

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Licenciatura en Ingeniería en Sistemas de Información Sistemas

Cursado: Bases de Datos II

L3 – Manejo de Procedimientos en PLO/SQL

Facilitador: Ing. Henry Lezcano

Integrantes:

Batista, Johel {8-914-587}

Pinilla, Miguel {8-975-2460}

Riley, Rolando {8-972-1033}

Villarreal, Andrés {8-970-1267}

Grupo: 1IF131

Fecha de Entrega: Domingo 09 de octubre de 2022.

Consigna del Problema Para Resolver

Para el caso desarrollado en clases correspondiente al registro académico de los estudiantes por cursos tomados.

Realizar lo siguiente:

- *Implemente el modelo físico normalizado*
- *Implemente los procedimientos para la inserción de la información del registro académico de los estudiantes:*
 - Para Estudiantes
 - Para Cursos
 - Para Calificaciones
- Implemente un cursor para determinar la calificación alfabética de los estudiantes y actualizar en la base de datos

Modelo Lógico-Relacional Normalizado

	Estudiantes	
PK	EST_ID nombre apellido carrera	number varchar 2 varchar 2 varchar 2

	Cursos	
PK	MAT_ID nombre	number varchar 2

	Calificaciones	
PK	FK1 EST_ID FK2 MAT_ID nota_A nota_N	number number char number

--Creación de la Tabla Estudiante

```
CREATE TABLE Estudiantes (  
    est_id INT,  
    nombre VARCHAR2 (15) NOT NULL,  
    apellido VARCHAR2(15) NOT NULL,  
    cedula VARCHAR2 (15) NOT NULL,  
    CONSTRAINT pk_Estudiantes PRIMARY KEY (est_id)  
);
```

--Secuencia del Id de la tabla Estudiantes

```
CREATE SEQUENCE est_seq  
    MINVALUE 1  
    START WITH 1  
    INCREMENT BY 1  
    CACHE 20;
```

--Procedimiento de Inserciones

```
CREATE OR REPLACE PROCEDURE AgregarEstudiante (  
    p_nombre Estudiantes.nombre%TYPE,  
    p_apellido Estudiantes.apellido%TYPE,  
    p_cedula Estudiantes.cedula%TYPE,  
    p_resultado OUT VARCHAR2 ) AS  
BEGIN  
    --Inicializacion de resultado  
    p_resultado := 'Insercion Exitosa';  
    --Se inserta una nueva tupla en la tabla Estudiantes  
    --A partir de esta secuencia se genera el Id_Estudiante
```

```

        INSERT INTO Estudiantes
        VALUES (est_seq.nextval, p_nombre, p_apellido, p_cedula);
COMMIT;
--Manejo de Excepciones de ID Duplicados para Estudiantes
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        p_resultado := 'Error en la Insercion';
    WHEN OTHERS THEN
        p_resultado := 'Error en la Insercion';
END AgregarEstudiante;
/

```

--Ejecución del Procedimiento de Inserciones de Datos

```

DECLARE
    v_resultado VARCHAR2(100);
BEGIN
    AgregarEstudiante('Adriana', 'Hernandez', '2-147-6987',
    v_resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarEstudiante('Leonardo', 'Garcia', '6-254-427', v_resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarEstudiante('Laura', 'Puerto', '8-658-1402', v_resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarEstudiante('Mario', 'Camacho', '6-254-427', v_resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarEstudiante('Oscar', 'Ovalle', '8-984-4127', v_resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);

```

END;

/

```
9  DBMS_OUTPUT.PUT_LINE(v_resultado);
10  AgregarEstudiante('Mario', 'Camacho', '6-254-427', v_resultado);
11  DBMS_OUTPUT.PUT_LINE(v_resultado);
12  AgregarEstudiante('Oscar', 'Ovalle', '8-984-4127', v_resultado);
13  DBMS_OUTPUT.PUT_LINE(v_resultado);
14  END;
15  /
```

Insercion Exitosa

Insercion Exitosa

Insercion Exitosa

Insercion Exitosa

Insercion Exitosa

PL/SQL procedure successfully completed.

SQL> select * from estudiantes;

EST_ID	NOMBRE	APELLIDO	CEDULA
3	Laura	Puerto	8-658-1402
4	Mario	Camacho	6-254-427
5	Oscar	Ovalle	8-984-4127
1	Adriana	Hernandez	2-147-6987
2	Leonardo	Garcia	6-254-427

SQL>

--Creación de la Tabla Cursos

```
CREATE TABLE Cursos (  
  mat_id INT,  
  nombre VARCHAR2(25) NOT NULL,  
  CONSTRAINT pk_Cursos PRIMARY KEY (mat_id)  
);
```

--Procedimiento de Inserciones de Datos

```
CREATE OR REPLACE PROCEDURE AgregarCurso (  
  p_mat_id Cursos.mat_id%TYPE,  
  p_nombre Cursos.nombre%TYPE,  
  p_resultado OUT VARCHAR2 ) AS  
BEGIN  
  --Inicializacion de resultado  
  p_resultado := 'Insercion Exitosa';  
  --inserta una nueva fila a la tabla Cursos  
  --cur_seq es para generar el ID del Curso  
  INSERT INTO Cursos  
  VALUES (p_mat_id, p_nombre);  
  COMMIT;  
  
  EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
      p_resultado := 'Error en la Insercion';  
    WHEN OTHERS THEN  
      p_resultado := 'Error en la Insercion';  
END AgregarCurso;  
/
```

--Bloque Anónimo para Ejecutar el Procedimiento de Inserciones

```
DECLARE
    v_resultado VARCHAR2(100);
BEGIN
    AgregarCurso(1, 'Matematica',v_Resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarCurso(2, 'Espanol', v_Resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
    AgregarCurso(3, 'Historia', v_Resultado);
    DBMS_OUTPUT.PUT_LINE(v_resultado);
END;
/
```

Probamos la excepción creando nuevamente las materias con el mismo id, produciendo duplicidad y ejecutó correctamente.

```
7 DBMS_OUTPUT.PUT_LINE(v_Resultado);
8 AgregarCurso(3, 'Historia', v_Resultado);
9 DBMS_OUTPUT.PUT_LINE(v_resultado);
10 END;
11 /
Error en la Insercion
Error en la Insercion
Error en la Insercion

PL/SQL procedure successfully completed.

SQL> select * from cursos
2 ;

    MAT_ID NOMBRE
-----
1 Matematica
2 Espanol
3 Historia

SQL>
```


--Creación de la Tabla de Calificaciones

```
CREATE TABLE Calificaciones (  
    est_id INT,  
    mat_id INT NOT NULL,  
    nota_n FLOAT NOT NULL,  
    nota_a CHAR(1),  
    CONSTRAINT fk_Estudiantes FOREIGN KEY (est_id) REFERENCES  
    Estudiantes (est_id),  
    CONSTRAINT fk_Cursos FOREIGN KEY (mat_id) REFERENCES Cursos  
    (mat_id),  
    CONSTRAINT pk_Calificaciones PRIMARY KEY (est_id, mat_id)  
);
```

--Procedimiento de Inserciones de Datos a la Tabla Calificaciones

```
CREATE OR REPLACE PROCEDURE AgregarCalificacion (  
    p_est_id Calificaciones.est_id%TYPE,  
    p_mat_id Calificaciones.mat_id%TYPE,  
    p_nota_n Calificaciones.nota_n%TYPE,  
    p_resultado OUT VARCHAR2 ) AS  
BEGIN  
    --Inicializacion de resultado  
    p_resultado := 'Insercion Exitosa';  
    --inserta una nueva fila a la tabla Calificaciones  
    INSERT INTO Calificaciones (est_id, mat_id, nota_n)  
    VALUES (p_est_id, p_mat_id, p_nota_n);  
COMMIT;  
  
--Manejo de excepciones a nivel de calificaciones duplicadas en un  
estudiante
```

```

EXCEPTION

    WHEN DUP_VAL_ON_INDEX THEN

        p_resultado := 'Error en la Insercion';

    WHEN OTHERS THEN

        p_resultado := 'Error en la Insercion';

END AgregarCalificacion;

/

```

--Ejecución del Proceso de Inserciones de Calificaciones

```

DECLARE

    P_resultado VARCHAR2(100);

BEGIN

    AgregarCalificacion(1,1,95, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(2,1,76, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(3,1,89, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(4,1,61, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(1,2,91, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(2,2,59, p_resultado);
    DBMS_OUTPUT.PUT_LINE(p_resultado);
    AgregarCalificacion(3,2,80, p_resultado);

```

```

DBMS_OUTPUT.PUT_LINE(p_resultado);
AgregarCalificacion(5,2,79, p_resultado);
DBMS_OUTPUT.PUT_LINE(p_resultado);
AgregarCalificacion(1,3,85, p_resultado);
DBMS_OUTPUT.PUT_LINE(p_resultado);
AgregarCalificacion(2,3,77, p_resultado);
DBMS_OUTPUT.PUT_LINE(p_resultado);
AgregarCalificacion(4,3,97, p_resultado);
DBMS_OUTPUT.PUT_LINE(p_resultado);
AgregarCalificacion(5,3,69, p_resultado);
DBMS_OUTPUT.PUT_LINE(p_resultado);
END;
/

```

```

11 DBMS_OUTPUT.PUT_LINE(p_resultado);
12 AgregarCalificacion(1,2,91, p_resultado);
13 DBMS_OUTPUT.PUT_LINE(p_resultado);
14 AgregarCalificacion(2,2,59, p_resultado);
15 DBMS_OUTPUT.PUT_LINE(p_resultado);
16 AgregarCalificacion(3,2,80, p_resultado);
17 DBMS_OUTPUT.PUT_LINE(p_resultado);
18 AgregarCalificacion(5,2,79, p_resultado);
19 DBMS_OUTPUT.PUT_LINE(p_resultado);
20 AgregarCalificacion(1,3,85, p_resultado);
21 DBMS_OUTPUT.PUT_LINE(p_resultado);
22 AgregarCalificacion(2,3,77, p_resultado);
23 DBMS_OUTPUT.PUT_LINE(p_resultado);
24 AgregarCalificacion(4,3,97, p_resultado);
25 DBMS_OUTPUT.PUT_LINE(p_resultado);
26 AgregarCalificacion(5,3,69, p_resultado);
27 DBMS_OUTPUT.PUT_LINE(p_resultado);
28 END;
29 /

```

Insercion Exitosa
Insercion Exitosa
Insercion Exitosa
Insercion Exitosa
Insercion Exitosa
PL/SQL procedure successfully completed
SQL> select * from Calificaciones;

EST_ID	MAT_ID	NOTA_N N
3	1	89
4	1	61
1	2	91
2	2	59
3	2	80
5	2	79
1	3	85
2	3	77
4	3	97
5	3	69
1	1	95

Insercion Exitosa
Insercion Exitosa
Insercion Exitosa
Insercion Exitosa
Insercion Exitosa

EST_ID	MAT_ID	NOTA_N N
2	1	76

12 rows selected.

3. PROGRAMA PL/SQL

Implemente un cursor para determinar la calificación alfabética de los estudiantes y actualizar en la base de datos.

DECLARE

```
CURSOR c_ConversionCal IS
SELECT * FROM Calificaciones
WHERE mat_id = '1';
```

v_calificaciones c_ConversionCal%ROWTYPE;

BEGIN

OPEN c_ConversionCal;

FETCH c_ConversionCal INTO v_calificaciones;

WHILE c_ConversionCal%FOUND LOOP

CASE

WHEN v_calificaciones.nota_n >= 91 THEN

v_calificaciones.nota_a := 'A';

WHEN v_calificaciones.nota_n >= 81 THEN

v_calificaciones.nota_a := 'B';

WHEN v_calificaciones.nota_n >= 71 THEN

v_calificaciones.nota_a := 'C';

WHEN v_calificaciones.nota_n >= 61 THEN

v_calificaciones.nota_a := 'D';

ELSE v_calificaciones.nota_a := 'F';

END CASE;

UPDATE Calificaciones SET nota_a = v_calificaciones.nota_a

```

        WHERE mat_id = v_calificaciones.mat_id AND
        est_id = v_calificaciones.est_id;

    FETCH c_ConversionCal INTO v_calificaciones;

    END LOOP;

    CLOSE c_ConversionCal;

    COMMIT;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('Valor no encontrado');

END;

/

```

--Creación de Vista con Calificaciones y los Estudiantes

```

    CREATE VIEW VistaNotas AS

    SELECT e.nombre Nombre_Estudiante, c.nombre Nombre_Curso, cal.nota_n,
    cal.nota_a

    FROM Calificaciones cal

    JOIN Cursos c ON c.mat_id = cal.mat_id

    JOIN Estudiantes e ON e.est_id = cal.est_id

    ORDER BY Nombre_Estudiante;

```

--Consulta a la vista

```

SELECT * FROM VistaNotas;

```

```
SQL> select * from vistanotas;
```

NOMBRE_ESTUDIAN	NOMBRE_CURSO	NOTA_N	N
Adriana	Matematica	95	
Adriana	Espanol	91	
Adriana	Historia	85	
Laura	Matematica	89	
Laura	Espanol	80	
Leonardo	Espanol	59	
Leonardo	Historia	77	
Leonardo	Matematica	76	
Mario	Matematica	61	
Mario	Historia	97	
Oscar	Historia	69	

NOMBRE_ESTUDIAN	NOMBRE_CURSO	NOTA_N	N
Oscar	Espanol	79	

```
12 rows selected.
```

```
SQL>
```