Documentação do Esquema do Banco de Dados Relacional

Alex Bruno Silveira Viana - 22152261 Gabrielly Ferreira Rodrigues - 22152262 Gilmar Eduardo Costa do Couto Júnior - 22152247

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

{gabrielly.rodrigues, alex.viana, gilmar.junior}@icomp.ufam.edu.br

1. Formalização do Esquema Relacional

De acordo com [Elmasri and Navathe 2010], um banco de dados relacional bem estruturado segue os princípios da normalização, garantindo a integridade dos dados e minimizando a redundância. O primeiro passo para esse processo é a **Primeira Forma Normal** (1NF), que estipula que o domínio de um atributo deve incluir apenas valores atômicos (indivisíveis). Além disso, cada atributo em uma tupla deve possuir um único valor dentro de seu domínio. Avançando para a **Segunda Forma Normal** (2NF), uma tabela atinge este estado se estiver na 1NF e exibir dependência funcional onde todos os atributos não-chave são funcionalmente dependentes da chave primária completa, eliminando dependências parciais. A **Terceira Forma Normal** (3NF), baseada na 2NF, enfatiza a prevenção de dependências transitivas. Neste estágio, uma tabela está na 3FN se estiver na 2FN, e todos os atributos não-chave são mutuamente independentes, garantindo que nenhum atributo não-chave dependa de outro atributo não-chave. Dessa forma, a equipe utilizou essas características como referencial para a normalização das tabelas.

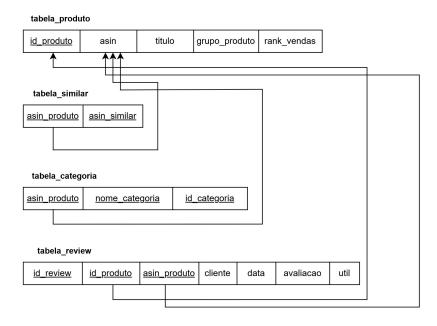


Figure 1. Diagrama correspondente ao Banco de Dados Relacional da equipe.

2. Dicionário de Dados

2.1. Tabela: tabela_produto

- id_produto: Identificador único do produto (Chave primária).
- asin: Código ASIN do produto.
- grupo_produto: Categoria ou grupo ao qual o produto pertence.
- rank_vendas: Classificação de vendas do produto.
- Restrições de Integridade:
 - Chave primária: id_produto.
 - Restrição de unicidade: asin (UNIQUE: garante que todos os valores nas colunas designadas sejam únicos em cada linha da tabela).

2.2. Tabela: tabela similar

- asin_produto: Código ASIN do produto principal (Chave primária, Chave estrangeira referenciando tabela_produto).
- asin_similar: Código ASIN do produto similar (Chave primária, Chave estrangeira referenciando tabela produto).
- Restrições de Integridade:
 - Chave primária: asin_produto, asin_similar.
 - Chave estrangeira: asin_produto referencia tabela_produto (Chave primária).

2.3. Tabela: tabela_categoria

- asin_produto: Código ASIN do produto (Chave primária, Chave estrangeira referenciando tabela_produto).
- nome categoria: Nome da categoria do produto.
- id_categoria: Identificador da categoria.
- Restrições de Integridade:
 - Chave Primária: asin_produto, nome_categoria, id_categoria.
 - Chave estrangeira: asin_produto referencia tabela_produto (Chave primária).

2.4. Tabela: tabela_review

- id_review: Identificador único da avaliação do tipo SERIAL (Chave primária).
- id_produto: Identificador único do produto ao qual a avaliação pertence (Chave estrangeira referenciando tabela_produto).
- asin_produto: Código ASIN do produto ao qual a avaliação pertence (Chave estrangeira referenciando tabela_produto).
- cliente: Identificador do cliente que fez a avaliação.
- data: Data da avaliação.
- avaliação: Classificação da avaliação.
- util: Número de votos úteis.
- Restrições de Integridade:
 - Chave erimária: id_review, id_produto, asin_produto.
 - Chave estrangeira: id_produto referencia tabela_produto (Chave primária).
 - Chave estrangeira: asin_produto referencia tabela_produto (Chave primária).

3. Execução dos Scripts

Esta seção detalha como o usuário pode executar os códigos, explicitando os comandos que devem ser utilizados no terminal para o funcionamento pleno do programa.

3.1. tp1_3.2.py

Para executar o código do arquivo "tp1_3.2.py", é preciso utilizar os seguintes comandos:

- python tp1_3.2.py --setup:
 Este comando é responsável por estabelecer a conexão com o banco de dados e criar as tabelas que serão utilizadas pelo programa.
- python tp1_3.2.py --insert:

 Comando que aciona a função que irá ler as informações do arquivo de entrada (neste caso amazon-meta.txt) e estabelecerá a conexão com o banco de dados, criando as tabelas que serão utilizadas pelo programa.
- python tp1_3.2.py --drop:
 Comando que finaliza a conexão com o banco de dados. Vale ressaltar que é preciso que o usuário saia do pgAdmin 4 antes de fechar a conexão em si.

3.2. tp1 3.3.py

Para executar o código do arquivo "tp1_3.3.py", é necessário realizar os seguintes passos:

• python tp1_3.3.py --dashboard'

Em seguida, o programa pedirá o ASIN de um produto para realizar as consultas no Banco de Dados:

```
Digite o ASIN do produto para iniciar:
```

Dessa forma, basta escrever um ASIN e pressionar "Enter", que o programa verificará se o produto existe para então apresentar o Dashboard completo dele.

3.3. Observações

- Altere no arquivo 'bd.ini' as informações da conexão de forma que correspondam às informações do seu PostgreSQL.
- Faça a mesma verificação no arquivo 'banco_de_dados.py' na linha 12: self.conn = psycopg2.connect(dbname='postgres', user='postgres', password='root', host='localhost')

A parte dbname='postgres' refere-se ao nome padrão do banco de dados ao qual a conexão é estabelecida inicialmente. Esta configuração geralmente é usada para administração e configuração do PostgreSQL, e em muitos casos, não é necessário alterá-la. Recomenda-se revisar e ajustar apenas as informações de usuário e senha conforme necessário para a configuração específica do ambiente.

References

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