

Računarstvo i informatika

*Katedra za računarstvo
Elektronski fakultet u Nišu*

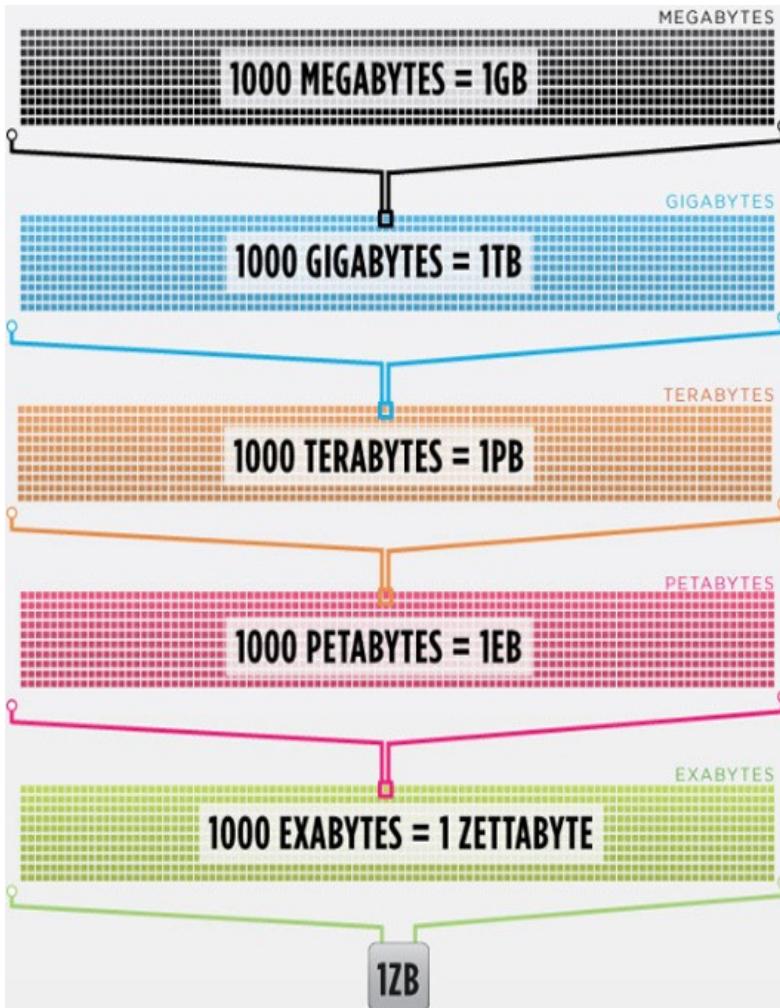
Napredne baze podataka **Graf baze podataka**

Zimski semestar 2020/2021

Sadržaj

- Graf baze podataka
- Neo4J

Graf baze podataka



- Količina podataka dostupona na Internetu postaje sve veća.
- Svaka dva dana kreira se količina informacija koja je bila dostupna na Internet-u do 2003. godine.
- **Sve veća povezanost podataka:**
 - Text (sadržaj)
 - HyperText (pokazivači)
 - RSS (grupisanje pokazivača)
 - Blogs (dodat pingback)
 - Tagging (grupisanje sličnih podataka)
 - RDF (opis povezanih podataka)
 - **GGG (sadržaj + pokazivači + veze + opis)**

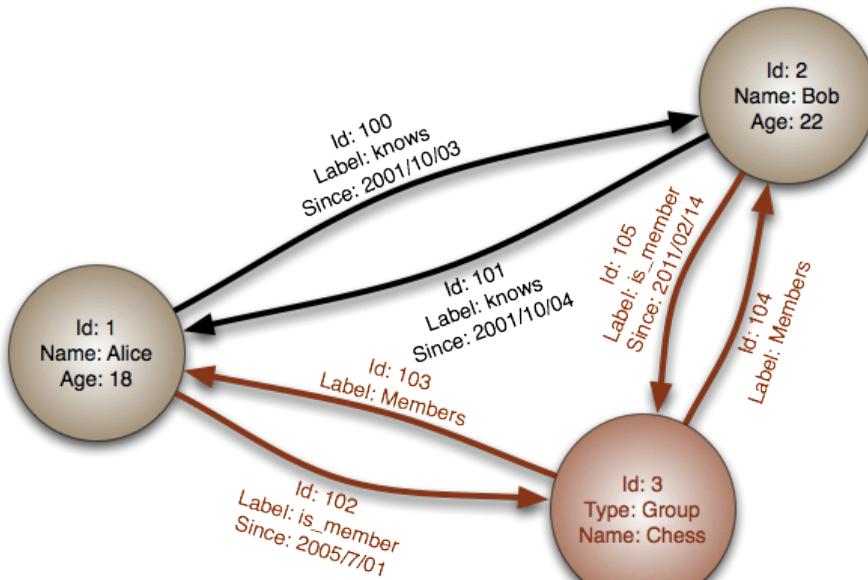
Graf baze podataka

- Bitna karakteristika podataka na Internetu je da su **polustruktuirani ili nestrukturiani**.
- **Primer:** Model podataka koji prikazuje podatke o filmovima: Glumci, Uloge, Lokacije, Datumi, Filmovi, Reditelji, Producenti, Ocene, Prikazivanja i sl.
- Kod ovog modela **broj veza između entiteta može drastično da varira**.
- **Svaki entitet** može da ima **proizvoljan broj veza** ka drugim entitetima.
- **Kompleksna i neregularna struktura**.
- Relaciona baza podataka nije pogodna za predstavljanje ovakvih modela podataka.

Graf baza podataka

- **Graf baze podataka** koriste **strukturu podataka tipa GRAF** (čvorovi, potezi i svojstva) za reprezentaciju i skladištenje podataka.
- **Po definiciji, graf baza podataka je bilo koji sistem za skladištenje podataka koji omogućava definisanje relacija nezavisno od indeksnih i lookup struktura.**
- To znači da **svaki element** može da sadrži **direktan pokazivač** ka drugim elementima bez potrebe za postojanje indeksnih struktura.
- Pored generalnih graf baza podataka, koje mogu da se koriste za skladištenje bilo kog grafa, postoje i specijalizovane graf baze podataka:
 - **Triplestore** (skladištenej podataka u obliku subjekat-predikat-objekat, RDF)
 - **Mrežne baze podataka** (baziraju se na korišćenju mrežnog modela)

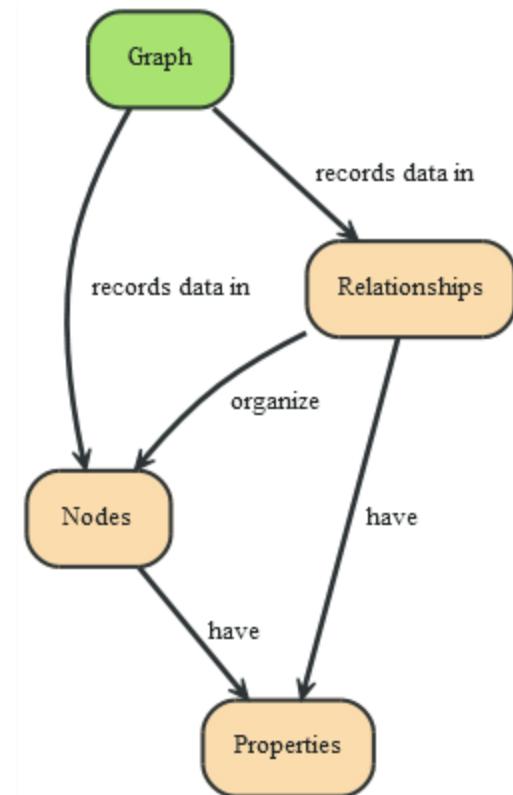
Graf baze podataka



- Graf baza podataka koristi strukturu tipa graf.
- **Elementi**: čvorovi, potezi i svojstva.
- **Koncept čvorova** je sličan konceptu objekta u OO programiranju.
- **Čvorovi** se koriste za reprezentaciju entiteta.
- **Svojstva** su podaci koji opisuju čvorove.
- **Potezi** su **relacije** koje povezuju čvorove međusobno ili čvorove sa svojstvima.
- Svojstva mogu biti pridružena i potezima.

Graf baze podataka

- Najjednostavnija graf baza podataka sadrži samo jedan čvor.
- Čvor može imati **neograničeni broj svojstava**.
- Veze između čvorova organizuju graf u različite strukture (lista, mapa, stablo) koje mogu biti povezane u još kompleksnije strukture.
- Za pretragu grafa koriste se **različiti algoritmi**.
- Da bi se ubrzao pristup i pretraga čvorova mogu se **indeksirati pojedina svojstva**.



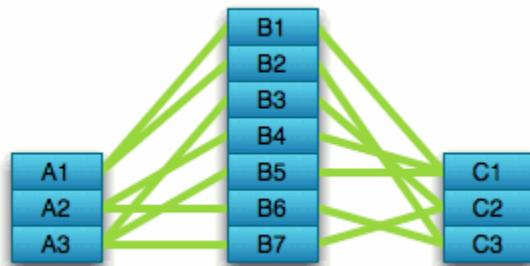
Graf baze podataka

- Graf baze podataka su pogodnije rešenje za **asocijativne skupove podataka**.
- Podaci iz graf baza podataka se **direktno mapiraju** na OO model.
- Dobro rešenje za **velike količine podataka** pošto ne zahtevaju skupe operacije spoja.
- Zahvaljujući **fleksibilnoj šemi** pogodne za razvoj ad-hoc aplikacija ili aplikacija koje brzo evaluiraju.
- Pogodne za operacije tipa: **obilazak grafa, traženje najkraćeg puta između dva čvora, provera da li postoji put između dva čvora, detektovanje paterna, određivanje suseda i sl.**

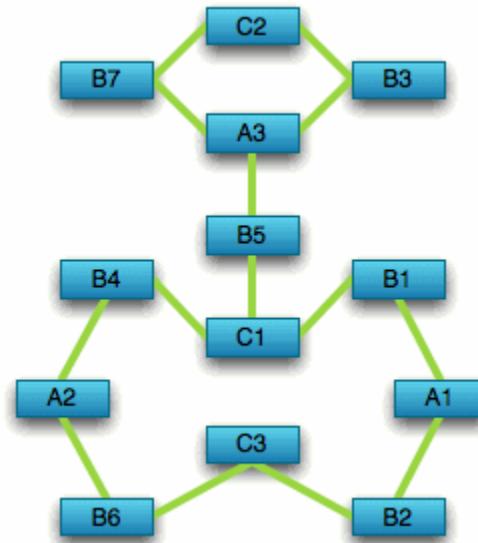
Graf baze podataka

- Poređenje sa relacionim bazama podataka

Relacione baze podataka su
**optimizovane za agregaciju
podataka**



Graf baze podataka su
**optimizovane za veze između
podataka**



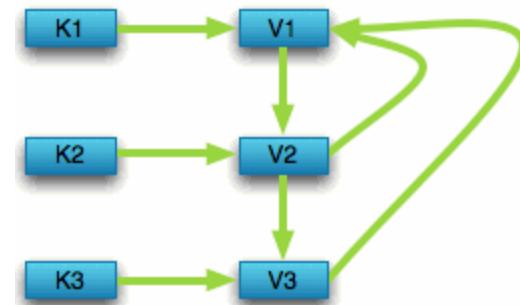
Graf baze podataka

- Poređenje sa key-value skladištima podataka

Key-value skladišta podataka su **optimizovana za jednostavne lookup pretrage**



Graf baze podataka su optimizovane za **obilaske povezanih podataka**



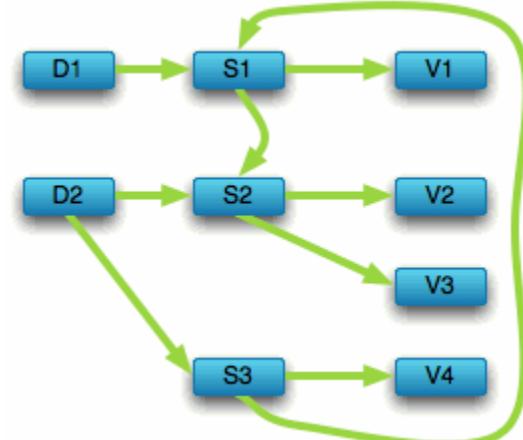
Graf baze podataka

- Poređenje sa key-value skladištima podataka

Key-value skladišta podataka
su **optimizovana za organizaciju podataka u strukture stabla**



Graf baze podataka omogućavaju **organizaciju podataka u različite kompleksne strukture povezanih podataka**



Neo4J

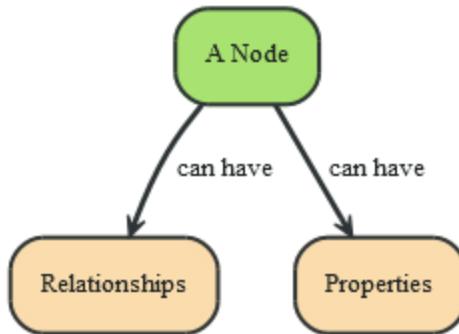
- **Creator:** Swedish media asset project, today Neo Technology company
- **Initial release:** 2006-2007 open source project, 2003 24/7 production, 2010 Neo4J server
- **Stable release:** 4.2.1
- **Programming language:** Java
- **Operating system:** Cross-platform
- **Type :** Graph database (embedded, server)
- **License:** dual licensing (open source + commercial)
- **Website:** <http://neo4j.com/>

Neo4J

- **Graf baza podataka + Lucene Index**
- Property graf
- Full ACID (atomicity, consistency, isolation, durability) – garantuje **konzistentnost podataka** za sve operacije koje se izvršavaju **unutar transakcija**.
- Brze pretraga podataka zahvaljujući algoritmima za obilazak stabla
- Embedded rešenje u Java aplikacijama
- Server rešenje
- **High Availability** (Enterprise Edition) – 32×10^9 čvorova, 32×10^9 relacija, 64×10^9 svojstava
- REST API

Neo4J

- **Osnovne jedinice** Neo4J grafa su **čvorovi i potezi (relacije)**.
- Neo4J dozvoljava da i **čvorovi i potezi poseduju svojstva**.
- Čvorovi se najčešće koriste za predstavljanje entiteta, ali zavisno od domena primene mogu imati i drugačiju ulogu.

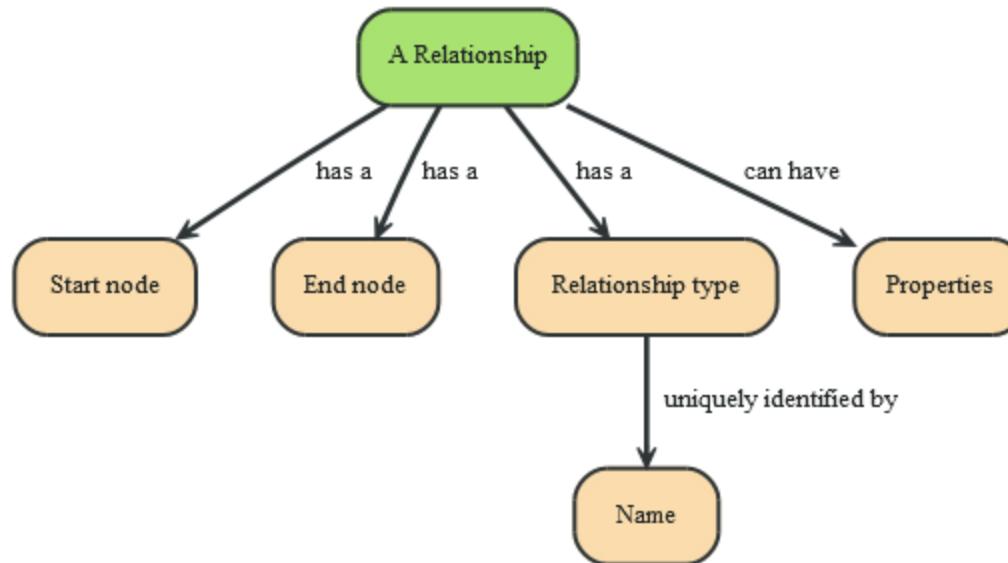


- Najednostavniji graf sadrži samo jedan čvor.



Neo4j

- Relacije između čvorova predstavljaju ključni element grafa.
- Omogućavaju pronalaženje povezanih podataka.

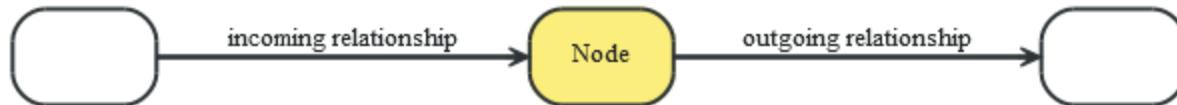


- Relacija povezuje dva čvora i garantovano poseduje **validne polazni i validni krajnji čvor**.

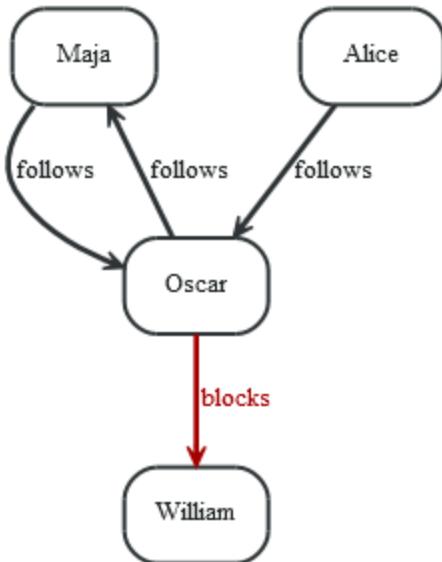


Neo4J

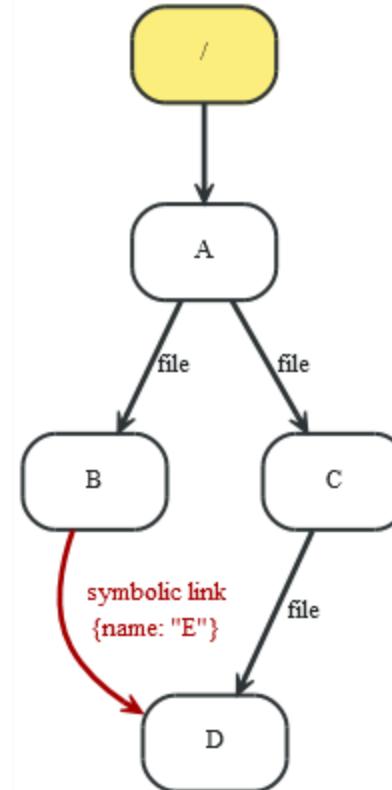
- Relacije su uvek usmerene tako da se razlikuju **dolazni i odlazni potezi**.
- Relacija se **uvek može obići bez obzira na smer**. Prilikom obilaska grafa najčešće se ignorišu relacije čiji smer ne odgovara potrebama.
- Svakoj relaciji je **dodeljen tip** odnosno labela.



Neo4J



Model jednostavne
društvene mreže



Model fajl sistema

Neo4J

- Svojstva predstavljaju **key-value parove**.
- Vrednosti svojstava mogu biti atomične vrednosti ili nizovi atomičnih vrednosti.
- U tabeli je dat spisak podržanih tipova podataka.

Type	Description	Value range
<code>boolean</code>		<code>true/false</code>
<code>byte</code>	8-bit integer	<code>-128 to 127</code> , inclusive
<code>short</code>	16-bit integer	<code>-32768 to 32767</code> , inclusive
<code>int</code>	32-bit integer	<code>-2147483648 to 2147483647</code> , inclusive
<code>long</code>	64-bit integer	<code>-9223372036854775808 to 9223372036854775807</code> , inclusive
<code>float</code>	32-bit IEEE 754 floating-point number	
<code>double</code>	64-bit IEEE 754 floating-point number	
<code>char</code>	16-bit unsigned integers representing Unicode characters	<code>u0000 to uffff (0 to 65535)</code>
<code>String</code>	sequence of Unicode characters	

Neo4J

- **Putanja (path)** predstavlja **kolekciju (najmanje jedan) čvorova koji su međusobno povezani relacijama.**
- Putanja se dobija kao **rezultat upita** odnosno rezultat obilaska grafa.
- **Obilazak grafa** podrazumeva **obilazak čvorova praćenjem veza** između čvorova a na **osnovu zadatog pravila.**

Neo4J

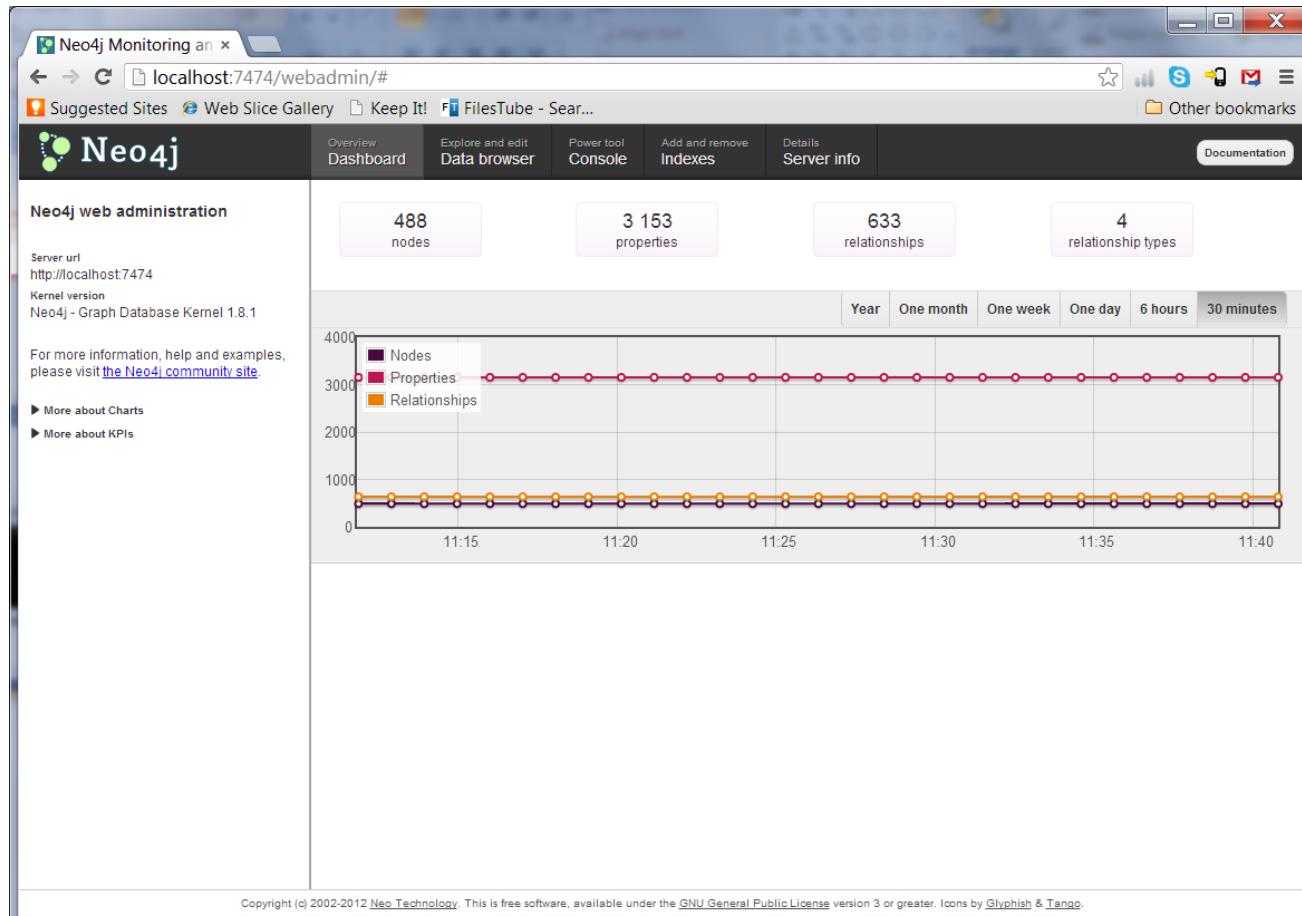
- Neo4J shell

```
[bash-4.4# cypher-shell -u neo4j -p test
Connected to Neo4j 3.5.0 at bolt://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
[neo4j> MATCH (n) RETURN count(n);
+-----+
| count(n) |
+-----+
| 0         |
+-----+
1 row available after 76 ms, consumed after another 0 ms
[neo4j> :exit

Bye!
bash-4.4#
```

Neo4j

- Neo4J Web administracija



Neo4J

- **REST API**
- Za reprezentaciju podataka se podrazumevano koristi JSON, kako za podatke koji se šalju Neo4J serveru tako i za podatke koje nEo4J server vraća klijentima
- Service root: GET i <http://localhost:7474/db/data/>
- **Podrška za streaming** – ne šalje se kompletan JSON dokument odjednom već u delovima

Neo4J

- REST API klijentske biblioteke

name	language / framework	URL
Java-Rest-Binding	Java	https://github.com/neo4j/java-rest-binding/
Neo4jClient	.NET	http://hg.readify.net/neo4jclient/
Neo4jRestNet	.NET	https://github.com/SepiaGroup/Neo4jRestNet
py2neo	Python	http://py2neo.org/
Bulbflow	Python	http://bulbflow.com/
neo4jrestclient	Python	https://github.com/versae/neo4j-rest-client
neo4django	Django	https://github.com/scholrly/neo4django
Neo4jPHP	PHP	https://github.com/jadell/Neo4jPHP
neography	Ruby	https://github.com/maxdemarzi/neography
neoid	Ruby	https://github.com/elado/neoid
node.js	JavaScript	https://github.com/thingdom/node-neo4j
Neocons	Clojure	https://github.com/michaelklishin/neocons

Neo4J

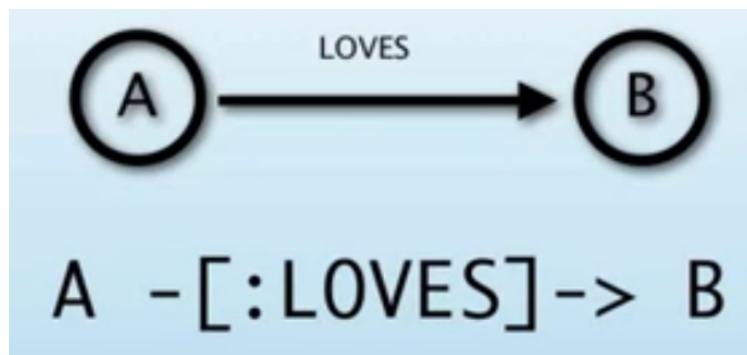
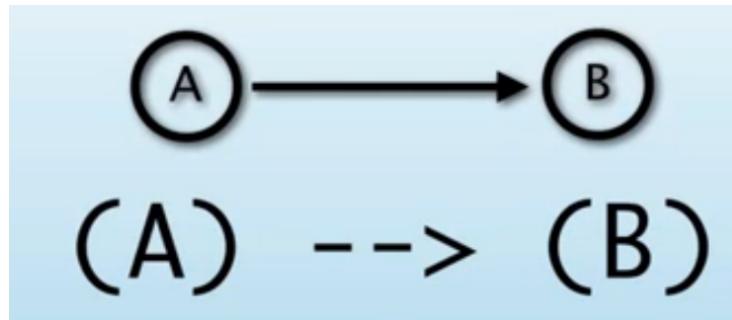
- **Cypher – deklarativni upitni jezik za grafove**
- Omogućava CRUD operacije nad Neo4J grafovima
- Izgrađen je na elementima SQL-a (WHERE i ORDER BY) i SPARQL-a (pattern matching).
- Namjenjen je za kreiranje ad-hoc upita korišćenjem različitih alata ali i za zadavanje upita korišćenjem REST API-ja ili neke klijentske biblioteke.

Neo4J

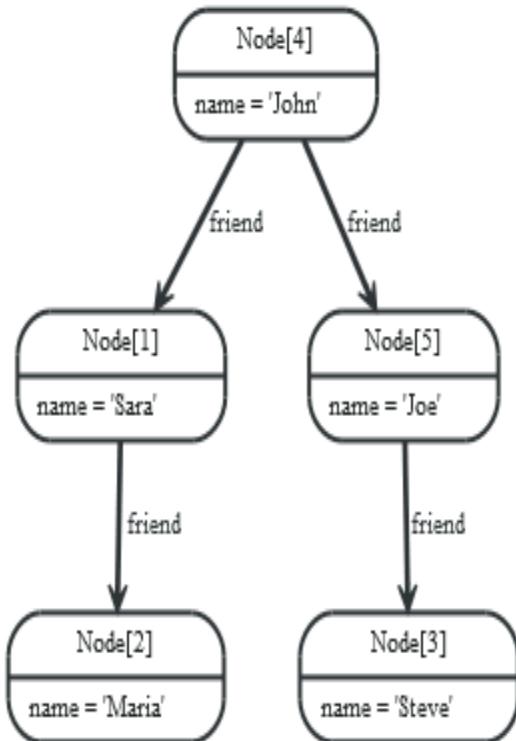
- Elementi Cypher jezika:
 - **START** – početna tačka za pretragu grafa. Dobija se na osnovu ID-a čvora ili pretragom indeksa.
 - **MATCH** – graf patter koji se traži u odnosu na početne tačke definisane u klauzuli START
 - **WHERE** – kriterijum za filtriranje
 - **RETURN** – koje podatke treba vratiti
 - **CREATE** – kreiranje čvorova i potega
 - **DELETE** – brisanej čvorova i potega
 - **SET** – definisanje vrednosti svojstava
 - **FOREACH** – ažuriranje elemenata u listi
 - **WITH** – podela upita na veći broj nezavisnih delova

Neo4J

- Zadavanje graf pattern-a



Neo4J



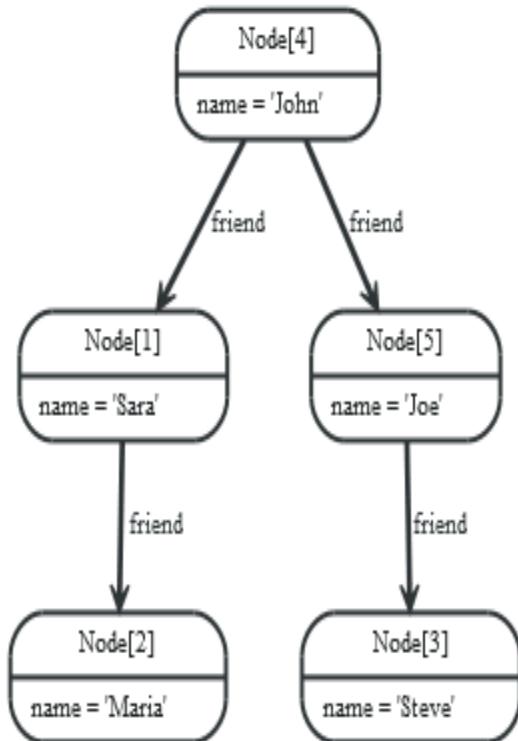
```
START user=node(5,4,1,2,3)
MATCH user-[:friend]->follower
WHERE follower.name =~ 'S.*'
RETURN user, follower.name
```

user	follower.name
Node[5]{name:"Joe"}	"Steve"
Node[4]{name:"John"}	"Sara"
2 rows	
0 ms	

Neo4J

- Neo4J koristi indekse za **lookup pretragu grafa** u cilju pronalaženja čvorova i relacija na koje će se primeniti grafovske operacije.
- Indeksi se koriste da obezbede **jedinstvenost vrednosti** pojedinih svojstava.
- Neo4J koristi različite mehanizme za implementaciju indeksa. Podrazumevani indeks je Lucene.
- Lucene je open-source biblioteka koja obezbeđuje mehanizme za indeksiranje i pretraživanje podataka.
- Indeksi mogu biti:
 - Exact indeksi - indeksira se kompletna vrednos svojstva. Indeksiraju se numeričke ili string vrednosti (sva svojstva mogu da se indeksiraju kao string vrednosti)
 - Full-text indeksi – indeksiraju se sve reči u tekstuallnom podatku

Neo4J

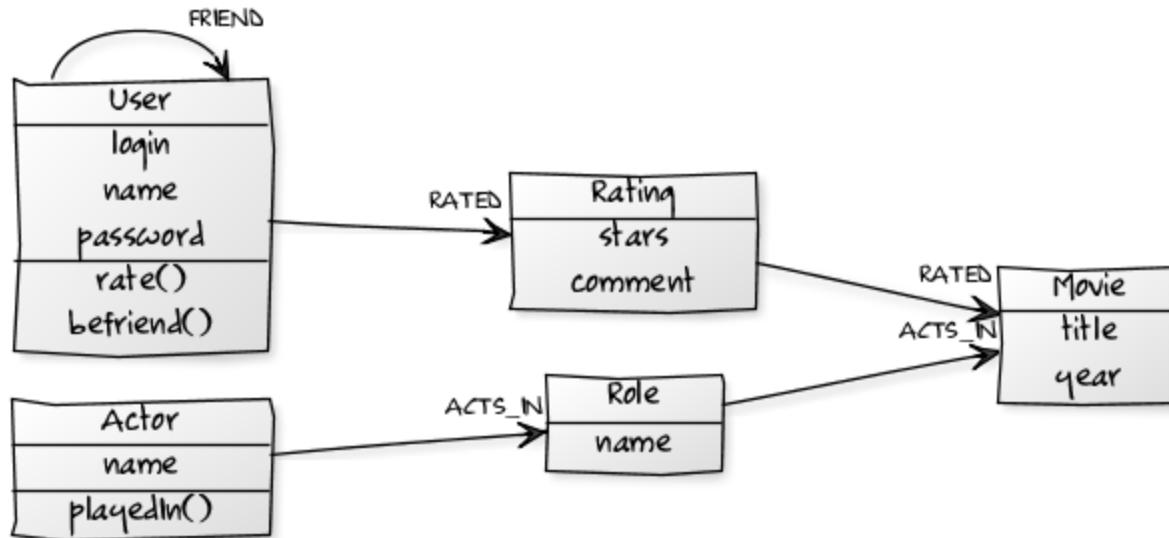


```
START john=node:node_auto_index(name = 'John')
MATCH john-[:friend]->()-[:friend]->fof
RETURN john, fof
```

john	fof
Node[4]{name:"John"}	Node[2]{name:"Maria"}
Node[4]{name:"John"}	Node[3]{name:"Steve"}
2 rows	
3 ms	

Neo4j

- cineasts_39_movies_446_actors sample dataset



Tipična primena

- Društvene mreže
- Sistemi za davanje preporuka/personalizaciju
(Recommender/personalization system)
- Knowledge graph
- Detekcija prevara i zloupotreba
- Lanac snabdevanja (supply chain management)
- Upravljanje identitetom i pravima pristupa
- Evidencija o mrežnoj infrastrukturi
- Upravljanje domenskim (master) modelom