

SISTEMA DE GERENCIAMENTO DE LOJA DE DISCOS UTILIZANDO PARADIGMA DECLARATIVO COM SQL

Giovanni Braga Soares Vasconcelos¹, Antônio Heitor Gomes Azevedo¹, Cauã Maia de Souza Nara¹

¹ Escola de Negócios, Tecnologia e Inovação - ARGO, Centro Universitário do Estado do Pará, Brasil, 5º Período de Ciência da Computação.

{giovanni23070008, antonio23070017, caua23070005}@aluno.cesupa.br

Paradigmas de Linguagens de Programação.

Prof^a. Suzana Lustosa de Souza

Belém, Pará. Abril de 2025.

RESUMO

Este artigo apresenta o desenvolvimento de um sistema de gerenciamento para loja de discos utilizando o paradigma declarativo de programação, especificamente com a linguagem SQL (Structured Query Language). O objetivo é demonstrar como o paradigma declarativo pode ser aplicado para solucionar problemas reais de gestão de informação comercial, onde o foco está em "o que" deve ser feito e não em "como" deve ser executado. O sistema desenvolvido permite gerenciar artistas, discos, clientes e vendas, possibilitando consultas eficientes e manipulação de dados de forma declarativa.

Palavras-chave: Paradigma Declarativo; SQL; Sistema de Informação; Gerenciamento de Loja de Discos.

ABSTRACT

This paper presents the development of a record store management system using the declarative programming paradigm, specifically with SQL (Structured Query Language). The goal is to demonstrate how the declarative paradigm can be applied to solve real information management problems, where the focus is on "what" should be done rather than "how" it should be executed. The developed system allows managing artists, records, customers, and sales, enabling efficient queries and data manipulation in a declarative manner.

Keywords: Declarative Paradigm; SQL; Information System; Record Store Management.

1. INTRODUÇÃO

Na atualidade, os sistemas de informação tornaram-se essenciais para a gestão eficiente de dados em diversas organizações comerciais. Estes sistemas são desenvolvidos utilizando diferentes paradigmas de programação, cada um com suas características, vantagens e desvantagens. Neste contexto, o presente artigo tem como objetivo desenvolver um sistema de gerenciamento para loja de discos utilizando o paradigma declarativo, especificamente com a linguagem SQL.

Como destacado por Date (2004), o paradigma declarativo se caracteriza por descrever "o que" um programa deve realizar, sem especificar explicitamente "como" realizar a tarefa. Este paradigma contrasta com o imperativo, onde o programador define explicitamente os passos que o computador deve seguir. O SQL, como uma linguagem declarativa, permite que os desenvolvedores se concentrem na definição da estrutura dos dados e nas operações que desejam realizar, enquanto o sistema gerenciador de banco de dados (SGBD) determina a melhor maneira de executar essas operações.

Um sistema de gerenciamento para loja de discos representa um caso ideal para aplicação do paradigma declarativo, pois envolve operações complexas de consulta e manipulação de dados estruturados, como informações sobre discos, artistas, clientes e vendas. Ao utilizar SQL, podemos expressar essas operações de forma concisa e eficiente, permitindo que o SGBD otimize a execução, como observado por Garcia-Molina et al. (2008).

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Paradigmas de Programação

Os paradigmas de programação são diferentes abordagens ou estilos de programação que definem como os programas são estruturados e executados. Cada paradigma possui características distintas que influenciam a forma como os problemas são modelados e resolvidos. Segundo Van Roy e Haridi (2004), os principais paradigmas incluem:

Paradigma Imperativo: Baseia-se na execução sequencial de comandos que alteram o estado do programa. O programador especifica explicitamente "como" uma tarefa deve ser realizada, detalhando os passos que o computador deve seguir. Linguagens como C, Java e Python suportam este paradigma.

Paradigma Declarativo: Neste paradigma, o programador descreve "o que" o programa deve fazer, sem especificar "como" fazer. O sistema subjacente determina a melhor maneira de executar a tarefa. SQL, Prolog e linguagens funcionais como Haskell são exemplos de linguagens declarativas.

Paradigma Funcional: Um subconjunto do paradigma declarativo, focado em funções matemáticas e evitando mudanças de estado e dados mutáveis. Exemplos incluem Haskell, Lisp e partes de JavaScript.

Paradigma Orientado a Objetos: Baseia-se na noção de "objetos" que encapsulam dados e comportamentos. Linguagens como Java, C++ e Python suportam este paradigma.

2.2 Paradigma Declarativo e SQL

Date (2004) enfatiza que o paradigma declarativo permite que os programadores se concentrem no problema a ser resolvido, em vez de se preocuparem com os detalhes algorítmicos de como resolver o problema. Isso pode levar a um código mais conciso, legível e menos propenso a erros.

O SQL (Structured Query Language) é uma linguagem declarativa projetada especificamente para gerenciar e manipular dados em sistemas de gerenciamento de banco de dados relacionais (RDBMS). Silberschatz et al. (2012) destacam as características principais do SQL como linguagem declarativa:

1. **Abstração de detalhes de implementação:** O programador especifica quais dados deseja recuperar ou manipular, e o SGBD determina como executar a operação de forma eficiente.
2. **Linguagem baseada em conjuntos:** O SQL trabalha com conjuntos de dados, permitindo operações em massa em vez de processamento registro por registro.
3. **Otimização automática:** Os SGBDs modernos incluem otimizadores de consulta que determinam o plano de execução mais eficiente para cada consulta SQL.
4. **Separação entre dados e processamento:** O SQL separa claramente a definição dos dados (esquema) das operações realizadas sobre eles.

2.3 Sistemas de Gerenciamento para Lojas de Discos

Um sistema de gerenciamento para loja de discos é um sistema de informação projetado para facilitar a gestão de produtos musicais e serviços oferecidos por uma loja especializada. Ramakrishnan e Gehrke (2003) sugerem que estes sistemas geralmente incluem funcionalidades como:

1. **Catálogo:** Registro e organização de discos, artistas e gêneros musicais.
2. **Gestão de Vendas:** Registro e acompanhamento de vendas e faturamento.
3. **Controle de Estoque:** Monitoramento da disponibilidade de produtos.
4. **Gestão de Clientes:** Cadastro e gerenciamento de informações sobre os clientes da loja.
5. **Análise de Vendas:** Geração de relatórios e estatísticas para suporte à decisão.

Estes sistemas são tipicamente implementados utilizando bancos de dados relacionais, tornando o SQL uma escolha natural para sua implementação, conforme argumentado por Elmasri e Navathe (2018).

3. METODOLOGIA

O desenvolvimento do sistema de gerenciamento para loja de discos seguiu uma abordagem sistemática, com foco na aplicação do paradigma declarativo utilizando SQL. As etapas da metodologia incluíram:

1. **Análise de requisitos:** Identificação das funcionalidades essenciais para um sistema de loja de discos, incluindo gestão de artistas, discos, clientes e vendas.
2. **Modelagem de dados:** Seguindo as recomendações de Silberschatz et al. (2012), foi definido o esquema relacional do banco de dados, com identificação das entidades, atributos e relacionamentos necessários.
3. **Implementação do esquema:** Criação das tabelas e relacionamentos utilizando comandos SQL DDL (Data Definition Language).
4. **Desenvolvimento de consultas:** Elaboração de consultas SQL para implementar as funcionalidades identificadas, como listagem de discos disponíveis, controle de vendas e geração de relatórios.
5. **Testes e validação:** Verificação da corretude e eficiência das consultas implementadas.

Para implementação, foi escolhido o SQL padrão ANSI, compatível com a maioria dos sistemas gerenciadores de banco de dados relacionais, como MySQL, PostgreSQL e SQL Server, conforme recomendado por Date (2004).

4. PARADIGMA ESCOLHIDO

4.1 Justificativa da Escolha do Paradigma Declarativo

O paradigma declarativo foi escolhido para este trabalho devido às seguintes razões, alinhadas com as observações de Garcia-Molina et al. (2008):

1. **Foco no "o que" em vez do "como":** Permite expressar as operações desejadas de forma concisa, sem se preocupar com os detalhes de implementação.
2. **Adequação ao domínio do problema:** Sistemas de gerenciamento de loja de discos envolvem principalmente operações de consulta e manipulação de dados estruturados, para as quais o paradigma declarativo é particularmente adequado.
3. **Otimização automática:** O uso de SQL permite que o SGBD otimize a execução das consultas, resultando em melhor desempenho sem exigir otimização manual.

Menor propensão a erros: A natureza declarativa do SQL reduz a complexidade do código e, conseqüentemente, a probabilidade de erros.

4. **Expressividade:** O SQL permite expressar operações complexas de manipulação de dados de forma sucinta e legível.

4.2 Critérios de Linguagem

A escolha do SQL como linguagem declarativa para implementação do sistema foi baseada nos seguintes critérios, conforme discutido por Ramakrishnan e Gehrke (2003):

1. **Legibilidade:** O SQL é uma linguagem com sintaxe próxima à linguagem natural, facilitando a leitura e compreensão do código.
2. **Expressividade:** Permite expressar operações complexas de consulta e manipulação de dados de forma concisa.
3. **Padronização:** O SQL é uma linguagem padronizada (ANSI/ISO), com ampla adoção na indústria e suporte por diversos sistemas gerenciadores de banco de dados.
4. **Eficiência:** Os SGBDs modernos incluem otimizadores avançados que garantem a execução eficiente das consultas SQL, como destacado por Silberschatz et al. (2012).
5. **Abstração de alto nível:** O SQL oferece uma abstração de alto nível para operações em bancos de dados, permitindo que o desenvolvedor se concentre na lógica de negócio.
6. **Portabilidade:** Consultas SQL padrão podem ser executadas em diferentes SGBDs com mínimas modificações.
7. **Escalabilidade:** O SQL e os SGBDs que o suportam são projetados para lidar com grandes volumes de dados e múltiplos usuários concorrentes.

5. IMPLEMENTAÇÃO DO SISTEMA

O sistema de gerenciamento para loja de discos foi implementado utilizando SQL para definir o esquema do banco de dados e as operações de manipulação e consulta. A implementação seguiu as melhores práticas descritas por Elmasri e Navathe (2018) e consistiu em uma abordagem estruturada que priorizou a normalização dos dados, a integridade referencial e o desenvolvimento de consultas eficientes que aproveitam o poder do paradigma declarativo.

5.1 Definição do Esquema do Banco de Dados

O esquema relacional foi projetado seguindo os princípios da normalização até a terceira forma normal (3FN), para evitar redundâncias e garantir a integridade dos dados. O modelo de dados inclui as seguintes tabelas:

- **Artistas:** Armazena informações sobre os artistas musicais.
- **Gêneros:** Contém os gêneros musicais para classificação dos discos.
- **Discos:** Armazena informações sobre os discos, incluindo título, artista, gênero, preço e quantidade em estoque.
- **Clientes:** Mantém dados sobre os clientes da loja.
- **Vendas:** Registra as vendas realizadas, incluindo informações sobre cliente e valor total.

- **Itens Venda:** Relaciona vendas e discos, registrando quantidade e preço unitário de cada item vendido.

5.2 Implementação de Funcionalidades

As principais funcionalidades implementadas incluem:

1. **Gestão de discos:** Cadastro, atualização e consulta de informações sobre discos.
2. **Gestão de clientes:** Cadastro e manutenção de dados dos clientes da loja.
3. **Controle de vendas:** Registro de vendas e atualização automática de estoque.
4. **Consultas e relatórios:** Implementação de consultas para obter informações relevantes, como:
 - a. Discos disponíveis em estoque
 - b. Total de vendas por cliente
 - c. Discos mais vendidos
 - d. Vendas por período
 - e. Vendas por gênero musical

Para otimizar certas operações frequentes, foi criada uma visão (view) que facilita a análise de discos, consolidando informações relevantes de múltiplas tabelas, seguindo as recomendações de Garcia-Molina et al. (2008).

5.3 Código Fonte e Disponibilidade

O código SQL completo do sistema está disponível no GitHub, no repositório <https://github.com/giovannibragasv/trabalho-paradigmas>. O código inclui:

Definições de Esquema (DDL):

- Criação de tabelas
- Definição de chaves primárias e estrangeiras
- Criação de índices para otimização de consultas
- Definição de visões (views)
- Criação de funções e procedimentos armazenados

Manipulação de Dados (DML):

- Inserção de dados de exemplo
- Scripts para atualização de registros
- Exemplos de exclusão de dados

Consultas de Exemplo:

- Consultas básicas de seleção
- Consultas com junções

- Consultas com agregações
- Consultas analíticas complexas

Scripts de Manutenção:

- Backup e restauração
- Verificação de integridade
- Otimização de desempenho

6. RESULTADOS E DISCUSSÃO

A implementação do sistema de gerenciamento para loja de discos utilizando o paradigma declarativo com SQL demonstrou resultados significativos, que confirmam as vantagens deste paradigma destacadas por Date (2004) e Silberschatz et al. (2012).

6.1 Análise da Implementação

A simplicidade e clareza do código SQL desenvolvido evidenciam a adequação do paradigma declarativo para o problema abordado. As consultas implementadas demonstram como operações complexas podem ser expressas de forma concisa e legível. Por exemplo, a consulta para identificar os discos mais vendidos:

```
SELECT D.Titulo, A.Nome AS Artista, SUM(IV.Quantidade) AS QuantidadeVendida,  
       SUM(IV.Quantidade * IV.PrecoUnitario) AS ValorTotal  
FROM ItensVenda IV  
JOIN Discos D ON IV.DiscoID = D.DiscoID  
JOIN Artistas A ON D.ArtistaID = A.ArtistaID  
GROUP BY D.DiscoID  
ORDER BY QuantidadeVendida DESC;
```

Esta consulta, em apenas algumas linhas, realiza junções entre tabelas, cálculos agregados e ordenação dos resultados, demonstrando o poder expressivo do SQL como linguagem declarativa.

6.2 Comparação com Outras Abordagens

Comparando com uma possível implementação utilizando o paradigma imperativo, a solução declarativa apresentou vantagens significativas:

1. **Concisão do código:** O código SQL é substancialmente mais conciso que uma implementação imperativa equivalente, como observado por Van Roy e Haridi (2004).
2. **Foco na lógica de negócio:** A implementação declarativa permitiu focar no "o que" em vez do "como", reduzindo a complexidade accidental e destacando a complexidade essencial do problema.

3. **Otimização automática:** A otimização automática das consultas pelo SGBD representa uma vantagem considerável em termos de desempenho, sem exigir esforço adicional de desenvolvimento, conforme destacado por Ramakrishnan e Gehrke (2003).

6.3 Desafios Encontrados

Apesar das vantagens, alguns desafios foram encontrados durante a implementação:

1. **Limitações para lógica procedural:** Para certas operações, como a atualização de estoque após vendas, o SQL puro apresenta limitações, podendo requerer o uso complementar de linguagens procedurais.
2. **Complexidade de consultas avançadas:** Consultas envolvendo múltiplos níveis de subconsultas ou agregações complexas podem se tornar difíceis de compreender e manter, um ponto também observado por Elmasri e Navathe (2018).

7. CONCLUSÕES

A implementação de um sistema de gerenciamento para loja de discos utilizando o paradigma declarativo com SQL demonstrou a eficácia desta abordagem para o desenvolvimento de sistemas de informação comercial, confirmando as observações teóricas de Date (2004) e Garcia-Molina et al. (2008). As principais conclusões obtidas são:

1. **Eficácia do paradigma declarativo:** O paradigma declarativo mostrou-se altamente adequado para o domínio do problema, permitindo expressar operações complexas de manipulação de dados de forma concisa e eficiente.
2. **Expressividade do SQL:** A linguagem SQL permitiu implementar todas as funcionalidades necessárias com um código claro e conciso, facilitando a manutenção e evolução do sistema.
3. **Abstração apropriada:** A abstração fornecida pelo SQL, que permite focar no "o que" em vez do "como", resultou em um desenvolvimento mais rápido e menos propenso a erros, como sugerido por Silberschatz et al. (2012).
4. **Otimização automática:** A capacidade dos SGBDs de otimizar automaticamente a execução das consultas SQL representou uma vantagem significativa, eliminando a necessidade de otimização manual, um benefício também destacado por Ramakrishnan e Gehrke (2003).
5. **Limitações:** Apesar das vantagens, o paradigma declarativo e o SQL possuem limitações, especialmente para implementação de lógicas procedurais complexas, que poderiam requerer o uso complementar de linguagens procedurais, como observado por Elmasri e Navathe (2018).

O sistema desenvolvido, embora simples, demonstra o potencial do paradigma declarativo para aplicações de gerenciamento de dados comerciais. Em trabalhos futuros, o sistema poderia ser expandido com funcionalidades adicionais, como gestão de fornecedores, controle de pedidos e integração com sistemas de e-commerce, além de melhoras para a experiência do usuário, com a adição de um sistema de reviews, avaliações, e contas, simulando uma rede social.

REFERÊNCIAS

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 8ª ed. Rio de Janeiro: Elsevier, 2004.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 7ª ed. São Paulo: Pearson, 2018.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. **Database Systems: The Complete Book**. 2ª ed. Upper Saddle River: Pearson, 2008.

RAMAKRISHNAN, R.; GEHRKE, J. **Database Management Systems**. 3ª ed. New York: McGraw-Hill, 2003.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 6ª ed. Rio de Janeiro: Elsevier, 2012.

VAN ROY, P.; HARIDI, S. **Concepts, Techniques, and Models of Computer Programming**. Cambridge: MIT Press, 2004.