

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONES
LICENCIATURA EN DESARROLLO DE SOFTWARE
BASE DE DATOS II
LABORATORIO NO.2

Integrantes:

Victor Rodriguez

Eladio Gonzalez

Diego Navarro

Alejandro Mosquera

1. Desarrolle un bloque anónimo PL/SQL para la captura de la data de la Base de Datos del Laboratorio No.1

```
2. -- 1
DECLARE
    --para la tabla Generos
    v_generos_nombre Generos.genero%TYPE;
    v_generos_id Generos.id_genero%TYPE NOT NULL := 1;

    --para la tabla Peliculas
    v_peliculas_id Peliculas.id_pelicula%TYPE NOT NULL := 1;
    v_peliculas_nombre Peliculas.pelicula%TYPE;

    --para la tabla actores
    v_actores_nombre Actores.actor%TYPE;
    v_actores_id Actores.id_actor%TYPE NOT NULL := 1;

    --para la tabla peliculas_actores
    v_p_a_id_actor Peliculas_Actores.peliact_id_actor%TYPE;
    v_p_a_id_pelicula Peliculas_Actores.peliact_id_pelicula%TYPE;

    v_variablecontrol INT := 1;

BEGIN
    -- Insercion de generos:
    v_generos_nombre := 'Ficcion';
    INSERT INTO Generos (id_genero, genero) VALUES (v_generos_id,
```

```

v_generos_nombre);

    v_generos_id := v_generos_id + 1;
    v_generos_nombre := 'Romance';
    INSERT INTO Generos (id_genero, genero) VALUES (v_generos_id,
v_generos_nombre);

--Insercion de peliculas
v_generos_id := 1;
v_peliculas_nombre := 'Volver al futuro';
v_peliculas_id := 1234;
INSERT INTO Peliculas (pelicula, id_pelicula, pel_id_genero)
VALUES (v_peliculas_nombre, v_peliculas_id, v_generos_id);

v_generos_id := 2;
v_peliculas_nombre := 'La boda de mi mejor amigo';
v_peliculas_id := 2345;
INSERT INTO Peliculas (pelicula, id_pelicula, pel_id_genero)
VALUES (v_peliculas_nombre, v_peliculas_id, v_generos_id);

v_generos_id := 1;
v_peliculas_nombre := 'Hombres de negro 2';
v_peliculas_id := 4507;
INSERT INTO Peliculas (pelicula, id_pelicula, pel_id_genero)
VALUES (v_peliculas_nombre, v_peliculas_id, v_generos_id);

--Insercion de actores
v_actores_nombre := 'Michael J Fox';
v_actores_id := 1;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

v_actores_nombre := 'Christopher Lee Thompson';
v_actores_id := 2;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

v_actores_nombre := 'Julia Roberts';
v_actores_id := 3;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

v_actores_nombre := 'Dermont Mulroney';
v_actores_id := 4;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

v_actores_nombre := 'Cameron Diaz';
v_actores_id := 5;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

v_actores_nombre := 'Tomy Lee Jones';
v_actores_id := 6;
INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

```

```

        v_actores_nombre := 'Will Smith';
        v_actores_id := 7;
        INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

        v_actores_nombre := 'Rip Torn';
        v_actores_id := 8;
        INSERT INTO Actores (id_actor, actor) VALUES (v_actores_id,
v_actores_nombre);

        --Insercion de peliculas_actores
        v_p_a_id_actor := 1;
        v_p_a_id_pelicula := 1234;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 2;
        v_p_a_id_pelicula := 1234;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 3;
        v_p_a_id_pelicula := 2345;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 4;
        v_p_a_id_pelicula := 2345;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 5;
        v_p_a_id_pelicula := 2345;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 6;
        v_p_a_id_pelicula := 4507;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 7;
        v_p_a_id_pelicula := 4507;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

        v_p_a_id_actor := 8;
        v_p_a_id_pelicula := 4507;
        INSERT INTO peliculas_actores (peliact_id_actor,
peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);

    EXCEPTIONS

    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('ERROR INESPERADO');

```

```
END;
```

```
terminal  bash x + v
92  INSERT INTO peliculas_actores (peliact_id_actor, peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);
93
94  v_p_a_id_actor := 5;
95  v_p_a_id_pelicula := 2345;
96  INSERT INTO peliculas_actores (peliact_id_actor, peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);
97
98  v_p_a_id_actor := 6;
99  v_p_a_id_pelicula := 4507;
100 INSERT INTO peliculas_actores (peliact_id_actor, peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);
101
102 v_p_a_id_actor := 7;
103 v_p_a_id_pelicula := 4507;
104 INSERT INTO peliculas_actores (peliact_id_actor, peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);
105
106 v_p_a_id_actor := 8;
107 v_p_a_id_pelicula := 4507;
108 INSERT INTO peliculas_actores (peliact_id_actor, peliact_id_pelicula) VALUES (v_p_a_id_actor, v_p_a_id_pelicula);
109
110 EXCEPTION
111 WHEN OTHERS THEN
112     DBMS_OUTPUT.PUT_LINE('ERROR INESPERADO');
113 END;
114 /
v_generos_nombre Generos.genero%TYPE;
*
```

2. 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.

```
--2
DECLARE
    ciudad NVARCHAR2(15) := 'Valencia';
    equipo NVARCHAR2(20);
    control NUMBER := 0;
BEGIN
    WHILE control < 3 LOOP

        CASE ciudad
            WHEN 'Madrid' THEN
                equipo := 'Real Madrid';
            WHEN 'Barcelona' THEN
                equipo := 'F.C. Barcelona';
            WHEN 'Valencia' THEN
                equipo := 'O porto furbo clu';
            ELSE
                equipo := 'Ciudad no registrada';
            END CASE;

        control := control + 1;
    END LOOP;
END;
```

```

        DBMS_OUTPUT.PUT_LINE('El equipo de ' || ciudad || ' es
' || equipo);
        control := control + 1;
    END LOOP;
END;
```

3. 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.

```

-- 3
DECLARE
    p_no_estudiante Estudiante.no_estudiante%TYPE := 123;
    p_cedula Estudiante.cedula%TYPE := '1234567890';
    p_nombre Estudiante.nombre%TYPE := 'Nombre del Estudiante';
    p_calificacion_final Estudiante.calificacionFinal%TYPE := 95;
BEGIN
    INSERT INTO Estudiante
    (no_estudiante, cedula, nombre, calificacionFinal)
    VALUES
    (p_no_estudiante, p_cedula, p_nombre, p_calificacion_final);

    SELECT nombre, calificacionFinal INTO p_nombre, p_calificacion_final
    FROM Estudiante
    WHERE no_estudiante = 123;
    DBMS_OUTPUT.PUT_LINE(p_nombre || ' ' || p_calificacion_final);
END;
/
```

4. 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.

```

DECLARE
    v_num INT;
    v_es_primo BOOLEAN := TRUE;
    v_divisor BINARY_INTEGER := 2;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Ingrese un número:');
    v_num := &num;

    IF v_num <= 1 THEN
        v_es_primo := FALSE;
    ELSE
        WHILE v_divisor * v_divisor <= v_num LOOP
            IF MOD(v_num, v_divisor) = 0 THEN
                v_es_primo := FALSE;
                EXIT;
            END IF;
        END LOOP;
    END IF;
```

```

        v_divisor := v_divisor + 1;
    END LOOP;
END IF;

IF v_es_primo THEN
    DBMS_OUTPUT.PUT_LINE(v_num || ' es un número primo.');
```

```

ELSE
    DBMS_OUTPUT.PUT_LINE(v_num || ' no es un número primo.');
```

```

END IF;
END;
/
```

```

Terminal  bash x + v
11      ELSE
12          WHILE v_divisor * v_divisor <= v_num LOOP
13              IF MOD(v_num, v_divisor) = 0 THEN
14                  v_es_primo := FALSE;
15                  EXIT;
16              END IF;
17              v_divisor := v_divisor + 1;
18          END LOOP;
19      END IF;
20
21      IF v_es_primo THEN
22          DBMS_OUTPUT.PUT_LINE(v_num || ' es un número primo.');
```

```

23      ELSE
24          DBMS_OUTPUT.PUT_LINE(v_num || ' no es un número primo.');
```

```

25      END IF;
26  END;
27  /
Enter value for num: 7
old 7:      v_num := &num;
new 7:      v_num := 7;
Ingrese un número:
7 es un número primo.

PL/SQL procedure successfully completed.

SQL> 
```

5. 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

```
SET SERVEROUTPUT ON
DECLARE
    v_controlador Cumpleanos.id_cumpleano%TYPE :=1;
    v_nombre Cumpleanos.nombre%TYPE;
    v_fecha Cumpleanos.fecha_cumpleanos%TYPE;
    v_id Cumpleanos.id_cumpleano%TYPE;
    v_nombre_consulta Cumpleanos.nombre%TYPE;
    v_fecha_consulta Cumpleanos.fecha_cumpleanos%TYPE;
BEGIN
    WHILE v_controlador <= 5 LOOP
        DBMS_OUTPUT.PUT('Ingrese el nombre: ');
        v_nombre := '&nombre';
        DBMS_OUTPUT.PUT('Ingrese la fecha (YYYY/MM/DD): ');
        v_fecha := TO_DATE('&&fecha', 'YYYY/MM/DD');
        INSERT INTO Cumpleanos (id_cumpleano, nombre,
fecha_cumpleanos)
        VALUES (v_controlador, v_nombre,v_fecha);

        DBMS_OUTPUT.PUT_LINE('id con valor de ---> ' ||
v_controlador);
        v_controlador := v_controlador + 1;
    END LOOP;

    DBMS_OUTPUT.PUT('Ingrese un ID para consultar: ');
    v_id := &id;
    SELECT nombre, fecha_cumpleanos INTO v_nombre_consulta,
v_fecha_consulta
    FROM Cumpleanos
    WHERE id_cumpleano = v_id;
    DBMS_OUTPUT.PUT_LINE('Nombre encontrado: ' || v_nombre_consulta);
    DBMS_OUTPUT.PUT_LINE('Fecha de cumpleaños encontrada: ' ||
TO_CHAR(v_fecha_consulta, 'YYYY/MM/DD'));
END;
/
```

```
Terminal  bash x + v
22  v_id := &id;
23  SELECT nombre, fecha_cumpleanos INTO v_nombre_consulta, v_fecha_consulta
24  FROM Cumpleanos
25  WHERE id_cumpleano = v_id;
26  DBMS_OUTPUT.PUT_LINE('Nombre encontrado: ' || v_nombre_consulta);
27  DBMS_OUTPUT.PUT_LINE('Fecha de cumpleaños encontrada: ' || TO_CHAR(v_fecha_consulta, 'YYYY/MM/DD'));
28  END;
29  /
old 11:          v_nombre := '&nombre';
new 11:          v_nombre := 'alejandro';
old 13:          v_fecha := TO_DATE('&fecha', 'YYYY/MM/DD');
new 13:          v_fecha := TO_DATE('2000-08-09', 'YYYY/MM/DD');
Enter value for id: 1
old 22:  v_id := &id;
new 22:  v_id := 1;
Ingrese el nombre: Ingrese la fecha (YYYY/MM/DD): id con valor de ---> 1
Ingrese el nombre: Ingrese la fecha (YYYY/MM/DD): id con valor de ---> 2
Ingrese el nombre: Ingrese la fecha (YYYY/MM/DD): id con valor de ---> 3
Ingrese el nombre: Ingrese la fecha (YYYY/MM/DD): id con valor de ---> 4
Ingrese el nombre: Ingrese la fecha (YYYY/MM/DD): id con valor de ---> 5
Ingrese un ID para consultar: Nombre encontrado: alejandro
Fecha de cumpleaños encontrada: 2000/08/09

PL/SQL procedure successfully completed.

SQL> 
```

Untitled > src > Script.sql VCS Initialization 216:1 (1209 chars, 30 line b