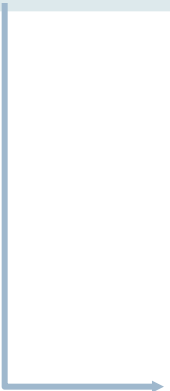


# **Tratamento de Exceções – PL/SQL**

# O que é uma Exceção?

```
DECLARE
    v_lname VARCHAR2(15);
BEGIN
    SELECT last_name INTO v_lname
    FROM employees
    WHERE first_name='John';
    DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
END;
```



Results Script Output Explain Autotrace DBMS Output OWA Output

Error starting at line 3 in command:

```
DECLARE
    v_lname VARCHAR2(15);
BEGIN
    SELECT last_name INTO v_lname FROM employees WHERE
    first_name='John';
    DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
END;
```

Error report:

ORA-01422: exact fetch returns more than requested number of rows

ORA-06512: at line 4

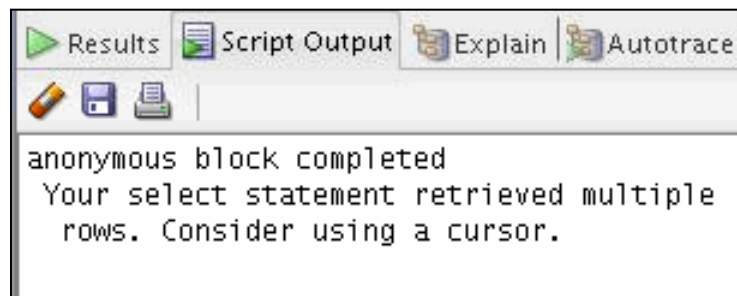
01422. 00000 - "exact fetch returns more than requested number of rows"

\*Cause: The number specified in exact fetch is less than the rows returned.

\*Action: Rewrite the query or change number of rows requested

# Manipulando a Exceção: Um exemplo

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees
  WHERE first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE (' Seu comando select recuperou
    várias linhas. Considere o uso de um cursor. ');
END;
/
```



# TRATAMENTO DE EXCEÇÕES

---

- ▶ Exceção é qualquer problema ocorrido que impossibilita o banco de dados de executar determinada instrução no bloco PL/SQL.
- ▶ A instrução pode ser um comando SQL ou uma instrução de atribuição.
- ▶ As exceções podem ser pré-definidas ou definidas pelo usuário.
- ▶ Quando um erro ocorre, o fluxo é desviado para a seção **EXCEPTION** do bloco PL/SQL correspondente, onde a exceção deve ser tratada.
- ▶ Se a seção **EXCEPTION** não existir, a mensagem de erro é devolvida para o usuário.



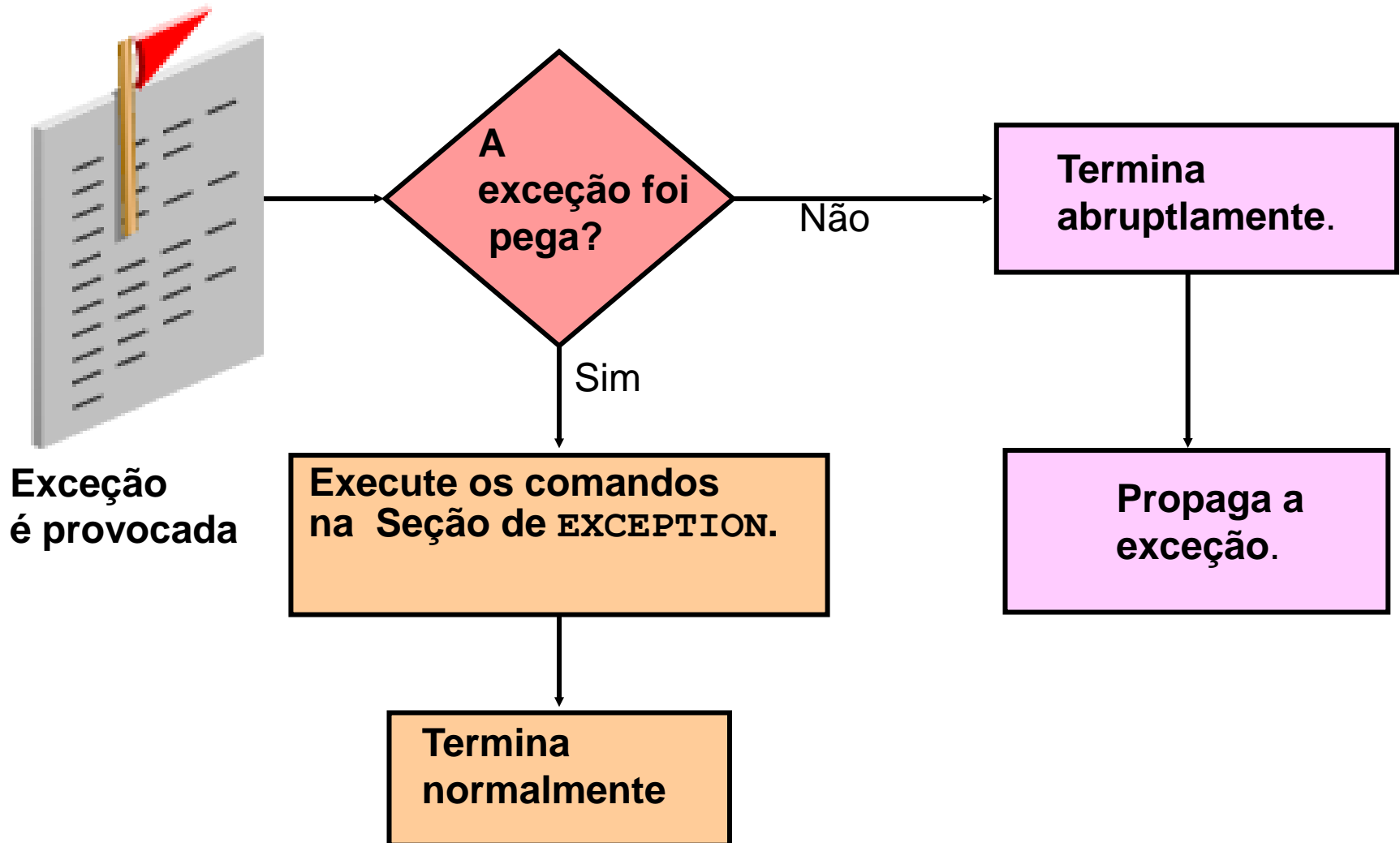
# Entendendo Exceções com PL/SQL

---

- ▶ Uma exceção é um erro PL/SQL que é provocado em tempo de execução.
- ▶ Uma exceção pode ser provocada:
  - ▶ Implicitamente pelo Servidor Oracle
  - ▶ Explicitamente pelo programa
- ▶ Uma exceção pode ser manipulada:
  - ▶ Aprisionando-a com um manipulador
  - ▶ Propagando ela ao ambiente de chamada



# Manipulando Exceções



# Tipos de Exceções

- ▶ Predefinida pelo Servidor Oracle
  - ▶ Não predefinida Servidor Oracle
- } provocada implicitamente
- 
- ▶ Definida pelo usuário
- } provocada explicitamente

---

<b>Exceção</b>	<b>Descrição</b>	<b>Orientações para tratamento</b>
Predefinida pelo Servidor Oracle	Um dos aproximadamente 20 erros mais frequentes no código PL/SQL	Você precisa need not declare these exceptions. They are predefined by the Oracle server and are raised implicitly.
Não predefinida Servidor Oracle	Qualquer outro erro padrão do servidor Oracle	Você precisa declará-las na seção declarativa; o servidor Oracle dispara o erro implicitamente, e você pode tratar o erro no manipulador de exceções.
Definida pelo usuário	Uma condição que o desenvolvedor determina como anormal	Você precisa declarar na secção declarative e disparar explicitamente.





# Sintaxe para tratar Exceções

EXCEPTION

```
WHEN exceção1 [OR exceção2 . . .] THEN  
    comando1;  
    comando2;
```

. . .

```
[WHEN exceção3 [OR exceção4 . . .] THEN  
    comando1;  
    comando2;  
    . . .]
```

```
[WHEN OTHERS THEN  
    comando1;  
    comando2;  
    . . .]
```

# Sintaxe para

## Manipulador de Exceção

EXCEPTION

```
WHEN exceção1 [OR exceção2 . . .] THEN  
    comando1;  
    comando2;  
    . . .
```

```
[WHEN exceção3 [OR exceção4 . . .] THEN  
    comando1;  
    comando2;  
    . . .]
```

```
[WHEN OTHERS THEN  
    comando1;  
    comando2;  
    . . .]
```

Você pode incluir qualquer número de tratadores dentro da Seção para tratar exceções específicas. Entretanto, você não pode ter vários tratadores para uma só exceção.

# Sintaxe para pegar Exceções

EXCEPTION

```
WHEN exceção1 [OR exceção2]
    comando1;
    comando2;
```

. . .

```
[WHEN exceção3 [OR exceção4]
    comando1;
    comando2;
```

. . .]

```
[WHEN OTHERS THEN
    comando1;
    comando2;
    . . .]
```

Vários manipuladores de exceção são permitidos.

Somente um manipulador é processado antes de concluir o bloco.

WHEN OTHERS é a última cláusula.

# Detectando Erros do Servidor Oracle pré-definidos

---

- ▶ Referencia o nome pré-definido na rotina de tratamento de exceções.
- ▶ Amostra de exceções pré-definidas:
  - ▶ `NO_DATA_FOUND`
  - ▶ `TOO_MANY_ROWS`
  - ▶ `INVALID_CURSOR`
  - ▶ `ZERO_DIVIDE`
  - ▶ `DUP_VAL_ON_INDEX`



# EXCEÇÕES PRÉ-DEFINIDAS

---

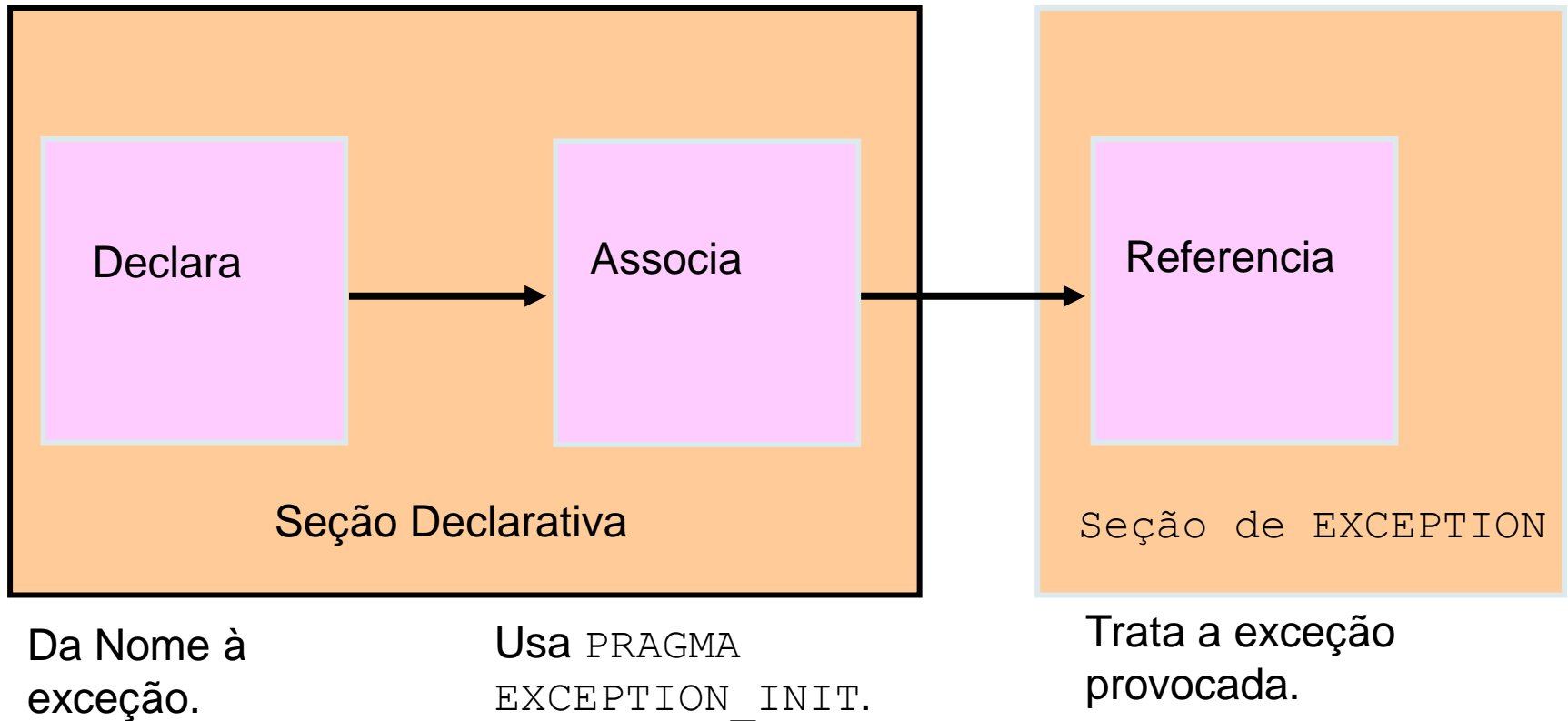
- ▶ Dup\_val\_on\_index - índice duplicado
  - ▶ Sqlcode = -1 Erro oracle = ORA-00001
  - ▶
  - ▶ Invalid Number - número inválido na conversão de valores
  - ▶ Sqlcode = -1722 Erro oracle = ORA-01722
  - ▶
  - ▶ Login\_denied - usuário/senha inválidos
  - ▶ Sqlcode = -1017 Erro oracle = ORA-01017
  - ▶
  - ▶ No\_Data\_Found - nenhuma linha recuperada
  - ▶ Sqlcode = +100 Erro oracle = ORA-1403
  - ▶
  - ▶ .Not\_logged\_on - não conectado ao Oracle
  - ▶ Sqlcode = -1012 Erro oracle = ORA-01012
  - ▶
- 



- 
- ▶ Program\_error      - erro interno do PL/SQL.
  - ▶ Sqlcode = -6501      Erro oracle = ORA-06501
  - ▶
  - ▶ Store\_error      - problema de memória.
  - ▶ Sqlcode = -6500      Erro oracle = ORA-06500
  - ▶
  - ▶ Timeout\_on\_resource      - problema de timeout (estouro de tempo no acesso a um objeto da base de dados)
  - ▶ Sqlcode = -51      Erro oracle = ORA-00051
  - ▶
  - ▶ Too\_many\_rows      - mais de uma linha recuperada
  - ▶ Sqlcode = -1427      Erro oracle = ORA-01427
  - ▶
  - ▶ Value\_error      - erro de conversão de dado ou valor atribuído a um campo ou coluna maior que os mesmos.
  - ▶ Sqlcode = -6502      Erro oracle = ORA-06502
  - ▶
  - ▶ Zero\_divide      - tentativa de dividir um número por zero.
  - ▶ Sqlcode = -1476      Erro oracle = ORA-01476
  - ▶
- 



# Detectando Erros do Servidor Oracle não pré-definidos



# Detectando Erros do Servidor Oracle não pré-definidos

---

- ▶ Exceções não pré-definidas são similares às exceções pré-definidas; entretanto, elas não são definidas como exceções PL/SQL no servidor Oracle. Eles são erros padrão de Oracle. Você cria exceções com erros padrão de Oracle usando a função `PRAGMA EXCEPTION_INIT`. Tais exceções são chamadas de exceções não pré-definidas.
- ▶ Você pode tratar um erro não pré-definido no servidor Oracle Server declarando ela primeiro. A exceção declarada é disparada implicitamente. Em PL/SQL, `PRAGMA EXCEPTION_INIT` diz ao compilador para associar um nome de exceção a um número de erro de Oracle. Isto facilita referenciar qualquer exceção interna pelo nome e escrever um tratamento de exceções específico para ele.
- ▶ **PS:** `PRAGMA` (também chamada de *pseudoinstructions*) é a palavra chave que seignifica que o comando é uma diretiva para o compilador, que não é processada quando o bloco PL/SQL é executado.

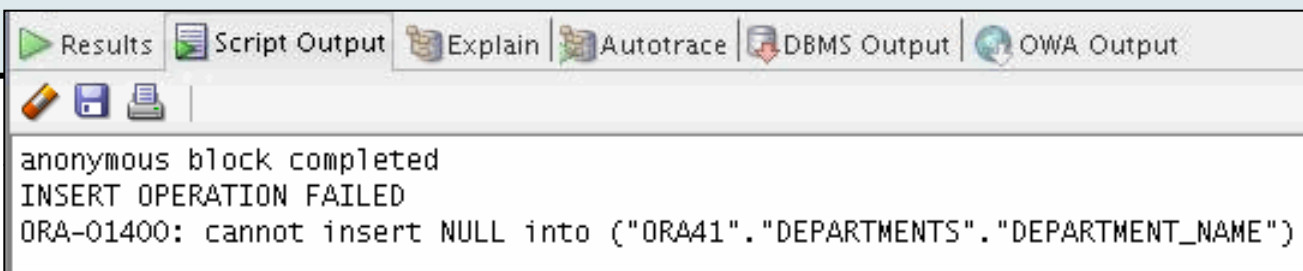




# Tratamento de Erros não-predefinidos: Um exemplo

- ▶ Para tratar o erro Oracle Server 01400 (“cannot insert NULL”):

```
DECLARE
  e_excec_insert EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_excec_insert, -01400);
BEGIN
  INSERT INTO departments
    (department_id, department_name) VALUES (280, NULL);
EXCEPTION
  WHEN e_excec_insert THEN
    DBMS_OUTPUT.PUT_LINE('INSERT OPERATION FAILED');
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
/
```



- 
- ▶ **Sintaxe :**
  - ▶ DECLARE
  - ▶ <nome exceção> EXCEPTION;
  - ▶
  - ▶ PRAGMA EXCEPTION\_INIT (nome exceção, código\_erro\_oracle);
  - ▶
  - ▶ BEGIN
  - ▶ ....
  - ▶
  - ▶ Para especificar qual exceção iniciará a execução do código, deve-se utilizar o comando WHEN.
  - ▶ Todas as exceções que não existe um tratamento específico, podem ser tratadas através da exceção OTHERS.
- 



# Funções para Detectar Exceções

---

- ▶ **SQLCODE:** Retorna o valor numérico para o código de erro (exceção interna)
- ▶ **SQLERRM:** Retorna a mensagem associada ao número de erro



- 
- ▶ Quando uma exceção ocorre, você pode identificar o código de erro associado ou mensagem de erro usando duas funções. Baseado nos valores do código ou a mensagem, você pode decidir o que fazer depois (ações de tratamento).

Função	Descrição
SQLCODE	Retorna o valor numérico para o código de erro (Você pode atribuir a ele uma variável NUMBER.)
SQLERRM	Retorna um literal que contém a mensagem associada com o número de erro

## Valores de SQLCODE: Exemplos

Valor SQLCODE	Descrição
0	Nenhuma exceção encontrada
1	Exceção definida pelo usuário
+100	Exceção NO_DATA_FOUND
<i>Número negativo</i>	Outro número de erro do servidor Oracle

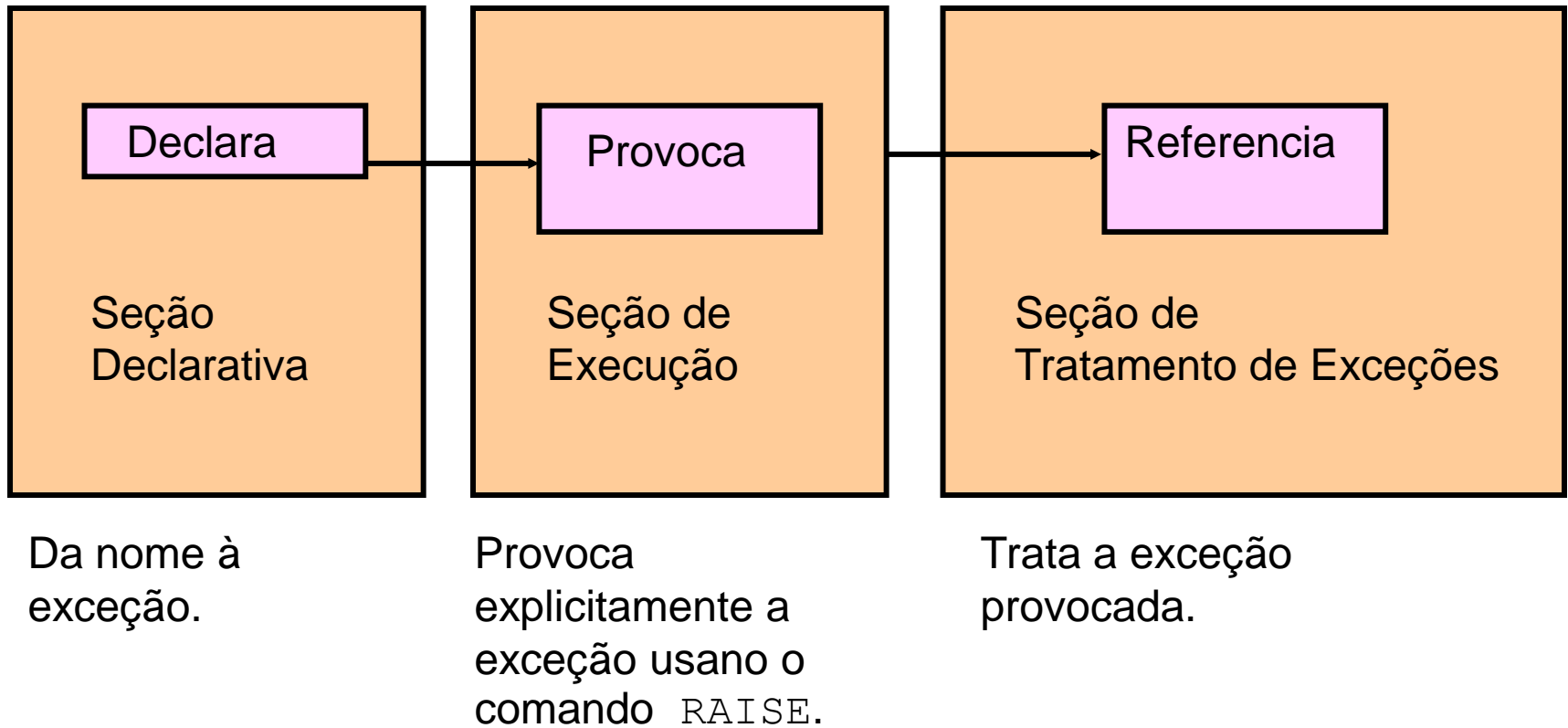


# Funções para Detectar Exceções

```
DECLARE
    v_codigo_error      NUMBER;
    v_mensagem_erro     VARCHAR2(255);
BEGIN
    ...
EXCEPTION
    ...
    WHEN OTHERS THEN
        ROLLBACK;
        v_codigo_error := SQLCODE ;
        v_mensagem_erro := SQLERRM ;
        INSERT INTO errors (e_user, e_date, error_code,
            error_message) VALUES (USER, SYSDATE, v_codigo_error,
            v_mensagem_erro);
END;
/
```

Você não pode usar `SQLCODE` ou `SQLERRM` diretamente em um comando SQL. No lugar, você deve atribuir seus valores a variáveis locais, e então usar as variáveis no comando SQL

# Tratando Exceções definidas pelo usuário



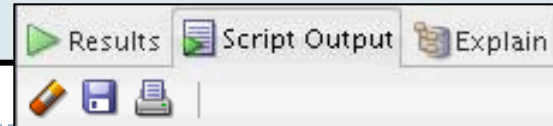
# Tratando Exceções definidas pelo usuário

```
DECLARE
  v_deptno NUMBER := 500;
  v_name VARCHAR2(20) := 'Testing';
  e_departamento_invalido EXCEPTION;
BEGIN
  UPDATE departments
  SET department_name = v_name
  WHERE department_id = v_deptno;
  IF SQL%NOTFOUND THEN
    RAISE e_departamento_invalido;
  END IF;
  COMMIT;
EXCEPTION
  WHEN e_departamento_invalido THEN
    DBMS_OUTPUT.PUT_LINE('No such department id.');
```

Diagram illustrating the execution flow of the PL/SQL block:

- 1. Declaration of the user-defined exception `e_departamento_invalido`.
- 2. Raising the exception `e_departamento_invalido` inside the `IF SQL%NOTFOUND` block.
- 3. Handling the exception in the `EXCEPTION` section.

```
END;
/
```



- 
- ▶ As exceções podem ser disparadas através da instrução RAISE
  - ▶ **Sintaxe :**
  - ▶
  - ▶ RAISE <nome da exceção>
  - ▶ No caso de exceções definidas pelo usuário, a exceção deve ser uma variável declarada na seção DECLARE, como sendo do tipo EXCEPTION.
  - ▶ Caso seja um erro do tipo ORA, pode ser associado ao erro Oracle através da declaração PRAGMA EXCEPTION\_INIT.
- 





---

**Exemplo:**

DECLARE

  X NUMBER := 0;

  Y NUMBER := 0;

  ESTOURO EXCEPTION;

BEGIN

  FOR X IN 1..1000 LOOP

    Y := X + (X/2);

    IF Y > 1000 THEN

      RAISE ESTOURO;

    END IF;

  END LOOP;

EXCEPTION

  WHEN ESTOURO THEN

    DBMS\_OUTPUT.PUT\_LINE('ESTOURO DE CAPACIDADE');

END;

---



# GERANDO INTERRUPÇÃO DO PROGRAMA

---

- ▶ O programa também provocar interrupção da execução através do comando *raise\_application\_error*.
  - ▶
  - ▶ **Sintaxe :**
  - ▶
  - ▶ `RAISE_APPLICATION_ERROR(código_do_erro, mensagem);`
  - ▶
  - ▶ Código do erro é o código a ser utilizado para o erro definido pelo usuário.
  - ▶ O Oracle reserva os códigos de 20000 até 20999 para esse propósito.
  - ▶ O erro será mostrado como qualquer outra mensagem de erro Oracle.
  - ▶ Você pode usar este procedimento para gerar mensagens de erros definidas pelo usuário desde procedimentos armazenados.
  - ▶ Você pode relatar erros para sua aplicação e evitar o retorno de exceções não tratadas.
  - ▶
- 



---

## ► Sintaxe:

```
raise_application_error (error_number,  
                        message[, {TRUE | FALSE}]);
```

- Você pode usar este procedimento para gerar mensagens de erro de sub-programas armazenados.
- Você pode relacionar erros para sua aplicação e evitar o retorno de exceções não tratadas.



- 
- ▶ Use o procedimento `RAISE_APPLICATION_ERROR` para comunicar uma exceção pré-definida interativamente retornando um código e mensagem de erro não padrão. Com `RAISE_APPLICATION_ERROR`, você pode reportar erros para sua aplicação e evita o retorno de exceções não tratadas.
  - ▶ Na sintaxe:

*Número do erro*    É um número especificado pelo usuário para as exceções entre – 20,000 e –20,999

*mensagem*        É a mensagem especificada pelo usuário para a exceção; é um string de caracteres de até 2,048 bytes.



# Procedimento

## `RAISE_APPLICATION_ERROR`

---

- ▶ É usado em dois diferentes lugares:
  - ▶ Seção de Execução
  - ▶ Seção de Exceção
- ▶ Retorna condições de erro ao usuário de forma consistente com outros erros do Servidor Oracle



# Procedimento

## RAISE\_APPLICATION\_ERROR

---

### ► Seção de execução:

```
BEGIN
...
DELETE FROM employees
  WHERE manager_id = v_mgr;
IF SQL%NOTFOUND THEN
  RAISE_APPLICATION_ERROR(-20202,
    'This is not a valid manager');
END IF;
...
```

### ► Seção de Exceção:

```
...
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR (-20201,
      'Manager is not a valid employee.');
```

END;  
/

# Outro Exemplo

---

- ▶ Aqui outro exemplo do uso do procedimento

`RAISE_APPLICATION_ERROR:`

- `DECLARE`  
    `e_nome EXCEPTION;`  
    `BEGIN`  
        `...`  
        `DELETE FROM employees`  
        `WHERE last_name = 'Higgins';`  
        `IF SQL%NOTFOUND THEN RAISE e_nome;`  
        `END IF;`  
    `EXCEPTION`  
        `WHEN e_nome THEN`  
            `RAISE_APPLICATION_ERROR (-20999, 'This is not a valid last name'); ...`  
    `END;`
- `/`



# Exemplo

---

```
SQL>ed exemplo11
set serveroutput on

DECLARE
  aux_salario emp.sal%type;
  aux_faixa   salgrade.grade%type;
BEGIN
  select  e.sal
         ,s.grade
  into    aux_salario
         ,aux_faixa
  from    emp e
         ,salgrade s
  where   e.sal between s.losal and s.hisal
  and     e.empno = &prm_empno;
  --
```

---





---

```
▶ if aux_faixa = 1 then
▶     aux_salario := aux_salario * 1.1;
▶     elsif aux_faixa = 2 then
▶         aux_salario := aux_salario * 1.2;
▶     elsif aux_faixa = 3 then
▶         aux_salario := aux_salario * 1.3;
▶     elsif aux_faixa = 4 then
▶         aux_salario := aux_salario * 1.4;
▶     elsif aux_faixa = 5 then
▶         aux_salario := aux_salario * 1.5;
▶     end if;
▶     --
▶     dbms_output.put_line('Salario de ' ||to_char(aux_salario));
▶     --
```

---



- 
- ▶ exception
  - ▶ when no\_data\_found then
  - ▶     dbms\_output.put\_line('Empregado não cadastrado ou '
  - ▶             '|| ' não se enquadra nas faixas pré-cadastradas');
  - ▶ when others then
  - ▶     raise\_application\_error(-20501,'Erro: '||sqlerrm);
  - ▶ END;
  - ▶ /
  - ▶ Enter value for prm\_empno: 111
  - ▶ old 12:   and    e.empno = &prm\_empno;
  - ▶ new 12:   and    e.empno = 111;
  - ▶ Empregado não cadastrado ou   não se enquadra nas faixas pré-cadastradas
  - ▶
  - ▶ PL/SQL procedure successfully completed.
- 



# Propagando Exceções em um Sub-bloco

---

Sub-blocos podem tratar uma exceção ou passar a exceção ao bloco que o contem.

```
DECLARE
    . . .
    e_ao_linhas    exception;
    e_integridade  exception;
    PRAGMA EXCEPTION_INIT (e_integridade,2292);
BEGIN
    FOR c_record IN emp_cursor LOOP
        BEGIN
            SELECT ...
            UPDATE ...
            IF SQL%NOTFOUND THEN
                RAISE e_ao_linhas;
            END IF;
        END;
    END LOOP;
EXCEPTION
    WHEN e_integridade THEN ...
    WHEN e_ao_linhas THEN ...
END;
/
```