

FIAP GRADUAÇÃO

Tecnologia em Análise e Desenvolvimento de Sistemas

Application Development For Databases

PROF. MILTON

Tratando Exceções

Objetivos

Ao concluir esta lição, você será capaz de:

- Definir exceções PL/SQL
- Reconhecer exceções não resolvidas
- Listar e usar diferentes tipos de handlers de exceções PL/SQL
- Interceptar erros inesperados
- Descrever o efeito da propagação da exceção em blocos aninhados
- Personalizar mensagens de exceção PL/SQL

Objetivos

Você aprendeu a criar blocos PL/SQL contendo uma seção declarativa e uma seção executável. Todos os códigos SQL e PL/SQL que precisam ser executados são gravados no bloco executável.

Até agora, pressupomos que o código funcionará bem se tomarmos cuidado com os erros que ocorrem no momento da compilação. No entanto, o código pode causar alguns erros inesperados durante o runtime. Nesta lição, você aprenderá a tratar esses erros no bloco PL/SQL.

Agenda

- Fundamentos das exceções PL/SQL
- Interceptando exceções

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

6

O Que É uma Exceção?

Examine o exemplo mostrado no slide. Não existem erros de sintaxe no código, o que significa que você conseguirá executar o bloco anônimo com sucesso. A instrução `SELECT` do bloco recupera o sobrenome John.

Entretanto, você verá o seguinte relatório de erro ao executar o código:

```
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
```

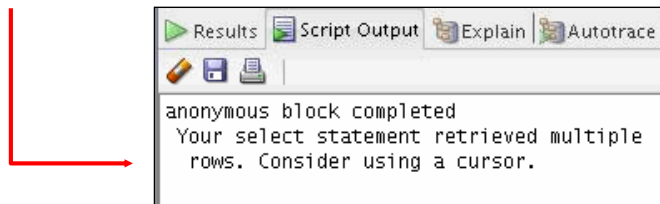
O código não funciona como esperado. Você esperava que a instrução `SELECT` recuperasse apenas uma linha; entretanto, ela recupera várias linhas. Esses erros que ocorrem durante o runtime são denominados *exceções*. Se ocorrer uma exceção, o bloco PL/SQL será encerrado. É possível tratar essas exceções no bloco PL/SQL.

Tratando a Exceção: 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 (' Your select statement retrieved
        multiple rows. Consider using a cursor. ');
END;
/

```



7

Tratando a Exceção: Exemplo

Você aprendeu a criar blocos PL/SQL contendo uma seção declarativa (iniciando com a palavra-chave `DECLARE`) e uma seção executável (iniciando e terminando com as palavras-chave `BEGIN` e `END`, respectivamente).

Para o tratamento de exceções, inclua outra seção opcional chamada *seção de exceções*.

- Essa seção começa com a palavra-chave `EXCEPTION`.
- Quando presente, essa deverá ser a última seção de um bloco PL/SQL.

Exemplo

No exemplo do slide, o código do slide anterior foi reescrito para tratar a exceção gerada. A saída do código também é mostrada no slide.

Quando você adiciona a seção `EXCEPTION` do código, o programa PL/SQL não é encerrado abruptamente. Se a exceção for gerada, o controle passará para a seção de exceções e todas as instruções dessa seção serão executadas. O bloco PL/SQL será encerrado normalmente, com sucesso.

Fundamentos das Exceções com PL/SQL ^{FIAP}

- Uma exceção é um erro PL/SQL gerado durante a execução de um programa.
- Uma exceção pode ser gerada:
 - Implicitamente pelo Oracle Server
 - Explicitamente pelo programa
- Uma exceção pode ser tratada:
 - Ao ser interceptada por um handler
 - Ao ser propagada para o ambiente de chamada

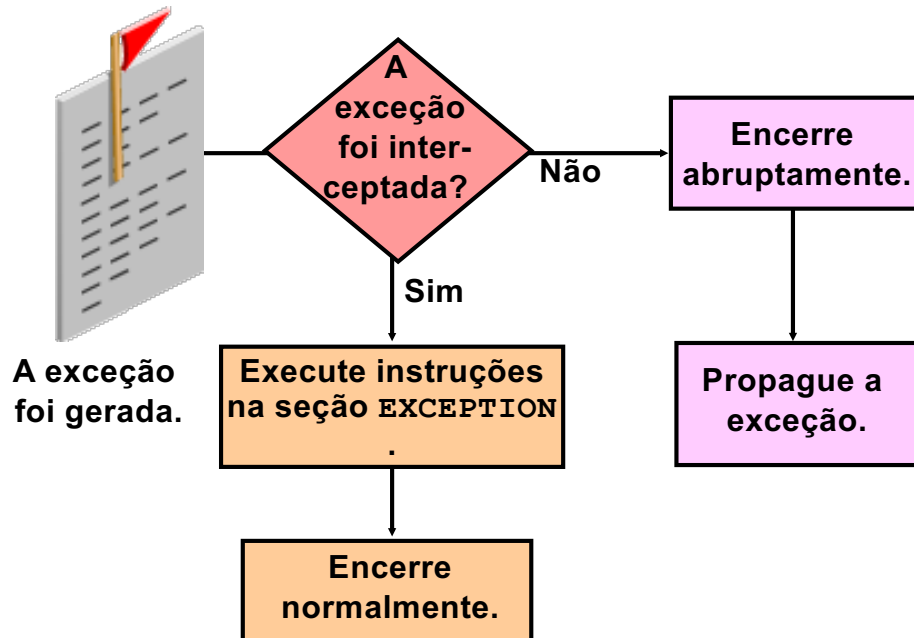
Fundamentos das Exceções com PL/SQL

Uma exceção é um erro no código PL/SQL gerado durante a execução de um bloco. Um bloco é sempre encerrado quando o código PL/SQL gera uma exceção, mas você pode especificar um handler de exceção para executar ações finais antes do encerramento do bloco.

Dois Métodos para Gerar uma Exceção

- Um erro Oracle ocorre, e a exceção associada é gerada automaticamente. Por exemplo, se o erro ORA-01403 ocorrer quando nenhuma linha for recuperada do banco de dados em uma instrução `SELECT`, o código PL/SQL gerará a exceção `NO_DATA_FOUND`. Esses erros são convertidos em exceções predefinidas.
- Dependendo da funcionalidade de negócios que o seu programa estiver implementando, será necessário gerar uma exceção explicitamente. Para gerar uma exceção explicitamente, execute a instrução `RAISE` no bloco. A exceção gerada pode ser predefinida ou definida pelo usuário. Existem alguns erros Oracle não predefinidos. Esses são os erros Oracle padrão não predefinidos. Você pode declarar explicitamente as exceções e associá-las a erros Oracle não predefinidos.

Tratando Exceções



Tratando Exceções

Interceptando uma Exceção

Inclua uma seção `EXCEPTION` no programa PL/SQL para interceptar exceções. Se a exceção for gerada na seção executável do bloco, o processamento será ramificado para o handler de exceção correspondente na seção de exceções do bloco. Se o código PL/SQL tratar a exceção com sucesso, ela não será propagada para o bloco que a contém ou para o ambiente de chamada. O bloco PL/SQL será encerrado com sucesso.

Propagando uma Exceção

Se a exceção for gerada na seção executável do bloco e não houver um handler de exceções correspondente, o bloco PL/SQL será encerrado com falha e a exceção se propagará a um bloco que o contém ou ao ambiente de chamada. O ambiente de chamada pode ser qualquer aplicação, como o SQL*Plus, que chame o programa PL/SQL.

Tipos de Exceções

- Predefinidas no Oracle Server
 - Não Predefinidas no Oracle Server
- Geradas implicitamente**
-
- Definidas pelo usuário
- Geradas explicitamente**

10

Tipos de Exceções

Há três tipos de exceções.

Exceção	Descrição	Instruções para Tratamento
Erro predefinido do Oracle Server	Um dos, aproximadamente, 20 erros que ocorrem com mais frequência na linguagem PL/SQL	Você não precisa declarar essas exceções. Elas são predefinidas pelo Oracle Server e são emitidas implicitamente.
Erro não predefinido do Oracle Server	Qualquer outro erro padrão do Oracle Server	Você precisa declará-los na seção declarativa; o servidor Oracle gera o erro implicitamente, e você pode detectar o erro no handler de exceção.
Erro definido pelo usuário	Uma condição que o desenvolvedor definiu como anormal	Você precisa declarar dentro da seção declarativa e emitir explicitamente.

Observação: Algumas ferramentas de aplicações com PL/SQL no lado cliente (como o Oracle Developer Forms) possuem suas próprias exceções.

Agenda

- Fundamentos das exceções PL/SQL
- Interceptando exceções

Sintaxe para Interceptar Exceções

```

EXCEPTION
  WHEN exception1 [OR exception2 . . .] THEN
    statement1;
    statement2;
    . . .
  [WHEN exception3 [OR exception4 . . .] THEN
    statement1;
    statement2;
    . . .]
  [WHEN OTHERS THEN
    statement1;
    statement2;
    . . .]

```

12

Sintaxe para Interceptar Exceções

É possível interceptar qualquer erro, incluindo um handler correspondente dentro da seção de tratamento de exceções do bloco PL/SQL. Cada handler consiste em uma cláusula **WHEN**, que especifica um nome de exceção, seguida de uma sequência de instruções a serem executadas quando essa exceção for gerada.

É possível incluir inúmeros handlers dentro de uma seção **EXCEPTION** para tratar exceções específicas. No entanto, não é possível ter vários handlers para uma única exceção.

A sintaxe de interceptação de exceções contém os seguintes elementos:

<i>exception</i>	É o nome padrão de uma exceção predefinida ou o nome de uma exceção definida pelo usuário declarada dentro da seção declarativa
<i>statement</i>	É uma ou mais instruções PL/SQL ou SQL
OTHERS	É uma cláusula de tratamento de exceções opcional que intercepta qualquer exceção que não foi explicitamente tratada

Diretrizes para Interceptar Exceções

FIAP

- A palavra-chave `EXCEPTION` inicia a seção de tratamento de exceções.
- São permitidos vários handlers de exceções.
- Apenas um handler será processado antes de deixar o bloco.
- `WHEN OTHERS` é a última cláusula.

14

Diretrizes para Interceptar Exceções

- Inicie a seção de tratamento de exceções do bloco com a palavra-chave `EXCEPTION`.
- Defina vários handlers de exceções, cada um com seu próprio conjunto de ações, para o bloco.
- Se ocorrer uma exceção, o código PL/SQL processará apenas um handler antes de deixar o bloco.
- Coloque a cláusula `OTHERS` depois de todas as outras cláusulas de tratamento de exceções.
- Você só pode ter uma cláusula `OTHERS`.
- As exceções não podem aparecer em instruções de atribuição ou instruções SQL.

Interceptando Erros Predefinidos do Servidor Oracle

- Faça referência ao nome predefinido na rotina de tratamento de exceções.
- Exemplos de exceções predefinidas:
 - NO_DATA_FOUND
 - TOO_MANY_ROWS
 - INVALID_CURSOR
 - ZERO_DIVIDE
 - DUP_VAL_ON_INDEX

15

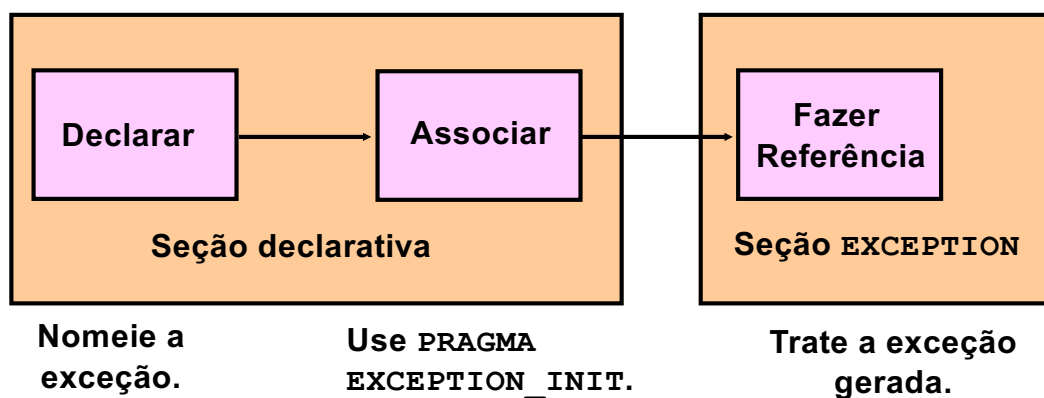
Interceptando Erros Predefinidos do Servidor Oracle

Intercepte um erro predefinido do Oracle Server fazendo referência a seu nome predefinido na rotina de tratamento de exceções correspondente.

Para obter uma lista completa das exceções predefinidas, consulte o manual *PL/SQL User's Guide and Reference*.

Observação: O código PL/SQL declara as exceções predefinidas no pacote STANDARD.

Interceptando Erros Não Predefinidos do Servidor Oracle



Interceptando Erros Não Predefinidos do Servidor Oracle

Exceções não predefinidas são semelhantes às predefinidas; contudo, elas não são definidas como exceções PL/SQL no Oracle Server. Elas são os erros Oracle padrão. Você pode criar exceções com os erros Oracle padrão usando a função `PRAGMA EXCEPTION_INIT`. Essas exceções são denominadas exceções não predefinidas.

Declare um erro não predefinido do Oracle Server para interceptá-lo. A exceção declarada será gerada implicitamente. No código PL/SQL, a função `PRAGMA EXCEPTION_INIT` instrui o compilador a associar um nome de exceção a um número de erro Oracle. Isso permite que você faça referência a qualquer exceção interna pelo nome e crie um handler específico para ela.

Observação: `PRAGMA` (também chamada *pseudoinstruções*) é a palavra-chave que identifica a instrução como uma diretiva de compilador, que não será processada quando o bloco PL/SQL for executado. Em vez disso, ela direciona o compilador PL/SQL a interpretar todas as ocorrências do nome da exceção dentro do bloco como o número de erro associado ao Oracle Server.

Interceptação de Erro Não Predefinido: Exemplo

Para interceptar o erro do Oracle Server 01400 (“cannot insert NULL”):

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

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

anonymous block completed
INSERT OPERATION FAILED
ORA-01400: cannot insert NULL into ("ORA41"."DEPARTMENTS"."DEPARTMENT_NAME")

19

Interceptação de Erro Não Predefinido: Exemplo

O exemplo ilustra as três etapas associadas à interceptação de um erro não predefinido:

1. Declare o nome da exceção na seção declarativa, usando a sintaxe:

exception EXCEPTION;

Na sintaxe, *exception* é o nome da exceção.

2. Associe a exceção declarada ao número de erro padrão do Oracle Server usando a função PRAGMA EXCEPTION_INIT. Use a seguinte sintaxe:

PRAGMA EXCEPTION_INIT(*exception*, *error_number*);

Na sintaxe, *exception* é a exceção declarada anteriormente e *error_number* é um número de erro padrão do Oracle Server.

3. Faça referência à exceção declarada na rotina correspondente de tratamento de exceções.

Exemplo

O exemplo do slide tenta inserir o valor NULL para a coluna `department_name` da tabela `departments`. Entretanto, a operação não é bem-sucedida porque `department_name` é uma coluna NOT NULL. Observe a seguinte linha do exemplo:

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

A função `SQLERRM` é usada para recuperar a mensagem de erro. Você aprenderá mais sobre `SQLERRM` nos próximos slides.

Funções para Interceptar Exceções

- **SQLCODE:** Retorna o valor numérico para o código de erro
- **SQLERRM:** Retorna a mensagem associada ao número de erro

Funções para Interceptar Exceções

Se ocorrer uma exceção, você poderá identificar o código de erro associado ou a mensagem de erro associada usando duas funções. Com base nos valores do código ou na mensagem, você poderá decidir quais as ações subsequentes a serem tomadas.

SQLCODE retorna o número de erro Oracle para exceções internas. **SQLERRM** retorna a mensagem associada ao número de erro.

Função	Descrição
SQLCODE	Retorna o valor numérico do código de erro (Você pode designá-lo a uma variável NUMBER .)
SQLERRM	Retorna dados de caracteres contendo a mensagem associada ao número do erro

Valores

SQLCODE: Exemplos

Valor SQLCODE	Descrição
0	Nenhuma exceção encontrada
1	Exceção definida pelo usuário
+100	Exceção NO_DATA_FOUND
<i>negative number</i>	Número de erro de outro servidor Oracle

Funções para Interceptar Exceções

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

21

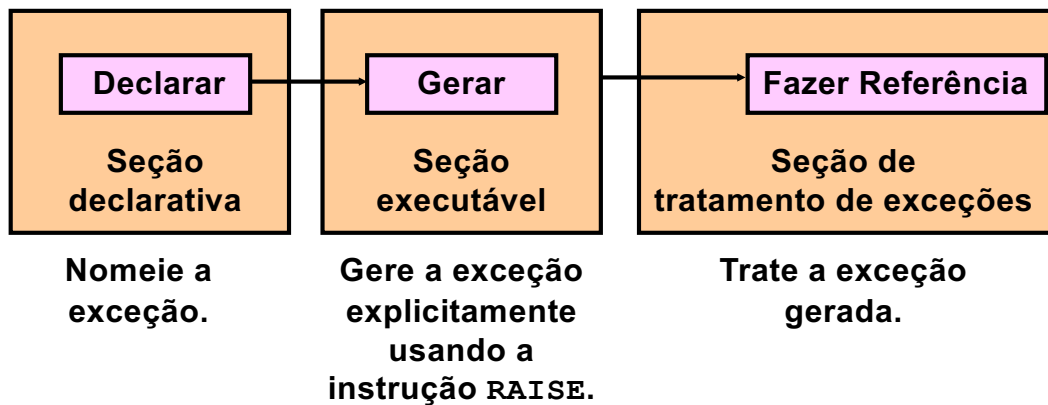
Funções para Interceptar Exceções (continuação)

Quando uma exceção for interceptada no handler de exceções WHEN OTHERS, você poderá usar um conjunto de funções genéricas para identificar esse erro. O exemplo do slide ilustra os valores de SQLCODE e SQLERRM sendo associados a variáveis e, em seguida, essas variáveis sendo usadas em uma instrução SQL.

Não é possível usar SQLCODE ou SQLERRM diretamente em uma instrução SQL. Em vez disso, você deverá designar seus valores a variáveis locais e, em seguida, usar as variáveis em instruções SQL, como é mostrado neste exemplo:

```
DECLARE
    err_num NUMBER;
    err_msg VARCHAR2(100);
BEGIN
    ...
EXCEPTION
    ...
    WHEN OTHERS THEN
        err_num := SQLCODE;
        err_msg := SUBSTR(SQLERRM, 1, 100);
        INSERT INTO errors VALUES (err_num, err_msg);
END;
```


Interceptando Exceções Definidas pelo Usuário



22

Interceptando Exceções Definidas pelo Usuário

O código PL/SQL permite que você defina suas próprias exceções de acordo com os requisitos da aplicação. Por exemplo, você pode solicitar que o usuário informe um número de departamento.

Defina uma exceção para tratar condições de erro nos dados de entrada. Verifique se o número de departamento existe. Se não existir, você deverá gerar uma exceção definida pelo usuário.

As exceções PL/SQL devem ser:

- Declaradas na seção declarativa de um bloco PL/SQL
- Geradas explicitamente com instruções RAISE
- Tratadas na seção EXCEPTION

Interceptando Exceções Definidas pelo Usuário

FIAP

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

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

- 1. Declaration of the exception `e_invalid_department`.
- 2. Execution of the `RAISE e_invalid_department;` statement within the `IF SQL%NOTFOUND THEN` block.
- 3. Execution of the exception handler `WHEN e_invalid_department THEN`.

Output window showing the results:

```
anonymous block completed
No such department id.
```

23

Interceptando Exceções Definidas pelo Usuário (continuação)

Para interceptar uma exceção definida pelo usuário, declare e gere essa exceção explicitamente.

1. Declare o nome da exceção definida pelo usuário dentro da seção declarativa.

Sintaxe:

```
exception EXCEPTION;
```

Na sintaxe, *exception* é o nome da exceção.

2. Use a instrução `RAISE` para gerar a exceção explicitamente na seção executável.

Sintaxe:

```
RAISE exception;
```

Na sintaxe, *exception* é a exceção declarada anteriormente.

3. Faça referência à exceção declarada na rotina correspondente de tratamento de exceções.

Exemplo

O bloco mostrado no slide atualiza o valor de `department_name` de um departamento. O usuário informa o número do departamento e o novo nome. Se o número do departamento fornecido não existir, nenhuma linha será atualizada na tabela `departments`. Uma exceção é gerada e uma mensagem é impressa para o usuário notificando-o de que um número de departamento inválido foi informado.

Observação: Use a instrução `RAISE` sozinha dentro de um handler de exceções para gerar

novamente a mesma exceção e propagá-la de volta para o ambiente de chamada.

Propagando Exceções em um Sub-bloco

Os sub-blocos podem tratar uma exceção ou transmiti-la para o bloco que os contém.

```
DECLARE
    . . .
    e_no_rows      exception;
    e_integrity    exception;
    PRAGMA EXCEPTION_INIT (e_integrity, -2292);
BEGIN
    FOR c record IN emp cursor LOOP
        BEGIN
            SELECT ...
            UPDATE ...
            IF SQL%NOTFOUND THEN
                RAISE e_no_rows;
            END IF;
        END;
    END LOOP;
EXCEPTION
    WHEN e_integrity THEN ...
    WHEN e_no_rows THEN ...
END;
/
```

24

Propagando Exceções em um Sub-bloco

Quando um sub-bloco trata uma exceção, ele é encerrado normalmente. O controle é retomado no bloco que o contém imediatamente após a instrução `END` do sub-bloco.

Entretanto, se um código PL/SQL gerar uma exceção e o bloco atual não tiver um handler para essa exceção, ela será propagada para os blocos que o contém até encontrar um handler. Se nenhum desses blocos tratar a exceção, será gerada uma exceção não tratada no ambiente host.

Quando a exceção se propagar para um bloco que contém outros, as ações executáveis restantes nesse bloco serão ignoradas.

Uma vantagem desse comportamento é que você pode incluir instruções que necessitam de um tratamento exclusivo de erros em blocos próprios, e deixar o tratamento geral de exceções para o bloco que os contém.

Observe no exemplo que as exceções (`no_rows` e `integrity`) são declaradas no bloco externo. No bloco interno, quando é gerada a exceção `no_rows`, o código PL/SQL procura fazer com que a exceção seja tratada no sub-bloco. Como a exceção não é resolvida no sub-bloco, a exceção é propagada para o bloco externo, onde o código PL/SQL encontra o handler.

Procedure RAISE_APPLICATION_ERROR FIAP

Sintaxe:

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

- Você pode usar esse procedure para gerar mensagens de erro definidas pelo usuário em subprogramas armazenados.
- É possível reportar erros à aplicação e evitar o retorno de exceções não resolvidas.

25

Procedure RAISE_APPLICATION_ERROR

Use o procedure RAISE_APPLICATION_ERROR para comunicar interativamente uma exceção predefinida, retornando uma mensagem e um código de erro não padrão. Com RAISE_APPLICATION_ERROR, é possível reportar erros à aplicação e evitar o retorno de exceções não tratadas.

Na sintaxe:

<i>error_number</i>	É um número especificado pelo usuário para a exceção entre – 20.000 e –20.999
<i>message</i>	É a mensagem especificada pelo usuário para a exceção; consiste em uma string de caracteres com até 2.048 bytes
TRUE FALSE	É um parâmetro booleano opcional (se TRUE, o erro será colocado na pilha de erros anteriores. Se o valor for FALSE, que é o default, o erro substituirá todos os erros anteriores.)

Procedure RAISE_APPLICATION_ERROR

- É usado em dois locais diferentes:
 - Seção executável
 - Seção de exceções
- Retorna condições de erro ao usuário, de modo consistente com outros erros do Oracle Server

Procedure RAISE_APPLICATION_ERROR (continuação)

É possível usar o procedure RAISE_APPLICATION_ERROR na seção executável, na seção de exceções ou nas duas seções de um programa PL/SQL. O erro retornado será consistente com o modo como o Oracle Server produz um erro predefinido, não predefinido ou definido pelo usuário. O número e a mensagem do erro são exibidos para o usuário.

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 de exceções:

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

27

Procedure RAISE_APPLICATION_ERROR (continuação)

O slide mostra que o procedure RAISE_APPLICATION_ERROR pode ser usado tanto na seção executável quanto na de exceções de um programa PL/SQL.

Veja um outro exemplo de uso do procedure RAISE_APPLICATION_ERROR:

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

Questionário

FIAP

É possível interceptar qualquer erro, incluindo um handler correspondente dentro da seção de tratamento de exceções do bloco PL/SQL.

- a. Verdadeiro
- b. Falso

28

Resposta: a

É possível interceptar qualquer erro, incluindo um handler correspondente dentro da seção de tratamento de exceções do bloco PL/SQL. Cada handler consiste em uma cláusula `WHEN`, que especifica um nome de exceção, seguida de uma sequência de instruções a serem executadas quando essa exceção for gerada. É possível incluir inúmeros handlers dentro de uma seção `EXCEPTION` para tratar exceções específicas. No entanto, não é possível ter vários handlers para uma única exceção.

Sumário

Nesta lição, você aprendeu a:

- Definir exceções PL/SQL
- Adicionar uma seção `EXCEPTION` ao bloco PL/SQL para lidar com exceções durante o runtime
- Tratar diferentes tipos de exceções:
 - Exceções predefinidas
 - Exceções não predefinidas
 - Exceções definidas pelo usuário
- Propagar exceções em blocos aninhados e chamar aplicações

Sumário

Nesta lição, você aprendeu a lidar com diferentes tipos de exceções. No código PL/SQL, uma condição de advertência ou erro durante o runtime é chamada de exceção. As exceções predefinidas são condições de erro que foram definidas pelo Oracle Server. As exceções não predefinidas podem ser quaisquer erros padrão do Oracle Server. As exceções definidas pelo usuário são exceções específicas da sua aplicação. A função `PRAGMA EXCEPTION_INIT` pode ser usada para associar um nome de exceção declarada a um erro do Oracle Server.

Você pode definir suas próprias exceções na parte declarativa de um bloco PL/SQL. Por exemplo, você pode definir a exceção `INSUFFICIENT_FUNDS` para identificar contas bancárias com saldo insuficiente.

Se ocorrer um erro, será gerada uma exceção. A execução normal é interrompida e o controle é transferido para a seção de tratamento de exceções do bloco PL/SQL. As exceções internas são geradas implicitamente (automaticamente) pelo sistema durante o runtime; contudo, as exceções definidas pelo usuário devem ser geradas explicitamente. Para tratar as exceções geradas, crie rotinas separadas chamadas handlers de exceções.

Exercício 8: Visão Geral

Este exercício aborda os seguintes tópicos:

- Criar e chamar exceções definidas pelo usuário
- Tratando exceções nomeadas do Oracle Server

Exercício 8: Visão Geral

Nestes exercícios, você criará handlers de exceção para uma exceção predefinida e uma exceção padrão do Oracle Server.