

Banco de Dados: Trabalho Prático 1

Gabriel da Silva Freitas, Gabriel Luciano Nunes, Guilherme Silveira Duarte

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

{gabriel.freitas,gabriel.luciano,guilherme.silveira}@icomp.ufam

1. Introdução

Este documento trata da criação de um banco de dados relacional para ser usado em uma loja de comércio eletrônico, com base nos metadados da rede de co-compra de produtos da Amazon. Descrevemos aqui o Diagrama de Esquema do Banco de Dados e um dicionário de dados, que apresenta as características de cada relação, atributo e restrição de integridade referencial ou de outro tipo que fazem parte do esquema. Além disso, abordamos as técnicas de normalização de alto nível utilizadas durante a criação do banco de dados.

2. Criação do Banco de Dados

Para a construção do esquema de relação, foi utilizada a técnica bottom-up, que consiste em criar entidades separadas e, em seguida, analisar as possíveis interações entre elas para formar as relações necessárias. Além disso, adotamos a forma normal 3FN, que tem como objetivo eliminar a redundância de dados no banco de dados. Após a análise das possíveis interações entre as entidades, foi obtido o esquema de relação, que pode ser visualizado na figura a seguir:

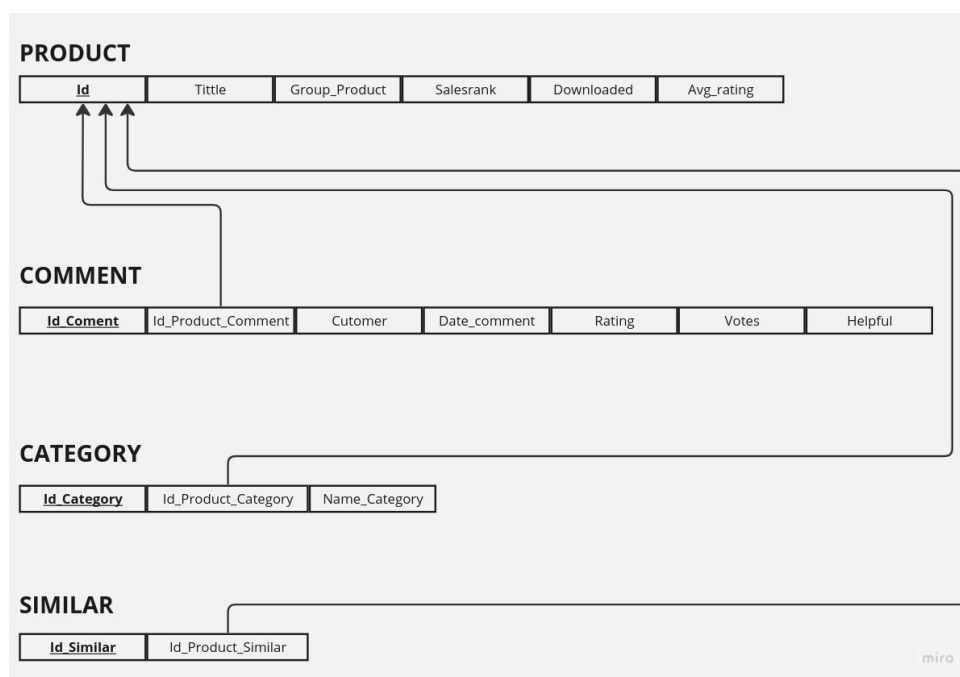


Figura 1. Diagrama dos esquemas do Banco de Dados

3. Forma Normal

A forma normal possui duas camadas que têm o poder de avaliar um banco de dados: A camada conceitual e a camada de implementação. A camada conceitual define como os usuários interpretam os esquemas de relação e o significado dos atributos, para tanto é necessário ter relações claras e bem definidas para realizar as consultas corretamente, por sua vez a camada de implementação ou armazenamento físico define como as tuplas são armazenadas e atualizadas [Navathe and Elmasri 2010].

A normalização de dados é um processo de análise da relação de esquemas baseados em chaves primárias com objetivo de minimizar redundâncias, minimizar operações de inserção, exclusão e atualização. Para isso, temos que a Terceira Forma Normal oferece tanto a simplicidade quanto a integração dos Dados. Visto que ela possui uma forma simples de organização - com as instâncias de cada tabela dependendo apenas de suas respectivas chaves primárias - conseguimos uma maior compreensão de usuários que estão em contato com banco de dados, facilitando a sua manutenção e gestão. Além disso, ela reforça a integridade de dados através da eliminação de redundância evitando alterações e modificações indesejadas.

4. Dicionário do Banco de Dados

Como mostrado na figura 1, criamos 4 relações: Product, Comment, Category e Similar. Descrevemos a seguir cada uma delas.

4.1. Product

Descreve os produtos digitais vendidos na loja. Ela possui seis atributos que são:

- **ID:** Identificador único de cada produto, esse valor é uma chave primária, portanto não pode ser repetida e nem nula. Também é o valor que é referenciado por chaves estrangeiras de outras tabelas.
- **Title:** Apresenta o título do produto é um varchar com tamanho máximo de 150 caracteres.
- **Group_Product:** Armazena o tipo do produto digital - que pode ser um livro, um CD, uma música - é um varchar com limite máximo de 40 caracteres.
- **Salesrank:** É um inteiro, indica a posição do produto no índice de vendas.
- **Downloaded:** Também inteiro, indica quantas vezes o produto foi baixado.
- **Avg_rating:** Inteiro, indica a média das avaliações do produto.

4.2. Comments

Esta tabela descreve as características dos comentários acerca dos produtos. Possui sete atributos:

- **Id_Comment:** Identificador único de cada comentário, este valor é uma chave primária.
- **Id_Product_Comment:** Este atributo é um identificador dos produtos em que o comentário está sendo referenciado, portanto esta é uma chave estrangeira que aponta para o atributo ID na tabela Product.
- **Cutomer:** Apresenta um código char(15), o qual identifica o usuário que comentou acerca de determinado produto.

- **Date_Comment:** Atributo do tipo date, o qual indica a data em que o comentário foi realizado.
- **Rating:** Valor inteiro que indica a média das notas dadas para o comentário.
- **Votes:** Valor inteiro que indica a quantidade de pessoas que avaliaram o comentário.
- **Helpful:** Valor inteiro, indica a utilidade do comentário.

4.3. Category

- **Id_Tupla_Category:** Identificador único de cada tupla da relação Category.
- **Id_Category:** Identificador de cada nome de categoria.
- **Id_Product_Category:** Este atributo é o identificador do produto no qual é classificado pelas categorias, por isso é uma chave estrangeira que aponta para o atributo ID na tabela Product.
- **Name_Category:** Indica o nome de cada categoria - Exemplo: pode ser um livro (book), de religião (Religion & Spirituality) ou de arte (Arts & Photography).

4.4. Similar

- **id_relation_similarity:** Atributo de tipo inteiro usado como identificador único de cada relação de similaridade entre dois produtos, esse valor é uma chave primária, portanto não pode ser repetida e nem nula.
- **id_product:** Atributo de tipo inteiro que identifica o produto com o qual se é estabelecida a similaridade. É a chave estrangeira que referencia a tabela Products no campo ID.
- **Product_Similar:** Atributo do tipo varchar que identifica o produto similar.

Referências

Navathe, S. and Elmasri, R. (2010). *Fundamentals of Database Systems*. Addison Wesley, 6th edition.