**FUNDAÇÃO OSWALDO ARANHA**

**CENTRO UNIVERSITÁRIO DE VOLTA REDONDA**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**RELATÓRIO TÉCNICO-CIENTÍFICO**

**EXCEÇÕES EM PL/SQL**

Autores

**CAIO DOMINGUES DA SILVA SANTOS**

**THOBIAS VICENTE DE SOUZA**

Orientador

**ROSENCLEVER LOPES GAZONI**

**Volta Redonda**

**Outubro/2019**

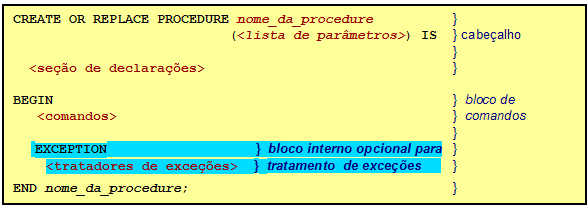
1. INTRODUÇÃO

Em PL/SQL, qualquer erro é uma exceção (*exception*), isto é, algo que não deveria ocorrer durante a execução do programa. Existem dois tipos de exceção:

* 1. **De sistema**: um erro definido pela Oracle, disparado pela *runtime engine* da PL/SQL, como “valores duplicados no index” ou “dado não encontrado”
  2. **Programada**: definida pelo programador, específica da aplicação. É disparada apenas em situações indicadas pelo código, e não automaticamente como as do sistema.

1. **O BLOCO**

As exceções são tratadas no bloco opcional “EXCEPTION” em procedures ou funções:



1. **TIPOS**

Para os exemplos, considere a utilização do Banco de Dados HR, disponível por padrão no *SQLDeveloper* e as saídas do servidor como: “SET SERVEROUTPUT ON”.

* 1. **Exceções comuns (principais)**
     1. NO\_DATA\_FOUND

Quando a query não retornou linhas. Exemplo: quando um SELECT INTO não retorna linhas.

CREATE OR REPLACE PROCEDURE select\_item (

t\_column VARCHAR2,

t\_name VARCHAR2

) AUTHID DEFINER

IS

temp VARCHAR2(30);

BEGIN

temp := t\_column;

SELECT COLUMN\_NAME INTO temp FROM USER\_TAB\_COLS WHERE TABLE\_NAME = UPPER(t\_name) AND COLUMN\_NAME = UPPER(t\_column);

temp := t\_name;

SELECT OBJECT\_NAME INTO temp FROM USER\_OBJECTS WHERE OBJECT\_NAME = UPPER(t\_name) AND OBJECT\_TYPE = 'TABLE';

EXCEPTION

**WHEN NO\_DATA\_FOUND THEN**

**DBMS\_OUTPUT.PUT\_LINE ('No Data found for SELECT on ' || temp);**

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('Unexpected error');

RAISE;

END;

/

BEGIN

select\_item(employees, 'nome\_de\_coluna\_que\_nao\_existe');

END;

/

* + 1. TOO\_MANY\_VALUES

Quando a inserção tenta inserir mais dados que a tabela possui em colunas. Exemplo:

BEGIN

INSERT INTO employees (21231, 'nome', 'sobrenome', 'email', '123', '17/10/19', '2', '1232', '1', '1', '1', 'este\_dado\_e\_uma\_coluna\_a\_mais');

END;

/

* + 1. DUP\_VAL\_ON\_INDEX

Quando a query tentou inserir uma chave primária já existente. Exemplo: considerando que já existe o funcionário com o ID = 1:

BEGIN

INSERT INTO employees (1, 'nome', 'sobrenome', 'email', '123', '17/10/19', '2', '1232', '1', '1', '1');

END;

/

* + 1. OTHERS

Genérica para tratamento de demais erros. Exemplo: reutilizando o código da procedure no primeiro caso:

BEGIN

select\_item(employees, 'nome\_de\_coluna\_que\_nao\_existe');

EXCEPTION

WHEN OTHERS THEN

DMBS\_OUTPUT.PUT\_LINE('OCORREU UM ERRO');

END;

/

* 1. **OUTRAS EXCEÇÕES**
     1. CURSOR\_ALREADY\_OPEN

Quando tenta abrir um cursor já aberto.

* + 1. INVALID\_CURSOR

Quando o cursor é inválido (não foi encontrado ou não existe)

* + 1. INVALID\_NUMBER

Quando o número é inválido, geralmente ocorre nas situações onde uma *String* de um número é passada para um campo/atributo/parâmetro de número.

* + 1. LOGIN\_DENIED

Quando o usuário e/ou senha foram recusados ao tentar realizar conexão com o banco de dados.

* + 1. NOT\_LOGGED\_ON

Quando se tenta executar uma ação que necessita de estar logado.

* + 1. PROGRAM\_ERROR

Erros de compilação do PL/SQL não identificados (geralmente erros na *runtime* *engine*).

* + 1. ROWTYPE\_MISMATCH

Quando o tipo de dado das linhas não é equivalente à query em questão.

* + 1. TOO\_MANY\_ROWS

Quando a query retornou muitas linhas.

* + 1. ZERO\_DIVIDE

Quando se tenta fazer uma divisão por zero.

1. **CONCLUSÃO**

As exceções são importantes para fornecerem um retorno legível e tratado ao usuário, desta forma, evita-se a exposição de informações que podem ser importantes (principalmente em sistemas Web onde o dado de *log* pode ser retornado diretamente ao usuário, expondo assim informações sensíveis do banco de dados).

Sendo assim, o tratamento de exceções além de ser mais agradável aos olhos, mantém os dados seguro, informa o tipo de erro e dependendo dos casos, a solução para eles, como um usuário preenchendo algo de forma incorreta.

Tratar as exceções é quase, senão, obrigatório para sistemas em ambiente de produção, tendo de aparecerem já mesmo no estágio de homologação, de forma que o *Quality Analist* ou *Tester* possam ver as exceções já tratadas.

**REFERÊNCIAS**

CIFERRI, Cristina D. A. Tratamento de Exceções. *In*: CIFERRI, Cristina D. A. **Tratamento de Exceções**. 07. ed. Brasil: USP, 15 jul. 2014. Disponível em: http://wiki.icmc.usp.br/images/c/c4/Mat07\_excecao.pdf. Acesso em: 17 out. 2019.

GONÇALVES, Eduardo Correa. Ratamento de Exceções de Sistema na linguagem PL/SQL. *In*: GONÇALVES, Eduardo Correa. **Tratamento de Exceções de Sistema na linguagem PL/SQL**. 1. ed. Brasil: DevMedia, 2012. Disponível em: https://www.devmedia.com.br/tratamento-de-excecoes-de-sistema-na-linguagem-pl-sql/26001. Acesso em: 17 out. 2019.