

Actividad 1

Investigar que es plsql

PL/SQL (Procedural Language /SQL) es un lenguaje de programación de bases de datos desarrollado por Oracle que se utiliza principalmente con la base de datos Oracle. Es una extensión del lenguaje SQL que añade características procedimentales como variables, bucles y estructuras de control.

En resumen, PL/SQL permite a los desarrolladores crear código más complejo y reutilizable para gestionar datos en la base de datos Oracle, complementando la funcionalidad de SQL con elementos procedimentales.



Características principales de PL/SQL:

Expansión de SQL: PL/SQL extiende la sintaxis de SQL permitiendo la creación de procedimientos almacenados, funciones, paquetes y otros elementos.

Procedimientos almacenados: PL/SQL permite definir procedimientos almacenados, que son bloques de código que se pueden reutilizar y ejecutar cuando se llaman.

Funciones: PL/SQL también permite definir funciones, que son similares a los procedimientos almacenados, pero devuelven un valor.

Paquetes: Los paquetes en PL/SQL agrupan procedimientos, funciones y variables relacionadas para facilitar la gestión del código.

Variables y tipos de datos: PL/SQL proporciona variables y tipos de datos para almacenar datos y realizar operaciones en ellos.

Estructuras de control: PL/SQL incluye estructuras de control de flujo como bucles (loops) e instrucciones condicionales (if/else) para controlar la ejecución del código.

Cursor: PL/SQL utiliza cursores para trabajar con los resultados de consultas SQL, permitiendo el procesamiento de los datos fila por fila.

Manipulación de datos: PL/SQL permite realizar operaciones de manipulación de datos, como inserciones, actualizaciones y eliminaciones, utilizando sentencias SQL dentro de bloques de código.

En PL/SQL, un **cursor** es una estructura de control que permite procesar los resultados de una consulta (sentencia SELECT) fila por fila. En esencia, un cursor es como un puntero a un conjunto de resultados, que permite recorrerlo y acceder a cada registro de forma individual.

Acceso a datos: Los cursores son esenciales para manejar conjuntos de resultados extensos en PL/SQL, permitiendo recorrerlos y procesarlos uno a uno o en bloques.

Control y flexibilidad: A diferencia de las consultas SELECT que devuelven todos los resultados de una vez, los cursores permiten un control más preciso sobre el acceso y la manipulación de los datos, como procesar filas condicionalmente.

Tipos de cursores: Existen cursores implícitos, que son creados automáticamente por PL/SQL para cada instrucción SELECT, y cursores explícitos, que el programador declara y controla directamente.

En PL/SQL, una **excepción** es una condición de error o un evento inesperado que ocurre durante la ejecución de un programa. Estas excepciones pueden ser errores predefinidos por el sistema o errores definidos por el usuario, y sirven para interrumpir el flujo normal del programa y permitir la ejecución de código de manejo de errores.

Condición de error: Una excepción es una situación anormal que se produce cuando se ejecuta un bloque PL/SQL.

Tipos de excepciones en PL/SQL:

Excepciones predefinidas: Son errores estándar definidos por PL/SQL, como ZERO_DIVIDE (división por cero) o DUP_VAL_ON_INDEX (valor duplicado en un índice único).

Excepciones definidas por el usuario: Son excepciones que el programador crea y define explícitamente para casos específicos de su aplicación.

Excepciones no predefinidas: Son errores que no están definidos por PL/SQL, pero que pueden ocurrir en la ejecución, como errores de acceso a datos.

Un bucle **FOR** en PL/SQL es una estructura de control que permite ejecutar un bloque de código repetidamente un número específico de veces. Se utiliza para iterar sobre un rango de valores, realizando una acción en cada iteración.

Se utiliza para realizar tareas repetitivas. Sirve para ejecutar el conjunto de sentencias un número específico de veces. Para ejecutar el bucle FOR, se proporcionan los valores inicial y final. Durante cada iteración, el Contador se incrementa en 1.

En PL/SQL, un bucle **WHILE** es una estructura de control que permite ejecutar repetidamente un bloque de código mientras una condición especificada sea verdadera. La condición se evalúa antes de cada iteración, y si es verdadera, el código dentro del bucle se ejecuta. Si la condición es falsa, el bucle termina y la ejecución continúa con la siguiente instrucción después del bucle WHILE.

Repetición: Ejecuta un bloque de código mientras una condición sea verdadera.

Control de flujo: Evalúa la condición antes de cada iteración.

Terminación: Termina cuando la condición se vuelve falsa.

El bucle WHILE es útil cuando no se sabe cuántas veces se necesita repetir un bloque de código, y la repetición depende de una condición que puede cambiar durante la ejecución del programa.

En PL/SQL, la sentencia **CASE** es una estructura de control que permite ejecutar diferentes bloques de código basados en la evaluación de una o más condiciones. Es similar a la sentencia IF-THEN-ELSE, pero proporciona una forma más estructurada y legible de manejar múltiples condiciones.

La sentencia CASE en PL/SQL permite evaluar un selector (una expresión) y compararlo con diferentes valores (WHEN). Si el selector coincide con un valor, se ejecuta el bloque de código asociado a ese WHEN. Si no hay coincidencias, se puede ejecutar un bloque de código ELSE (opcional).

Que es una **tabla dual**

En el contexto de bases de datos, una tabla DUAL es una tabla especial que existe por defecto en Oracle y otras bases de datos, aunque no es estrictamente necesaria. Tiene una sola columna y una sola fila, y se utiliza principalmente para realizar cálculos o devolver valores de funciones del sistema (como la fecha actual) cuando no se necesita acceder a datos de otras tablas.

Propósito: La tabla DUAL se usa cuando se necesita realizar una consulta SELECT que no requiere datos de una tabla real. Por ejemplo, para obtener la fecha actual, o para realizar cálculos que no dependen de datos de una tabla.

Estructura: Tiene una sola columna, típicamente llamada DUMMY, y solo una fila con un valor, como 'X'.

Uso común: Un ejemplo común es SELECT SYSDATE FROM DUAL;, que devolverá la fecha y hora actual del sistema.

Alternativas:

En otras bases de datos como SQL Server, PostgreSQL, MySQL y MariaDB, no es necesario usar una tabla DUAL porque pueden devolver valores de funciones del sistema directamente sin una cláusula FROM, o usar cualquier tabla con una fila si se requiere.