

| GUÍA 1.3.5:

Excepciones Definidas por el Usuario

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	IL1.3

Antecedentes generales

En esta guía encontrarás los contenidos asociados a excepciones definidas por el usuario en PL/SQL, 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: EXCEPCIONES DEFINIDAS POR EL USUARIO

Objetivos de Aprendizaje

1. Entender las excepciones definidas por el usuario en PL/SQL.
2. Aprender a manejar excepciones en bloques PL/SQL para garantizar la robustez del código.
3. Desarrollar bloques PL/SQL que procesen datos y generen información, manejando adecuadamente las excepciones.
4. Implementar ejemplos prácticos y realizar actividades para consolidar los conceptos aprendidos.

2. Declaración y Uso de Excepciones Definidas por el Usuario

2.1 Declaración de Excepciones

- **Sintaxis:** Declaración en la sección de declaraciones del bloque PL/SQL.

Ejemplo de declaración:

```
DECLARE
    ex_custom_error EXCEPTION;
```

2.2 Uso de RAISE para Lanzar Excepciones

- **RAISE:** Utilizado para lanzar una excepción definida por el usuario.

Ejemplo de uso de RAISE:

```
DECLARE
    ex_invalid_salary EXCEPTION;
    v_salary emp.sal%TYPE := -1000;
BEGIN
    IF v_salary < 0 THEN
        RAISE ex_invalid_salary;
    END IF;
EXCEPTION
    WHEN ex_invalid_salary THEN
        DBMS_OUTPUT.PUT_LINE('Invalid salary detected.');
```

END;
/

3. Manejo de Excepciones en Bloques PL/SQL

3.1 Bloques de Excepción con Excepciones Definidas por el Usuario

- **Declaración en la sección de declaraciones:** Define la excepción en la parte declarativa del bloque.
- **Uso de RAISE_APPLICATION_ERROR:** Proporciona un mensaje de error más descriptivo.

Ejemplo de manejo con RAISE_APPLICATION_ERROR:

```
DECLARE
    ex_invalid_discount EXCEPTION;
    v_discount NUMBER := 150;
BEGIN
    IF v_discount > 100 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Discount cannot exceed 100%.');
    END IF;
EXCEPTION
    WHEN ex_invalid_discount THEN
        DBMS_OUTPUT.PUT_LINE('Error: Invalid discount value.');
```

END;
/

4. Ejemplos Prácticos

4.1 Ejemplo: Validación de Precios

Manejo de excepciones al validar un precio en una tabla de productos:

```
DECLARE
    ex_invalid_price EXCEPTION;
    v_product_id products.product_id%TYPE := 23; /* probar con producto 16*/
    v_price products.unit_price%TYPE;
BEGIN
    SELECT unit_price INTO v_price
    FROM products
    WHERE product_id = v_product_id;

    IF v_price > 45 THEN
        RAISE ex_invalid_price;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Price is valid.');
```

EXCEPTION

```
    WHEN ex_invalid_price THEN
        DBMS_OUTPUT.PUT_LINE('Error: Price exceeds $45.');
```

WHEN OTHERS THEN

```
    DBMS_OUTPUT.PUT_LINE('Unexpected error: ' || SQLERRM);
```

END;
/

4.2 Ejemplo: Validación de Cantidad en Pedidos

Manejo de excepciones al validar la cantidad de un pedido en una tabla de órdenes:

```
DECLARE
    ex_invalid_quantity EXCEPTION;
    v_order_id orders.order_id%TYPE := 21;
    v_product_id order_items.product_id%TYPE := 38;
    v_quantity order_items.quantity%TYPE;
BEGIN
    SELECT quantity INTO v_quantity
    FROM order_items
    WHERE order_id = v_order_id and
           product_id = v_product_id;

    IF v_quantity < 2 THEN
        RAISE ex_invalid_quantity;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Quantity is valid.');
```

EXCEPTION

```
    WHEN ex_invalid_quantity THEN
        DBMS_OUTPUT.PUT_LINE('Error: Quantity must be at least 2.');
```

WHEN OTHERS THEN

```
        DBMS_OUTPUT.PUT_LINE('Unexpected error: ' || SQLERRM);
```

END;

/

5. Actividades Prácticas

Actividad 1: Validación de Precios

1. **Objetivo:** Crear un bloque PL/SQL que valide el precio de un producto (tabla products) y maneje excepciones definidas por el usuario.
2. **Pasos:**
 - Consulta el precio de un producto.
 - Maneja la excepción si el precio es negativo o mayor a un valor máximo permitido.

Actividad 2: Validación de Fechas de Pedido

1. **Objetivo:** Crear un bloque PL/SQL que valide la fecha de un pedido (tabla orders) y maneje excepciones definidas por el usuario.
2. **Pasos:**
 - Consulta la fecha de un pedido.
 - Maneja la excepción si la fecha es futura o pasada más allá de un límite razonable.



Conclusión de la Sesión 2

- Manejo de excepciones definidas por el usuario proporciona flexibilidad para manejar condiciones específicas que el desarrollador quiera manejar de forma personalizada.

Recursos Adicionales

- Informáticas, B. L. •. [@bilbaolabs]. (2024, mayo 6). 17. *EXCEPCIONES / EXCEPTION en PL/SQL - ORACLE*. Youtube.

<https://www.youtube.com/watch?v=KWdoe7oq-hk>