



Sistemas de Informação

Prof. Me. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

Criação de Funções

Objetivos

- Descrever a utilização das funções
- Criar funções armazenadas
- Invocar uma função
- Remover uma função
- Diferenciar entre um procedimento e uma função

Objetivo da lição

Criar e invocar funções.

Visão geral de Funções Armazenadas

Uma função:

- É um bloco PL/SQL nomeado que deve retornar um valor
- É armazenado no banco de dados como um objeto de esquema para ser executado várias vezes
- É chamado como parte de uma expressão ou é utilizado para fornecer um valor de parâmetro

Visão geral de funções armazenadas

Uma função é um bloco PL/SQL nomeado que pode aceitar parâmetros, ser invocado e retorna um valor. Em geral, você utiliza uma função para calcular um valor. Funções e procedimentos são estruturados. Uma função deve retornar um valor para o ambiente de chamada, enquanto um procedimento retorna zero ou mais valores ao seu ambiente de chamada. Como um procedimento, uma função possui um cabeçalho, uma seção declarativa, uma seção executável e uma seção opcional de tratamento de exceções. Uma função deve ter uma cláusula RETURN no cabeçalho e pelo menos uma declaração RETURN na seção executável.

As funções podem ser armazenadas no banco de dados como objetos de esquema. Uma função que é armazenada no banco de dados é referida como uma função armazenada. As funções também podem ser criadas em aplicativos do lado do cliente.

As funções promovem a reutilização e facilitam a manutenção. Quando válidos, esses objetos podem ser usados em qualquer aplicativo. Se os requisitos de negócio mudarem, apenas uma função precisa ser atualizada.

Uma função também pode ser chamada como parte de uma expressão SQL ou como parte de uma expressão PL/SQL. No contexto de uma expressão SQL, uma função deve obedecer a regras específicas para controlar os efeitos colaterais. Em uma

expressão PL/SQL, o identificador de função atua como uma variável cujo valor depende dos parâmetros passados para ele.

Sintaxe para Criar Função

- O bloco PL/SQL deve ter obrigatoriamente pelo menos uma declaração RETURN.

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter1 [mode1] datatype1, ...)]
RETURN datatype IS|AS
[local_variable_declarations; ...]
BEGIN
  -- actions;
  RETURN expression;
END [function_name];
```

→ Bloco PL/SQL

Sintaxe para criar funções

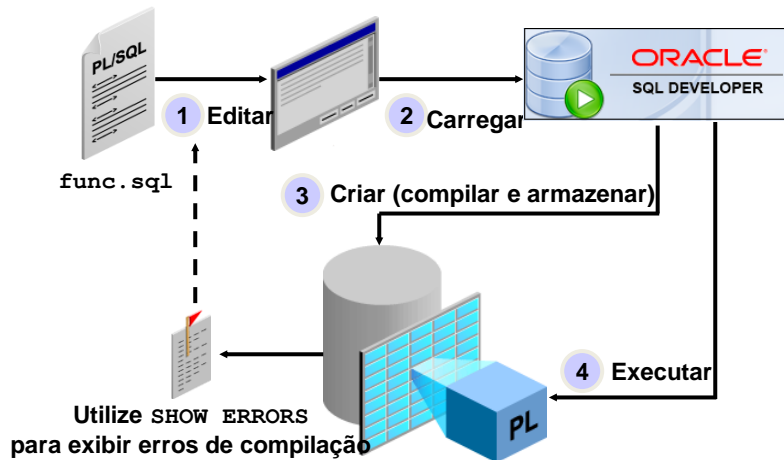
Uma função é um bloco PL/SQL que retorna um valor. Uma declaração RETURN deve ser fornecida para retornar um valor com um tipo de dados consistente com a declaração da função.

Você cria novas funções com a instrução CREATE FUNCTION, que pode declarar uma lista de parâmetros, deve retornar um valor e deve definir as ações a serem executadas pelo bloco PL/SQL padrão.

Você deve considerar os seguintes pontos sobre a instrução CREATE FUNCTION:

- A opção REPLACE indica que, se a função existir, ela será descartada e substituída pela nova versão .
- O tipo de dados RETURN não deve incluir uma especificação de tamanho.
- O bloco PL/SQL começa com BEGIN após a declaração de quaisquer variáveis locais e termina com um END, seguido opcionalmente pelo nome_da_função.
- Deve haver pelo menos uma declaração de expressão RETURN.
- Você não pode referenciar host variables ou vincular variáveis no bloco PL/SQL de uma função armazenada.

Desenvolvendo Funções Armazenadas



Como desenvolver funções armazenadas

O diagrama ilustra os passos básicos no desenvolvimento de uma função. Para desenvolver uma função armazenada, execute as seguintes etapas:

1. Crie um arquivo usando seu editor de texto ou código favorito para editar a sintaxe da função e salve o código em um arquivo normalmente com uma extensão `.sql`.
2. Carregue o código.
3. Execute a instrução `CREATE FUNCTION` para compilar e armazenar a função no banco de dados.
4. Depois da compilação bem-sucedida, invoque a função a partir de um ambiente ou aplicativo PL/SQL.

Retornando um Valor

- Adicione uma cláusula `RETURN` com o tipo de dados no cabeçalho da função.
- Inclua uma declaração `RETURN` na seção executável.

Várias instruções de RETORNO são permitidas em uma função (geralmente dentro de uma instrução `IF`). Apenas uma declaração `RETURN` é executada, pois após o retorno do valor, o processamento do bloco é encerrado.

Use os comandos `SHOWS ERRORS` ou `SHOW ERRORS FUNCTION function_name` para ver erros de compilação.

Função: Exemplo

- Crie a função:

```
CREATE OR REPLACE FUNCTION get_sal
(id employees.employee_id%TYPE) RETURN NUMBER IS
sal employees.salary%TYPE := 0;
BEGIN
SELECT salary
INTO sal
FROM employees
WHERE employee_id = id;
RETURN sal;
END get_sal;
/
```

- Invoque a função como uma expressão ou como um valor de parâmetro:

```
EXECUTE dbms_output.put_line(get_sal(100))
```

Função armazenada: Exemplo

A função `get_sal` é criada com um único parâmetro de entrada e retorna o salário como um número. Execute o comando como mostrado ou guarde-o em um arquivo de script e execute o script para criar a função `get_sal`.

A função `get_sal` segue uma prática de programação comum de usar uma única instrução `RETURN` que retorna um valor atribuído a uma variável local. Se a sua função tiver uma seção de exceção, ela também pode conter uma declaração `RETURN`.

Invoque uma função como parte de uma expressão PL / SQL, porque a função retornará um valor para o ambiente de chamada. A segunda caixa de código usa o `EXECUTE` para chamar o procedimento `DBMS_OUTPUT.PUT_LINE` cujo argumento é o valor de retorno da função `get_sal`. Neste caso, `get_sal` é invocado primeiro para calcular o salário do empregado com ID 100. O valor salarial retornado é fornecido como o valor do parâmetro `DBMS_OUTPUT.PUT_LINE`, que exibe o resultado (se você executou `SET SERVEROUT`).

Nota: Uma função deve sempre retornar um valor. O exemplo não retorna um valor se uma linha não for encontrada para um determinado ID. Idealmente, crie um manipulador de exceção para retornar um valor também.

Formas de execução de uma função

- Invocar como parte de uma expressão PL/SQL

- Utilizando uma variável host para obter o resultado

```
VARIABLE salary NUMBER  
EXECUTE :salary := get_sal(100)
```

- Utilizando uma variável local para obter o resultado

```
DECLARE sal employees.salary%type;  
BEGIN  
    sal := get_sal(100); ...  
END;
```

- Utilize em uma instrução SQL (sujeito a restrições)

```
EXECUTE dbms_output.put_line(get_sal(100))
```

```
SELECT job_id, get_sal(employee_id) FROM employees;
```

Função em Expressões SQL: Exemplo

```
CREATE OR REPLACE FUNCTION tax(value IN NUMBER)
  RETURN NUMBER IS
BEGIN
  RETURN (value * 0.08);
END tax;
/
SELECT employee_id, last_name, salary, tax(salary)
FROM   employees
WHERE  department_id = 100;
```

Function created.

| EMPLOYEE_ID | LAST_NAME | SALARY | TAX(SALARY) |
|-------------|-----------|--------|-------------|
| 108 | Greenberg | 12000 | 960 |
| 109 | Faviet | 9000 | 720 |
| 110 | Chen | 8200 | 656 |
| 111 | Sciarra | 7700 | 616 |
| 112 | Urman | 7800 | 624 |
| 113 | Popp | 6900 | 552 |

6 rows selected.

Função em Expressões SQL: Exemplo

O exemplo no slide mostra como criar uma função para calcular o imposto de renda. A função aceita um parâmetro NUMBER e retorna o imposto de renda calculado com base em uma simples taxa fixa de imposto de 8%.

A função de imposto é invocada como uma expressão na cláusula SELECT juntamente com o ID do empregado, o sobrenome e o salário dos funcionários de um departamento com ID 100. O resultado de retorno da função de imposto é exibido com o resultado normal a partir da consulta.

Locais para chamar funções definidas pelo usuário

As funções definidas pelo usuário funcionam como funções de uma única linha e podem ser utilizadas em:

- Uma lista da instrução SELECT
- Expressões condicionais das cláusulas WHERE e HAVING
- Na cláusula VALUES da declaração INSERT
- Na cláusula SET da instrução UPDATE

Locais para chamar funções definidas pelo usuário

Uma função definida pelo usuário PL/SQL pode ser chamada a partir de qualquer expressão SQL em que uma função de linha única embutida pode ser chamada.

Exemplo:

```
SELECT employee_id, tax(salary)
FROM employees
WHERE tax(salary) > (SELECT MAX(tax(salary))
                     FROM employees
                     WHERE department_id = 30)
ORDER BY tax(salary) DESC;
```

| EMPLOYEE_ID | TAX(SALARY) |
|-------------|-------------|
| 100 | 1920 |
| 101 | 1360 |
| 102 | 1360 |
| 145 | 1120 |
| 146 | 1080 |
| ... | ... |

10 rows selected.

Removendo uma Função

- Removendo uma função armazenada:

- Você pode remover uma função utilizando a seguinte sintaxe:

```
DROP FUNCTION function_name
```

- Exemplo:

```
DROP FUNCTION get_sal;
```

- Todos os privilégios concedidos em uma função são revogados quando a função é removida.
- A sintaxe CREATE OR REPLACE é equivalente a apagar uma função e a recriá-la. Os privilégios concedidos na função permanecem os mesmos quando esta sintaxe é usada.

Removendo Funções

Para remover uma função execute o comando DROP FUNCTION SQL.

CREATE OR REPLACE Versus DROP e CREATE

A cláusula REPLACE na sintaxe CREATE OR REPLACE é equivalente a apagar uma função e recriá-la. Quando você usa a sintaxe CREATE OR REPLACE, os privilégios concedidos neste objeto a outros usuários permanecem os mesmos. Quando você DROP uma função e, em seguida, recriá-la, todos os privilégios concedidos nesta função são revogados automaticamente.

Procedimentos X Funções

| Procedimentos | Funções |
|---|--|
| Execute como uma instrução PL/SQL | Invocado como parte de uma expressão |
| Não contém cláusula RETURN no cabeçalho | Deve conter uma cláusula RETURN no cabeçalho |
| Pode retornar valores (se houver) nos parâmetros de saída | Deve retornar um único valor |
| Pode conter uma instrução RETURN sem um valor | Deve conter pelo menos uma declaração RETURN |

Diferenças entre procedimentos e funções

Você cria um procedimento para armazenar uma série de ações para execução posterior. Um procedimento pode conter zero ou mais parâmetros que podem ser transferidos para e do ambiente de chamada, mas um procedimento não precisa retornar um valor. Um procedimento pode chamar uma função para auxiliar suas ações.

Nota: Um procedimento contendo um único parâmetro OUT seria melhor reescrito como uma função retornando o valor.

Você cria uma função quando deseja calcular um valor que deve ser retornado para o ambiente de chamada. Uma função pode conter zero ou mais parâmetros que são transferidos do ambiente de chamada. As funções normalmente retornam apenas um valor único e o valor é retornado através de uma declaração RETURN. As funções usadas nas instruções SQL não devem usar os parâmetros do modo OUT ou IN OUT. Embora uma função que use parâmetros de saída possa ser usada em um procedimento ou bloco PL / SQL, ele não pode ser usado em instruções SQL.

Resumo

Nesta lição, você aprendeu a:

- Escrever uma função PL/SQL utilizando a instrução `CREATE FUNCTION`
- Invocar uma função como parte de uma expressão PL/SQL
- Utilizar funções em instruções SQL
- Remover uma função do banco de dados utilizando a instrução SQL `DROP FUNCTION`

OBRIGADO



profalexandre.barcelos@fiap.com.br



<https://www.linkedin.com/in/alexandrebarcelos>

FIAP

Copyright © 2023 | Professor Me. Alexandre Barcelos
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente
proibido sem consentimento formal, por escrito, do professor/autor.