

Tratamento de Exceções

Laboratório de Bases de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Tratamento de Exceções

- Exceções
 - erros e imprevistos que podem ocorrer durante a execução de um bloco PL/SQL
- SGBD
 - ao encontrar um erro ou um imprevisto, aborta a execução e procura por uma área de exceções
- Tipos
 - predefinidas
 - definidas pelo usuário

Comando

EXCEPTION

WHEN nome_da_exceção

THEN relação de comandos;

WHEN nome_da_exceção

THEN relação de comandos;

Exemplo

```
set serveroutput on;
DECLARE
    nome aluno.nome_aluno%Type;
    sexo aluno.sexo_aluno%Type;
BEGIN
    nome := 'Pedro'; sexo := 'm';
    INSERT INTO aluno VALUES (1, nome, sexo);
    dbms_output.put_line ('Inserção executada com sucesso.');
```

EXCEPTION

```
    WHEN Dup_Val_On_Index
        THEN dbms_output.put_line ('Aluno já cadastrado. ');
    WHEN Others
        THEN dbms_output.put_line ('Erro no cadastramento. ');
END;
```

Exceções Predefinidas

- **CURSOR_ALREADY_OPEN**
 - ocorre quando se tenta abrir um cursor que já está aberto
- **DUP_VAL_ON_INDEX**
 - ocorre quando se tenta armazenar um valor duplicado em uma coluna de uma tabela que possui chave única ou primária
- **INVALID_CURSOR**
 - ocorre quando se tenta realizar uma operação ilegal em um cursor

Exceções Predefinidas

- **INVALID_NUMBER**
 - ocorre na tentativa de converter uma *string* para um número, quando a *string* não representa um número válido
- **LOGIN_DENIED**
 - ocorre na tentativa de conexão com o BD com um usuário/senha inválido
- **NO_DATA_FOUND**
 - ocorre quando o comando **SELECT ... INTO** não retorna nenhuma linha

Exceções Predefinidas

- **NO_LOGGED_ON**
 - ocorre na tentativa de acessar o banco de dados sem que se esteja conectado a ele
- **PROGRAM_ERROR**
 - ocorre em caso de problemas internos do PL/SQL
- **ROWTYPE_MISMATCH**
 - ocorre se o retorno do cursor e a variável PL/SQL para retorno do cursor forem de tipos incompatíveis

Exceções Predefinidas

- **STORAGE_ERROR**
 - ocorre se não houver memória suficiente para a execução de um bloco PL/SQL ou caso a memória esteja com problemas
- **TIMEOUT_ON_RESOURCE**
 - ocorre quando acontecer um *timeout* enquanto aguarda-se um recurso
- **TOO_MANY_ROWS**
 - ocorre quando um comando **SELECT ... INTO** retornar mais de uma linha

Exceções Predefinidas

- `VALUE_ERROR`
 - ocorre quando houver um erro aritmético, de conversão, truncagem ou tamanho
- `ZERO_DIVIDE`
 - ocorre na tentativa de dividir qualquer número por zero
- `OTHERS` “Look up an error message”
 - trata outros erros, usando as funções `SQL_CODE` (número do erro) e `SQLERRM` (texto do erro)

Exceções Definidas pelo Usuário

- Características
 - precisam ser declaradas e chamadas explicitamente
- Declaração
 - realizada na área de declaração
 - sintaxe: `nome_da_exceção EXCEPTION`
- Utilização
 - realizada na área de comandos
 - sintaxe: `RAISE nome_da_exceção`

Exceções Definidas pelo Usuário

DECLARE

nome_exceção **EXCEPTION**;

exceção tem que ser declarada!

BEGIN

relação_de_comandos;

IF ...

THEN **RAISE** nome_exceção;

END IF;

relação_de_comandos;

EXCEPTION

WHEN nome_exceção

THEN relação_de_comandos;

END;

Exemplo (1/2)

```
set serveroutput on;
DECLARE
  nusp aluno.NUSP%Type;
  nome aluno.nome_aluno%Type;
  sexo aluno.sexo_aluno%Type;
  valida_campos EXCEPTION;
BEGIN
  nusp := 4;
  nome := 'Daniel';
  -- sexo := 'm';
  IF (nusp IS NULL) OR (sexo IS NULL) OR (nome IS NULL)
    THEN RAISE valida_campos;
    ELSE INSERT INTO aluno VALUES (nusp, nome, sexo);
  END IF;
```

Exemplo (2/2)

EXCEPTION

WHEN valida_campos

THEN dbms_output.put_line ('Preencher todos os campos.');

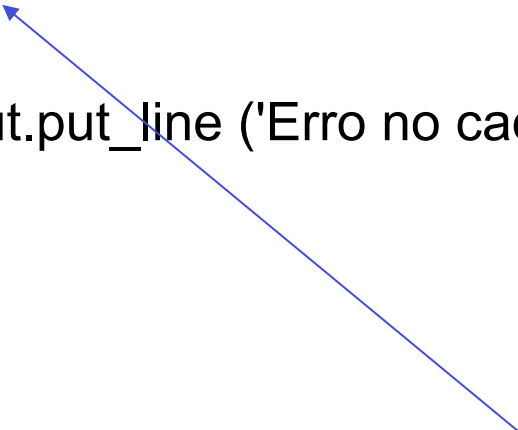
WHEN Dup_Val_On_Index

THEN dbms_output.put_line ('Aluno já cadastrado.');

WHEN Others

THEN dbms_output.put_line ('Erro no cadastramento.');

END;



Diferentes tipos de exceção
podem ser combinadas na
área de exceções !

EXCEPTION_INIT

- Manipulação de exceção sem nome
 - exceção OTHERS
 - PRAGMA EXCEPTION_INIT
- Pragma
 - **renomeia** um erro Oracle identificado por seu número
 - permite que a exceção seja referenciada pelo novo nome
 - diretiva de compilação

COMANDO

DECLARE

nome_exceção **EXCEPTION**;

PRAGMA EXCEPTION_INIT (nome_exceção, código_erro);

BEGIN

relação_de_comandos;

...

EXCEPTION

WHEN nome_exceção

THEN relação_de_comandos;

END;

Exemplo (1/2)

```
set serveroutput on;
```

```
DECLARE
```

```
  nusp aluno.NUSP%Type;
```

```
  nome aluno.nome_aluno%Type;
```

```
  sexo aluno.sexo_aluno%Type;
```

```
  obrigatorio EXCEPTION;
```

```
  PRAGMA EXCEPTION_INIT (obrigatorio, -1400);
```

```
BEGIN
```

```
  nusp := 4;
```

```
  nome := 'Daniel';
```

```
  -- sexo := 'm';
```

```
  INSERT INTO aluno VALUES (nusp, nome, sexo);
```


Exemplo (2/2)

EXCEPTION

WHEN obrigatorio

THEN dbms_output.put_line ('Preencher todos os campos.');

WHEN Dup_Val_On_Index

THEN dbms_output.put_line ('Aluno já cadastrado.');

WHEN Others

THEN dbms_output.put_line ('Erro no cadastramento.');

END;

Diferentes tipos de exceção
podem ser combinadas na
área de exceções !