**Que es PL/SQL**

PL/SQL, que significa "Procedural Language/SQL", es un lenguaje de programación extendido de SQL, diseñado por Oracle para integrarse con su base de datos. Permite a los desarrolladores combinar sentencias SQL con construcciones de programación procedimental, como bucles, condiciones y estructuras de datos, para realizar operaciones complejas en la base de datos.

En esencia, PL/SQL es un lenguaje de programación que se ejecuta dentro de la base de datos Oracle. Esto significa que, a diferencia de SQL puro, PL/SQL permite:

**Escribir procedimientos y funciones almacenados:**

Estas son unidades de código que se compilan y almacenan en la base de datos, permitiendo la reutilización y la ejecución eficiente de operaciones complejas.

**Controlar el flujo de la ejecución:**

PL/SQL ofrece estructuras de control como IF, THEN, ELSE, FOR, WHILE, etc., para realizar diferentes acciones basadas en condiciones.

**Manejar errores:**

PL/SQL proporciona mecanismos para manejar excepciones (errores) de manera controlada, evitando que la base de datos se detenga abruptamente.

**Utilizar variables y tipos de datos:**

Permite declarar variables para almacenar datos temporales y utilizar diferentes tipos de datos, como números, cadenas, fechas, etc.

**Interactuar con la base de datos de manera más eficiente:**

Al integrar SQL con lógica de programación, PL/SQL permite realizar operaciones complejas en la base de datos de forma más eficiente que con SQL puro.

En resumen, PL/SQL es una herramienta esencial para los desarrolladores que trabajan con bases de datos Oracle, permitiéndoles crear aplicaciones más complejas y eficientes que aprovechan al máximo las capacidades de la base de datos.

**¿Qué es un cursor?**

En PL/SQL, un cursor es una estructura de control con nombre que permite a un programa acceder a las filas de un conjunto de resultados, una a la vez, o en grupos, en lugar de procesar todo el conjunto de resultados de una sola vez. Es una herramienta esencial para trabajar con consultas que devuelven múltiples filas, permitiendo la manipulación y el procesamiento de los datos fila por fila.

**Acceso selectivo:**

Los cursores permiten a los desarrolladores seleccionar y procesar datos específicos dentro de un conjunto de resultados.

**Procesamiento fila por fila:**

En lugar de manejar todos los resultados a la vez, los cursores permiten iterar a través de cada fila del conjunto de resultados, realizando operaciones específicas en cada una.

**Tipos de cursores:**

PL/SQL ofrece cursores implícitos (creados automáticamente para ciertas sentencias) y explícitos (declarados por el desarrollador para un control más preciso).

**Control sobre la manipulación:**

Los cursores explícitos brindan un mayor control sobre la apertura, recuperación y cierre del cursor, permitiendo acciones personalizadas y manejo de errores.

**Atributos del cursor:**

Los cursores tienen atributos como %FOUND, %NOTFOUND, %ROWCOUNT y %ISOPEN que proporcionan información sobre su estado y resultados.

**Utilidad:**

Los cursores son especialmente útiles para:

* Realizar operaciones complejas en cada fila de un conjunto de resultados.
* Manipular datos en diferentes tablas basadas en los resultados de una consulta.
* Implementar lógica de negocio que requiere procesamiento individual de datos.

**Que es un for loop**

En PL/SQL, un bucle FOR se utiliza para ejecutar un bloque de código un número específico de veces. Este tipo de bucle es útil cuando se conoce de antemano la cantidad de iteraciones que se necesitan. Se define un rango numérico, y el bucle se ejecuta para cada valor dentro de ese rango.

**Características importantes:**

* El contador se incrementa automáticamente en cada iteración.
* No es necesario declarar la variable contador.
* El bucle se detiene cuando el contador alcanza el valor fin.
* Se pueden usar las sentencias EXIT y EXIT WHEN para salir del bucle antes de que se complete el rango, según tutorialspoint.com.
* Se puede usar la sentencia CONTINUE para omitir la iteración actual y pasar a la siguiente, según Oracle.
* Se puede utilizar la palabra clave REVERSE para recorrer el rango en orden descendente, según IBM.

**EJEMPLO**

BEGIN

FOR i IN 1..5 LOOP

DBMS\_OUTPUT.PUT\_LINE('Iteración número: ' || i);

END LOOP;

END;

/

**Que es un While**

En PL/SQL, un bucle WHILE es una estructura de control que permite repetir un bloque de código mientras una condición especificada sea verdadera. Es un bucle controlado por la entrada, lo que significa que la condición se evalúa antes de cada iteración del bucle. Si la condición es verdadera, el bloque de código dentro del bucle se ejecuta. Si la condición es falsa, el control del programa pasa a la siguiente instrucción fuera del bucle.

**Características principales:**

**Controlado por la entrada:**

La condición se evalúa antes de cada iteración.

**Repetición condicional:**

El bloque de código se ejecuta repetidamente mientras la condición sea verdadera.

**Posible no ejecución:**

Si la condición es falsa desde el principio, el bucle no se ejecuta ninguna vez.

**Necesidad de control interno:**

Para evitar bucles infinitos, es crucial que dentro del bucle haya alguna instrucción que eventualmente cambie el valor de la condición a falso.

**EJEMPLO**

DECLARE

contador NUMBER := 1;

BEGIN

WHILE contador <= 5 LOOP

DBMS\_OUTPUT.PUT\_LINE('Contador: ' || contador);

contador := contador + 1;

END LOOP;

END;

/

**Que es un If**

En PL/SQL, un IF es una estructura de control condicional que permite ejecutar diferentes bloques de código según una condición booleana. Básicamente, evalúa una condición y, si es verdadera, ejecuta un conjunto de sentencias; opcionalmente, también puede especificar un bloque de sentencias a ejecutar si la condición es falsa, o incluso múltiples condiciones alternativas.

**Tipos de estructuras IF en PL/SQL:**

IF ... THEN ... END IF;: Ejecuta un bloque de código si la condición es verdadera.

IF ... THEN ... ELSE ... END IF;: Ejecuta un bloque de código si la condición es verdadera, y otro bloque si es falsa.

IF ... THEN ... ELSIF ... THEN ... ELSE ... END IF;: Ejecuta un bloque de código de una serie de condiciones mutuamente excluyentes.

**Ejemplo simple:**

IF edad > 18 THEN

DBMS\_OUTPUT.PUT\_LINE('Es mayor de edad');

END IF;

**Ejemplo con ELSE:**

IF salario > 50000 THEN

DBMS\_OUTPUT.PUT\_LINE('Empleado bien pagado');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Empleado con salario regular');

END IF;

**Ejemplo con ELSIF:**

IF puntuacion >= 90 THEN

DBMS\_OUTPUT.PUT\_LINE('Aprobado con Sobresaliente');

ELSIF puntuacion >= 70 THEN

DBMS\_OUTPUT.PUT\_LINE('Aprobado');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Reprobado');

END IF;

**Que es un case**

En PL/SQL, CASE es una instrucción de control que permite ejecutar diferentes conjuntos de sentencias (o expresiones) dependiendo del valor de una variable o expresión. Es similar a una estructura if-then-else pero con la capacidad de evaluar múltiples condiciones de manera más clara y concisa, especialmente cuando se trata de múltiples opciones.

**¿Cómo funciona?**

La instrucción CASE en PL/SQL puede ser de dos tipos:

**1. CASE simple:**

Evalúa una expresión y compara su valor con diferentes valores de coincidencia. Si encuentra una coincidencia, ejecuta las sentencias asociadas a esa coincidencia.

**2. CASE buscada:**

Evalúa múltiples expresiones booleanas y ejecuta las sentencias asociadas a la primera expresión que sea verdadera.

**Sintaxis general:**

CASE selector

WHEN valor1 THEN

-- Sentencias a ejecutar si selector = valor1

WHEN valor2 THEN

-- Sentencias a ejecutar si selector = valor2

ELSE

-- Sentencias a ejecutar si no hay coincidencia

END CASE;

**Ejemplo con CASE simple:**

DECLARE

v\_dia\_semana VARCHAR2(20);

v\_numero\_dia NUMBER := 3;

BEGIN

CASE v\_numero\_dia

WHEN 1 THEN

v\_dia\_semana := 'Lunes';

WHEN 2 THEN

v\_dia\_semana := 'Martes';

WHEN 3 THEN

v\_dia\_semana := 'Miércoles';

ELSE

v\_dia\_semana := 'Día inválido';

END CASE;

DBMS\_OUTPUT.PUT\_LINE('El día de la semana es: ' || v\_dia\_semana);

END;

/