

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
ICC200 - BANCO DE DADOS I
2024/2

Professor: Altigran Soares da Silva (alti@icomp.ufam.edu.br)

Monitora: Giovanna Andrade (giovanna.andrade@icomp.ufam.edu.br)

Alunas:

Maria Gabriela Moraes de Sá - 22250537

Maria Giovanna Gonçalves Sales - 22251138

Juúle Yoshie Sarkis Hanada - 22251135

RELATÓRIO - TRABALHO PRÁTICO 1

1. Metodologia de Design do projeto desenvolvido

A implementação do projeto ocorreu nas seguintes etapas, conforme descrito em Fundamentals of Database System de Elmasri e Navathe (2010) [1]:

1.1 Coleta e Análise de Requisitos

- Usuários: Membros da equipe, monitores e professor da disciplina.
- Ambiente de execução: Local em máquinas de configurações variadas
- Sistemas Operacionais: MacOS, Linux e Windows.
- Linguagem de programação: Python.
- SGBD: PostgreSQL
- Acesso ao SGBD: O script deverá se conectar diretamente como PostgreSQL, sem bibliotecas de mapeamento Objeto-Relacional.
- Input: O arquivo “Amazon product co-purchasing network metadata”, que se trata de um dataset do Stanford Network Analysis Project (SNAP), disponível em: <https://snap.stanford.edu/data/amazon-meta.html>.
- Output: Resultado das consultas.
 - a) Dado um produto, listar os 5 comentários mais úteis e com maior avaliação e os 5 comentários mais úteis e com menor

avaliação

- b) Dado um produto, listar os produtos similares com maiores vendas do que ele
- c) Dado um produto, mostrar a evolução diária das médias de avaliação ao longo do intervalo de tempo coberto no arquivo de entrada
- d) Listar os 10 produtos líderes de venda em cada grupo de produtos
- e) Listar os 10 produtos com a maior média de avaliações úteis positivas por produto
- f) Listar a 5 categorias de produto com a maior média de avaliações úteis positivas por produto
- g) Listar os 10 clientes que mais fizeram comentários por grupo de produto

1.2 Design conceitual do banco de dados

Nesta etapa, foi criado um modelo conceitual de alto nível em forma de diagrama, priorizando as características de expressividade e formalidade.

Foram definidos nesta etapa: entidades, relacionamentos, atributos, atributos-chave, cardinalidade e restrições. Foi adotada a estratégia de design **Bottom-up**, que consiste em iniciar o esquema com abstrações básicas e adicionar novas abstrações posteriormente. Essa abordagem considera as relações entre atributos individuais como ponto inicial para construção do esquema.

Guidelines informais seguidos na construção do modelo conceitual:

- Semântica clara no esquema, mantendo cada tabela correspondente a uma entidade ou um relacionamento.
- Redução de informações redundantes, de forma que operações de inserção, deleção ou modificação não gerem anomalias.

- Redução de valores *null*, que podem ocorrer em excesso quando há atributos não se aplicam a todos os itens da tabela. Os valores *null* são problemáticos pois tem comportamentos imprecisos e podem gerar interpretações variadas.
- Impossibilitar a geração de tuplas espúrias (propriedade de *nonadditive join* e *lossless join*).

O modelo foi normalizado de acordo com a **Terceira Forma Normal** com o objetivo de minimizar redundâncias de dados e anomalias de inserção, deleção e atualização. Para isso, cada esquema foi analisado em relação às suas chaves e dependências funcionais.

Regras aplicadas na normalização:

- Primeira Forma Normal: atributos devem ser atômicos e indivisíveis.
- Segunda Forma Normal: atributos que não são chave devem ser funcionalmente dependentes da chave primária. Se a chave primária for composta, eles não podem ser dependentes de apenas uma parte da chave primária, e sim da chave completa.
- Terceira Forma Normal: atributos não primários não devem ser transitivamente dependentes da chave primária, ou seja, um atributo não pode ser funcionalmente dependente de outro atributo que não seja a chave primária.

O diagrama resultante desta etapa se encontra na seção 2 do relatório.

1.3 Escolha do DBMS

O alvo do projeto é o PostgreSQL [2] conforme a especificação do trabalho, que se trata de um Sistema Gerenciador de Banco de Dados Objeto Relacional (ORDBMS) de código aberto.

1.4 Design lógico do banco de dados (*data model mapping*)

Nesta etapa, o modelo conceitual foi mapeado para o SGBD específico PostgreSQL. O resultado desta etapa foi a linguagem de definição

de dados (DDL) do PostgreSQL que especifica o esquema de banco de dados.

1.5 Design físico do banco de dados

Em relação ao design físico do banco de dados, não foi realizada nenhuma alteração na configuração do SGBD, optando-se por manter a configuração padrão do PostgreSQL.

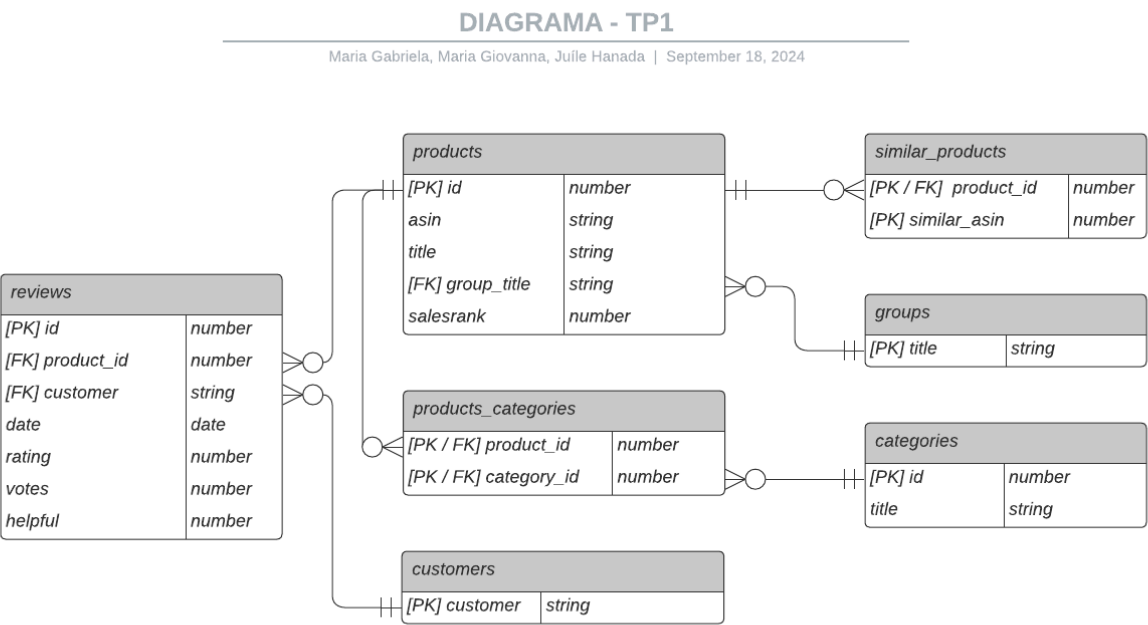
1.6 Implementação do banco de dados

Esta etapa inclui a implementação e execução do script 1 (tp1_3.2.py), que realiza a criação e a população do esquema de banco de dados a partir do processamento do arquivo de entrada.

1.8 Operação do banco de dados

Esta etapa inclui a implementação e execução do script 2 (tp1_3.3.py), que realiza consultas no banco de dados a partir da entrada fornecida pelo usuário.

2. Diagrama



3. Dicionário de Dados

Os dicionários de dados consistem em um conjunto de metadados sobre o banco de dados que contém, entre outras informações [1]:

- Descrição do esquema de banco de dados
- Informação sobre o projeto físico do banco de dados, como estrutura de armazenamento, caminhos de acesso, tamanhos de arquivos etc
- Descrição dos tipos de usuários, suas responsabilidades e direitos de acesso
- Descrições em alto nível das transações e aplicações do banco de dados e o relacionamento entre usuários e transações.
- A relação entre as transações dos bancos de dados e os itens referenciados por eles
- Estatísticas de uso como frequências de queries, transações e contagem de acesso a diferentes porções do banco
- A história de qualquer alteração realizada no banco de dados e suas aplicações, incluindo a descrição dos motivos das mudanças (proveniência de dados).

3.1 Tabelas e atributos

PRODUCTS			
Armazena informações dos produtos cadastrados.			
Nome da Variável	Tipo	Constraint	Descrição
id	number	PK	Identificador único do produto.
asin	string		Código ASIN único do produto contendo 10 caracteres alfanuméricos.
group_title	string	FK	Referência para a tabela groups, representando o grupo ao qual o produto pertence.

title	string		Nome ou título do produto.
salesrank	number		Posição do produto no ranking de vendas.

A tabela products possui os seguintes relacionamentos:

- Relacionamento 1:N com a tabela reviews, significando que um produto pode ter várias avaliações.
- Relacionamento 1:N com a tabela similar_products, significando que um produto pode ter vários produtos similares.
- Relacionamento 1:N com a tabela products_categories, significando que um produto pode pertencer a várias categorias de produtos.
- Relacionamento N:1 com a tabela groups, significando que o produto pertence a um grupo.
- Relacionamento N:N com a tabela categories, significando que o produto pode estar em várias categorias e as categorias podem conter vários produtos. Esse relacionamento levou à criação da tabela products_categories.

REVIEWS			
Armazena avaliações feitas por clientes em relação aos produtos.			
Nome da Variável	Tipo	Constraint	Descrição
id	number	PK	Identificador único da avaliação.
product_id	number	FK	Referência para a tabela products, representando o produto avaliado.
customer	string	FK	Referência para a tabela customers, representando o cliente que realizou a avaliação.

date	date		Data da avaliação.
rating	number		Nota dada pelo cliente ao produto.
votes	number		Número de votos que a avaliação recebeu.
helpful	number		Número de votos úteis para a avaliação.

A tabela reviews possui os seguintes relacionamentos:

- Relacionamento N:1 com a tabela products, significando que a avaliação está ligada a um produto.
- Relacionamento N:1 com a tabela customers, significando que a avaliação é feita por um cliente.

SIMILAR_PRODUCTS			
Relaciona produtos com outros produtos similares.			
Nome da Variável	Tipo	Constraint	Descrição
product_id	number	PK, FK	Referência para a tabela products, representando o produto principal.
similar_asin	number	PK	ASIN do produto similar.

GROUPS			
Armazena os diferentes grupos dos produtos.			
Nome da Variável	Tipo	Constraint	Descrição
title	string	PK	Nome ou título do grupo ao qual o produto pertence.

similar_asin	number	PK	ASIN do produto similar.
--------------	--------	----	--------------------------

CATEGORIES			
Armazena as categorias de produtos.			
Nome da Variável	Tipo	Constraint	Descrição
id	number	PK	Identificador único da categoria.
title	string		Nome da categoria.

PRODUCTS_CATEGORIES			
Relaciona produtos com suas categorias.			
Nome da Variável	Tipo	Constraint	Descrição
product_id	number	PK, FK	Referência para a tabela products.
category_id	number	PK, FK	Referência para a tabela categories.

CUSTOMERS			
Armazena informações dos clientes.			
Nome da Variável	Tipo	Constraint	Descrição
customer	string	PK	Identificador único do cliente.

3.2 Restrições de integridade

- Chaves Primárias (PK): Asseguram que cada registro seja único em sua respectiva tabela. As PKs estão presentes em todas as tabelas.

- Chaves Estrangeiras (FK): Mantêm a integridade referencial entre tabelas, garantindo que as relações entre os registros sejam válidas.

3.3 Usuários

- Administradores: Têm acesso completo ao banco de dados para criar, modificar ou excluir tabelas e registros. São responsáveis pelo gerenciamento de produtos, categorias, e grupos, além de manter a integridade das associações entre produtos e similares.
- Clientes/Usuários Finais: São representados na tabela customers e podem adicionar avaliações e interagir com os produtos.

4. Referências

- [1] ELMASRI, Ramez; NAVATHE, Shamkant B. Fundamentals of Database Systems, 6th edition. Addison Wesley, 2010. Capítulos 10 e 11
- [2] PostgreSQL: The world's most advanced open source relational database:
<https://www.postgresql.org/>
- [3] PostgreSQL Python <https://www.postgresqltutorial.com/postgresql-python/>