Guia de Configuração e Execução da Aplicação CLI de Banco de Dados

Grupo Minha Vida é um DeepFake

October 31, 2024

1 Introdução

Este documento fornece instruções detalhadas para configurar e executar uma aplicação de linha de comando (CLI) que realiza consultas SQL em um banco de dados PostgreSQL. Siga os passos cuidadosamente para preparar o ambiente e executar a aplicação com sucesso.

2 Estrutura do Projeto

Abaixo está uma explicação de cada arquivo Python presente no projeto, com seu respectivo propósito:

- **app.py**: Arquivo principal da aplicação CLI. Ele exibe o menu interativo no terminal, gerencia a interação com o usuário e chama as funções de consulta de acordo com as opções selecionadas.
- **database_config.py**: Arquivo de configuração da conexão com o banco de dados PostgreSQL. Ele contém a função `conectar_banco()`, que utiliza variáveis de ambiente para estabelecer uma conexão segura com o banco de dados.
- **queries.py**: Este arquivo contém funções para cada uma das consultas SQL usadas na aplicação. Cada função encapsula uma consulta específica e retorna os resultados para que o `app.py` possa exibi-los ao usuário.
- **utils.py**: Contém funções utilitárias para formatar e manipular os dados obtidos do banco de dados. Isso inclui a formatação de resultados em tabelas e o tratamento de dados complexos, como blobs de imagem, que são salvos como arquivos temporários para visualização.

3 Configuração do Ambiente

3.1 Instalação de Dependências

Para rodar este projeto, é necessário instalar a biblioteca `python-dotenv` para gerenciar variáveis de ambiente. Execute o seguinte comando no terminal para instalá-la:

```
pip install python-dotenv
```

3.2 Configuração do Arquivo `.env`

Crie um arquivo `.env` na pasta raiz do projeto para configurar as variáveis de conexão ao banco de dados. Insira as seguintes configurações no arquivo `.env`:

```
DB_NAME=nome_do_banco
DB_USER=usuario
DB_PASSWORD=senha
DB_HOST=localhost
DB_PORT=5432
```

3.3 Explicação dos Campos do `.env`

Para preencher corretamente as variáveis acima, é necessário acessar as configurações do seu banco de dados PostgreSQL:

- **DB_NAME**: Nome do banco de dados ao qual deseja conectar. Utilize um cliente SQL (como `pgAdmin` ou `psql`) para verificar o nome do banco ou crie um novo usando:

  ```
  CREATE DATABASE nome_do_banco;
  ```

- **DB_USER**: Usuário autorizado a acessar o banco. O usuário padrão do PostgreSQL é `postgres`, mas você pode criar um novo usuário com:

  ```
  CREATE USER usuario WITH PASSWORD 'senha';
  GRANT ALL PRIVILEGES ON DATABASE nome_do_banco TO usuario;
  ```

- **DB_PASSWORD**: Senha associada ao `DB_USER`. Defina ou altere a senha com:

  ```
  ALTER USER usuario WITH PASSWORD 'nova_senha';
  ```

- **DB_HOST**: Endereço do servidor do banco de dados. Use `localhost` para conexões locais. Se estiver em outro servidor, insira o IP ou domínio.

- **DB_PORT**: Porta onde o PostgreSQL está escutando (padrão: 5432). Verifique o arquivo `postgresql.conf` caso a porta seja personalizada.

4 Preparando o Projeto para Execução

4.1 Estrutura do Projeto

Certifique-se de que todos os arquivos estejam organizados conforme abaixo:

```
meu_projeto/
        app.py                # Arquivo principal da aplica  o CLI
        database_config.py    # Configura  o de conex o ao banco de dados
        queries.py            # Arquivo com as fun  es de consulta SQL
        utils.py              # Fun  es utilit rias para formata  o de resultados
        .env.example          # Exemplo das vari veis de ambiente
        requirements.txt      # Lista de depend ncias
```

4.2 Arquivo `.env.example`

Para facilitar a configuração, inclua um arquivo `.env.example` com placeholders para as variáveis de ambiente, como no exemplo:

```
DB_NAME=nome_do_banco
DB_USER=usuario
DB_PASSWORD=senha
DB_HOST=localhost
DB_PORT=5432
```

4.3 Criação do `requirements.txt`

O arquivo `requirements.txt` lista as dependências do projeto. Gere-o automaticamente com o comando:

```
pip freeze > requirements.txt
```

Ou, insira manualmente as dependências principais:

```
psycopg2
python-dotenv
pillow
```

5 Instruções de Uso (README)

Recomenda-se incluir um arquivo `README.md` com instruções detalhadas para configuração e execução do projeto. Um exemplo de conteúdo do `README.md`:

```
# Projeto de Banco de Dados CLI

Este projeto é uma aplicação de linha de comando que permite executar consultas SQL
em um banco de dados PostgreSQL.

## Pré-requisitos
- Python 3.7+
- PostgreSQL
- Dependências listadas em ‘requirements.txt‘

## Instalação
1. Clone o projeto ou extraia os arquivos em um diretório local.
2. Instale as dependências:
   ‘‘‘bash
   pip install -r requirements.txt

Configure as variáveis de ambiente:
Crie um arquivo .env baseado no arquivo .env.example.
Preencha as variáveis com as credenciais do banco de dados.
Estrutura do Projeto
app.py: Arquivo principal da aplicação CLI.
database_config.py: Configuração de conexão ao banco de dados.
queries.py: Funções para consultas SQL.
utils.py: Funções utilitárias para formatação de resultados.
.env.example: Exemplo das variáveis de ambiente.
requirements.txt: Lista de dependências do projeto.
Executando o Projeto
Certifique-se de que o banco de dados PostgreSQL está em execução e acessível.
Execute a aplicação com o comando:
bash
Copiar código
python app.py
```

6 Considerações Finais

Após configurar o ambiente e preencher as variáveis de conexão, a aplicação CLI estará pronta para uso. Certifique-se de que o banco de dados PostgreSQL está configurado corretamente e que as permissões de acesso estão adequadas para o usuário especificado.