Centro Paula Souza

Fatec Jacareí
Prof. Francisco de Moura

BANCO DE DADOS RELACIONAL

Stored Procedures (Procedimentos Armazenados)

Semestre: 1º/2025

Professora: Lucineide Pimenta

Objetivos da aula

Fatec
Jacareí
Prof. Francisco de Moura

- **Objetivos Gerais:**
  - Conhecer como criar e usar **Stored Procedures** no PostgreSQL.
  - Conhecer como automatizar processos dentro do banco de dados.
  - Conhecer como alterar e excluir Procedures.
  - Demonstrar exemplos práticos nos **bancos de dados Escola e Sistema Bancário**.
- **Objetivos Específicos:**
  - Criar **Stored Procedures** para simplificar operações.
  - Entender como **parâmetros e controle de fluxo** funcionam dentro de uma Procedure.
  - Aplicar Procedures para **automatizar processos** no banco.
  - Modificar Procedures sem precisar deletá-las e recriá-las.
  - Excluir Procedures que não são mais necessárias.
  - Implementar essas técnicas no **projeto ABP**.

Crie a estrutura do banco de dados

Fatec
Jacareí
Prof. Francisco de Moura

create database bd_escola;

CREATE TABLE cursos (id_curso SERIAL PRIMARY KEY, nome VARCHAR(100) NOT NULL);

CREATE TABLE alunos (id_aluno SERIAL PRIMARY KEY, nome VARCHAR(100) NOT NULL, idade INT, id_curso INT REFERENCES cursos(id_curso));

CREATE TABLE notas (id_nota PRIMARY KEY, disciplina VARCHAR(100) NOT NULL, nota float, id_aluno INT REFERENCES alunos(id_aluno));

INSERT INTO cursos (nome) VALUES ('Engenharia');

INSERT INTO cursos (nome) VALUES ('Análise de Sistemas'), ('Computação'), ('Matemática');

INSERT INTO alunos (nome, idade, id_curso) VALUES ('João Silva', 22, 1);

INSERT INTO alunos (nome, idade, id_curso) VALUES ('Marina Lima', 16, 3), ('Maria Souza', 20, 3), ('Carlos Lima', 25, 4), ('Lucas Pereira', 18, 3);

INSERT INTO notas (id_nota, id_aluno, disciplina, nota) VALUES (101, 1, 'Matemática', 8.5),(102, 2, 'História', 9.0);

UPDATE alunos SET idade = 16 WHERE nome = 'João Silva';

UPDATE alunos SET idade = 17, id_curso = 1 WHERE nome = 'Marina Lima';

*Select * from cursos;*

*Select * from alunos;*

Crie a estrutura do banco de dados

Fatec Jacareí
Prof. Francisco de Moura

```
create database bd_bancofinanceiro;

-- TABELA: Cliente
CREATE TABLE cliente (
    id_cliente SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    cpf VARCHAR(11) UNIQUE NOT NULL,
    data_nascimento DATE
);

-- TABELA: Agência
CREATE TABLE agencia (
    id_agencia SERIAL PRIMARY KEY,
    nome VARCHAR(50),
    cidade VARCHAR(50)
);

-- TABELA: Conta
CREATE TABLE conta (
    id_conta SERIAL PRIMARY KEY,
    id_cliente INT REFERENCES cliente(id_cliente),
    id_agencia INT REFERENCES agencia(id_agencia),
    saldo NUMERIC(10,2) DEFAULT 0
);

-- TABELA: Transação
CREATE TABLE transacao (
    id_transacao SERIAL PRIMARY KEY,
    id_conta INT REFERENCES conta(id_conta),
    tipo VARCHAR(20) CHECK (tipo IN ('Saque', 'Depósito',
'Transferência')),
    valor NUMERIC(10,2),
    data_transacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Crie a estrutura do banco de dados

Fatec Jacareí
Prof. Francisco de Moura

```
-- TABELA: Endereço
CREATE TABLE endereco (
    id_endereco SERIAL PRIMARY KEY,
    id_cliente INT REFERENCES cliente(id_cliente),
    rua VARCHAR(100),
    bairro VARCHAR(50),
    cidade VARCHAR(50),
    estado VARCHAR(2)
);
```

//Inserção de Dados

```
-- Clientes
INSERT INTO cliente (nome, cpf, data_nascimento) VALUES
('Carlos Andrade', '12345678900', '1985-02-10'),
('Mariana Costa', '98765432100', '1990-11-25');
-- Agências
INSERT INTO agencia (nome, cidade) VALUES
('Agência Centro', 'São Paulo'),
('Agência Norte', 'Rio de Janeiro');
```

```
-- Contas
INSERT INTO conta (id_cliente, id_agencia, saldo) VALUES
(1, 1, 1000.00),
(2, 2, 2500.50);

-- Transações
INSERT INTO transacao (id_conta, tipo, valor) VALUES
(1, 'Depósito', 500.00),
(1, 'Saque', 200.00),
(2, 'Depósito', 1000.00);

-- Endereços
INSERT INTO endereco (id_cliente, rua, bairro, cidade, estado) VALUES
(1, 'Rua das Flores, 123', 'Jardins', 'São Paulo', 'SP'),
(2, 'Av. Brasil, 456', 'Centro', 'Rio de Janeiro', 'RJ');
```

O que são Stored Procedures?

Fatec
Jacareí
Prof. Francisco de Moura

- **Definição:**
  - **Stored Procedures** são blocos de código SQL armazenados no banco de dados.
  - Permitem executar **processos automáticos** com um simples comando.
- **Por que usar Stored Procedures?**
  - Automatiza processos frequentes.
  - Melhora a segurança, evitando SQL direto nas aplicações.
  - Reduz tráfego entre aplicação e banco de dados.
- **Exemplo (Banco Escolar):**
  - Criar uma procedure para **cadastrar um novo aluno**, evitando repetição de código.
- **Exemplo (Sistema Bancário):**
  - Criar uma procedure para **realizar transferências entre contas** automaticamente.

Criando Stored Procedures no PostgreSQL

Fatec Jacareí
Prof. Francisco de Moura

- **Sintaxe básica:**

```
CREATE OR REPLACE PROCEDURE nome_da_procedure()
LANGUAGE plpgsql
AS $$
BEGIN
    -- Bloco de comandos SQL
END;
$$;
```

Estrutura de uma Procedure:

1. **CREATE PROCEDURE** → Cria a procedure.
2. **LANGUAGE plpgsql** → Define a linguagem usada.
3. **AS .....** → Define o bloco de código.
4. **BEGIN ... END** → Bloco de execução dos comandos SQL.

✅ Agora vamos criar exemplos práticos!

Exemplo: Procedure para Cadastrar Alunos (Banco Escolar)

Fatec Jacareí
Prof. Francisco de Moura

- **Problema:**
  Queremos criar uma procedure para **inserir um novo aluno** sem precisar repetir o comando INSERT manualmente.
- **Solução:** Criar uma **Stored Procedure** para cadastrar um aluno.

```
CREATE OR REPLACE PROCEDURE cadastrar_aluno(
    p_nome VARCHAR,
    p_idade INT,
    p_curso VARCHAR)
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO alunos (nome, idade, curso)
    VALUES (p_nome, p_idade, p_curso);
END;
$$;
```

Chamando a Procedure:

CALL cadastrar_aluno('Carlos Andrade', 20, 'Computação');

✅ **Agora podemos cadastrar alunos apenas chamando a Procedure!**

Exemplo: Procedure para Cadastrar Alunos (Banco Escolar)

Fatec Jacareí
Prof. Francisco de Moura

Problema:

Como faço para cadastrar mais alunos?

- Se quiser cadastrar **vários alunos de uma vez**, você pode chamar a procedure várias vezes seguidas, assim:

CALL cadastrar_aluno('Carlos Andrade', 20, 'Computação');

CALL cadastrar_aluno('Mariana Silva', 22, 'Engenharia');

CALL cadastrar_aluno('João Pereira', 19, 'Matemática');

Isso funciona bem para poucos registros, mas se precisarmos cadastrar **muitos alunos de uma vez**, podemos usar **laços (LOOP ou FOREACH) dentro da procedure** ou **inserir os dados direto em massa usando INSERT INTO ... SELECT**.

Exemplo: Procedure para Cadastrar Alunos (Banco Escolar)

Fatec Jacareí
Prof. Francisco de Moura

- Qual a vantagem de usar CALL ao invés de INSERT diretamente?

Comparação	CALL (Procedure)	INSERT direto
Reutilização	Pode ser chamado várias vezes sem reescrever a lógica	Precisa reescrever o comando toda vez
Manutenção	Fácil de modificar (basta alterar a procedure)	Mudanças exigem alteração em todo o código que usa INSERT
Segurança	Evita SQL direto na aplicação, reduzindo riscos de SQL Injection	Maior risco se não for bem tratado na aplicação
Performance	Reduz tráfego entre aplicação e banco, pois encapsula a lógica	Envia múltiplos comandos SQL ao banco de dados
Automação	Pode incluir regras de negócio e validações diretamente na procedure	Requer que a aplicação gerencie essas regras

Exemplo: Procedure para Cadastrar Alunos (Banco Escolar)

Fatec Jacareí
Prof. Francisco de Moura

- Qual a vantagem de usar CALL ao invés de INSERT diretamente?

- **Conclusão:**

  O CALL **(Procedure)** é mais **modular, reutilizável e seguro** do que escrever INSERT diretamente no código da aplicação.

  Além disso, facilita **manutenção** e **automação**, pois podemos adicionar regras e validações dentro da procedure.

Exemplo: Procedure para Transferência Bancária (Sistema Bancário)

Fatec
Jacareí
Prof. Francisco de Moura

- **Problema:**
  Precisamos criar um procedimento que permita **transferir dinheiro entre contas**, garantindo que:
  - O **remetente tenha saldo suficiente**.
  - O saldo seja **atualizado corretamente**.
- **Solução:** Criar uma **Stored Procedure** para transferências.

Exemplo: Procedure para Transferência Bancária (Sistema Bancário)

Fatec Jacareí
Prof. Francisco de Moura

```
CREATE OR REPLACE PROCEDURE realizar_transferencia(
    p_conta_origem INT,
    p_conta_destino INT,
    p_valor DECIMAL)
LANGUAGE plpgsql
AS $$
BEGIN
    -- Verifica se há saldo suficiente
    IF (SELECT saldo FROM contas WHERE id_conta = p_conta_origem) >= p_valor THEN
        -- Debita o valor da conta de origem
        UPDATE contas
        SET saldo = saldo - p_valor
        WHERE id_conta = p_conta_origem;

        -- Credita o valor na conta de destino
        UPDATE contas
        SET saldo = saldo + p_valor
        WHERE id_conta = p_conta_destino;
    ELSE
        RAISE EXCEPTION 'Saldo insuficiente para a transferência!';
    END IF;
END;
$$;
```

- **Executando uma transferência:**

```
CALL realizar_transferencia(1, 2, 500.00);
```

✅ **Agora as transferências podem ser feitas automaticamente com uma única chamada!**

Alterar Stored Procedures

Fatec Jacareí
Prof. Francisco de Moura

Problema:
Queremos mudar o nome de uma Procedure sem recriá-la do zero.

- **Solução:** No PostgreSQL, **não é possível renomear diretamente uma Procedure**, então precisamos:
  - Criar uma nova Procedure com o novo nome.
  - Excluir a Procedure antiga.

- **Passo 1 - Criar a nova Procedure:**

```
CREATE OR REPLACE PROCEDURE nova_transferencia(
    p_conta_origem INT,
    p_conta_destino INT,
    p_valor DECIMAL)
LANGUAGE plpgsql
AS $$
BEGIN
    -- Lógica da transferência aqui
END;
$$;
```

- **Passo 2 - Excluir a Procedure antiga:**

```
DROP PROCEDURE realizar_transferencia;
```

✅ Agora, a Procedure foi renomeada corretamente!

Excluir Stored Procedures

Fatec
Jacareí
Prof. Francisco de Moura

Problema:

Temos uma Procedure que não é mais necessária e queremos removê-la.

- **Solução:**

  DROP PROCEDURE nome_da_procedure;

- **Exemplo:**

  DROP PROCEDURE cadastrar_aluno;

✅ A Procedure **cadastrar_aluno** foi excluída com sucesso!

Ajustes nos Bancos de Dados

Fatec Jacareí
Prof. Francisco de Moura

- **Banco Escolar:**
  - Criamos a procedure **cadastrar_aluno**.
  - **Novo requisito:** Adicionar um campo *data_cadastro* para registrar quando o aluno foi inserido.
- **Banco do Sistema Bancário:**
  - Criamos a procedure **realizar_transferencia**.
  - **Novo requisito:** Criar uma tabela *historico_transferências* para armazenar registros das transferências.

Ajustes nos Bancos de Dados

Fatec
Jacareí
Prof. Francisco de Moura

- **Criando o histórico de transferências:**

```
CREATE TABLE historico_transferencias (
    id SERIAL PRIMARY KEY,
    id_conta_origem INT,
    id_conta_destino INT,
    valor DECIMAL(10,2),
    data_transacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- **Ajustando a Procedure para salvar histórico:**

```
INSERT INTO historico_transferencias (id_conta_origem, id_conta_destino, valor)
VALUES (p_conta_origem, p_conta_destino, p_valor);
```

✅ Agora o sistema **registra todas as transferências realizadas!**

Atividade Prática (Individual)

Fatec
Jacareí
Prof. Francisco de Moura

- **Criar** uma Procedura chamada **curso2** no **Banco Escolar**.
- **Renomear** essa Procedure para **novo_cadastro_curso**.
- **Excluir** a Procedure **novo_cadastro_curso**.
- **Criar** uma Procedure para **cadastrar um novo curso** no Banco Escolar.
- **Chamar** a Procedure e adicionar um curso.
- **Criar** uma Procedure para aplicar **um bônus de 5%** no saldo de todas as contas no Sistema Bancário.
- **Executar** a Procedure e verificar os novos saldos.

Entrega do Requisito (BDR.03)

Fatec
Jacareí
Prof. Francisco de Moura

- **O que deve ser entregue?**
  - ✅ **Stored Procedures aplicadas ao projeto ABP**.
  - ✅ **Criação de pelo menos 2 Procedures úteis para o projeto**.
  - ✅ **Uso correto de parâmetros para automatizar processos**.
  - ✅ **Requisito atendido: BDR.03 - Stored Procedures**.
- **Como será avaliado?**
  - ✓ Implementação correta das **Stored Procedures**.
  - ✓ Automação de processos no **banco de dados do projeto**.
  - ✓ Eficiência na execução dos procedimentos.

Referências Bibliográfica da Aula

Fatec
Jacareí
Prof. Francisco de Moura

📌 **Livros:**

Elmasri & Navathe (2010). Sistemas de Banco de Dados.

Silberschatz et al. (2011). Sistemas de Banco de Dados.

📌 **Links úteis:**

📎 PostgreSQL Docs
📎 W3Schools SQL Guide

Bibliografia Básica

Fatec
Jacareí
Prof. Francisco de Moura

- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro, Elsevier: Campus, 2004.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 7 ed. São Paulo: Pearson, 2018.
- SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. F. **Sistema de banco de dados**. Rio de Janeiro: Elsevier Brasil, 2016.

Bibliografia Complementar

Fatec
Jacareí
Prof. Francisco de Moura

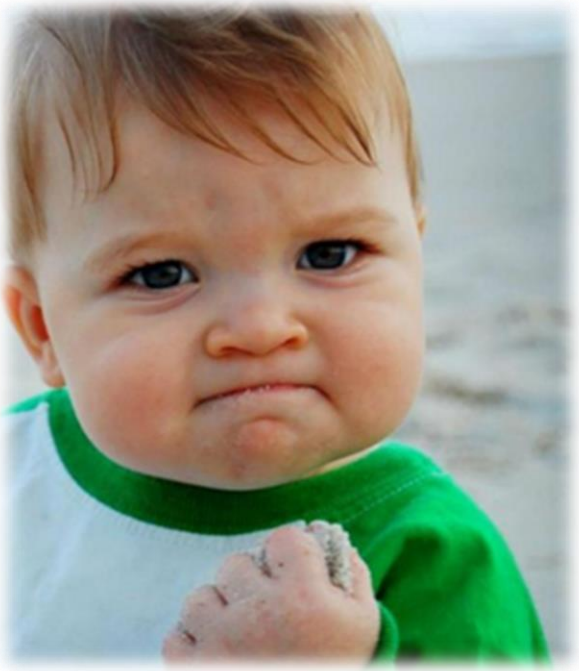
- BEAULIEU, A. **Aprendendo SQL**. São Paulo: Novatec, 2010.
- GILLENSON, M. L. **Fundamentos de Sistemas de Gerência de Banco de Dados**. Rio de Janeiro: LTC, 2006.
- MACHADO, F. N. R. **Banco de Dados:** Projeto e Implementação. São Paulo: Érica, 2005.
- OTEY, M; OTEY, D. **Microsoft SQL Server 2005:** Guia do Desenvolvedor. Rio de Janeiro: Ciência Moderna, 2007.
- RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento de Bancos de Dados**. 3 ed. Porto Alegre: Bookman, 2008.
- ROB, P; CORONEL, C. **Sistemas de Banco de Dados:** Projeto, Implementação e Gerenciamento**.** 8 ed. São Paulo: Cencage Learning, 2011.
- TEOREY, T; LIGHTSTONE, S; NADEAU, T. **Projeto e Modelagem de Bancos de Dados.** São Paulo: Campus, 2006.

Dúvidas?

Fatec
Jacareí
Prof. Francisco de Moura

Considerações Finais

Fatec
Jacareí
Prof. Francisco de Moura



Professor(a):
Lucineide Pimenta

Bom descanso à todos!