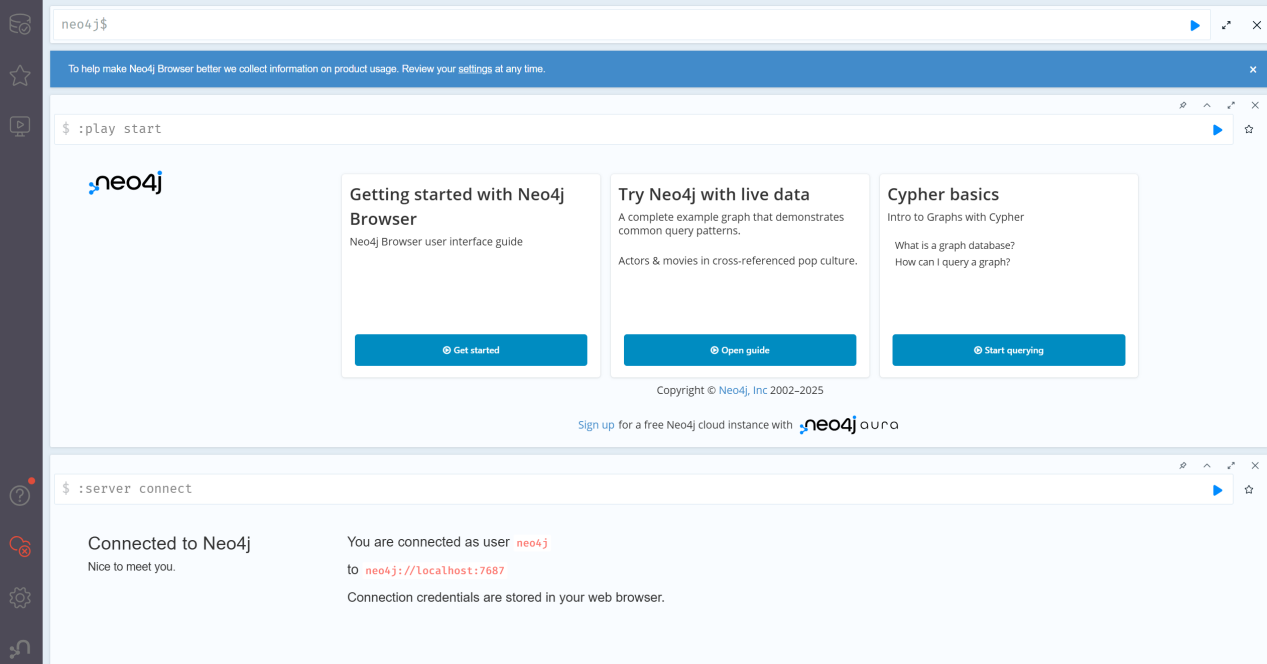
# 实验3实验报告

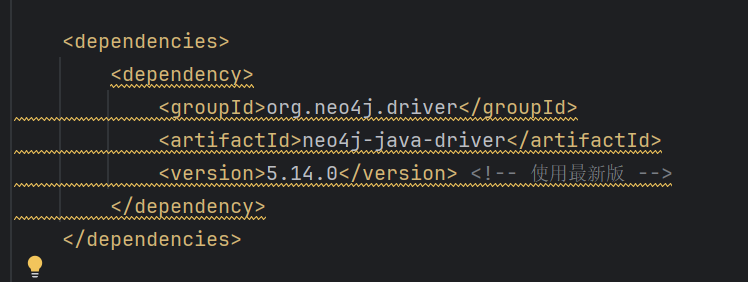
### 姓名：谢宝玛 学号：1120233506

## 安装neo4j,启动



## 新建Java项目

注：尝试添加jar包，但是报错，所以使用maven添加依赖



## 编写程序，创建知识图谱

构建一个“学术领域知识图谱”，包含以下实体和关系：

节点类型（实体）：

Person：研究人员（属性：name, affiliation）

Paper：论文（属性：title, year）

Conference：会议（属性：name, location）

关系类型：

AUTHORED：(Person)-[:AUTHORED]->(Paper)

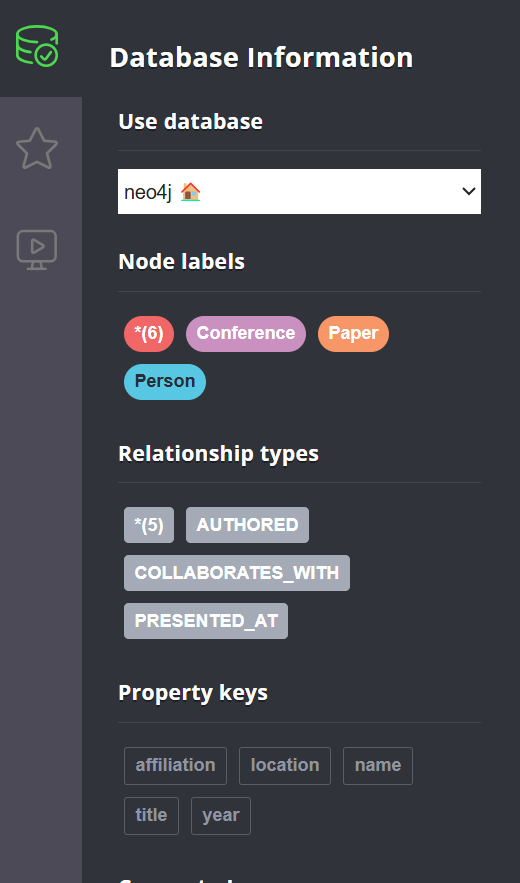
PRESENTED\_AT：(Paper)-[:PRESENTED\_AT]->(Conference)

COLLABORATES\_WITH：(Person)-[:COLLABORATES\_WITH]->(Person)

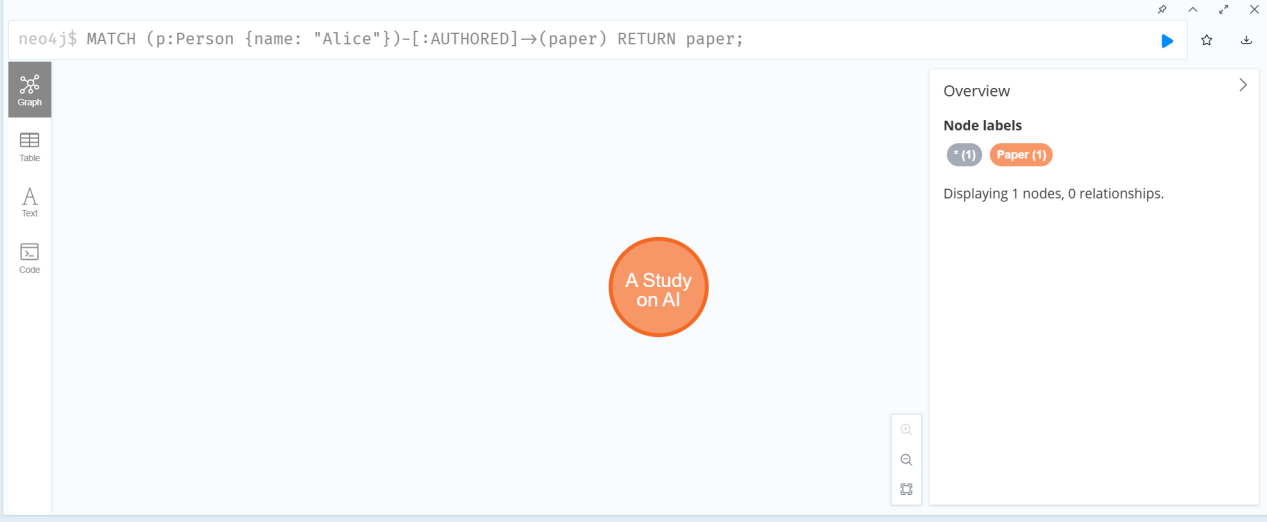
源码：

package com.alibaba.cloud;  
  
import org.neo4j.driver.\*;  
  
public class AcademicGraph implements AutoCloseable {  
 private final Driver driver;  
  
 public AcademicGraph(String uri, String user, String password) {  
 driver = GraphDatabase.driver(uri, AuthTokens.basic(user, password));  
 }  
  
 @Override  
 public void close() {  
 driver.close();  
 }  
  
 // 创建研究人员  
 public void createPerson(String name, String affiliation) {  
 try (Session session = driver.session()) {  
 session.run("MERGE (p:Person {name: $name, affiliation: $affiliation})",  
 Values.parameters("name", name, "affiliation", affiliation));  
 }  
 }  
  
 // 创建论文  
 public void createPaper(String title, int year) {  
 try (Session session = driver.session()) {  
 session.run("MERGE (p:Paper {title: $title, year: $year})",  
 Values.parameters("title", title, "year", year));  
 }  
 }  
  
 // 创建会议  
 public void createConference(String name, String location) {  
 try (Session session = driver.session()) {  
 session.run("MERGE (c:Conference {name: $name, location: $location})",  
 Values.parameters("name", name, "location", location));  
 }  
 }  
  
 // 创建作者与论文的关系  
 public void createAuthorship(String personName, String paperTitle) {  
 try (Session session = driver.session()) {  
 session.run(  
 "MATCH (p:Person {name: $personName}), (pa:Paper {title: $paperTitle}) " +  
 "MERGE (p)-[:AUTHORED]->(pa)",  
 Values.parameters("personName", personName, "paperTitle", paperTitle));  
 }  
 }  
  
 // 创建论文和会议的关系  
 public void createPresentation(String paperTitle, String confName) {  
 try (Session session = driver.session()) {  
 session.run(  
 "MATCH (pa:Paper {title: $paperTitle}), (c:Conference {name: $confName}) " +  
 "MERGE (pa)-[:PRESENTED\_AT]->(c)",  
 Values.parameters("paperTitle", paperTitle, "confName", confName));  
 }  
 }  
  
 // 创建合作关系  
 public void createCollaboration(String name1, String name2) {  
 try (Session session = driver.session()) {  
 session.run(  
 "MATCH (a:Person {name: $name1}), (b:Person {name: $name2}) " +  
 "MERGE (a)-[:COLLABORATES\_WITH]->(b)",  
 Values.parameters("name1", name1, "name2", name2));  
 }  
 }  
  
 // 示例数据写入  
 public static void main(String[] args) {  
 try (AcademicGraph graph = new AcademicGraph("bolt://localhost:7687", "neo4j", "13883880881xs")) {  
 graph.createPerson("Alice", "Tsinghua University");  
 graph.createPerson("Bob", "MIT");  
  
 graph.createPaper("A Study on AI", 2023);  
 graph.createPaper("Graph Databases", 2022);  
  
 graph.createConference("ICML", "Vienna");  
 graph.createConference("NeurIPS", "New Orleans");  
  
 graph.createAuthorship("Alice", "A Study on AI");  
 graph.createAuthorship("Bob", "Graph Databases");  
  
 graph.createPresentation("A Study on AI", "ICML");  
 graph.createPresentation("Graph Databases", "NeurIPS");  
  
 graph.createCollaboration("Alice", "Bob");  
 }  
 }  
}

## 查看数据库



查询 Alice 写过哪些论文：



查看哪些人合作过

