

| GUÍA 2.1.2:

Construcción de procedimientos almacenados para procesar información masiva

Sigla	Asignatura	Experiencia de Aprendizaje
BDY1103	Taller de Base de Datos	EA2: Implementa programas PL/SQL en la base de datos para construir una solución integral de procesamiento y generación de información o ser usados en otros procesos y/o aplicaciones.
Tiempo	Modalidad de Trabajo	Indicadores de logro
4h	Individual	IL2.1

Antecedentes generales

En esta guía encontrarás los contenidos asociados a la construcción de procedimientos almacenados para procesar información masiva, junto con ejemplos y actividades prácticas a desarrollar.

Requerimientos para esta actividad

En esta actividad, los y las estudiantes deberán utilizar SQL Developer y seguir las instrucciones indicadas por el/la docente.



Sesión 1: Construcción de procedimientos almacenados para procesar información masiva

Objetivos de Aprendizaje

1. Comprender cómo construir procedimientos almacenados con y sin parámetros.
2. Aprender a procesar información masiva utilizando estos procedimientos.
3. Integrar procedimientos y funciones en otros programas PL/SQL y sentencias SQL.

1. Introducción a los Procedimientos Almacenados

1.1 ¿Qué son?

- Procedimientos Almacenados: Subprogramas que realizan tareas específicas. Se almacenan en la base de datos y pueden aceptar parámetros de entrada y salida. A diferencia de las funciones, no tienen que devolver un valor, aunque pueden hacerlo usando parámetros de salida.

1.2 Ventajas

- Reutilización de Código: Facilita la reutilización de lógica compleja en múltiples lugares.
- Encapsulación: Permite encapsular la lógica de negocio.
- Rendimiento: Reduce la sobrecarga de comunicación cliente-servidor al ser ejecutadas en el servidor.

1.3 Estructura Básica

```
CREATE OR REPLACE PROCEDURE procedure_name IS
BEGIN
    -- Código del procedimiento
END;
/
```

2. Procedimientos sin Parámetros

2.1 Concepto

- Procedimientos sin Parámetros: Ejecutan una tarea específica sin necesidad de datos externos. Útiles para operaciones estándar como tareas de mantenimiento.

2.2 Ejemplo Práctico: Generación de Informe de Empleados

Problema: Generar un informe simple que liste todos los empleados.

```
CREATE OR REPLACE PROCEDURE generate_employee_report IS
BEGIN
    FOR employee IN (SELECT empno, ename, job, sal FROM emp) LOOP
        DBMS_OUTPUT.PUT_LINE('Emp ID: ' || employee.empno || ', Name: ' ||
employee.ename ||
                                ', Job: ' || employee.job || ', Salary: ' ||
employee.sal);
    END LOOP;
END;
/
```

2.3 Ejecución del Procedimiento

```
BEGIN
    generate_employee_report;
END;
/
```

3. Procedimientos con Parámetros

3.1 Concepto

- Procedimientos con Parámetros: Permiten pasar datos al procedimiento. Se usan para operaciones que requieren información específica para su ejecución.

3.2 Tipos de Parámetros

- IN: Solo entrada.
- OUT: Solo salida.
- IN OUT: Entrada y salida.

3.3 Ejemplo Práctico: Actualización de Salario

Problema: Actualizar el salario de un empleado dado su ID y el nuevo salario.

```
CREATE OR REPLACE PROCEDURE update_employee_salary(
    p_empno IN NUMBER,
    p_new_salary IN NUMBER
) IS
BEGIN
    UPDATE emp
    SET sal = p_new_salary
    WHERE empno = p_empno;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No employee found with ID: ' || p_empno);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Salary updated for employee ID: ' ||
p_empno);
    END IF;
END;
/
```

3.4 Ejecución del Procedimiento

```
BEGIN
    update_employee_salary(7839,                6000);
END;
/
```

4. Ejemplos Prácticos

Ejemplo 1: Procedimiento sin Parámetros

Objetivo: Listar todos los productos.

```
CREATE OR REPLACE PROCEDURE list_all_products IS
BEGIN
    FOR product IN (SELECT product_id, product_name, unit_price FROM prod-
ucts) LOOP
        DBMS_OUTPUT.PUT_LINE('Product ID: ' || product.product_id || ',
Name: ' || product.product_name ||
                                ', Price: ' || product.unit_price);
    END LOOP;
END;
/

begin
    list_all_products;
end;
/
```

Ejemplo 2: Procedimiento con Parámetros

Objetivo: Actualizar el precio de un producto dado su ID y el nuevo precio.

```
CREATE OR REPLACE PROCEDURE update_product_price(
    p_product_id IN NUMBER,
    p_new_price IN NUMBER
) IS
BEGIN
    UPDATE products
    SET unit_price = p_new_price
    WHERE product_id = p_product_id;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No product found with ID: ' || p_product_id);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Price updated for product ID: ' || p_pro-
duct_id);
    END IF;
END;
/

begin
    update_product_price(32, 55);
end;
/
```

5. Actividades Prácticas

Actividad 1: Procedimiento sin Parámetros

1. Objetivo: Crear un procedimiento que liste todos los departamentos.
2. Tabla: dept
4. Pasos:
 - Crear el procedimiento list_all_departments.
 - Imprimir el deptno y dname de cada departamento.

Actividad 2: Procedimiento con Parámetros

1. Objetivo: Crear un procedimiento que actualice la ubicación de una tienda dada su ID y la nueva ubicación.
2. Tabla: stores
3. Pasos:
 - Crear el procedimiento update_store_location.
 - Aceptar store_id y new_location como parámetros.

Conclusión de la Sesión 1

- Resumen: Los procedimientos almacenados son esenciales para automatizar tareas en bases de datos.

Recursos Adicionales

PL/SQL. *Procedimientos y Funciones*. (2016, enero 1). PL/SQL. Procedimientos y Funciones. <https://elbauldelprogramador.com/plsql-procedimientos-y-funciones/>

DiscoDurodeRoer [@DiscoDurodeRoer]. (2018, agosto 23). *Funciones y procedimientos / Ejercicios PL/SQL #4*. Youtube. <https://www.youtube.com/watch?v=IRp2ReMONx0>