**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**

**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**

**DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN, CONTROL Y EVALUACIÓN DE RECURSOS INFORMÁTICOS**

**LIC. EN ING. DE SISTEMAS DE INFORMACIÓN**

**SISTEMAS DE BASE DE DATOS II**

**Laboratorio N°3**

**Profesor:**

**Ing. Henry Lezcano**

**Integrantes:**

Cutire, Fernando (8-972-906)

Díaz, Gabriel (20-53-5198)

Escobar, Jorge (2-747-1772)

Feng, William (8-977-446)

**Grupo: 1IF131**

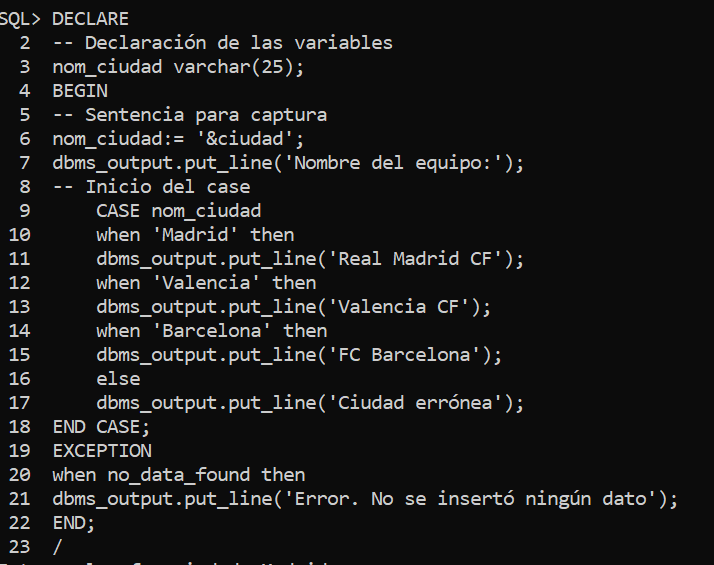
**Caso:**

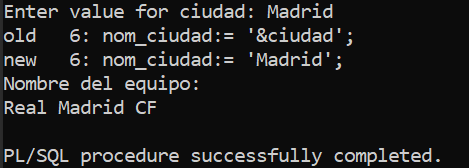
REALIZAR LAS IMPLEMENTACIONES DE LOS SIGUIENTES BLOQUES ANÓNIMOS. Trate de incluir la estructura integral de los bloques de ser necesario controlando las exception. Adicional aplique las guías de estilo de programación del PL/SQL. Debe crear las relaciones con las restricciones que sean necesarias para los programas solicitados de ser necesario.

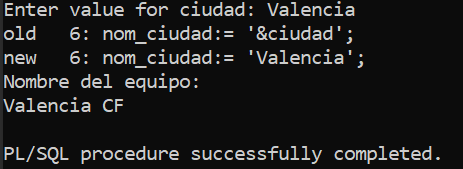
1. Desarrolle un bloque anónimo que capture (&captura) el nombre de una ciudad española y mande a línea de comando el nombre del equipo que representa la ciudad. El ejercicio será para 3 ciudades. Utilice la estructura del CASE por la estructura de control IF-THEN-ELSE. No se permiten las mismas ciudades por Equipo.

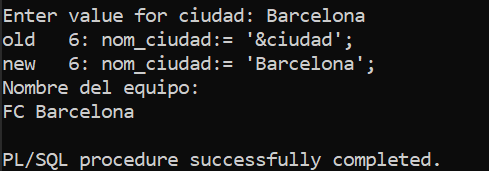
| DECLARE -- Declaración de las variables  nom\_ciudad varchar(25); BEGIN -- Sentencia para captura  nom\_ciudad:= '&ciudad'; dbms\_output.put\_line('Nombre del equipo:'); -- Inicio del case   CASE nom\_ciudad  when 'Madrid' then  dbms\_output.put\_line('Real Madrid CF');  when 'Valencia' then  dbms\_output.put\_line('Valencia CF');  when 'Barcelona' then  dbms\_output.put\_line('FC Barcelona');  else  dbms\_output.put\_line('Ciudad errónea'); END CASE; EXCEPTION when no\_data\_found then dbms\_output.put\_line('Error. No se insertó ningún dato'); END; / |
| --- |

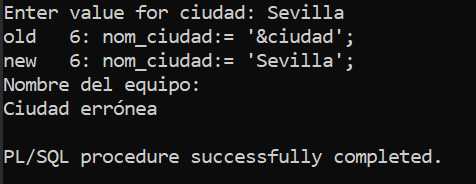
**Ejecución del programa**









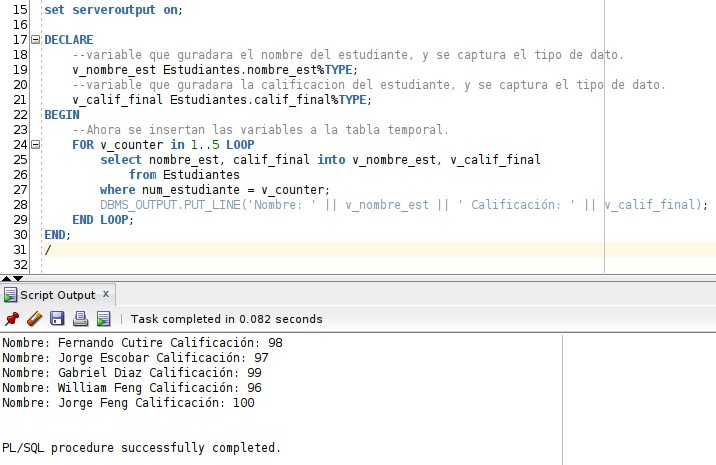


2. Desarrolle un bloque anónimo que cargue en una relación o tabla de base de datos llamada estudiante con el número de estudiante, cédula, nombre y calificación final. Luego que realice una consulta a esta tabla de estudiante para mostrar en la línea de comando el nombre del estudiante con la calificación final obtenida.

**Código del bloque anónimo:**

| create table Estudiantes ( num\_estudiante number(4) not null, ced\_est varchar2(12) not null, nombre\_est varchar(100) not null, calif\_final number(3) not null, constraint pk\_num\_est primary key (num\_estudiante) );  insert into Estudiantes values (0001, '8-950-100', 'Fernando Cutire', 98); insert into Estudiantes values (0002, '8-950-200', 'Jorge Escobar', 97); insert into Estudiantes values (0003, '8-950-300', 'Gabriel Diaz', 99); insert into Estudiantes values (0004, '8-950-400', 'William Feng', 96); insert into Estudiantes values (0005, '8-950-500', 'Jorge Feng', 100);  set serveroutput on;  DECLARE  --variable que guardará el nombre del estudiante, y se captura el tipo de dato.  v\_nombre\_est Estudiantes.nombre\_est%TYPE;  --variable que guardará la calificación del estudiante, y se captura el tipo de dato.  v\_calif\_final Estudiantes.calif\_final%TYPE; BEGIN  --Ahora se insertan las variables a la tabla temporal.  FOR v\_counter in 1..5 LOOP  select nombre\_est, calif\_final into v\_nombre\_est, v\_calif\_final  from Estudiantes  where num\_estudiante = v\_counter;  DBMS\_OUTPUT.PUT\_LINE('Nombre: ' || v\_nombre\_est || 'Calificación: ' || v\_calif\_final);  END LOOP; END; / |
| --- |

**Ejecución del programa:**

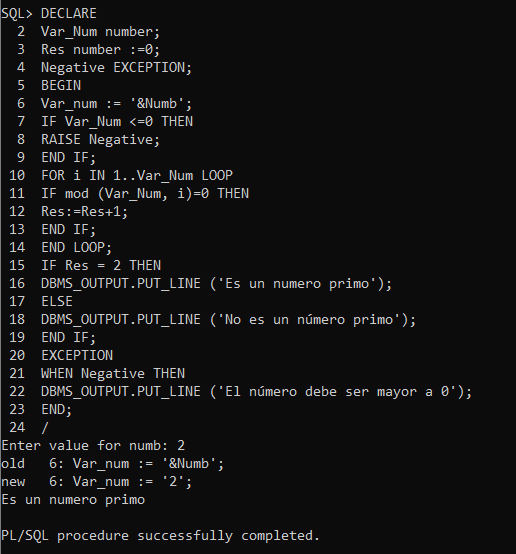


3. Desarrolle un bloque anónimo que capture un número entero y determine si este número es primo o no lo es, adicionalmente muestre el resultado en la línea de comando.

**CÓDIGO DEL BLOQUE ANÓNIMO**

| DECLARE  Var\_Num number;  Res number :=0;  Negative EXCEPTION; BEGIN  Var\_num := '&Numb';  IF Var\_Num <=0 THEN  RAISE Negative;  END IF;  FOR i IN 1..Var\_Num LOOP  IF mod (Var\_Num, i)=0 THEN  Res:=Res+1;  END IF;  END LOOP;  IF Res = 2 THEN  DBMS\_OUTPUT.PUT\_LINE ('Es un número primo');  ELSE  DBMS\_OUTPUT.PUT\_LINE ('No es un número primo');  END IF;  EXCEPTION  WHEN Negative THEN  DBMS\_OUTPUT.PUT\_LINE ('El número debe ser mayor a 0');  END;  / |
| --- |
|  |

**Ejecución de programa**

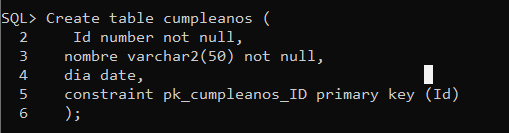
****

4. Desarrolle un bloque anónimo que implemente un proceso de repetición para almacenar en una relación de base de datos llamada cumpleaños la identificación que corresponde al contador que controla el ciclo de repetición, nombre y día de cumpleaños de 5 estudiantes de su grupo. Luego un bloque adicional que me permita capturar la identificación y haga una consulta a la relación cumpleaños para conocer el nombre y el día de cumpleaños en línea de comando.

| SQL> Create table cumpleanos (  Id number not null,  nombre varchar2(50) not null,  dia date,  constraint pk\_cumpleanos\_ID primary key (Id)  ); |
| --- |

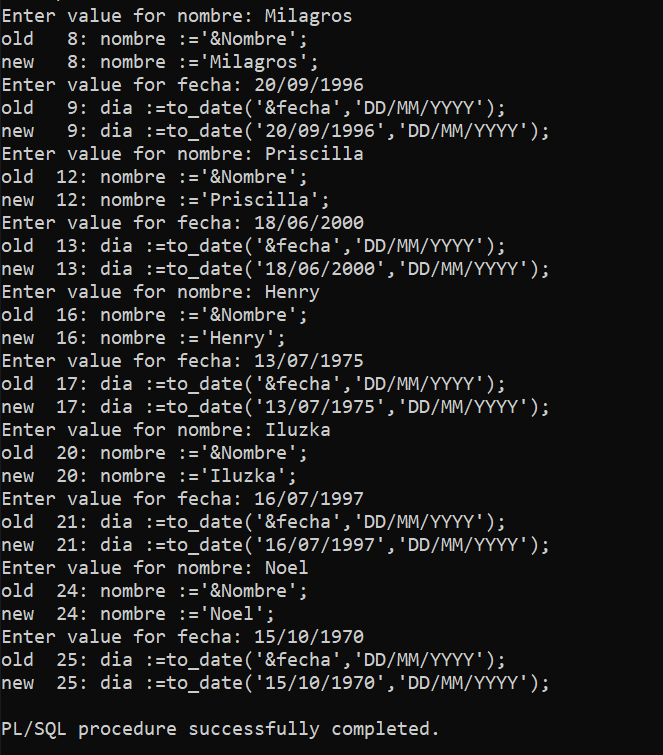
**CÓDIGO DEL PRIMER BLOQUE ANÓNIMO**

Ejecución del programa



Programa:

| Declare   IDC number;  nombre cumpleanos.nombre%TYPE;  dia cumpleanos.dia%TYPE; Begin  FOR IDC in 1..5 LOOP  IF IDC = 1 THEN  nombre :='&Nombre';  dia :=to\_date('&fecha','DD/MM/YYYY');  INSERT INTO cumpleanos VALUES(IDC,nombre,dia);  ELSIF IDC = 2 THEN  nombre :='&Nombre';  dia :=to\_date('&fecha','DD/MM/YYYY');  INSERT INTO cumpleanos VALUES(IDC,nombre,dia);  ELSIF IDC = 3 THEN  nombre :='&Nombre';  dia :=to\_date('&fecha','DD/MM/YYYY');  INSERT INTO cumpleanos VALUES(IDC,nombre,dia);  ELSIF IDC = 4 THEN  nombre :='&Nombre';  dia :=to\_date('&fecha','DD/MM/YYYY');  INSERT INTO cumpleanos VALUES(IDC,nombre,dia);  ELSIF IDC = 5 THEN  nombre :='&Nombre';  dia :=to\_date('&fecha','DD/MM/YYYY');  INSERT INTO cumpleanos VALUES(IDC,nombre,dia);  END IF;  END LOOP; END;  / |
| --- |



**CÓDIGO DEL SEGUNDO BLOQUE ANÓNIMO**

| Set serveroutput on; DECLARE  IDC cumpleanos.ID%TYPE;  nombre1 cumpleanos.nombre%TYPE;  dia1 cumpleanos.dia%TYPE; BEGIN  IDC := &IDC;  Select nombre,dia INTO nombre1, dia1  From cumpleanos  Where ID=IDC;  DBMS\_OUTPUT.PUT\_LINE('nombre: ' || nombre1 || 'Dia de nacimiento: ' || dia1); END; / |
| --- |

