**Evidencia día 4 semana 12**

**Procedimientos Almacenados**



**Descripción**

En esta Clase se realizo la implemtacionProcedimientos almacenados.

**Reflexión**

Para profundisar en el tema analize este ejemplo.

Los procedimientos o funciones son esctructuras que aglutinan una serie de instruciones con el fin de evitar tener que repetir código, esto produce una mejora en la legibilidad y eficiencia del código, así como un mejor control de errores.

En el SGBD PostgreSQL un procedimiento no devuelve ningún resultado, encambi una función si devuelve un resultado.

Buenas prácticas:

Documentar el código generado, incluyendo detalles como autor del procedimiento, fecha de creación y versión actual, así como un registro de los cambios que dicho procedimiento ha sufrido a lo largo de su vida.

Asegurarse de que los tipos de datos están correctamente definidos, evita problemas de complilación o transformación de tipos que baja el rendimiento de la base de datos.

Utilizar las llamadas a procedimientos desde las aplicaciones, siempre que sea posible que actuan como caja negra frente a la base de datos y protege a las aplicaciones de posibles cambios de estructuras lógicas de las consultas.

Evitar la generación de código dinámico, evita errores a largo plazo si las estructuras de datos cambian, asi como evitar problemas de seguridad (denominados Inyección SQL)

Evitar llamadas a funciones de manera innecesaria, ejemplo llamar a la función now():date para pedir la hora cada vez que seestime en vez de almacenarla en una variable temporal en la que solo se la llame una vez.

Ejemplo de sintaxis, funcion que inserta nuevos clientes menores de edad en una tabla poder hacerlos promociones.

Crear la tabla donde se reflejaran los nuevos cliente menores de edad

CREATE TABLE ventas.tb\_promocion\_joven

(

promocion\_joven\_id SERIAL PRIMARY KEY,

client\_name char(40) NOT NULL,

age integer,

address char(120),

city char(25),

country char(50) NOT NULL,

contact\_email char(100),

phone char(15),

created\_date date,

create\_by\_user char(50)

);

Crear una funcion que separe a los clientes jovenes en tb\_promocion joven

CREATE OR REPLACE FUNCTION ventas.promocion\_joven()

RETURNS trigger AS

$$

/\*

\* Procedimiento: promocion\_joven

\* Autor: Diego Calvo Barreno

\* Fecha creación: 2017-05-29

\* Versión: 1.0

\* Parámetros: sin parámetros

\* Descripción: Procedimiento que cargará en la tabla tb\_promocion\_joven

\* los clientes que al darse de alta tienen menos de 18 años

\*/

DECLARE

-- Define la estructura temporal que servirá para realizar las operaciones

temp\_row ventas.tb\_promocion\_joven%rowtype;

BEGIN

IF (NEW.age < 18) THEN

-- Capturar datos del nuevo cliente creado

temp\_row.promocion\_joven\_id := NEW.client\_id;

temp\_row.client\_name := NEW.client\_name;

temp\_row.age := NEW.age;

temp\_row.address := NEW.address;

temp\_row.city := NEW.city;

temp\_row.country := NEW.country;

temp\_row.contact\_email := NEW.contact\_email;

temp\_row.phone := NEW.phone;

-- Capturar el usuario y el momento de la operación

temp\_row.create\_by\_user := user;

temp\_row.created\_date := now();

BEGIN

-- Inserta los valores en la tabla CDC

INSERT INTO ventas.tb\_promocion\_joven SELECT temp\_row.\*;

END;

-- Devuelve el la estructura grabada NEW

RETURN NEW;

END IF;

RETURN NULL;

END; -- Fin del procedimiento

$$

LANGUAGE plpgsql;

Crear el disparador que activa la funcion

-- Definición del disparador que llama a la función anteriormente descrita

CREATE TRIGGER tg\_promocion\_joven

-- Este disparador se activa con los eventos de inserción, actualización y borrado

AFTER INSERT ON ventas.tb\_client

FOR EACH ROW

EXECUTE PROCEDURE ventas.promocion\_joven()

;

-- Inserciones que prueban la funcion

INSERT INTO ventas.tb\_client ( client\_id, client\_name, age, address, city, country, contact\_email, phone, created\_date ) VALUES

('10000', 'Susana', '15','C/ Real 1', 'Soria', 'España', NULL, NULL, NULL),

('10001', 'Rosana', '16','C/ Real 1', 'Soria', 'España', NULL, NULL, NULL),

('10002', 'Javier', '35','C/ Real 2', 'Soria', 'España', NULL, NULL, NULL);