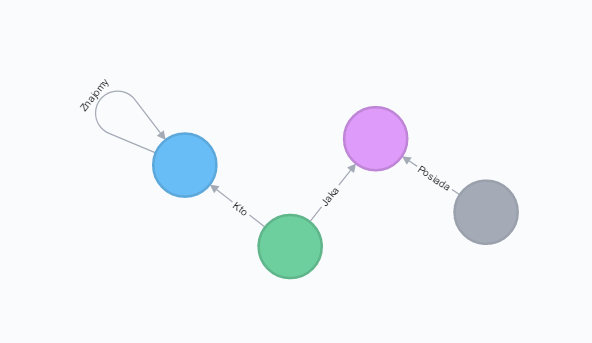
Wiktor Łęczyński

Zadanie - grafowe bazy danych

1)



Schemat bazy danych

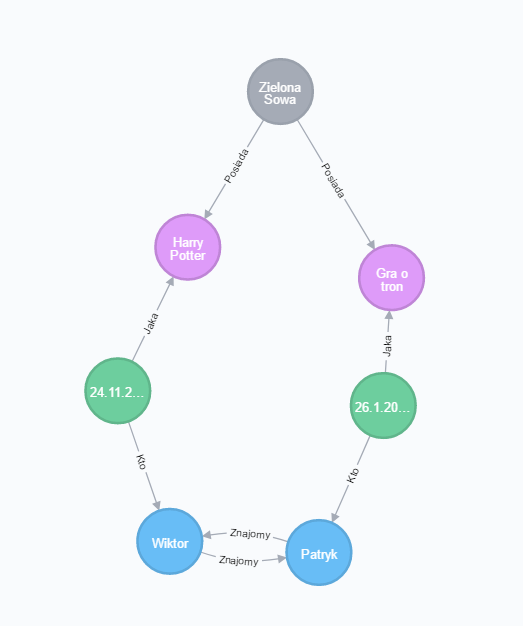
Opis schematu

-Czytelnik (niebieski) atrybuty :imie , nazwisko , miejscowość , może posiadać wskażnik do znajomych czytelników

-Książka (fiolet) atrybuty:tytuł , autor

-Wypożyczenie(zielony) atrybuty : id , data ,status, wskażnik do czytelnika i książki

-Wydawnictwo (szary) atrybuty : nazwa , wskażnik do książki

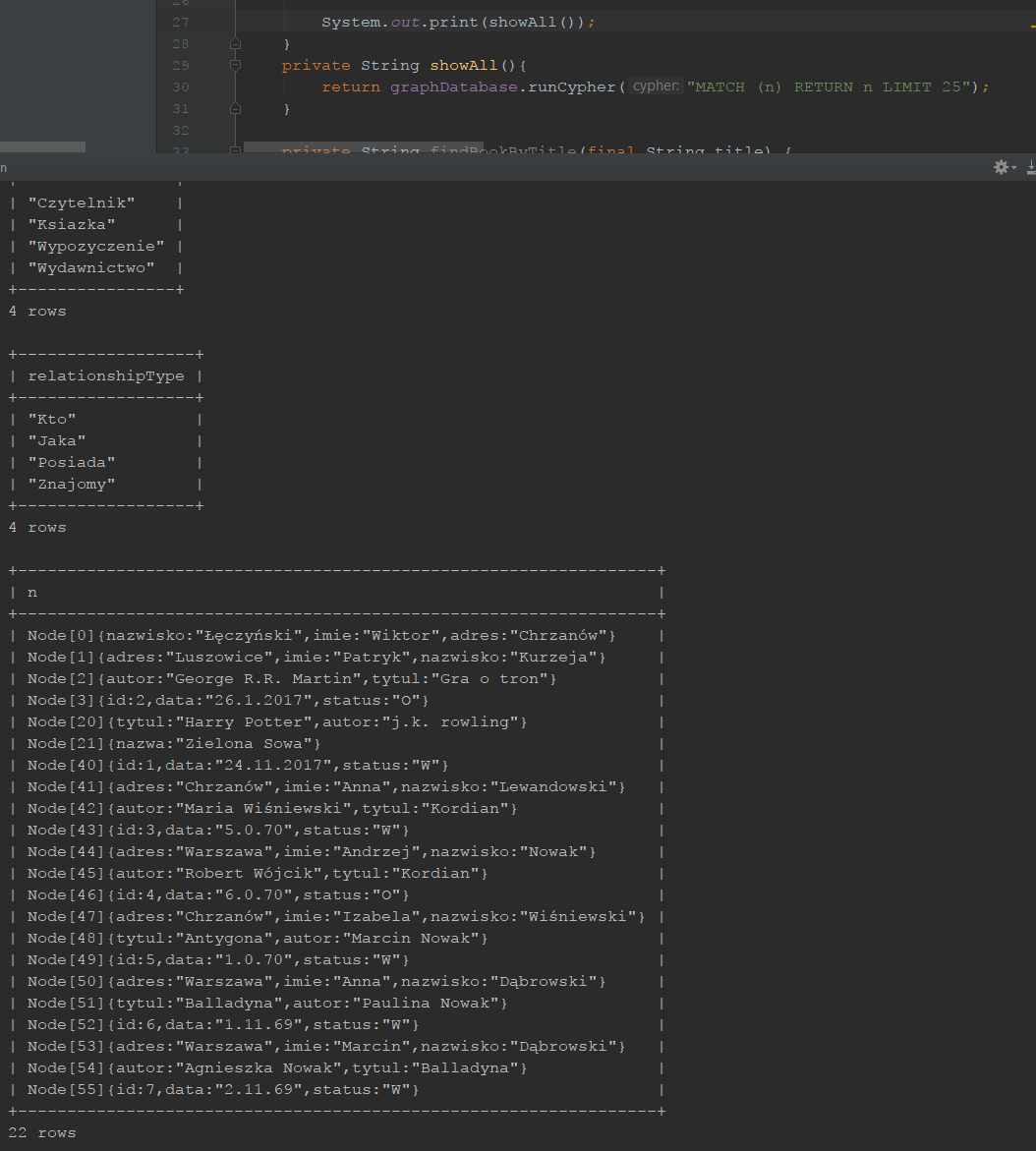


2)

CREATE (czytelnik:Czytelnik {imie:"Wiktor",nazwisko:"Łęczyński",adres:"Chrzanów"}) RETURN czytelnik  
 CREATE (czytelnik:Czytelnik {imie:"Patryk",nazwisko:"Kurzeja",adres:"Luszowice"}) RETURN czytelnik  
 CREATE (ksiazka:Ksiazka {tytul:"Harry Potter",autor:"j.k. rowling"}) RETURN ksiazka  
 CREATE (ksiazka:Ksiazka {tytul:"Gra o tron",autor:"George R.R. Martin"}) RETURN ksiazka  
  
 Create (wypozyczenie:Wypozyczenie {id:1,data:"24.11.2017",status:"W"}) Return wypozyczenie  
 Create (wypozyczenie:Wypozyczenie {id:2,data:"26.1.2017",status:"O"}) Return wypozyczenie  
  
 Create (wydawnictwo:Wydawnictwo {nazwa:"Zielona Sowa"}) Return wydawnictwo  
  
 Match (czytelnik:Czytelnik),(wypozyczenie:Wypozyczenie)  
 where czytelnik.imie="Wiktor" and czytelnik.nazwisko="Łęczyński" and  
 wypozyczenie.id=1  
 Create (wypozyczenie)-[r:Kto]->(czytelnik)  
 return wypozyczenie  
  
 Match (ksiazka:Ksiazka),(wypozycznie:Wypozyczenie)  
 where ksiazka.tytul="Harry Potter" and wypozycznie.id=1  
 Create (wypozycznie)-[r:Jaka]->(ksiazka)  
 return wypozycznie  
  
  
 Match (czytelnik:Czytelnik),(wypozyczenie:Wypozyczenie)  
 where czytelnik.imie="Patryk" and czytelnik.nazwisko="Kurzeja"  
 and wypozyczenie.id=2  
 Create (wypozyczenie)-[r:Kto]->(czytelnik)  
 return wypozyczenie  
  
 Match (ksiazka:Ksiazka),(wypozycznie:Wypozyczenie)  
 where ksiazka.tytul="Gra o tron" and wypozycznie.id=2  
 Create (wypozycznie)-[r:Jaka]->(ksiazka)  
 return wypozycznie  
  
  
 Match(wydawnictwo:Wydawnictwo),(ksiazka:Ksiazka) where  
 ksiazka.tytul="Harry Potter" and  
 wydawnictwo.nazwa="Zielona Sowa"  
 Create(wydawnictwo)-[r:Posiada]->(ksiazka)  
 return wydawnictwo  
  
 Match(wydawnictwo:Wydawnictwo),(ksiazka:Ksiazka) where  
 ksiazka.tytul="Gra o tron" and  
 wydawnictwo.nazwa="Zielona Sowa"  
 Create(wydawnictwo)-[r:Posiada]->(ksiazka)  
 return wydawnictwo  
  
 Match (czytelnik:Czytelnik),(czytelnik2:Czytelnik)  
 where czytelnik.imie="Patryk" and czytelnik.nazwisko="Kurzeja"  
 and czytelnik2.imie="Wiktor"  
 and czytelnik2.nazwisko="Łęczyński"  
 Create (czytelnik)-[r:Znajomy]->(czytelnik2)  
 return czytelnik  
  
 Match (czytelnik:Czytelnik),(czytelnik2:Czytelnik)  
 where czytelnik.imie="Patryk" and czytelnik.nazwisko="Kurzeja"  
 and czytelnik2.imie="Wiktor"  
 and czytelnik2.nazwisko="Łęczyński"  
 Create (czytelnik2)-[r:Znajomy]->(czytelnik)  
 return czytelnik  
   
 Match(wypozyczenia:Wypozyczenie{id:1})-[r]->(b)  
 return r,wypozyczenia,b

3) private void generator(){  
 String[] imiona = {"Marcin", "Maria", "Anna", "Małgorzata", "Andrzej", "Agnieszka", "Paulina", "Jagoda", "Izabela", "Robert"};  
 String[] nazwiska={"Nowak","Nowak","Wiśniewski","Dąbrowski","Lewandowski","Wójcik"};  
 String[] tytuly={"Antygona","Balladyna","Wesele","Król Edyp","Kordian","Hamlet"};  
 String[] miasta={"Chrzanów","Kraków","Warszawa"};  
 Random generator = new Random();  
 for(int i=0;i<5;i++) {  
 String imie = imiona[generator.nextInt(imiona.length)];  
 String nazwisko = nazwiska[generator.nextInt(nazwiska.length)];  
 String miasto = miasta[generator.nextInt(miasta.length)];  
//System.out.println("Tworze czytelnika"+imie+nazwisko);  
 graphDatabase.runCypher("CREATE (czytelnik:Czytelnik {imie:" + "'" + imie + "'" + "," +  
 "nazwisko:" + "'" + nazwisko + "'" +  
 ",adres:" + "'" + miasto + "'" + "})");  
  
 String tytul = tytuly[generator.nextInt(tytuly.length)];  
 String imie1 = imiona[generator.nextInt(imiona.length)];  
 String nazwisko1 = nazwiska[generator.nextInt(nazwiska.length)];  
//System.out.println("Tworze ksiazke "+tytul);  
 graphDatabase.runCypher("CREATE (ksiazka:Ksiazka {tytul:" + "'" + tytul + "'" + ",autor:" + "'" + imie1 + " " + nazwisko1 + "'" + "})");  
  
 Date date=new Date(generator.nextInt());  
 String[] statusy={"O","W"};  
 String status=statusy[generator.nextInt(statusy.length)];  
 String sdata=date.getDay()+"."+date.getMonth()+"."+date.getYear();  
  
 String s=getID();  
 String s1=s.substring(65,75);  
 String[] parts=s1.split(" ");  
 Integer id=Integer.*parseInt*(parts[0]);  
 id++;  
  
 graphDatabase.runCypher("Create (wypozyczenie:Wypozyczenie {id:"+id+",data:"+ "'" +sdata+ "'" +",status:"+ "'" +status+ "'" +"})");  
  
 graphDatabase.runCypher("Match (czytelnik:Czytelnik),(wypozyczenie:Wypozyczenie) where czytelnik.imie="+ "'" +imie+ "'" +" and czytelnik.nazwisko="+ "'" +nazwisko+ "'" +" and wypozyczenie.id="+id+" Create (wypozyczenie)-[r:Kto]->(czytelnik)");  
 graphDatabase.runCypher("Match (ksiazka:Ksiazka),(wypozycznie:Wypozyczenie)where ksiazka.tytul="+ "'" +tytul+ "'" +" and wypozycznie.id="+id+" Create (wypozycznie)-[r:Jaka]->(ksiazka)");  
 }  
 }

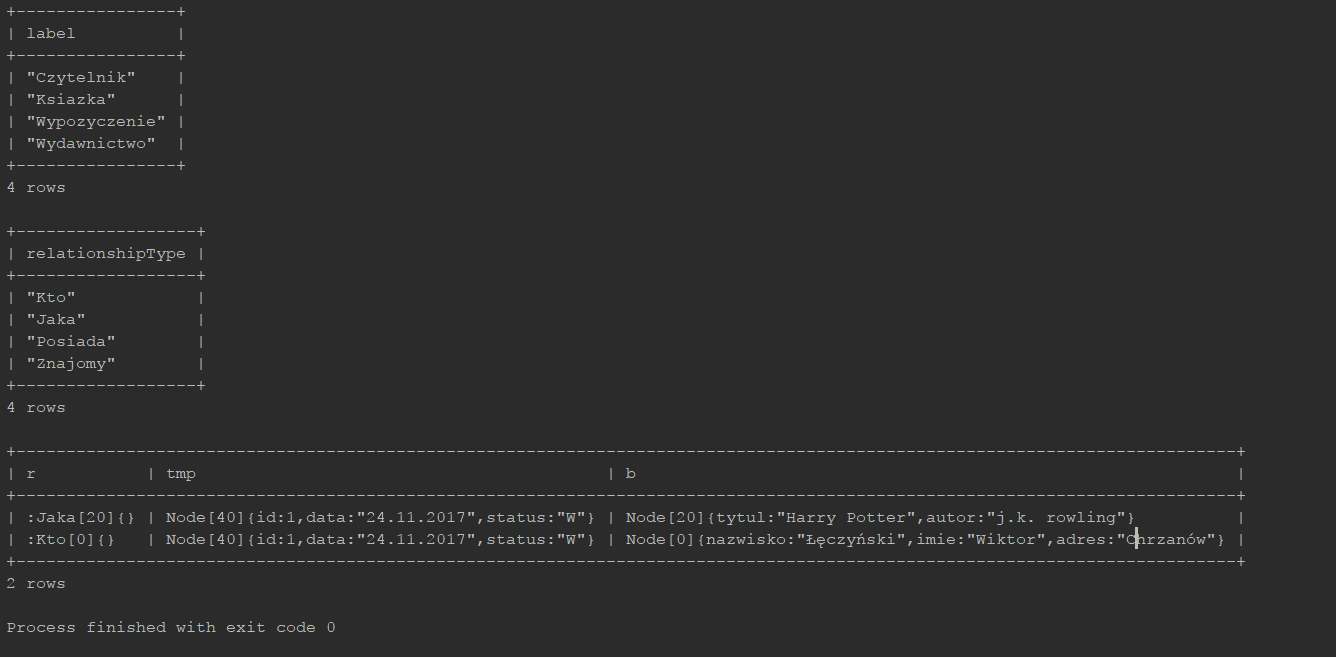
private String getID(){  
 return graphDatabase.runCypher("Match(wypozyczenia:Wypozyczenie) return wypozyczenia.id order by wypozyczenia.id DESC limit 1");  
}



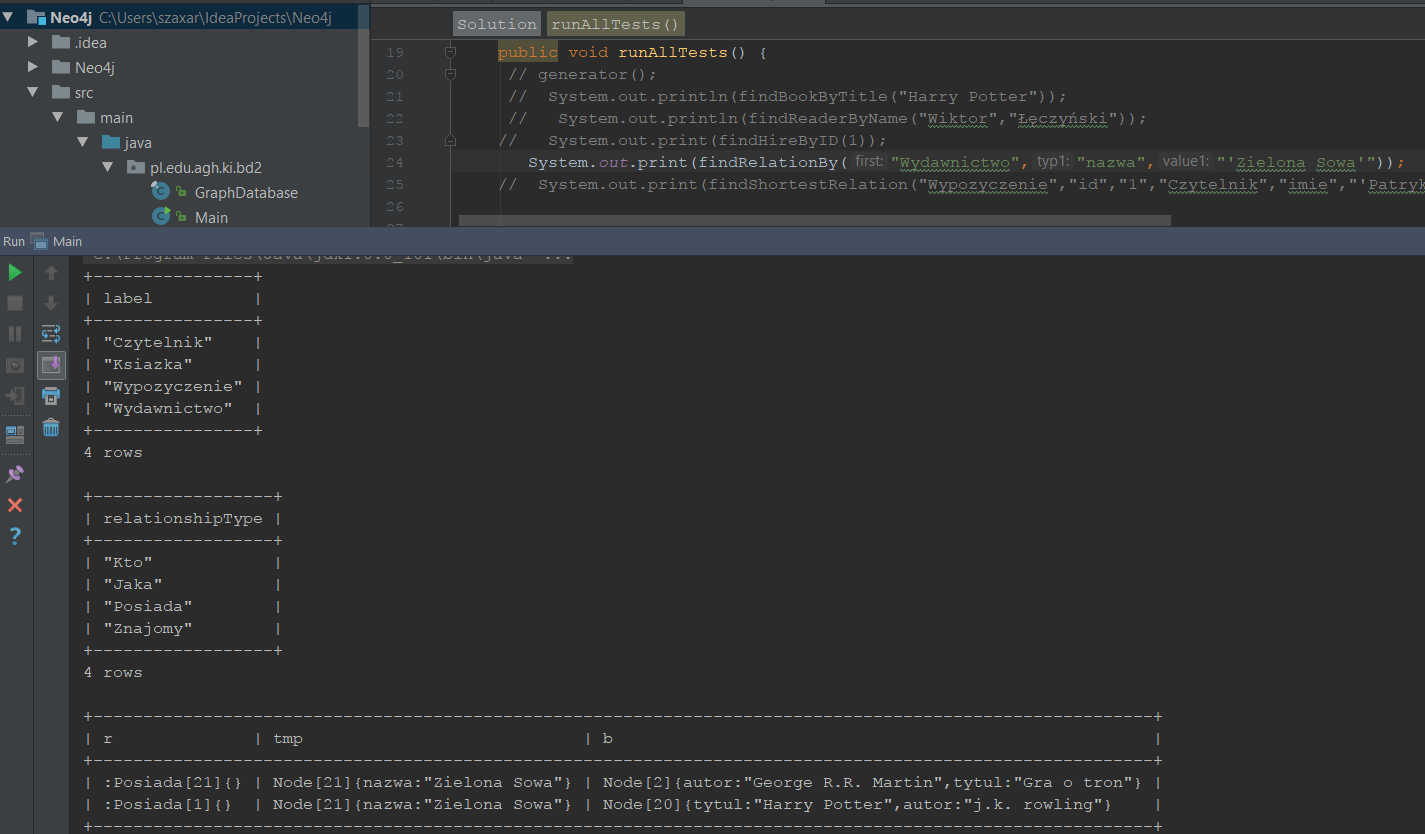
4) private String findRelationBy(final String first,final String typ1,final String value1){  
  
 return graphDatabase.runCypher("Match(tmp:"+first+"{"+typ1+":"+value1+"})-[r]->(b)\n" +  
 "return r,tmp,b");  
}

Test

System.*out*.print(findRelationBy("Wypozyczenie","id","1"));



System.*out*.print(findRelationBy("Wydawnictwo","nazwa","'Zielona Sowa'"));

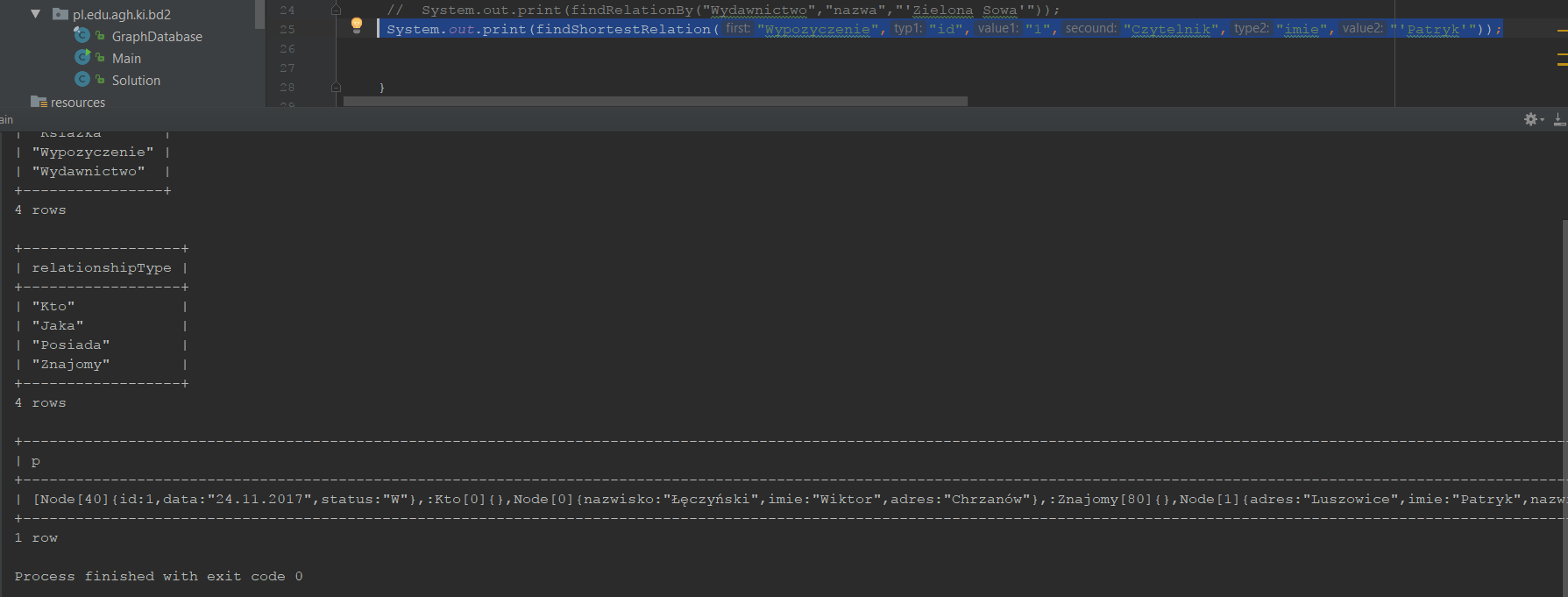


5)

private String findShortestRelation(final String first,final String typ1,final String value1,final String secound , final String type2 ,  
 final String value2 ){  
 return graphDatabase.runCypher("Match p=shortestPath((n1:"+first+"{"+typ1+":"+value1+"})-[\*..6]->(n2:"+secound  
 +"{"+type2+":"+value2+"}))\n" +  
 "return p");  
  
  
}

Test

System.*out*.print(findShortestRelation("Wypozyczenie","id","1","Czytelnik","imie","'Patryk'"));



System.out.print(findShortestRelation("Wypozyczenie","id","2","Czytelnik","imie","'Wiktor'"));

