

Trabalho Prático 1 - Banco de Dados Relacional

Kaike Ribas da Silva Maciel, Aline Silmara Menezes Sales

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Av. Gen. Rodrigo Octávio, 6200 Setor Norte Campus Universitário, Manaus – AM – Brasil

aline.sales, kaike.maciel@icomp.ufam.edu.br

1. Introdução

Este documento apresenta o esquema de um banco de dados relacional desenvolvido para gerenciar informações de produtos vendidos em uma loja de comércio eletrônico, incluindo avaliações e comentários de clientes. O projeto segue a técnica ascendente (bottom-up) de modelagem e adere a formas normais de alto nível, como a Terceira Forma Normal (3NF) ou a Forma Normal de Boyce-Codd (BCNF), com o objetivo de garantir a integridade e eficiência dos dados. O sistema foi elaborado para armazenar e organizar informações sobre produtos, vendas, categorias e interações dos clientes, otimizando a consulta e análise desses dados.

Além disso, o banco de dados foi projetado para oferecer suporte a um dashboard, que permitirá realizar consultas complexas sobre o desempenho dos produtos e as avaliações dos usuários. O documento inclui o diagrama do banco de dados e um dicionário de dados detalhado, descrevendo as relações, atributos e restrições de integridade, assegurando a correta estruturação e funcionalidade do sistema implementado.

2. Esquema Banco de dados

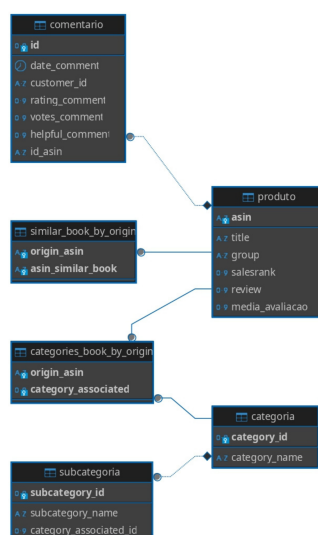


Figura 1. Modelo Relacional

O diagrama relacional apresentado é estruturado para modelar o banco de dados de uma loja de comércio eletrônico, onde são registrados dados de produtos, avaliações

de clientes, categorias e subcategorias. Este modelo visa garantir a integridade dos dados e otimizar as operações de busca, filtragem e recomendações de produtos similares, utilizando a técnica de normalização em um esquema relacional. Cada entidade foi projetada com chaves primárias e estrangeiras, assegurando a integridade referencial entre os dados e evitando redundâncias. Além disso, o modelo está em conformidade com as diretrizes de normalização de alto nível, assegurando a eliminação de anomalias de inserção, atualização e exclusão.

O modelo é composto por cinco entidades principais que se relacionam entre si:

2.1. Produto:

A tabela de produtos armazena as principais informações sobre cada item, incluindo o título, o grupo ao qual pertence, o ranking de vendas, o número de avaliações e a média das avaliações recebidas. O campo `asin`, que é o código identificador único de cada produto, é a chave primária dessa tabela. Essa entidade é central para o modelo, pois diversas outras entidades, como avaliações e produtos similares, fazem referência a ela.

2.2. Categoria e Subcategoria:

Para organizar os produtos de forma eficiente, foram criadas as tabelas de categoria e subcategoria. A tabela de categorias contém o nome de cada categoria, e cada subcategoria faz referência a uma categoria associada através de uma chave estrangeira. Essa hierarquia ajuda a estruturar os produtos de forma organizada, permitindo consultas mais rápidas e precisas ao filtrar itens com base em categorias e subcategorias.

2.3. Comentário (Review):

As avaliações feitas pelos clientes são armazenadas na tabela comentário, que registra a data do comentário, o identificador do cliente, a nota dada ao produto, o número de votos recebidos pelo comentário e os votos úteis. Esta tabela faz referência direta ao `asin` do produto avaliado, criando um vínculo entre as avaliações e os produtos.

2.4. Similar_book_by_origin (Produtos Similares):

Essa tabela tem a função de mapear produtos similares, estabelecendo uma relação entre o produto de origem e os produtos recomendados. Cada registro contém o `asin` do produto de origem e o `asin` do produto similar. A chave primária dessa tabela é composta por esses dois campos, e ambos fazem referência à tabela de produtos, o que permite uma recomendação consistente e precisa de itens relacionados.

2.5. Categories_book_origin:

Esta tabela intermediária vincula os produtos às suas respectivas categorias. O campo `origin_asin` faz referência à tabela de produtos, e o campo `category_associated` faz referência à tabela de categorias. Com isso, um produto pode ser associado a uma ou mais categorias, permitindo flexibilidade na classificação de itens.

Este modelo relacional foi desenhado para garantir a consistência e a eficiência nas operações de leitura e escrita, além de oferecer escalabilidade para futuras expansões e integrações de dados adicionais, como novas categorias, avaliações e recomendações de produtos similares. A implementação das chaves primárias e estrangeiras assegura que os dados sejam corretamente relacionados, sem perda de integridade.

Dicionário de Dados

Tabela: Produto

Atributo	Tipo de Dado	Restrições	Descrição
asin	VARCHAR(120)	PK, NOT NULL	Código identificador único do produto
title	TEXT	NOT NULL	Título ou nome do produto
group	TEXT	-	Grupo ou categoria do produto
salesrank	INTEGER	-	Ranking de vendas do produto
review	INTEGER	-	Número de avaliações do produto
media_avaliacao	FLOAT	-	Média das avaliações do produto

Tabela: Categoria

Atributo	Tipo de Dado	Restrições	Descrição
category_id	INTEGER	PK, NOT NULL	Identificador único da categoria
category_name	TEXT	UNIQUE, NOT NULL	Nome da categoria

Tabela: Subcategoria

Atributo	Tipo de Dado	Restrições	Descrição
subcategory_id	INTEGER	PK, NOT NULL	Identificador único da subcategoria
subcategory_name	TEXT	NOT NULL	Nome da subcategoria
category_associated_id	INTEGER	FK (Categoria.category_id), NOT NULL	Referência à categoria associada

Tabela: Comentário

Atributo	Tipo de Dado	Restrições	Descrição
id	SERIAL	PK, NOT NULL	Identificador único do comentário
date_comment	DATE	-	Data do comentário
customer_id	VARCHAR(120)	-	Identificador do cliente que comentou
rating_comment	INTEGER	-	Nota dada pelo cliente no comentário
votes_comment	INTEGER	-	Número de votos que o comentário recebeu
helpful_comment	INTEGER	-	Número de votos úteis que o comentário recebeu
id_asin	VARCHAR(120)	FK (Produto.asin), NOT NULL	Referência ao produto comentado

Tabela: Similar_book_by_origin

Atributo	Tipo de Dado	Restrições	Descrição
origin_asin	VARCHAR(120)	PK, FK (Produto.asin), NOT NULL	Referência ao produto de origem
asin_similar_book	VARCHAR(120)	FK (Produto.asin), NOT NULL	Referência ao produto similar

Tabela: Categories_book_by_origin

Atributo	Tipo de Dado	Restrições	Descrição
origin_asin	VARCHAR(120)	PK, FK (Produto.asin), NOT NULL	Referência ao produto de origem
category _associated	INTEGER	FK (Categoria.category_id), NOT NULL	Referência à categoria associada

3. Formas Normais Aplicadas

O diagrama relacional apresentado foi normalizado para garantir a consistência e otimização das operações no banco de dados. Abaixo, descrevemos como cada forma normal foi aplicada.

Primeira Forma Normal (1NF)

Para que o banco de dados esteja na **1ª Forma Normal (1NF)**, cada tabela deve garantir que seus atributos tenham *valores atômicos*. Ou seja, cada célula da tabela deve conter um único valor, e não um conjunto de valores. Além disso, deve haver uma *chave primária* que identifique exclusivamente cada registro.

- As tabelas *Produto*, *Categoria*, *Subcategoria*, *Comentário*, *Similar_book_by_origin* e *Categories_book_by_origin* atendem a esse requisito. Todos os atributos contêm valores atômicos, e cada tabela possui uma chave primária para identificar os registros de forma única.

Segunda Forma Normal (2NF)

Para que o banco de dados esteja na **2ª Forma Normal (2NF)**, ele deve estar em **1NF** e todos os atributos não chave devem ser *totalmente dependentes da chave primária*. Isso significa que não pode haver dependências parciais — nenhum atributo pode depender apenas de uma parte de uma chave primária composta.

- A tabela *Produto* está em **2NF**, pois todos os atributos (como *title*, *group*, *sales_rank*) dependem unicamente da chave primária *asin*. - A tabela *Comentário* também está em **2NF**, pois atributos como *rating_comment* e *votes_comment* dependem diretamente da chave primária *id*. - Para as tabelas com chaves primárias compostas, como *Similar_book_by_origin*, todos os atributos dependem das chaves compostas (*origin_asin* e *asin_similar_book*).

Terceira Forma Normal (3NF)

A **3ª Forma Normal (3NF)** exige que o banco de dados esteja em **2NF** e que *não haja dependências transitivas* entre os atributos. Isso significa que nenhum atributo não chave deve depender de outro atributo não chave.

- Em *Produto*, todos os atributos não chave (como *title* e *group*) dependem diretamente de *asin*, sem dependências transitivas. - A tabela *Subcategoria* está em **3NF**, pois o atributo *subcategory_name* depende da chave primária *subcategory_id*, e não de *category_associated_id*. - A tabela *Comentário* está em **3NF**, pois todos os atributos dependem diretamente da chave primária *id*.

Forma Normal de Boyce-Codd (BCNF)

A **BCNF** é uma versão mais rigorosa da **3NF**. Para estar em **BCNF**, a tabela deve estar em **3NF**, e cada determinante de uma dependência funcional deve ser uma superchave.

- O banco de dados atende aos requisitos de **BCNF** em todas as tabelas. Em *Similar_book_by_origin*, por exemplo, ambas as chaves (chave primária composta) determinam de forma única o relacionamento entre os produtos.

Quarta Forma Normal (4NF)

A **4ª Forma Normal (4NF)** lida com *dependências multivaloradas*. Para estar em **4NF**, a tabela deve estar na **BCNF** e não deve ter dependências multivaloradas.

- As tabelas no esquema não apresentam dependências multivaloradas, já que todos os atributos são valores atômicos. As relações entre produtos, categorias e comentários são bem definidas e não há situações em que um único registro de chave primária tenha múltiplos valores independentes para um atributo. Portanto, o banco de dados está em conformidade com a **4ª Forma Normal**.

Quinta Forma Normal (5NF)

A **5ª Forma Normal (5NF)**, ou **Forma de Projeção-Junção (PJ/NF)**, garante que não há anomalias de junção ao decompor tabelas. Isso significa que as tabelas não podem ser decompostas em tabelas menores sem que as junções necessárias criem dados inconsistentes.

- As tabelas *Similar_book_by_origin* e *Categories_book_by_origin* mostram exemplos de junção de tabelas sem anomalias. As relações entre produtos e categorias, e entre produtos e seus similares, são estruturadas de forma a garantir que as junções mantenham a integridade dos dados. Portanto, o banco de dados atende aos requisitos da **5ª Forma Normal**.

4. Conclusão

O desenvolvimento do esquema relacional para o banco de dados de uma loja de comércio eletrônico, apresentado neste trabalho, segue de maneira rigorosa os princípios da modelagem de dados e normalização. Ao aplicar as formas normais até a quinta forma (5NF), garantimos um sistema robusto, capaz de manter a integridade dos dados, evitar redundâncias e otimizar consultas e operações de escrita.

Através das entidades definidas — como produtos, categorias, subcategorias, avaliações e produtos similares — o banco de dados oferece uma estrutura eficiente para organizar grandes volumes de informações, permitindo a flexibilidade necessária para a gestão e análise dos dados de vendas e interações dos usuários. As implementações de chaves primárias e estrangeiras asseguram a consistência dos relacionamentos entre as tabelas, facilitando futuras expansões e integrações.

Com esse projeto, conseguimos não apenas implementar um banco de dados funcional e escalável, mas também demonstrar a importância de boas práticas de modelagem, como a normalização, para garantir o sucesso de operações complexas em um ambiente de comércio eletrônico. A arquitetura criada está preparada para suportar consultas avançadas, relatórios e dashboards, sendo uma solução eficaz para o gerenciamento de dados e a tomada de decisões estratégicas baseadas nas informações armazenadas.

Referências

- [1] Date, C. J. (2003). *An Introduction to Database Systems*. Addison-Wesley.
- [2] Elmasri, R., & Navathe, S. B. (2011). *Fundamentals of Database Systems* (6th ed.). Pearson.
- [3] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts* (6th ed.). McGraw-Hill.
- [4] Connolly, T., & Begg, C. (2014). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson.
- [5] Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems* (3rd ed.). McGraw-Hill.