

UA4B. PL/SQL

4.14.1. Excepciones II

Ahora que ya sabemos lo que son las excepciones, cómo capturarlas y manejarlas y cómo definir y lanzar las nuestras propias. Es la hora de comentar algunos detalles sobre el uso de las mismas.

El alcance de una excepción sigue las mismas reglas que el de una variable, por lo que si nosotros redefinimos una excepción que ya es global para el bloque, la definición local prevalecerá y no podremos capturar esa excepción a menos que el bloque en la que estaba definida esa excepción fuese un bloque nombrado, y podremos capturarla usando la sintaxis: `nombre_bloque.nombre_excepcion`.

Las excepciones predefinidas están definidas globalmente. No necesitamos (ni debemos) redefinir las excepciones predefinidas.

```
DECLARE
    no_data_found EXCEPTION;
BEGIN
    SELECT * INTO ...
EXCEPTION
    WHEN no_data_found THEN      --captura la excepción local, no
                                --la global
END;
```

Cuando manejamos una excepción no podemos continuar por la siguiente sentencia a la que la lanzó.

```

DECLARE

    ...

BEGIN

    ...
    INSERT INTO familias VALUES
(id_fam, nom_fam, NULL, oficina);
    INSERT INTO agentes VALUES
(id_ag, nom_ag, login, password, 0, 0, id_fam, NULL);
    ...
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        --manejamos la excepción debida a que el nombre de
        --la familia ya existe, pero no podemos continuar por
        --el INSERT INTO agentes, a no ser que lo pongamos
        --explícitamente en el manejador

END;

```

Pero sí podemos encerrar la sentencia dentro de un bloque, y ahí capturar las posibles excepciones, para continuar con las siguientes sentencias.

```

DECLARE
    id_fam NUMBER;
    nom_fam VARCHAR2(40);
    oficina NUMBER;
    id_ag NUMBER;
    nom_ag VARCHAR2(60);
    usuario VARCHAR2(20);
    clave VARCHAR2(20);
BEGIN
    ...
    BEGIN
        INSERT INTO familias VALUES (id_fam, nom_fam, NULL, oficina);
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            SELECT identificador INTO id_fam FROM familias WHERE nombre = nom_fam;
    END;
    INSERT INTO agentes VALUES (id_ag, nom_ag, login, password, 1, 1,

```

...

END;

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**
<<http://creativecommons.org/licenses/by-sa/4.0/>>