

### Prácticas sobre conceptos básicos de: PL-SQL

Antes de hacer los ejercicios, analizar los ejemplos que os he realizado para que analicéis de forma práctica como plantearlos.

#### DEFINICIÓN DE VARIABLES Y TIPOS DE DATOS.

**Ejemplo1.** Se trata de definir variables y tipos de datos. Algunas con contenido y mostrar mensajes en pantalla. En este caso se trata de una operación algebraica que consiste en incrementar a una bonificación dada el importe de las horas por su salario hora. Se trata de practicar con variables y tipos de datos.

```
declare
salario NUMBER;
horas_trabajadas number :=40;
salario_hora number:= 22.50;
bonificacion number := 150;
contador number :=0;
begin
salario := (horas_trabajadas * salario_hora) + bonificacion;
contador := contador +1;
dbms_output.put_line(salario);
dbms_output.put_line(contador);
end;
```

```
1 declare
2 salario NUMBER;
3 horas_trabajadas number :=40;
4 salario_hora number:= 22.50;
5 bonificacion number := 150;
6 contador number :=0;
7 begin
8 salario := (horas_trabajadas * salario_hora) + bonificacion;
9 contador := contador +1;
10 dbms_output.put_line(salario);
11 dbms_output.put_line(contador);
12 end;
13
```

Resultados

Explicar

Describir

SQL Guardado

Historial

1050  
1

Sentencia procesada.

0,01 segundos

## ESTRUCTURAS BÁSICAS DE CONTROL

### Usos de IF :

#### Ejemplo2

```
declare
fecha_nacimiento varchar2(10) := '17-04-2020';
fecha_actual varchar2(10) := '20-04-2020';
salario NUMBER;
horas_trabajadas number := 40;
salario_hora number := 22.50;
bonificacion number := 150;
contador number := 0;
begin
if fecha_nacimiento > fecha_actual then
salario := (horas_trabajadas * salario_hora) + bonificacion;
contador := contador + 1;
end if;
dbms_output.put_line(salario);
dbms_output.put_line(contador);
end;
```

#### Ejemplo2:

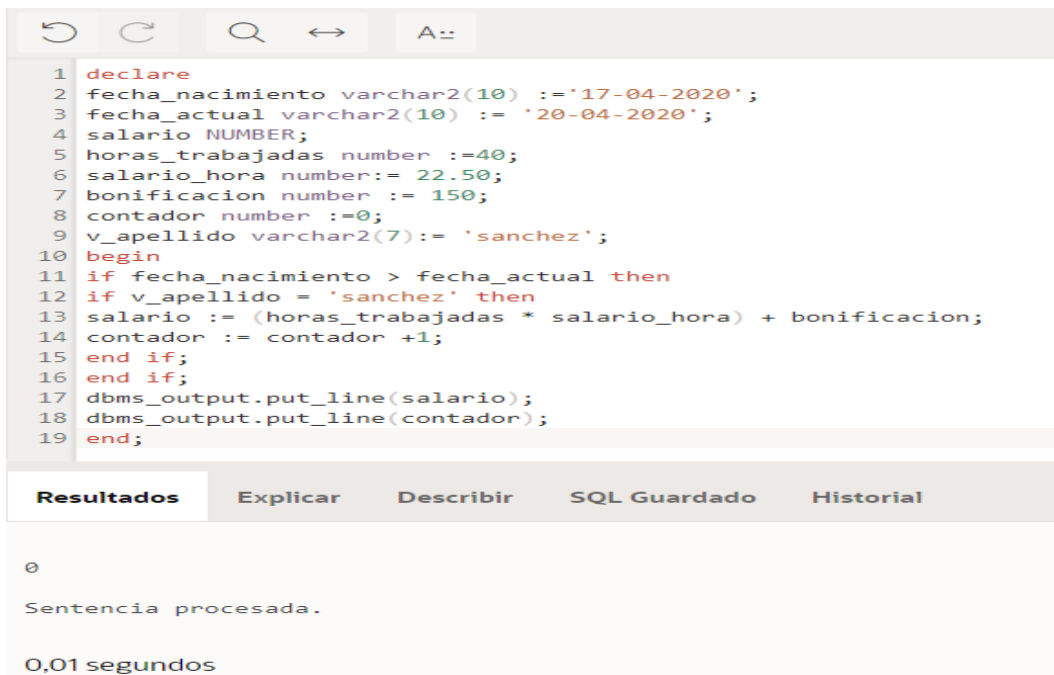
```
1 declare
2 fecha_nacimiento varchar2(10) := '17-04-2020';
3 fecha_actual varchar2(10) := '20-04-2020';
4 salario NUMBER;
5 horas_trabajadas number := 40;
6 salario_hora number := 22.50;
7 bonificacion number := 150;
8 contador number := 0;
9 begin
10 if fecha_nacimiento > fecha_actual then
11 salario := (horas_trabajadas * salario_hora) + bonificacion;
12 contador := contador + 1;
13 end if;
14 dbms_output.put_line(salario);
15 dbms_output.put_line(contador);
16 end;
17
```

Resultados	Explicar	Describir	SQL Guardado	Historial
0				
Sentencia procesada.				
0,01 segundos				

### Ejemplo3

```
declare
fecha_nacimiento varchar2(10) := '17-04-2020';
fecha_actual varchar2(10) := '20-04-2020';
salario NUMBER;
horas_trabajadas number := 40;
salario_hora number := 22.50;
bonificacion number := 150;
contador number := 0;
v_apellido varchar2(7) := 'sanchez';
begin
if fecha_nacimiento > fecha_actual then
if v_apellido = 'sanchez' then
salario := (horas_trabajadas * salario_hora) + bonificacion;
contador := contador + 1;
end if;
end if;
dbms_output.put_line(salario);
dbms_output.put_line(contador);
end;
```

Ejemplo3:



```
1 declare
2 fecha_nacimiento varchar2(10) := '17-04-2020';
3 fecha_actual varchar2(10) := '20-04-2020';
4 salario NUMBER;
5 horas_trabajadas number := 40;
6 salario_hora number := 22.50;
7 bonificacion number := 150;
8 contador number := 0;
9 v_apellido varchar2(7) := 'sanchez';
10 begin
11 if fecha_nacimiento > fecha_actual then
12 if v_apellido = 'sanchez' then
13 salario := (horas_trabajadas * salario_hora) + bonificacion;
14 contador := contador + 1;
15 end if;
16 end if;
17 dbms_output.put_line(salario);
18 dbms_output.put_line(contador);
19 end;
```

**Resultados** Explicar Describir SQL Guardado Historial

0

Sentencia procesada.

0,01 segundos

### Ejemplo4

#### Usos de CASE:

Al igual que IF, la sentencia CASE selecciona una secuencia de sentencias a ejecutar:

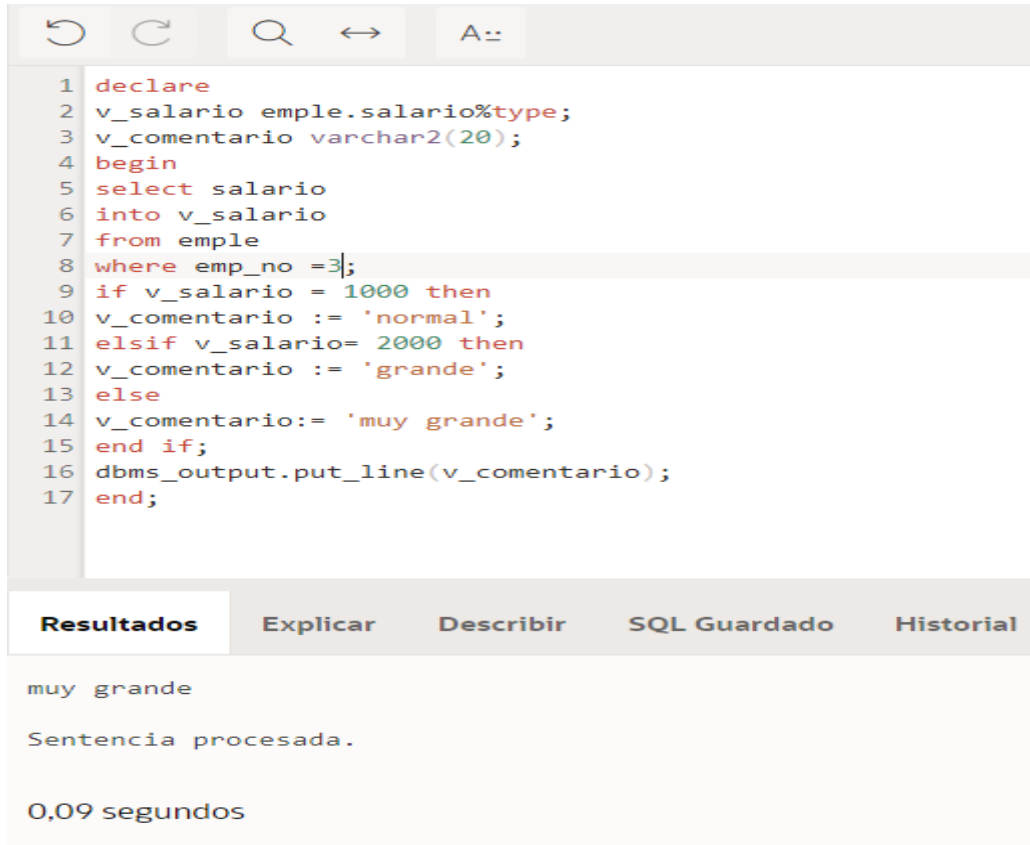
```
declare
fecha_nacimiento varchar2(10) := '17-04-2020';
fecha_actual varchar2(10) := '20-04-2020';
salario NUMBER;
horas_trabajadas number := 40;
salario_hora number := 22.50;
bonificacion number := 150;
contador number := 0;
v_apellido varchar2(7) := 'sanchez';
begin
case fecha_nacimiento
when '17-04-2020' then dbms_output.put_line('correcto');
else dbms_output.put_line('falso');
end case;
end;
```

```
1 declare
2 fecha_nacimiento varchar2(10) := '17-04-2020';
3 fecha_actual varchar2(10) := '20-04-2020';
4 salario NUMBER;
5 horas_trabajadas number := 40;
6 salario_hora number := 22.50;
7 bonificacion number := 150;
8 contador number := 0;
9 v_apellido varchar2(7) := 'sanchez';
10 begin
11 case fecha_nacimiento
12 when '17-04-2020' then dbms_output.put_line('correcto');
13 else dbms_output.put_line('falso');
14 end case;
15 end;
16
```

Resultados	Explicar	Describir	SQL Guardado	Historial
correcto				
Sentencia procesada.				
0,01 segundos				

Ejemplo5 . Más usos de IF , IF-THEN, ELSIF y ELSE a partir de una recuperación con SELECT.

Uso de %type



The screenshot shows a SQL IDE interface. At the top is a toolbar with icons for undo, redo, search, and other functions. Below the toolbar is a text editor containing a PL/SQL script. The script declares a variable `v_salario` of type `emple.salario%type` and a variable `v_comentario` of type `varchar2(20)`. It then begins a block where it selects the salary for employee number 3 into `v_salario`. Following this, it uses an `if-elsif-else` structure to assign a comment to `v_comentario` based on the salary value: 'normal' for 1000, 'grande' for 2000, and 'muy grande' for any other value. The script ends by printing the comment using `dbms_output.put_line` and then terminates with `end;`.

Below the script editor is a results pane with tabs for 'Resultados', 'Explicar', 'Describir', 'SQL Guardado', and 'Historial'. The 'Resultados' tab is active, showing the output of the script execution. The output consists of three lines: the comment 'muy grande', a status message 'Sentencia procesada.', and the execution time '0,09 segundos'.

```
1 declare
2 v_salario emple.salario%type;
3 v_comentario varchar2(20);
4 begin
5 select salario
6 into v_salario
7 from emple
8 where emp_no =3;
9 if v_salario = 1000 then
10 v_comentario := 'normal';
11 elsif v_salario= 2000 then
12 v_comentario := 'grande';
13 else
14 v_comentario:= 'muy grande';
15 end if;
16 dbms_output.put_line(v_comentario);
17 end;
```

**Resultados**   Explicar   Describir   SQL Guardado   Historial

muy grande

Sentencia procesada.

0,09 segundos

### **EJERCICIOS**

Estableciendo las tres partes básicas de un programa PL-SQL como en los ejemplos anteriores, realizar procesos de prueba con las definiciones de variables, usos de IF, CASE y SELECT.

Hacer las prácticas siguientes:

a) Construir un bloque con : declare , begin y end en el que se controle un resultado algebraico de multiplicación y división por = 1000, mayor que 1000 y menor que mil con mensajes asociados. Tenéis que presentar el programa hecho y el resultado en el documento word.

b) Construir un bloque con : declare , begin y end en el que se controle por el contenido de una variable texto fijado si coincide o no y sus mensajes oportunos. Tenéis que presentar el programa hecho y el resultado en el documento word.

c) Construir un bloque con : declare , begin y end en el que se controle un resultado numérico y un dato de un texto fijados en variables si es correcto o no y sus mensajes oportunos. Tenéis que presentar el programa hecho y el resultado en documento word.

d) Construir un bloque con : declare , begin y end. Utilizar un acceso a la tabla EMPLE para recoger el dato de apellido de un determinado empleado y preguntar si coincide con el fijado por vosotros en variable. Controlar que se cumpla o no con mensajes asociados. Tenéis que presentar el programa hecho y el resultado en documento word.

e) Construir un bloque con : declare , begin y end . Utilizar un acceso a la tabla EMPLE para recoger el dato de apellido y el del salario de un determinado empleado y preguntar si coincide con los fijados por vosotros en las variables. Controlar que se cumpla o no con mensajes asociados. Utilizar estructura sentencia IF. Tenéis que presentar el programa hecho y el resultado en documento word.

f) Construir un bloque con : declare , begin y end . Utilizar un acceso a la tabla EMPLE para recoger el dato de apellido de un determinado empleado y preguntar si coincide con el fijado por nosotros en las variables asociadas. Utilizar estructura sentencia CASE. Controlar que se cumpla o no con mensajes asociados. Tenéis que presentar el programa hecho y el resultado en documento word.