



| GUÍA 1.4.2: Evaluación Procedimientos, Funciones y Packages

Sigla	Asignatura	Experiencia de Aprendizaje
BDY1103	Taller de Base de Datos	EA1: Desarrolla bloques PL/SQL para procesar datos y generar información relevante para el negocio.
Tiempo	Modalidad de Trabajo	Indicadores de logro
4h	Individual	IL1.4



Antecedentes generales

En esta guía encontrarás los contenidos asociados a la evaluación de procedimientos y funciones almacenadas, y packages, 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: EVALUACIÓN DE PROCEDIMIENTOS ALMACENADOS, FUNCIONES ALMACENADAS Y PACKAGES

Objetivos de Aprendizaje

1. Entender el rol de procedimientos almacenados, funciones almacenadas y packages en soluciones de base de datos.
2. Evaluar cuándo y por qué utilizar cada uno de estos componentes en una solución integral.
3. Analizar ejemplos prácticos y realizar actividades que evalúen y mejoren su implementación.

1. Introducción a Procedimientos Almacenados, Funciones Almacenadas y Packages

1.1 Definiciones

- Procedimientos Almacenados: Subprogramas que realizan una tarea específica y pueden aceptar parámetros. No tienen que devolver un valor, pero pueden realizar cambios en la base de datos.
- Funciones Almacenadas: Subprogramas que aceptan parámetros y devuelven un valor. Se usan para realizar cálculos y pueden ser invocadas en consultas SQL.
- Packages: Conjunto de procedimientos y funciones empaquetados. Facilitan la modularización y encapsulación del código.

1.2 Ventajas

- Reutilización de Código: Promueven la reutilización y el mantenimiento eficiente del código.
- Encapsulación: Permiten encapsular la lógica de negocio y proteger los datos.
- Rendimiento: Mejoran el rendimiento al reducir la sobrecarga en la comunicación cliente-servidor.

1.3 Escenarios de Uso

- Procedimientos Almacenados: Automatización de tareas, ejecución de operaciones complejas.
- Funciones Almacenadas: Cálculos y transformación de datos.
- Packages: Organización de la lógica de negocio y agrupación de subprogramas relacionados.



2. Evaluación de su Uso en Soluciones de Negocio

2.1 Procedimientos Almacenados

- Automatización: ¿Necesitamos realizar tareas automatizadas que impliquen múltiples pasos?
- Ejemplos de Uso: Importación de datos, generación de informes.

2.2 Funciones Almacenadas

- Transformación de Datos: ¿Requerimos realizar cálculos o transformaciones en datos?
- Ejemplos de Uso: Cálculo de impuestos, conversión de formatos.

2.3 Packages

- Modularización: ¿Buscamos organizar y encapsular subprogramas relacionados?
- Ejemplos de Uso: Librerías de funciones, módulos de aplicaciones.

2.4 Evaluación Comparativa

- Escalabilidad: ¿Qué componente se adapta mejor a las necesidades futuras?
- Mantenimiento: ¿Cuál ofrece mejor mantenimiento y actualización?
- Rendimiento: ¿Qué opción ofrece mejor rendimiento en el contexto dado?



3. Ejemplos Prácticos de Implementación

3.1 Evaluación de Procedimientos Almacenados

Ejemplo: Automatización de la Actualización de Inventarios

-- Este bloque PL/SQL ilustra cómo un procedimiento almacenado podría usarse -- para actualizar el inventario automáticamente, y se evaluará su necesidad -- y efectividad en comparación con otras soluciones.

Creación de procedimiento:

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

Llamada a procedimiento:

```
BEGIN
    -- Llamada al procedimiento almacenado
    UpdatePrice(32, 50);
    DBMS_OUTPUT.PUT_LINE('Price updated successfully.');
END;
/
```

Evaluación:

- **Automatización:** Evalúa si la tarea es recurrente y necesita automatización.
- **Rendimiento:** Evalúa si reduce la carga de trabajo en comparación con la ejecución manual.

3.2 Evaluación de Funciones Almacenadas

Ejemplo: Cálculo de Descuento de Producto

-- Este bloque PL/SQL simula una función almacenada que calcula descuentos, -- para evaluar cuándo y cómo esta función es necesaria en consultas.

Creación de función:

```
CREATE OR REPLACE FUNCTION CalculateDiscount(p_price IN NUMBER, p_discount_rate IN NUMBER) RETURN NUMBER IS
BEGIN
    RETURN p_price - (p_price * p_discount_rate / 100);
END;
```

Llamada a función:



```
DECLARE
    v_discounted_price NUMBER;
BEGIN
    -- Uso de la función almacenada
    v_discounted_price := CalculateDiscount(100, 10);
    DBMS_OUTPUT.PUT_LINE('Discounted Price: ' || v_discounted_price);
END;
/
```

Evaluación:

- **Transformación de Datos:** Evalúa si la función permite simplificar cálculos en diferentes partes del sistema.
- **Reutilización:** Evalúa la facilidad de reutilización de la lógica de cálculo en múltiples contextos, por ejemplo, una función que calcule el dígito verificador de un RUT.



3.3 Evaluación de Packages

Ejemplo: Librería de Utilidades

- Este bloque PL/SQL muestra cómo un package puede agrupar funciones relacionadas,
- y se evaluará la conveniencia de su uso para modularización y encapsulación.

Creación de package:

```
CREATE OR REPLACE PACKAGE UtilityPackage IS
    FUNCTION ConvertToUpper(p_input VARCHAR2) RETURN VARCHAR2;
    PROCEDURE PrintWelcome(p_name VARCHAR2);
END UtilityPackage;

CREATE OR REPLACE PACKAGE BODY UtilityPackage IS
    FUNCTION ConvertToUpper(p_input VARCHAR2) RETURN VARCHAR2 IS
        BEGIN
            RETURN UPPER(p_input);
        END;

    PROCEDURE PrintWelcome(p_name VARCHAR2) IS
        BEGIN
            DBMS_OUTPUT.PUT_LINE('Welcome ' || ConvertToUpper(p_name) ||
' !');
        END;
END UtilityPackage;
```

Llamada a función dentro del package:

```
DECLARE
    v_name          VARCHAR2(100)      := 'John';
BEGIN
    -- Uso del package
    UtilityPackage.PrintWelcome(v_name);
END;
/
```

Evaluación:

- **Organización:** Evalúa si agrupar subprogramas en un package mejora la organización y claridad.
- **Encapsulación:** Evalúa cómo encapsular la lógica de negocio en un package mejora la seguridad y el mantenimiento.



4. Actividades Prácticas

Actividad 1: Análisis Comparativo

1. **Objetivo:** Evaluar la necesidad de procedimientos, funciones y packages en un contexto de negocio.
2. **Pasos:**
 - Proporciona un escenario de negocio.
 - Discute cuál de los componentes es más adecuado para implementar la solución y por qué.
 - Justifica la elección considerando mantenimiento, rendimiento y escalabilidad.

Escenario de la actividad:

La empresa necesita un sistema para gestionar los descuentos de productos y actualizar inventarios automáticamente basado en ventas.

Análisis:

- **Procedimientos Almacenados:** Para la actualización automática del inventario.
- **Funciones Almacenadas:** Para calcular descuentos en tiempo real.
- **Packages:** Para agrupar la lógica de descuentos y gestión de inventarios.

Actividad 2: Evaluación de Ejemplos Reales

1. **Objetivo:** Evaluar ejemplos de procedimientos, funciones y packages en una base de datos real.
2. **Pasos:**
 - Revisa los ejemplos proporcionados por tu docente.
 - Analiza su implementación, beneficios y posibles áreas de mejora.
 - Proporciona recomendaciones para optimizar o mejorar la implementación actual.

Ejemplo de la actividad:

Revisa la implementación de los packages entregados por tu docente, los que contienen subprogramas para gestionar información y propone mejoras en su modularización.

Conclusión de la Sesión 1

- **Evaluación** de procedimientos, funciones y packages es crucial para la arquitectura de soluciones eficientes.