### GESTIÓN DE EXCEPCIONES

La ejecución de una aplicación finaliza al presentarse algún error pero con la definición de manejadores de errores es posible hacer una serie de acciones y continuar con la ejecución de la aplicación. El tratamiento de errores es muy importante en los sistemas de bases de datos con el fin de responder a cualquier problema que ocurra en la ejecución de cualquier aplicación.

Para el tratamiento de errores PL/SQL tiene una sección denominada EXCEPTION donde se realiza el tratamiento y control de errores. Cuando un proceso detecta un error pasa el control a la zona EXCEPTION y busca automáticamente si existe tratamiento a la excepción del error detectado.

Existen dos métodos para provocar una excepción

1. Cuando se produce un error de Oracle se genera automáticamente la excepción asignada. Por ejemplo, el error ORA 01001 indica que se ha producido una operación ilegal con un cursor. En este caso PL provoca la excepción INVALID\_CURSOR.
2. Provocar explícitamente una excepción, es decir que el usuario sea el que la predefina.

Si el bloque no tiene área de excepciones este se interrumpe dando el servidor los mensajes pertinentes y que tenga. Si tiene área de excepciones el error se propaga a dicha área, enviando el mensaje que considere oportuno.

Sintaxis:

EXCEPTION

WHEN EXCEP1 [ OR EXCEP2 ... ] THEN

ORDENES;

[WHEN EXCEP3 [ OR EXCEP 4 …] THEN

ORDENES;

[WHEN OTHERS] THEN

ORDENES;

…]

Donde:

**EXCEP1, EXCEP2,...**: Son los nombres de las excepciones estándar o definidas por el usuario

**OTHERS**: Es la que irrumpe en cualquier otro caso (como la clase EXCEPTION en Java o NET)

Las excepciones pueden ser:

**Implícitas**: Éstas pueden ser predefinidas o no predefinidas por el servidor Oracle.

**Explicitas**: Definidas por el usuario.

### EXCEPCIONES INTERNAS DE PL / SQL (PREDEFINIDAS POR EL SERVIDOR)

Este tipo de excepciones se disparan ante un determinado error detectado por PL/SQL. No hay que declararlas.

Disponemos de las siguientes:

|  |  |  |
| --- | --- | --- |
| ***Excepciones predefinidas de Oracle*** | | |
| Error Oracle | Excepción **Equivalente** | Descripción |
| ORA – 0001 | DUP\_VAL\_ON\_INDEX | Violación de una restricción de unidad |
| ORA – 0051 | TIMEOUT\_ON\_RESOURCE | Se produjo un fin de intervalo mientras se esperaba un cierto recurso |
| ORA – 0061 | TRANSACTION\_BACKED\_OUT | La transacción fue cancelada debido a un bloqueo |
| ORA – 1001 | INVALID\_CURSOR | Operación ilegal con un cursor |
| ORA – 1012 | NOT\_LOGGED\_ON | No existe conexión con Oracle |
| ORA – 1017 | LOGIN\_DENIED | Nombre de usuario o contraseña inválidos |
| ORA – 1403 | NO\_DATA\_FOUND | No se ha encontrado ningún dato |
| ORA – 1422 | TOO\_MANY\_ROWS | Hay más de una línea que corresponde a una orden SELECT… INTO |
| ORA – 1476 | ZERO\_DIVIDE | División por cero |
| ORA – 1722 | INVALID\_NUMBER | Falló la conversión a un número |
| ORA – 6500 | STORAGE\_ERROR | Error interno PL/SQL, generado cuando PL/SQL se queda sin memoria |
|  |  |  |
| ORA – 6501 | PROGRAM\_ERROR | Error interno PL/SQL |
| ORA – 6502 | VALUE\_ERROR | Error de truncamiento aritmético o de conversión |
| ORA – 6504 | ROWTYPE\_MISMATCH | Una variable de cursor del HOST y una variable del cursor PL/SQL tienen tipo de fila incompatibles |
| ORA – 6511 | CURSOR\_ALREADY\_OPEN | Se ha intentado abrir un cursor que ya estaba abierto |
| ORA – 6530 | ACCESS\_INTO\_NULL \* | Se ha intentado asignar valores a los atributos de un objeto que tiene el valor NULL |
| ORA – 6531 | COLLECTION\_IS\_NULL \* | Se ha intentado aplicar métodos de colección distintos de EXISTS a una tabla o array PL/SQL con valor NULL |
| ORA – 6532 | SUBSCRIPT\_OUTSIDE\_LIMIT \* | Una referencia a una tabla anidada o índice de arrray se encuentra fuera del rango declarado |
| ORA – 6533 | SUBSCRIPT\_BEYOND\_COUNT \* | Una referencia a una tabla animada o índice de array es mayor que el número de elementos de la colección. |

EJEMPLO:

Hacer un bloque anónimo, que introduciendo dos números por teclado divida n1/n2.

DECLARE

NUM1 NUMBER:=&n1;

NUM2 NUMBER :=&n2;

COCIENTE NUMBER;

BEGIN

COCIENTE:= NUM1/NUM2;

DBMS\_OUTPUT.PUT\_LINE('COCIENTE:'||COCIENTE);

EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('IMPOSIBLE DIVIDIR POR 0');

END;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

RESULTADO:

COCIENTE:5

Si ejecutamos dividiendo por cero:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

RESULTADO:

IMPOSIBLE DIVIDIR POR 0

### EXCEPCIONES DEFINIDAS POR EL USUARIO

Para poder crear una excepción definida a nivel de usuario hay que declararla en la zona de declaraciones.

Sintaxis:

NB\_EXCEPCION EXCEPTION;

**NB\_EXCEPCION** es el nombre de la excepción.

Para arrancar una excepción de usuario y pasar el control a la sección de excepciones se utiliza la orden RAISE seguida del nombre de la excepción.

EJEMPLO

Definir un cursor que seleccione el apellido y comisión de los empleados ordenado por comisión de la tabla emp. Insertar en otra tabla los registros que no tengan comisión nula.

1.- Crear la tabla PRUEBAEXCEPCIONES.

CREATE TABLE PRUEBAEXCEPCIONES (APE VARCHAR2(15),COM NUMBER);

2.- Listamos el apellido y comisión de los empleados para comprobar cuantos tienen comisión nula.

SELECT APELLIDO, COMISION FROM EMP ORDER BY 2 ASC;

Tabla

El contenido generado por IA puede ser incorrecto.

Vemos que solo los seis primeros tienen comisión

3.- Creamos el bloque anónimo para que inserte en la tabla recién creada los empleados con comisión. Cuando encentre un nulo cortará la inserción ya que provocaremos una excepción.

DECLARE

CURSOR C1 IS

SELECT APELLIDO, COMISION FROM EMP ORDER BY 2 ASC;

V\_APE EMP.APELLIDO%TYPE;

V\_COM EMP.COMISION%TYPE;

CONT NUMBER:=0;

ERROR\_COMIS EXCEPTION;

BEGIN

OPEN C1;

FETCH C1 INTO V\_APE, V\_COM;

WHILE C1%FOUND LOOP

IF V\_COM IS NULL THEN

RAISE ERROR\_COMIS;

END IF;

CONT:=C1%ROWCOUNT;

INSERT INTO PRUEBAEXCEPCIONES VALUES (V\_APE,V\_COM);

COMMIT;

FETCH C1 INTO V\_APE, V\_COM;

END LOOP;

EXCEPTION

WHEN ERROR\_COMIS THEN

DBMS\_OUTPUT.PUT\_LINE (CONT ||' FILAS INSERTADAS');

END;

RESULTADO:

6 FILAS INSERTADAS

4.- Comprobamos que ha insertado únicamente los registros con comisión no nula.

Tabla

El contenido generado por IA puede ser incorrecto.

### EXCEPCIONES ASOCIADAS A ERRORES DE ORACLE. (NO PREDEFINIDOS POR EL SERVIDOR)

Estas se definen basándose en el número de error. Supongamos que queremos gestionar una excepción para comprobar si en un momento se intenta asignar un valor nulo en una columna que no lo permite. Este tipo de error devuelve el código -1400 para asociarlo a una excepción, utilizaremos la orden:

**PRAGMA EXCEPTION\_INIT** (nb\_excepcion, numero de error de oracle);

La excepción tiene que declararse primero y luego asociarle un error.

EJEMPLO

Realizamos un bloque anónimo sin control de errores para comprobar el error que provoca.

Estamos intentando introducir un nulo en la columna DNOMBRE y no permite nulos. Error=-1400

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Asociamos el error con pragma:

DECLARE

ERROR\_NULOS EXCEPTION;

PRAGMA EXCEPTION\_INIT(ERROR\_NULOS,-1400);

BEGIN

INSERT INTO DEPT VALUES (88,NULL,'MADRID');

COMMIT;

EXCEPTION

WHEN ERROR\_NULOS THEN

DBMS\_OUTPUT.PUT\_LINE('UNA COLUMNA NO ADMITE NULOS');

END;

RESULTADO:

UNA COLUMNA NO ADMITE NULOS

Ejemplo si el valor es más largo que lo admitido en la columna:

DECLARE

ERROR\_NULOS EXCEPTION;

PRAGMA EXCEPTION\_INIT(ERROR\_NULOS,-1400);

ERROR\_NUMEROS EXCEPTION;

PRAGMA EXCEPTION\_INIT(ERROR\_NUMEROS,-1438);

BEGIN

INSERT INTO DEPT VALUES (88333,NULL,'MADRID');

COMMIT;

EXCEPTION

WHEN ERROR\_NULOS THEN

DBMS\_OUTPUT.PUT\_LINE('UNA COLUMNA NO ADMITE NULOS');

WHEN ERROR\_NUMEROS THEN

DBMS\_OUTPUT.PUT\_LINE('NÚMERO DEMASIADO LARGO PARA LA COLUMNA');

END;

RESULTADO:

NÚMERO DEMASIADO LARGO PARA LA COLUMNA

### FUNCIONES ASOCIADAS CON EXCEPCIONES

Cuando se produce una excepción se puede identificar el código de error asociado o el mensaje mediante dos funciones:

**- SQLCODE:** Devuelve el número de error de Oracle de las excepciones internas. Se le asigna a una variable number. Vale 0 si no encuentra ninguna excepción, 1 si la excepción es indefinida por el usuario, 100 si es del tipo NO\_DATA\_FOUND y un número negativo si es error del servidor.

**- SQLERRM**: Devuelve datos carácter que contienen el mensaje de error asociado.

Ejemplo:

Declare

codigo dept.dept\_no%type;

begin

select dept\_no into codigo

from dept

where dnombre = 'VENTAS6';

dbms\_output.put\_line ('El departamento de Ventas tiene codigo: '||codigo);

exception

when TOO\_MANY\_ROWS then

dbms\_output.put\_line ('Coinciden en nombre varios departamentos');

when OTHERS then

dbms\_output.put\_line (to\_char(SQLCODE) || ' ' || SQLERRM);

end;

RESULTADO:

100 ORA-01403: no data found

### UTILIZACIÓN DE RAISE\_APPLICATION\_ERROR

Existe una función predefinida RAISE\_APPLICATION\_ERROR con la que se pueden crear mensajes de error y definir y enviar mensajes de error.

Sintaxis:

RAISE\_APPLICATION\_ERROR(Numero de error, Mensaje de error)

Número de error. Parámetro de la función cuyo valor debe encontrarse entre –20.000 y –20.999.

Mensaje de error. Parámetro de la función que es el mensaje asociado al error (Máximo 512 bytes).

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.