

SQL BASICO

Para esta atividade utilizar a ferramenta SQL Server

1. Criar um **E-commerce** simples utilizando comandos SQL.
 - a. Utilizar comando CREATE DATABASE;
 - b. Utilizar comando CREATE TABLE;
 - c. Fazer relacionamento com FOREIGN KEY e REFERENCES;
 - d. Inserir dados com comando INSERT;
 - e. Fazer comando SELECT com clausula WHERE;
2. Criar um **Marketplace** simples utilizando comandos SQL.
 - a. Utilizar comando CREATE DATABASE;
 - b. Utilizar comando CREATE TABLE;
 - c. Fazer relacionamento com FOREIGN KEY e REFERENCES;
 - d. Inserir dados com comando INSERT;
 - e. Fazer comando SELECT com clausula WHERE;
3. Criar um **Classificado** simples utilizando comandos SQL.
 - a. Utilizar comando CREATE DATABASE;
 - b. Utilizar comando CREATE TABLE;
 - c. Fazer relacionamento com FOREIGN KEY e REFERENCES;
 - d. Inserir dados com comando INSERT;
 - e. Fazer comando SELECT com clausula WHERE;
4. Criar um **Rede Social** simples utilizando comandos SQL.
 - a. Utilizar comando CREATE DATABASE;
 - b. Utilizar comando CREATE TABLE;
 - c. Fazer relacionamento com FOREIGN KEY e REFERENCES;
 - d. Inserir dados com comando INSERT;
 - e. Fazer comando SELECT com clausula WHERE;

Salve os scripts de cada projeto em pastas diferentes e suba no GitHub, você pode separar os scripts de cada pasta em DDL e DML. Ex:

➔ Pasta Classificado:

- DDL.sql
- DML.sql

Entrega

Para esta entrega utilizar o Fluxo de entrega quando não existe repositório criado, segue os passos:

1. Criar repositório no GitHub;
2. Clonar repositório em uma pasta na maquina: `$ git clone [url]`;
3. Fazer alterações necessárias;
4. Preparar alterações para controle de versão: `$ git add .` ou `$ git add [file]`;
5. Gravar alterações de arquivos permanentemente: `$ git commit -m "[descriptive message]"`;
6. Carrega todos os commits de branch local para o GitHub: `$ git push [nome-repo] [branch]`.

Em seguida enviar o link do repositório na plataforma;