

Gestión de bases de datos

Excepciones

Introducción

Es una advertencia o condición de error. Se controlan dentro de su propio bloque. Si se produce un error, se ejecuta el bloque exception.

DECLARE

-- Declaraciones

BEGIN

-- Ejecución

EXCEPTION

-- Excepción

END;

Ejemplos

Ejemplo de excepciones NO_DATA_FOUND Y ZERO_DIVIDE. Cualquier otra excepción se captura en el bloque WHEN OTHERS.

```
DECLARE
    -- Declaraciones
BEGIN
    -- Ejecución
EXCEPTION
    WHEN NO_DATA_FOUND THEN - - se ejecuta si ocurre la excepción NO_DATA_FOUND
    WHEN ZERO_DIVIDE THEN - - se ejecuta si ocurre la excepción ZERO_DIVIDE
    WHEN OTHERS THEN /* se ejecuta si ocurre una excepción de un tipo no tratado en los anteriores */
END;
```

Ejecución exception

Cuando ocurre un error, se ejecuta el bloque EXCEPTION y una vez finalizada la ejecución del bloque EXCEPTION no se continua ejecutando el bloque anterior.

Si no existe un bloque de excepción apropiado, se ejecutará el bloque WHEN OTHERS (si existe) que debe ser el último manejador de excepciones.

Existen **excepciones predefinidas** en el sistema, pero también se puede definir desde cero con la sentencia **RAISE**.

Ejecución predefinidas

NOMBRE EXCEPCIÓN	NÚMERO DE ERROR	DESCRIPCIÓN
ACCESS_INTO_NULL	ORA-06530	Se intentan asignar valores a un objeto que no se había inicializado
CASE_NOT_FOUND	ORA-06592	Ninguna opción WHEN dentro de la instrucción CASE captura el valor, y no hay instrucción ELSE.
COLLECTION_IS_NULL	ORA-06531	Se intenta utilizar un varray o una tabla anidada que no estaba inicializada
CURSOR_ALREADY_OPEN	ORA-06511	Se intenta abrir un cursor que ya se había abierto
DUP_VAL_ON_INDEX	ORA-00001	Se intentó añadir una fila que provoca que un índice único repita valores
INVALID_CURSOR	ORA-01001	Se realizó una operación ilegal sobre un cursor

Ejecución predefinidas

NOMBRE EXCEPCIÓN	NÚMERO DE ERROR	DESCRIPCIÓN
INVALID_NUMBER	ORA-01722	Falla la conversión de carácter a número
LOGIN_DENIED	ORA-01017	Se intenta conectar con Oracle usando un nombre de usuario y contraseña inválidos
NO_DATA_FOUND	ORA-01403	SELECT de fila única no devolvió valores
PROGRAM_ERROR	ORA-06501	Error interno de Oracle
ROWTYPE_MISMATCH	ORA-06504	Hay incompatibilidad de tipos entre el cursor y las variables a las que se intentan asignar sus valores
STORAGE_ERROR	ORA-06500	No hay memoria suficiente

Ejecución predefinidas

NOMBRE EXCEPCIÓN	NÚMERO DE ERROR	DESCRIPCIÓN
SUBSCRIPT_BEYOND_COUNT	ORA-06533	Se hace referencia a un elemento de un varray o una tabla anidada usando un índice mayor que los elementos que poseen
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	Se hace referencia a un elemento de un varray o una tabla anidada usando un índice cuyo valor está fuera del rango legal
SYS_INVALID_ROWID	ORA-01410	Se convierte un texto en un número de identificación de fila (ROWID) y el texto no es válido
TIMEOUT_ON_RESOURCE	ORA-06501	Error interno de Oracle
ROWTYPE_MISMATCH	ORA-00051	Se consumió el máximo tiempo en el que Oracle permite esperar al recurso
TOO_MANY_ROWS	ORA-01422	SELECT de fila única devuelve más de una fila
VALUE_ERROR	ORA-06502	Hay un error aritmético, de conversión, de redondeo o de tamaño en una operación
ZERO_DIVIDE	ORA-01476	Se intenta dividir entre el número cero.

Ejemplo

```
DECLARE
  x NUMBER :=0;
  y NUMBER := 3;
  res NUMBER;
BEGIN
  res:=y/x;
  DBMS_OUTPUT.PUT_LINE(res);
EXCEPTION
  WHEN ZERO_DIVIDE THEN
    DBMS_OUTPUT.PUT_LINE('No se puede dividir por cero') ;
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error inesperado') ;
END;
```


SQLCODE Y SQLERRM

Funciones predefinidas de las excepciones. Son muy útiles para aclarar el significado de la excepción OTHERS.

- ❑ **SQLCODE**: Devuelve el código del error producido.
- ❑ **SQLERRM**: Devuelve el mensaje de Oracle asociado al número de error.

Ejemplo:

```
EXCEPTION
```

```
.....
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' || SQLCODE || 'mensaje: ' || SQLERRM);
```

```
END;
```

Ejercicio 1

Escribir un bloque que divida un número entre cero. Capturar la excepción WHEN OTHERS y mostrar por pantalla el código y mensaje de error.

EXCEPCIONES DE USUARIO. RAISE

Los usuarios pueden lanzar sus propias excepciones.

Para ello:

1. Declarar un nombre para la excepción en DECLARE
miExcepcion EXCEPTION;
2. En BEGIN utilizar la instrucción RAISE para lanzar la excepción:
RAISE miExcepcion;
3. En EXCEPTION capturar el nombre de la excepción
EXCEPTION
...
when miExcepcion THEN

EJEMPLO

Escribir un bloque que si un número es menor que cero, lance una excepción: VALOR_NEGATIVO.

```
DECLARE
    VALOR_NEGATIVO EXCEPTION;
    valor NUMBER;
BEGIN
    valor := -1;
    IF valor < 0 THEN
        RAISE VALOR_NEGATIVO;
    END IF;
    EXCEPTION
        WHEN VALOR_NEGATIVO THEN
            dbms_output.put_line('El valor no puede ser negativo');
END;
```

Ejercicio 2

Sobre la tabla Estudiantes. Escribir un bloque que muestre el número total de estudiantes y lanzar una excepción `NO_ALUMNOS` si el número es igual a 0, indicando con un mensaje que no hay alumnos.

Ejercicio 3

Incorporar una nueva excepción MUCHOS_ALUMNOS, si el total de alumnos es mayor o igual que 5.

RAISE APPLICATION ERROR

Otra forma de lanzar excepciones propias es utilizar la función `RAISE_APPLICATION_ERROR` que simplifica los tres pasos anteriores:

```
RAISE_APPLICATION_ERROR(Nº ERROR, MENSAJE, {TRUE|FALSE});
```

Lanza un error cuyo número debe estar entre -20000 y -20999 y muestra el mensaje indicado. El tercer parámetro es opcional, por defecto `TRUE` e indica si el error se añade a la pila de errores existentes.

RAISE APPLICATION ERROR EJEMPLO

```
BEGIN
  DELETE FROM estudiantes WHERE codigo=8;
  IF SQL%NOTFOUND THEN
    RAISE_APPLICATION_ERROR(-20001,'No existe el
estudiante');
  END IF;
END;
/
```




CEU

*Centro de Estudios
Profesionales*

Fundación San Pablo Andalucía