PL/SQL Excepciones

Antonio Espín Herranz

Control de excepciones

- Manejo de excepciones.
- Excepciones predefinidas.
- Excepciones predefinidas por los usuarios.
- Reglas de alcance.
- Personalización de excepciones.

Control de Excepciones

- El lenguaje PL/SQL proporciona un mecanismo de detección de errores que permite dar una respuesta y prever una salida correcta del programa cuando se produce.

- Se definen dentro de la sección **EXCEPTION**, es opcional y sólo se define cuando el bloque deba capturar errores.

```
DECLARE
    -- Declaración de variables y excepciones.;
    BEGIN
        -- Instrucciones;
    EXCEPTION
        --Tratamiento de las excepciones;
    END;
/
```

Control de Excepciones

- **Reglas**:
  - Definir y **proporcionar un nombre a cada error** (diferente para los errores de usuario y errores de Oracle).
  - **Asociar una entrada** en la **sección EXCEPTION** para cada nombre de error definido **en la sección DECLARE**.
  - **Definir el tratamiento** que hay que realizar en la sección EXCEPTION.

Excepciones predefinidas

- Todos los errores en Oracle poseen un número de identificación exclusivo.
- En el paquete estándar de Oracle está definida toda la lista de excepciones predefinidas.
- Para tratar los restantes errores de Oracle, siempre es posible utilizar la palabra OTHERS y llamar a las funciones SQLCODE y SQLERRM.

Lista de errores

Excepcion	Se ejecuta ...	SQLCODE
ACCESS_INTO_NULL	El programa intentó asignar valores a los atributos de un objeto no inicializado	-6530
COLLECTION_IS_NULL	El programa intentó asignar valores a una tabla anidada aún no inicializada	-6531
CURSOR_ALREADY_OPEN	El programa intentó abrir un cursor que ya se encontraba abierto. Recuerde que un cursor de ciclo FOR automáticamente lo abre y ello no se debe especificar con la sentencia OPEN	-6511
DUP_VAL_ON_INDEX	El programa intentó almacenar valores duplicados en una columna que se mantiene con restricción de integridad de un índice único (unique index)	-1
INVALID_CURSOR	El programa intentó efectuar una operación no válida sobre un cursor	-1001
INVALID_NUMBER	En una sentencia SQL, la conversión de una cadena de caracteres hacia un número falla cuando esa cadena no representa un número válido	-1722
LOGIN_DENIED	El programa intentó conectarse a Oracle con un nombre de usuario o password inválido	-1017
NO_DATA_FOUND	Una sentencia SELECT INTO no devolvió valores o el programa referenció un elemento no inicializado en una tabla indexada	100
NOT_LOGGED_ON	El programa efectuó una llamada a Oracle sin estar conectado	-1012
PROGRAM_ERROR	PL/SQL tiene un problema interno	-6501

Lista de errores II

Excepcion	Se ejecuta ...	SQLCODE
ROWTYPE_MISMATCH	Los elementos de una asignación (el valor a asignar y la variable que lo contendrá) tienen tipos incompatibles. También se presenta este error cuando un parámetro pasado a un subprograma no es del tipo esperado	-6504
SELF_IS_NULL	El parámetro SELF (el primero que es pasado a un método MEMBER) es nulo	-30625
STORAGE_ERROR	La memoria se terminó o está corrupta	-6500
SUBSCRIPT_BEYOND_COUNT	El programa está tratando de referenciar un elemento de un arreglo indexado que se encuentra en una posición más grande que el número real de elementos de la colección	-6533
SUBSCRIPT_OUTSIDE_LIMIT	El programa está referenciando un elemento de un arreglo utilizando un número fuera del rango permitido (por ejemplo, el elemento "-1")	-6532
SYS_INVALID_ROWID	La conversión de una cadena de caracteres hacia un tipo rowid falló porque la cadena no representa un número	-1410
TIMEOUT_ON_RESOURCE	Se excedió el tiempo máximo de espera por un recurso en Oracle	-51
TOO_MANY_ROWS	Una sentencia SELECT INTO devuelve más de una fila	-1422
VALUE_ERROR	Ocurrió un error aritmético, de conversión o truncamiento. Por ejemplo, sucede cuando se intenta calzar un valor muy grande dentro de una variable más pequeña	-6502

Ejemplo

```
set serveroutput on;
declare
  person personas%rowtype;
  begin
  -- Si en la tabla hay mas de una fila, la sentencia devolverá mas
    de un resultado, provocando la excepción: too_many_rows.
   select * into person from personas;
    exception
    when too_many_rows then
      dbms_output.put_line('salta la excepcion');
      rollback;
end;
/
```

Excepciones de Usuario

- A parte de las excepciones definidas dentro de Oracle, el usuario puede declarar sus propias excepciones.
- Pero en este caso es el usuario el que las desencadena en caso de ser necesario utilizando la instrucción RAISE.
- Esquema:

```
Declare
        Nombre_error EXCEPTION;              -- Se declara.
Begin
        …
        If (anomalia)
                Then RAISE Nombre_error;     -- Ante una circunstancia se provoca.
EXCEPTION
        WHEN Nombre_error THEN               -- Se captura.
                (tratamiento)
END;
```

Provocar excepciones

```
DECLARE
 v_div NUMBER;
BEGIN
    SELECT 1/0 INTO v_div FROM DUAL;
EXCEPTION
 WHEN OTHERS THEN
 RAISE_APPLICATION_ERROR(-20001,'No se puede dividir por cero');
END;
-- La captura de la división por cero también se puede hacer así:
Begin
    …
Exception
    when divide_zero then
                -- Tratamiento
End;
```

Propagación de Excepciones

- A instrucción RAISE a parte de generar excepciones también se puede utilizar para provocar una excepción en un bloque interno y que sea capturada por otro bloque mas externo.
- Y cada bloque hacer una parte del tratamiento de la excepción.

Ejemplo

- Si un pedido tiene líneas asociadas se eliminan, así como el pedido.

```
Declare
-- Variable asociada a la col numped de la tabla pedidos.
Vnumped pedidos.numped%TYPE;
Begin
        Vnumped:=1;
        Declare
                Vnumero integer;
        Begin
                Select count(*) into vnumero from lineasped where numped=vnumped;
                If vnumero > 0 then
                        Raise too_many_rows;
                End if;
        Exception
                When too_many_rows then
                        Delete from lineasped where numped=vnumped;
                        Raise no_data_found;   -- Salta al bloque externo y captura la excepción, para continuar con el
                                               -- tratamiento
        End;
Exception
        When no_data_found then
                Delete pedidos where numped=vnumped;
End;
/
```

Visibilidad de Excepciones

- Es igual que en las variables, las declaraciones de un bloque externo son visibles para los subbloques.

- Cuando tenemos dos bloques anidados y declaramos una excepción en el bloque externo, podemos verla dentro del interno, a no ser que la volvamos a definir dentro del interno (**la enmascaramos**).

Visibilidad de Excepciones

```
Declare
    Sobrepasar Exception;
Begin
    …
        Declare
            Sobrepasar Exception;
        Begin
            …
        Exception
            …
        End;
Exception
…
End;
```

El sub-bloque no puede generar la excepción definida en el bloque de nivel superior, ya que está enmascarada por la definición local.
Se podría si el bloque tuviera una etiqueta utilizando esta sintaxis:
Nombre_bloque.nombre_excepcion

Ejemplo: Tratar varias excepciones

```
Set serveroutput on;
DECLARE
  num_row  number_table%ROWTYPE;
BEGIN
  select *
  into num_row
  from number_table;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
   dbms_output.put_line('No  hay datos!');
  WHEN TOO_MANY_ROWS THEN
   dbms_output.put_line('Demasiados  resultados!');
  WHEN OTHERS THEN
   dbms_output.put_line('Error');
END;
/
```