

# Procedimientos almacenados

Procedimientos: Ejecutan una serie de comandos.

Funciones: Retornan un valor después de procesar alguna información.

Trigger: Ejecutan acciones automáticas cuando ocurre un evento, como insertar o actualizar datos.

## Variables

- Son herramientas que se pueden usar dentro de estos tres tipos de bloques para guardar y manipular información mientras se ejecuta el código.
- Es un espacio de almacenamiento temporal que se utiliza para guardar información que puede cambiar durante la ejecución del código.

```
DECLARE
    var1 TYPE;
    var2 TYPE;
    var3 TYPE;
cursor_nombre CURSOR FOR SELECT columna FROM tabla;

-- Almacenar una variable con una consulta:
SELECT columna1 INTO var1 FROM tabla WHERE condicion;

-- Almacenar múltiples variables con una consulta
SELECT columna2, columna3 INTO var2, var3 FROM tabla WHERE condicion;
```

## ESTRUCTURA

Procedimientos

```
CREATE OR REPLACE PROCEDURE nombre(parametros)
LANGUAGE plpgsql AS $$
DECLARE -- Opcional
    -- Cursor
    nombre CURSOR FOR SELECT variables FROM tabla;

    -- Variables
    var1 TYPE;
BEGIN
    -- CODE
END $$;
```

## Funciones

```
CREATE OR REPLACE FUNCTION nombre(parametros)
RETURN tipo_dato AS $$
DECLARE
    -- Variables
BEGIN
    -- CODIGO
    -- IFS
    -- Return
END;
$$ LANGUAGE plpgsql;
```

## Trigger

```
CREATE OR REPLACE FUNCTION nombre(parametros)
RETURN trigger AS $$
BEGIN
    -- Code
    RETURN data;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER nombre
{ BEFORE | AFTER } { INSERT | UPDATE | DELETE }
ON nombreTabla
FOR EACH ROW
EXECUTE FUNCTION nombre_funcion();
```

# TIPOS DE CONTROL DE FLUJO

## IF

```
IF condicion THEN
    -- Codigo que se ejecuta si la condicion es verdadera
ELSIF otra_condicion THEN
    -- Codigo que se ejecuta si es la otra condicion
ELSE
    -- Codigo que se ejecuta si ninguna de las dos es
END IF;
```

## CASE

- Variable especifica

```

CASE variable
    WHEN valor1 THEN
        -- Codigo
    WHEN valor2 THEN
        -- Codigo
    ELSE
        -- Codigo salida
END CASE;

```

- Condición general

```

CASE
    WHEN condicion THEN
        -- Codigo
    WHEN condicion2 THEN
        -- Codigo
    ELSE
        -- Codigo
END CASE;

```

## LOOP (bucle)

```

LOOP
    -- Codigo dentro del bucle
    EXIT WHEN condicion; -- Salir del bucle si se cumple la condicion
END LOOP;

```

# Ejemplos e Información

Procedimiento :

## Ejemplo Procedimientos

```

CREATE OR REPLACE PROCEDURE ProcesarEmpleados()
LANGUAGE plpgsql AS $$
DECLARE

    -- Declara el cursor que selecciona id y nombre de tabla empleados
    empleado_cursor CURSOR FOR SELECT id, nombre FROM empleados;

    -- Declara variables para almacenar los valores de cada fila
    emp_id INT;
    emp_nombre VARCHAR(50);

BEGIN
    -- Abre el cursor para iniciar el recorrido
    OPEN empleado_cursor;

```

```
-- Bucle para recorrer todas las filas del cursor
LOOP
    -- Cada vez que se ejecuta un FETCH, el cursor avanza una
    fila en el conjunto de resultados.

    -- Extrae la siguiente fila del cursor y la guarda en
variables
    FETCH empleado_cursor INTO emp_id, emp_nombre;

    -- Verifica si el FETCH no extrajo ninguna fila (Fin)
    EXIT WHEN NOT FOUND;

    -- Operaciones con los datos extradios (Mostrar el nombre
del empleado)
    RAISE NOTICE 'Empleado Procesado: ID = %, Nombre = %',
emp_id, emp_nombre;

    END LOOP;

    -- Cierra el cursor después de completar el bucle
    CLOSE empleado_cursor;
END $$;
```