Soliz Rocabado (rodrifer@gmail.com)

June 14, 2007

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Triggers

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Hemos visto las ventajas que las funciones nos proporcionan al simplificar acciones en nuestra base de datos.

Pero estas acciones requieren la intervención de una persona encargada de ejecutar las funciones cuando se requiera de su actuación.

Los Triggers al contrario son funciones que se ejecutan de forma automática en respuesta a ciertos eventos que ocurren en la base de datos.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Cuando se ejecuta un trigger?

- Antes o después de una inserción (INSERT)
- Antes o después de una actualización (UPDATE)
- Antes o después de un borrado (DELETE)

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

La definición de un trigger consta de dos partes,

1. La definición de la función asociada al trigger que es la que ejecuta la acción en respuesta al evento (INSERT, UPDATE, DELETE)
2. La definición formal de trigger que indicará:
 - (a) De que tabla se esperarán los eventos
 - (b) A que tipo de evento se responderá

Una función tipo trigger es similar a una función normal, salvo por unos detalles:

```
CREATE OR REPLACE FUNCTION nombrefuncion (param, param)
  RETURNS trigger
  AS $$
  DECLARE
    variable ;
    variable ;
  BEGIN
    sentencia ;
    sentencia ;
    RETURN retorno ;
  END;
  $$
Language 'plpgsql';
```

Primero, el tipo de retorno ya no es un tipo normal de PostgreSQL, sino el tipo especial **trigger**.

Segundo, dependiendo de las acciones del trigger, el parámetro en el comando **RETURN** puede variar, eso lo veremos mas adelante :-)

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Luego de elaborada la función trigger, debemos definir formalmente el trigger de este modo:

```
CREATE OR REPLACE TRIGGER nombretrigger
  [AFTER | BEFORE] [INSERT | UPDATE | DELETE]
  ON nombretabla
  FOR EACH [ROW | STATEMENT]
  EXECUTE PROCEDURE funcion ;
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

CREATE TRIGGER nombretrigger = Creación del trigger

[AFTER | BEFORE] = Antes o después del evento

[INSERT | UPDATE | DELETE] = El tipo de evento al que se responderá

ON nombretabla = De que tabla se esperarán los eventos

FOR EACH [ROW | STATEMENT] = Para cada fila o para cada sentencia

EXECUTE PROCEDURE función = La función que se va a ejecutar (con sus parámetros si es que los necesitara)

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Hagamos un ejemplo para ver las cosas un poco mas claras. Tomemos otra vez nuestra tabla item:

```
CREATE TABLE item (  
  item_id      serial NOT NULL,  
  nombre      varchar(150) NOT NULL,  
  tipo        varchar(100) NOT NULL,  
  cantidad    int4 NOT NULL DEFAULT 0,  
  precio_compra numeric(7,2) NOT NULL,  
  precio_venta numeric(7,2) NOT NULL,  
  CONSTRAINT item_id_pk PRIMARY KEY (item_id)  
);
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Nuestra tabla **item** tiene la capacidad de almacenar la información mas actual e importante para la tienda de computadoras, pero hay otro tipo de información que no es capaz de almacenar (por lo menos por ahora), la información del cambio de datos. Nuestra tabla **item** no tiene memoria de los cambios que han sufrido sus tuplas, si alguien viene y cambia el precio de venta de un artículo, no hay forma de obtener el precio anterior.

Si alguien borra un artículo, tampoco sabremos que artículo ha sido borrado ni por quien. Esto nos lleva a la primera gran área de utilización de los triggers, la auditoría de tablas.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Entonces cuales son las acciones que debe realizar nuestro trigger?

- Necesitamos llevar un registro de todos los cambios relevantes en nuestra tabla **item**, para empezar queremos tener un seguimiento de los cambios en los precios de los items.
- También queremos almacenar en algún lugar todos aquellos items que hayan sido eliminados, también queremos guardar sus respectivos historiales de cambios.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Antes de empezar a definir el trigger es necesario conocer una cuantas variables por defecto que nos provee PostgreSQL:

current_user = el nombre del usuario que esta actualmente conectado a la base de datos y que ejecuta las sentencias.

current_date = La fecha actual (del servidor, no del cliente)

current_time = La hora actual

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

PostgreSQL también maneja unas cuantas variables al momento de ejecutar un trigger, estas son:

NEW = Variable compuesta que almacena los nuevos valores de la tupla que se esta modificando

OLD = Variable compuesta que almacena los valores antiguos de la tupla que se esta modificando

TG_OP = Variable tipo string que indica que tipo de evento está ocurriendo (INSERT, UPDATE, DELETE)

TG_ARGV = Variable tipo arreglo que almacena los parametros de la función del trigger, podemos accederlos de la forma TG_ARGV[0], TG_ARGV[1], etc.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Como último paso antes de empezar a definir el trigger, debemos crear una tabla que vaya a almacenar los cambios de precio en nuestra tabla **item**, la llamaremos **item_actualizado**.

```
CREATE TABLE item_actualizado (  
  item_id          int4 NOT NULL,  
  nombre          varchar(150) NOT NULL,  
  precio_anterior numeric(7,2) NOT NULL,  
  precio_actualizado numeric(7,2) NOT NULL,  
  autor           varchar(100) NOT NULL,  
  fecha_cambio   date NOT NULL,  
  CONSTRAINT item_id_fk FOREIGN KEY (item_id) REFERENCES item(item_id)  
);
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

```
CREATE FUNCTION item_actualizado_tri()
RETURNS trigger
AS $$
BEGIN
    IF (TG_OP = 'UPDATE') THEN
        INSERT INTO item_actualizado VALUES (
            OLD.item_id, OLD.nombre, OLD.precio_venta,
            NEW.precio_venta, current_user, current_date);
    END IF;
    RETURN NULL;
END;
$$
Language 'plpgsql';
```

La nombramos con un sufijo **tri** para darnos cuenta que es una función perteneciente a un trigger y no confundirla con otra que tengamos en nuestro servidor.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Vamos a la definición formal del trigger:

```
CREATE TRIGGER actualizar_item
AFTER UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE item_actualizado_tri();
```

Listo, cada vez que se actualice una tupla en la tabla **item**, se reflejará el cambio en la tabla **item_actualizado**. Pero era eso lo que queríamos?, no totalmente, volvamos a la función del trigger.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

```
CREATE FUNCTION item_actualizado_tri()
RETURNS trigger
AS $$
BEGIN
    IF (TG_OP = 'UPDATE') THEN
        IF (OLD.precio_venta != NEW.precio_venta) THEN
            INSERT INTO item_actualizado VALUES (
                OLD.item_id, OLD.nombre, OLD.precio_venta,
                NEW.precio_venta, current_user, current_date);
        END IF;
    END IF;
    RETURN NULL;
END;
$$
Language 'plpgsql';
```

Ahora si, solo se insertarán tuplas en la tabla **item_actualizado** cuando se cambie el valor de **precio_venta** .

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Modifiquemos algún **precio_venta** de la tabla **item** y luego veamos la tabla **item_actualizado**.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Ejercicios:

- A la tabla **item_actualizado** añadir un atributo **hora_cambio** que almacene la hora de la modificación.
- Hacer otro trigger que haga el mismo tratamiento pero con los items eliminados de la tabla **item**.

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Modificamos la tabla:

```
CREATE TABLE item_actualizado (  
  item_id          int4 NOT NULL,  
  nombre           varchar(150) NOT NULL,  
  precio_anterior  numeric(7,2) NOT NULL,  
  precio_actualizado numeric(7,2) NOT NULL,  
  autor            varchar(100) NOT NULL,  
  fecha_cambio     date NOT NULL,  
  hora_cambio      timetz NOT NULL,  
  CONSTRAINT item_id_fk FOREIGN KEY (item_id) REFERENCES item(item_id)  
);
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Modificamos la función:

```
CREATE FUNCTION item_actualizado_tri()
  RETURNS trigger
  AS $$
  BEGIN
    IF (TG_OP = 'UPDATE') THEN
      IF (OLD.precio_venta != NEW.precio_venta) THEN
        INSERT INTO item_actualizado VALUES (
          OLD.item_id, OLD.nombre, OLD.precio_venta, NEW.precio_venta,
          current_user, current_date, current_time);
      END IF;
    END IF;
    RETURN NULL;
  END;
  $$
Language 'plpgsql';
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Gracias a nuestra nueva tabla **item_actualizado** ahora podemos obtener mas información acerca de nuestra tienda de computadoras.

Podemos saber:

- Que precio tenia cierto ítem en cierta temporada
- Cuanto tiempo se ha mantenido el precio estable
- En que temporadas del año los precio caen o se disparan
- Quienes han modificado los precios de los items

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

Ahora hagamos el trigger que haga el seguimiento de los items que han sido eliminados de nuestra tabla **item**:

Primero creamos la tabla **item_eliminado**:

```
CREATE TABLE item_eliminado (  
    item_id          int4 NOT NULL,  
    nombre           varchar(150) NOT NULL,  
    precio_compra    numeric(7,2) NOT NULL,  
    precio_venta     numeric(7,2) NOT NULL,  
    autor            varchar(100) NOT NULL,  
    fecha_eliminación date NOT NULL,  
    hora_eliminacion timetz NOT NULL  
);
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

```
CREATE FUNCTION item_eliminado_tri()  
RETURNS trigger  
AS $$  
BEGIN  
    IF (TG_OP = 'DELETE') THEN  
        INSERT INTO item_eliminado VALUES (  
            OLD.item_id, OLD.nombre, OLD.precio_compra, OLD.precio_venta,  
            current_user, current_date, current_time);  
    END IF;  
    RETURN NULL;  
END;  
$$  
Language 'plpgsql';
```

Triggers

Triggers

Ejecución de un trigger

Definición de un trigger

Partes de un trigger

Definición formal del trigger

Nuestro primer trigger

Que es lo que queremos?

Variables del sistema

Variables del trigger

Función trigger

Otro trigger

```
CREATE TRIGGER eliminar_item
AFTER DELETE ON item
FOR EACH ROW
EXECUTE PROCEDURE item_eliminado_tri();
```

Pero aquí nos surge un problema, solo podremos eliminar items que no tengan sus correspondientes cambios en la tabla **item_actualizado**, una solución seria eliminar todo su registro de cambios antes de hacer la eliminación.

- Triggers
- Triggers
- Ejecución de un trigger
- Definición de un trigger
- Partes de un trigger
- Definición formal del trigger
- Nuestro primer trigger
- Que es lo que queremos?
- Variables del sistema
- Variables del trigger
- Función trigger
- Otro trigger