**Sintaxis lenguaje procedimental PL/pgSQL**

* **Anotaciones**
  + Todo lo que se encuentre entre corchetes (**[]**) será opcional
  + Las palabras en **azul** deberán cambiarse por los elementos del lenguaje mencionados en su nombre
  + Las palabras en color **rosa** representan palabras reservadas del lenguaje, por lo que siempre se escribirán de la misma forma. Aunque es posible escribirlas en mayúsculas o minúsculas
  + Los elementos en color **verde** representan caracteres que conservan su valor

Actividad 9

Lenguaje procedimental PL/pgSQL

* **Declaración de un bloque**

**[ <<etiqueta>> ]**

**[ declare**

**declaraciones ]**

**begin**

**sentencias**

**...**

**end [ etiqueta ];**

* **Ejecución de un bloque anónimo**

**do $$**

**[ <<etiqueta>> ]**

**[ declare**

**declaraciones ]**

**begin**

**sentencias**

**...**

**end [ etiqueta ];**

**$$**

* **RAISE**

**RAISE NOTICE ‘formato’ [ , *expresión* [, ... ] ];**

* **Declaración de variables**

**nombre\_de\_la\_variable tipo\_de\_dato [ := expresión ];**

* **Declaración de constantes**

**nombre\_de\_la\_constante constant tipo\_de\_dato := expresión;**

* **SELECT INTO**

**SELECT**

**lista\_de\_columnas**

**INTO**

**[ TEMPORARY | TEMP | UNLOGGED ] [ TABLE ] nombre\_de\_las\_variables**

**FROM**

**tabla**

**WHERE**

**condición;**

* **Copiar el tipo de dato de una columna**

**nombre\_de\_la\_variable nombre\_de\_la\_tabla.nombre\_de\_la\_columna;**

* **%type**

**nombre\_de\_la\_variable variable%type;**

* **Declaración de una variable de tipo fila**

**nombre\_de\_la\_variable nombre\_de\_la\_tabla%rowtype;**

* **Acceso a una columna de tipo fila**

**nombre\_de\_la\_variable.nombre\_de\_la\_columna**

* **Declaración de una variable de tipo registro**

**nombre\_de\_la\_variable record;**

* **Acceso a una columna de tipo registro**

**nombre\_de\_la\_variable.nombre\_de\_la\_columna**

**Actividad 10**

**Funciones definidas por el usuario**

* **Creación de una función**

**CREATE [ OR REPLACE ] FUNCTION nombre\_función(lista\_parámetros)**

**[ RETURNS valor\_retorno | RETURNS TABLE ( nombre\_columna tipo\_columna [, ...] ) ]**

**LANGUAGE PLPGSQL**

**AS**

**$$**

**DECLARE**

**-- declaración de variables**

**BEGIN**

**-- lógica**

**END;**

**$$**

* **Eliminación de una función**

**DROP FUNCTION [ IF EXISTS ] nombre [ ( [ [ modo\_argumento ] [ nombre\_argumento ] tipo\_dato [, ...] ] ) ] [, ...];**

* **Mostrar las funciones definidas por el usuario**

**\df**

* **Declaración de parámetros**

**nombre [ CONSTANT ] tipo [ NOT NULL ] [ { DEFAULT | := | = } expresión ];**

**Actividad 11**

**Estructuras de control: IF-THEN, CASE, LOOP-WHILE-REPEAT**

* **IF**

**IF condición THEN**

**sentencias;**

**END IF;**

* **CASE**

**CASE expresión\_búsqueda**

**WHEN expresión\_1 [, expresión\_2, ...] THEN**

**-- sentencias-WHEN**

**[ ... ]**

**[ELSE**

**-- sentencias-ELSE ]**

**END CASE;**

* **LOOP**

**[ <<etiqueta>> ]**

**LOOP**

**-- sentencias**

**END LOOP;**

* **WHILE**

**[ <<etiqueta>> ]**

**WHILE condición LOOP**

**-- sentencias**

**END LOOP;**

* **FOR-LOOP (rango de enteros)**

**[ <<etiqueta>> ]**

**FOR contador\_ciclo IN [ REVERSE ] de..hasta [ BY paso ] LOOP**

**-- sentencias**

**END LOOP [ etiqueta ];**

* **FOR-LOOP (conjunto de resultados)**

**[ <<etiqueta>> ]**

**FOR variable\_temporal IN consulta LOOP**

**-- sentencias**

**END LOOP [ etiqueta ];**

**Actividad 12**

**Manejo de errores**

* **RAISE**

**RAISE [ nivel ] ‘formato’ [ , expresión [, ... ] ];**

* **ASSERT**

**ASSERT condición [, mensaje ];**

* **EXCEPTION**

**<<etiqueta>>**

**DECLARE**

**BEGIN**

**-- sentencias**

**EXCEPTION**

**WHEN condición [OR condición...] THEN**

**-- manejo de la excepción**

**[WHEN condición [OR condición...] THEN**

**-- manejo de la excepción**

**]**

**[WHEN OTHERS THEN**

**-- manejo de otras excepciones**

**]**

**END;**

**Actividad 13**

**Procedimientos almacenados**

* **Creación de un procedimiento almacenado**

**CREATE [OR REPLACE] PROCEDURE nombre\_procedimiento(lista\_parametros)**

**LANGUAGE PLPGSQL**

**AS $$**

**DECLARE**

**-- declaración de variables**

**BEGIN**

**-- cuerpo del procedimiento**

**-- transacciones**

**END;**

**$$**

* **Eliminación de un procedimiento almacenado**

**DROP PROCEDURE [IF EXISTS] nombre\_procedimiento (lista\_argumentos)[CASCADE | RESTRICT];**

* **Eliminación de varios procedimientos almacenados**

**DROP PROCEDURE [IF EXISTS] nombre\_procedimiento1, nombre\_procedimiento2, ...;**

**Actividad 14**

**Cursores**

* **Declaración de cursores**

**DECLARE nombre\_cursor REFCURSOR;**

* **Declaración de cursores enlazados a consultas**

**nombre\_cursor [ [ NO ] SCROLL ] CURSOR [( nombre tipo\_dato, nombre tipo\_dato, ... )] FOR consulta;**

**Nota:** ConSCROLL se especifica si el cursor se puede desplazar hacia atrás, si se usa NO SCROLL, el cursor no se podrá desplazar hacia atrás

* **Apertura de cursores no enlazados**

**OPEN cursor\_no\_enlazado [ [ NO ] SCROLL ] FOR consulta;**

* **Apertura y vinculación de cursores no enlazados, de manera dinámica**

**OPEN cursor\_no\_enlazado [ [ NO ] SCROLL ] FOR EXECUTE cadena\_de\_consulta [USING EXPRESSION [, ... ] ];**

* **Apertura de cursores enlazados**

**OPEN nombre\_cursor [ (nombre := valor, nombre := valor,...)];**

* **Recuperación de filas**

**FETCH [ dirección { FROM | IN } ] nombre\_cursor INTO variable\_objetivo;**

* **Avanzando filas**

**MOVE [ dirección { FROM | IN } ] nombre\_cursor;**

* **Actualización de filas**

**UPDATE nombre\_tabla SET columna = valor, ... WHERE CURRENT OF nombre\_cursor;**

* **Eliminación de filas**

**DELETE** **FROM** **nombre\_tabla WHERE CURRENT OF nombre\_cursor;**

* **Cierre de cursores**

**CLOSE nombre\_cursor;**

**Actividad 15**

**Disparadores SQL (trigger)**

* **Creación de una función trigger**

**CREATE FUNCTION función\_trigger()**

**RETURNS TRIGGER**

**LANGUAGE PLPGSQL**

**AS $$**

**BEGIN**

***-- lógica del trigger***

**END;**

**$$**

* **Creación de un trigger**

**CREATE TRIGGER nombre\_trigger**

**{BEFORE | AFTER} { evento }**

**ON nombre\_tabla**

**[FOR [EACH] { ROW | STATEMENT }]**

**EXECUTE FUNCTION función\_trigger**

* **Creación de un trigger**

**DROP TRIGGER [IF EXISTS] nombre\_trigger ON nombre\_tabla [ CASCADE | RESTRICT ];**

* **Renombrar un trigger**

**ALTER TRIGGER nombre\_trigger ON nombre\_tabla RENAME TO nuevo\_nombre;**

* **Habilitación de un trigger**

**ALTER TABLE nombre\_tabla ENABLE TRIGGER nombre\_trigger | ALL;**

* **Deshabilitación de un trigger**

**ALTER TABLE nombre\_tabla DISABLE TRIGGER nombre\_trigger | ALL;**