**UNIVERSIDADE SENAI JOINVILLE**

**Felipe Rafael Rothbarth**

**OT7-BD – PROCEDURES**

**JOINVILLE**

**1. O que são PROCEDURES?**

Procedures, ou procedimentos armazenados, são blocos de código SQL armazenados diretamente no banco de dados. Elas são definidas uma vez e podem ser reutilizadas para executar tarefas específicas, como cálculos, validações ou manipulação de dados.

São semelhantes a funções, mas geralmente não retornam valores diretamente.

Facilitam a execução de operações complexas que envolvem múltiplas instruções SQL.

**2. Quando usar e quando se deve aplicar as PROCEDURES?**

Quando usar:

Quando há necessidade de automatizar tarefas repetitivas.

Em operações complexas que envolvem múltiplas instruções SQL.

Para encapsular lógica de negócio no banco de dados, reduzindo redundâncias.

Quando evitar:

Quando o código é mais adequado no nível da aplicação (backend), onde há maior flexibilidade para gerenciar lógicas complexas.

Em casos em que a manutenção do código SQL diretamente no banco de dados se torne difícil.

**3. Como criar uma PROCEDURE (estrutura)?**

Uma procedure é criada usando o comando CREATE PROCEDURE.

Estrutura básica:

DELIMITER //

CREATE PROCEDURE NomeDaProcedure (

IN parametroEntrada TipoDado,

OUT parametroSaida TipoDado

)

BEGIN

-- Bloco de instruções SQL

SELECT coluna INTO parametroSaida

FROM tabela

WHERE condicao;

END //

DELIMITER ;

**Componentes**:

DELIMITER: Altera o delimitador padrão para suportar múltiplas instruções.

IN: Indica um parâmetro de entrada.

OUT: Indica um parâmetro de saída.

BEGIN ... END: Define o bloco de código executado pela procedure.

Exemplo prático:

DELIMITER //

CREATE PROCEDURE ObterSalarioVendedor (

IN idVendedor INT,

OUT salario DECIMAL(10,2)

)

BEGIN

SELECT salario INTO salario

FROM vendedor

WHERE idvendedor = idVendedor;

END //

DELIMITER ;

**Chamando a Procedure:**

CALL ObterSalarioVendedor(2, @salarioResultado);

SELECT @salarioResultado;

**4. É seguro usar as PROCEDURES? Por quê?**

Sim, mas com ressalvas.

Vantagens de Segurança:

Controle de acesso: Podem ser configuradas para que usuários específicos tenham acesso apenas às procedures, sem visibilidade direta sobre as tabelas.

Prevenção de SQL Injection: Parâmetros usados corretamente ajudam a evitar manipulações maliciosas de consultas SQL.

Centralização: A lógica de negócios fica encapsulada no banco de dados, reduzindo dependência direta no código da aplicação.

Riscos:

Complexidade na manutenção: Procedures grandes ou mal documentadas podem ser difíceis de atualizar e depurar.

Dependência de plataforma: Procedures são específicas ao banco de dados usado (MySQL, SQL Server, etc.), dificultando a portabilidade.

**5. Principais instruções para usar as PROCEDURES**

CREATE PROCEDURE: Cria uma nova procedure.

CALL NomeDaProcedure(): Executa uma procedure armazenada.

DELIMITER: Altera o delimitador padrão para suportar blocos de código SQL.

IN/OUT/INOUT: Define os tipos de parâmetros.

IN: Apenas entrada.

OUT: Apenas saída.

INOUT: Entrada e saída.

SET: Define valores em parâmetros dentro da procedure.

DROP PROCEDURE NomeDaProcedure: Remove uma procedure existente.

Exemplo com instruções variadas:

DELIMITER //

CREATE PROCEDURE AtualizarSalarioVendedor (

IN idVendedor INT,

IN aumento DECIMAL(10,2),

OUT novoSalario DECIMAL(10,2)

)

BEGIN

UPDATE vendedor

SET salario = salario + aumento

WHERE idvendedor = idVendedor;

SELECT salario INTO novoSalario

FROM vendedor

WHERE idvendedor = idVendedor;

END //

DELIMITER ;

**Chamando e testando:**

CALL AtualizarSalarioVendedor(2, 500.00, @novoSalario);

SELECT @novoSalario;

**REFERENCIAS BIBLIOGRAFICAS**

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson, 2011.

DATE, C. J. Introdução a Sistemas de Bancos de Dados. 8. ed. São Paulo: Pearson, 2004.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 6. ed. Porto Alegre: Bookman, 2009.

MURACH, Joel. Murach's MySQL. 2. ed. Fresno: Mike Murach & Associates, 2019.

BEIGHLEY, Lynn. Aprendendo SQL: Uma Introdução Interativa. 2. ed. Rio de Janeiro: Alta Books, 2019.

ORACLE CORPORATION. Procedures and Functions. MySQL Documentation. Disponível em: https://dev.mysql.com/doc/. Acesso em: 11 nov. 2024.

STACK, Mark. MySQL Stored Procedures. MySQL Tutorials, 2020. Disponível em: https://www.mysqltutorial.org/mysql-stored-procedures-tutorial.aspx. Acesso em: 11 nov. 2024.

ELMASRI, Ramez; NAVATHE, Shamkant. Fundamentals of Database Systems. 7. ed. Boston: Pearson, 2015.