



INSTITUTO SUPERIOR DO LITORAL DO PARANÁ

Prof. Luiz Efigênio

Disciplina: Administração de Banco de Dados II

Tema: *Sistemas de Informação – Triggers e Stored Procedures*

1. Introdução

Nos **Sistemas de Informação**, os **bancos de dados** são a base para o armazenamento e manipulação de informações críticas. A administração eficiente dessas estruturas é essencial para manter a **integridade, segurança e desempenho** das operações.

Duas ferramentas poderosas disponíveis em **Sistemas Gerenciadores de Banco de Dados (SGBDs)** — como **MySQL, PostgreSQL, SQL Server e Oracle** — são as **Triggers (gatilhos)** e as **Stored Procedures (procedimentos armazenados)**.

Esses recursos permitem **automatizar tarefas, padronizar processos e garantir a consistência** das informações, reduzindo erros humanos e otimizando o fluxo de trabalho.

2. Trigger (Gatilho)

Uma **Trigger** é um bloco de código SQL que é executado **automaticamente** em resposta a um evento específico dentro do banco de dados, como a inserção, atualização ou exclusão de registros em uma tabela.

2.1. Funções e objetivos

- **Automação:** executa comandos sem intervenção do usuário.
- **Integridade dos dados:** impede operações inválidas ou inconsistentes.

- **Auditoria:** registra automaticamente mudanças nas tabelas.
- **Sincronização:** atualiza tabelas relacionadas em tempo real.

2.2. Estrutura básica de uma Trigger

2.3. Exemplo prático

```
CREATE TRIGGER nome_trigger
{BEFORE | AFTER} {INSERT | UPDATE | DELETE}
ON nome_tabela
FOR EACH ROW
BEGIN
    -- Código SQL a ser executado
END;

CREATE TRIGGER log_insercao AFTER INSERT ON alunos
FOR EACH ROW
BEGIN
    INSERT INTO log_alunos (id_aluno, data_acao, acao)
    VALUES (NEW.id, NOW(), 'Inserção de novo aluno');
END;
```

Explicação:

Toda vez que um novo registro é inserido na tabela **alunos**, uma linha correspondente é automaticamente gravada na tabela **log_alunos**, servindo como histórico de auditoria.

2.4. Boas práticas

- Evite colocar **lógica de negócio complexa** dentro de triggers.
- Documente bem a função de cada trigger.
- Use **nomes descritivos** e padronizados (ex: **trg_aft_insert_vendas**).
- Teste o desempenho — triggers mal planejadas podem afetar o tempo de resposta do banco.

3. Stored Procedure (Procedimento Armazenado)

As **Stored Procedures** são conjuntos de instruções SQL armazenadas e executadas diretamente no servidor de banco de dados. Elas funcionam como **funções reutilizáveis**, capazes de receber parâmetros de entrada e saída, processar informações e retornar resultados.

3.1. Vantagens

- **Melhor desempenho:** compiladas e otimizadas pelo SGBD.
- **Segurança:** evita SQL injection, pois os comandos são pré-definidos.
- **Centralização da lógica de negócio:** garante consistência em múltiplas aplicações.
- **Facilidade de manutenção:** alterações são feitas em um único local.

3.2. Estrutura básica

3.3. Exemplo prático

Uso da Procedure:

```
DELIMITER $$

CREATE PROCEDURE registrar_venda (
    IN id_cliente INT,
    IN valor_total DECIMAL(10,2)
)
BEGIN
    INSERT INTO vendas (cliente_id, valor, data_venda)
    VALUES (id_cliente, valor_total, NOW());
END$$

DELIMITER ;

CALL registrar_venda(3, 120.50);
```



3.4. Boas práticas

- Sempre **valide os parâmetros de entrada**.
- Utilize nomes padronizados (sp_, proc_, etc.).
- Mantenha o código limpo e modular.
- Prefira **Stored Procedures** para operações repetitivas e críticas.

4. Diferenças entre Trigger e Stored Procedure

Característica	Trigger	Stored Procedure
Execução	Automática (disparada por evento)	Manual (executada por comando CALL)
Dependência de evento	Sim	Não
Foco principal	Integridade e auditoria	Processos e regras de negócio
Chamada por aplicação	Não	Sim (pode ser chamada via API ou script)
Criação de logs	Muito comum	Menos comum



5. Aplicações em Sistemas de Informação

Em um **Sistema de Informação**, o uso de Triggers e Stored Procedures ajuda a:

- **Automatizar processos empresariais**, como controle de estoque ou cálculo de comissões;
- **Garantir consistência de dados**, evitando inconsistências em transações;
- **Aumentar a segurança**, limitando o acesso direto às tabelas;
- **Melhorar a performance**, centralizando operações no próprio servidor de banco de dados.

Por exemplo, em um sistema de vendas:

- Uma **Trigger** pode registrar toda inserção na tabela **vendas** em um log de auditoria;
- Uma **Stored Procedure** pode calcular automaticamente descontos e registrar a transação, reduzindo o risco de erros manuais.

6. Conclusão

Triggers e Stored Procedures são ferramentas indispensáveis na **Administração de Bancos de Dados**, permitindo criar sistemas de informação mais **inteligentes, seguros e automatizados**.

Quando bem implementadas, essas funcionalidades contribuem para a **eficiência operacional** e para a **qualidade dos dados**, aspectos essenciais em qualquer ambiente corporativo moderno.



7. Referências Bibliográficas

- DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 8ª ed. Rio de Janeiro: Elsevier, 2004.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 7ª ed. Pearson, 2019.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de Banco de Dados*. 6ª ed. AMGH, 2020.
- ORACLE Corporation. *Database PL/SQL Language Reference*. Oracle Documentation, 2023.
- MySQL Documentation. *MySQL 8.0 Reference Manual – Triggers and Stored Programs*. Oracle, 2024.



Atividade

Desenvolva **uma Trigger** e **uma Stored Procedure** relacionadas entre si, utilizando um cenário realista (como controle de estoque, vendas, biblioteca, ou cadastro de alunos).

Envie o **script SQL completo por e-mail até o dia 17**, com o **assunto:**
[ADB-4P] Trigger e Store Procedure