

第十六周作业：Neo4j 知识图谱实践

作业内容

安装 Neo4j 并实践知识图谱的构建和查询操作。

Neo4j 安装指南

方法一：使用 Docker 安装（推荐）

1. 安装 Docker Desktop

- 访问 [Docker Desktop 官网](#)
- 下载并安装 Docker Desktop for Windows

2. 使用 Docker 运行 Neo4j

```
docker run \
  --name neo4j \
  -p7474:7474 -p7687:7687 \
  -e NEO4J_AUTH=neo4j/password \
  -d neo4j:latest
```

3. 访问 Neo4j Browser

- 打开浏览器访问：<http://localhost:7474>
- 用户名：neo4j
- 密码：password（首次登录后需要修改）

方法二：直接安装 Neo4j Desktop

1. 下载 Neo4j Desktop

- 访问 [Neo4j 官网](#)
- 下载 Neo4j Desktop（Windows版本）

2. 安装步骤

- 运行安装程序
- 创建账户或使用本地账户
- 创建新的数据库项目
- 启动数据库

3. 配置连接

- 默认端口：7474 (HTTP), 7687 (Bolt)
- 默认用户名：neo4j
- 首次登录需要设置新密码

Python 环境配置

安装依赖

```
pip install -r requirements.txt
```

主要依赖包：

- `neo4j`：Neo4j Python 驱动
- `pandas`：数据处理（可选）

快速开始

1. 连接 Neo4j 数据库

```
from neo4j_connection import Neo4jConnection

# 创建连接
conn = Neo4jConnection(
    uri="bolt://localhost:7687",
    user="neo4j",
    password="your_password"
)

# 测试连接
conn.test_connection()
```

2. 创建节点和关系

运行 `basic_operations.py` 查看基本操作示例。

3. 构建知识图谱

运行 `knowledge_graph_example.py` 查看完整的知识图谱构建示例。

示例代码说明

`neo4j_connection.py`

- Neo4j 数据库连接类
- 提供连接管理和基本操作封装

`basic_operations.py`

- 创建节点 (Node)
- 创建关系 (Relationship)
- 基本查询操作
- 更新和删除操作

knowledge_graph_example.py

- 构建电影知识图谱示例
- 包含演员、导演、电影等实体
- 复杂查询示例 (Cypher 查询)

Cypher 查询语言基础

创建节点

```
CREATE (p:Person {name: 'Alice', age: 30})
```

创建关系

```
MATCH (a:Person {name: 'Alice'}), (b:Person {name: 'Bob'})  
CREATE (a)-[:KNOWS]->(b)
```

查询节点

```
MATCH (p:Person)  
RETURN p
```

查询关系

```
MATCH (a:Person)-[r:KNOWS]->(b:Person)  
RETURN a, r, b
```