

ÉRICA PALOMINO

TEMA 5

LENGUAJES DE BASES DE DATOS. PL/SQL

erica.palomino@escuelaartegranada.com

ESCUELAARTEGRANADA

CONOCIENDO PL/SQL

Lo primero es recordar que SQL no es un lenguaje de programación, es un **lenguaje de CONSULTAS**.

PL/SQL amplía SQL con los elementos característicos de los lenguajes de programación: variables, sentencias de control de flujo, bucles...

Con PL/SQL vamos a poder programar las unidades de programa de la base de datos ORACLE, están son:

- Procedimiento almacenados
- Funciones
- Triggers
- Scripts

CONOCIENDO PL/SQL

PL/SQL **NO** es **CASE-SENSITIVE**, es decir, no diferencia entre mayúsculas y minúsculas

Comentarios:

- -- Comentario de línea simple
- /* */ Comentario de multilínea

De asignación	:=
Aritméticos	+ - * / ** (exponente) mod (resto de la división)
Relacionales	= <> < > <= >=
Lógicos	And Or Not
Concatenación	

BLOQUES PL/SQL

BLOQUES PL/SQL

Es la unidad mínima de programación en PL/SQL

Está dividida en 3 partes diferenciadas y delimitadas por palabras clave:

```
declare
    /*Parte declarativa*/

begin
    /*Parte de ejecucion*/

[ exception ]
    /*Parte de excepciones*/

end;
```



SECCIÓN DE DECLARACIÓN

SECCIÓN DE DECLARACIÓN

Se declaran las variables que se van a necesitar.

Siempre indicamos nombre y tipo de dato.

También se declaran aquí otros elementos:

- Excepciones
- Constantes
- Cursores

Sintaxis genérica:

Nombre_Variable [CONSTANT] tipo_dato [:=valor inicial];

SECCIÓN DE DECLARACIÓN

DECLARE

```
v_location VARCHAR2(15) := 'Granada';  
PI CONSTANT NUMBER := 3.1416;  
v_nombre tabla_empleados.nombre%TYPE;  
  
reg_datos micursor%ROWTYPE;
```

BEGIN

```
/*Parte de ejecucion*/
```

EXCEPTION

```
/*Parte de excepciones*/
```

END;



SECCIÓN DE EJECUCIÓN

SECCIÓN DE EJECUCIÓN

Dentro de un bloque PL/SQL se pueden mostrar mensajes personalizados utilizando para ello la función:

DBMS_OUTPUT.PUT_LINE(cadena)

```
Begin
    dbms_output.put_line('Hola mundo');
End;
```

EJERCICIO

Crear un bloque PL/SQL que muestre el mensaje:

Hola mundo

Hoy es un bonito día

Adiós mundo

SECCIÓN DE EJECUCIÓN

La cadena de caracteres que queremos mostrar se puede concatenar con variables o con otros símbolos utilizando el operador de concatenación:

```
Declare
    nombre varchar2(10);
Begin
    nombre:='Juan';
    dbms_output.put_line('Hola '||nombre||' qué tal?');
End;
```

SECCIÓN DE EJECUCIÓN

Dentro de un bloque PL/SQL se pueden utilizar **cualquier sentencia SQL**.

La única sentencia SQL que sufre algún cambio es **SELECT**.

En PL/SQL, el resultado de un SELECT debe ser recogido en una **variable**

Para ello se añade la cláusula **INTO** a todas las que ya conocemos de SELECT

EJERCICIOS

Mostrar el nombre de la sala con código S3.

Mostrar el nombre y la capacidad de la sala con código S3

Mostrar el nombre de la sala con más capacidad.

Mostrar el nombre de la película que se proyecta en la sala S2
a las 23.00

Mostrar la capacidad y el número de filas de la sala llamada
Oceanía

IMPORTANTE

PL/SQL permite definir el tipo de una variable copiando del tipo de otra.

Mediante la utilización de las cláusulas %TYPE y %ROWTYPE.

- %type Copia el tipo de una variable o columna existente.

v_nombre duenios.nombre%type;

- %rowtype Copia el tipo de una fila existente.

v_fila duenios%rowtype;

EJERCICIOS

Mostrar todos los datos de la película que se proyecta en la S4 a las 20.00

Mostrar todos los datos de la sala más grande del cine.

Mostrar los datos de la sala en la que se proyecta 'El autor' a las 17.00

Mostrar los datos de la sesión más vendida.

Mostrar el nombre de la película de la que se han vendido más entradas en total.

Mostrar la nacionalidad de la película más vista en una sesión.

IMPORTANTE

Podemos pedir datos al usuario utilizando **ACCEPT**

Esto se hace siempre fuera del bloque

ACCEPT variable **PROMPT** mensaje

Para utilizar el la variable creada con ACCEPT debemos poner delante el símbolo **&**

EJERCICIOS

Mostrar todos datos de la película con el título que el usuario introduzca por teclado.

Pedir al usuario un código de sala y mostrar sus datos.

Mostrar la calificación por edad de una película que el usuario introduzca por teclado.

Pedir al usuario un número de sala y una hora y mostrar qué película se proyecta en esa sala a esa hora.

Mostrar el número de entradas vendidas para una determinada película.
(Pedir código de película, de sala y hora)

Mostrar el horario de una determinada película en una determinada sala.



ESTRUCTURAS DE CONTROL

ESTRUCTURAS DE CONTROL

En PL/SQL solo disponemos de la estructura condicional IF

IF (expresion) **THEN**

-- Instrucciones

ELSIF (expresión) **THEN**

-- Instrucciones

ELSE

-- Instrucciones

END IF;

EJERCICIOS

Pedir al usuario un número y decir si el número es positivo o negativo.

Pedir al usuario su edad y decir si es mayor de edad o no.

Comprobar si la sala 4 está llena para la película p4 a las 20.00.

Comprobar si quedan más de 10 entradas para ver "Astérix y Obélix: el reino medio" a las 18.00

Pedir al usuario un nombre de película y comprobar si hay aún entradas disponibles para esa película a la hora que el usuario desee. El usuario deberá indicar cuántas entradas quiere comprar. En caso de que haya entradas, el bloque se las venderá a 6,5 euros. En caso contrario le indicará al usuario que elija otra hora.

EJERCICIOS

Escribir un bloque PL/SQL tal que el usuario dará dos nombres de salas y el bloque le indicará qué sala es más grande, indicando además la diferencia entre una sala y la otra.

Pedir al usuario un nombre de película. El bloque PL/SQL debe indicar al usuario si la película es nacional o extranjera, en caso de que sea nacional, además deberá mostrar todos los datos de la película.



ESTRUCTURAS REPETITIVAS

ESTRUCTURAS DE REPETICIÓN

En PL/SQL existen 3 estructuras de repetición:

- **WHILE**

```
WHILE (condición) LOOP  
    -- Instrucciones  
END LOOP;
```

- **LOOP**

```
LOOP  
    -- Instrucciones  
    IF (expresión) THEN  
        -- Instrucciones  
    EXIT;  
    END IF;  
END LOOP;
```

- **FOR**

```
FOR contador IN  
    minimo .. maximo LOOP  
    -- Instrucciones  
END LOOP;
```


EJERCICIOS WHILE

Hacer un bloque PL/SQL que muestre una cuenta regresiva desde 10 hasta 0.

Hacer un bloque PL/SQL que pida al usuario dos números y muestre una cuenta desde el mayor al menor. Controlar qué número es mayor y qué número es menor.

Hacer un bloque PL/SQL que muestre por pantalla la información de todas las salas que hay en la base de datos

EJERCICIOS LOOP

Hacer un bloque PL/SQL que muestre una cuenta regresiva desde 10 hasta 0.

Hacer un bloque PL/SQL que pida al usuario dos números y muestre una cuenta desde el mayor al menor. Controlar qué número es mayor y qué número es menor.

Hacer un bloque PL/SQL que muestre por pantalla la información de todas las salas que hay en la base de datos

EJERCICIOS FOR

Hacer un bloque PL/SQL que muestre una cuenta regresiva desde 10 hasta 0.

Hacer un bloque PL/SQL que pida al usuario dos números y muestre una cuenta desde el mayor al menor. Controlar qué número es mayor y qué número es menor.

Hacer un bloque PL/SQL que muestre por pantalla la información de todas las salas que hay en la base de datos

EJERCICIOS FOR

Hacer un bloque PL/SQL que pida al usuario

Hacer un bloque PL/SQL que pida al usuario dos números y muestre una cuenta desde el mayor al menor. Controlar qué número es mayor y qué número es menor.

Hacer un bloque PL/SQL que muestre por pantalla la información de todas las salas que hay en la base de datos



ITERAR SOBRE CONSULTAS

ITERAR SOBRE CONSULTAS

Podemos aprovecharnos de la estructura **FOR** para recorrer las filas de la tabla resultado de una consulta, al final lo que resulta es una estructura parecida a **FOREACH**:

```
FOR iterador IN ( CONSULTA ) LOOP  
    -- instrucciones  
END LOOP;
```

El iterador funciona como una variable **ROWTYPE** por lo que para acceder a los datos hay que especificar la columna correspondiente. No hace falta declararlo.

EJERCICIOS

Usando FOR calcula la ocupación total de una sala indicada por el usuario.

Pide al usuario una nacionalidad. Se mostrarán las calificaciones de todas las películas de dicha nacionalidad indicando si es 'TP' que es 'Para todos los públicos', si es '7' que es 'Para mayores de 7 años' y si es 18; 'Para mayores de 18 años'.

Pedir al usuario una sala, hacer un FOR inverso en el que, si la ocupación es mayor a 50 se reducirá en un 10%, en caso contrario se aumenta un 5%.



MANEJO DE ERRORES

MANEJO DE ERRORES

Hasta ahora, si nos daba algún fallo en el código o en la consulta, se paraba la ejecución por completo y había que reiniciar el programa.

Sin embargo, existe una forma de 'capturar' esos errores para modificar la ejecución y que no termine de forma temprana. Es lo que se llama **manejo de excepciones**.

MANEJO DE ERORRES

Para implementar el manejo de excepciones en PL/SQL utilizaremos el bloque **EXCEPTIONS**, siguiendo la estructura:

BEGIN

--instrucciones--

EXCEPTION

WHEN e1 THEN

--instrucciones--

WHEN e2 THEN

--instrucciones--

END;

Los errores más comunes son **NO_DATA_FOUND** y **TOO_MANY_ROWS**.

MANEJO DE ERRORES

También podemos declarar nuestros propios errores dentro del bloque **DECLARE**, creando una variable de tipo **EXCEPTION**. (**e1 EXCEPTION**)

Para lanzar el error se utiliza la sentencia **RAISE** dentro de la condición que queramos, seguido del nombre que le hayamos dado a la variable. (**RAISE e1**)

Para recoger el error dentro del bloque **EXCEPTION** utilizamos el nombre dentro de la condición **WHEN**. (**WHEN e1 THEN**)