ATRIBUTOS PARA MANEJO DE ERRORES

Oracle, implícitamente, abre un cursor cada vez que procesa una sentencia SQL que no esté asociada a un cursor explícito.

Este cursor implícito creado por Oracle se llama ***SQL*** y dispone también de algunos atributos que pueden ser de gran utilidad para conocer cierta información de la ejecución de las sentencias SELECT INTO, INSERT, UPDATE Y DELETE.

El valor de estos atributos siempre se referirá a la última orden SQL ejecutada:

***SQL%NOTFOUND*:** Devuelve verdadero si la última orden ha fallado.

***SQL%FOUND*:** Devuelve verdadero si la última orden se ha ejecutado con éxito.

***SQL%ROWCOUNT*:** Devuelve el número de filas afectadas por la última orden SQL.

EJEMPLO

```
DECLARE
. . .

BEGIN
 . . .
UPDATE EMPLEADOS SET dpto=12 WHERE nombre='Carlos Arias';

IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('Se ha producido un error durante la actualización');
END IF;
. . .
END;
```

MANEJO DE ERRORES

En PL/SQL una advertencia o condición de error es llamada una *excepción*. Estas pueden ser definidas en forma interna (en tiempo de ejecución de un programa) o explícitamente por el usuario. Ejemplos de excepciones definidas en forma interna son la división por cero y la falta de memoria en tiempo de ejecución. Estas mismas condiciones excepcionales tienen sus nombres propios y pueden ser referenciadas con ellos: *zero_divide* y *storage_error*.

También se pueden definir excepciones a medida y nombrarlas de alguna forma. Cuando ocurre un error se alcanza la excepción, esto quiere decir que se ejecuta la porción del programa donde ésta se encuentra implementada, transfiriéndose el control a ese bloque de sentencias. Las excepciones definidas por el usuario deben ser alcanzadas explícitamente utilizando la sentencia *raise*.

Con las excepciones ser pueden manejar los errores cómodamente sin necesidad de mantener múltiples chequeos por cada sentencia escrita. También provee claridad en el código desde el momento en que permite mantener las rutinas correspondientes al tratamiento de los errores en forma separada de la lógica del negocio.

Excepciones predefinidas

Las excepciones predefinidas no necesitan ser declaradas. Simplemente se utilizan cuando estas son gatilladas por algún error determinado.

La siguiente es la lista de las excepciones predeterminadas por PL/SQL y una breve descripción de cuándo son accionadas:

Nombre Excepción	Gatillada cuando…
ACCESS_INTO_NULL	El programa intentó asignar valores a los atributos de un objeto no inicializado
COLLECTION_IS_NULL	El programa intentó asignar valores a una tabla anidada aún no inicializada
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
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)
INVALID_CURSOR	El programa intentó efectuar una operación no válida sobre un cursor
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
LOGIN_DENIED	El programa intentó conectarse a Oracle con un nombre de usuario o password inválido
NO_DATA_FOUND	Una sentencia SELECT INTO no devolvió valores o el programa referenció un elemento no inicializado en una tabla indexada
NOT_LOGGED_ON	El programa efectuó una llamada a Oracle sin estar conectado
PROGRAM_ERROR	PL/SQL tiene un problema interno
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
STORAGE_ERROR	La memoria se terminó o está corrupta
SUBSCRIPT_OUTSIDE_LIMIT	El programa está referenciando un elemento de un arreglo utilizando un número fuera del rango permitido (por ejemplo, el elemento "-1")

TIMEOUT_ON_RESOURCE	Se excedió el tiempo máximo de espera por un recurso en Oracle
TOO_MANY_ROWS	Una sentencia SELECT INTO devuelve más de una fila
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
ZERO_DIVIDE	El programa intentó efectuar una división por cero

Excepciones definidas por el usuario

PL/SQL permite al usuario definir sus propias excepciones, las que deberán ser declaradas y gatilladas explícitamente utilizando otros comandos del lenguaje.

Declaración

Las excepciones sólo pueden ser declaradas en el segmento "Declare" de un bloque, subprograma o paquete. Se declara una excepción escribiendo su nombre seguida de la palabra clave EXCEPTION. Las declaraciones son similares a las de variables, pero recuerde que una excepción es una condición de error, no un ítem de datos. Aun así, las mismas reglas de alcance aplican tanto sobre variables como sobre las excepciones.

Ejemplo:

```
DECLARE
  error_01    EXCEPTION;
```

Reglas de Alcance

Una excepción no puede ser declarada dos veces en un mismo bloque. Tal como las variables, una excepción declarada en un bloque es local a ese bloque y global a todos los sub-bloques que comprende.

La sentencia "RAISE"

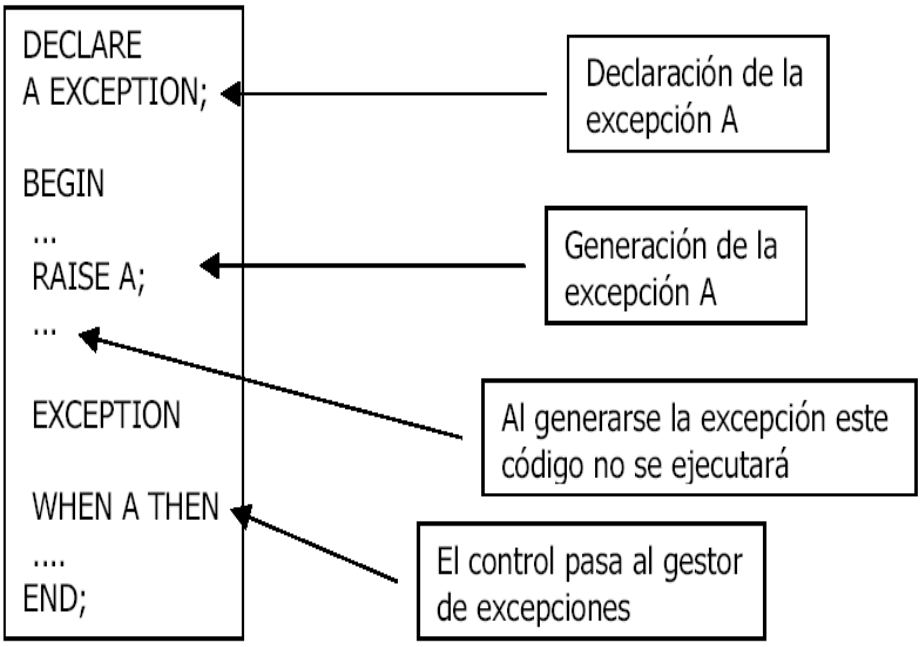
La sentencia RAISE permite gatillar una excepción en forma explícita. Es factible utilizar esta sentencia en cualquier lugar que se encuentre dentro del alcance de la excepción.

Ejemplo:

```
DECLARE
  Fuera_de_rango   EXCEPTION;          -- declaración de la excepción
  total            NUMBER(4);
BEGIN
  …
  IF total < 1 THEN
       RAISE Fuera_de_rango;                  -- llamado a la excepción
  END IF;
EXCEPTION
  WHEN Fuera_de_rango THEN
       -- manejar el error aquí
  WHEN OTHERS THEN
       …
END;
```

Finalmente, cabe destacar la existencia de la excepción OTHERS, que simboliza cualquier condición de excepción que no ha sido declarada. Se utiliza comúnmente al final del bloque de excepciones para absorber cualquier tipo de error que no ha sido previsto por el programador.

Resumen



Uso de SQLCODE y SQLERRM

Al manejar una excepción es posible apoyarse con las funciones predefinidas SQLCode y SQLErrm para aclarar al usuario la situación de error acontecida.

Sqlcode siempre retornará el número del error de Oracle y un "0" (cero) en caso exitoso al ejecutarse una sentencia SQL.

Por otra parte, *Sqlerrm* retornará el correspondiente mensaje de error para la situación ocurrida. También es posible entregarle a la función SQLERRM un número negativo que represente un error de Oracle y ésta devolverá el mensaje asociado.

Estas funciones son muy útiles cuando se utilizan en el bloque de excepciones, para aclarar el significado de la excepción OTHERS, cuando ésta ocurre.

Estas funciones no pueden ser utilizadas directamente en una sentencia SQL, pero sí se puede asignar su valor a alguna variable de programa y luego usar esta última en alguna sentencia.

Ejemplo:

```
DECLARE
  err_num     NUMBER;
  err_msg     VARCHAR2(100);
BEGIN
  …
EXCEPTION
  WHEN OTHERS THEN
       err_num := SQLCODE;
       err_msg := SUBSTR(SQLERRM, 1, 100);
       INSERT INTO errores VALUES(err_num, err_msg);
END;
```

EJEMPLO Manejo de Errores definidos por el usuario

CLASES

Varchar	Varchar	number (3)	number(3)
depto	**curso**	**Num_alumnos**	**Max_alumnos**
HIS	101	41	40
COMP	307	34	40

```
Declare
Muchos_alumnos exception;
Numero_alumnos number(3);
Maximo_alumnos number(3);

Begin
Select num_alumnos, max_alumnos
into numero_alumnos, maximo_alumnos from clases
Where depto= 'HIS' and curso= '101';

If numero_alumnos > maximo_alumnos then
Raise muchos_alumnos;
End if;

Exception
When muchos_alumnos then
Dbms_output.put_line ('el cupo máximo del curso esta sobrepasado');
End;
```