

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

SISTEMAS DE BASES DE DATOS II

ASIGNACIÓN No. 2

CURSORES

NOMBRE: ANDREINA GÓMEZ.

CÉDULA: 8-939-1682.

DOCENTE: ING. HENRY J. LEZCANO P.

05 DE OCTUBRE.

II SEMESTRE 2020.

1. Implemente el bloque pl/sql de la ppt No.4, donde en el área de ejecución del bloque, la información extraída por el cursor sea cargada en una relación o tabla diseñada por usted, establezca controles para el proceso la inserción de la información.

Creación de tablas
Create table estudiantes (Id_est number primary key, Nombre varchar2(30), Apellido varchar2(30), Licenciatura varchar2(100)); Create table Est_sistemasdeinfo(V_id_est number primary key, V_nombre varchar2(30), V_apellido varchar2(30));

Insertando datos a la Tabla Estudiantes
Insert into Estudiantes values (101, 'Tom', 'Hopper', 'Mecánica'); Insert into Estudiantes values (102, 'David', 'Castañeda', 'Mecánica'); Insert into Estudiantes values (103, 'Emmy', 'Raver', 'Logística y Cadena de suministro'); Insert into Estudiantes values (104, 'Robert', 'Sheehan', 'Logística y Cadena de suministro'); Insert into Estudiantes values (105, 'Aidan', 'Gallagher', 'Sistemas de Información'); Insert into Estudiantes values (106, 'Justin', 'Min', 'Mecánica'); Insert into Estudiantes values (107, 'Elle', 'Page', 'Sistemas de Información'); Insert into Estudiantes values (108, 'Eudora', 'Patch', 'Sistemas de Información'); Insert into Estudiantes values (109, 'Harlan', 'Cooper', 'Sistemas de Información'); Insert into Estudiantes values (110, 'Archie', 'Andrews', 'Mecánica'); Insert into Estudiantes values (111, 'Betty', 'Cooper', 'Logística y Cadena de suministro'); Insert into Estudiantes values (112, 'Veronica', 'Lodge', 'Logística y Cadena de suministro'); Insert into Estudiantes values (113, 'Jughead', 'Jones', 'Sistemas de Información'); Insert into Estudiantes values (114, 'Cheryl', 'Blossom', 'Logística y Cadena de suministro'); Insert into Estudiantes values (115, 'Toni', 'Topaz', 'Mecánica');

**Bloque pl/sql (Donde este guarda los estudiantes que pertenecen a la
Licenciatura de Sistemas de Información en la Tabla
Est_sistemasdeinfo.).**

```
DECLARE
    va_id_est      estudiantes.id_est%TYPE;
    va_nombre      estudiantes.nombre%TYPE;
    va_apellido     estudiantes.apellido%TYPE;
    va_licenciatura estudiantes.licenciatura%TYPE := 'Sistemas de Información';

    CURSOR c_alumno IS
    SELECT id_est, nombre, apellido
    FROM estudiantes
    WHERE licenciatura = va_licenciatura;

    BEGIN
    OPEN c_alumno;
    LOOP
    FETCH c_alumno INTO va_id_est, va_nombre, va_apellido;
    EXIT WHEN c_alumno%NOTFOUND;

    INSERT INTO Est_sistemasdeinfo (v_id_est, v_nombre, v_apellido)
    VALUES (va_id_est, va_nombre, va_apellido);

    END LOOP;
    CLOSE c_alumno;
    COMMIT;
    END;
    /
```

EVIDENCIAS

Datos de la Tabla Estudiantes

```
SQL> select* from estudiantes;
```

ID_EST	NOMBRE	APELLIDO

LICENCIATURA		

101	Tom	Hopper
Mecánica		

102	David	Castañeda
Mecánica		

103	Emmy	Raver
Logística y Cadena de suministro		

ID_EST	NOMBRE	APELLIDO

LICENCIATURA		

104	Robert	Sheehan
Logística y Cadena de suministro		

105	Aidan	Gallagher
Sistemas de Información		

106	Justin	Min
Mecánica		

ID_EST	NOMBRE	APELLIDO

LICENCIATURA		

107	Elle	Page
Sistemas de Información		

108	Eudora	Patch
Sistemas de Información		

109	Harlan	Cooper
Sistemas de Información		

```

ID_EST NOMBRE                                APELLIDO
-----
LICENCIATURA
-----
110 Archie                                Andrews
Mecánica
111 Betty                                Cooper
Logística y Cadena de suministro
112 Veronica                            Lodge
Logística y Cadena de suministro

ID_EST NOMBRE                                APELLIDO
-----
LICENCIATURA
-----
113 Jughead                            Jones
Sistemas de Información
114 Cheryl                            Blossom
Logística y Cadena de suministro
115 Toni                                Topaz
Mecánica

15 rows selected.

```

BLOQUE PL/SQL

```

SQL> DECLARE
2   va_id_est      estudiantes.id_est%TYPE;
3   va_nombre      estudiantes.nombre%TYPE;
4   va_apellido    estudiantes.apellido%TYPE;
5   va_licenciatura estudiantes.licenciatura%TYPE := 'Sistemas de Información';
6
7   CURSOR c_alumno IS
8     SELECT id_est, nombre, apellido
9     FROM estudiantes
10    WHERE licenciatura = va_licenciatura;
11
12 BEGIN
13   OPEN c_alumno;
14   LOOP
15     FETCH c_alumno INTO va_id_est, va_nombre, va_apellido;
16     EXIT WHEN c_alumno%NOTFOUND;
17
18     INSERT INTO Est_sistemasdeinfo (v_id_est, v_nombre, v_apellido)
19     VALUES (va_id_est, va_nombre, va_apellido);
20
21
22   END LOOP;
23   CLOSE c_alumno;
24   COMMIT;
25 END;
26 /

```

PL/SQL procedure successfully completed.

Consulta de la información guardada.

```
SQL> select* from est_sistemasdeinfo;
```

V_ID_EST	V_NOMBRE	V_APELLIDO
----------	----------	------------

105	Aidan	Gallagher
-----	-------	-----------

```
SQL>
```

2. Complete el bloque pl/sql de la ppt No.8 de manera que podamos observar que información está extrayendo el cursor basado en variables de acoplamiento.

```
SQL>
SQL> DECLARE
  2  V_id_salon          salones.id_salon%TYPE;
  3  V_edificio          salones.edificio%TYPE;
  4  V_departamento     clases.departamento%TYPE;
  5  V_curso             clases.curso%TYPE;
  6
  7  CURSOR v_cursor IS
  8  SELECT edificio
  9  FROM salones, clases
 10 WHERE salones.id_salon = clases.id_clase
 11 AND departamento = v_departamento
 12 AND curso = v_curso;
 13
 14 BEGIN
 15 V_departamento := 'HIS';
 16 V_curso := 'a1';
 17
 18 OPEN v_cursor;
 19 FETCH v_cursor into v_edificio;
 20 While v_cursor%FOUND
 21 LOOP
 22 DBMS_OUTPUT.PUT_LINE(v_curso);
 23 FETCH v_cursor into v_edificio;
 24 END LOOP;
 25 CLOSE v_cursor;
 26 END;
 27 /

PL/SQL procedure successfully completed.
```

3. Modifique el bloque anterior aplicando el concepto de cursores parametrizados.

```
SQL> DECLARE
  2  CURSOR v_cursor (
  3  V_departamento    clases.departamento%TYPE,
  4  V_curso            clases.curso%TYPE
  5  ) IS
  6
  7  SELECT edificio
  8  FROM salones, clases
  9  WHERE salones.id_salon = clases.id_clase
 10  AND departamento = v_departamento
 11  AND curso = v_curso;
 12
 13 BEGIN
 14
 15  OPEN v_cursor ('HIS', 'a1');
 16  CLOSE v_cursor;
 17  END;
 18  /

PL/SQL procedure successfully completed.
```


4. Implemente el bloque pl/sql de la ppt No.17 donde se valida el uso de los atributos para los cursores implícitos y la cláusula SELECT.

```
SQL> create table recibirdatos(  
2  columna_a varchar2(50),  
3  columna_b varchar2(50));  
  
Table created.
```

```
SQL> SET SERVEROUTPUT ON;  
SQL> DECLARE  
2  V_edificio salones%ROWTYPE;  
3  
4  BEGIN  
5  SELECT * INTO v_edificio  
6  FROM salones  
7  WHERE id_salon =1;  
8  
9  IF SQL%NOTFOUND THEN  
10 INSERT INTO recibirdatos(columna_a) values('NOT FOUND');  
11 END IF;  
12  
13 EXCEPTION  
14 WHEN NO_DATA_FOUND THEN  
15 INSERT INTO recibirdatos(columna_a) values('NOT FOUND, EXCEPTION HANDLER');  
16 END;  
17 /  
  
PL/SQL procedure successfully completed.
```

5. Diseñe bloque pl/sql que extraiga información de una relación o tabla de base de datos por medio de un cursor y esta se actualizada en el área de ejecución del proceso estableciendo los controles para el proceso de actualización.

```
SQL> Set serveroutput on;
SQL> DECLARE
  2  V_nombre_serie serie.nombre_serie%TYPE;
  3  V_capitulos serie.capitulos%TYPE;
  4  V_temporadas serie.temporadas%TYPE;
  5  Cursor c_series is
  6  Select nombre_serie, capitulos
  7  FROM series
  8  Where temporadas >= 2
  9  BEGIN
10  OPEN c_series;
11  Update series
12  Set capitulos >=10;
13  Dbms_output.put_line ('Serie con historia larga');
14  LOOP
15  Fetch c_series into v_nombre_serie, v_capitulos, v_temporadas;
16  Exit when c_series%NOTFOUND;
17  Dbms_output.put_line(v_nombre_serie);
18  End loop;
19  Close c_series;
20  Commit;
21  End;
22  /
```