

4

Excepciones ORACLE

Bases de datos II
Universidad del Cauca
Ing. Wilson Ortega

Introducción

- Los errores en tiempo de ejecución de PL/SQL pueden provenir de fallas de diseño, errores de código, hardware, etc
- Es difícil anticiparse a todos los errores pero se pueden controlar las excepciones para permitir que el programa siga
- Lo ideal es evitar que mensajes de error confusos lleguen hasta el usuario.

Definición

- En PL/SQL un error en la ejecución es llamado excepción.
- Puede ser:
 - Definida internamente
 - ORA-22056 (value *string* is divided by zero)
 - Definida por el usuario
 - Fondos insuficientes para la transacción
- Cuando ocurre un error se lanza una excepción y se transfiere el control a la sección del bloque encargada de su manejo.
- Los nombres predefinidos de excepciones oracle están disponibles en :
http://docs.oracle.com/cd/B19306_01/appdev.102/b14261/errors.htm

Control de excepciones

Ejemplo

```
SET SERVEROUTPUT ON
DECLARE
v_bono NUMBER:= 10000;
v_num_empleados NUMBER := 0;
v_bono_empleado NUMBER;
BEGIN

    v_bono_empleado := v_bono / v_num_empleados;
    DBMS_OUTPUT.PUT_LINE('Bono por empleado: ' || v_bono_empleado);

EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE('Olvidó introducir el número de empleados');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Ocurrió un error. Contacte al administrador');
END;
```

Control de excepciones

Ejemplo

```
SET SERVEROUTPUT ON
DECLARE
v_nombre VARCHAR(50);
BEGIN
    SELECT nombre INTO v_nombre FROM programa
    WHERE programa_id = 234;
    DBMS_OUTPUT.PUT_LINE('Nombre programa: ' || v_nombre);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No existe el programa');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Ocurrió un error. Contacte al administrador');
END ;
```

Excepciones definidas por el usuario

- Se definen en la parte declarativa del bloque
 - Ej.
DEFINE
saldo_agotado EXCEPTION;
- Se deben lanzar manualmente usando
 - RAISE
 - DBMS_STANDARD.RAISE_APPLICATION_ERROR

Excepciones definidas por el usuario

Ejemplo

```
SET SERVEROUTPUT ON
DECLARE
  v_promedio NUMBER(2,1);
  est_reprobado EXCEPTION;
BEGIN
  SELECT promedio INTO v_promedio FROM estudiante WHERE codigo = 23;
  IF v_promedio < 3 THEN
    RAISE est_reprobado;
  END IF;
  DBMS_OUTPUT.PUT_LINE('Estudiante becado!');
EXCEPTION
  WHEN est_reprobado THEN
    DBMS_OUTPUT.PUT_LINE('No puede optar a beca debido al promedio');
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No existe el estudiante');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió un error. Contacte al administrador');
END ;
```

Usando EXCEPTION_INIT

- Es posible manejar excepciones que no tienen un nombre predefinido por oracle.
 - WHEN OTHERS
 - Pragma EXCEPTION_INIT
- pragma EXCEPTION_INIT le dice al compilador que asocie un nombre de excepción a un número de error de Oracle. De esta manera en el bloque de excepciones puede usar ese nombre despues del WHEN

Usando EXCEPTION_INIT

Ejemplo (1)

```
SET SERVEROUTPUT ON
DECLARE
  TYPE t_array_nombres IS VARRAY(5) OF VARCHAR2(50);
  v_nombres t_array_nombres := t_array_nombres();
BEGIN
  v_nombres(2) := 'Pepe';
END ;
```

Informe de error:

ORA-06533: Subscript beyond count

ORA-06512: at line 5

06533. 00000 - "Subscript beyond count"

***Cause:** An in-limit subscript was greater than the count of a varray
or too large for a nested table.

***Action:** Check the program logic and explicitly extend if necessary.

Usando EXCEPTION_INIT

Ejemplo (2)

```
SET SERVEROUTPUT ON
DECLARE
    TYPE t_array_nombres IS VARRAY(5) OF VARCHAR2(50);
    v_nombres t_array_nombres := t_array_nombres();
    INDICE_FUERA_RANGO exception;
    PRAGMA EXCEPTION_INIT(INDICE_FUERA_RANGO, -6533);
BEGIN
    v_nombres(2) := 'Pepe';
EXCEPTION
    WHEN INDICE_FUERA_RANGO THEN
        DBMS_OUTPUT.PUT_LINE('Está asignando un valor fuera de la capacidad del
array');
END ;
```

Definir mensajes de error propios

- `RAISE_APPLICATION_ERROR` permite definir un error `ORA-n` propio.
- Es útil para retornar ese número al programa que llama un procedimiento almacenado o función y así controlarlo.

- Sintaxis:

`Raise_application_error(numero, mensaje[, {TRUE | FALSE}]);`

Número: Un número negativo (-20000..-20999)

Mensaje: Mensaje a mostrar (Hasta 2048 bytes).

Definir mensajes de error propios

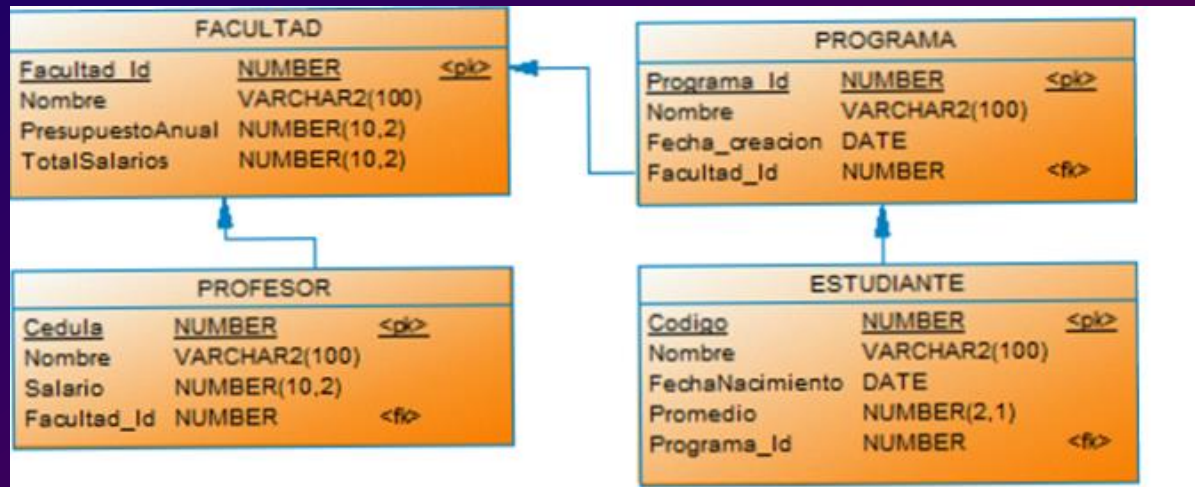
- Si el parámetro opcional es puesto en TRUE, el error se pone en la pila de las excepciones lanzadas anteriormente. Es útil si se quiere ver la traza de todas las excepciones que se lanzaron.
- Si es puesto en FALSE (por defecto), el error reemplaza los anteriores.

Definir mensajes de error propios

Ejemplo

```
SET SERVEROUTPUT ON
DECLARE
v_nombre VARCHAR(50);
v_prog_id NUMBER := &v;
BEGIN
    IF v_prog_id <= 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El id de programa es cero o negativo');
    END IF;
    SELECT nombre INTO v_nombre FROM programa
    WHERE programa_id = v_prog_id;
    DBMS_OUTPUT.PUT_LINE('Nombre programa: ' || v_nombre);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No existen datos para el id introducido');
        RAISE_APPLICATION_ERROR(-20002, 'No hay datos en la tabla para el id
introducido', true);
END ;
```

Ejercicios



- Cree un procedimiento almacenado que pida al usuario la cédula de un profesor y muestre su nombre y salario. Controle la excepción que se lanza si no existe la cédula en la tabla. En caso de que el profesor son tenga una facultad asociada, lance la excepción SIN_FACULTAD y muestre el mensaje "Sin facultad".
- Cree una función que pida al usuario el id de una facultad y retorne el número de profesores asociados. Si no tiene profesores asociados se lanza un código de error ORA-20002 con el mensaje "Facultad sin profesores".

Bibliografía

- **Oracle® Database PL/SQL Language Reference**
-11g Release 1 (11.1) - 2009