

# V 6

---

1. postoji biblioteka sa predavanja koja je vazna za projekat

2. Cypher jezik - sintaksa

- **Čvorovi:**

(**p:Person**)

(**p**)

(**:Person**)

(**p:Person:Professor**)

3.

- **p** - alias
- **Person** - tip cvora
- moze se izostaviti tip
- moze se izostaviti alias, ali :
- cvor moze imati vise tipova

- **Grane:**

[**:FRIENDS**]

[**f**]

[**f:FRIENDS**]

4.

- u ovom slucaju je sличno, razlika je sto moze biti samo jedan tip

- Osobine:

```
{username:"pera",  
password:"perica"}
```

```
(:User{username:"pera",  
password:"perica"})
```

```
[:FRIENDS  
{koliko_dugo:10}]
```

5.

- i grane i cvorovi imaju osobine
- prvi primer ispod forme sintakse
  - prikazuje označavanje osobine cvora
- drugi primer prikazuje osobinu grane

- Povezivanje čvorova:

```
(u:User{ime:"Miša"})
```

```
[:FRIEND]->
```

```
(p:User{ime:"Saša"})
```

6.

7. Naredba Create

- **Create:**

```
CREATE (u:User  
{username:"Danijela",  
vocation: "professor"})
```

8. Upiti operatorom Match

- Match:

```
MATCH (movie:Movie)
RETURN movie.title
MATCH (movie:Movie
{title:"Cloud Atlas"})
RETURN movie
```

- - prvo je vracanje svih naziva filmova
  - drugo je vracanje filma sa odredjenom osobinom naziv

9. notacije Grane :

- `--` - postoji veza, ali ne i koji tip veze
- `<-- / -->` - isto, samo usmereno
- `<- / ->` - korisceno samo kada se govori o osobini veza
- `[r*1..4]` - postavljanje uslova kroz koliko koraka je dozvoljeno proci od jednog do drugog cvora

10. primer kreiranja grane

- Create:

```
MATCH
```

```
(a:User{username: "Danijela"}),(b:User{username: 'Nikola'})
CREATE (a)-[r:je_prijatelj { kolikoDugo: 5 }]->(b)
```

- Merge:

```
MERGE (u:User{age:61, name: "Nikola"})
```

11.

- Radi update ili ubacivanje neceg novog

12. set naredba

- Set:

```
MATCH (n:{name: 'Chris' }) SET n.username = 'chrisdkemper'
RETURN n
```

- - ubacivanje nove osobine

- Delete/Remove:

```
MATCH (p:Person) DELETE p
```

```
MATCH (n:User)-[p:je_prijatelj]->(u)DELETE p
```

```
MATCH (u:User) REMOVE u.username
```

13.

- moguce je bristi cvorove, grane, atribute

14. WHERE u kombinaciji sa MATCH

- Where:

```
MATCH (n:Person) WHERE n.age > 18 AND (n.name = 'Chris'  
OR n.name = "Kane") AND (n)- [:RELATED {relation:  
"brother"}]-() RETURN n
```

- 

15. ORDER BY

- Match:

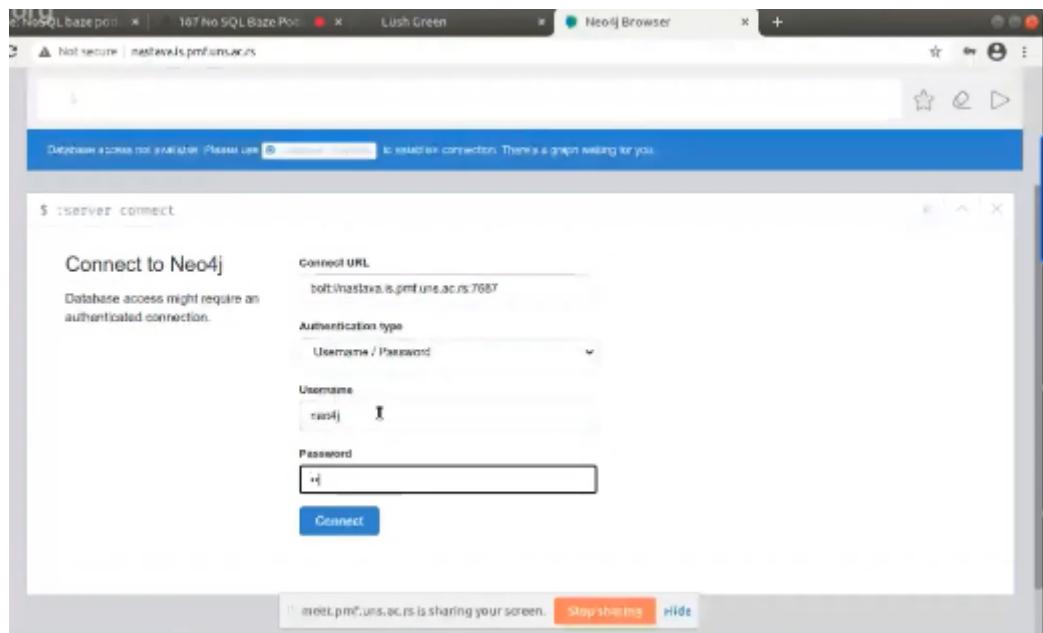
```
MATCH (n) RETURN n ORDER BY n.age, n.name DESC
```

- 

16. VPN nakacivanje

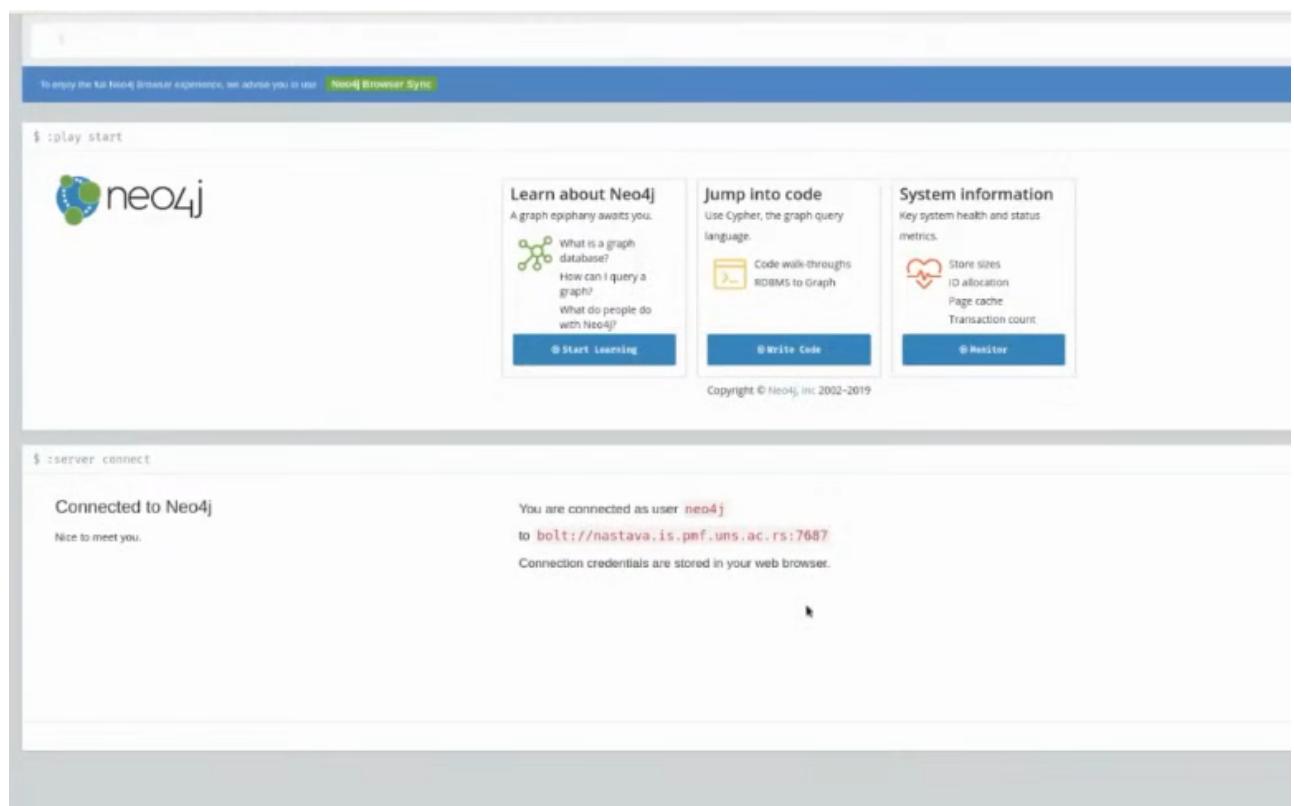
- Adresa: nastava.is.pmf.uns.ac.rs:7474
- Username: neo4j
- Password: admin

17.



18.

19. cuvanje svog export-a kao backup, u slucaju da ako se desi kolaps



20.



21.



22.



23.

24.

The screenshot shows the Neo4j Database Information page. At the top, there's a title bar with the URL "neo4jsecure | https://neo4j:7474". Below it is a sidebar with a star icon and a folder icon. The main area is titled "Database Information" and has a section titled "Node Labels". It displays a grid of node labels: "Asset", "Author", "Book", "BookAuthor", "Cinema", "City", "Country", "Drug", "Film", "Gamer", "Game", "Genre", "Movie", "Nation", "Person", "Player", "Pub", "Publisher", "Review", "Role", "Title", "User", "VideoGame", and "Worker".

- info DBa
  - nazivi cvorova DBa (u ovom slučaju su oni pobrisani, ali su ostali kao zapamceni ovde nepotrebno)
    - nemaju u sebi podatke
    - nije uradila complete reset

25.

The screenshot shows the Neo4j browser interface. On the left, there's a sidebar with a "Graph" icon, a search bar, and sections for "Node Labels" and "Relationship Types". The "Node Labels" section lists: "Asset", "Author", "Book", "BookAuthor", "Cinema", "City", "Country", "Drug", "Film", "Gamer", "Game", "Genre", "Movie", "Nation", "Person", "Player", "Pub", "Publisher", "Review", "Role", "Title", "User", "VideoGame", and "Worker". The "Relationship Types" section lists: "ACTED\_IN", "CONNECTED\_TO", "DIRECTED", "FOLLOWS", "PRODUCED", "REVIEWED", "USED\_BY", and "WRITES". In the center, there's a graph visualization with nodes represented by colored circles (blue, orange, red) and relationships represented by lines connecting them. A query at the top reads: "\$ MATCH p=()-[r:WRITES]->() RETURN p LIMIT 25". Below the graph, it says "Displaying 16 nodes, 22 relationships."

- nazivi grana
  - npr. ovde nisu obrisani podaci

26. i ostalo

The screenshot shows the Neo4j browser interface. The top section, titled 'Property Keys', lists variousNeo4j properties in a grid:

- id, accesscomplexity, age
- avg\_rating, bookID, born
- club, community, created
- date, description, distribution
- filmID, gender, id, me
- isOld, isbn, isac13, isold
- language, modified, name
- nationality, naziv, nazvit
- naziv1, naziv2, nickname
- nickname, num\_pages
- pagesize
- pagerank\_with\_direction, period
- position, rating, ratings\_count
- released, role, roles
- summary, tagline
- total\_reviews\_count, title, title
- type, version, vulnerability\_id
- votesCount

The bottom section, titled 'Connected as', shows the current user connection details:

- Username: neo4j
- Role: admin
- Admin:  server user list  
 server user add

The 'Database' section displays the database configuration:

- Version: 3.5.13
- Editor: Community
- Name: graph.db
- Size: 36.15 MB
- Implementation:  cypher  
 queries

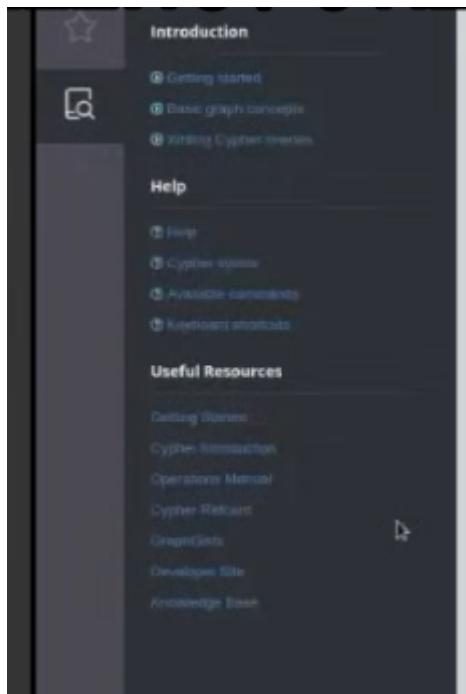
- property keys
- connection diag.
- db diag.

The screenshot shows the 'Sample Scripts' section of the Neo4j browser. It contains a list of pre-written Cypher scripts categorized by type:

- Basic Queries:**
  - Create an index
  - Create unique property constraint
  - Get some data: [▶](#)
  - Hello World! [▶](#)
- Common Procedures:**
- Data Profiling:**
- Example Graphs:**
  - Movie Graph [▶](#)
  - Northwind Graph [▶](#)

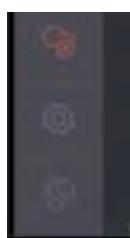
27.

- neki pomocni skriptovi
- bavicemo se sa **Movie Graph**



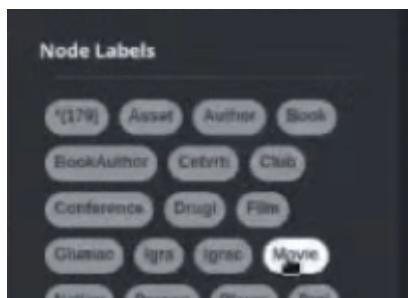
28.

- sekcija od dokumentacija

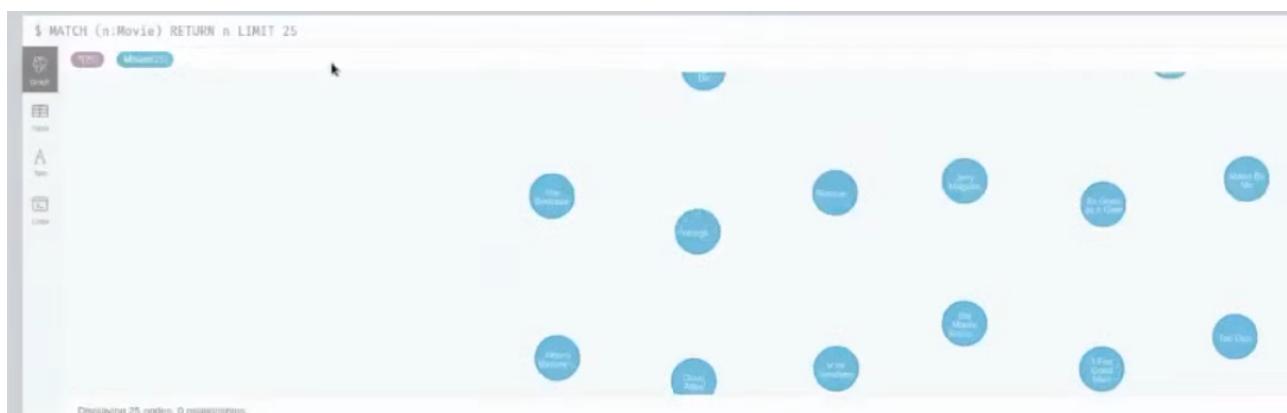


29.

- call service, settings, about



30.



31.

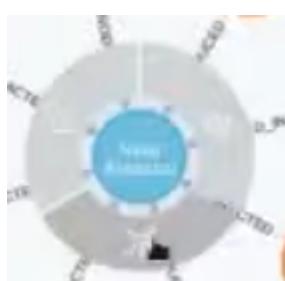


- klikom na izvrsen upit kopira se na traku za unos
  - isto tu je collapse card, close card opcija
- moguce je izmeniti sastav te naredbe
- klik na play



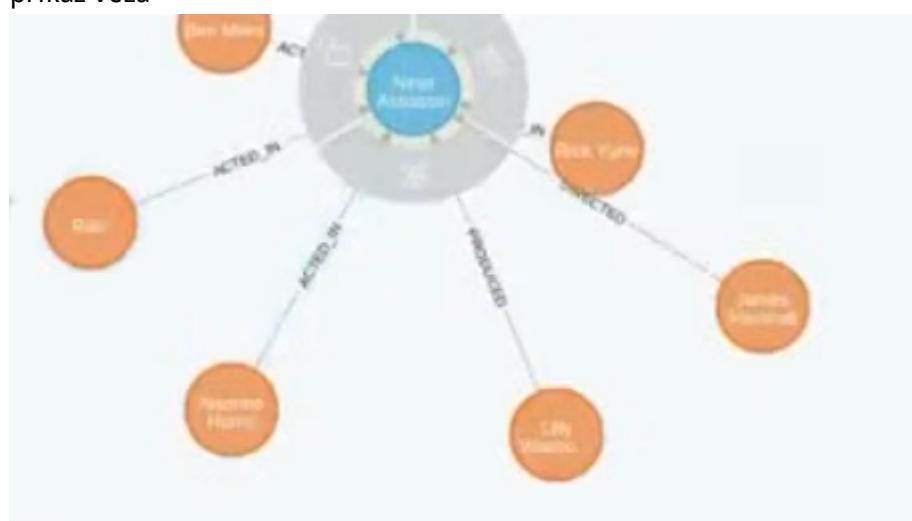
33.

- 
- atributi cvora



34.

- prikaz veza



- 



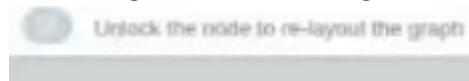
35.

- takodje za neki vezani cvor je moguce sagledati osobine



36.

- lock / unlock
- ovo i ne igra neku bitnu ulogu



37.

- Dismiss - isto ne igra neku ulogu



38.

- tabelarni prikaz

```
$ MATCH (n:Movie) RETURN n
```

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with icons for search, home, and logs. The main area displays two nodes representing movies. The first node has the ID 'n' and contains the following JSON data:

```
{  
    "title": "The Matrix",  
    "tagline": "Welcome to the Real  
world",  
    "released": 1999  
}
```

The second node also has the ID 'n' and contains the following JSON data:

```
{  
    "title": "The Matrix Reloaded",  
    "tagline": "Free your mind",  
    "released": 2003  
}
```

At the bottom of the browser window, a status message reads: "Started streaming 38 records after 3 ms and completed after 5 ms."

- dosta je neprirodno posmatrati podatke tabelarno
  - jedino ima smisla ako za npr. izlistavanje naziva nekog filma
  - uglavnom koristimo grafovski prikaz

```
$ MATCH (n:Movie) RETURN n
```

"n"
{"title": "The Matrix", "tagline": "Welcome to the Real World", "released": 1999}
{"title": "The Matrix Reloaded", "tagline": "Free your mind", "released": 2003}
{"title": "The Matrix Revolutions", "tagline": "Everything that has a beginning has an end", "released": 2003}
{"title": "The Devil's Advocate", "tagline": "Evil has its winning ways", "released": 1997}
{"title": "A Few Good Men", "tagline": "In the heart of the nation's capital, in a courthouse of the U.S. government, one man will stop at nothing to keep his honor, and one will stop at nothing to find the truth.", "released": 1992}
{"title": "Top Gun", "tagline": "I feel the need, the need for speed.", "released": 1986}
{"title": "Jerry Maguire", "tagline": "The rest of his life begins now."}

39.

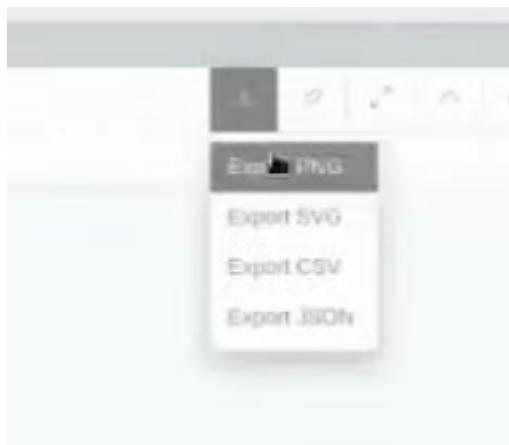
- tekstualni prikaz

```
$ MATCH (n:Movie) RETURN n
```

Server version	Neo4j/3.5.13
Server address	nastava.is.pmf.uns.ac.rs:7687
Query	MATCH (n:Movie) RETURN n
Summary	{ "statement": { "text": "MATCH (n:Movie) RETURN n", ... }}
Response	[ { "keys": [ ... ] } ]

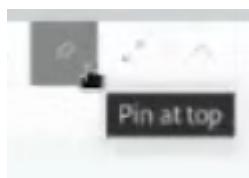
40.

- deo sa code-om je servisna info.



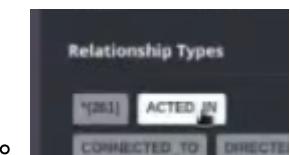
41.

- export

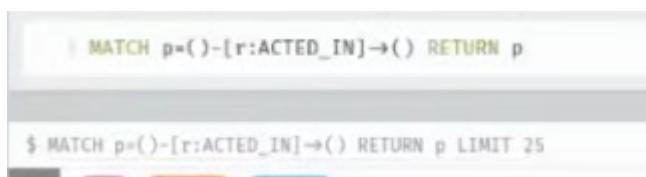


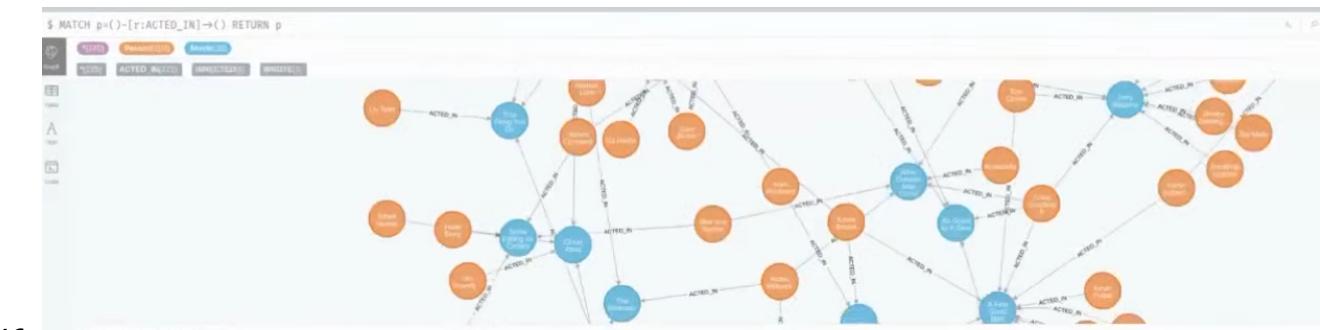
42.

### 43. za grane



44.





46.

- movie : 38, ali nas to nije zanimalo, ni 140 cvorova, vec 170 veza

## Zadaci

### 1. Kreirati čvor, koji reprezentuje novi film (specifirati title, released, tagline)

47.

- `create (m:Movie{title:"Zona Zamfirova", released:2002, tagline:"Ene ga kuce u caksire"})`
- `Match (n:Movie) return n`
- `Match (n:Movie{title:"Zona Zamfirova"}) return n`

### 2. Kreirati čvor, koji reprezentuje novu osobu (specifirati name, born)

48.

- `create(p:Person {name:"Vojin Cetkovic", born:1971})`
- `Match (p:Person{name:"Vojin Cetkovic"}) return p`

### 3. Kreirati granu, koja povezuje novokreirani film i novokreiranu osobu (specifirati role)

49.

- 2 nacina:
  - `MATCH (p:Person{name:"Vojin Cetkovic"}),(m:Movie{title:"Zona Zamfirova"}) CREATE (p)-[r:ACTED_IN{roles:"Mane Kujundzija"}]->(m)] return r`
  - `MATCH (p:Person),(m:Movie) WHERE p.name:"Vojin Cetkovic" and m.title:"Zona Zamfirova" CREATE (p)-[r:ACTED_IN{roles:"Mane Kujundzija"}]->(m)] return r`
  - POGRESNO je NAPISANO na prvu loptu **role** umesto **roles** i ono je to prihvatile stoga, nije kao u RBP da ce da pazi na uniformnost, vec moramo sami

4. Prikazati čvorove tipe Movie, koji imaju naslov *The godfather* (ili neki drgi dostupan naslov)

50.

- `MATCH (m:Movie {title:"The godfather"}) return m`

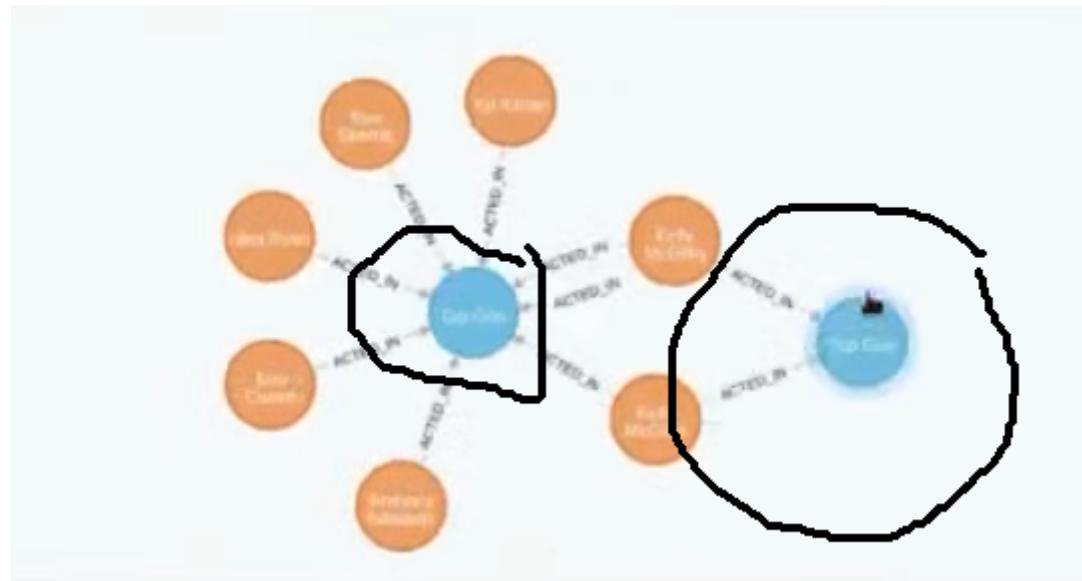
5. Prikazati osobe, koje su glumile u filmu sa naslovom *The godfather* (ili neki drgi dostupan naslov)

51.

- `match (p)-[r:ACTED_IN]->(m:Movie{title:"The godfather"}) RETURN p`

ili ( OVO JE NESTO SPOREDNO, kao zackoljica )

- `match p=()-[r:ACTED_IN]->(m:Movie{title:"The godfather"}) RETURN p`



Moj ubog pokusaj

```
MATCH (p:Person) WHERE (p)-[:ACTED_IN]->(m:Movie{title:"The godfather"})  
RETURN p
```

6. Prikazati naslove i godine filmova, u kojima *Marlon Brando* (ili neki drgi dostupan glumac) glumi

52.

- sada ne zelimo ceo cvor
  - vec naslov i godinu filma
- `match (m:Movie)<-[:ACTED_IN]-(p:Person{title:"Marlon Brando"}) return m.title, m.released`

## 7. Prikazati osobe, koje glume u filmovima, u kojima i *Al Pacino* (ili neki drgi dostupan glumac) glumi

53.

- `match (:Person{name:"Al Pacino"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(p) return p`

ili

- `match (p)-[:ACTED_IN]->(m)<-[:ACTED_IN]-(p1:Person{name:"Al Pacino"}) return p`

moj neki rad

```
match (p:Person), (ap:Person{name:"Al Pacino"}) WHERE (ap)-[:ACTED_IN]->()
() return p
```

## 8. Dodati nadimak postojećoj osobi.

54.

- `match (p{name: "Marlon Brando"}) set p.nickname = "Bud" return p`

## 9. Prikazati sve osobe, sortirane po imenima, rastuće

55.

- `match (p:Person) return p.name ORDER BY p.name ASC`

pogresan pristup

- `match (p:Person) return p ORDER BY p.name ASC`

### Zadaci

10. Prikazati sve osobe, koje su rođene pre 1970. godine.

56.

- `match (p:Person) where p.born < 1970 return p`

greska: moze da postoji jos neki tip sa born

- `match (p) where p.born < 1970 return p`

bez where ne moze

## 11. Obrisati atribut nadimak, koji je prethodno dodat, za osobu *Marlon Brando* (ili neka druga dostupna osoba).

57.

- `match (p{name: "Marlon Brando"}) remove p.nickname return p`

## 12. Dodati osobinu `isOld` sa vrednošću `true`, za sve osobe rođene 1930.

58.

- `merge (p:Person{born: 1930}) on match set p.isOld = TRUE return p`

moglo je i ovako

- `match (p:Person{born: 1930}) set p.isOld = TRUE return p`

## 13. Obrisati osobu sa imenom *Marlon Brando* (ili neka druga dostupna osoba).

59.

- `match (p{name: "Marlon Brando"}) delete p`, ALI DESI SE crash,

ERROR Neo.ClientError.Schema.ConstraintValidationFailed  
Cannot delete node<#0131>, because it still has relationships. To delete this node, you must first delete its relationships.

- pa sada je potrebno delovati tako da moramo brisati veze sa ovim cvorom, ALI POSTOJI JEDNOSTAVNIJI NACIN:

- `match (p{name: "Marlon Brando"}) detach delete p`
- "razvezao"

## 14. Prikazati sve producente.

60.

- producent je **osoba** vezan vezom `[:PRODUCED]`
- `match (p:Person)-[:PRODUCED]->(m) return p`

## 15. Prikazati sve režisere, koji su režirali filmove, u kojima je producent *Joel Silver*.

61.

- `match (d:Person)-[:DIRECTED]->(m)<-[:PRODUCED]-(:Person{name:"Joel Silver"}) return d`

## 16. Prikazati sve filmove, koji imaju review sa ratingom 100.

62.

- `match (m)<-[REVIEWED{rating: 100}]->(p) return m`

63. odluka je dizajna da nema director, producent, actor,...; nego vec po glagolu u vezi da se vrsti struktuiranje

## 17. Prikazati osobe, koje su i producenti i reziseri istog filma u isto vreme.

64.

- `match (p:Person)-[:PRODUCED]->()-[:DIRECTED]-(p) return p, m`

## 7 vezba

---

65. Ucitavanje csv fajla u bazu, npr. `book1.csv`

66. Direktno iz Cypher

67. nista obimno u ovom slucaju

68. mongoDB fazon dataset nije dobar, jer ima puno ugnjezdavanja

69. odsekla je i uzela 2000 linija

70. za kolonu gde se podaci ponavljaju generise se cvor

- prica o vezivanju cvorova
  - npr. `publisher_date` je property veze izmedju cvora `book` i `publisher`

71. radi ilustracije pravimo poseban cvor, npr. `review`

72. Zadatak da se otklone greske/nepotrebnosti iz fajla: prosireni razmaci, preugracak naziv lepo prikazati

bookID	title	authors
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPré
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPré
3	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling
4	Harry Potter and the Prisoner of Azkaban (Harry Potter #3)	J.K. Rowling/Mary GrandPré
5	Harry Potter Boxed Set: Books 1-5 (Harry Potter #1-5)	J.K. Rowling/Mary GrandPré
6	Unhinged Harry Potter Book Seven News: "Half-Blood Prince" Analysis and Speculation	W. Michael Zimmerman
7	Harry Potter Collection (Harry Potter #1-6)	J.K. Rowling
8	The Ultimate Hitchhiker's Guide: Five Complete Novels and One Story (Hitchhiker's Guide to the Galaxy #1-5)	Douglas Adams
9	The Ultimate Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #1-9)	Douglas Adams
10	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #1)	Douglas Adams
11	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #2)	Douglas Adams
12	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #3)	Douglas Adams
13	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #4)	Douglas Adams
14	The Hitchhiker's Guide to the Galaxy (Hitchhiker's Guide to the Galaxy #5)	Douglas Adams
15	A Short History of Nearly Everything	Douglas Adams
16	Bryson's African Diary	Bill Bryson
17	Bryson's Dictionary of Troublesome Words: A Writer's Guide to Getting It Right	Bill Bryson
18	In a Sunburned Country	Bill Bryson
19	I'm a Stranger Here Myself: Notes on Returning to America After Twenty Years Away	Bill Bryson
20	The Lost Continent: Travels in Small Town America	Bill Bryson
21	Neither Here nor There: Travels in Europe	Bill Bryson
22	Notes from a Small Island	Bill Bryson
23	The Mother Tongue: English and How It Got That Way	Bill Bryson
24	J.R.R. Tolkien 4-Book Boxed Set: The Hobbit and The Lord of the Rings	J.R.R. Tolkien
25	The Hobbit (The Hobbit and The Lord of the Rings #1-3)	J.R.R. Tolkien
26	The Fellowship of the Ring (The Lord of the Rings #1)	J.R.R. Tolkien
27	The Return of the King (The Lord of the Rings #2)	J.R.R. Tolkien
28	The Lord of the Rings (The Lord of the Rings #3)	J.R.R. Tolkien/Alan Lee
29	The Lord of the Rings: Weapons and Warfare	Chris Smith/Christopher Lee/Richard Taylor
30	The Lord of the Rings: Complete Visual Companion	Jude Fisher
31	Agile Web Development with Rails: A Pragmatic Guide	Dave Thomas/David Heinemeier Hansson/Lee
32	Hatchet (Brian's Saga #1)	Gary Paulsen
33	Hatchet: A Guide for Using "Hatchet" in the Classroom	Donna Ikenberry/Edward Sorensen/Kelli Vassanca
34	Guts: The True Stories behind Hatchet and the Brian Books	Gary Paulsen
35	Molly Hatchet - 3 at the Best	Molly Hatchet
36	Hatchet (Jewell's Guide on Contemporary Fiction)	Doreen Kernan
37	The Changing Sea (Changeling #1)	Angela Knight/Sahara Kelly/Judy Mayes/Martie
38	The Changeling (Changeling #2)	Della Strommen
39	The Changeling (Changeling #3)	Frances A. McMillip
40	The Changeling (Changeling #4)	Zilpha Keatley Snyder
41	The Changeling (Changeling #5)	Kate Hosley
42	The Changeling (Changeling #6)	Philippe Carr

73.

## 74. postoji i prezentacija

## Učitavanje csv fajlova u bazu

- Razni načini za direktno učitavanje:
    - LOAD CSV Cypher komanda (za manje i srednje data set-ove, do 10 miliona instanci).
    - Neo4j-admin bulk import tool - upotreba iz komandne linije, za obimne količine podataka
    - Kettle import tool - dobro radi sa velikim količinama podataka
  - - `load csv` je koristan za manjeg obima importe
    - `bulk import` iz cli-a
    - `kettle import tool`

## Učitavanje csv fajlova u bazu

- Load csv:
    - Moguće je učitati podatke sa nekog URL-ja
    - Direktno mapiranje na kompleksne grafovske strukture
    - Rešavanje konverzija
    - ...
    - Rad sa lokalnim fajlovima: *file:///*
    - Rad sa fajlovima sa web-a: *https://host/path/data.csv*

75.

Load csv - primer:

- Books.csv – na Google drive-u kao Excel sheet
- Potrebno je dobiti link sa *Publish to web* opcijom:  
[https://docs.google.com/spreadsheets/d/e/2PACX-1vRiKjm2Wo5Gb1E4FrE6rrBw79sqlpvnAqvy3uQKKfmH\\_Oy6-GLCbS\\_CCCrmacxiAv-XfsACKGnU\\_YJx/pubhtml](https://docs.google.com/spreadsheets/d/e/2PACX-1vRiKjm2Wo5Gb1E4FrE6rrBw79sqlpvnAqvy3uQKKfmH_Oy6-GLCbS_CCCrmacxiAv-XfsACKGnU_YJx/pubhtml)
- Ili staviti u odgovarajući folder (zapisano u neo4j.conf)

76.

- potreba za posedovanjem lokalnog neo4j

```
[dtesendic@nastava: ~]$ cd /etc/neo4j
[dtesendic@nastava: /etc/neo4j]$ ls
neo4j.conf  neo4j.logrotate
[dtesendic@nastava: /etc/neo4j]$
```

77.



78.

```
[dtesendic@nastava: ~]$ cd /etc/neo4j
[dtesendic@nastava: /etc/neo4j]$ ls
neo4j.conf  neo4j.logrotate
[dtesendic@nastava: /etc/neo4j]$
```

79.

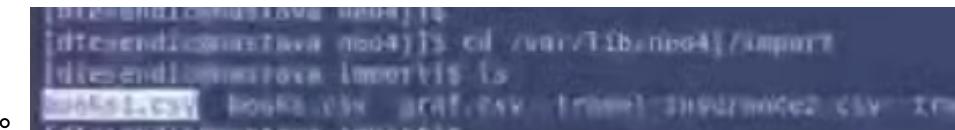
```
[dtesendic@nastava: /etc/neo4j]$ nano neo4j.conf
[dtesendic@nastava: /etc/neo4j]$ cat neo4j.conf
# Neo4j configuration
#
# For more details and a complete list of settings, please see
# https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/
#
# The name of the database to mount
dbms.active_database=graph.db
#
# Paths of directories in the installation.
dbms.directories.data=/var/lib/neo4j/data
dbms.directories.plugins=/var/lib/neo4j/plugins
dbms.directories.certificates=/var/lib/neo4j/certificates
dbms.directories.logs=/var/log/neo4j
dbms.directories.lib=/usr/share/neo4j/lib
dbms.directories.run=/var/run/neo4j
#
# This setting constrains all LOAD CSV import files to be under the IMPORT directory. Remove or comment it out to
# allow files to be loaded from anywhere in the filesystem; this introduces possible security problems. See the
# LOAD CSV section of the manual for details.
#dbms.directories.import=/var/lib/neo4j/import
#
# Whether requests to Neo4j are authenticated.
# To disable authentication, uncomment this line
#dbms.security.auth_enabled=false
#
# Enable this to be able to upgrade a store from an older version.
#dbms.upgrade=true
[dtesendic@nastava: /etc/neo4j]$
```

80.

```
[dtesendic@nastava: /etc/neo4j]$ nano neo4j.conf
[dtesendic@nastava: /etc/neo4j]$ cat neo4j.conf
#
# Miscellaneous configuration
#cypher.default_language=cql
#
# Enable this to specify a parser other than the default one
#cypher.default_language=cql
#
# Determines if Cypher will allow using file URLs when loading data using
# LOAD CSV. Setting this value to 'false' will cause Neo4j to fail LOAD
# clauses that load data from the file system.
#dbms.security.allow_csv_import_from_file_urls=true
[dtesendic@nastava: /etc/neo4j]$
```

81.

82. kopiranje csv-ova na namestenu putanju

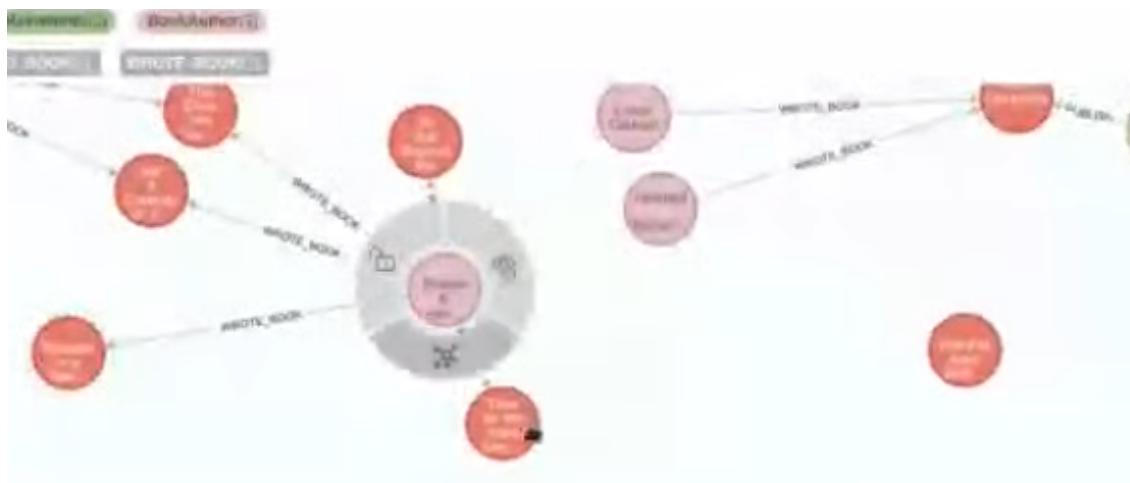


83.

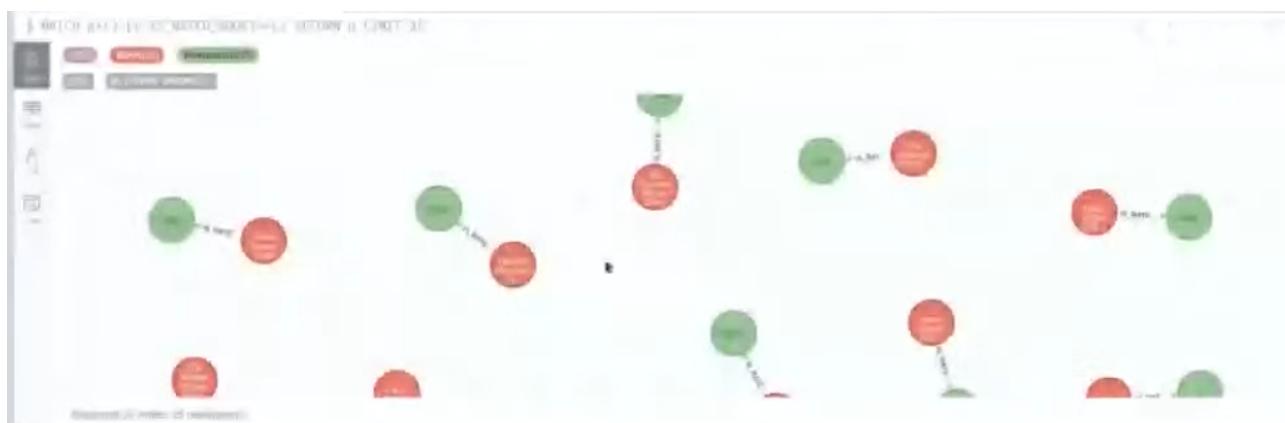
```
CREATE CONSTRAINT ON (b:Book) ASSERT b.bookID IS UNIQUE;
CREATE CONSTRAINT ON (a:BookAuthor) ASSERT a.name IS UNIQUE;
CREATE CONSTRAINT ON (p:Publisher) ASSERT p.name IS UNIQUE;
LOAD CSV WITH HEADERS FROM "file:///books1.csv" AS line
WITH line, SPLIT(line.authors, '/') AS authors,
SPLIT(line.publication_date,"/") as
dateComponents
MERGE (book: Book {bookID:line.bookID, title:line.title, isbn:line.isbn,
isbn13:line.isbn13,
language: line.language_code, num_pages:line.num_pages})
MERGE (publisher:Publisher {name:line.publisher})
MERGE (reviewInfo:ReviewInfo {avg_rating:line.average_rating,
ratings_count:line.ratings_count,
text_reviews_count:line.text_reviews_count})
FOREACH (author IN authors| MERGE (auth:BookAuthor {name: author}) MERGE
(auth)-[:WROTE_BOOK]->(book))
MERGE (book)-[:IS_RATED_BOOK]->(reviewInfo)
MERGE (publisher)-[:PUBLISHED_BOOK {date:date({day:
toInteger(dateComponents[1]),
month: toInteger(dateComponents[0]), year:
toInteger(dateComponents[2])})}]->(book)
```



84.

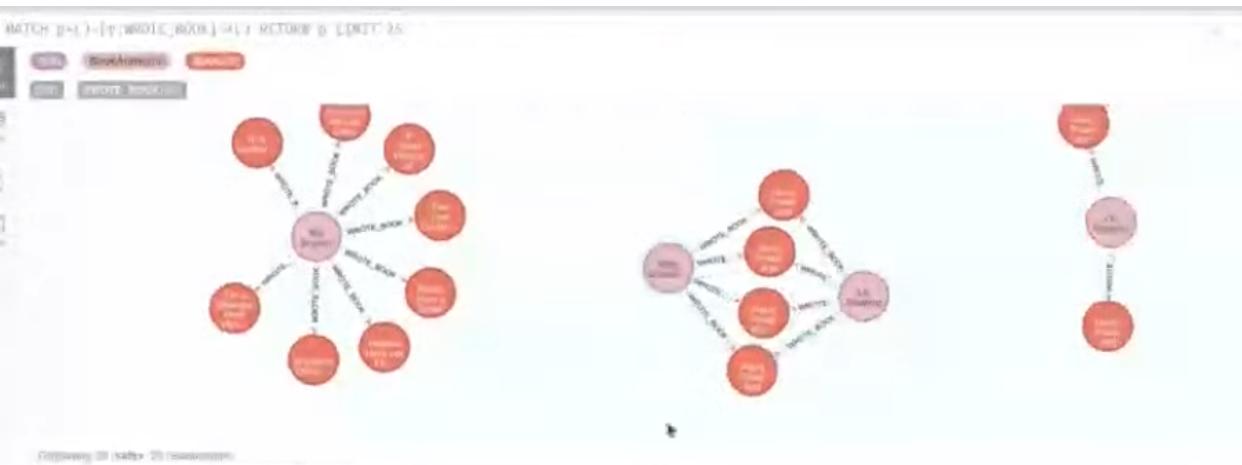
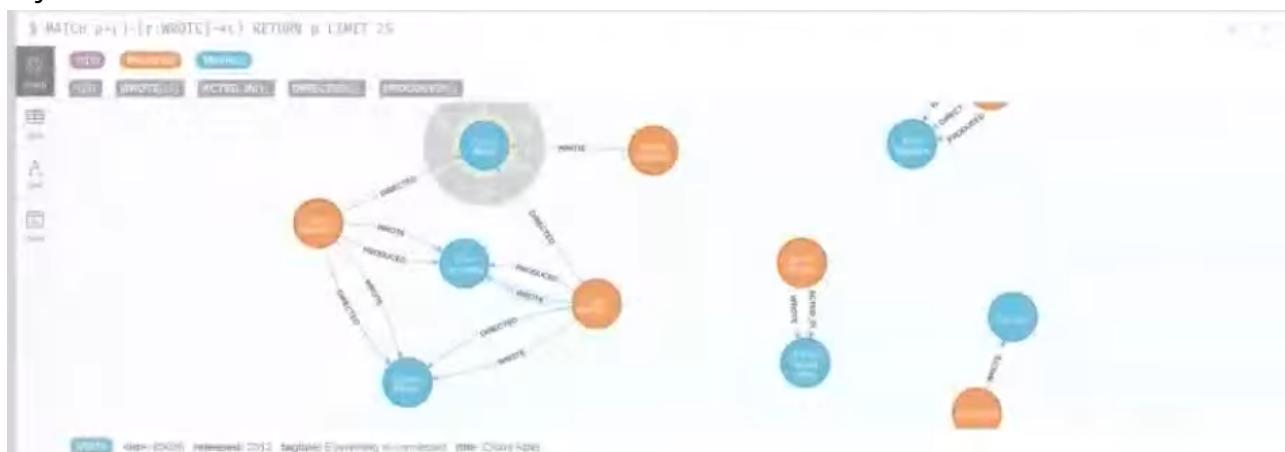


85.



86.

### 87. NIJE OVO HTELA DA PRIKAZE



88.

- Load csv – primer: Books.csv
  - Preprocesiranje fajla
  - Identifikovati čvorove i veze
  - Insert-ovanje

89.

- preprocesiranje je pomenuto gore

- <https://neo4j.com/developer/guide-import-csv/>
- <https://neo4j.com/docs/cypher-manual/>

90.

91. pregled gornjeg koda:

```
CREATE CONSTRAINT ON (b:Book) ASSERT b.bookID IS UNIQUE;
```

```
CREATE CONSTRAINT ON (a:BookAuthor) ASSERT a.name IS UNIQUE;
```

```
CREATE CONSTRAINT ON (p:Publisher) ASSERT p.name IS UNIQUE;
```

- 
- splitovanje
  - LOAD CSV WITH HEADERS FROM "file:///books1.csv" AS line  
WITH line, SPLIT(line.authors, ',') AS authors, SPLIT(line.publication\_date, "/") as dateComponents
- lepse merge od insert, jer radi update
  - MERGE (book: Book {bookID:line.bookID, title:line.title, isbn:line.isbn, isbn13:line.isbn13, language: line.language\_code, num\_pages:line.num\_pages})
- lep primerak kako se radi
  - FOREACH (author IN authors| MERGE (auth:BookAuthor {name: author}) MERGE (auth)-[:WROTE\_BOOK]->(book))
- interesantno zbog datuma
  - MERGE (publisher)-[:PUBLISHED\_BOOK {date:date({day: toInteger(dateComponents[1]), month: toInteger(dateComponents[0]), year: toInteger(dateComponents[2])})} ]->(book)

- Přístup z Java:

- Dependency:

```
<groupId>org.neo4j.driver</groupId>
<artifactId>neo4j-java-driver</artifactId>
<version>4.2.0</version>
```

92.

- z Java

- Přístup z Java:

```
driver = GraphDatabase.driver("bolt://hostAddress:7687", AuthTokens.basic("admin", "admin"));
Session session = driver.session();
session.writeTransaction(new TransactionWork<Integer>() {
    @Override
    public Integer execute(Transaction tx) {
        Map<String, Object> parameters = new HashMap<String, Object>();
        parameters.put("title", title); parameters.put("released", released);
        tx.run("CREATE (m:Movie {title:$title, released:$released})", parameters);
        for(int i=0;i<actors.length;i++){
            tx.run("MATCH (a:Movie{title:$title}),(b:Person{name:'" + actors[i] + "'})"
                  "CREATE (b)-[r:ACTED_IN]->(a)", parameters);
        }
    }
});
```

- tj.

```
driver = GraphDatabase.driver("bolt://hostAddress:7687", AuthTokens.basic("admin", "admin"));
Session session = driver.session();
session.writeTransaction(new TransactionWork<Integer>() {
    @Override
    public Integer execute(Transaction tx) {
        Map<String, Object> parameters = new HashMap<String, Object>();
        parameters.put("title", title); parameters.put("released", released);
        tx.run("CREATE (m:Movie {title:$title, released:$released})", parameters);
        for(int i=0;i<actors.length;i++){
            tx.run("MATCH (a:Movie{title:$title}), (b:Person{name:'" + actors[i] + "'})"
                  "CREATE (b)-[r:ACTED_IN]->(a)", parameters);
        }
    }
});
```

93. ZA PROJEKAT MOZE NA OBA NACINA, Najlakse cypher

94. mora se iskoristiti graphDataScience biblioteka