

# Introdução à Banco de Dados - Trabalho I

Abel Severo Rocha, Ana Carla de Araújo Fernandes, Natasha Araújo Caxias

<sup>1</sup>Instituto de Computação (IComp) – Universidade Federal do Amazonas (UFAM)  
Av. Gen. Rodrigo Octávio, 6200, Coroado I – 69080-900 – Manaus – AM – Brasil

{abel.severo}{ana.carla}{natasha.caxias}@icomp.ufam.edu.br

## 1. Introdução

Este trabalho documenta o projeto e a implementação de um banco de dados relacional para um sistema de comércio eletrônico, com base no conjunto de dados Amazon Product Co-purchasing Network Metadata (*snap\_amazon.txt*). Inclui-se a elaboração do diagrama de esquema, o dicionário de dados e a justificativa das decisões de modelagem.

## 2. Observações sobre a modelagem

O esquema foi concebido a partir da técnica ascendente (bottom-up), sendo normalizado para a Forma Normal de Boyce-Codd (BCNF). Esta escolha garante que todas as dependências funcionais sejam baseadas exclusivamente em chaves candidatas, eliminando redundâncias.

A Figura 1 apresenta o diagrama elaborado para representar o esquema do banco de dados proposto. Nele, é possível observar a estrutura das relações, bem como a identificação das chaves primárias responsáveis por garantir a unicidade dos registros e das chaves estrangeiras, que estabelecem os vínculos entre as diferentes tabelas, assegurando a integridade referencial do modelo.

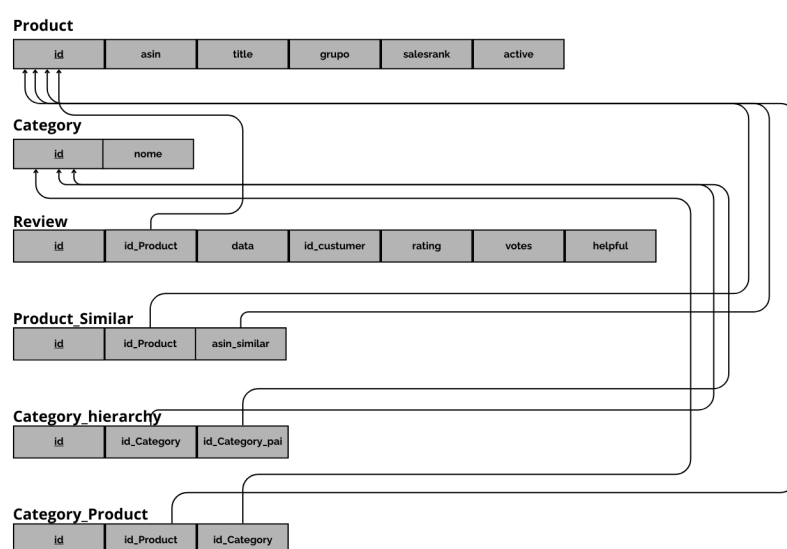


Figura 1. Esquema do banco de dados

Para tratar os produtos descontinuados, foi incluído de um novo atributo denominado *ativo* na tabela **Product**. Assim, foi possível representar de maneira explícita o status de cada produto, e manter que poderia comprometer a consistência dos dados. Dessa forma, preserva-se a integridade referencial, garantindo que as relações entre produtos e seus similares sejam mantidas, mesmo quando determinados itens deixam de estar disponíveis no catálogo.

Além disso, para lidar com atributos que poderiam violar os princípios de normalização avançada BCNF, foi criada uma *View*. Essa solução assegura que o esquema relacional esteja em conformidade com os padrões estabelecidos, ao mesmo tempo em que mantém a flexibilidade necessária para consultas e análises.

Nesse contexto, também se destacou a necessidade de tratar informações provenientes das avaliações (*reviews*). Identificou-se que alguns atributos, como *avg\_rating* e *downloaded*, eram redundantes, uma vez que poderiam ser calculados a partir de outros dados já armazenados. Dessa forma, a adoção de *Views*, combinada com a reformulação dos atributos relacionados às avaliações, contribui para uma estrutura mais consistente, clara e alinhada.



Figura 2. View com seus atributos

Durante o processo de modelagem e implementação do banco de dados, optou-se pela criação de índices com o objetivo de otimizar o desempenho das consultas. Essa prática contribui para reduzir o tempo de resposta em operações recorrentes, especialmente em tabelas com grande volume de registros.

Outro aspecto relevante é a separação dos comandos SQL em dois arquivos distintos: *schema.sql*, destinado à definição da estrutura do banco de dados, e *queries.sql*, responsável pelo armazenamento das consultas utilizadas. Essa divisão favorece a clareza do código e facilita a manutenção futura.

### 3. Dicionário de Dados

#### Tabela: Product

**Descrição da tabela:** Contém os registros de produtos comercializados na Amazon, sejam eles ativos ou descontinuados. Cada produto é identificado por um *id interno* e por um *ASIN* (Amazon Standard Identification Number), código único no ecossistema Amazon. Esta tabela é central no banco, sendo referenciada por outras entidades, como categorias, revisões e produtos similares.

#### Atributos:

- **id:** Identificador interno sequencial do produto.
- **asin:** Código único do produto na Amazon. Ex.: “0000557685”.
- **title :** Nome ou título do produto.

- **grupo:** Grupo ou classificação genérica do produto (ex.: “Books”, “Electronics”).
- **salesrank:** Classificação de vendas.
- **active:** Indica se o produto está ativo (TRUE) ou descontinuado (FALSE).

#### **Restrições de Integridade:**

- Chave Primária: id
- Restrição de Unicidade: asin deve ser único.
- Valor Padrão: active assume TRUE se não especificado.

#### **Tabela: Category**

**Descrição da tabela:** Armazena a lista de categorias disponíveis no sistema, organizando os produtos em grupos temáticos ou funcionais.

#### **Atributos:**

- **id:** Identificador único da categoria.
- **nome:** Nome textual da categoria (ex.: “Literatura Estrangeira”, “Tecnologia”).

#### **Restrições de Integridade:**

- Chave Primária: id
- Obrigatoriedade: nome não pode ser nulo.

#### **Tabela: Review**

**Descrição da tabela:** Armazena as avaliações feitas por clientes sobre os produtos. Cada linha corresponde a uma revisão única, vinculada a um produto.

#### **Atributos:**

- **id:** Identificador único da revisão.
- **id\_Product:** Produto avaliado, referenciando Product(id).
- **data:** Data da revisão.
- **id\_customer:** Identificador do cliente que fez a revisão.
- **rating:** Avaliação do produto em escala de 1 a 5 estrelas.
- **votes :** Número de votos recebidos pela revisão.
- **helpful:** Número de votos considerados úteis.

#### **Restrições de Integridade:**

- Chave Primária: id
- Chave Estrangeira: id\_Product → Product(id)
- Restrição de Domínio: rating deve estar entre 1 e 5
- Valores Padrão: votes e helpful iniciam em 0

#### **Tabela: Product\_Similar**

**Descrição da tabela:** Registra produtos considerados similares. Um produto pode ter diversos itens similares, auxiliando recomendações.

#### **Atributos:**

- **id:** Identificador único do registro.
- **id\_Product:** Produto de referência.
- **asin\_similar:** ASIN de um produto considerado similar.

#### **Restrições de Integridade:**

- Chave Primária: id
- Chaves Estrangeiras:
  - id\_Product → Product(id)
  - asin\_similar → Product(asin)
- Restrição de Unicidade: (id\_Product, asin\_similar) não pode se repetir

### **Tabela: Category\_Hierarchy**

**Descrição da tabela:** Representa a hierarquia das categorias, permitindo estruturar categorias em níveis (ex.: “Livros” > “Ficção” > “Fantasia”).

#### **Atributos:**

- **id:** Identificador do relacionamento hierárquico.
- **id\_Category:** Categoria filha.
- **id\_Category\_pai:** Categoria pai.

#### **Restrições de Integridade:**

- Chave Primária: id
- Chaves Estrangeiras:
  - id\_Category → Category(id)
  - id\_Category\_pai → Category(id)
- Restrição de Unicidade: (id\_Category, id\_Category\_pai) deve ser único
- Restrição de Domínio: id\_Category ≠ id\_Category\_pai

### **Tabela: Category\_Product**

**Descrição da tabela:** Implementa a relação muitos-para-muitos entre produtos e categorias. Um produto pode pertencer a várias categorias e uma categoria pode englobar diversos produtos.

#### **Atributos:**

- **id :** Identificador único do relacionamento.
- **id\_Product:** Produto associado à categoria.
- **id\_Category:** Categoria atribuída ao produto.

#### **Restrições de Integridade:**

- Chave Primária: id
- Chaves Estrangeiras:
  - id\_Product → Product(id)
  - id\_Category → Category(id)
- Restrição de Unicidade: (id\_Product, id\_Category) deve ser único