

**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**  
**LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN**

**SISTEMAS DE BASES DE DATOS II**  
**LABORATORIO No. 4**  
**IMPLEMENTACIONES DE BLOQUES ANONIMOS**

**Integrantes:**

**ANDREINA GÓMEZ 8-939-1682**

**STEFANIE AROSEMENA 8-885-1747**

**EMANOL GONZALEZ 3-745-1637**

**JOSÉ QUINTERO 8-952-698**

**DOCENTE: ING. HENRY J. LEZCANO P.**

**GRUPO: 1IF121**

**FECHA: 30 SEPTIEMBRE.**

**AÑO 2020.**

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.

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
v_Ciudad varchar2(15) ;
```

```
BEGIN
```

```
v_Ciudad := '&v_Ciudad';
```

```
CASE
```

```
when v_Ciudad= 'Madrid' then
```

```
dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al  
"REAL MADRID"');
```

```
when v_Ciudad= 'Barcelona' then
```

```
dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al "El  
BARCELONA"');
```

```
when v_Ciudad= 'Valencia' then
```

```
dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al "EL  
VALENCIA FC"');
```

```
ELSE dbms_output.put_line ('no encontrado');
```

```
END CASE;
```

```
END;
```

```
/
```

## Evidencias

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_Ciudad varchar2(15) ;
  3  BEGIN
  4  v_Ciudad := '&v_Ciudad';
  5  CASE
  6  when v_Ciudad= 'Madrid' then
  7  dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al "REAL MADRID"');
  8  when v_Ciudad= 'Barcelona' then
  9  dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al "El BARCELONA"');
  10 when v_Ciudad= 'Valencia' then
  11 dbms_output.put_line ('La ciudad '||v_Ciudad ||' tiene como equipo al "EL VALENCIA FC"');
  12 ELSE dbms_output.put_line ('no encontrado');
  13 END CASE;
  14 END;
  15 /
Enter value for v_ciudad: Barcelona
old  4: v_Ciudad := '&v_Ciudad';
new  4: v_Ciudad := 'Barcelona';
La ciudad Barcelona tiene como equipo al "El BARCELONA"

PL/SQL procedure successfully completed.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_Ciudad varchar2(15) ;
  3  BEGIN
  4  v_Ciudad := '&v_Ciudad';
  5  CASE
  6  when v_Ciudad= 'Madrid' then
  7  dbms_output.put_line ('La ciudad '||v_Ciudad ||'tiene como equipo al "REAL MADRID"');
  8  when v_Ciudad= 'Barcelona' then
  9  dbms_output.put_line ('La ciudad '||v_Ciudad ||'tiene como equipo al "El BARCELONA"');
  10 when v_Ciudad= 'Valencia' then
  11 dbms_output.put_line ('La ciudad '||v_Ciudad ||'tiene como equipo al "EL VALENCIA FC"');
  12 ELSE dbms_output.put_line ('no encontrado');
  13 END CASE;
  14 END;
  15 /
Enter value for v_ciudad: Madrid
old  4: v_Ciudad := '&v_Ciudad';
new  4: v_Ciudad := 'Madrid';
La ciudad Madridtiene como equipo al "REAL MADRID"

PL/SQL procedure successfully completed.
```

## Como anotación

```
SQL> conn lab4
Enter password:
Connected.
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_Ciudad varchar2(15) ;
  3  BEGIN
  4  v_Ciudad := ||'|| &v_Ciudad ||'||;
  5  CASE
  6  when v_Ciudad = 'Madrid' then
  7  dbms_output.put_line ('Real Madrid');
  8  END CASE;
  9  END;
  10 /
Enter value for v_ciudad: madrid
old  4: v_Ciudad := ||'|| &v_Ciudad ||'||;
new  4: v_Ciudad := ||'|| madrid ||'||;
v_Ciudad := ||'|| madrid ||'||;
*
ERROR at line 4:
ORA-06550: line 4, column 13:
PLS-00103: Encountered the symbol "||" when expecting one of the following:
( - + case mod new not null <an identifier>
<a double-quoted delimited-identifier> <a bind variable>
continue avg count current exists max min prior sql stddev
sum variance execute forall merge time timestamp interval
date <a string literal with character set specification>
<a number> <a single-quoted SQL string> pipe
<an alternatively-quoted string literal with character set specification>
<an alternatively
```

No me salió a la primera

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_Ciudad varchar2(15) ;
  3  v_opcion varchar2(15) ;
  4
  5  BEGIN
  6  v_Ciudad := &v_Ciudad ;
  7  v_opcion := v_Ciudad;
  8  CASE v_opcion
  9  when 'Madrid' then
10  dbms_output.put_line ('Real Madrid');
11  when ' Barcelona ' then
12  dbms_output.put_line ('El Barcelona');
13  when 'Valencia' then
14  dbms_output.put_line ('El Valencia F');
15  END CASE;
16  END;
17  /
Enter value for v_ciudad: Madrid
old  6: v_Ciudad := &v_Ciudad ;
new  6: v_Ciudad := Madrid ;
v_Ciudad := Madrid ;
*
ERROR at line 6:
ORA-06550: line 6, column 14:
PLS-00201: identifier 'MADRID' must be declared
ORA-06550: line 6, column 1:
PL/SQL: Statement ignored
```

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_Ciudad varchar(10);
  3  BEGIN
  4  v_Ciudad := &v_Ciudad;
  5  CASE v_Ciudad
  6  when 'MADRID' then
  7  dbms_output.put_line('Real Madrid');
  8  when 'BARCELONA' then
  9  dbms_output.put_line('El Barcelona');
10  when 'VALENCIA' then
11  dbms_output.put_line('El Valencia F');
12  END CASE;
13  END;
14  /
Enter value for v_ciudad: Madrid
old  4: v_Ciudad := &v_Ciudad;
new  4: v_Ciudad := Madrid;
v_Ciudad := Madrid;
*
ERROR at line 4:
ORA-06550: line 4, column 13:
PLS-00201: identifier 'MADRID' must be declared
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

2. Desarrolle un bloque anónimo que cargue en una relación o tabla de base de datos llamada estudiante con el número de estudiante, cedula, 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.

```
SQL> create table estudiantes(  
  2  n_est number not null,  
  3  nombre_est varchar2(30) not null,  
  4  cedula_est varchar2(20) not null,  
  5  nota_final number(5) not null,  
  6  constraint pk_estudiantes_n_est primary key(n_est));  
  
Table created.  
  
SQL> insert into estudiantes values(101, 'Teresa', '8-9234-7898', 100);  
1 row created.  
  
SQL> insert into estudiantes values(102, 'Coraline', '8-5674-1002', 95);  
1 row created.  
  
SQL> insert into estudiantes values(103, 'Patrick', '4-8695-1234', 75);  
1 row created.  
  
SQL> insert into estudiantes values(104, 'Nathan', '9-7896-1723', 50);  
1 row created.  
  
SQL> insert into estudiantes values(105, 'Anne', '7-8975-1000', 61);  
1 row created.  
  
SQL> insert into estudiantes values(106, 'Gilbert', '8-670-7890', 100);  
1 row created.  
  
SQL> insert into estudiantes values(107, 'Diego', '8-7654-1892', 90);  
1 row created.  
  
SQL> commit;
```

```
SQL> Declare
  2   va_nota_final number(5);
  3   va_nombre_est varchar2(30);
  4   Begin
  5   select nota_final
  6 into va_nota_final
  7   from estudiantes
  8   where nota_final = 50;
  9   select nombre_est
10 into va_nombre_est
11   from estudiantes
12   where nombre_est= 'Nathan';
13   END;
14   /

PL/SQL procedure successfully completed.
```

```
SQL> Declare
  2   va_nota_final estudiantes.nota_final%TYPE;
  3   va_nombre_est estudiantes.nombre_est%TYPE;
  4   Begin
  5   Select
  6   nota_final,
  7   nombre_est
  8   into
  9   va_nota_final,
10   va_nombre_est
11   from estudiantes
12   where n_est = 101;
13   DBMS_OUTPUT.PUT_LINE(va_nombre_est || va_nota_final);
14   End;
15   /

PL/SQL procedure successfully completed.
```

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

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
v_numprimo number;
```

```
v_contador number := 0 ;
```

```
v_auxiliar number(10,0) := 0;
```

```
BEGIN
```

```
v_numprimo := & v_numprimo;
```

```
FOR i IN 1.. v_numprimo LOOP
```

```
v_auxiliar := v_numprimo/i;
```

```
v_auxiliar := v_auxiliar*i;
```

```
IF v_auxiliar =1 THEN
```

```
v_contador := v_contador+1;
```

```
ELSIF v_auxiliar=v_numprimo THEN
```

```
v_contador := v_contador+1;
```

```
ELSE NULL;
```

```
END IF;
```

```
END LOOP;
```

```
IF v_contador = 2 THEN
```

```
dbms_output.put_line ('El numero '|| v_numprimo||' es primo');
```

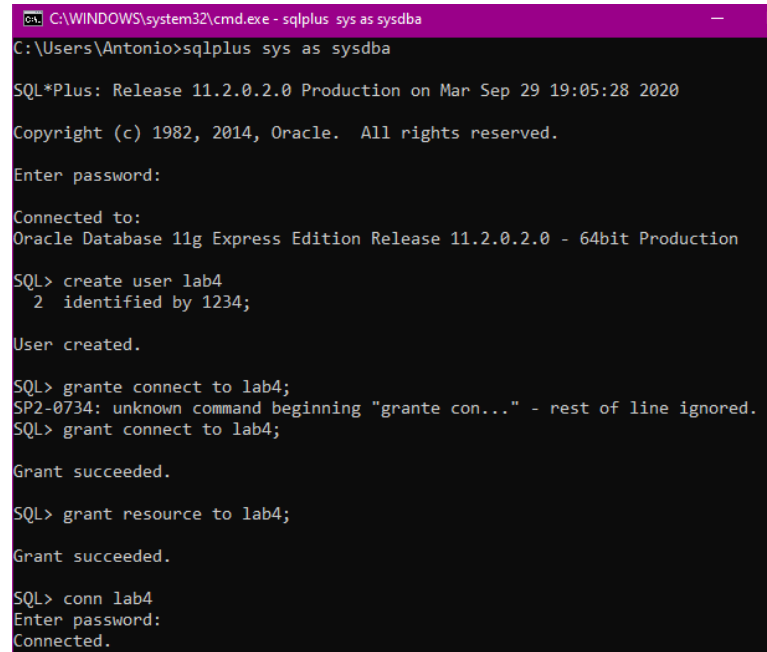
```
ELSE dbms_output.put_line ('El numero '|| v_numprimo||' no es primo');
```

END IF;

END;

/

## Evidencias



```
C:\WINDOWS\system32\cmd.exe - sqlplus sys as sysdba
C:\Users\Antonio>sqlplus sys as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Mon Sep 29 19:05:28 2020

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> create user lab4
      2 identified by 1234;

User created.

SQL> grante connect to lab4;
SP2-0734: unknown command beginning "grante con..." - rest of line ignored.
SQL> grant connect to lab4;

Grant succeeded.

SQL> grant resource to lab4;

Grant succeeded.

SQL> conn lab4
Enter password:
Connected.
```



```

C:\WINDOWS\system32\cmd.exe - sqlplus sys as sysdba

SQL> conn lab4
Enter password:
Connected.
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_numprimo number;
  3  v_contador number := 0 ;
  4  v_auxiliar number(10,0) := 0;
  5  BEGIN
  6  v_numprimo := & v_numprimo;
  7  FOR i IN 1.. v_numprimo LOOP
  8  v_auxiliar := v_numprimo/i;
  9  v_auxiliar := v_auxiliar*i;
 10  IF v_auxiliar =1 THEN
 11  v_contador := v_contador+1;
 12  ELSIF v_auxiliar=v_numprimo THEN
 13  v_contador := v_contador+1;
 14  ELSE NULL;
 15  END IF;
 16  END LOOP;
 17  IF v_contador = 2 THEN
 18  dbms_output.put_line ('El numero '|| v_numprimo||' es primo');
 19  ELSE dbms_output.put_line ('El numero '|| v_numprimo||' no es primo');
 20  END IF;
 21  END;
 22  /
Enter value for v_numprimo: 1
old 6: v_numprimo := & v_numprimo;
new 6: v_numprimo := 1;
El numero 1 no es primo

PL/SQL procedure successfully completed.

```

```

C:\WINDOWS\system32\cmd.exe - sqlplus sys as sysdba

PL/SQL procedure successfully completed.

SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  v_numprimo number;
  3  v_contador number := 0 ;
  4  v_auxiliar number(10,0) := 0;
  5  BEGIN
  6  v_numprimo := & v_numprimo;
  7  FOR i IN 1.. v_numprimo LOOP
  8  v_auxiliar := v_numprimo/i;
  9  v_auxiliar := v_auxiliar*i;
 10  IF v_auxiliar =1 THEN
 11  v_contador := v_contador+1;
 12  ELSIF v_auxiliar=v_numprimo THEN
 13  v_contador := v_contador+1;
 14  ELSE NULL;
 15  END IF;
 16  END LOOP;
 17  IF v_contador = 2 THEN
 18  dbms_output.put_line ('El numero '|| v_numprimo||' es primo');
 19  ELSE dbms_output.put_line ('El numero '|| v_numprimo||' no es primo');
 20  END IF;
 21  END;
 22  /
Enter value for v_numprimo: 7
old 6: v_numprimo := & v_numprimo;
new 6: v_numprimo := 7;
El numero 7 es primo

PL/SQL procedure successfully completed.

```

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.

Creación de Tabla Cumpleaneros	
<pre>create table cumpleaneros (   cod_contador number primary key,   nombre varchar (30) not null,   fecha_nac date not null);</pre>	
	<pre>SQL&gt; create table cumpleaneros (   2  cod_contador number primary key,   3  nombre varchar (30) not null,   4  fecha_nac date not null);  Table created.</pre>

Creación de Bloque Anónimo
<pre>DECLARE   va_cod_contador cumpleaneros.cod_contador%TYPE;   va_nombre cumpleaneros.nombre%TYPE;   va_fecha_nac cumpleaneros.fecha_nac%TYPE;   contador number := 0; BEGIN   for i in 1..5 loop   case i   when 1 then   va_cod_contador := i;   va_nombre := '&amp;nombre';   va_fecha_nac := to_date('&amp;fecha_nac','DD/MM/YYYY');   insert into cumpleaneros values(va_cod_contador, va_nombre,   va_fecha_nac);</pre>

```

when 2 then
    va_cod_contador := i;
    va_nombre := '&nombre';
    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
    insert into cumpleaneros values(va_cod_contador, va_nombre,
va_fecha_nac);
when 3 then
    va_cod_contador := i;
    va_nombre := '&nombre';
    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
    insert into cumpleaneros values(va_cod_contador, va_nombre,
va_fecha_nac);
when 4 then
    va_cod_contador := i;
    va_nombre := '&nombre';
    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
    insert into cumpleaneros values(va_cod_contador, va_nombre,
va_fecha_nac);
else
    va_cod_contador := i;
    va_nombre := '&nombre';
    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
    insert into cumpleaneros values(va_cod_contador, va_nombre,
va_fecha_nac);
end case;
end loop;
END;
/

```

```

SQL> DECLARE
2     va_cod_contador cumpleaneros.cod_contador%TYPE;
3     va_nombre cumpleaneros.nombre%TYPE;
4     va_fecha_nac cumpleaneros.fecha_nac%TYPE;
5     contador number := 0;
6     BEGIN
7     for i in 1..5 loop
8     case i
9     when 1 then
10    va_cod_contador := i;
11    va_nombre := '&nombre';
12    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
13    insert into cumpleaneros values(va_cod_contador, va_nombre, va_fecha_nac);
14    when 2 then
15    va_cod_contador := i;
16    va_nombre := '&nombre';
17    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
18    insert into cumpleaneros values(va_cod_contador, va_nombre, va_fecha_nac);
19    when 3 then
20    va_cod_contador := i;
21    va_nombre := '&nombre';
22    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
23    insert into cumpleaneros values(va_cod_contador, va_nombre, va_fecha_nac);
24    when 4 then
25    va_cod_contador := i;
26    va_nombre := '&nombre';
27    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
28    insert into cumpleaneros values(va_cod_contador, va_nombre, va_fecha_nac);
29    else
30    va_cod_contador := i;
31    va_nombre := '&nombre';
32    va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
33    insert into cumpleaneros values(va_cod_contador, va_nombre, va_fecha_nac);
34    end case;
35    end loop;
36    END;
37    /

```

### Inserción de Datos en el bucle

```

Enter value for nombre: Andreina
old 11:  va_nombre := '&nombre';
new 11:  va_nombre := 'Andreina';
Enter value for fecha_nac: 28/11/1999
old 12:  va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
new 12:  va_fecha_nac := to_date('28/11/1999','DD/MM/YYYY');
Enter value for nombre: Stefanie
old 16:  va_nombre := '&nombre';
new 16:  va_nombre := 'Stefanie';
Enter value for fecha_nac: 27/10/1999
old 17:  va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
new 17:  va_fecha_nac := to_date('27/10/1999','DD/MM/YYYY');
Enter value for nombre: Emanol
old 21:  va_nombre := '&nombre';
new 21:  va_nombre := 'Emanol';
Enter value for fecha_nac: 5/4/2000
old 22:  va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
new 22:  va_fecha_nac := to_date('5/4/2000','DD/MM/YYYY');
Enter value for nombre: Jose
old 26:  va_nombre := '&nombre';
new 26:  va_nombre := 'Jose';
Enter value for fecha_nac: 15/12/1999
old 27: va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
new 27: va_fecha_nac := to_date('15/12/1999','DD/MM/YYYY');
Enter value for nombre: Teresa
old 31:  va_nombre := '&nombre';
new 31:  va_nombre := 'Teresa';
Enter value for fecha_nac: 6/7/1999
old 32:  va_fecha_nac := to_date('&fecha_nac','DD/MM/YYYY');
new 32:  va_fecha_nac := to_date('6/7/1999','DD/MM/YYYY');

PL/SQL procedure successfully completed.

```

### Consulta

```
SQL> select* from cumpleaneros;
```

COD_CONTADOR	NOMBRE	FECHA_NA
1	Andreina	28/11/99
2	Stefanie	27/10/99
3	Emanol	05/04/00
4	Jose	15/12/99
5	Teresa	06/07/99