| GUÍA 1.4.5:
Evaluación de Triggers

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 triggers en base de datos, junto con ejemplos y actividades prácticos 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 TRIGGERS y CONSTRUCCIÓN DE SOLUCIONES INTEGRALES

Objetivos de Aprendizaje

1. Entender el rol que cumplen los triggers 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 Triggers

1.1 Definiciones

- **Triggers**: Subprogramas que se ejecutan automáticamente en respuesta a eventos específicos en una tabla o vista (INSERT, UPDATE, DELETE).

1.2 Ventajas

- **Automatización**: Ejecutan automáticamente tareas basadas en eventos.
- **Integridad de Datos**: Ayudan a mantener la integridad y consistencia de los datos.

1.3 Escenarios de Uso

- **Validación de Datos**: Garantizar que los datos cumplen con ciertos criterios.
- **Registro de Cambios**: Mantener un historial de cambios en los datos.

2. Evaluación de Triggers en Soluciones de Negocio

2.1 Tipos de Triggers

- Before: Se ejecutan antes del evento.
- After: Se ejecutan después del evento.
- Instead Of: Se ejecutan en lugar del evento (principalmente para vistas).

2.2 Evaluación Comparativa

- Automatización: Evalúa si los triggers son necesarios para automatizar tareas que no pueden ser manejadas eficientemente por otros medios.
- Rendimiento: Considera el impacto en el rendimiento, especialmente en operaciones de gran volumen.
- Mantenimiento: Evalúa la facilidad o dificultad de mantener triggers en comparación con otras soluciones.

2.3 Ejemplos de Evaluación

Ejemplo: Mantenimiento de un Registro de Cambios

-- Este bloque PL/SQL ilustra cómo un trigger puede ser usado para
-- mantener un registro de cambios en una tabla, evaluando su utilidad.

```
CREATE OR REPLACE TRIGGER track_changes
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
   INSERT INTO changes_log (emp_id, old_value, new_value, change_date)
   VALUES (:OLD.emp_id, :OLD.salary, :NEW.salary, SYSDATE);
END;
/
```

Evaluación:

- **Automatización**: Evalúa la necesidad de automatizar el registro de cambios.
- **Impacto en Rendimiento**: Analiza el impacto en el rendimiento para operaciones de gran volumen.

3. Construcción de Soluciones Integrales con Procedimientos, Funciones, Packages y Triggers

3.1 Ejemplo de Solución Integral

Escenario: Gestión de una tienda que necesita automatizar la actualización de inventarios, calcular descuentos, y registrar cambios.

Componentes:

- **Procedimientos Almacenados**: Para actualizar inventarios.
- **Funciones Almacenadas**: Para calcular descuentos.
- **Packages**: Para organizar la lógica de descuentos y gestión de inventarios.
- **Triggers**: Para registrar cambios en el inventario.

-- Ejemplo de cómo se combinan todos los componentes para una solución integral.
-- Este es un pseudocódigo que describe la interacción sin detallar las implementaciones.

```
CREATE OR REPLACE PROCEDURE UpdateInventory(p_item_id IN NUMBER,
p_new_quantity IN NUMBER);

CREATE OR REPLACE FUNCTION CalculateDiscount(p_price IN NUMBER, p_dis-
count_rate IN NUMBER) RETURN NUMBER;

CREATE OR REPLACE PACKAGE InventoryPackage IS
      FUNCTION GetStockLevel(p_item_id IN NUMBER) RETURN NUMBER;
      PROCEDURE Restock(p_item_id IN NUMBER, p_quantity IN NUMBER);
```

```
END InventoryPackage;

CREATE OR REPLACE TRIGGER LogInventoryChanges
AFTER UPDATE ON inventory
FOR EACH ROW
BEGIN
   INSERT INTO inventory_log (item_id, old_quantity, new_quantity,
change_date)
   VALUES (:OLD.item_id, :OLD.quantity, :NEW.quantity, SYSDATE);
END;

BEGIN
   -- Integración de los componentes
   UpdateInventory(1001, 20);
   DBMS_OUTPUT.PUT_LINE('New price after discount: ' || CalculateDis-
count(100, 10));
   DBMS_OUTPUT.PUT_LINE('Stock Level: ' || InventoryPackage.GetStock-
Level(1001));
END;
/
```

Evaluación:

- **Coherencia**: Evalúa cómo los componentes trabajan juntos para resolver el problema.
- **Escalabilidad**: Considera cómo la solución puede escalar con el aumento de datos o cambios en requisitos.

4. Actividades Prácticas

Actividad 1: Construcción de una Solución Integral

1. Objetivo: Diseñar una solución integral usando procedimientos, funciones, packages y triggers para un escenario de negocio.
2. Pasos:
    - Proporciona un escenario de negocio.
    - Diseña una solución que integre todos los componentes.
    - Describe cómo cada componente contribuye a la solución.

Actividad 2: Evaluación de Implementaciones

1. Objetivo: Evaluar implementaciones reales de procedimientos, funciones, packages y triggers en una base de datos.
2. Pasos:
    - Revisa las implementaciones presentadas por el docente.
    - Evalúa su efectividad y discute propuestas de mejora.
    - Discute en grupo los hallazgos y las recomendaciones.

Conclusión de la Sesión 2

- Evaluación de Triggers: Importante para la automatización basada en eventos.

Mejores Prácticas

1. Documentación y Pruebas: Siempre documenta y prueba cada componente para asegurar su correcto funcionamiento.
2. Modularización: Mantén la lógica modular y encapsulada para facilitar el mantenimiento y la actualización.
3. Optimización: Evalúa el rendimiento y busca optimizaciones continuas en la implementación de los componentes.

Recursos Adicionales

- *PL/SQL. Disparadores o Triggers*. (2016, enero 1). PL/SQL. Disparadores o Triggers. https://elbauldelprogramador.com/plsql-disparadores-o-triggers/
- tom. (2023, mayo 26). *Triggers en Oracle SQL: Potentes Disparadores para Mejorar la Funcionalidad*. Limirai. https://limirai.com/triggers-en-oracle-sql-potentes-disparadores-para-mejorar-la-funcionalidad/