**FUNDAMENTOS PL-SQL**

PL-SQL es el lenguaje que tiene Oracle para trabajar con lógica sobre los datos.

Es un lenguaje de programación.

Crear una estructura de código para que realice acciones dependiendo de variables.

Oracle utiliza Estructura de bloques. Son bloques de código que permiten poder realizar acciones secuenciales dentro de objetos.

En una estructura de bloque podemos utilizar multitud de objetos.

* Variables y tipos
* Estructuras de control
* Procedimientos almacenados

En Oracle a estos bloques de código se le llaman bloques anónimos.

Dichos bloques anónimos se pierden cuando terminemos la sesión, pero podemos almacenar bloques de código en objetos para llamarlos posteriormente.

* Procedimientos almacenados: Son objetos persistentes que realizan una secuencia de acciones
* Paquetes: Un conjunto de procedimientos y tipos
* Triggers: Disparadores que se ejecutan al realizar cambios sobre objetos
* Funciones: Bloques de código que devuelven un valor.

Todos los bloques de código están diferenciados por una estructura:

DECLARE

Declaración de variables del programa (opcional)

BEGIN

Lógica del programa

EXCEPTION

Zona si el programa da error (opcional)

END;

Una variable es el nombre que damos a un elemento del programa dónde almacenaremos un dato.

Para la declaración de la variable no podemos utilizar espacios, no podemos empezar por número o no podemos utilizar palabras reservadas.

En Pl-sql no hay diferencia entre mayúsculas y minúsculas.

Siempre se declara la variable con su tipo de dato. Es opcional asignarle un valor.

Si no le asignamos un valor, la variable contiene un null.

Para asignar valores a las variables se utiliza la sintaxis **:=**

declare

numero int;

begin

--la variable es null;

numero := 19;

end;

Imprimir por pantalla:

Dbms\_output.put\_line(‘IMPRIMIENDO POR PANTALLA’);

Si deseamos sacar más valores necesitamos concatenarlos para representarlos junto a texto. ||

Ejemplo de primer PL-SQL

*--EJEMPLO PRIMER PL-SQL*

declare

numero int := 12;

begin

dbms\_output.put\_line('El número vale ' || numero);

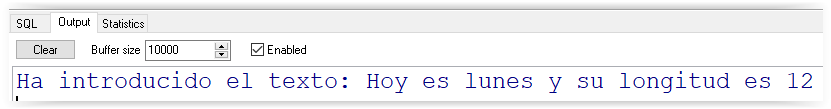
*--CAMBIAMOS EL VALOR DEL NUMERO*

numero := 19;

dbms\_output.put\_line('El número vale ahora ' || numero);

end;

Ejemplo PL-SQL con variables que introduce el usuario



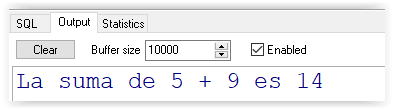
Ejemplo PL-SQL para sumar dos números con un programa

--REALIZAR UN PROGRAMA DONDE PEDIREMOS DOS NUMEROS AL USUARIO.

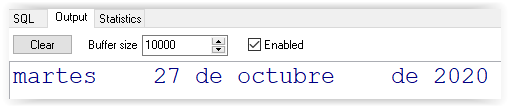
--MOSTRAREMOS POR PANTALLA LA SUMA.

--qué información tengo que guardar?

--numero1 numero2 suma



Ejemplo para pedir una fecha al usuario y mostrar el formato largo de fecha



**CODIGO SQL**

**COMBINAR SENTENCIAS PL-SQL CON SENTENCIAS SQL**

am

Dentro de cualquier bloque pl-sql podemos incluir consultas de acción, pero no podemos incluir consultas de selección.

No podemos hacer un SELECT para mostrar datos.

Podemos utilizar las variables para las sentencias de acción.

Ejemplo:

Debemos crear un bloque pl-sql dónde incrementaremos el salario de los empleados de un departamento que nos indiquen con una variable.

declare

v\_deptno int;

begin

*--pedimos un número de departamento al usuario*

v\_deptno := &deptno;

update emp set salario = salario + 1 where dept\_no = v\_deptno;

end;

Crearemos un bloque pl-sql dónde pediremos al usuario una acción a realizar sobre la tabla departamentos. La acción será insertar o modificar. Si pone otro valor, mostraremos un mensaje de error.

Pediremos también los datos del departamento a insertar o modificar.

**TIPADO DE VARIABLES COLUMNA**

Existe una definición de variables para cuando vamos a utilizarlas con columnas de la tabla.

Dicha definición nos permite indicar el TIPO DE DATO de una columna existente para una variable declarada.

De esa forma, estamos controlando posibles errores, ya que la variable se ajusta a la definición.

Se utiliza la palabra TYPE.

declare

v\_variable TABLA.COLUMNA%TYPE

begin

end;

Ejemplo con TYPE

