BASES DE DATOS UD-6

PROGRAMACIÓN DE BASES DE DATOS

ÍNDICE

[INTRODUCCIÓN PL/SQL (PROCEDURAL LANGUAGE / STRUCTURED QUERY LANGUAGE) 3](#_Toc134525926)

[a) Principales ventajas 3](#_Toc134525927)

[b) Arquitectura 3](#_Toc134525928)

[c) Interacción con el usuario 4](#_Toc134525929)

[d) Paquete 4](#_Toc134525930)

[CONCEPTOS BÁSICOS 4](#_Toc134525931)

[A. BLOQUE 4](#_Toc134525932)

[B. COMENTARIOS 5](#_Toc134525933)

[C. SOPORTE PARA COMANDOS SQL 5](#_Toc134525934)

[D. OPERADORES 6](#_Toc134525935)

[E. VARIABLES 7](#_Toc134525936)

[1) Tipos de datos 7](#_Toc134525937)

[2) Variables de sustitución 7](#_Toc134525938)

[F. CURSOR IMPLÍCITO 8](#_Toc134525939)

[1) Curso implícito para comandos LMD 8](#_Toc134525940)

[2) Cursor implícito para recuperar datos de una consulta 9](#_Toc134525941)

[G. GESTIÓN DE EXCEPCIONES: EXCEPCIONES PREDEFINIDAS 9](#_Toc134525942)

[H. ESTRUCTURAS DE CONTROL 10](#_Toc134525943)

[1) If Else 10](#_Toc134525944)

[2) Case 10](#_Toc134525945)

[3) Loop – while 11](#_Toc134525946)

[4) For 11](#_Toc134525947)

[TIPOS DE PROGRAMAS 12](#_Toc134525948)

# INTRODUCCIÓN PL/SQL (PROCEDURAL LANGUAGE / STRUCTURED QUERY LANGUAGE)

PL/SQL es un lenguaje de programación que amplía la funcionalidad de SQL. Permite usar sentencias SQL para manipular datos y sentencias de control de flujo para procesar los datos.

Los programas creados con PL/SQL se pueden almacenar en la base de datos como cualquier otro objeto (tablas, índices, vistas…), quedando disponibles para su ejecución.

PL/SQL fue creado por Oracle y actualmente todos los gestores de bases de datos utilizan algún lenguaje parecido para poder programar las bases de datos.

1. Principales ventajas
   1. PL/SQL combina la potencia de SQL para la manipulación de datos, con la potencia de los lenguajes de programación para procesar los datos.
   2. La ejecución se produce en el servidor, lo que implica menor tráfico en la red y menos carga en el cliente. Además, el código se agrupa en bloques, lo que mejora el rendimiento.
   3. Los disparadores de bases de datos permiten implementar reglas complejas de negocio y auditoría de datos. Por ejemplo, si se introduce un registro en una tabla de facturas se pueden recalcular de forma automática campos de otras tablas, o se puede invalidar la operación si no se cumplen ciertos requisitos de negocio.
2. Arquitectura

PL/SQL es una tecnología integrada en el servidor Oracle capaz de ejecutar bloques de código en coordinación con el motor SQL.

También es posible que esté integrada en herramientas externas al servidor.

Antes de almacenarse el programa PL/SQL, el código es analizado sintácticamente y se comprueba que las referencias a objetos de la base de datos sea correcta (por ejemplo que las tablas existan).

En caso de que todo sea correcto, el motor PL/SQL se encarga de enviar al motor SQL los comandos SQL integrados en el bloque.

Por último ejecutará las órdenes procedimentales PL/SQL combinándolas con el resultado de los comandos SQL.

Normalmente el motor PL/SQL está en el servidor de la base de datos (SGBD), pero en caso de que esté en una herramienta externa, ejecutará las órdenes procedimentales y enviará los comandos SQL al servidor.

1. Interacción con el usuario

PL/SQL está diseñado para trabajar con grandes volúmenes de información eficazmente, pero no para interactuar con el usuario. No dispone de órdenes que permitan capturar datos desde teclado ni mostrar resultados por pantalla.

No obstante, con fines de depuración de programas, Oracle incorpora el paquete DBMS\_OUTPUT (permite visualizar textos en pantalla) y variables de sustitución (permite introducir valores por teclado).

1. Paquete

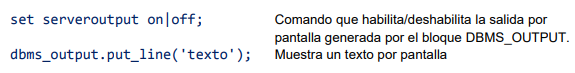
Un paquete es un contenedor que agrupa programas de la base de datos. En un entorno de producción se crean paquetes por área de negocio o para agrupar utilidades que dan soporte a otras aplicaciones.

Para ejecutar un programa que está contenido en un paquete se utiliza la nomenclatura:



Algunos paquetes ya están predefinidos en la base de datos. Los más relevantes:

* STANDARD: incorpora funciones de uso general (ROUND, NVL, ABS, TO\_CHAR…). Este es un caso particular en el que no se indica el nombre del paquete en las llamadas a los programas. Se ejecuta round(25.6) en lugar de standard.round(25.6).
* DBMS\_OUTPUT: incorpora funciones para depurar programas PLSQL. Para emplearlo debemos conocer dos utilidades:



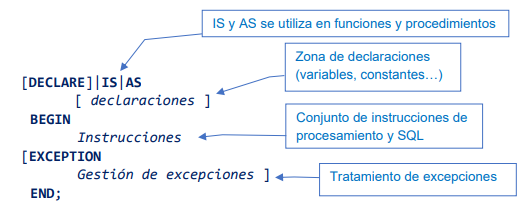
# CONCEPTOS BÁSICOS

## BLOQUE

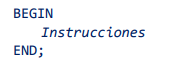
El bloque es la unidad de trabajo en PL/SQL. Se divide en tres partes claramente definidas:

* Zona de declaraciones (opcional).
* Procesamiento de instrucciones.
* Tratamiento de excepciones (opcional).

Sintaxis básica:



El bloque mínimo tendrá la siguiente estructura:



Es posible realizar anidación de bloques, introduciendo bloques dentro de la sección de instrucciones y de tratamiento de excepciones (no en la zona declarativa):

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

## COMENTARIOS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## SOPORTE PARA COMANDOS SQL

En un bloque PL/SQL se pueden ejecutar comandos SQL para manipulación y consulta de datos.

Los comandos de manipulación se integran directamente. Por ejemplo:

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Los comandos de consulta de datos no muestran el resultado en pantalla directamente, sino que lo guardan en un área de memoria denominada cursor, a la que se accede utilizando variables. Por esa razón la integración de un comando SELECT dentro de un bloque PL/SQL no es directa. Por ejemplo:



## OPERADORES

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

## VARIABLES

Las variables del programa se definen en la zona de declaraciones del bloque. La sintaxis es:



Se puede inicializar una variable al declararla. Ejemplos:



1. Tipos de datos

Los tipos de datos que se pueden emplear son:

Tabla

Descripción generada automáticamente

1. Variables de sustitución

Con la finalidad de depurar programas (nunca con la de que sean empleadas por usuarios finales), es posible tomar valores por teclado a través de variables de sustitución.

Solo pueden emplearse en bloques PL/SQL anónimos, y fuera de estructuras repetitivas (bucles LOOP).

Estas variables no es necesario declararlas en la zona de declaración, sino que se utilizan directamente poniendo su nombre precedido por &:



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si se quiere utilizar una variable de sustitución varias veces en el programa, será necesario cargar su valor en una variable normal. Por ejemplo:

Texto

Descripción generada automáticamente

## CURSOR IMPLÍCITO

Un cursor implícito es un área de memoria donde se guarda información sobre el último comando SQL ejecutado en un bloque PL/SQL.

Un cursor implícito no es necesario declararlo como cursor en la zona de declaraciones.

Tiene limitaciones:

* Los datos del cursor estarán disponibles únicamente tras ejecutar el comando sql, no podrán recuperarse más adelante.
* No es válido para consultas SELECT que devuelvan más de un registro.

Para salvar estas limitaciones se utilizan los cursores explícitos.

1. Curso implícito para comandos LMD

Un cursor implícito dispone de varios atributos con información sobre la ejecución del último comando, entre ellos:

* SQL%ROWCOUNT devuelve el número de filas afectadas por el último comando (INSERT, UPDATE, DELETE).

1. Cursor implícito para recuperar datos de una consulta

En el caso de los datos devueltos por una consulta un cursor implícito soportará solo los SELECT que tienen un único registro como resultado (solo devuelve una fila).

Si se utiliza un cursor implícito para obtener datos de una consulta que devuelve más de una fila, se producirá la excepción TOO\_MANY\_ROWS. Si no devuelve ninguna fila se generará NO\_DATA\_FOUND.

Si se quiere recuperar el registro resultado de una consulta, solo será necesario declarar las variables que van a recoger los datos devueltos. El tipo de datos de esas variables debe ser el mismo de los datos devueltos por la consulta.

El formato básico es:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## GESTIÓN DE EXCEPCIONES: EXCEPCIONES PREDEFINIDAS

Las excepciones sirven para tratar errores, situaciones y mensajes de aviso. En el bloque PL/SQL existe una zona de excepciones cuyo formato es:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una excepción está asociada a un determinado evento. Cuando ocurre ese evento, automáticamente se deja de ejecutar la sección de instrucciones del bloque PL/SQL y se pasa a la sección de excepciones.

En ella se busca el manejador (WHEN) del evento que se ha producido y ejecuta el código asociado. Si no lo encuentra se ejecutará el manejador genérico (WHEN OTHERS).

El manejador se encargará de realizar una acción, por ejemplo dar marcha atrás a los cambios (rollback) o mostrar un mensaje de error o aviso.

Una vez ejecutado el código asociado a la excepción, se finaliza el programa PL/SQL.

Es posible crear excepciones personalizadas, aunque existen excepciones predefinidas como:

* NO\_DATA\_FOUND: un comando SELECT no ha devuelto ninguna fila (no tiene por qué ser un error)
* TOO\_MANY\_ROWS: un comando SELECT ha devuelto más de una fila (requiere usar un cursor explícito)

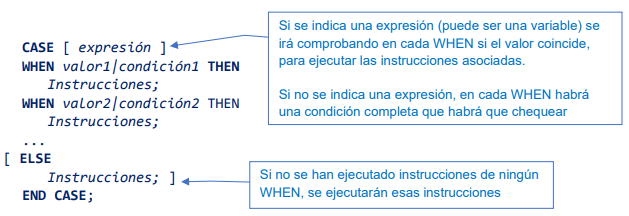
## ESTRUCTURAS DE CONTROL

1. If Else

Escala de tiempo

Descripción generada automáticamente con confianza media

1. Case



Escala de tiempo

Descripción generada automáticamente

1. Loop – while

Texto

Descripción generada automáticamente

1. For

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# TIPOS DE PROGRAMAS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente