



Manejo de Excepciones en PL/SQL

CONCEPTOS

● Qué es una excepción?

- Es la identificación de un tipo de evento que se levanta en SQL

● Cómo se levanta la excepción?

- Cuando ocurre un error de Oracle.
- Levantando el error explícitamente.

● Cómo se maneja el error?

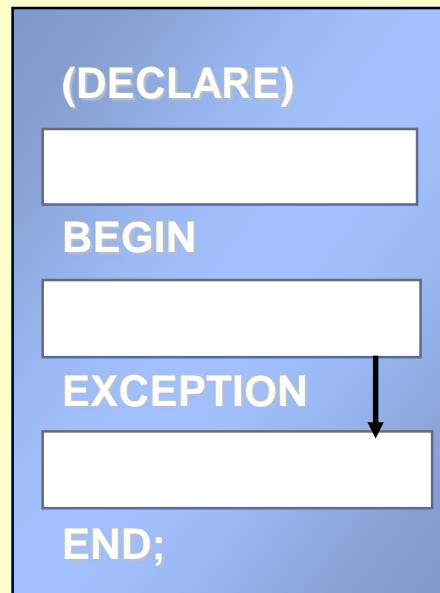
- Atrapándolo con un manejador.
- Propagándolo al ambiente que se encuentra por encima del PL/SQL que se ejecuta.

Manejo de excepciones

Atrapar la excepción

Se levanta una excepción

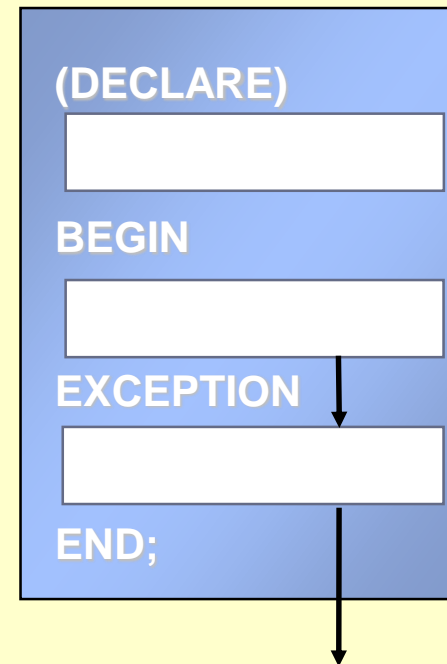
Se atrapa la excepción



Propagar la excepción

Se levanta la excepción

La excepción no se previó



La excepción se propaga al ambiente desde el cual se llama

TIPOS DE EXCEPCIONES

PROPIOS DEL ORACLE:

- PREDEFINIDOS (Servidor)
- NO PREDEFINIDOS

Se levantan de forma implícita

- DEFINIDOS POR EL USUARIO

Se levantan explícitamente

TIPOS DE EXCEPCIONES

EXCEPCIÓN	DESCRIPCIÓN	MANEJO
Errores de ORACLE predefinidos	Son unos 20 errores que ocurren más frecuentemente en el código PL/SQL	No necesita declarar estas excepciones. Son errores predefinidos en el Servidor de Oracle y son levantados implícitamente
Errores de ORACLE no predefinidos	Cualquier otro error estándar del servidor ORACLE	Se pueden declarar en la sección declarativa y el Servidor de Oracle los levanta implícitamente
Errores definidos por el usuario	El usuario determina una condición como anormal	Se deben declarar en la sección declarativa y se levantan explícitamente

SINTAXIS

EXCEPTION

```
WHEN exception1 [OR exception2 . . .] THEN
```

```
    statement1;
```

```
    statement2;
```

```
    . . .
```

```
[WHEN exception3 [OR exception4 . . .] THEN
```

```
    statement1;
```

```
    statement2;
```

```
    . . .]
```

```
[WHEN OTHERS THEN
```

```
    statement1;
```

```
    statement2;
```

```
    . . .]
```



GUÍA PARA TRATAMIENTO DE EXCEPCIONES

- WHEN OTHERS debe ser la última cláusula.
- EXCEPTION : Con esta palabra se inicia el tratamiento de excepción para un bloque BEGIN-END
- Si una de las condiciones de excepción se cumple, ya no se trata el resto

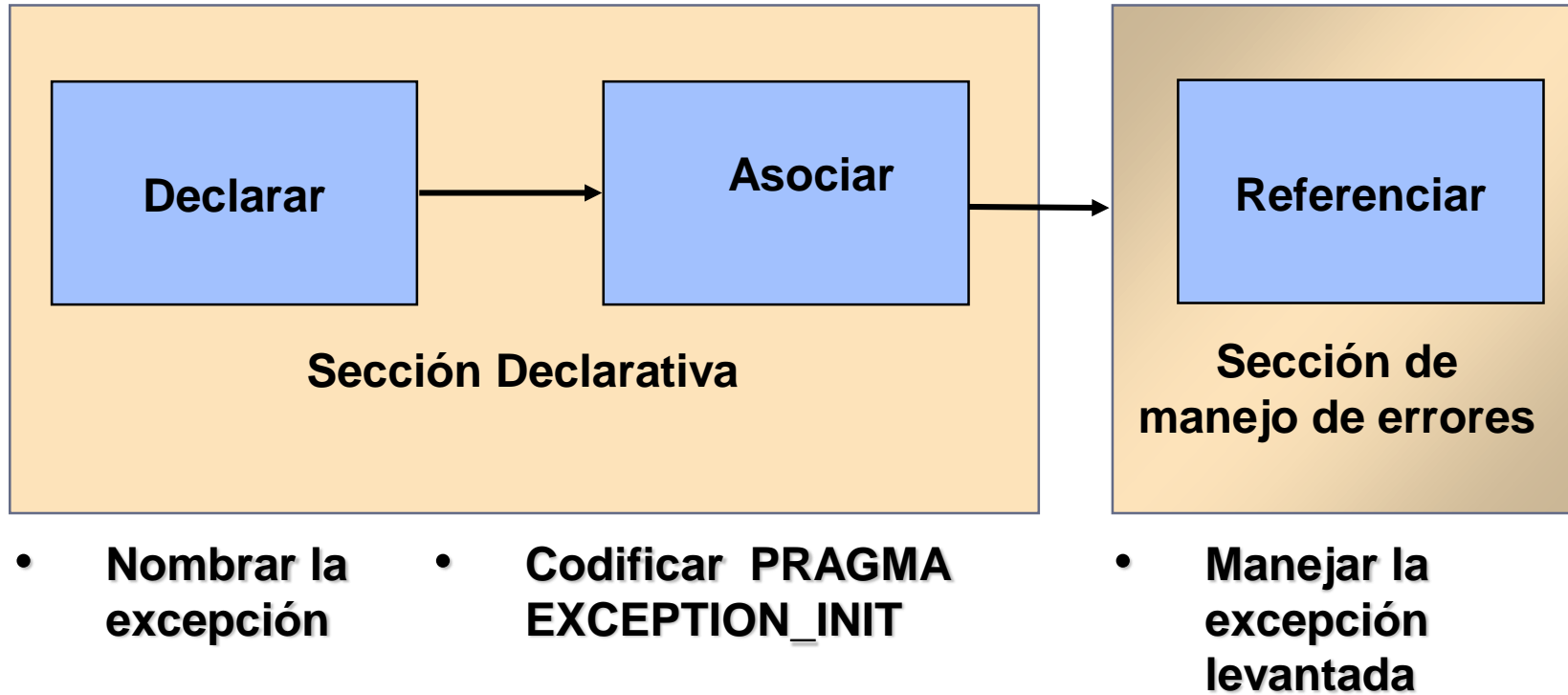
Resumen de errores predefinidos

ERRORES PREDEFINIDOS EN ORACLE	
MENSAJE	
CURSOR_ALREADY_OPEN	ORA-06511
DUP_VAL_ON_INDEX	ORA-00001
INVALID_NUMBER	ORA-01001
INVALID_CURSOR	ORA-01722
LOGIN_DENIED	ORA-01017
NO_DATA_FOUND	ORA-01403
NOT_LOGGED_ON	ORA-01012
PROGRAM_ERROR	ORA-06501
ROWTYPE_MISMATCH	ORA-06504
STORAGE_ERROR	ORA-06500
TIMEOUT_ON_RESOURCE	ORA-00051
TOO_MANY_ROWS	ORA-01422
VALUE_ERROR	ORA-06502
ZERO_DIVIDE	ORA-01476

Ejemplos de excepciones predefinidas

```
CREATE OR REPLACE PROCEDURE elim_articulo
(p_id_articulo IN b_articulos.id%TYPE) IS
v_id    b_articulos.id%TYPE;
BEGIN
  /* Verifico antes si no tengo registro de ventas */
  BEGIN
    SELECT  id_articulo INTO  v_id
    FROM    b_detalle_ventas
    WHERE   id_articulo = p_id_articulo
    AND     ROWNUM = 1;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      -- No existen ventas, por lo tanto puedo borrar --
      DELETE FROM  b_articulos
      WHERE  id = p_id_articulo;
      COMMIT;
  END;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE ('Error inesperado ');
END elim_articulo;
```

ERRORES NO PREDEFINIDOS



EJEMPLO DE ERRORES NO PREDEFINIDOS

Error ORA-02292 (restricción de integridad violada)

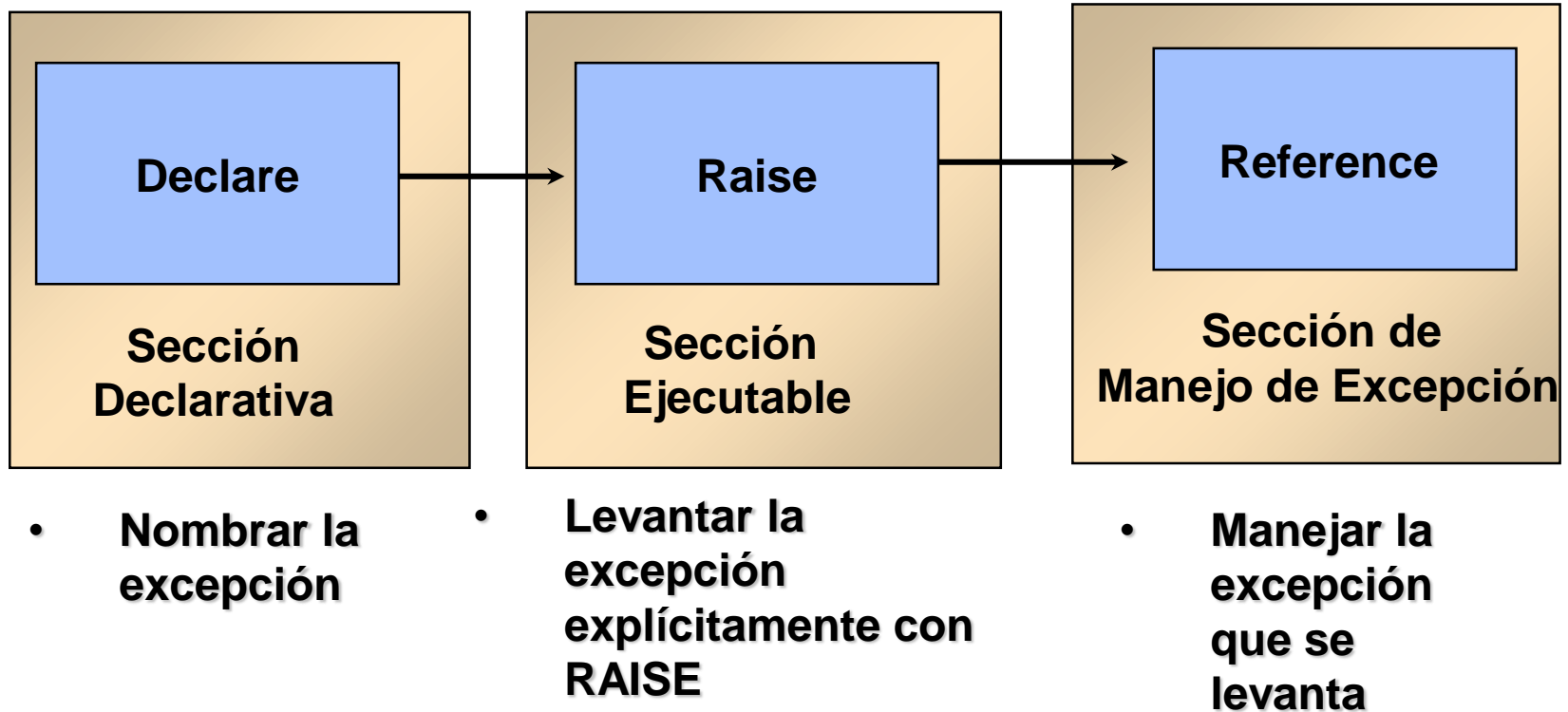
```
DECLARE
    e_tiene_hijos  EXCEPTION;
    PRAGMA EXCEPTION_INIT (e_tiene_hijos, -2292);
BEGIN
    DELETE FROM b_articulos WHERE id = &id;
    COMMIT;
EXCEPTION
    WHEN e_tiene_hijos THEN
        DBMS_OUTPUT.PUT_LINE ('no puede borrar; tiene
        ítems que hacen referencia al artículo');
END;
/
```

1

2

3

EXCEPCIONES PREDEFINIDAS POR EL USUARIO



EJEMPLO DE EXCEPCIÓN DEFINIDA POR EL USUARIO

```
DECLARE
  v_id    b_articulos.id%type := &ID;
  e_tiene_hijos EXCEPTION;
  v_cant  NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_cant
    FROM B_detalle_ventas
   WHERE id_articulo = v_id;
  -- Si encuentra, es error --
  IF NVL(v_cant,0) > 0 THEN
    RAISE e_tiene_hijos;
  ELSE
    DELETE FROM b_articulos WHERE id = v_id;
    COMMIT;
  END IF;
EXCEPTION
  WHEN e_tiene_hijos THEN
    DBMS_OUTPUT.PUT_LINE ('no puede borrar; tiene ítems
que hacen referencia al artículo');
END;
```

1

2

3

FUNCIONES PARA ATRAPAR EXCEPCIONES

● **WHEN OTHERS**

- Atrapa las excepciones por las que no se preguntó
- Debe ser la última excepción manejada

● **SQLCODE**

- Retorna el valor numérico del error

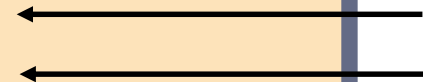
● **SQLERRM**

- Retorna el mensaje asociado con el error

EJEMPLO

```

. . .
  v_error_code      NUMBER;
  v_error_message   VARCHAR2 (255) ;
BEGIN
. . .
EXCEPTION
. . .
  WHEN OTHERS THEN
    ROLLBACK;
    v_error_code := SQLCODE;
    v_error_message := SQLERRM;
    DBMS_OUTPUT.PUT_LINE
(TO_CHAR(v_error_code) ||
  ': ' || v_error_message);
END;
```



PROPAGACIÓN DE EXCEPCIONES

Un bloque puede
manejar el error o
pasarlo al siguiente
bloque

```
DECLARE
...
e_no_rows      exception;
e_integrity     exception;
PRAGMA EXCEPTION_INIT (
                e_integrity, -2292);
BEGIN
  FOR c_record IN emp_cursor LOOP

    BEGIN
      SELECT . . .
      UPDATE . . .
      IF SQL%NOTFOUND THEN
        RAISE e_no_rows;
      END IF;
    EXCEPTION
      WHEN e_integrity THEN . . .
      WHEN e_no_rows THEN . . .
    END;

  END LOOP;
EXCEPTION
  WHEN NO_DATA_FOUND THEN ...
  WHEN TOO_MANY_ROWS THEN ...
END;
```


CÓDIGOS DE ERROR DEFINIDOS POR EL USUARIO

Oracle permite también definir *números de error y mensajes* que son retornados a la aplicación cliente, la que, una vez recibida, puede manejar el error basado en el error especificado por el usuario

Los errores especificados por el usuario pueden retornarse usando:

RAISE_APPLICATION_ERROR(nro_error, 'texto');

CÓDIGOS DE ERROR DEFINIDOS POR EL USUARIO

- La cláusula **RAISE** termina la ejecución del programa, deshace todos los efectos del procedimiento y retorna un número de error y un mensaje que especifica el usuario.
- El número de error debe manejarse en el rango de **-20000** a **-20999**, y el texto se refiere a una expresión de tipo carácter.

CÓDIGOS DE ERROR DEFINIDOS POR EL USUARIO - Ejemplo

```
CREATE OR REPLACE FUNCTION F_RECUPERAR_EMP
(PCED NUMBER) RETURN VARCHAR2 IS
  V_NOMBRE VARCHAR2(100);
BEGIN
  SELECT NOMBRE||' '||APELLIDO INTO V_NOMBRE
  FROM   B_EMPLEADOS E
  WHERE  E.CEDULA = PCED;
  RETURN V_NOMBRE;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20002, 'NO SE
    ENCUENTRA EL EMPLEADO ');
END;
/
```