

Arquitetura e Administração de Bases de Dados  
Oracle11g PL/SQL Programming

**João Costa**  
jcosta@isec.pt

1

## Agenda

---

- ▶ Tratamento de exceções

---

▶ 2

2023/24 - LEI - AABD - PL/SQL

2

## Executar um bloco com erros

### Erros comuns

- ▶ Usar “=” em vez de “:=”  
 ORA-06550: linha 5, coluna 1:  
 PLS-00103: Encontrado o símbolo "=" quando era esperado ...
- ▶ Não declarar uma variável  
 ORA-06550: linha 5, coluna 1:  
 PLS-00201: o identificador 'A' deve ser declarado
- ▶ Escrever mal o nome de uma variável  
 ORA-06550: linha 5, coluna 1:  
 PLS-00201: o identificador 'AD' deve ser declarado
- ▶ Não terminar um comando com “;”  
 ORA-06550: linha 5, coluna 1:  
 PLS-00103: Encontrado o símbolo "=" quando era esperado ...
- ▶ Comando SELECT que não devolve dados  
 ORA-01403: não foram encontrados dados

▶ 3

2023/24 - LEI - AABD - PL/SQL

3

## Executar um bloco com erros

- ▶ Não terminar um comando com “;”

```

SQL> DECLARE
2  lv_basket_num NUMBER(3);
3  lv_created_date DATE;
4  lv_qty_num NUMBER(2);
5  lv_sub_num NUMBER(5,2);
6  lv_days_num NUMBER(3);
7  lv_shopper_num NUMBER(3) := 25;
8  BEGIN
9  SELECT idBasket, dtcreated, quantity, subtotal
10 INTO lv_basket_num, lv_created_date, lv_qty_num, lv_sub_num
11 FROM bb_basket
12 WHERE idShopper = lv_shopper_num
13 AND orderplaced = 0;
14 lv_days_num := SYSDATE - lv_created_date;
15 DBMS_OUTPUT.PUT_LINE(lv_basket_num||' * '||lv_created_date||' * '||
16 lv_qty_num||' * '||lv_sub_num||' * '||lv_days_num);
17 END;
18 /
DBMS_OUTPUT.PUT_LINE(lv_basket_num||' * '||lv_created_date||' * '||
*
ERROR at line 15:
ORA-06550: line 15, column 2:
PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the
following:
. ( * @ % & = - + ; < / > at in is mod not ren
<an exponent (**)> <> or != or ~= >= <= <> and or like
between ||
The symbol "." was substituted for "DBMS_OUTPUT" to continue.
SQL>
  
```

▶ 4

2023/24 - LEI - AABD - PL/SQL

4

## Tratamento de excepções

- ▶ Usado para capturar situações de erro
  - ▶ Assumir o processamento para que a aplicação possa continuar
- ▶ Num bloco PL/SQL, colocado na secção **EXCEPTION**

```
DECLARE
  -- variable declaration
BEGIN
  -- body
EXCEPTION
  -- WHEN conditions
END;
```

▶ 5

2023/24 - LEI - AABD - PL/SQL

5

## Tratamento de excepções

### Tipos de erros

- ▶ Erros do Oracle
  - ▶ Pré-definidos
  - ▶ Não pré-definidos
- ▶ Erros definidos pelo utilizador

▶ 6

2023/24 - LEI - AABD - PL/SQL

6

## Erros pré-definidos do Oracle

EXCEPTION NAME	SQLCODE	DESCRIPTION
DUP_VAL_ON_INDEX	-0001	It is raised when duplicate values are attempted to be stored in a column with unique index.
INVALID_CURSOR	-1001	It is raised when attempts are made to make a cursor operation that is not allowed, such as closing an unopened cursor.
NO_DATA_FOUND	-1403	It is raised when a SELECT INTO statement returns no rows.
TOO_MANY_ROWS	-1422	It is raised when a SELECT INTO statement returns more than one row.
ZERO_DIVIDE	-1476	It is raised when an attempt is made to divide a number by zero.
VALUE_ERROR	-6502	It is raised when an arithmetic, conversion, truncation, or size constraint error occurs.
CASE_NOT_FOUND	-6592	It is raised when none of the choices in the WHEN clause of a CASE statement is selected, and there is no ELSE clause.

▶ 7

2023/24 - LEI - AABD - PL/SQL

7

## Erros pré-definidos: Exemplo

```
DECLARE
  cod NUMBER;
BEGIN
  SELECT  codigo_autor INTO cod
  FROM    autores
  WHERE   codigo_autor = 999;
END;
/
*
```

ERROR at line 1:  
ORA-01403: no data found  
ORA-06512: at line 1

▶ 8

2023/24 - LEI - AABD - PL/SQL

8

## Adicionar tratamento de exceções

```

DECLARE
    cod NUMBER;
BEGIN
    SELECT  codigo_autor INTO cod
    FROM    autores
    WHERE   codigo_autor = 999;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('AUTOR INEXISTENTE!!');
END;
/

AUTOR INEXISTENTE!!
PL/SQL procedure successfully completed.

```

▶ 9

2023/24 - LEI - AABD - PL/SQL

9

## Erros não pré-definidos

- ▶ Não têm nomes pré-definidos
- ▶ Podem ser “*apanhados*” com  
WHEN OTHERS

```

BEGIN
    DELETE autores
    WHERE  codigo_autor = 1;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro');
END;
/

```

▶ 10

2023/24 - LEI - AABD - PL/SQL

10

## Erros não pré-definidos (cont.)

Também podemos associar um nome a um código de erro

- 1º Passo – Identificar possíveis erros

```
BEGIN
  DELETE FROM LIVROS;
END;
/

BEGIN
*
ERROR at line 1:
ORA-02292: integrity constraint (LEIBD1C01.FK_VENDA_ID_LIVRO_LIVROS) violated
- child record found
```

► 11

2023/24 - LEI - AABD - PL/SQL

11

## Erros não pré-definidos (cont.)

- 2º passo - Associar uma variável do tipo **exception** ao código do erro e tratar a exceção

```
DECLARE
  EX_LIVROS EXCEPTION;
  PRAGMA EXCEPTION_INIT( EX_LIVROS, -2292 );
BEGIN
  -- ...

  DELETE FROM LIVROS;

EXCEPTION
  WHEN EX_LIVROS THEN
    DBMS_OUTPUT.PUT_LINE('ALGUNS DOS LIVROS FORAM VENDIDOS');
END;
/

ALGUNS DOS LIVROS FORAM VENDIDOS!!

PL/SQL procedure successfully completed.
```

Declarar uma exceção

Associar um nº de erro do Oracle ao nome da exceção

Ocorre um erro de violação da FK se ainda houver itens no cesto.

O tratamento da exceção é executado se o comando **DELETE** despoletar o erro -2292.

► 12

2023/24 - LEI - AABD - PL/SQL

12

## Excepções definidas pelo utilizador

- ▶ Não é despoletado nenhum erro pelo sistema
- ▶ Podem servir para aplicar as regras de negócio
- ▶ Uma vez despoletado o erro (**RAISE**)
  - ▶ Os restantes comandos não são executados
- ▶ O processamento passa para a secção **EXCEPTION**

▶ 13

2023/24 - LEI - AABD - PL/SQL

13

## Excepções definidas pelo utilizador

```

DECLARE
  EX_UPDATE EXCEPTION;
BEGIN
  UPDATE LIVROS
    SET QUANT_EM_STOCK = QUANT_EM_STOCK -1
    WHERE CODIGO_LIVRO = 21;

  IF SQL%NOTFOUND THEN
    RAISE EX_UPDATE;
  END IF;
EXCEPTION
  WHEN EX_UPDATE THEN
    DBMS_OUTPUT.PUT_LINE('LIVRO INEXISTENTE!!');
END;
/

LIVRO INEXISTENTE!!

PL/SQL procedure successfully completed.

```

Declarar uma excepção

Se não forem actualizadas linhas, lançar excepção

Tratar Excepção

▶ 14

2023/24 - LEI - AABD - PL/SQL

14

## Excepções – Conceitos Adicionais

### ▶ **WHEN OTHERS**

- ▶ Apanha todos os erros não tratados especificamente noutra secção de tratamento de exceções
- ▶ Usado para tratar erros inesperados

▶ 15

2023/24 - LEI - AABD - PL/SQL

15

## Excepções – Conceitos Adicionais

- ▶ **SQLCODE** - Identifica o código do erro ocorrido
- ▶ **SQLERRM** - Identifica a mensagem associada ao erro
- ▶ Funções usadas especialmente em testes
  - ▶ Para identificar erros inesperados

▶ 16

2023/24 - LEI - AABD - PL/SQL

16



## SQLCODE and SQLERRM: Exemplo

- ▶ Guardar informação de erros numa tabela da BD usando estas funções

```
-- .. Plsql code

WHEN OTHERS THEN
  COD := SQLCODE;
  MSG := SQLERRM;

  INSERT INTO LOG (DATA, CODIGO, MENSAGEM)
    VALUES (SYSDATE, COD, MSG);

  DBMS_OUTPUT.PUT_LINE('OCORREU UM ERRO!!');
  DBMS_OUTPUT.PUT_LINE('VERIFICAR O LOG!!');
```

É necessário guardar previamente os valores em variáveis, antes de inserir na tabela

▶ 17

2023/24 - LEI - AABD - PL/SQL

17

## Propagação de excepções

- ▶ Quando ocorre um erro num bloco
  - ▶ Processamento transfere-se para a secção `EXCEPTION` desse bloco
- ▶ Se o bloco for inserido noutra bloco
  - ▶ Se não existirem secções de tratamento de excepções
    - ▶ O processamento passa para a secção `EXCEPTION` do bloco exterior

▶ 18

2023/24 - LEI - AABD - PL/SQL

18

## RAISE\_APPLICATION\_ERROR

- ▶ Permite atribuir um nº e uma mensagem a um erro

```
DECLARE
    QUANT LIVROS.QUANT_EM_STOCK%TYPE;
BEGIN
    SELECT QUANT_EM_STOCK INTO quant
    FROM    LIVROS
    WHERE CODIGO_LIVRO = 21;

    IF quant <=0 THEN
        RAISE_APPLICATION_ERROR(-20000,'Não há livros em stock') ;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('LIVRO INEXISTENTE!!');
END;
/
```

▶ 19

2023/24 - LEI - AABD - PL/SQL

19

## RAISE\_APPLICATION\_ERROR

- ▶ Permite atribuir um nº e uma mensagem a um erro

```
DECLARE
    QUANT LIVROS.QUANT_EM_STOCK%TYPE;
BEGIN
    SELECT QUANT_EM_STOCK INTO quant
    FROM    LIVROS
    WHERE CODIGO_LIVRO = 21;

    IF quant <=0 THEN
        RAISE_APPLICATION_ERROR(-20000,'Não há livros em stock') ;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('LIVRO INEXISTENTE!!');
END;
/
```

```
ERROR at line 1
ORA-2000: Não há livros em stock!!
ORA-06512: at line 1
```

▶ 20

2023/24 - LEI - AABD - PL/SQL

20

## Exercícios

---

1. Apague o livro com o código 1
  - ▶ Apanhar a exceção e fazer o output da mensagem
  
2. Inserir uma venda do livro com o código 2
  - ▶ Lançar uma exceção se a quantidade a vender for superior á quantidade em stock