

## Projeto de Banco de Dados – E-commerce

### Introdução

Este projeto implementa a **modelagem lógica de um banco de dados para um sistema de e-commerce**, contemplando clientes PF e PJ, múltiplas formas de pagamento, controle de entregas com status e rastreio, além de relacionamentos entre fornecedores, produtos e pedidos.

O objetivo é aplicar conceitos de **modelagem conceitual e lógica (EER)**, criar o **script SQL de criação do esquema**, inserir dados para testes e elaborar **consultas complexas** que demonstrem o uso de diferentes cláusulas SQL.

### Modelagem Lógica

#### Entidades e Relacionamentos

- **Cliente**: pode ser PF ou PJ (restrição de exclusividade entre CPF e CNPJ).
- **Produto**: associado a um fornecedor.
- **Fornecedor**: fornece produtos.
- **Vendedor**: pode ser analisado se também é fornecedor.
- **Pedido**: vinculado a um cliente.
- **ItemPedido**: detalha os produtos de cada pedido.
- **Pagamento**: múltiplas formas possíveis por pedido.
- **Entrega**: status e código de rastreio.

### Scripts

- schema.sql → Criação do banco de dados e tabelas com constraints.
- insert.sql → Inserção de dados de teste (clientes, produtos, pedidos, pagamentos e entregas).
- queries.sql → Consultas SQL complexas para análise dos dados.

### Consultas SQL

As queries de exemplo demonstram:

- Recuperações simples com **SELECT**
- Filtros com **WHERE**
- Atributos derivados com **SUM**
- Ordenações com **ORDER BY**

- Filtros em grupos com **HAVING**
- Junções complexas com **JOIN**

Exemplos de perguntas respondidas:

- Quantos pedidos foram feitos por cada cliente?
- Algum vendedor também é fornecedor?
- Relação de produtos, fornecedores e estoques.
- Relação de nomes dos fornecedores e nomes dos produtos.
- Clientes que gastaram mais de R\$ 1000.

## Próximos Passos

1. Crie um repositório no GitHub chamado ecommerce-db.
2. Adicione os arquivos README.md, schema.sql, insert.sql e queries.sql.
3. Faça o commit e push para o repositório.

## Como Executar

1. Clone este repositório:
- ```
git clone https://github.com/seuusuário/ecommerce-db.git
```
2. Execute o script schema.sql em seu SGBD (MySQL ou MariaDB).
  3. Popule o banco com insert.sql.
  4. Teste as consultas em queries.sql.

## 5. Conclusão

6. Este projeto demonstra a aplicação prática de **modelagem de banco de dados** em um cenário de e-commerce, cobrindo desde a criação do esquema até consultas complexas.

## Passo 1 – Criar o repositório no GitHub

1. Acesse [GitHub](#).
2. Clique em **New Repository**.
3. Dê o nome: ecommerce-db.
4. Adicione uma descrição: *Projeto de banco de dados para e-commerce com modelagem lógica, scripts SQL e consultas complexas.*
5. Marque como **Public**.

6. Clique em **Create Repository**.

 Passo 2 – Preparar os arquivos localmente

No seu computador, crie uma pasta chamada ecommerce-db e coloque dentro:

- README.md (documentação que já montamos)
- schema.sql (criação das tabelas)
- insert.sql (inserção de dados)
- queries.sql (consultas complexas)

 Passo 3 – Subir os arquivos para o GitHub

Opção A – Pelo site

1. Abra o repositório recém-criado.
2. Clique em **Upload files**.
3. Selecione os arquivos da pasta ecommerce-db.
4. Clique em **Commit changes**.
5. Opção B – Pelo Git no terminal
6. Se você já tem Git instalado:
7. # Clonar o repositório vazio
8. git clone <https://github.com/seusuuario/ecommerce-db.git>
9. # Entrar na pasta
10. cd ecommerce-db
11. # Copiar os arquivos para dentro da pasta
12. # (README.md, schema.sql, insert.sql, queries.sql)
13. # Adicionar os arquivos ao controle de versão
14. git add .
15. # Criar o commit
16. git commit -m "Projeto de banco de dados e-commerce"
17. # Enviar para o GitHub
18. git push origin main

 Passo 4 – Conferir

- Abra o repositório no GitHub.
- Verifique se todos os arquivos estão lá.
- O README.md será exibido automaticamente na página inicial do repositório.