



MANEJO DE EXCEPCIONES EN PL/SQL

En la mayoría de los lenguajes de programación se emplean EXCEPCIONES para tratar errores en tiempo de ejecución. En Oracle también podemos definir excepciones. Estas definiciones hacen que, cuando se produce un error, salte una excepción y pase el control a la sección excepción correspondiente. Las acciones a seguir se incluyen en la sección EXCEPTION definida al final del bloque BEGIN-END.

```
BEGIN  
  
...  
  
EXCEPTION  
    WHEN <nombre_excepción> THEN  
        <bloque de sentencias>;  
    WHEN <nombre_excepción> THEN  
        <bloque de sentencias>;  
  
...  
    [WHEN OTHERS THEN <bloque de sentencias>;]  
END;
```

Existen dos tipos de excepciones:

- las excepciones predefinidas y
- las excepciones definidas por el usuario.

EXCEPCIONES PREDEFINIDAS

Son aquellas que se disparan automáticamente al producirse determinados errores. Las más frecuentes son:

- **too_many_rows**: Se produce cuando SELECT ... INTO devuelve más de una fila.
- **no_data_found**: se produce cuando un SELECT... INTO no devuelve ninguna fila.
- **value_error**: se produce cuando hay un error aritmético o de conversión.
- **zero_divide**: se produce cuando hay una división entre 0.
- **dupval_on_index**: se produce cuando se intenta almacenar un valor que crearía duplicados en la clave primaria o en una columna con restricción UNIQUE.
- **invalid_number**: se produce cuando se intenta convertir una cadena a un valor numérico.

Al estar ya definidas en Oracle no se declaran en la sección DECLARE.

EXCEPCIONES DEFINIDAS POR EL USUARIO

Son las que define el usuario. Para poder trabajar con excepciones definidas por el usuario, es necesario que se realicen 3 pasos:

1. Definición del nombre de la excepción en la sección DECLARE. La sintaxis es

Nombre_de_excepción EXCEPTION
2. Lanzar la excepción. Se hace mediante la orden RAISE ;
3. Definir las acciones a seguir si se lanza la excepción. Esto se hace en la sección EXCEPTION.

WHEN nombre_de_excepción THEN ...

Normalmente definiremos excepciones asociadas a un tipo concreto de programa de PL/SQL (procedimientos, funciones, etc.) que veremos en sesiones posteriores.

Ejemplo 1:

Crea un bloque de código en la que dado el nif '11111111A' se muestre su nombre y teléfono.

```
DECLARE
    auxnombre empleado.nombre%type;
    auxtelefono empleado.telefono%type;
BEGIN
    select nombre, telefono into auxnombre, auxtelefono
    from empleado where nif='11111111A';
    dbms_output.put_line('Nombre '||auxnombre||' telefono '||auxtelefono);

    EXCEPTION
    WHEN no_data_found then
        dbms_output.put_line ('No existe empleado con ese nif');
END;
```

Ejemplo 2:

Crea un bloque de código en el que dado el nif '11111111A' y el valor máximo de 10, incluya a ese empleado como recepcionista siempre que no existan ya más de 10 recepcionistas.

Si al darlo de alta como recepcionista vemos que no está dado de alta como empleado, se dará de alta tanto de empleado como de recepcionista y se mostrará un mensaje indicando que falta por completar sus datos.

DECLARE

auxnombre empleado.nombre%type;

auxtelefono empleado.telefono%type;

total number(3);

pasa_tope exception;

BEGIN

select count(*) into total from emprecepcion;

if total>10 then **raise pasa_tope;** end if;

select nombre, telefono **into auxnombre, auxtelefono**

from empleado where nif='11111111A';

dbms_output.put_line ('Nombre '||auxnombre||' pasa a tareas de recepcionista');

insert into emprecepcion values('11111111A');

EXCEPTION

WHEN no_data_found then

insert into empleado(nif) values('11111111A');

insert into emprecepcion values('11111111A');

dbms_output.put_line ('No existía. Se ha dado de alta como recepcionista, falta completar sus datos en empleado');

WHEN pasa_tope then

dbms_output.put_line ('Se supera el tope asignado para recepcionistas');

END;