

UNIVERSIDAD TECNOLÓGICA DE PANAMA
FACULTAD DE INGENIERIA DE SISTEMAS
LICENCIATURA EN INGENIERIA DE SOFTWARE

BASE DE DATOS II ORACLE PROGRAMACION PL/SQL

Fundamentos del Lenguaje PL-SQL -ORACLE



Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

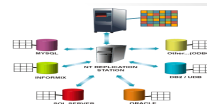
{ 1 }

1

CONTENIDO

Capítulo IV. Procedimientos

- **Fundamentos de Lenguaje PL/SQL**
- **Cursores**
- **Procedimientos**




Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 2 }

2

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



El diagrama muestra la arquitectura de Oracle PL/SQL. En el centro hay un icono de servidor con el texto 'PL/SQL ENGINE'. Arriba a la izquierda hay un icono de cliente con el texto 'CLIENT'. Arriba a la derecha hay un icono de servidor con el texto 'ORACLE SERVER'. Abajo a la izquierda hay un icono de servidor con el texto 'ORACLE SERVER'. Abajo a la derecha hay un icono de servidor con el texto 'ORACLE SERVER'. En el centro, debajo del 'PL/SQL ENGINE', hay un icono de servidor con el texto 'ORACLE SERVER'. Las flechas indican la comunicación entre los componentes.

INTRODUCCION

Para el desarrollador es necesario conocer en forma previa la sintaxis básica del lenguaje PL/SQL:

- *Las reglas sintácticas son los bloques, componentes de cualquier lenguaje de programación.*
- *Los componentes de un bloque PL-SQL.*
- *Las declaraciones de variables y tipos de datos*
- *Las estructuras procedimentales básica.*
- *Los cursores y subprogramas*

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

3

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE

The diagram illustrates the Oracle database architecture. At the top center is a server icon labeled "SERVER". To its left is a client icon labeled "CLIENT". Below the client is a network icon labeled "NETWORK". Below the network is a box labeled "SQL*NET". To the right of the network is another network icon labeled "NETWORK". Below this network is a server icon labeled "SERVER". At the bottom right is a database icon labeled "DATABASE". Arrows indicate the flow of communication between these components.

INTRODUCCION

Para el desarrollador es necesario conocer en forma previa la sintaxis básica del lenguaje PL_SQL:

- Las guías de estilo de programación de PL-SQL y se presentan técnicas que ayudan a escribir códigos bien elegantes y de fácil comprensión.
- Las ordenes de PL-SQL son procedimentales, al igual que las ordenes de SQL. Incluyen declaraciones de variables, llamadas a procedimientos y estructuras de bucles. Con referencia a las ordenes de SQL estas nos permiten acceder a las bases de datos.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

4

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

- Los bloques representan la unidad básica de cualquier programa PL-SQL.
- Todos los PL-SQL están compuestos por bloques, que pueden estar situados de forma secuencial (uno detrás de otro) o pueden estar anidados (uno dentro de otro).
- Hay diferentes de tipos de bloques:
 - Los bloques anónimos ('anonymous Blocks') se construyen, por regla general, de manera dinámica y se ejecuta una sola vez.
 - Los bloques nominado ('named Blocks') son bloques dinámicos con una etiqueta que le da al bloque un nombre. Se construyen, por regla general, de manera dinámica y se ejecuta una sola vez.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 5 }

5

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

- Hay diferentes de tipos de bloques:
 - Los *subprogramas* son procedimientos, paquetes y funciones almacenados en la base de datos. Estos bloques no cambian, por regla general, después de su construcción y se ejecutan múltiples veces.
 - Los subprogramas se ejecutan explícitamente, mediante una llamada al procedimiento, paquetes o funciones.
 - Los disparadores ('triggers') son bloques nominados que también se almacenan en la base de datos. Tampoco cambian, generalmente, después de su construcción se ejecutan múltiples veces.
 - Los disparadores se ejecutan de manera implícita cada vez que tiene lugar un suceso de disparo. El suceso de disparo es una orden del lenguaje DML que se ejecuta sobre una tabla de la base de datos, entre estar ordenes están INSERT, UPDATE y DELETE.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 6 }

6

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

1. Ejemplo de un bloque anónimo:

```
DECLARE
/* Declaración de las variables que se usan en este bloque */
v_Num1 NUMBER := 1;
v_Num2 NUMBER := 2;
v_String1 VARCHAR(50) := 'Hello World!';
v_String2 VARCHAR(50) := '–This message brought to you by PL/SQL!';
V_OutputStr VARCHAR2(50);
BEGIN
/* Primero inserta dos filas en temp_table, utilizando los valores de las variable */
INSERT INTO temp_table (num_col, char_col) VALUES (v_num1, v_String1);
INSERT INTO temp_table (num_col, char_col) VALUES (v_num2, v_String2);
/*Ahora consulta temp_table para las dos filas que se acaban de insertar y las presenta en pantalla utilizando el paquete
DBMS_OUTPUT */
SELECT char_cal INTO v_OutputStar
From temp_table
WHERE num_col = v_num1;
DBMS_OUTPUT.PUT_LINE(v_OutputStar);
/*SELECT char_cal INTO v_OutputStar
From temp_table
WHERE num_col = v_num2;
DBMS_OUTPUT.PUT_LINE(v_OutputStar); */
END;
```

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

2. Ejemplo de un bloque nominado, esta etiqueta se coloca antes de la palabra clave:

```
<<InsertIntoTemp>>
DECLARE
/* Declaración de las variables que se usan en este bloque */
v_Num1 NUMBER := 1;
v_Num2 NUMBER := 2;
v_String1 VARCHAR(50) := 'Hello World!';
v_String2 VARCHAR(50) := '–This message brought to you by PL/SQL!';
V_OutputStr VARCHAR2(50);
BEGIN
/* Primero inserta dos filas en temp_table, utilizando los valores de las variable */
INSERT INTO temp_table (num_col, char_col) VALUES (v_num1, v_String1);
INSERT INTO temp_table (num_col, char_col) VALUES (v_num2, v_String2);
/*Ahora consulta temp_table para las dos filas que se acaban de insertar y las presenta en pantalla utilizando
el paquete DBMS_OUTPUT */
SELECT char_cal INTO v_OutputStar
From temp_table
WHERE num_col = v_num1;
DBMS_OUTPUT.PUT_LINE(v_OutputStar);
SELECT char_cal INTO v_OutputStar
From temp_table
WHERE num_col = v_num2;
DBMS_OUTPUT.PUT_LINE(v_OutputStar);
END InsertIntoTemp;
```

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

3. Podemos transformar un bloque en un procedimiento almacenado reemplazando la palabra clave **DECLARE** con las palabras claves **CREATE OR REPLACE PROCEDURE**

```
CREATE OR REPLACE PROCEDURE InsertIntoTemp AS
/* Declaración de las variables que se usan en este bloque */
v_Num1 NUMBER := 1;
v_Num2 NUMBER := 2;
v_String1 VARCHAR(50) := 'Hello World!';
v_String2 VARCHAR(50) := '-This message brought to you by PL/SQL!';
v_OutputStr VARCHAR2(50);
BEGIN
/* Primero inserta dos filas en temp_table, utilizando los valores de las variable */
INSERT INTO temp_table (num_col, char_col) VALUES (v_num1, v_String1);
INSERT INTO temp_table (num_col, char_col) VALUES (v_num2, v_String2);
/* Ahora consulta temp_table para las dos filas que se acaban de insertar y las presenta en pantalla utilizando
el paquete DBMS_OUTPUT */
SELECT char_col INTO v_OutputStr
WHERE num_col = v_num1;
DBMS_OUTPUT.PUT_LINE(v_OutputStr);
SELECT char_col INTO v_OutputStr
WHERE num_col = v_num2;
DBMS_OUTPUT.PUT_LINE(v_OutputStr);
END;
```

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 9 }

9

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

El Bloque PL-SQL

4. Podemos construir un disparador sobre la tabla **temp_table** para que solo introduzca valores positivos en **num_col**. Este disparador se activara cada vez que se inserte una nueva columna en **temp_table** o actualice una fila ya existente

```
CREATE OR REPLACE TRIGGER OnlyPositive
BEFORE INSERT OR UPDATE OF num_col
ON temp_table
FOR EACH ROW
BEGIN
IF :new.num_col < 0 THEN
RAISE_APPLICATION_ERROR (-2000, 'Please insert a positive value');
END IF;
END OnlyPositive ;
```

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 10 }

10

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

ESTRUCTURA BASICA DE UN BLOQUE

Todos los bloques tienen tres secciones diferenciadas:

```
DECLARE
/*Sección declarativa */
BEGIN
/*Sección ejecutable */
EXCEPTION
/* Sección Excepciones */
END;
```

- **La sección declarativa:** donde se localizan todas la variables, cursores y tipos usados por los bloques. También podemos declarar en esta sección las funciones y procedimientos locales, que solo están disponibles para este bloque.
- **La sección ejecutable:** donde se lleva a cabo el trabajo del bloque. En esta sección pueden aparecer tantos ordenes de SQL como ordenes procedimentales.
- **La sección de excepciones:** aquí se lleva a cabo el tratamiento de errores. El código que se incluya en esta sección no se ejecuta a menos que ocurra un error.
- Las palabras claves del bloque **DECLARE, BEGIN, EXCEPTION y END** delimitan cada una de las secciones. El punto y coma también es obligatorio, forma parte de la sintaxis del bloque.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 11 }

11

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

ESTRUCTURA BASICA DE UN BLOQUE

Ejemplo de un bloque anónimo con todas la sección:

```
DECLARE
/* Inicio de la sección declarativa */
v_StudentID NUMBER(5) := 10000; -- Variable numérica inicializada con 10,000
v_FirstName VARCHAR2(50);      -- Cadena de caracteres de longitud variable con longitud máxima de 20
BEGIN
/* Inicio de la seccion de ejecutable */
-- Recupera el nombre del estudiante con ID igual a 10,000
SELECT firt_name INTO v_FirstName
FROM students
WHERE id = v_StudentID;
EXCEPTION
/* Inicio de la sección de excepciones */
WHEN NO_DATA_FOUND THEN
-- Manejo de la codificación de error
INSERT INTO log_table (info)
VALUES ('Student 10,000 does not exists!');
END;
```

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

{ 12 }

12

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

UNIDADES LEXICAS

Cualquier programa PL/SQL esta compuesto por unidades léxicas: los bloques que son los componentes del lenguaje. Esencialmente, una unidad léxica es una secuencia de caracteres, donde los caracteres pertenecen a un conjunto de caracteres permitidos en el lenguaje PL/SQL.

Las letras mayúsculas y minúsculas	A-Z y a-z
Los dígitos	0 - 9
Espacios en blanco	Tabuladores, caracteres de espaciados y retornos de carro
Símbolos matemáticos	+ - * / < > =
Símbolos de puntuación	() { } [] ; ' " @ # % \$ ^ & _

Cualquier elemento de estos conjuntos puede ser usado como parte de un programa PL/SQL. No se diferencia entre la mayúscula y minúscula, excepto en el interior de una cadena delimitada por comillas.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

[13]

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

UNIDADES LEXICAS

Las unidades léxicas pueden clasificarse en *identificadores*, *delimitadores*, *literales* y *comentarios*.

IDENTIFICADORES

- Son usados para dar nombre a los objetos PL/SQL, como variables, cursores, tipos, programas.
- Constan de una letra, seguida por una secuencia opcional de caracteres, que pueden incluir letras, números, signos de dólar(\$), caracteres de subrayado y símbolos de almohadilla(#). Los demás caracteres no son permitidos.
- La longitud máxima de un identificador es de 30 caracteres, y todos los caracteres son significativos.

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

[14]

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE

ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

UNIDADES LEXICAS

Las unidades léxicas pueden clasificarse en *identificadores*, *delimitadores*, *literales* y *comentarios*.

IDENTIFICADORES LEGALES E ILEGALES

Identificadores Legales	Identificadores Ilegales
x	x+y
v_StudentID	_temp_
pempvari	First Name
v1	Este_es_un_identificador_muy_largo
v2	1_variable
social_security_#	

No hay diferencia entre mayúsculas y minúsculas por lo que estos identificadores son equivalentes

```
Room_Description
Room_description
ROOM_DESCRIPTION
rOom_DeSCRIPTION
```

Implementación de Base de Datos
II Por. Ing. Henry Lezcano I
Semestre del 2024

15

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE

ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

UNIDADES LEXICAS

Las unidades léxicas pueden clasificarse en *identificadores*, *delimitadores*, *literales* y *comentarios*.

DELIMITADORES

Los delimitadores son símbolos formados por uno o mas caracteres que tienen un significado especial para PL/SQL. Son usadas para separar unos identificadores de otros.

Símbolo	Descripción	Símbolo	Descripción
+	Operador de Suma	-	Operador de resta
*	Operador de Multiplicación	/	Operador de división
=	Operador de Igualdad	<	Operador menor que
>	Operador Mayor que	(Delimitador inicial de expresión
)	Delimitador Final de Expresión	;	Terminador de orden
.	Selector de Componente	,	Separador de elemento
'	Delimitador de Cadena de Caracteres	@	Indicador de enlace a base de datos
:	Indicador de variable de asignacion	"	Delimitador de cadena entrecomillada
<>	Operador distinto de	**	Operador de Exponenciación

Implementación de Base de Datos

II Por. Ing. Henry Lezcano I

Semestre del 2024

16

4.1. Fundamentos del Lenguaje PL-SQL -ORACLE



ESTRUCTURA DE LOS BLOQUES DE PROGRAMA

UNIDADES LEXICAS

Las unidades léxicas pueden clasificarse en *identificadores*, *delimitadores*, *literales* y *comentarios*.

DELIMITADORES-Continuación

Los delimitadores son símbolos formados por uno o mas caracteres que tienen un significado especial para PL/SQL. Son usadas para separar unos identificadores de otros.

Símbolo	Descripción	Símbolo	Descripción
~=	Operador distinto de , !=	!=	Operador distinto que , <>
<=	Operador menor igual que	^=	Operador distinto que , ~=
:=	Operador de asignación	>=	Operador mayor igual a
..	Operador de rango	=>	Operador de asociación
<<	Delimitador de comienzo de etiqueta		Operador de concatenación
--	Indicador de comentario una sola línea	>>	Delimitador de fin de etiqueta
/	Indicador de cierre de comentario multilíneas	/	Indicador de cierre de comentario multilíneas
<tab>	Carácter de tabulación	<space>	Espacio
%	Indicador de atributo	<cr>	Retorno de carro