Manipulação de Bancos de Dados

Manipulação de Bancos de Dados

- Conexão com bancos de dados.
- Consultas SQL.
- Diferenças entre bancos relacionais e não relacionais.
- Uso de bibliotecas como SQLite e SQLAlchemy no Python.

O Que São Bancos de Dados?

- **Um banco de dados é usado para armazenar e gerenciar informações.**
- Relacionais:
  - Organizam dados em tabelas.
  - Utilizam SQL (Structured Query Language).
  - Ex.: MySQL, PostgreSQL, SQLite.
- Não Relacionais:
  - Dados em documentos, grafos ou pares chave-valor.
  - Ex.: MongoDB, Firebase.

Conexão com Bancos de Dados Relacionais no Python

- SQLite:
- Banco relacional embutido no Python.
- Não exige configuração de servidor.

```
import sqlite3

# Criar ou conectar ao banco
conexao = sqlite3.connect('meu_banco.db')
cursor = conexao.cursor()

# Fechar conexão
cursor.close()
conexao.close()
```

Comandos SQL Básicos

Criar Tabelas:

```sql
CREATE TABLE usuarios (
    id INTEGER PRIMARY KEY,
    nome TEXT NOT NULL,
    email TEXT UNIQUE
);
```

Inserir Dados:

```
sql
Copiar código
INSERT INTO usuarios (nome, email) VALUES ('Ana', 'ana@email.com');
```

Consultar Dados:

```sql
sql
Copiar código
SELECT * FROM usuarios;
```

Atualizar Dados:

```sql
sql
Copiar código
UPDATE usuarios SET email = 'novo@email.com' WHERE id = 1;
```

Excluir Dados:

```sql
sql
Copiar código
DELETE FROM usuarios WHERE id = 1;
```

Exemplo Completo em Python

```python
import sqlite3

# Conectar ao banco
conexao = sqlite3.connect('meu_banco.db')
cursor = conexao.cursor()

# Criar tabela
cursor.execute('''
CREATE TABLE IF NOT EXISTS usuarios (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nome TEXT NOT NULL,
    email TEXT UNIQUE NOT NULL
);
''')

# Inserir dados
cursor.execute('INSERT INTO usuarios (nome, email) VALUES (?, ?)', ('João', 'joao@email.c
om'))
conexao.commit()

# Consultar dados
cursor.execute('SELECT * FROM usuarios')
for linha in cursor.fetchall():
    print(linha)

# Fechar conexão
cursor.close()
conexao.close()
```

Uso de SQLAlchemy (ORM - Object Relational Mapper)

- Permite manipular bancos de dados com objetos Python.
- Instalação:

```
bash

Copiar código

pip install sqlalchemy
```

Exemplo de Modelo:

```python
from sqlalchemy import create_engine, Column, Integer, String
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

Base = declarative_base()

class Usuario(Base):
    __tablename__ = 'usuarios'
    id = Column(Integer, primary_key=True)
    nome = Column(String, nullable=False)
    email = Column(String, unique=True, nullable=False)

# Configuração do banco
engine = create_engine('sqlite:///meu_banco.db')
Base.metadata.create_all(engine)
Session = sessionmaker(bind=engine)
session = Session()

# Inserir usuário
novo_usuario = Usuario(nome="Ana", email="ana@email.com")
session.add(novo_usuario)
session.commit()

# Consultar usuários
usuarios = session.query(Usuario).all()
for usuario in usuarios:
    print(usuario.nome, usuario.email)
```

Exercícios Durante a Aula

Exercício 1: Conexão com Banco SQLite

Assistido

Enunciado: Crie um script Python que se conecte a um banco SQLite chamado `meu_banco.db` e exiba uma mensagem confirmando a conexão.

Exercício 2: Criar e Inserir Dados em uma Tabela

Enunciado: Crie uma tabela `produtos` com colunas `id`, `nome` e `preco`. Insira três produtos e exiba os dados inseridos.

```python
import sqlite3

conexao = sqlite3.connect('meu_banco.db')
cursor = conexao.cursor()

cursor.execute('''
CREATE TABLE IF NOT EXISTS produtos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nome TEXT NOT NULL,
    preco REAL NOT NULL
);
''')

produtos = [('Notebook', 2500.0), ('Celular', 1500.0), ('Câmera', 1200.0)]
cursor.executemany('INSERT INTO produtos (nome, preco) VALUES (?, ?)', produtos)
conexao.commit()

cursor.execute('SELECT * FROM produtos')
for produto in cursor.fetchall():
    print(produto)

cursor.close()
conexao.close()
```

Exercício 3: Atualizar e Excluir Dados

- **Enunciado**: Atualize o preço do produto `Notebook` para 2700.0 e exclua o produto `Câmera`.

```
Copiar código
cursor.execute("UPDATE produtos SET preco = ? WHERE nome = ?", (2700.0, 'Notebook'))
cursor.execute("DELETE FROM produtos WHERE nome = ?", ('Câmera',))
conexao.commit()
```

Exercício 4: Consultar Dados

Enunciado: Escreva um script que consulte todos os produtos no banco de dados e exiba apenas os com preço maior que 2000.

```python
cursor.execute("SELECT * FROM produtos WHERE preco > ?", (2000,))
for produto in cursor.fetchall():
    print(produto)
```

Exercício 1: Uso de SQLAlchemy

Enunciado: Crie um modelo para a tabela `usuarios` usando SQLAlchemy. Adicione dois usuários e exiba seus nomes e e-mails.

Exercício 2: Relacionamento entre Tabelas

Enunciado: Crie uma tabela `pedidos` que tenha um relacionamento com a tabela `usuarios`, onde cada pedido pertence a um usuário.

Exercício 3: Criar Banco Não Relacional com MongoDB

Enunciado: Use a biblioteca `pymongo` para criar um banco MongoDB. Insira um documento na coleção `produtos`.