



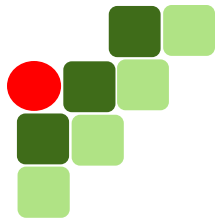
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Araraquara

Banco de Dados II

Cristiane Yaguinuma

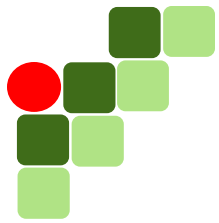
cristiane.yaguinuma@ifsp.edu.br

- Tratamento de exceções PL/SQL



Roteiro da aula

- ▶ **Introdução ao tratamento de exceções**
- ▶ **Sintaxe da seção EXCEPTION**
- ▶ **Tipos de exceções**
- ▶ **Exercícios**



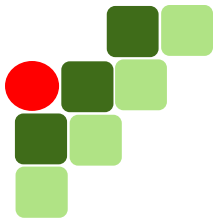
Tratamento de exceções

- ▶ Uma exceção é um erro PL/SQL lançado durante a execução de um bloco de programa
- ▶ As exceções e o tratamento de exceção são o mecanismo pelo qual programas PL/SQL reagem e lidam com os erros em tempo de execução



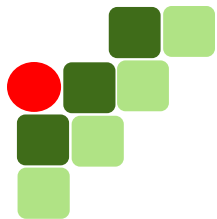
Tratamento de exceções

- ▶ Uma exceção pode ser lançada:
 - Implicitamente pelo servidor Oracle
 - Explicitamente pelo programa
- ▶ Uma exceção pode ser tratada:
 - Por meio de um handler
 - Por meio da propagação para o ambiente de chamada



Exemplo de exceção

```
SET SERVEROUTPUT ON
DECLARE
    lname VARCHAR2(15);
BEGIN
    SELECT last_name INTO lname
    FROM employees
    WHERE first_name='John';
    DBMS_OUTPUT.PUT_LINE ('O sobrenome de John é : '
        || lname);
END;
/
```



Exemplo de exceção

- ▶ No exemplo, não há erros de sintaxe visíveis no código, porém o seguinte erro é apresentado quando o código é executado:

```
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
```

- ▶ Os erros que ocorrem em tempo de execução são chamados de exceções
- ▶ Quando uma exceção ocorre, o bloco PL/SQL é terminado

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
    lname VARCHAR2(15);
```

```
BEGIN
```

```
    SELECT last_name INTO lname
```

```
    FROM employees
```

```
    WHERE first_name='John';
```

```
    DBMS_OUTPUT.PUT_LINE ('O sobrenome de John é : '  
    || lname);
```

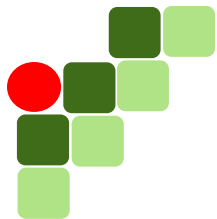
```
EXCEPTION
```

```
    WHEN TOO_MANY_ROWS THEN
```

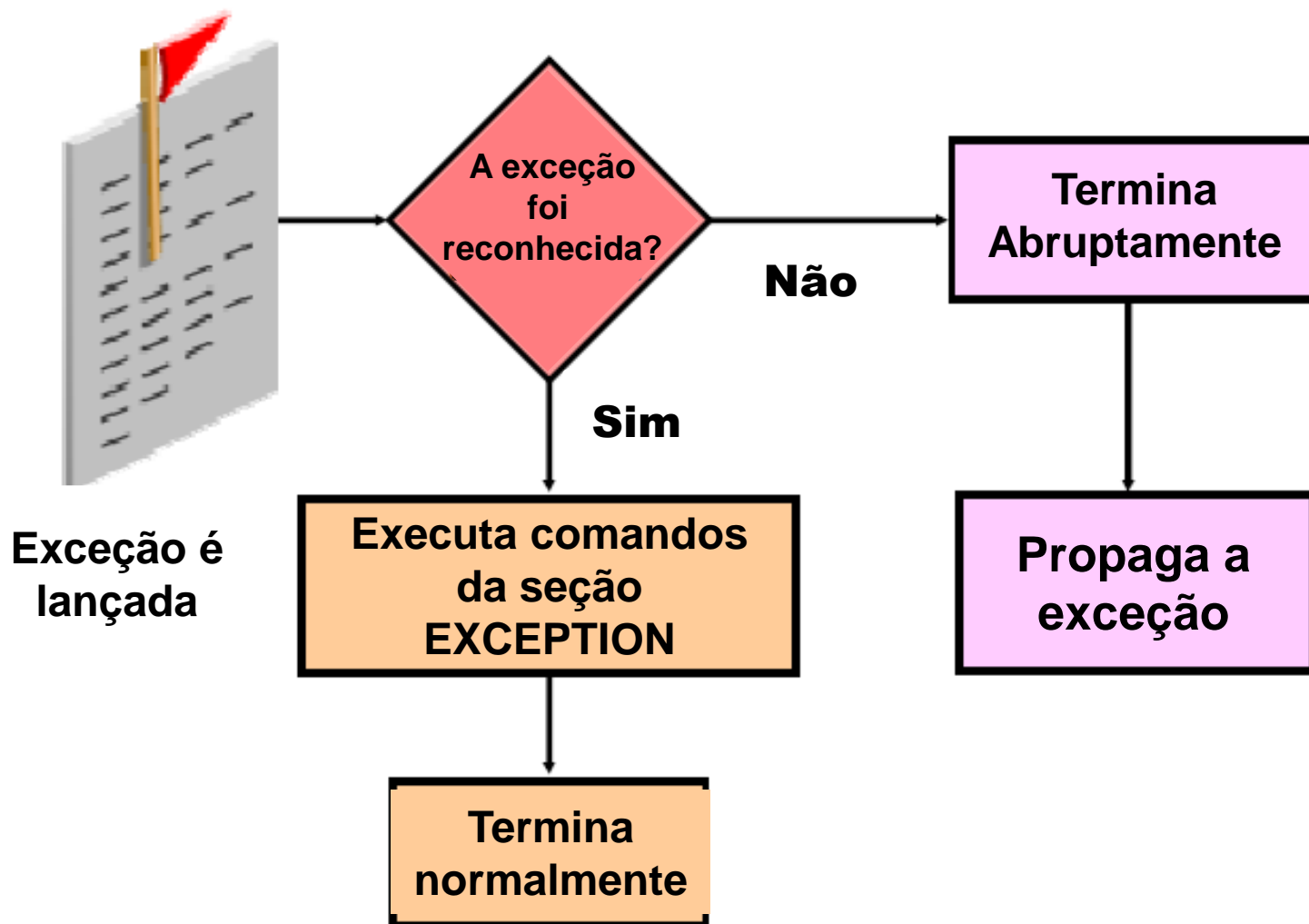
```
        DBMS_OUTPUT.PUT_LINE ('O comando SELECT INTO  
        recuperou muitas tuplas. Considere o uso de um  
        cursor.');
```

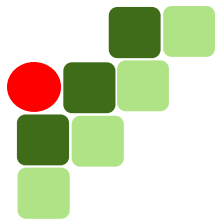
```
END;
```

```
/
```



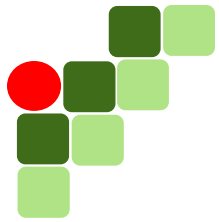
Tratando Exceções





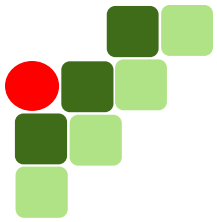
Tratando Exceções

- ▶ **Identificando uma exceção**
 - Incluir uma seção de EXCEPTION em seu programa PL/SQL para reconhecer exceções
 - Se a exceção for lançada na seção executável do bloco de programa, o processamento é transferido ao handler da exceção correspondente no bloco de seção de exceções



Propagando Exceções

- ▶ **Propagando a Exceção**
 - Se a exceção for lançada na seção executável do bloco e não houver handler da exceção correspondente, o bloco PL/SQL terminará com falha e a exceção será propagada para o bloco de fechamento ou para o ambiente de chamada
 - O ambiente de chamada pode ser qualquer aplicação



Sintaxe da seção EXCEPTION

```
EXCEPTION
```

```
  WHEN exception1 [OR exception2 . . .] THEN  
    statement1;  
    statement2;
```

```
  . . .
```

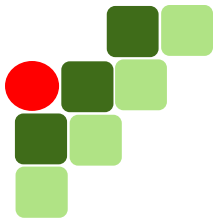
```
[WHEN exception3 [OR exception4 . . .] THEN  
  statement1;  
  statement2;  
  . . .]
```

```
[WHEN OTHERS THEN  
  statement1;  
  statement2;  
  . . .]
```



Sintaxe da seção EXCEPTION

- ▶ A sintaxe para o reconhecimento de uma exceção incluem os seguintes elementos:
 - Exception → é o nome padrão de uma exceção predefinida ou o nome de uma exceção definida pelo usuário e declarada na seção de declaração
 - Statement → um ou mais comandos PL/SQL ou comandos SQL
 - OTHERS → É uma cláusula opcional que reconhece qualquer exceção que não tem handler explícito

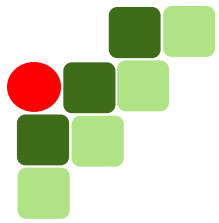


Sintaxe da seção EXCEPTION

► WHEN OTHERS

- Para reconhecer qualquer exceção não específica, use o handler de exceção OTHERS
- Deve ser o último handler de exceção definido

```
WHEN NO_DATA_FOUND THEN
    statement1;
    ...
WHEN TOO_MANY_ROWS THEN
    statement1;
    ...
WHEN OTHERS THEN
    statement1;
```



Tipos de Exceções

Predefinida pelo Servidor Oracle

Não predefinida pelo Servidor Oracle

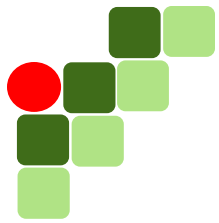


**Lançada
implicitamente**

Definida pelo usuário

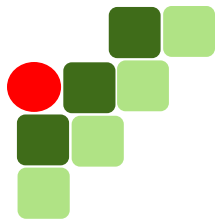


**Lançada
explicitamente**



Tipos de Exceções

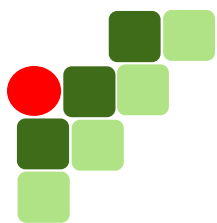
Exceção	Descrição	Direções para tratamento
Erro predefinido pelo Servidor Oracle	Um dos erros que ocorrem mais frequentemente nos códigos PL/SQL	Não é necessário declarar essas exceções. Elas são predefinidas pelo servidor Oracle e são lançadas implicitamente
Erro não predefinido pelo servidor Oracle	Qualquer outro erro padrão do servidor Oracle	É necessário declarar na seção declarativa. O servidor Oracle lança o erro implicitamente, e o erro poderá ser tratado através do handler de exceção
Erro definido pelo usuário	Uma condição que o desenvolvedor determina como anormal	É necessário declarar na seção declarativa e lançá-la explicitamente



Exceções Predefinidas do Oracle

- ▶ Exemplos de exceções predefinidas
 - NO_DATA_FOUND
 - TOO_MANY_ROWS
 - INVALID_CURSOR
 - CURSOR_ALREADY_OPEN
 - ZERO_DIVIDE

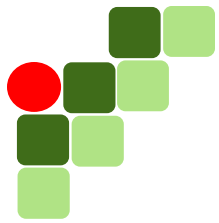
- ▶ Mais exemplos
 - [Predefined PL/SQL Exceptions](#)



Exceções Predefinidas do Oracle

Exception Name	Oracle Server Error Number	Description
ACCESS_INTO_NULL	ORA-06530	Attempted to assign values to the attributes of an uninitialized object
CASE_NOT_FOUND	ORA-06592	None of the choices in the WHEN clauses of a CASE statement are selected, and there is no ELSE clause.
COLLECTION_IS_NULL	ORA-06531	Attempted to apply collection methods other than EXISTS to an uninitialized nested table or VARRAY
CURSOR_ALREADY_OPEN	ORA-06511	Attempted to open an already open cursor
DUP_VAL_ON_INDEX	ORA-00001	Attempted to insert a duplicate value
INVALID_CURSOR	ORA-01001	Illegal cursor operation occurred.
INVALID_NUMBER	ORA-01722	Conversion of character string to number failed.
LOGIN_DENIED	ORA-01017	Logging on to the Oracle server with an invalid username or password
NO_DATA_FOUND	ORA-01403	Single row SELECT returned no data.
NOT_LOGGED_ON	ORA-01012	The PL/SQL program issues a database call without being connected to the Oracle server.
PROGRAM_ERROR	ORA-06501	PL/SQL has an internal problem.
ROWTYPE_MISMATCH	ORA-06504	The host cursor variable and PL/SQL cursor variable involved in an assignment have incompatible return types.

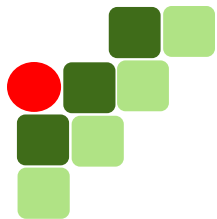
Exception Name	Oracle Server Error Number	Description
STORAGE_ERROR	ORA-06500	PL/SQL ran out of memory, or memory is corrupted.
SUBSCRIPT_BEYOND_COUNT	ORA-06533	Referenced a nested table or VARRAY element by using an index number larger than the number of elements in the collection
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	Referenced a nested table or VARRAY element by using an index number that is outside the legal range (for example, -1)
SYS_INVALID_ROWID	ORA-01410	The conversion of a character string into a universal ROWID fails because the character string does not represent a valid ROWID.
TIMEOUT_ON_RESOURCE	ORA-00051	Time-out occurred while the Oracle server was waiting for a resource.
TOO_MANY_ROWS	ORA-01422	Single-row SELECT returned multiple rows.
VALUE_ERROR	ORA-06502	Arithmetic, conversion, truncation, or size-constraint error occurred.
ZERO_DIVIDE	ORA-01476	Attempted to divide by zero



Funções para reconhecer exceções

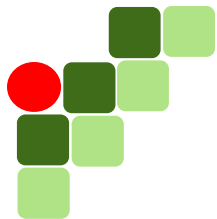
- ▶ Quando uma exceção ocorre, pode-se identificar o código do erro associado ou a mensagem de erro usando duas funções:
 - **SQLCODE**: Retorna o valor numérico para o código do erro.
 - Pode-se associar ele a uma variável do tipo **NUMBER**
 - **SQLERRM**: Retorna a mensagem associada com o número do erro

```
CREATE TABLE ERRORS (  
    e_date DATE,  
    error_code NUMBER,  
    error_msg VARCHAR2(255));  
  
DECLARE  
    errcode NUMBER;  
    errmsg VARCHAR2(255);  
  
BEGIN  
    ...  
EXCEPTION  
    ...  
WHEN OTHERS THEN  
    errcode := SQLCODE ;  
    errmsg := SQLERRM ;  
    INSERT INTO errors (e_date, error_code, error_msg)  
    VALUES (SYSDATE, errcode, errmsg);  
  
END;  
/
```



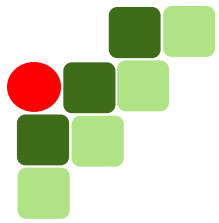
Exercícios

1. Defina um procedimento usando o comando **SELECT INTO**, que imprima na tela o nome completo de um empregado, dado o seu sobrenome como parâmetro de entrada
 - a) Faça o tratamento de exceção para que o procedimento imprima na tela a mensagem: “Nenhum empregado encontrado com o sobrenome XXX”, caso nenhum empregado seja encontrado
 - b) Faça o tratamento de exceção para apresentar a mensagem: “O comando retornou mais linhas do que o permitido. Considere o uso de um cursor”, caso a consulta retorne mais de uma tupla

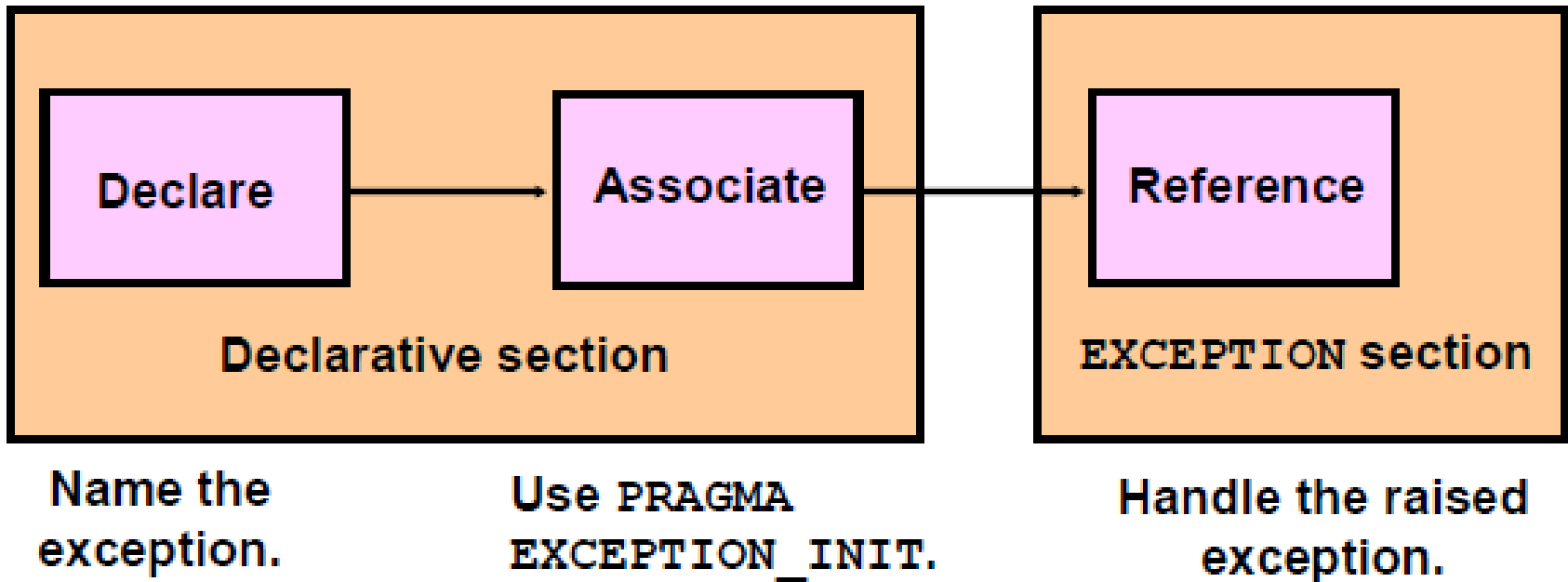


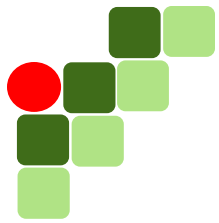
Exercícios

2. **Altere o exercício 1 para que, caso algum outro erro ocorra, sejam exibidos na tela o SQLCODE e o SQLERRM do erro.**
 - **Altere o WHERE do exercício 1 para last_name = 0. Qual o erro obtido?**



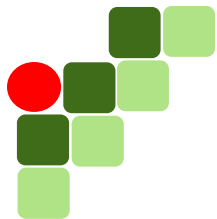
Erros Não Predefinidos do Servidor Oracle





Erros Não Predefinidos no Servidor Oracle

- ▶ Para criar exceções com erros não predefinidos pode-se usar a função `PRAGMA EXCEPTION_INIT`
- ▶ `PRAGMA EXCEPTION_INIT` diz ao compilador para associar um nome de exceção com um número de erro
 - Permite referenciar qualquer exceção interna pelo nome e designar um handler específico para ele



Erros Não Predefinidos

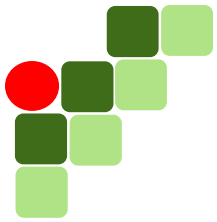
- ▶ Três passos para lançar erro não predefinido:
 1. Declarar o nome da exceção na seção declarative (DECLARE), usando a sintaxe:

exception_name EXCEPTION;

2. Associar a exceção declarada com o erro padrão do servidor Oracle usando a função PRAGMA EXCEPTION_INIT. Use a sintaxe:

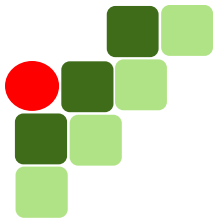
**PRAGMA EXCEPTION_INIT(exception_name, error
number) ;**

3. Referenciar a exceção na seção EXCEPTION



Exemplo: erro não predefinido

- ▶ Lançar o erro de servidor Oracle -01400 (“cannot insert NULL”)
 - Tentar inserir valor NULL na coluna department_name da tabela departments
 - Essa operação não é bem sucedida pois department_name é uma coluna NOT NULL
- ▶ Para exibir a mensagem de erro:
`DBMS_OUTPUT.PUT_LINE (SQLERRM) ;`



Exemplo: erro não predefinido

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
    insert_excep EXCEPTION;
```

```
    PRAGMA EXCEPTION_INIT(insert_excep, -01400);
```

```
BEGIN
```

```
    INSERT INTO departments(department_id,  
        department_name) VALUES (280, NULL);
```

```
EXCEPTION
```

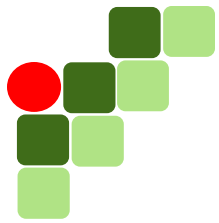
```
    WHEN insert_excep THEN
```

```
        DBMS_OUTPUT.PUT_LINE('INSERT OPERATION FAILED');
```

```
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
```

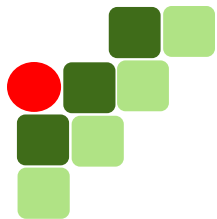
```
END;
```

```
/
```



Listagem de erros do Oracle

- ▶ Database Error Messages
 - ORA-00910 to ORA-01497



Exercício

- ▶ Criar uma tabela de log de erros

```
CREATE TABLE ERRORS (  
    e_date DATE,  
    error_code NUMBER,  
    error_msg VARCHAR2(255) );
```

- ▶ Criar um bloco anônimo que trate o erro
 - ORA-01438: value larger than specified precision allowed for this column
- ▶ Utilize alguma coluna de tipo NUMBER e insira um valor maior que a capacidade dela
- ▶ Inserir os dados relativos ao erro na tabela ERRORS

DECLARE

errcode NUMBER;

errmsg VARCHAR2(255);

insert_excep2 EXCEPTION;

PRAGMA EXCEPTION_INIT(insert_excep2, -01438);

BEGIN

INSERT INTO departments(department_id,
department_name) VALUES (11111, 'teste');

EXCEPTION

WHEN insert_excep2 THEN

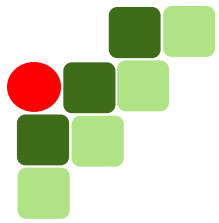
errcode := SQLCODE;

errmsg := SQLERRM;

INSERT INTO errors (e_date, error_code, error_msg)
VALUES (SYSDATE, errcode, errmsg);

END;

/



Exceções Definidas pelo Usuário

- ▶ **PL/SQL permite que o usuário defina suas próprias exceções**
- ▶ **Passos:**
 - 1. Declarar o nome da exceção definida pelo usuário na seção declarativa;**
 - 2. Usar o comando RAISE para levantar a exceção explicitamente na seção executável do bloco;**
 - 3. Referenciar a exceção declarada com a rotina de manipulação correspondente.**

DECLARE

invalid_department EXCEPTION;

name VARCHAR2(20) := '&name';

deptno NUMBER := &deptno;

BEGIN

UPDATE departments

SET department_name = name

WHERE department_id = deptno;

IF SQL%NOTFOUND THEN

RAISE invalid_department;

END IF;

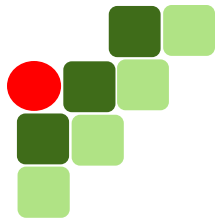
EXCEPTION

WHEN **invalid_department** THEN

DBMS_OUTPUT.PUT_LINE('Não existe o department id informado');

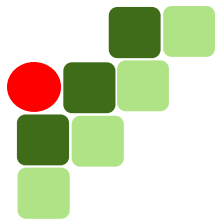
END;

/



Exercício

- ▶ Faça um procedimento que atualize o salário de um empregado, sendo que o id do empregado e o novo salário são fornecidos como parâmetros de entrada
 - O salário somente pode ser atualizado se for menor que o maior salário da função do empregado
 - Caso o salário esteja fora dos limites, lance uma exceção chamada `salary_too_high` que informe que o salário está acima do limite permitido



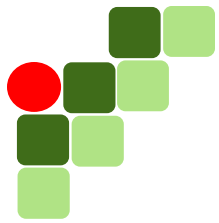
Procedure

RAISE_APPLICATION_ERROR

- Procedimento para emitir mensagens de erro definidas pelo usuário em subprogramas armazenados

- Sintaxe:

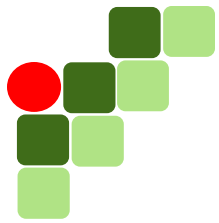
```
raise_application_error(error_number, message  
[, {TRUE | FALSE}]);
```



Procedure RAISE_APPLICATION_ERROR

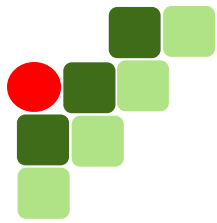
```
raise_application_error(error_number, message  
[, {TRUE | FALSE}]);
```

Item	Descrição
Error_number:	é um número específico para o erro no intervalo de números inteiros negativos de -20000..-20999
Message	É uma mensagem especificada pelo usuário para o erro; é uma cadeia com até 2.048 caracteres
TRUE FALSE	É um parâmetro booleano opcional (Se TRUE, o erro é colocado na pilha de erros anteriores. Se FALSE, que é o default, o erro substitui todos os erros anteriores.



Procedure RAISE_APPLICATION_ERROR

- ▶ Pode ser usado em dois locais
 - Seção executável (entre BEGIN e END)
 - Seção de exceções (EXCEPTION)
- ▶ Permite retornar mensagens de erro de uma forma consistente com outros erros do servidor Oracle



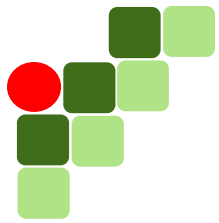
Procedure RAISE_APPLICATION_ERROR

SEÇÃO EXECUTÁVEL

```
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 EXCEÇÃO

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



Procedure RAISE_APPLICATION_ERROR

```
DECLARE
```

```
    e_name EXCEPTION;
```

```
BEGIN
```

```
    DELETE FROM employees
```

```
    WHERE last_name = 'Damon';
```

```
    IF SQL%NOTFOUND THEN
```

```
        RAISE e_name;
```

```
    END IF;
```

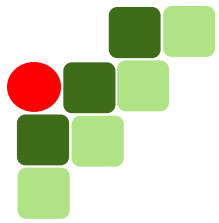
```
EXCEPTION
```

```
    WHEN e_name THEN
```

```
        RAISE_APPLICATION_ERROR(-20999, 'O sobrenome não é  
válido');
```

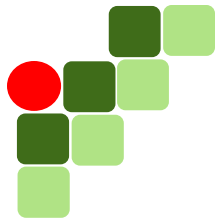
```
END;
```

```
/
```



Exercício

- ▶ Desenvolva um procedimento que divida o salário de um funcionário pelo número de membros de sua família e exiba o resultado na tela. Os parâmetros de entrada devem ser o id do empregado e o número de membros da família
 - Faça o tratamento de exceção caso haja erro de divisão por zero (-01476). Neste caso, exiba mensagem de que não há membros na família.
 - Caso o número de membros da família seja um valor negativo, exiba uma mensagem de erro usando `RAISE_APPLICATION_ERROR` informando que números negativos são inválidos.



Referências

- ▶ Oracle Database PL/SQL Language Reference
 - Handling PL/SQL Errors