PL-pgSQL - Módulo I

- **Linguagem Procedural para Banco de Dados:**

- Caracteristicas

- Proporcionar a possibilidade da criação de funções ou procedimentos armazenados (functions/stored procedures) e gatilhos (triggers);

- Adicionar estruturas de controle de fluxo à linguagem SQL;

- Herdar e manipular todos os tipos de dados, funções e operadores existentes no banco de dados;

- Considerada confiável por parte do servidor;

- São de fácil utilização.

PL-pgSQL - Vantagens

- Redução do número de "round trips" desnecessários entre clientes e servidores;
- Possibilidade de agrupar vários comandos SQL em um único procedimento;
- Redução do número de "query parsing";
- Aumento da segurança do código;
- Possível redução do consumo de rede.

PL-pgSQL – Tipos de Funções

- Query Language Functions (Funções escritas em SQL);
- Procedural Language Functions (funções escritas em, por exemplo, PL/pgSQL);
- Funções internas;
- Funções em linguagem C.

PL-pgSQL – Criando Função

```
CREATE [or REPLACE] FUNCTION
   nome_funcao(par_1 tipo, par_2 tipo ... par_n)
RETURNS [SETOF] [VOID || DATATYPE || ESCALAR]
   AS
$$
DECLARE
BEGIN
RETURN;
END;
$$ LANGUAGE 'PLpgSQL';
```

PL-pgSQL – Funções escritas em SQL

- **Executam uma lista de comandos SQL;**
- **Retornam o resultado da última consulta da lista;**
- **Se a última consulta não retornar valores um null será retornado;**
- **Alternativamente uma função SQL poderá retornar um conjunto de valores (SETOF tabela/tipo);**
- **Todos os comandos devem ser separados por “;”**
- **A menos que a função seja declarada para retornar VOID, o último comando deverá ser um SELECT;**

PL-pgSQL – Funções escritas em SQL

- **Se o retorno for VOID o último comando não poderá ser um SELECT;**
- **Qualquer comando poderá ser utilizado (INSERT, SELECT, UPDATE e DELETE).**

PL-pgSQL – Funções escritas em SQL

- **Exemplos:**

```
CREATE FUNCTION limparMovimento() RETURNS
   void AS '
DELETE FROM movimento;
' LANGUAGE SQL;
```

- **Para executar a função**

```
select * from movimento;
SELECT limparMovimento();
select * from movimento;
```

PL-pgSQL – Funções escritas em SQL

- **Exemplos:**

```
CREATE FUNCTION inserirMovimento() RETURNS
   void AS $$
insert into movimento values('ped1',1,20,53.00);
insert into movimento values('ped1',3,15,29.70);
insert into movimento values('ped1',4,10,15.40);
insert into movimento values('ped2',4,12,18.48);
insert into movimento values('ped2',3,10,19.80);
insert into movimento values('ped3',1,15,39.75);
$$ LANGUAGE SQL;
```

PL-pgSQL – Funções escritas em SQL

- **Para executar a função:**

```
select * from movimento;
select inserirMovimento();
select * from movimento;
```

PL-pgSQL – Funções escritas em SQL

- **Exemplos:**

```
CREATE FUNCTION atualizarMovimento(char(10),
   integer, integer) returns
integer AS
$$
UPDATE movimento set  qtde = $3 where
nro_ped = $1 and cod_prod = $2;
select 1;
$$
LANGUAGE SQL;
```

PL-pgSQL – Funções escritas em SQL

- **Para executar a função:**

```
select * from movimento
select atualizarMovimento('ped1', 1,21);
select * from movimento
select atualizarMovimento('ped1', 1,20);
select * from movimento
```

PL-pgSQL – Funções escritas em SQL

- **Exemplo:**

```
CREATE OR REPLACE FUNCTION
atualizarPrecoProduto(integer, decimal(5,2)) returns
    setof produto AS $$
UPDATE produto set preco = preco * (1.00 + $2/100.00)
where cod_prod = $1;
select * from produto ;
$$
LANGUAGE SQL;
```

PL-pgSQL – Funções escritas em SQL

- **Para executar a função:**

```
select * from produto;
select * from atualizarPrecoProduto(1, 10.00);
select * from produto;
```