

Manejo de Excepciones

Curso : Base de datos II

Profesor : Jesús Canales Guando

Alumno :
Máximo Huamán Allcahuamán

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif, uppercase font, centered within a solid red square.

Manejo de Excepciones

¿Qué es una excepción?

Se produce una excepción cuando se descubre un error durante la ejecución de un programa que interrumpe el normal funcionamiento del programa.

Ejemplos de excepciones :

- Ingresar un nombre de usuario y/o contraseña incorrectos.
- Olvidar incluir @ en una dirección de correo electrónico.
- Ingresar un número de tarjeta de crédito incorrectamente.
- Ingresar una fecha de vencimiento que ya pasó.

Manejo de Excepciones

¿Qué es un controlador de excepciones?

Al escribir código, los programadores deben anticipar los tipos de errores que pueden ocurrir durante la ejecución. Un controlador de excepciones es un código que define las acciones de recuperación que se realizarán cuando se genere una excepción (es decir, cuando se produzca un error).

¿Por qué es importante el manejo de excepciones?

- Para proteger al usuario de errores (los mensajes de error pueden frustrar usuarios/clientes y esto no es bueno).
- Proteger la base de datos (los errores pueden ser costosos, en tiempo y recursos).

Manejo de Excepciones

La sección de excepciones empieza con la palabra clave **EXCEPTION**

```
DECLARE
    v_country_name countries.country_name%TYPE := 'Korea, South';
    v_elevation countries.highest_elevation%TYPE;
BEGIN
    SELECT highest_elevation INTO v_elevation
    FROM countries WHERE country_name = v_country_name;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('Country name, ' || v_country_name || ',
        cannot be found. Re-enter the country name using the correct
        spelling. ');
END;
```

Manejo de Excepciones

TRAPPING :

- La sección de manejo de excepciones captura sólo aquellas excepciones que se especifican; cualquier otra excepción no está atrapada a menos que use el controlador de excepciones **OTHERS**.

- El controlador **OTHERS** captura todas las excepciones que no están ya atrapados.

- Si se usa, **OTHERS** debe ser el último controlador de excepciones definido

```
EXCEPTION
    WHEN exception1 [OR exception2 . . .] THEN
        statement1;
        statement2;
    . . .
    [WHEN OTHERS THEN
        statement1;
        statement2;
    . . .]
```

Captura de excepciones de Oracle Server

Servidor Oracle predefinido error :

Errores PL / SQL más comunes (alrededor de 20 más o menos que se nombran)

Oracle no predefinido error del servidor :

Otros errores de PL/SQL (sin nombre)

Error definido por el usuario :

Definidos por el programador



Captura de excepciones de Oracle Server

Detección de errores predefinidos de Oracle Server

Hacer referencia al nombre predefinido en la excepción. Ejemplo

- **NO_DATA_FOUND**
- **TOO_MANY_ROWS**
- **INVALID_CURSOR**
- **ZERO_DIVIDE**
- **DUP_VAL_ON_INDEX**

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees WHERE job_id = 'ST_CLERK';
  DBMS_OUTPUT.PUT_LINE('The last name of the ST_CLERK is: ' ||
v_lname);
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved multiple
                           rows. Consider using a cursor.');
```

END;

Captura de excepciones definidas por el usuario

- Además de los errores predefinidos de Oracle, los programadores pueden crear sus propias definiciones de error.
- Los errores definidos por los usuarios aparte del que presenta Oracle Server también pueden ser definidos por el programador y debe ser planteado por el programador cuando ocurran excepciones.
- Con un error definido por el usuario, el programador crea un código de error y un mensaje de error.
- Un ejemplo de error definido por el usuario podría ser `INVALID_MANAGER_ID`

Captura de excepciones definidas por el usuario

```
DECLARE
  e_invalid_department EXCEPTION; 1
  v_name VARCHAR2(20) := 'Accounting';
  v_deptno NUMBER := 27;
BEGIN
  UPDATE departments
    SET      department_name = v_name
    WHERE    department_id = v_deptno;
  IF SQL%NOTFOUND THEN
    RAISE e_invalid_department; 2
  END IF;
EXCEPTION
  WHEN e_invalid_department 3
    THEN DBMS_OUTPUT.PUT_LINE('No such department id.');
```

END;

Captura de excepciones definidas por el usuario

La declaración RAISE

- Se puede utilizar la declaración **RAISE** para generar excepciones.
- Puede utilizar el procedimiento **RAISE_APPLICATION_ERROR** para retornar mensajes de error definidos por el usuario almacenado en subprogramas.
- La principal ventaja de utilizar este procedimiento en lugar de **RAISE**, es que **RAISE_APPLICATION_ERROR** te permite para asociar tu propio número de error y significativo mensaje con la excepción.

Captura de excepciones definidas por el usuario

La sintaxis de **RAISE_APPLICATION_ERROR**

- **TRUE** | **FALSE** es un parámetro booleano opcional
- Si es **TRUE**, el error se coloca en la pila de errores.
- Si es **FALSE**, el valor predeterminado reemplaza todos los anteriores errores.
- Puede utilizar **RAISE_APPLICATION_ERROR** en dos diferentes lugares: sección ejecutable y sección de excepciones

```
RAISE_APPLICATION_ERROR (error_number,  
                           message [, {TRUE | FALSE}]);
```