Nombre: Santiago López
Semestre: 4to A
Docente: Ing. Edwin Buenaño
Materia: Gestión de Base de Datos
Tema: Ejercicios con lazos
Fecha: 15/11/2025

---------- **CREACION DEL USUARIO Y PERMISOS** ----------

CONNECT SYSTEM
ORACLE
CREATE USER EJE_LAZOS IDENTIFIED BY EL;
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO EJE_LAZOS;
CONNECT EJE_LAZOS
CLAVE: EL

```
SQL> CONNECT SYSTEM
Introduzca la contrase±a:
Conectado.
SQL> CREATE USER EJE_LAZOS IDENTIFIED BY EL;

Usuario creado.

SQL> GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO EJE_LAZOS;

Concesi¾n terminada correctamente.

SQL> CONNECT EJE_LAZOS
Introduzca la contrase±a:
Conectado.
```

---------- **CREACION DE LAS TABLAS** ----------

```
CREATE TABLE TABLA_SUMA (
   N1 NUMBER,
   N2 NUMBER,
   SUMA_TOTAL NUMBER
);

CREATE TABLE TABLA_PARES_IMPARES (
   N1 NUMBER,
   N2 NUMBER,
   SUMA_PARES NUMBER,
   PRODUCTO_IMPARES NUMBER
);

CREATE TABLE TABLA_POTENCIA (
   BASE NUMBER,
   EXPO NUMBER,
   RESULTADO NUMBER
);
```

```
SQL> CREATE TABLE TABLA_SUMA (
  2       N1 NUMBER,
  3       N2 NUMBER,
  4       SUMA_TOTAL NUMBER
  5  );

Tabla creada.

SQL> CREATE TABLE TABLA_PARES_IMPARES (
  2       N1 NUMBER,
  3       N2 NUMBER,
  4       SUMA_PARES NUMBER,
  5       PRODUCTO_IMPARES NUMBER
  6  );

Tabla creada.

SQL> CREATE TABLE TABLA_POTENCIA (
  2       BASE NUMBER,
  3       EXPO NUMBER,
  4       RESULTADO NUMBER
  5  );

Tabla creada.
```

---------- CREACION PROCEDURE 1 ----------

Crear un procedure que reciba 2 numeros(n1 , n2) controlar que n2 sea mayor que n1, y devuelva la suma de los numeros contenidos entre n1 y n2 , sin incluir los extremos n1 y n2

Execute suma_intermedios(5,12) -> 6+7+8+9+10+11 -> 51
Tabla_suma=(5, 12, 51)

```
CREATE OR REPLACE PROCEDURE SUMA_INTERMEDIOS(N1_IN NUMBER,
N2_IN NUMBER)
IS
SUMA NUMBER := 0; I NUMBER; ERROR_N2 EXCEPTION;
BEGIN
   IF N2_IN <= N1_IN THEN
      RAISE ERROR_N2;
   END IF;

   I := N1_IN + 1;
```

```
  LOOP
     EXIT WHEN I >= N2_IN;
     SUMA := SUMA + I;
     I := I + 1;
  END LOOP;

  INSERT INTO TABLA_SUMA (N1, N2, SUMA_TOTAL)
                  VALUES (N1_IN, N2_IN, SUMA);

  COMMIT;

 EXCEPTION
  WHEN ERROR_N2 THEN RAISE_APPLICATION_ERROR(-20001, 'ERROR: N2
debe ser mayor que N1');
END SUMA_INTERMEDIOS;
.
/
```

```
SQL> CREATE OR REPLACE PROCEDURE SUMA_INTERMEDIOS(N1_IN NUMBER, N2_IN NUMBER)
  2  IS
  3  SUMA NUMBER := 0; I NUMBER; ERROR_N2 EXCEPTION;
  4  BEGIN
  5      IF N2_IN <= N1_IN THEN
  6          RAISE ERROR_N2;
  7      END IF;
  8
  9      I := N1_IN + 1;
 10
 11      LOOP
 12          EXIT WHEN I >= N2_IN;
 13          SUMA := SUMA + I;
 14          I := I + 1;
 15      END LOOP;
 16
 17      INSERT INTO TABLA_SUMA (N1, N2, SUMA_TOTAL)
 18                          VALUES (N1_IN, N2_IN, SUMA);
 19
 20      COMMIT;
 21
 22    EXCEPTION
 23      WHEN ERROR_N2 THEN RAISE_APPLICATION_ERROR(-20001, 'ERROR: N2 debe ser mayor que N1');
 24  END SUMA_INTERMEDIOS;
 25  .
SQL> /

Procedimiento creado.
```

---------- EJECUCION PROCEDURE 1 ----------

```
EXEC SUMA_INTERMEDIOS (5,12);
EXEC SUMA_INTERMEDIOS (4,10);
EXEC SUMA_INTERMEDIOS (3,6);
```

```
SQL> SELECT * FROM TABLA_SUMA;

ninguna fila seleccionada

SQL> EXEC SUMA_INTERMEDIOS (5,12);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_SUMA;

        N1         N2 SUMA_TOTAL
---------- ---------- ----------
         5         12         51

SQL> EXEC SUMA_INTERMEDIOS (4,10);

Procedimiento PL/SQL terminado correctamente.

SQL> EXEC SUMA_INTERMEDIOS (3,6);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_SUMA;

        N1         N2 SUMA_TOTAL
---------- ---------- ----------
         5         12         51
         4         10         35
         3          6          9
```

---------- CREACION PROCEDURE 2 ----------

Crear un procedure que reciba 2 numeros diferentes(n1 , n2) , y devuelva a una tabla la suma de los numeros pares contenidos entre n1 y n2 , sin incluir los extremos n1 y n2 , y el producto de los numeros impares contenidos entre n1 y n2 , sin incluir los extremos n1 y n2

```
CREATE OR REPLACE PROCEDURE PARES_IMPARES(N1 IN NUMBER, N2 IN
NUMBER) AS
   SUMA_P NUMBER := 0;
   PROD_I NUMBER := 1;
   I NUMBER;
   A NUMBER;
   B NUMBER;
BEGIN
   IF N1 = N2 THEN
      RAISE_APPLICATION_ERROR(-20002, 'Los números deben ser diferentes');
   END IF;

   A := N1;
   B := N2;
```

```
    IF B < A THEN
        I := A;
        A := B;
        B := I;
    END IF;

    I := A + 1;

    WHILE I < B LOOP
        IF MOD(I, 2) = 0 THEN
            SUMA_P := SUMA_P + I;
        ELSE
            PROD_I := PROD_I * I;
        END IF;

        I := I + 1;
    END LOOP;

    INSERT INTO TABLA_PARES_IMPARES VALUES (A, B, SUMA_P, PROD_I);

END;
.
/
```

```
SQL> CREATE OR REPLACE PROCEDURE PARES_IMPARES(N1 IN NUMBER, N2 IN NUMBER) AS
  2      SUMA_P NUMBER := 0;
  3      PROD_I NUMBER := 1;
  4      I NUMBER;
  5      A NUMBER;
  6      B NUMBER;
  7  BEGIN
  8      IF N1 = N2 THEN
  9          RAISE_APPLICATION_ERROR(-20002, 'Los números deben ser diferentes');
 10      END IF;
 11
 12      A := N1;
 13      B := N2;
 14
 15      IF B < A THEN
 16          I := A;
 17          A := B;
 18          B := I;
 19      END IF;
 20
 21      I := A + 1;
 22
 23      WHILE I < B LOOP
 24          IF MOD(I, 2) = 0 THEN
 25              SUMA_P := SUMA_P + I;
 26          ELSE
 27              PROD_I := PROD_I * I;
 28          END IF;
 29
 30          I := I + 1;
 31      END LOOP;
 32
 33      INSERT INTO TABLA_PARES_IMPARES VALUES (A, B, SUMA_P, PROD_I);
 34
 35  END;
 36  .
SQL> /

Procedimiento creado.
```

---------- EJECUCION PROCEDURE 2 ----------

Caso 1: Números en orden normal

EXECUTE PARES_IMPARES(3,11);

```
SQL> SELECT * FROM TABLA_PARES_IMPARES;

ninguna fila seleccionada

SQL> EXECUTE PARES_IMPARES(3,11);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_PARES_IMPARES;

        N1         N2 SUMA_PARES PRODUCTO_IMPARES
---------- ---------- ---------- ----------------
         3         11         28              315
```

Caso 2: Números en orden invertido (N2 < N1)

EXECUTE PARES_IMPARES(20, 6);

```
        N1         N2 SUMA_PARES PRODUCTO_IMPARES
---------- ---------- ---------- ----------------
         3         11         28              315

SQL> EXECUTE PARES_IMPARES(20, 6);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_PARES_IMPARES;

        N1         N2 SUMA_PARES PRODUCTO_IMPARES
---------- ---------- ---------- ----------------
         3         11         28              315
         6         20         78         43648605
```

---------- CREACION PROCEDURE 3 ----------

USANDO WHILE. CREAR UN PROCEDURE QUE RECIBA 2 NUMEROS EL PRIMER NUMERO ES LA BASE, Y EL SEGUNDO ES EL EXPONENTE, SE BUSCA OBTENER EL RESULTADO DE LA POTENCIACION (BASE ELEVADA AL EXPONENTE) // WHILE.

```
CREATE OR REPLACE PROCEDURE POTENCIA_WHILE(BASE IN NUMBER,
EXPO IN NUMBER) AS
   RESULTADO NUMBER := 1;
```

```
   CONTADOR NUMBER := 1;
BEGIN
   WHILE CONTADOR <= EXPO LOOP
      RESULTADO := RESULTADO * BASE;
      CONTADOR := CONTADOR + 1;
   END LOOP;

   INSERT INTO TABLA_POTENCIA VALUES (BASE, EXPO, RESULTADO);

END;
.
/
```

```
SQL> CREATE OR REPLACE PROCEDURE POTENCIA_WHILE(BASE IN NUMBER, EXPO IN NUMBER) AS
  2      RESULTADO NUMBER := 1;
  3      CONTADOR NUMBER := 1;
  4  BEGIN
  5      WHILE CONTADOR <= EXPO LOOP
  6          RESULTADO := RESULTADO * BASE;
  7          CONTADOR := CONTADOR + 1;
  8      END LOOP;
  9
 10      INSERT INTO TABLA_POTENCIA VALUES (BASE, EXPO, RESULTADO);
 11
 12  END;
 13  .
SQL> /

Procedimiento creado.
```

---------- EJECUCION PROCEDURE 3 ----------

```
EXECUTE POTENCIA_WHILE(3,4);
EXECUTE POTENCIA_WHILE(2,2);
SELECT * FROM TABLA_POTENCIA;
```

```
SQL> SELECT * FROM TABLA_POTENCIA;

ninguna fila seleccionada

SQL> EXECUTE POTENCIA_WHILE(3,4);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_POTENCIA;

      BASE       EXPO  RESULTADO
---------- ---------- ----------
         3          4         81

SQL> EXECUTE POTENCIA_WHILE(2,2);

Procedimiento PL/SQL terminado correctamente.

SQL> SELECT * FROM TABLA_POTENCIA;

      BASE       EXPO  RESULTADO
---------- ---------- ----------
         3          4         81
         2          2          4
```