FIAP

Stored Procedures e Functions

DATABASE APPLICATION DEVELOPMENT

Leonardo Uchida
leonardo@fiap.com.br

Procedures

FIAP

É um subprograma/objeto que possui um nome único e consiste em uma série de comandos PL/SQL. Tem por objetivo manipular dados.

- ✓ Blocos anônimos nomeados
- ✓ Podem ser armazenados no banco de dados
- ✓ São pré-compilados no banco de dados
- ✓ Reutilizáveis em todas as linguagens que interagem com BD
  - ✓ Podem receber valores como parâmetro
  - ✓ Podem retornar valores
- ✓ Não podem ser chamadas diretamente através do comando SELECT.
- ✓ Melhor performance (Shared pool)

Procedures

Sintaxe:

```
CREATE OR REPLACE PROCEDURE <nome_ate_30_caracteres>
        (parametros IN/OUT <datatype>
        ..)
[ IS | AS ]
        <declaração de variáveis, cursores, constantes>
BEGIN
        <execução>
EXCEPTION
        <tratamento de exceção>
END;
```

Procedures

FIAP

Exemplo:

```
create or replace procedure hello_world
as
      v_mensagem varchar2(100);
begin
   v_mensagem:='Hello World via procedure.';
   dbms_output.put_line (v_mensagem);
end;
```

Execução da procedure:

```
set serveroutput on;

exec hello_world;
Hello World via procedure.

PL/SQL procedure successfully completed.
```

```
begin
    hello_world;
end;
/
Hello World via procedure.

PL/SQL procedure successfully completed.
```

Procedures

FIAP

Parâmetros:

IN => Somente entrada. O valor do parâmetro não pode ser sobrescrito durante a execução. (DEFAULT)

- CREATE [OR REPLACE] PROCEDURE proc1 (param_name1 IN datatype, param_name12 IN datatype ...)

OUT => Somente saída. O valor do parâmetro não pode ser atribuído na chamada do procedimento.

- CREATE [OR REPLACE] PROCEDURE proc2 (param_name OUT datatype)

INOUT => Entrada e Saída. A variável de parâmetro pode ser populada na chamada e sobrescrita na execução.

- CREATE [OR REPLACE] PROCEDURE proc3 (param_name IN OUT datatype)

Procedures

FIAP

Exemplo com parâmetros:

```
create or replace procedure hello_world (p_msg in varchar2, p_qtd_chars out number)
as
begin
   p_qtd_chars := nvl(length(p_msg),0);
   dbms_output.put_line (p_msg);
end;
```

Execução da procedure:

```
set serveroutput on;

declare
   mensagem varchar2(1000);
   qtd      number;
begin
  mensagem := 'Aula de PL/SQL';
  hello_world(mensagem, qtd);
  dbms_output.put_line('A mensagem "' || mensagem || '" tem ' || qtd || ' caracteres.');
end;
/
A mensagem "Aula de PL/SQL" tem 14 caracteres.
PL/SQL procedure successfully completed.
```

Functions

FIAP

É um subprograma/objeto que possui um nome único e consiste em uma série de comandos PL/SQL. Tem por objetivo computar/calcular dados e sempre deve retornar um valor.

- ✓ Blocos anônimos nomeados
- ✓ Podem ser armazenados no banco de dados
- ✓ São pré-compilados no banco de dados
- ✓ Reutilizáveis em todas as linguagens que interagem com BD
  - ✓ Podem receber valores como parâmetro
  - ✓ Devem retornar valores (Mandatório)
- ✓ Podem ser chamadas diretamente através do comando SELECT, mas não podem implementar DML.
- ✓ Melhor performance (Shared pool)

Functions

FIAP

Sintaxe:

```
CREATE OR REPLACE FUNCTION <nome_ate_30_caracteres>
        (parametros IN/OUT <datatype>..) RETURN datatype
[ IS | AS ]
        <declaração de variáveis, cursores, constantes>
BEGIN
        <execução>
        return Valor;
EXCEPTION
        <tratamento de exceção>
END;
```

Functions

FIAP

Exemplo:

```
create or replace function soma_dois_numeros
(p_n1 in number, p_n2 in number)
return number
is
    v_soma    number;
begin
    v_soma := p_n1 + p_n2;
    return v_soma;
end;
```

Execução da function:

```
select soma_dois_numeros(1,2) from dual;

SOMA_DOIS_NUMEROS(1,2)
----------------------
                     3
```

```
declare
  resultado number;
begin
  resultado := soma_dois_numeros(1,2);
  dbms_output.put_line(resultado);
end;
3
PL/SQL procedure successfully completed.
```

Built-in Functions

FIAP

São funções nativas/core do Oracle database.

Exemplos:

- **TO_CHAR**
- **TO_NUMBER**
- **TO_DATE**
- **INSTR**
- **SUBSTR**
- **UPPER**
- **LOWER**
- **INITCAP**
- **LENGTH**
- **SYSDATE**
- **TRUNC**

Procedures x Functions

FIAP

Procedures	Functions
Utilizado para processar DML e manipular dados	Utilizado para executar pequenos processamentos / calculos
Não pode ser chamado através do comando SELECT	Uma função que não implementa comandos de DML podem ser chamados através do comando SELECT
Retorno de valor somente quando utilizado um parämetro OUT	Comando RETURN do bloco de execução retorna um valor
Não mandatório ter parametro de retorno	É mandatório retornar um valor
RETURN simplesmente encerra a execução e devolve o controle para o executor	RETURN encerra a execução através do retorno de um valor
Tipo de dado de retorno não é especificado no cabeçalho do objeto	Tipo de dado de retorno obrigatório no cabeçalho do objeto

Exercícios

FIAP

Functions:

1) Crie uma função chamada fun_qtd_func_depto que retorne a quantidade total de funcionário de um determinado departamento. Caso não exista funcionário cadastrado para o departamento informado, ou ocorrer qualquer erro imprevisto, retornar o valor 0 (zero).

2) Desenvolva uma função denominada fun_nome_veículo que vai ter um retorno varchar2. Essa função vai receber como parâmetro a placa do veículo e selecionar as seguintes colunas: Modelo + Cor + e tipo do automóvel. Concatene essas colunas na variável de retorno da função. Caso a placa do veículo não esteja cadastrada, parar o processamento emitindo a seguinte mensagem de erro “Placa <nrplaca> do veículo não está cadastrada. Favor informar um código válido !”.

3) Desenvolva uma função denominada fun_nome_grupo que vai ter um retorno varchar2. Essa função vai receber como parâmetro o código do grupo de veículo e selecionar o nome do grupo na tabela LOC_GRUPO. Caso o código do grupo não esteja cadastrado, parar o processamento emitindo a seguinte mensagem de erro “Código do grupo não está cadastrado. Favor informar um código válido !”.

4) Crie uma função chamada fun_val_med_loc_cliente que retorne o valor total médio gasto por determinado cliente. Caso não exista o cliente cadastrado, ou ocorrer qualquer erro imprevisto, retornar o valor 0 (zero).

Exercícios

FIAP

Procedures:

1) Desenvolva um procedimento chamado prc_gera_estrelas_cli que tem como objetivo atualizar o número de estrelas de cada cliente(coluna nr_estrelas: tabela LOC_CLIENTE). O número de estrelas que um cliente pode possuir está entre 1 e 5 estrelas. Para ser possível identificar qual estrela o cliente possui, utilize a função FUN_VAL_MED_LOC_CLI para obter o valor médio total gasto pelo cliente e utilize a tabela abaixo, que determina em qual faixa o valor médio de locação o cliente se encaixa. No final do processamento, confirme as transações realizadas e cria uma instrução SQL à parte que verifica se a estrela gerada para cada cliente está de acordo com as locações hoje cadastradas.

De 0,00 A 200,00	1
De 201,00 A 300,00	2
De 301,00 A 450,00	3
De 451,00 A 600,00	4
Maior que 600,00	5

2) Criar um procedimento de nome prc_insere_proprietario para inserir dados na tabela loc_proprietario , este procedimento recebera como parâmetro todas as colunas da tabela loc_proprietario com exceção da chave primária que deve ser obtida automaticamente selecionando o ultimo valor da chave incrementando + 1. Utilize todos os exceptions necessários para garantir que o tratamento de erro(mensagens) seja gerenciado pela aplicação.

3) Criar um procedimento de nome prc_insere_grupo para inserir dados na tabela loc_grupo, este procedimento recebera como parâmetro todas as colunas da tabela loc_grupo, com exceção da chave primária que deve ser obtida automaticamente selecionando o ultimo valor da chave incrementando + 1. Utilize todos os exceptions necessários para garantir que o tratamento de erro(mensagens) seja gerenciado pela aplicação. Caso ocorra alguma anormalidade, exiba uma mensagem de erro significativa.