

PROGRAMAÇÃO EM BANCO DE DADOS

Prof. Roberson Alves

PROGRAMAÇÃO EM BANCO: OBJETOS BÁSICOS

- **CONSISTE EM ESCREVER ALGORITMOS DENTRO DO BANCO DE DADOS COM LINGUAGEM PROCEDURAL;**
- **UMA ESPÉCIE DE SQL ESTENDIDO OU AVANÇADO;**
- **PROCEDIMENTOS ARMAZENADOS(STORED PROCEDURES);**
- **GATILHOS(TRIGGERS).**

PROGRAMAÇÃO EM BANCO: VANTAGENS

- **Os usuários não precisam saber detalhes da implementação, ou das tabelas que serão usadas;**
- **Se um conjunto de procedimentos armazenados suporta todas as funcionalidades que os usuários precisam, esses usuários nunca precisarão acessar as tabelas diretamente;**
- **Usuários podem ter acesso aos procedimentos armazenados, mesmo que não tenham acesso direto as tabelas ou visões destas;**
- **Procedimentos armazenados podem encapsular as regras do negócio;**
- **Todos os usuários podem usar o mesmo procedimento para assegurar consistência ao acesso a dados e modificação; e**
- **Reduz o tráfego na rede.**

PROGRAMAÇÃO EM BANCO: DESVANTAGENS

- **Aumenta a carga no servidor de banco de dados;**
- **Linguagem não é padronizada entre os diferentes bancos de dados.**

PROGRAMAÇÃO EM BANCO: ORACLE

```
1 CREATE OR REPLACE PROCEDURE aumenta_sal (p_empno IN emp.empno%TYPE) IS
2 BEGIN
3     UPDATE
4         scott.emp
5     SET
6         sal = sal * 1.10
7     WHERE
8         empno = p_empno;
9 END aumenta_sal;
10 /
```

PROGRAMAÇÃO EM BANCO: FIREBIRD

```
CREATE PROCEDURE SP SELECIONA CLIENTES (  
    pTIPO VARCHAR(1)  
)  
RETURNS (  
    rCOD_CLIENTE INTEGER,  
    rNOME VARCHAR(100),  
    rTIPO VARCHAR(15)  
)  
as  
begin  
    for  
        SELECT cod_cliente, NOME, TIPO from cliente  
        where tipo = :ptipo  
        INTO :rcod_cliente, :rNOME, :rTIPO  
    do  
        begin  
            if (:rtipo = 'F') then  
                rtipo = 'Pessoa Física';  
            else  
                if (:rtipo = 'J') then  
                    rtipo = 'Pessoa Jurídica';  
                suspend;  
            end  
        end  
    end
```

PROGRAMAÇÃO EM BANCO: SQL SERVER

```
CREATE PROCEDURE uspFindProductByModel (  
    @model_year SMALLINT,  
    @product_count INT OUTPUT  
) AS  
BEGIN  
    SELECT  
        product_name,  
        list_price  
    FROM  
        production.products  
    WHERE  
        model_year = @model_year;  
  
    SELECT @product_count = @@ROWCOUNT;  
END;
```


PROGRAMAÇÃO EM BANCO DE DADOS

- Todos os bancos de dados comerciais possuem uma **linguagem procedural** auxiliar para a definição de **procedimentos armazenados**
 - Definição de regras de negócio
 - Especificação de restrições de integridade não possíveis no modelo relacional
 - Cálculo de atributos derivados
 - Auditoria
 - Adição de funcionalidades ao banco

PROCEDIMENTOS ARMAZENADOS

- Também conhecidos como Stored Procedures. São um poderoso recurso de banco de dados. Nada mais que programas desenvolvidos em determinada linguagem de script e armazenados no servidor, local onde serão processados;
- Eles também são conhecidos como funções, este é o motivo pelo qual quando nos referenciamos a uma stored procedure no PostgreSQL devemos utilizar o nome de **Function**;
- A partir do PG11 é possível utilizar construções **CREATE PROCEDURE**.

PROGRAMAÇÃO NO POSTGRESQL

- PostgreSQL não possui uma única linguagem procedural, este SGBD aceita várias linguagens e pode ser estendido para outras
 - PL/pgSQL
 - PL/Tcl
 - PL/Perl
 - PL/Python
 - Entre outras não distribuídas com o SGBD: PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme e PL/sh

PROGRAMAÇÃO NO POSTGRESQL

- **PL/pgSQL** (*Procedural Language for the PostgreSQL*) é uma linguagem procedural carregável desenvolvida para o SGBD PostgreSQL, que possui as seguintes características:
 - é utilizada para criar funções e gatilhos;
 - possibilita a execução de processamentos complexos;
 - permite adicionar estruturas de controle à linguagem SQL;
 - é fácil de ser utilizada;
 - é compatível com a PL/SQL padrão.

TRIGGERS EM POSTGRESQL

- **Um dos principais usos de linguagens procedurais em bancos de dados é a definição de gatilhos**
- **Gatilhos são execuções disparadas pelo banco em função de EVENTOS que ocorrem**
- **Um evento ocorre**
 - **Em uma tabela**
 - **De acordo com uma operação, por DML (INSERT, UPDATE ou DELETE)**
 - **Antes ou depois (AFTER ou BEFORE)**

TRIGGERS

- Identificadores de correlação – variáveis de vínculo PL/PGSQL
 - sempre vinculados à tabela desencadeadora do gatilho(trigger)

<div>instrução</div> <div>identificador</div>	old	new
INSERT	NULL	valores que serão inseridos
UPDATE	valores antes da atualização	novos valores para a atualização
DELETE	valores antes da remoção	NULL