

Documentação: TP1 - Bancos de Dados I

Jamily Oliveira Pereira, Lorena Flávia Marinho de Souza, Lucas Saraiva de Moura

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

{ jamily.oliveira, lorena.flavia, lucas.saraiva}@icomp.ufam.edu.br

Apresentação

Objetivo deste trabalho prático é projetar e implementar um banco de dados sobre produtos vendidos em uma loja de comércio eletrônico, incluindo avaliações e comentários de usuários sobre estes produtos. O trabalho consiste na criação de um Banco de Dados Relacional contendo dados sobre compras de produtos e elaboração de um *Dashboard*, um painel para monitoramento dos dados de compra, gerando uma série de relatórios. O arquivo de entrada para o banco de dados será o *"Amazon product co-purchasing network metadata"*

1. Forma de Alto Nível

Uma Forma Normal é uma regra que deve ser obedecida por uma tabela para que esta seja considerada “bem projetada” [Heuser 2009]. Essas formas definem adequadamente os elementos (atributos) que serão utilizados na base de dados para evitar anomalias. O processo de normalização consiste em seis regras, entre elas a 4^o Forma normal.

Uma tabela está na quarta forma normal, quando, além de estar na 3FN, não tem dependências multivaloradas ou relacionamentos ternários ou superiores. Com base nisso, foi criado o esquema relacional abaixo seguindo o modelo 4FN:

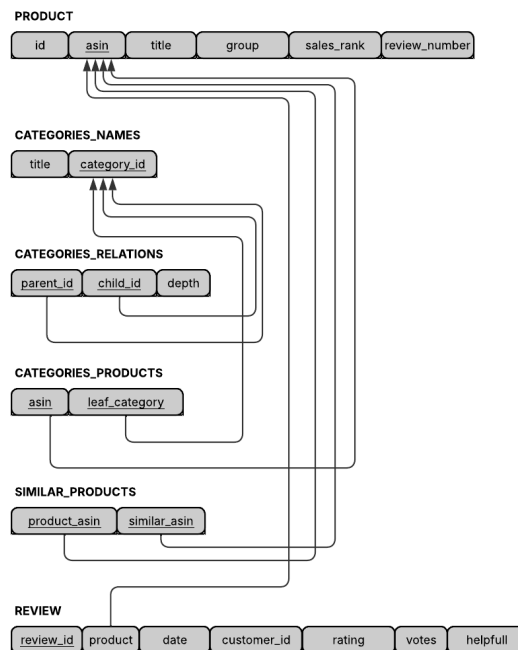


Figura 1. Diagrama do esquema relacional

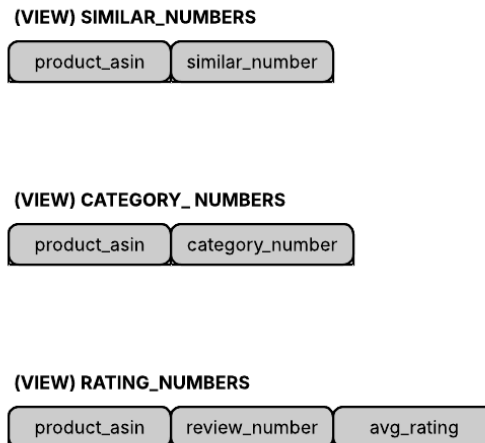


Figura 2. Diagrama do esquema relacional (views)

2. Dicionário de dados

Uma das fases da Forma Normal é a Decomposição do Banco de Dados, onde é dividido uma tabela complexa em tabelas menores e mais simples. Para este trabalho, foi decomposto a tabela referencial em 6 tabelas menores, nas quais são:

Tabela product

Essa tabela foi organizada para ter os atributos principais do produto:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
id	O id de uma linha no txt	SERIAL	Nada
asin	O identificador de um produto	VARCHAR(50)	PRIMARY KEY
title	Título do produto	TEXT	Nada
"group"	Grupo do produto	TEXT	Nada
sales_rank	Classificação de vendas	INT	Nada
review_number	Total de reviews	INT	Nada

Essa tabela possui como a chave primária o atributo asin, como forma de identificar um específico produto. Dessa forma, todas as outras tabelas se relacionarão com esse atributo, garantindo a coerência dos produtos.

Tabela Categories_Names

Para relacionar cada id de categoria ao seu nome, foi criado esta tabela possuindo os atributos:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
title	Título da categoria	TEXT	Nada
category_id	Identificador de uma categoria	VARCHAR(50)	PRIMARY KEY

Essa tabela possui como atributos como chave primária a category_id, com o objetivo de ter atributos únicos e não nulos para identificar uma categoria, facilitando a organização e a integridade dos dados.

Tabela Categories_Relations

Essa tabela serve para ter a categoria superior e a menor categoria em uma hierarquia de categorias, tendo como atributos:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
parent_id	Identificador da categoria pai	VARCHAR(50)	PRIMARY KEY e FOREIGN KEY
child_id	Identificador da categoria filha	VARCHAR(50)	PRIMARY KEY e FOREIGN KEY
depth	Profundidade entre as categorias	INT	Nada

Para isso, ela possui chave primária composta com os atributos parent_id e child_id, na qual referencia category_id da tabela category_names, para não criar uma categoria id diferente das existentes. Além disso, ela possui um valor inteiro da profundidade entre do parent_id até o child_id, com o intuito de facilitar a modelagem da hierarquia entre categorias.

Tabela Categories_Products

Esta tabela foi criada para relacionar as categorias ao produto, contendo os atributos:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
asin	Identificador do produto	INTEGER	PRIMARY KEY e FOREIGN KEY
leaf_category	Identificador da categoria-folha	INTEGER	PRIMARY KEY e FOREIGN KEY

Essa tabela possui chave primária composta pelos atributos asin, que referencia o identificador da tabela product, e leaf_category, que referencia a tabela categories_names. Ela relaciona os produtos às categorias às quais pertencem, eliminando as redundâncias geradas pela hierarquia de categorias. O atributo correspondente à categoria-folha permite inferir a categoria pai: para identificar todas as categorias relacionadas a um produto, basta conhecer as categorias-folhas às quais ele pertence. Para descobrir categorias mais gerais (não-folhas), é suficiente consultar a tabela categories_relations, verificando se uma categoria geral está relacionada como pai de alguma categoria-folha.

Tabela similar_products

Essa tabela existe para listar todos os produtos similar a um produto específico, contendo os atributos:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
product_asin	Identificador do produto	VARCHAR(50)	PRIMARY KEY e FOREIGN KEY
similar_asin	Identificador do produto similar	VARCHAR(50)	PRIMARY KEY e FOREIGN KEY

Essa tabela possui uma chave primária composta, na qual possui product_asin e similar_asin, que referenciam asin da tabela product. Esses atributos foram escolhidos como chave primária para garantir que cada produto tenha seu produto similar e que ambas as colunas não se repitam, de forma a criar uma repetição.

Tabela Review

Essa tabela serve para especificar tudo sobre a review de um produto, contendo os atributos:

Nome do Atributo	Descrição	Domínio	Restrições de Integridade
review_id	Identificador da review	VARCHAR(50)	PRIMARY KEY
product	Identificador do produto da review	TEXT	FOREIGN KEY
"data"	Data da review	DATE	Nada
customer_id	Identificador do cliente	VARCHAR(50)	Nada
rating	Avaliação do produto	INT	Nada
votes	Quantidade de votos na review	INT	Nada
helpfull	Quantidade de votos de utilidade, o quanto a review foi útil	INT	Nada

Para isso, possuí review_id para facilitar a busca de um review. Além disso, possuí um atributo product que referencia asin de product, para garantir a correlação de uma review com seu específico produto.

Views:

As *Virtual Views* são relações definidas por uma consulta sobre outras relações. As views virtuais não são armazenadas no banco de dados, mas podem ser consultadas como se existissem. O processador de consultas irá substituir a view por sua definição para executar a consulta. Embora as views (virtuais), não existam fisicamente, elas são definidas por uma expressão muito parecida com uma consulta. Também podem ser consultadas como se existissem fisicamente e, em alguns casos, podemos até mesmo modificar views.

No presente trabalho usamos as views para apresentar os valores que estavam presentes no arquivo original de entrada, porém não foram adicionadas em nenhuma das relações utilizadas.

View similar_count:

Esta view possui como atributos product_asin e total_similar. O objetivo desta view é contar quantos produtos similares um determinado produto possui e mostrar esse valor no atributo similar_count.

Dicionário de dados:

- product_asin: o identificador de um produto
- similar_count: contagem de quantos similares este produto tem

View category_count:

Esta view possui como atributos asin e total_category. O objetivo desta view é contar quantas categorias um determinado produto está inserido e mostrar esse valor no atributo total_category.

Dicionário de dados:

- asin: o identificador de um produto
- total_category: contagem de quantas categorias este produto está inserido

View rating_count:

Esta view possui como atributos product, total_reviews e avg_rating. O objetivo desta view é contar quantas reviews um determinado produto recebeu e mostrar também a média das avaliações que esse produto recebeu.

Dicionário de dados:

- product: o identificador de um produto
- total_reviews: contagem de quantas reviews este produto recebeu
- avg_rating: média das avaliações

3. Restrições**Referências**

Heuser, C. A. (2009). *Projeto de Banco de Dados*. Bookman, Porto Alegre, 6 edition. Instituto de Informática da UFRGS.