



**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS**  
**COMPUTACIONALES**



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

**LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN**  
**SISTEMAS DE BASES DE DATOS II.**

**ASIGNACIÓN 3**

**Profesor**  
**Henry Lezcano**

**Estudiantes**

Solis, Michael 8-958-1219  
Sánchez, Ana 8-967-832  
Tejada, Ángel 8-969-974  
Samudio, Nedith 8-968-1471  
Urriola, Vicente 8-892-2296

**Grupo**  
**1IF131**

**UNIVERSIDAD TECNOLÓGICA DE PANAMA  
FACULTAD DE INGENIERIA DE SISTEMAS  
COMPUTACIONALES LICENCIATURA EN INGENIERIA DE  
SISTEMAS DE INFORMACION**

**SISTEMAS DE BASES DE  
DATOS II**

**ASIGNACION No. 3**

REALIZAR LAS IMPLEMENTACIONES DE LOS SIGUIENTES BLOQUES ANONIMOS. Trate de incluir integral de los bloques de ser necesario controlando las exception. Adicional aplique las guías de estilo de programación 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 unidades. Utilice la estructura del CASE por la estructura IF-THEN-ELSE. No se permiten las mismas ciudades por equipo.

```
DECLARE
ciudad varchar2(30);
BEGIN
ciudad := '&ciudad';
dbms_output.put_line('Nombre del equipo de la ciudad:');
CASE ciudad
when 'Madrid' then dbms_output.put_line('Real Madrid');
when 'Valencia' then dbms_output.put_line('Valencia C.F. ');
when 'Barcelona' then dbms_output.put_line('Barcelona F.C');
else dbms_output.put_line('No se encontró la ciudad');
END CASE;
END;
/
```

```

SQL> DECLARE
  2  ciudad varchar2(30);
  3  BEGIN
  4  ciudad := '&ciudad';
  5  dbms_output.put_line('Nombre del equipo:');
  6      CASE ciudad
  7          when 'Madrid' then dbms_output.put_line('Real Madrid');
  8          when 'Valencia' then dbms_output.put_line('Valencia C.F. ');
  9          when 'Barcelona' then dbms_output.put_line('Barcelona');
 10          else dbms_output.put_line('No se encontró la ciudad');
 11      END CASE;
 12  END;
 13  /
Enter value for ciudad: Madrid
old  4: ciudad := '&ciudad';
new  4: ciudad := 'Madrid';
Nombre del equipo:
Real Madrid

PL/SQL procedure successfully completed.

```

```

SQL> DECLARE
  2  ciudad varchar2(30);
  3  BEGIN
  4  ciudad := '&ciudad';
  5  dbms_output.put_line('Nombre del equipo:');
  6      CASE ciudad
  7          when 'Madrid' then dbms_output.put_line('Real Madrid');
  8          when 'Valencia' then dbms_output.put_line('Valencia C.F. ');
  9          when 'Barcelona' then dbms_output.put_line('Barcelona');
 10          else dbms_output.put_line('No se encontró la ciudad');
 11      END CASE;
 12  END;
 13  /
Enter value for ciudad: Valencia
old  4: ciudad := '&ciudad';
new  4: ciudad := 'Valencia';
Nombre del equipo:
Valencia C.F.

PL/SQL procedure successfully completed.

```

```

SQL> DECLARE
  2  ciudad varchar2(30);
  3  BEGIN
  4  ciudad := '&ciudad';
  5  dbms_output.put_line('Nombre del equipo:');
  6      CASE ciudad
  7      when 'Madrid' then dbms_output.put_line('Real Madrid');
  8      when 'Valencia' then dbms_output.put_line('Valencia C.F. ');
  9      when 'Barcelona' then dbms_output.put_line('Barcelona F.C ');
 10      else dbms_output.put_line('No se encontró la ciudad');
 11      END CASE;
 12  END;
 13  /
Enter value for ciudad: Barcelona
old  4: ciudad := '&ciudad';
new  4: ciudad := 'Barcelona';
Nombre del equipo:
Barcelona F.C

PL/SQL procedure successfully completed.

```

```

SQL> DECLARE
  2  ciudad varchar2(30);
  3  BEGIN
  4  ciudad := '&ciudad';
  5  dbms_output.put_line('Nombre del equipo:');
  6      CASE ciudad
  7      when 'Madrid' then dbms_output.put_line('Real Madrid');
  8      when 'Valencia' then dbms_output.put_line('Valencia C.F. ');
  9      when 'Barcelona' then dbms_output.put_line('Barcelona F.C ');
 10      else dbms_output.put_line('No se encontró la ciudad');
 11      END CASE;
 12  END;
 13  /
Enter value for ciudad: Barcelona
old  4: ciudad := '&ciudad';
new  4: ciudad := 'Barcelona';
Nombre del equipo:
Barcelona F.C

PL/SQL procedure successfully completed.

```

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.

/\*Creación de la tabla Estudiante\*/

Create table Estudiante(

```
n_estudiante NUMBER p-rimary key,  
cedula VARCHAR2(20),  
nombre VARCHAR2(20),  
calificacion VARCHAR2(20)  
);
```

```
SQL> Create table Estudiante(  
2  n_estudiante NUMBER primary key,  
3  cedula VARCHAR2(20),  
4  nombre VARCHAR2(20),  
5  calificacion VARCHAR2(20)  
6  );  
  
Table created.
```

/\*Inserción de los datos\*/

```
Insert into Estudiante values (16, '8-967-832', 'Ana Sánchez' , 91);
```

```
SQL> Insert into Estudiante values (16, '8-967-832', 'Ana Sánchez' , 91);  
  
1 row created.
```

/\*Lectura de la tabla Estudiante\*/

DECLARE

```
v_nombre estudiante.nombre%type;
```

```
v_calif estudiante.calificacion%type;
```

BEGIN

```
Select nombre, calificacion into v_nombre, v_calif From Estudiante;
```

```
DBMS_OUTPUT.put_line('Nombre: '||v_nombre || '||'Calificacion: ' || v_calif);
```

END;

```
SQL> DECLARE  
2  v_nombre estudiante.nombre%type;  
3  v_calif estudiante.calificacion%type;  
4  BEGIN  
5  Select nombre, calificacion into v_nombre, v_calif From Estudiante;  
6  DBMS_OUTPUT.put_line('Nombre: '||v_nombre || '||'Calificacion: ' || v_calif);  
7  END;  
8  
9  /  
Nombre: Ana Sánchez Calificacion: 91  
  
PL/SQL procedure successfully completed.
```

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.

```
DECLARE
numero number;
divisores number;
BEGIN
numero := '&numero';
divisores := 0;
for i in 1..numero loop
    if mod(numero,i) = 0 then divisores := divisores +1;
    end if;
end loop;
if divisores = 2 then
    dbms_output.put_line('Es primo');
else dbms_output.put_line('No es primo');
end if;
END;
/
```

```
SQL> DECLARE
  2  numero number;
  3  divisores number;
  4  BEGIN
  5  numero := '&numero';
  6  divisores := 0;
  7  for i in 1..numero loop
  8  if mod(numero,i) = 0 then divisores := divisores +1;
  9  end if;
 10  end loop;
 11  if divisores = 2 then
 12  dbms_output.put_line('Es primo');
 13  else dbms_output.put_line('No es primo');
 14  end if;
 15  END;
 16  /
Enter value for numero: 10
old  5: numero := '&numero';
new  5: numero := '10';
No es primo

PL/SQL procedure successfully completed.
```

```

SQL> DECLARE
  2  numero number;
  3  divisores number;
  4  BEGIN
  5  numero := '&numero';
  6  divisores := 0;
  7  for i in 1..numero loop
  8  if mod(numero,i) = 0 then divisores := divisores +1;
  9  end if;
 10  end loop;
 11  if divisores = 2 then
 12  dbms_output.put_line('Es primo');
 13  else dbms_output.put_line('No es primo');
 14  end if;
 15  END;
 16  /
Enter value for numero: 7
old   5: numero := '&numero';
new   5: 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.

```

CREATE TABLE CUMPLEANIOS(
  id_est number NOT NULL,
  nombre_est varchar2(30) NOT NULL,
  cumpleaños_est date
);

```

```

SQL> CREATE TABLE CUMPLEANIOS(
  2      id_est number NOT NULL,
  3      nombre_est varchar2(30) NOT NULL,
  4      cumpleaños_est date
  5  );
Table created.

```

```

DECLARE
  v_id_est NUMBER;
  v_nombre_est CUMPLEANIOS.nombre_est%TYPE;
  v_cumpleanio_est CUMPLEANIOS.cumpleanio_est%TYPE;
BEGIN
  FOR v_id_est in 1..5 LOOP

```

```

    IF v_id_est = 1 THEN
        v_nombre_est:='&nombre ';
        v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
        INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est,
v_cumpleanio_est);
    ELSIF v_id_est = 2 THEN
        v_nombre_est:='&nombre ';
        v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
        INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est,
v_cumpleanio_est);
    ELSIF v_id_est = 3 THEN
        v_nombre_est:='&nombre ';
        v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
        INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est,
v_cumpleanio_est);
    ELSIF v_id_est = 4 THEN
        v_nombre_est:='&nombre ';
        v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
        INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est,
v_cumpleanio_est);
    ELSIF v_id_est = 5 THEN
        v_nombre_est:='&nombre ';
        v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
        INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est,
v_cumpleanio_est);
    END IF;
END LOOP;
END;
/

```



```

SQL> DECLARE
2     v_id_est NUMBER;
3     v_nombre_est CUMPLEANIOS.nombre_est%TYPE;
4     v_cumpleanio_est CUMPLEANIOS.cumpleanio_est%TYPE;
5 BEGIN
6     FOR v_id_est in 1..5 LOOP
7         IF v_id_est = 1 THEN
8             v_nombre_est:='&nombre ';
9             v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
10            INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est, v_cumpleanio_est);
11        ELSIF v_id_est = 2 THEN
12            v_nombre_est:='&nombre ';
13            v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
14            INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est, v_cumpleanio_est);
15        ELSIF v_id_est = 3 THEN
16            v_nombre_est:='&nombre ';
17            v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
18            INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est, v_cumpleanio_est);
19        ELSIF v_id_est = 4 THEN
20            v_nombre_est:='&nombre ';
21            v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
22            INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est, v_cumpleanio_est);
23        ELSIF v_id_est = 5 THEN
24            v_nombre_est:='&nombre ';
25            v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
26            INSERT INTO CUMPLEANIOS VALUES (v_id_est, v_nombre_est, v_cumpleanio_est);
27        END IF;
28    END LOOP;
29 END;
30 /
Enter value for nombre: Rosa
old 8:          v_nombre_est:='&nombre ';
new 8:          v_nombre_est:='Rosa ';
Enter value for fecha: 15/10/2000
old 9:          v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
new 9:          v_cumpleanio_est:= to_date('15/10/2000', 'DD/MM/YYYY');
Enter value for nombre: Carlos
old 12:         v_nombre_est:='&nombre ';
new 12:         v_nombre_est:='Carlos ';
Enter value for fecha: 06/12/1999
old 13:         v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
new 13:         v_cumpleanio_est:= to_date('06/12/1999', 'DD/MM/YYYY');
Enter value for nombre: Manuel

```

```

Enter value for nombre: Manuel
old 16:          v_nombre_est:='&nombre ';
new 16:          v_nombre_est:='Manuel ';
Enter value for fecha: 25/08/2001
old 17:          v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
new 17:          v_cumpleanio_est:= to_date('25/08/2001', 'DD/MM/YYYY');
Enter value for nombre: Isabel
old 20:          v_nombre_est:='&nombre ';
new 20:          v_nombre_est:='Isabel ';
Enter value for fecha: 18/02/2001
old 21:          v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
new 21:          v_cumpleanio_est:= to_date('18/02/2001', 'DD/MM/YYYY');
Enter value for nombre: Daniel
old 24:          v_nombre_est:='&nombre ';
new 24:          v_nombre_est:='Daniel ';
Enter value for fecha: 09/11/1998
old 25:          v_cumpleanio_est:= to_date('&fecha', 'DD/MM/YYYY');
new 25:          v_cumpleanio_est:= to_date('09/11/1998', 'DD/MM/YYYY');

PL/SQL procedure successfully completed.

SQL> █

```

```

DECLARE
    v_id_est CUMPLEANIOS.id_est%TYPE;
    v_est_nombre CUMPLEANIOS.nombre_est%TYPE;
    v_est_fecha CUMPLEANIOS.cumpleanio_est%TYPE;
BEGIN
    v_id_est:='&identificacion';
    SELECT nombre_est, cumpleanio_est INTO v_est_nombre, v_est_fecha
FROM CUMPLEANIOS
    WHERE id_est=v_id_est;
    DBMS_OUTPUT.PUT_LINE('NOMBRE:'|| ' '||v_est_nombre||' '||'FECHA:'||
' '||v_est_fecha);
    EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Este número de identificación de no existe');
END;
/

```

```

SQL> DECLARE
2     v_id_est CUMPLEANIOS.id_est%TYPE;
3     v_est_nombre CUMPLEANIOS.nombre_est%TYPE;
4     v_est_fecha CUMPLEANIOS.cumpleanio_est%TYPE;
5 BEGIN
6     v_id_est:='&identificacion';
7     SELECT nombre_est, cumpleanio_est INTO v_est_nombre, v_est_fecha FROM CUMPLEANIOS
8     WHERE id_est=v_id_est;
9     DBMS_OUTPUT.PUT_LINE('NOMBRE:'||' '||v_est_nombre||' '||'FECHA:'||' '||v_est_fecha);
10    EXCEPTION
11    WHEN NO_DATA_FOUND THEN
12        DBMS_OUTPUT.PUT_LINE('Este número de identificación de no existe');
13 END;
14 /
Enter value for identificacion: 3
old 6:      v_id_est:='&identificacion';
new 6:      v_id_est:='3';
NOMBRE: Manuel  FECHA: 25/08/01

PL/SQL procedure successfully completed.

```

La asignación No.3 es en Equipo de trabajo. Deben ir imprimiendo los procesos que van creando como prueba de resultado, pegarlo en una página de Word y Subirlo a Ecampus. La presentación de cada equipo será la próxima clase.