

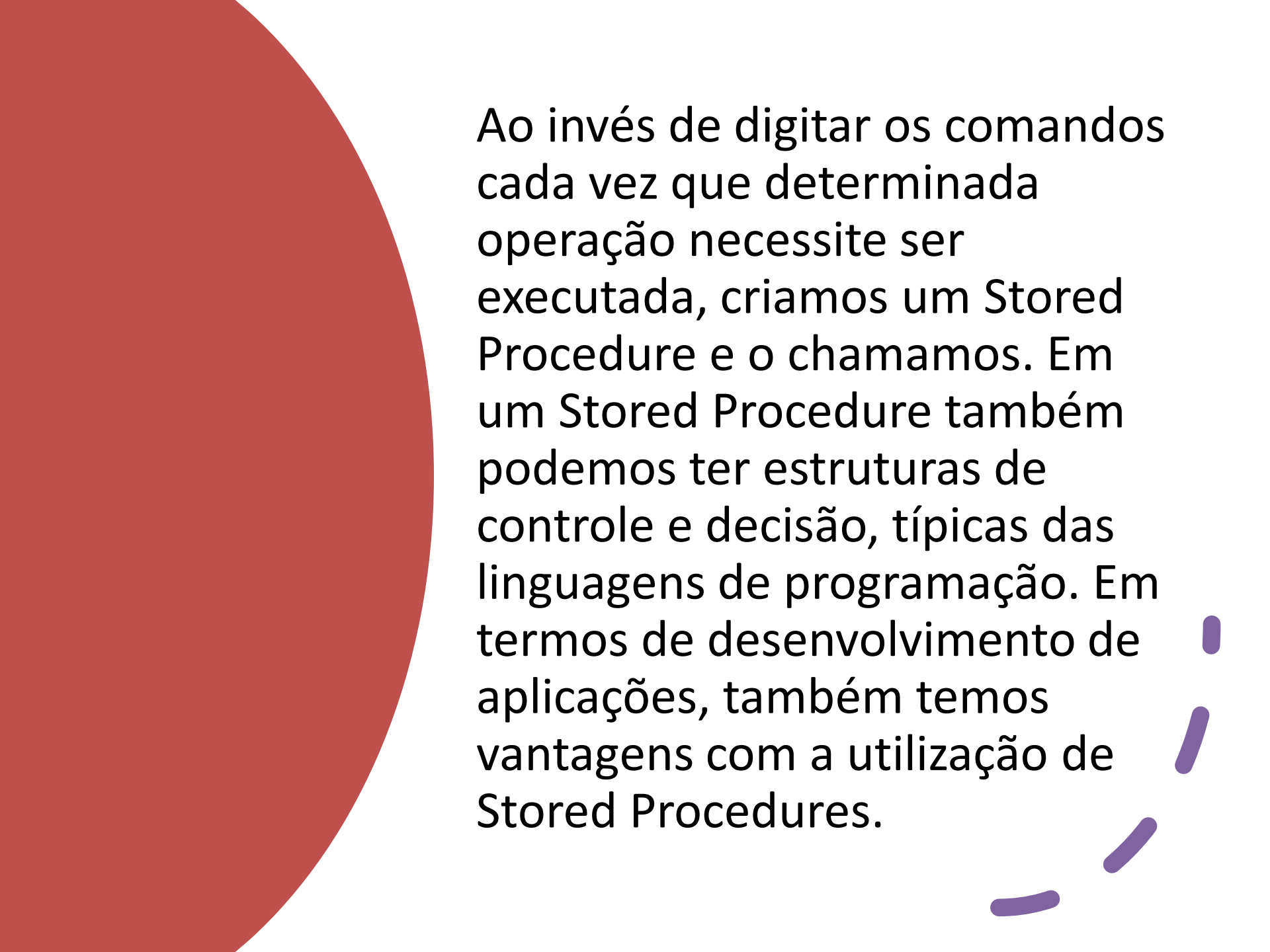


Stored Procedures no SQL Server

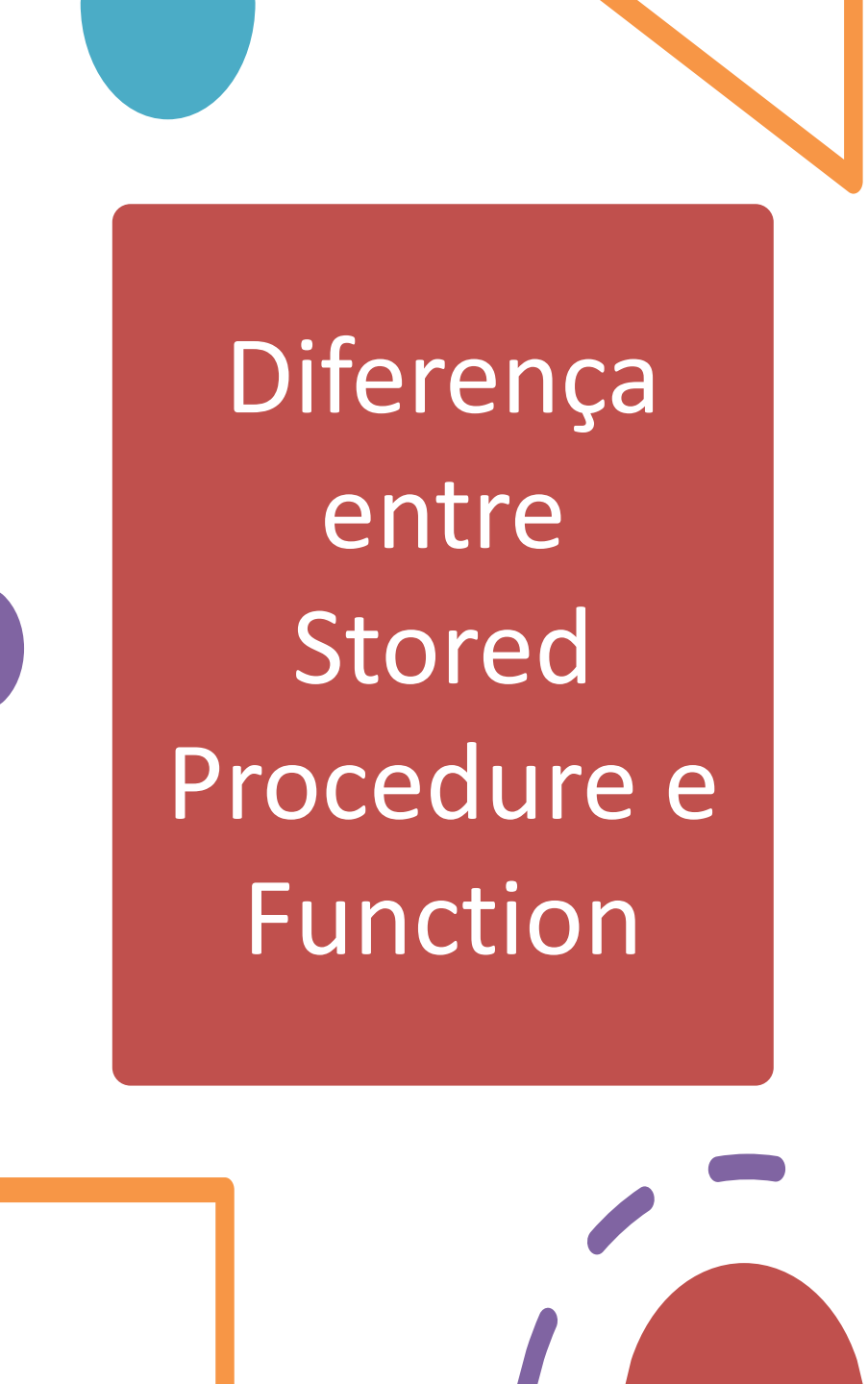
Banco de Dados II e III

Stored Procedure

Stored Procedure (procedimento armazenado) é um conjunto de comandos, ao qual é atribuído um nome. Este conjunto fica armazenado no Banco de Dados e pode ser chamado a qualquer momento tanto pelo SGBD (sistema Gerenciador de Banco de Dados) quanto por um sistema que faz interface com o mesmo. A utilização de Stored Procedures é uma técnica eficiente de executarmos operações repetitivas.



Ao invés de digitar os comandos cada vez que determinada operação necessite ser executada, criamos um Stored Procedure e o chamamos. Em um Stored Procedure também podemos ter estruturas de controle e decisão, típicas das linguagens de programação. Em termos de desenvolvimento de aplicações, também temos vantagens com a utilização de Stored Procedures.



Diferença entre Stored Procedure e Function

- A diferença entre a Stored Procedure e a Function, esta relacionada a obrigatoriedade que a function tem em retornar valores, enquanto a Stored Procedure não retorna nada.

Criando Stored Procedure

```
CREATE PROCEDURE nome_do_stored_procedure  
[  
    {@parametro  
    tipo_de_dados_parametro}[=valor_default] [output]  
]  
[,...n]  
AS  
    comando1,  
    comando2,  
    comando3,  
    ...,  
    comando2
```

Exemplo

```
Create Procedure BuscaCliente  
@nomeBusca varchar(30)  
As  
Select CodCli, NomeCli from Clientes  
Where nome Like '%' + @nomeBusca + '%'
```

Cria um procedimento que recebe um parâmetro (@nomeBusca) e mostra todos os clientes cujo nome contenha o nome informado.

Alterando Stored Procedure

```
ALTER PROCEDURE nome_do_stored_procedure  
[  
    {@parametro  
    tipo_de_dados_parametro}[=valor_default] [output]  
]  
[,...n]  
AS  
    comando1,  
    comando2,  
    comando3,  
    ...,  
    comando2
```

Exemplo

```
Alter Procedure BuscaCliente  
@codBusca int  
As  
Select * Clientes  
Where codcli = @codBusca
```

Cria um procedimento que recebe um parâmetro (@nomeBusca) e mostra todos os clientes cujo nome contenha o nome informado.

Trabalhando com Stored Procedure

Excluindo a Stored Procedure:

Drop Procedure NomedaSuaProcedure;

Exemplo → Drop Procedure nomeBusca

Executando a Stored Procedure:

Execute NomedaSuaProcedure;

Exec NomedaSuaProcedure;

Exemplo → Exec nomeBusca 'Maria'

Execute codBusca 2



Vantagens: Desempenho

Ex.: Seja a consulta

```
SELECT codigop, nome, COUNT(*)  
FROM Projeto p, Alocacao a  
WHERE p.codproj = a.codigop  
GROUP BY p.codproj, p.nome
```

Se vários usuários realizarem esta consulta o tráfego de rede será alto.

No entanto, se criarmos uma *stored procedure* para executar esta consulta, os usuários necessitaram apenas de um comando para executar a consulta anterior:

EXEC nomeProcedimento;



Sintaxe e Regras

Sintaxe

CREATE PROCEDURE <nome> [parâmetro]

AS <instrução SQL>

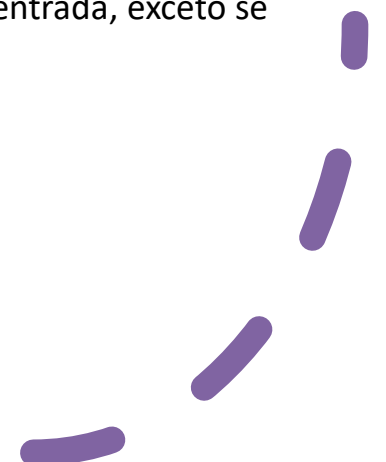
Regras

O **nome** da *procedure* deve seguir as regras para criação de identificadores

Nome do parâmetro deve iniciar por @ e deve ser único na lista de argumentos, seguido do seu tipo

@mes int, @ano int, @nome varchar

Todos os parâmetros são considerados de entrada, exceto se houver OUTPUT após sua definição



Stored Procedures Outros exemplos

```
CREATE PROCEDURE MostraEmpregadosDep  
    @nomeDep varchar(50)
```

```
AS
```

```
    SELECT e.mat, e.nome, e.endereco,  
    e.salario  
    FROM Empregados e, Departamento d  
    WHERE e.codD = d.codD and  
        d.nomeD = @nomeDep
```

Uma chamada a este procedimento seria:

```
EXEC MostraEmpregadosDep 'Informatica'
```



Stored Procedures -Outros exemplos

Exemplo: Criar uma procedure que exiba o título e a editora de cada livro.

```
CREATE PROCEDURE listar  
AS
```

```
SELECT l.titulo, e.nome  
FROM livro l, editora e  
WHERE l.codEdit =
```

```
e.codEdit
```

```
EXEC listar
```



Stored Procedures - Outros exemplos

- **Exemplo:** Criar uma *procedure* que exiba o total de salários pagos a um determinado setor da empresa.

```
CREATE PROCEDURE sptotalSal  
@setor char(3)  
AS
```

```
SELECT SUM(salario)  
FROM funcionario  
WHERE setor = @setor
```

```
EXEC sptotalSal 'INF'
```

Stored Procedures - Outros exemplos

```
create procedure spInsereAluno
@nomeAluno varchar(50)
,@cpfAluno char(11)
as
begin
    declare @idAluno int
    if exists (select cpfAluno from tbAluno where cpfAluno like
@cpfAluno)
    begin
        print ('Não é possível Cadastrar! CPF ' +@cpfAluno + 'CPF já
existe!')
    end
    else
    begin
        insert tbAluno(nomeAluno, cpfAluno) values
            (@nomeAluno, @cpfAluno)
        set @idAluno = (select max(idAluno) from tbAluno)
        print('Aluno '+@nomeAluno+' cadastrado com sucesso com
codigo '+convert(varchar(5), @idAluno) )
    end
end
end
```

Lista 1

Utilize o banco de dados escola e execute cada uma das procedures.

1. Criar uma stored procedure “Busca_Aluno” que receba o código do aluno e retorne seu nome e data de nascimento.
2. Criar uma stored procedure “Insere_Aluno” que insira um registro na tabela de Alunos.
3. Criar uma stored procedure “Aumenta_Preco” que, dados o nome do curso e um percentual, aumente o valor do curso com a porcentagem informada
4. Criar uma stored procedure “Exibe_Turma” que, dado o nome da turma exiba todas as informações dela.
5. Criar uma stored procedure “Exibe_AlunosdaTurma” que, dado o nome da turma exiba os seus alunos.

Lista 1

Usando o bdEscola criar procedures que executem o que se pede:

6- Criar uma stored procedure para inserir alunos, verificando pelo cpf se o aluno existe ou não, e informar essa condição via mensagem

7- Criar uma stored procedure que receba o nome do curso e o nome do aluno e matricule o mesmo no curso pretendido