

| GUÍA 2.2.5:

Integración de packages en soluciones de negocio

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	IL2.2



Antecedentes generales

En esta guía encontrarás los contenidos asociados a la construcción de packages con constructores públicos y privados, y su integración con soluciones de negocio, 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 2: Integración de Packages en Soluciones de Negocio

Objetivos de Aprendizaje

1. Comprender la estructura y uso de packages en PL/SQL.
2. Desarrollar packages con constructores públicos y privados.
3. Integrar packages en soluciones de procesamiento y generación de información.

1. Integración de Packages en Soluciones de Negocio

1.1 Concepto

- **Uso de Packages:** Facilitan la creación de soluciones integrales, proporcionando un punto de acceso consistente para procedimientos y funciones relacionados.

1.2 Ejemplo: Package para Generación de Informes de Ventas

Problema: Crear un package que incluya procedimientos para generar informes de ventas por productos y tiendas.

Especificación del Package

```
CREATE OR REPLACE PACKAGE sales_reporting IS
    PROCEDURE generate_product_sales_report;
    PROCEDURE generate_store_sales_report;
END sales_reporting;
/
```

Cuerpo del Package

```
CREATE OR REPLACE PACKAGE BODY sales_reporting IS
    -- Procedimiento Privado
    PROCEDURE print_line(p_text IN VARCHAR2) IS
    BEGIN
        DBMS_OUTPUT.PUT_LINE(p_text);
    END print_line;

    -- Procedimiento Público
    PROCEDURE generate_product_sales_report IS
        CURSOR c_products IS
            SELECT products.product_name, SUM(order_items.quantity) AS total_quantity, SUM(order_items.unit_price * order_items.quantity) AS total_sales
            FROM order_items
            JOIN products ON order_items.product_id = products.product_id
            GROUP BY product_name;
    BEGIN
        print_line('Product Sales Report:');
    END generate_product_sales_report;
END sales_reporting;
```



```

    FOR r_product IN c_products LOOP
        print_line('Product: ' || r_product.product_name);
        print_line('Total Quantity Sold: ' || r_product.total_quantity);
        print_line('Total Sales: ' || r_product.total_sales);
        print_line('-----');
    END LOOP;
END generate_product_sales_report;

-- Procedimiento Público
PROCEDURE generate_store_sales_report IS
    CURSOR c_stores IS
        SELECT store_name, SUM(quantity) AS total_quantity, SUM(or-
der_items.unit_price * quantity) AS total_sales
        FROM order_items
        JOIN orders ON order_items.order_id = orders.order_id
        JOIN stores ON orders.store_id = stores.store_id
        GROUP BY store_name;
BEGIN
    print_line('Store Sales Report:');
    FOR r_store IN c_stores LOOP
        print_line('Store: ' || r_store.store_name);
        print_line('Total Quantity Sold: ' || r_store.total_quantity);
        print_line('Total Sales: ' || r_store.total_sales);
        print_line('-----');
    END LOOP;
END generate_store_sales_report;
END sales_reporting;
/

```

Ejecución del Package

```

begin
    sales_reporting.generate_product_sales_report;
    sales_reporting.generate_store_sales_report;
end;
/

```

3. Ejemplos Prácticos

Ejemplo 1: Package para Gestión de Pedidos con Lógica Privada

Objetivo: Crear un package para gestionar pedidos que incluya procedimientos privados para cálculos detallados.

Especificación del Package

```

CREATE OR REPLACE PACKAGE order_processing IS
    PROCEDURE process_order(p_order_id IN NUMBER);
END order_processing;
/

```

Cuerpo del Package

```
CREATE OR REPLACE PACKAGE BODY order_processing IS
  -- Procedimiento Privado
  FUNCTION calculate_order_total(p_order_id IN NUMBER) RETURN NUMBER IS
    v_order_total NUMBER;
  BEGIN
    SELECT SUM(unit_price * quantity)
    INTO v_order_total
    FROM order_items
    WHERE order_id = p_order_id;

    RETURN v_order_total;
  END calculate_order_total;

  -- Procedimiento Público
  PROCEDURE process_order(p_order_id IN NUMBER) IS
    v_total NUMBER;
  BEGIN
    v_total := calculate_order_total(p_order_id);
    DBMS_OUTPUT.PUT_LINE('Order ID: ' || p_order_id || ', Total: ' ||
v_total);
  END process_order;
END order_processing;
/
```

Ejecución del Package

```
begin
  order_processing.process_order(8);
end;
/
```

Ejemplo 2: Package para Gestión de Departamentos con Lógica Privada

Objetivo: Crear un package para gestionar departamentos con cálculos de promedio de salario como lógica privada.

Especificación del Package

```
CREATE OR REPLACE PACKAGE dept_management IS
  PROCEDURE dept_average_salary(p_deptno IN NUMBER);
END dept_management;
/
```

Cuerpo del Package

```
CREATE OR REPLACE PACKAGE BODY dept_management IS
  -- Procedimiento Privado
  FUNCTION calculate_average_salary(p_deptno IN NUMBER) RETURN NUMBER IS
    v_avg_salary NUMBER;
  BEGIN
    SELECT AVG(sal)
    INTO v_avg_salary
```

```
FROM emp
WHERE deptno = p_deptno;

RETURN v_avg_salary;
END calculate_average_salary;

-- Procedimiento Público
PROCEDURE dept_average_salary(p_deptno IN NUMBER) IS
    v_avg_salary NUMBER;
BEGIN
    v_avg_salary := calculate_average_salary(p_deptno);
    DBMS_OUTPUT.PUT_LINE('Department ID: ' || p_deptno || ', Average
Salary: ' || v_avg_salary);
END dept_average_salary;
END dept_management;
/
```

4. Actividad Práctica

Actividad 1: Package para Gestión de Inventario

Objetivo: Crear un package para gestionar inventarios.

Tabla: products

Pasos:

- Agregar la columna quantity_in_stock number(10) a la tabla PRODUCTS.
- Crear el package `inventory_management` que incluya un procedimiento para actualizar cantidades de productos y una función privada para calcular el valor total de inventario.

Conclusión de la Sesión 2

- **Resumen:** Se implementaron packages con lógica avanzada y constructores privados.