

Você está em

DevMedia

Artigo

Manipulação de erros em PL/pgSQL

Nesse artigo vamos tratar da manipulação de erros em um procedimento armazenado escrito na linguagem PL/pgSQL.



Anotar



Marcado como lido

Artigos



Banco de Dados



Manipulação de erros em PL/pgSQL

Manipulação de erros em PL/pgSQL

Olá pessoal,



Você está em

DevMedia

Por padrão, qualquer erro ocorrido em uma função PL/pgSQL aborta sua execução, e conseqüentemente a transação. Podemos capturar e tratar esses erros usando um bloco com uma cláusula de exceção. Sua sintaxe é uma extensão da sintaxe de um bloco comum.

Sintaxe:

```
[ <<label>> ]  
  [ DECLARE  
    declarações ]  
  BEGIN  
    afirmações  
  EXCEPTION  
    WHEN condição [ OR condição ... ] THEN  
      Faça algo  
    [ WHEN condição [ OR condição ... ] THEN  
      Faça algo  
    ... ]  
  END;
```

Uma condição de erro só é executada quando acontece o tal erro (óbvio), ou seja, se não ocorrer nenhum erro durante a execução do bloco normal, todos os comandos são executados normalmente até o END final. Mas se um erro acontece durante a execução, a execução normal do bloco onde houve o erro é abandonada e o controle passa para o bloco de EXCEPTION. Se o erro ocorrido está explícito no bloco então o código correspondente é executado, senão passa para o código de tratamento de erro genérico, caso exista.

Em <http://www.postgresql.org/docs/8.2/interactive/errcodes-appendix.html> é possível verificar a lista dos

Você está em

DevMedia

Quando acontece um erro do PostgreSQL a exception associada é sinalizada automaticamente. Erro gerado pelo PostgreSQL.

ou

Uma exception é explicitamente sinalizada através do comando RAISE dentro do bloco PL/pgSQL. Erro gerado intencionalmente pelo usuário.

Vejamos os exemplos abaixo.

Quando tentamos escrever um valor igual a um outro valor já existente em uma coluna de uma tabela e essa coluna é chave primária, o PostgreSQL gera um erro de chave duplicada. Esse é um erro interno gerado automaticamente pelo PostgreSQL.

Agora imagine que eu tenho uma função que recebe um valor inteiro como parâmetro e que esse parâmetro não pode receber um valor superior a 1000. Nesse caso, eu poderia gerar um erro através do comando RAISE, todas as vezes que alguém chamasse essa função passando um valor superior a 1000.

Você está em

DevMedia

Em um banco de dados que contém uma tabela usuario com cod (integer) e nome (varchar).

Create function teste_except() returns integer

As \$\$

Declare

x integer;

y integer;

Begin

INSERT INTO usuario values (1, 'Hesley Py');

BEGIN

UPDATE usuario SET nome = 'Hesley' WHERE cod = 1;

x := 1;

y := x / 0;

EXCEPTION WHEN division_by_zero THEN

Você está em

DevMedia

```
return x;
```

```
End;
```

```
$$ Language 'plpgsql';
```

Após a execução do código acima digite:

```
Select teste_except();
```

Neste exemplo estamos causando, propositadamente, um erro de divisão por 0. Ao final da execução você perceberá que o insert será executado corretamente, mas o update não. Isso acontece pq a transação só é interrompida (ROLLBACK) dentro do bloco onde aconteceu o erro. No exemplo mostrado o erro acontece somente no bloco (Begin ... End) onde ocorre o update.

A instrução RAISE é utilizada para gerar mensagens informativas e causar erros.

Sintaxe:

Você está em

DevMedia

O nível EXCEPTION (como visto acima) causa um erro e interrompe a transação corrente; os outros níveis apenas geram mensagens com diferentes níveis de prioridade.

Dentro da cadeia de caracteres de formatação, o caractere % é substituído pelo valor, na forma de cadeia de caracteres, da variável.

Exemplo:

```
RAISE NOTICE "O valor da variável é %", variavel;
```

No exemplo, o símbolo % será substituído pelo valor presente em variavel.

Outro exemplo, agora interrompendo a transação com a mensagem de erro fornecida:

```
RAISE EXCEPTION "ID inexistente --> %", id_usuario;
```

Você está em

DevMedia

Create function exclui_usuario(id integer) returns void

As \$\$

Begin

 If id != 1 then

 DELETE FROM usuario WHERE cod = id;

 Else

 RAISE EXCEPTION 'O usuário código % não pode ser removido', id;

 End if;

End;

\$\$ Language 'plpgsql';

Ao chamar essa função a partir de uma outra, a função chamadora deve possuir uma cláusula exception para tratar esse possível erro.

Espero que seja útil.

Você está em

DevMedia

Anotar



Marcado como lido



Por Hesley

Em 2007

Falar com professor - Tire a sua dúvida.



Poste aqui a sua dúvida, nessa seção só você e o consultor podem ver os seus comentários.

[Enviar dúvida](#)[Planos de estudo](#)[Fale conosco](#)[Plano para Instituição de ensino](#)[Assinatura para empresas](#)[Assine agora](#)

Você está em

DevMedia



2

