CICLO: [DAM]
MÓDULO DE [BASES DE DATOS]

[Tarea Nº 06]

Alumno:
[Juan Carlos Filter Martín]
[15456141A]

Contenido

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Archivo JuanCarlosFilterMartinTarea6.SQL

2. **(RA05_d)** Se han definido y utilizado guiones para automatizar tareas y **(RA05_g)** Se han utilizado estructuras de control de flujo.

Bloque anónimo PL/SQL para calcular el 'salario anual' de un empleado:

✗ Tiene que pedir nombre y apellido por una variable de sustitución

```
nombre_apellidos VARCHAR2(100) :='&Nombre_y_apellidos';
```

✗ Tiene que pedir el salario mensual en otra variable de sustitución

```
salario_mensual NUMBER :=&salario_mensual;
```

Este bloque PL/SQL debe:

✗ Calcular el salario anual : ('salario mensual' * 12) + 'prima' y en función del salario mensual le indicamos que prima le corresponde mediante diferentes IF

- Si tiene más de 21000 = prima +3000

```
BEGIN

    IF salario_mensual * 12 > 21000 THEN
        prima := 3000;
```

- Si tiene entre 12000 y 21000 (incluidos) = prima +1800

```
ELSIF salario_mensual * 12 BETWEEN 12000 AND 21000 THEN
    prima := 1800;
```

- Si tiene menos de 12000 = prima+1000
  (finalmente con un ELSE ya que solo quedaría esta opción)

```
ELSE
    prima := 1000;
END IF;
```

✗ Añadimos a la variable salario_anual el salario mensual*12 + la prima correspondiente y mostramos el resultado por pantalla.

```
salario_anual := (salario_mensual * 12) + prima;
 DBMS_OUTPUT.PUT_LINE('El sueldo anual para el empleado "' || nombre_apellidos
 || '" con un sueldo mensual de "' || salario_mensual || '" es de "' || salario_anual || '"');
```

Bloque anónimo PL/SQL [completo]

Se va a mostrar todo el bloque PL/SQL indicando que hace cada línea:

```
--Necesario para poder imprimir por pantalla en PL/SQL
SET SERVEROUTPUT on;

DECLARE

    nombre_apellidos VARCHAR2(100) :='&Nombre_y_apellidos'; -- variable que almacena nombre_apellido

1   salario_mensual NUMBER :=&salario_mensual; -- variable que almacena el salario

    prima NUMBER; -- variable que almacena el plus del salario

    salario_anual NUMBER; -- Variable que almacena el salario anual que obtenemos como resultado final

BEGIN

2   IF salario_mensual * 12 > 21000 THEN  -- Si es mayor a  21000...
        prima := 3000;
3   ELSIF salario_mensual * 12 BETWEEN 12000 AND 21000 THEN -- Y si es entre 12000 y 21000...
        prima := 1800;
4   ELSE -- Sino ...
        prima := 1000;
    END IF;

5   salario_anual := (salario_mensual * 12) + prima;
6    DBMS_OUTPUT.PUT_LINE('El sueldo anual para el empleado "' || nombre_apellidos
     || '" con un sueldo mensual de "' || salario_mensual || '" es de "' || salario_anual || '"');

END;
```

✗ En primer lugar tenemos que activar el servicio para poder mostrar por pantalla

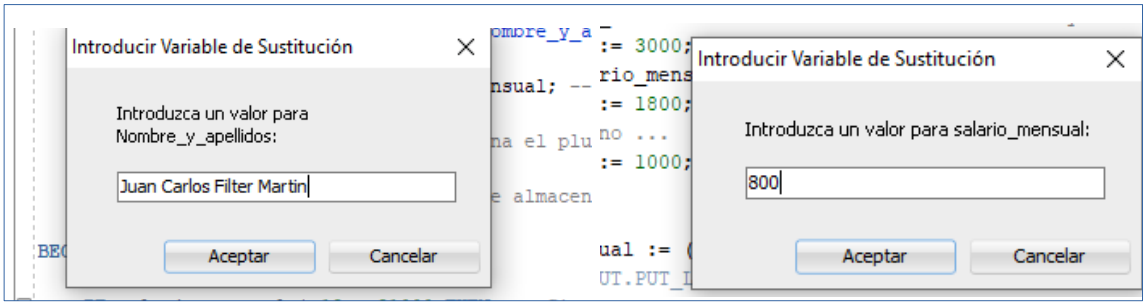
SET SERVEROUTPUT on

1. Declaramos las variables de sustitución y las variables necesarias ‘prima’ y ‘salario_anual’
2. Condición IF que entra si el salario mensual * 12 es mayor a 21000
3. Entra si el salario mensual * 12 está entre 12000 y 21000 (incluidos)
4. Por ultimo si nada de lo anterior se cumple entra en el ELSE que sería si es menor a 12000
5. Se almacena en salario anual el resultado de salario mensual*12 + prima correspondiente.
6. Mostrar por pantalla con la concatenación indicada.

Resultado por pantalla

Prueba 1:

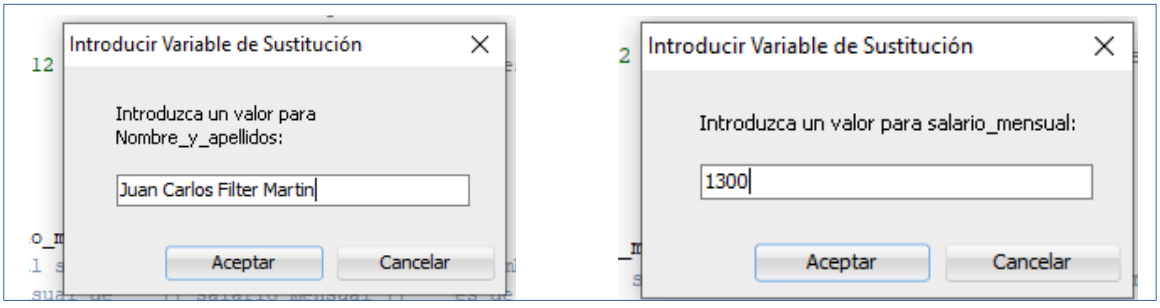
nombre_apellidos: Juan Carlos Filter Martin
salario_mensual: 800




```
END;
El sueldo anual para el empleado "Juan Carlos Filter Martin" con un sueldo mensual de "800" es de "10600"

Procedimiento PL/SQL terminado correctamente.
```

Prueba 2:

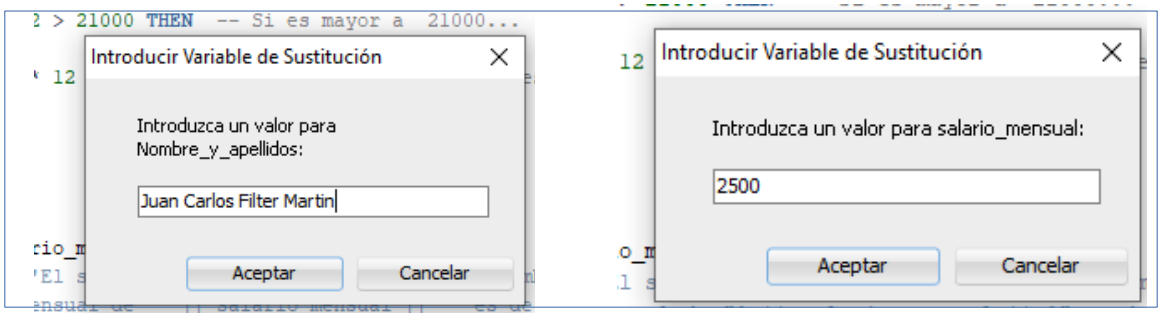
nombre_apellidos: Juan Carlos Filter Martin
salario_mensual: 1300




```
END;
El sueldo anual para el empleado "Juan Carlos Filter Martin" con un sueldo mensual de "1300" es de "17400"

Procedimiento PL/SQL terminado correctamente.
```

Prueba 3:

nombre_apellidos: Juan Carlos Filter Martin
salario_mensual: 2500




```
END;
El sueldo anual para el empleado "Juan Carlos Filter Martin" con un sueldo mensual de "2500" es de "33000"

Procedimiento PL/SQL terminado correctamente.
```

3. **(RA05_f)** Se han definido procedimientos y funciones de usuario y **(RA05_j)** Se han utilizado excepciones

Función PL/SQL

Se va a crear una función llamada 'getNombreApellidos' que va a consultar en la tabla 'Employees' del esquema HR para devolver nombre y apellido de un empleado pasándole el id de empleado.

- Función llamada getNombreApellidos

```
CREATE OR REPLACE FUNCTION getNombreApellidos(emplo
  nombre_apellidos VARCHAR2(100);
```

- Parámetro de entrada el id de empleado y devuelve como salida "Nombre Apellidos" es la concatenación de los campos 'first_name' y 'last_name'.

```
CREATE OR REPLACE FUNCTION getNombreApellidos(employee_id IN NUMBER) RETURN VARCHAR2 IS
  nombre_apellidos VARCHAR2(100);
```

Esta función PL/SQL hace lo siguiente:

- Realizamos una consulta a la tabla 'Employees' con el id del empleado pasado por parámetro y recogemos el nombre y apellido que se va a concatenar y guardar en la variable nombre_apellido

- Y va a ser devuelto mediante un return a nombre y apellidos que es la salida de la función

```
BEGIN
  SELECT first_name || ' ' || last_name INTO nombre_apellidos FROM Employees
  WHERE employee_id = getNombreApellidos.employee_id;

  RETURN nombre_apellidos;
```

- Si NO existe un empleado con dicho id *(WHEN NO_DATA_FOUND THEN)* ...

  va a devolver una excepción como cadena de texto
  "No existe un empleado con id "ID"

CONVERSION EXPLÍCITA A VARCHAR2

```
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN 'No existe un empleado con id ' || TO_CHAR(employee_id);
END;
```

- Si existe un empleado con el id entonces devolvemos la cadena nombre_apellido mediante el return y este va a ser enviado cuando se llame a la función

```
    WHERE employee_id = getNombre

    RETURN nombre_apellidos;
EXCEPTION
```

```
CREATE OR REPLACE FUNCTION getNombreApellidos(employee_id IN NUMBER) RETURN VARCHAR2 IS
  nombre_apellidos VARCHAR2(100);
```

Función PL/SQL [completo]

```
1  CREATE OR REPLACE FUNCTION getNombreApellidos(employee_id IN NUMBER) RETURN VARCHAR2 IS
     nombre_apellidos VARCHAR2(100);

   BEGIN
2    SELECT first_name || ' ' || last_name INTO nombre_apellidos FROM Employees
     WHERE employee_id = getNombreApellidos.employee_id;

3    RETURN nombre_apellidos;
4  EXCEPTION
     WHEN NO_DATA_FOUND THEN
       RETURN 'No existe un empleado con id ' || TO_CHAR(employee_id);
   END;
```

1. Función getNombreApellidos que recoge por parámetros el id y retorna la concatenación de nombre_apellido.
2. Consulta a la tabla Employees con el id del empleado pasado por parámetro (mediante el where se compara ambos id) y recogemos el nombre y apellido que se va a concatenar y guardar en la variable nombre_apellido.
3. Devolvemos la concatenación del nombre y apellido.
4. Si no existe el id mandará un mensaje de error con el texto indicado.

Crear Bloque PL/SQL para ejecutar la función getNombreApellidos

```
--Ejecución de la función getNombreApellidos

1  DECLARE
     resultado VARCHAR2(100); --Declaramos una variable que va a  obtener el resultado de la llamada a la función

   BEGIN
2  --Diferentes resultados con id: 105, 201 y 1111
     resultado := getNombreApellidos(105);
     DBMS_OUTPUT.PUT_LINE(resultado);

     resultado := getNombreApellidos(201);
     DBMS_OUTPUT.PUT_LINE(resultado);

     resultado := getNombreApellidos(1111);
     DBMS_OUTPUT.PUT_LINE(resultado);
   END;
```

1. Se declara una variable que va a obtener el resultado de la función
2. Se va a llamar a la función, indicándole por parámetros diferentes id y almacenándolo en resultado.

Ejecución de la función

1. Compilamos la función

```
CREATE OR REPLACE FUNCTION getNombreApellidos(employee_id IN NUMBER) RETURN VARCHAR2 IS
  nombre_apellidos VARCHAR2(100);

BEGIN
  SELECT first_name || ' ' || last_name INTO nombre_apellidos FROM Employees
  WHERE employee_id = getNombreApellidos.employee_id;

  RETURN nombre_apellidos;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN 'No existe un empleado con id ' || TO_CHAR(employee_id);
END;

--Ejecución de la función getNombreApellidos
```

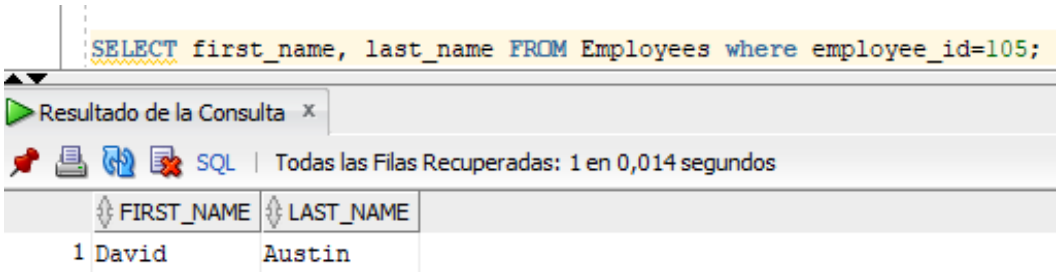
```
Salida de Script
Tarea terminada en 0,117 segundos

Function GETNOMBREAPELLIDOS compilado
```

2. Ejecutamos el bloque PL/SQL para llamar a la función con los id: 105, 201 y 1111

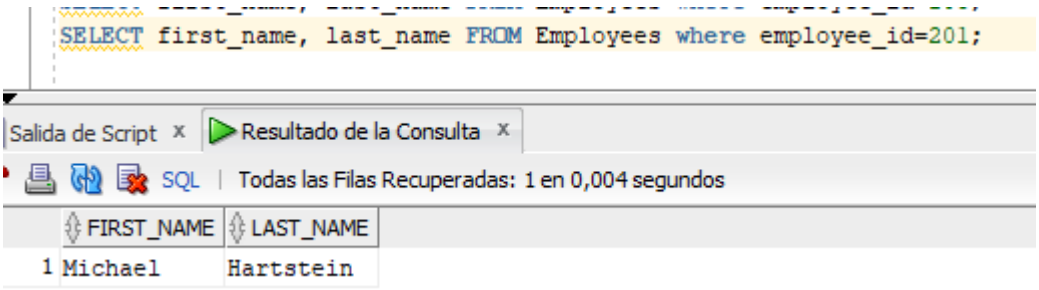
```
DECLARE
  resultado VARCHAR2(100); --Declaramos una variak

BEGIN
--Diferentes resultados con id: 105, 201 y 1111
  resultado := getNombreApellidos(105);
  DBMS_OUTPUT.PUT_LINE(resultado);

  resultado := getNombreApellidos(201);
  DBMS_OUTPUT.PUT_LINE(resultado);

  resultado := getNombreApellidos(1111);
  DBMS_OUTPUT.PUT_LINE(resultado);
END;
```

```
Salida de Script
Tarea terminada en 0,14 segundos

Function GETNOMBREAPELLIDOS compilado

David Austin
Michael Hartstein
No existe un empleado con id 1111

Procedimiento PL/SQL terminado correctamente.
```

El ID 105 : corresponde a David Austin



El ID 201 : corresponde a Michael Hartstien



El ID 1111 : No corresponde a ningún empleado