



1. Excepcions

NF3. Excepcions, cursors i transaccions

UF3 - Llenguatges SQL: DCL i extensió procedimental

Desenvolupament d'Aplicacions Multiplataforma

M02 – Bases de dades. Versió 1.0

© M^a Carmen Brito Ruiz



1.1. Excepciones

1.1.1. Tipos de excepciones

1.2. Ejemplos de excepciones

1.1. Excepciones

Las excepciones se usan para tratar errores en tiempo de ejecución, ya sean errores como situaciones definidas por el usuario. El formato de una excepción es el siguiente:

```
...  
EXCEPTION  
  
WHEN <nombre_excepción1> THEN <instrucciones1>;  
WHEN <nombre_excepción2> THEN <instrucciones2>;  
  
...  
  
WHEN <nombre_excepciónN> THEN <instruccionesN>;  
[WHEN OTHERS THEN <instrucciones>;]  
  
END <nombre_programa>;
```

1.1.1. Tipos de excepciones

➤ Excepciones internas predefinidas

Este tipo de excepciones están predefinidas por Oracle y se disparan automáticamente al producirse determinados errores. Este tipo de excepciones no se han de declarar en la sección DECLARE.

Excepcions

Código error Oracle	SQL CODE	Exception	Se disparan cuando...
ORA-010403	+100	NO_DATA_FOUND	Una sentencia SELECT...INTO no devuelve ninguna fila.
ORA-01422	-1422	TOO_MANY_ROWS	Una sentencia SELECT...INTO no devuelve más de una fila.
ORA-00001	-1	DUP_VAL_ON_INDEX	Intentar almacenar un valor que estaba duplicado en el campo clave o en una columna con restricción UNIQUE.
ORA-06530	-6530	ACCESS_INTO_NULL	Intentar acceder a atributos de un objeto no inicializado
ORA-06531	-6531	COLLECTION_IS_NULL	Intentar acceder a elementos de una colección que no ha sido inicializada
ORA-06511	-6511	CURSOR_ALREADY_OPEN	Intentar abrir un cursor que ya está abierto.
ORA-01001	-1001	INVALID_CURSOR	Intentar realizar una operación no permitida sobre un cursor, como puede ser cerrar un cursor que no está abierto.
ORA-01722	-1722	INVALID_NUMBER	Fallo al intentar convertir una cadena a un valor numérico.



Excepcions

Código error Oracle	SQL CODE	Exception	Se disparan cuando...
ORA-01017	-1017	LOGIN_DENIED	Intentar conectar a Oracle con un usuario o clave incorrecta
ORA-01012	-1012	NOT_LOGGED_ON	Intentar acceder a la base de datos sin estar conectado a Oracle
ORA-06501	-6501	PROGRAM_ERROR	Hay un problema interno en la ejecución del programa
ORA-06504	-6504	ROWTYPE_MISMATCH	La variable del cursor del HOST y la variable del cursor PL/SQL pertenecen a tipos incompatibles
ORA-06533	-6533	SUB_SCRIPT_OUTSIDE_LIMIT	Intentar acceder a una tabla anidada o a un array con un valor de índice ilegal
ORA-06500	-6500	STORAGE_ERROR	El bloque PL/SQL se ejecuta fuera de memoria
ORA-00051	-51	TIMEOUT_ON_RESOURCE	Se excede el tiempo de espera para un recurso
ORA-06502	-6502	VALUE_ERROR	Un error de tipo aritmético, de conversión, de truncamiento, etc.
ORA-01476	-1476	ZERO_DIVIDE	Se intenta la división entre cero

Desenvolupament d'Aplicacions Multiplataforma – M02 Bases de dades

UF3: Llenguatges SQL: DCL i extensió procedimental - NF3: Excepcions, cursors i transaccions -

EA 3.3.1. Excepcions

Versió 1.0 - © M^a Carmen Brito

➤ Excepciones definidas por el usuario.

Son las excepciones que el propio usuario (programador) define y programa. Para su uso hay que dar los tres pasos siguientes:

- a) Se han de declarar en la sección DECLARE de la siguiente forma:

```
<nombre_excepción> EXCEPTION;
```

- b) Se disparan con la orden RAISE en la sección ejecutable del programa:

```
RAISE <nombre_excepción>;
```

- c) Tratamiento de la excepción, teniendo en cuenta el siguiente formato:

```
WHEN <nombre_excepción> THEN <tratamiento>;
```

➤ Otras excepciones

Son errores internos que no tienen asignada una excepción, sino un código y un mensaje de error, a los que se accede con las funciones `SQLCODE` y `SQLERRM`. El tratamiento es el mismo que en el resto de excepciones, cuando se produce el error se transfiere el control a la sección `EXCEPTION` y se tratará el error en la cláusula `WHEN OTHERS`.

No se pueden acceder desde órdenes SQL *Plus a las funciones `SQLCODE` y `SQLERRM`, pero se pueden pasar a este entorno a través de algunas soluciones como puede ser la siguiente:

```
...  
WHEN OTHERS THEN  
    :codigo_error:=SQLCODE;  
    :mensaje_error:=SQLERRM;  
END;
```


1.2. Ejemplos de excepciones

Ejemplo 1: En este ejemplo se trabaja con excepciones internas predefinidas por el propio DBMS, en este caso, trabajamos con Oracle.

```
SET VERIFY OFF
SET ECHO OFF
SET SERVEROUTPUT ON
DECLARE
    var_empno employees.employee_id%TYPE;
BEGIN
    SELECT employee_id
    INTO var_empno
    FROM employees;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('ERROR: no se han
encontrado datos');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE ('ERROR: devuelve más de una
fila');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('ERROR!!! (Sin definir)');
END;
/
SET VERIFY ON
SET ECHO ON
```

Ejecución del ejemplo 1:

- Si no hay datos en la tabla:

```
bloque anónimo terminado  
ERROR: no se han encontrado datos
```

- Si la consulta devuelve más de una fila:

```
bloque anónimo terminado  
ERROR: devuelve más de una fila
```

Ejemplo 2: Comprueba si un empleado existe en la tabla, consultando por el código del empleado.

El programa ha de dar los siguientes resultados:

- Si existe el empleado:

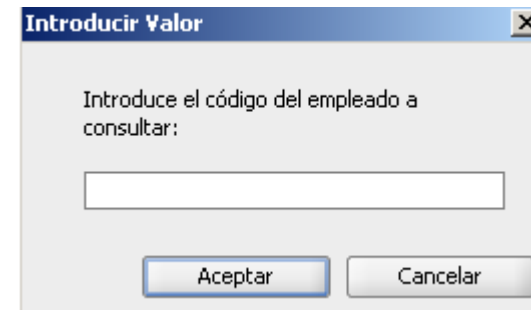
```
bloque anónimo terminado
El empleado con código 100 se llama Steven
```

- Si no existe el empleado:

```
bloque anónimo terminado
ERROR: no se han encontrado datos
```

- Si la consulta devuelve más de una fila (aunque en este caso no es posible, pues se consulta por el campo clave):

```
bloque anónimo terminado
ERROR: devuelve más de una fila
```



```
SET VERIFY OFF
SET ECHO OFF
SET SERVEROUTPUT ON
ACCEPT var PROMPT 'Introduce el código del empleado a consultar:'
DECLARE
  var_ename employees.first_name%TYPE;

BEGIN
  SELECT first_name
  INTO var_ename
  FROM employees
  WHERE employee_id = &var;
  DBMS_OUTPUT.PUT_LINE ('El empleado con código '||&var||' se llama
  '||var_ename);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('ERROR: no se han encontrado datos');
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE ('ERROR: devuelve más de una fila');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE ('ERROR!!! (Sin definir)');
END;
/
SET VERIFY ON
SET ECHO ON
```

Ejemplo 3: Comprueba si un empleado existe en la tabla, consultando por el código del empleado. En caso de existir el empleado, nos indica si el nombre del empleado empieza por la letra S o no.

- Si existe el empleado y además empieza el nombre por S:

```
bloque anónimo terminado
```

```
EL EMPLEADO 100 existe
```

```
La primera posición de su nombre sí que es una S, se llama Steven
```

- Si existe el empleado, pero no empieza por S:

```
bloque anónimo terminado
```

```
EL EMPLEADO 101 existe
```

```
ERROR:la primera posición no es una S
```

- Si no existe el empleado: bloque anónimo terminado
 No hay filas

```
-- Declaración de variables y de excepciones

SET VERIFY OFF
SET ECHO OFF
SET SERVEROUTPUT ON
ACCEPT var PROMPT 'QUE EMPLEADO DESEAS CONSULTAR: '
DECLARE

    codigo_error NUMBER(6);          -- controla el error devuelto por oracle
    var_empno employees.employee_id%TYPE;
    var_ename employees.first_name%TYPE;
    no_contiene_S EXCEPTION;
    no_hay_filas EXCEPTION;
    PRAGMA EXCEPTION_INIT(no_hay_filas, +100);
```

```
-- Inicio del programa: se consulta y se imprime si existe el socio y se  
-- comprueba si la primera posición del nombre es una S. En caso de que no  
-- sea, se llama a la excepción programada por el programador.
```

```
BEGIN
```

```
    SELECT employee_id, first_name
```

```
    INTO var_empno, var_ename
```

```
    FROM employees
```

```
    WHERE employee_id=&var;
```

```
    DBMS_OUTPUT.PUT_LINE('EL EMPLEADO ' || var_empno || ' existe');
```

```
    IF SUBSTR(var_ename,1,1)<>'S' THEN
```

```
        RAISE no_contiene_S;
```

```
    ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('La primera posición de su nombre sí que  
es una S, se llama ' || var_ename);
```

```
    END IF;
```

```
EXCEPTION          -- programación de las excepciones
  WHEN no_contiene_S THEN
    DBMS_OUTPUT.PUT_LINE('ERROR:la primera posición no es una
S');
  WHEN no_hay_filas THEN
    DBMS_OUTPUT.PUT_LINE ('No hay filas');
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE ('DEVUELVE MAS DE UNA FILA');
  WHEN OTHERS THEN
    codigo_error:=SQLCODE;
    DBMS_OUTPUT.PUT_LINE ('ERROR!!!!!!!!!!' || codigo_error);
END;

/

SET ECHO ON

SET VERIFY ON
```

Desenvolupament d'Aplicacions Multiplataforma – M02 Bases de dades

UF3: Llenguatges SQL: DCL i extensió procedimental - NF3: Excepcions, cursors i transaccions -

EA 3.3.1. Excepcions

Versió 1.0 - © M^a Carmen Brito

Preguntes!!!!

