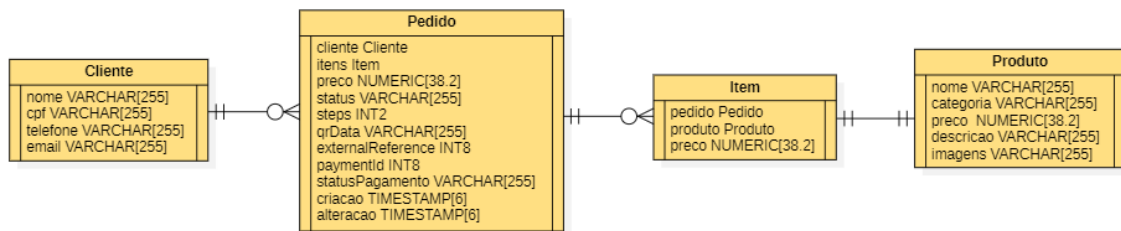


Aplicação TechFood – Fase 3

Modelagem de dados



No projeto optamos em utilizar na estrutura de aplicação o banco de dados PostgreSQL que é um banco de dados relacional, SQL.

Optamos por este banco de dados, visando algumas vantagens como:

- **Open Source**, uma solução de código aberto. Onde nesse momento por se tratar de uma solução acadêmica nos poupa de gastar com licenciamento, diferente da utilização de um Oracle, por exemplo;
- **Modelo Relacional Estruturado** onde os dados são organizados em tabelas, e as relações entre essas tabelas são claramente definidas. Melhorando a estrutura e integridade referencial entre as entidades.
- **Consultas Complexas** suportando **joins**, **subconsultas**, **views** e **procedures**, altamente sofisticadas;
- **Segurança Robusta** com controle avançado de acesso a dados;
- **Desempenho e Escalabilidade** com a possibilidade de crescimento da rede e consequentemente do número de clientes e pedidos. O PostgreSQL suporta um grande volume de dados e permite ser escalado de acordo com o crescimento dos dados;
- **Comunidade Ativa**, que contribui constantemente para o desenvolvimento do software, criando extensões, ferramentas e melhorando a documentação.

Para nos apoiar no processo de integração e criação deste banco de dados na nuvem optamos por utilizar o Amazon RDS. Quando unimos a solução da Amazon com o banco PostgreSQL estamos falando de uma solução poderosa e fácil de usar para quem precisa de um banco de dados relacional robusto e escalável, sem a sobrecarga de gerenciar a infraestrutura e as tarefas administrativas.

Estrutura

- **Cliente**

id_cliente (PK): Identificador único do cliente

nome: Nome do cliente

email: E-mail do cliente

telefone: Telefone do cliente

data_registro: Data de registro do cliente

- **Produto**

id_produto (PK): Identificador único do produto

nome: Nome do produto

descricao: Descrição do produto

preco: Preço do produto

estoque: Quantidade disponível em estoque

- **Pedido**

id_pedido (PK): Identificador único do pedido

id_cliente (FK): Chave estrangeira referenciando Cliente

data_pedido: Data do pedido

status: Status do pedido (e.g., "Em preparação", "Pronto", "Entregue")

valor_total: Valor total do pedido

- **ItemPedido**

id_item_pedido (PK): Identificador único do item no pedido

id_pedido (FK): Chave estrangeira referenciando Pedido

id_produto (FK): Chave estrangeira referenciando Produto

quantidade: Quantidade do produto no pedido

preco_unitario: Preço unitário do produto no momento do pedido

subtotal: Subtotal do item (quantidade * preço_unitario)

Relacionamentos

Cliente para Pedido: Um cliente pode fazer vários pedidos (Relacionamento 1).

Pedido para ItemPedido: Um pedido pode ter vários itens (Relacionamento 1).

Produto para ItemPedido: Um produto pode estar em vários itens de diferentes pedidos (Relacionamento 1).

Diagrama Entidade-Relacionamento (DER)

Cliente (1) ←---- (N) Pedido

Pedido (1) ←---- (N) ItemPedido

Produto (1) ←---- (N) ItemPedido

Descrição dos Relacionamentos

Cliente e Pedido: Um cliente pode fazer múltiplos pedidos, portanto, temos um relacionamento de 1 para N entre Cliente e Pedido.

Pedido e ItemPedido: Cada pedido pode ter múltiplos itens, o que nos leva a um relacionamento de 1 para N entre Pedido e ItemPedido.

Produto e ItemPedido: Um produto pode estar em vários pedidos diferentes, então o relacionamento entre Produto e ItemPedido também é de 1 para N.

Essas tabelas e relacionamentos representam uma estrutura que permite rastrear os pedidos dos clientes, os produtos que cada pedido contém e os valores associados. Isso forma a base do sistema da lanchonete e pode ser facilmente estendido conforme necessário, por exemplo, adicionando promoções, cupons de desconto, ou dados adicionais dos produtos e clientes.