

Trabalho Prático I - Projeção e Implementação de um Banco de Dados Relacional

Karen H. F. Ponce de Leão - 22250541

Ricardo E. X. Silva - 22250556

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)

Av. Gen. Rodrigo Octávio, 6200 Setor Norte Campus Universitário, Manaus – AM – Brasil

{karen.ferreira, ricardo.silva}@icomp.ufam.edu.br

1. Análise dos Dados

Utilizando o exemplo fornecido na especificação do trabalho, observamos que o Banco de Dados seria semelhante ao apresentado na Figura 1. No entanto, armazenar dados dessa maneira seria inviável. Portanto, identificou-se a necessidade de aplicar as formas normais para reduzir a redundância e manter a integridade dos dados. Dessa forma, o banco de dados foi melhor estruturado com a adição de algumas tabelas.

ID	ASIN	TITLE	GROUP	SALES RANK	SIMILAR	CATEGORIES	REVIEWS
15	1559362022	Wake Up and Smell the Coffee	Books	518927	1559360968 1559361247 1559360828 1559361018 0743214552	Books[283155] Subjects[1000] Literature & Fiction[17] Drama[2159] United States[2160] Arts & Photography[1] Performing Arts[521000] Theater[2154]	2002-5-13 customer: A2IG0A66Y6O8TQ rating: 5 votes: 3 helpful: 2 2002-6-17 customer: A2OIN4AUH84KNE rating: 5 votes: 2 helpful: 1 2003-1-2 customer: A2HN382JNT1CIU rating: 1 votes: 6 helpful: 1 2003-6-27 customer: A39QMV9ZKRJXO5 rating: 4 votes: 1 helpful: 1 2004-2-17 customer: AUUVMSTQ1TXDI rating: 1 votes: 2 helpful: 0 2004-10-13 customer: ASXYF0Z3UH4HB rating: 5 votes: 1 helpful: 1

Figura 1. Esquema do Banco de Dados inicial

2. Estruturação do Banco de Dados com as Formas Normais

Primeiro vamos explicar como foi feita a modelação do nosso banco de dados, como chegamos nas tabelas finais através da normalização, na próxima sessão de Dicionário de Dados entraremos em detalhes sobre as tabelas.

2.1. Primeira Forma Normal (1FN)

Para começar a estruturar melhor nosso Banco de Dados, iniciamos pelo conceito mais básico: os atributos multivalorados e compostos. Esses atributos podem causar anomalias ao realizar operações no banco de dados. Para eliminar essas anomalias, utilizamos a primeira forma normal (1FN). A 1FN identifica a existência desses atributos e cria uma tabela separada para cada um deles, relacionando-os com a chave primária da tabela principal.

No nosso exemplo, temos como atributos multivalorados “Similar” que armazena mais de um valor para uma única chave (neste caso, o atributo ASIN). Já o atributo composto é “Review”, que guarda várias informações sobre um comentário de um produto.

No caso do atributo “Categories” podemos ver que além de ser um atributo multivalorado, que são as diversas categorias e subcategorias de um produto, ele também é composto, uma vez que cada categoria possui seu próprio ID.

Depois de aplicar a primeira forma normal, temos então uma nova versão das tabelas no nosso Banco de Dados que pode ser observada na Figura 2.

PRODUCT					SIMILAR	
ID	ASIN	TITLE	GROUP	SALESRANK	ASIN	SIMILAR
15	1559362022	Wake Up and Smell the Coffee	Books	518927	1559362022	1559360968
	1559362022				1559362022	1559361247
	1559362022				1559362022	1559360828
	1559362022				1559362022	1559361018
	1559362022				1559362022	0743214552

REVIEW						CATEGORY		
ASIN	DATE	CUSTOMER	RATING	VOTES	HELPFUL	ASIN	KEY_CATEGORY	NAME
1559362022	2002-5-13	A2IGOA66Y6O8TQ	5	3	2	1559362022	283155	Books
1559362022	2002-6-17	A2OIN4AUH84KNE	5	2	1	1559362022	1000	Subjects
1559362022	2003-1-2	A2HN382JNT1CIU	1	6	1	1559362022	17	Literature & Fiction
1559362022	2003-6-27	A39QMV9ZKRJXO5	4	1	1	1559362022	2159	Drama
1559362022	2004-2-17	AUUVMTQ1TXDI	1	2	0	1559362022	2160	United States
1559362022	2004-10-13	A5XYF0Z3UH4HB	5	1	1	1559362022	1	Arts & Photography
						1559362022	521000	Performing Arts
						1559362022	2154	Theater

Figura 2. Esquema do Banco de Dados Após a 1FN. Os atributos em *itálico* são as chaves estrangeiras para a tabela PRODUCT, ou seja, todas as tabelas tem uma chave estrangeira para PRODUCT esta sendo o atributo “ASIN”

2.2. Segunda Forma Normal (2FN)

A segunda forma normal (2FN) na normalização de bancos de dados trata da dependência entre a chave primária e os atributos não chave. Para que um banco de dados esteja na 2FN, todos os atributos não chave devem depender completamente e funcionalmente da chave primária. Se um atributo não chave depender apenas parcialmente da chave primária, é necessário criar uma nova tabela para esse atributo relacionando de outra forma.

Na nossa tabela **Category** nós armazenamos o “Id” (Key_Category) e o “Nome” de cada categoria, entretanto, o nome da categoria não depende completamente do nosso par de chaves (Asin, Key_Category), este “Nome” está relacionado somente com o ID da categoria, portanto dividimos essa tabela **Category** em duas: **Category** onde armazena ID e Nome de cada categoria, e uma tabela **Product_Category** onde armazena o ASIN do produto e o ID da categoria que ele pertence. Essa divisão elimina a dependência parcial, garantindo uma melhor normalização do banco de dados. Assim temos um novo esquema do Banco de Dados mostrado na Figura 3.

2.3. Terceira Forma Normal (3FN)

Esta etapa de normalização visa reduzir a duplicação de dados e garantir a integridade referencial. A Terceira Forma Normal (3NF) funciona com a ideia de que não pode haver

SIMILAR		PRODUCT					PRODUCT_CATEGORY		CATEGORY	
ASIN	SIMILAR	ID	ASIN	TITLE	GROUP	SALES RANK	ASIN	KEY_CATEGORY	KEY_CATEGORY	NAME
1559362022	1559360968	15	1559362022	Wake Up and Smell the Coffee	Books	518927	1559362022	283155	283155	Books
1559362022	1559361247						1559362022	1000	1000	Subjects
1559362022	1559360828						1559362022	17	17	Literature & Fiction
1559362022	1559361018						1559362022	2159	2159	Drama
1559362022	0743214552						1559362022	2160	2160	United States
							1559362022	1	1	Arts & Photography
							1559362022	521000	521000	Performing Arts
							1559362022	2154	2154	Theater

Figura 3. Esquema do Banco de Dados Após a 2FN. Os atributos em *itálico* são as chaves estrangeiras

um atributo não-chave dependendo de outro atributo não-chave. Quando isso ocorre, é necessário criar uma nova tabela para eliminar essa dependência. Além disso, todos os atributos não-chave devem depender completamente da chave primária (2NF).

No nosso esquema, não observamos tal erro, portanto não foi necessária a aplicação dessa normalização.

2.4. Quarta Forma Normal (4FN)

No esquema apresentado na Figura 3 nós estamos armazenando todas categorias e subcategorias em uma única tabela sem distinção, o que nos leva implicitamente a lidar com uma dependência multivalorada e sem ter uma clara distinção dos dados. Para tratar isso, aplicamos a Quarta Forma Normal que elimina esse tipo de dependência e evita possíveis redundâncias.

Sendo assim criamos uma tabela chamada **Product_Subcategory**, nesta tabela para cada chave da tabela **Product_Category** temos uma subcategoria associada a ela, formando então uma tabela separada para armazenar a informação de subcategoria de um produto, onde o conjunto desses três atributos serve como uma chave única.

Após a aplicação da 4FN chegamos ao fim da nossa estruturação do banco de dados. Para uma melhor visualização do esquema, na Figura 4 deixamos apenas o esquema das tabelas.

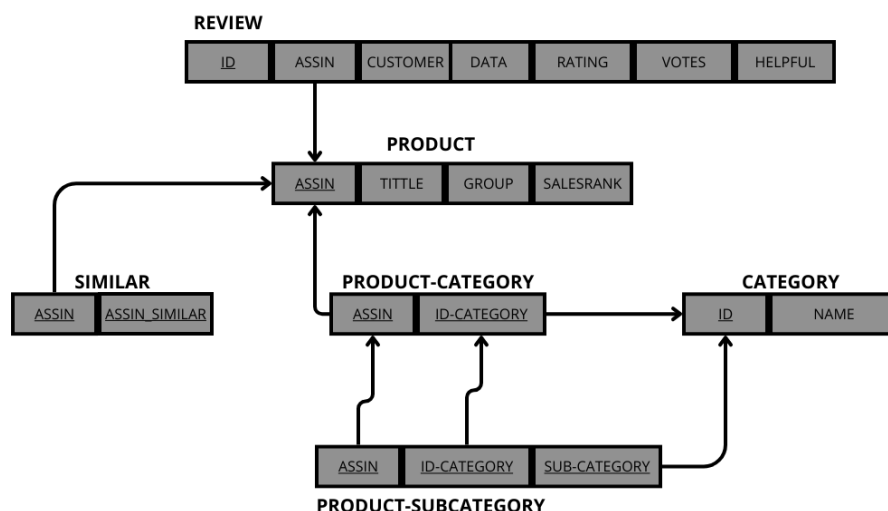


Figura 4. Esquema Final do Banco de Dados

3. Dicionário de Dados

Nesta seção vamos descrever agora cada detalhe de cada uma das tabelas do nosso Banco de Dados além de uma decisões de projetos relacionada a cada tabela.

3.1. Tabela Product

Nesta tabela desconsideramos o atributo “ID” pois vimos que o ASIN era o suficiente para identificar um produto, assim também sendo utilizado como Chave Primária na relação. Na tabela abaixo mostramos mais detalhes sobre a mesma. Desconsideramos produtos que foram descontinuados uma vez que não tinha informações o suficiente para agregar no nosso banco de dados.

Atributo	Tipo	Descrição	Restrição
Asin	VARCHAR	Código único do Produto	PRIMARY KEY
Tittle	VARCHAR	Título do produto	
Group	VARCHAR	Grupo que pertence	
Salesrank	INTEGER	Sua posição no Rank de vendas	

3.2. Tabela Category

Esta relação guarda todas as categorias (sendo elas principais ou subcategorias) juntamente com o seu ID.

Atributo	Tipo	Descrição	Restrição
Category_key	INTEGER	Código único da categoria	PRIMARY KEY
Name	VARCHAR	Nome da Categoria	NOT NULL

3.3. Tabela Product_Category

Esta relaciona um produto a sua categoria principal. Aqui levamos em conta que a primeira categoria que aparecia na lista de categorias do arquivo seria sua categoria principal, uma vez que a mesma se repetia em cada lista.

Atributo	Tipo	Descrição	Restrição
Asin	VARCHAR	Código do Produto	PRIMARY KEY FOREIGN KEY (Product.asin)
Id_category	INTEGER	Código da Categoria	PRIMARY KEY FOREIGN KEY (Category.category_key)

3.4. Tabela Product_Subcategory

Esta tabela faz a relação de um produto com suas subcategorias.

Atributo	Tipo	Descrição	Restrição
Asin	VARCHAR	Código do Produto	PRIMARY KEY FOREIGN KEY (Product_Category.asin)
Id_category	INTEGER	Código da Categoria	PRIMARY KEY FOREIGN KEY (Product_Category.id_category)
Subcategory	INTEGER	Código da Subcategoria	PRIMARY KEY FOREIGN KEY (Category.category_key)

3.5. Tabela Similar

Esta tabela guarda o Asin de um produto e todos os Asin de produtos similares a ele.

Atributo	Tipo	Descrição	Restrição
Asin	VARCHAR	Código do Produto	PRIMARY KEY FOREIGN KEY (Product.assin)
Assin_similar	INTEGER	Código da Produto Similar	PRIMARY KEY

3.6. Tabela Review

Nesta relação tivemos dificuldade de decidir nossa Chave Primária, uma vez que por exemplo, uma pessoa pode deixar mais de um comentário em um mesmo produto o que já eliminaria ambos atributos como chave, e não achamos também muito adequado utilizar uma chave composta de asin, customer e data por conta da complexidade de manipulação depois. Então colocamos um atributo ID auto incrementado, uma vez que cada tupla terá valores únicos independente dos demais atributos. Nossa relação então ficou como mostrado na tabela abaixo.

Atributo	Tipo	Descrição	Restrição
ID	SERIAL	ID único para cada Review	PRIMARY KEY
Assin	VARCHAR	Código da Produto Similar	FOREIGN KEY (Product.assin)
Data	DATE	Data que foi feita a Review	
Costumer	VARCHAR	Identificador do Cliente	
Rating	INTEGER	Avaliação do Produto	
Votes	INTEGER	Avaliação da Review	
Helpful	INTEGER	O quão útil foi a Review	