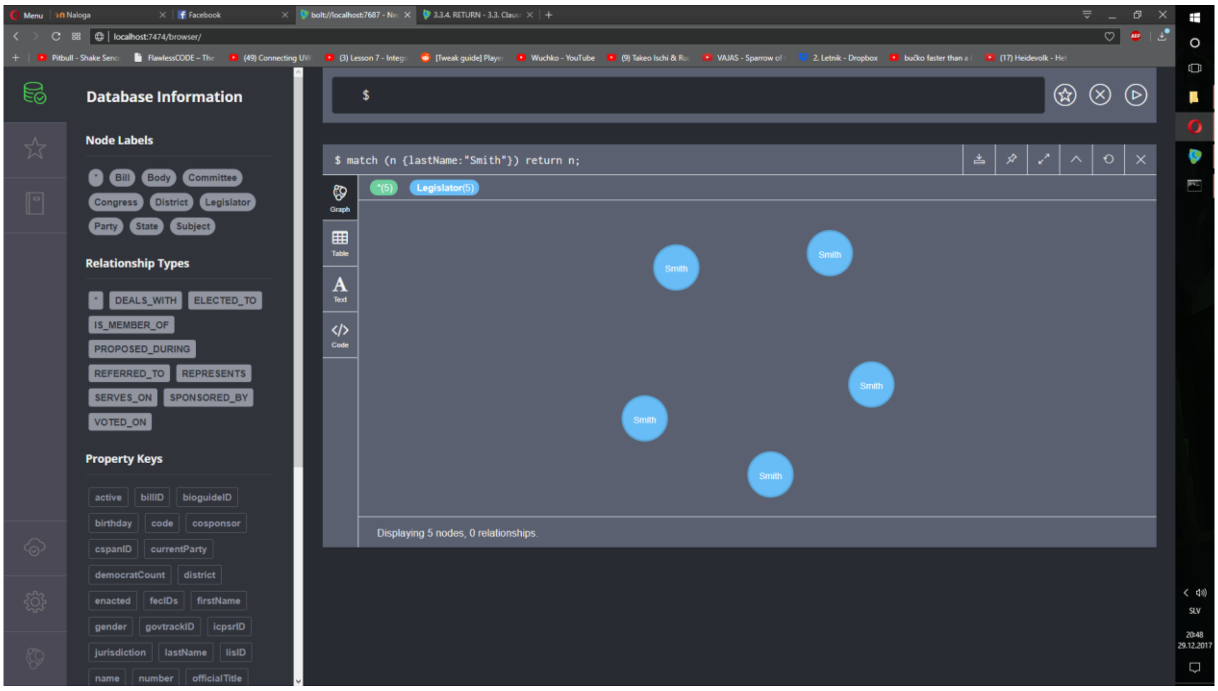
Neo4j

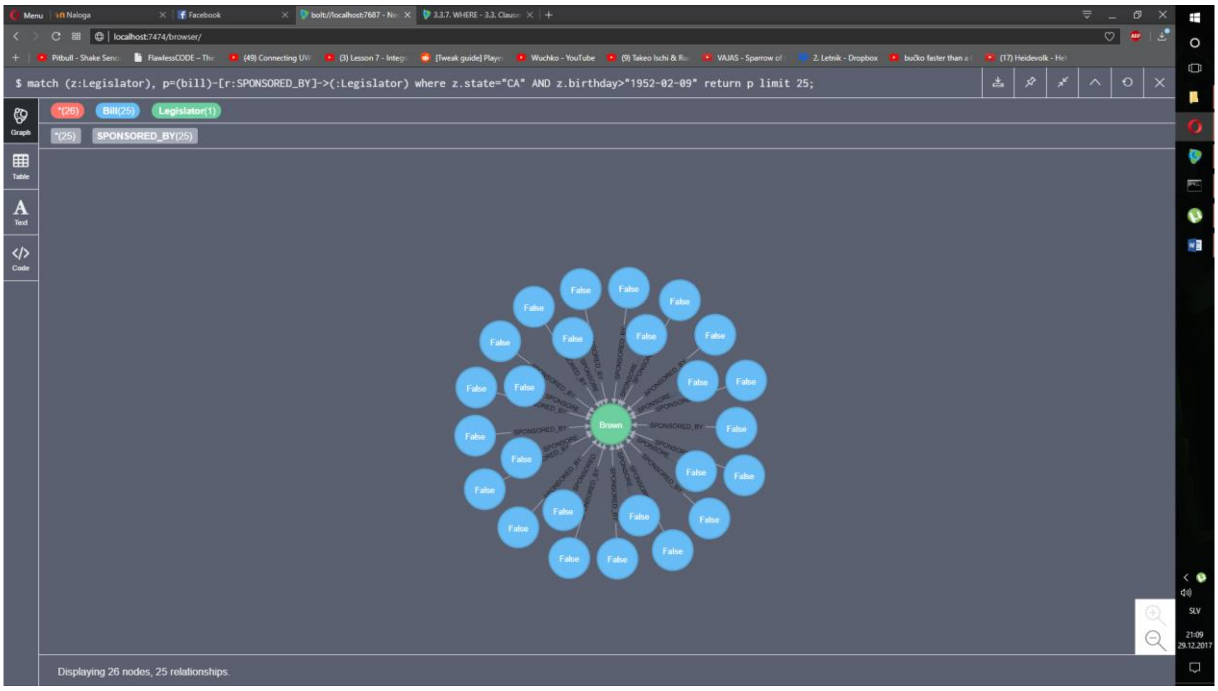
1. match (n {lastName:«Smith«}) return n;



```
$ match (n {lastName:"Smith"}) return n;
```

Displaying 5 nodes, 0 relationships.

2. match (z:Legislator), p=(bill)-[r:SPONSORED_BY]->(:Legislator) where z.state="CA" AND z.birthday>"1952-02-09" return p limit 25;



```
$ match (z:Legislator), p=(bill)-[r:SPONSORED_BY]->(:Legislator) where z.state="CA" AND z.birthday>"1952-02-09" return p limit 25;
```

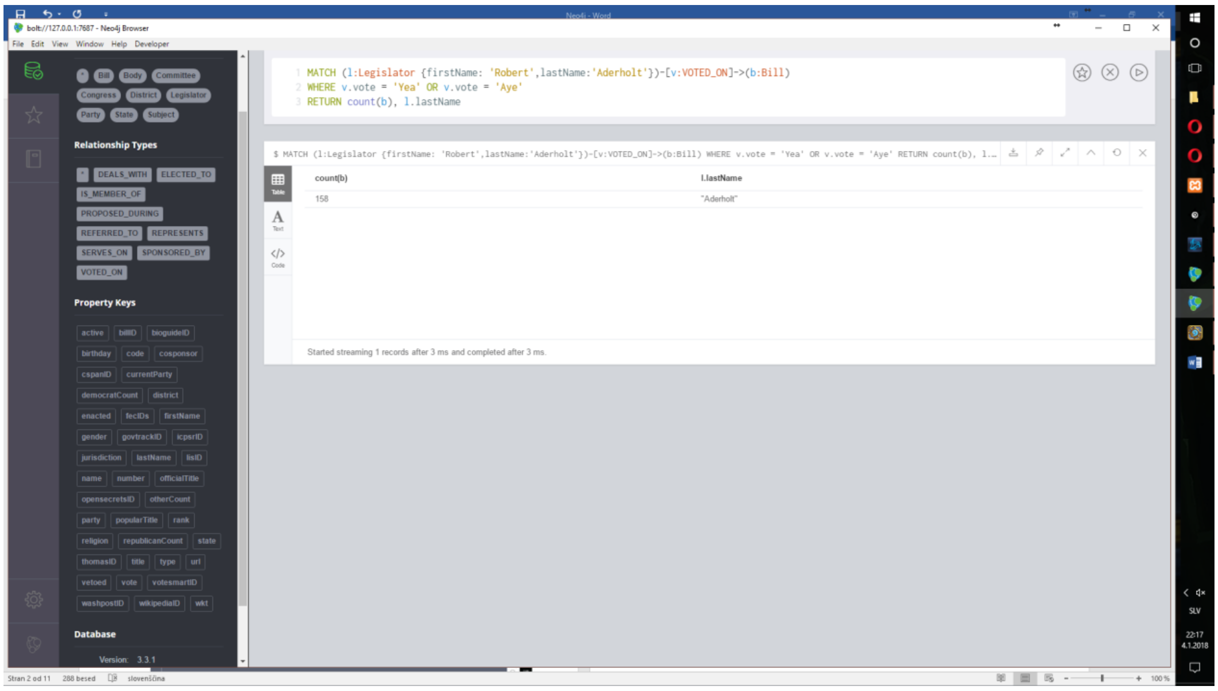
Displaying 26 nodes, 25 relationships.

3)

MATCH (l:Legislator {firstName: 'Robert',lastName:'Aderholt'})-[v:VOTED_ON]->(b:Bill)

WHERE v.vote = 'Yea' OR v.vote = 'Aye'

RETURN count(b), l.lastName


bolt://127.0.0.1:7687 - Neo4j Browser
File Edit View Window Help Developer
* Bill Body Committee Congress District Legislator Party State Subject
Relationship Types
* DEALS_WITH ELECTED_TO IS_MEMBER_OF PROPOSED_DURING REFERRED_TO REPRESENTS SERVES_ON SPONSORED_BY VOTED_ON
Property Keys
active billID bioguideID birthday code cosponsor cspanID currentParty democratCount district enacted fecIDs firstName gender govtrackID icpsrID jurisdiction lastName lisID name number officialTitle opensecretsID otherCount party popularTitle rank religion republicanCount state thomasID title type url vetoed vote votesmartID washpostID wikipediaID wkt
Database
Version: 3.3.1
Table
Text
Code
Stran 2 od 11 288 besed slovenščina

```
MATCH (l:Legislator {firstName: 'Robert',lastName:'Aderholt'})-[v:VOTED_ON]->(b:Bill)
WHERE v.vote = 'Yea' OR v.vote = 'Aye'
RETURN count(b), l.lastName
```

```
$ MATCH (l:Legislator {firstName: 'Robert',lastName:'Aderholt'})-[v:VOTED_ON]->(b:Bill) WHERE v.vote = 'Yea' OR v.vote = 'Aye' RETURN count(b), l...
```

count(b)	l.lastName
158	"Aderholt"

Started streaming 1 records after 3 ms and completed after 3 ms.

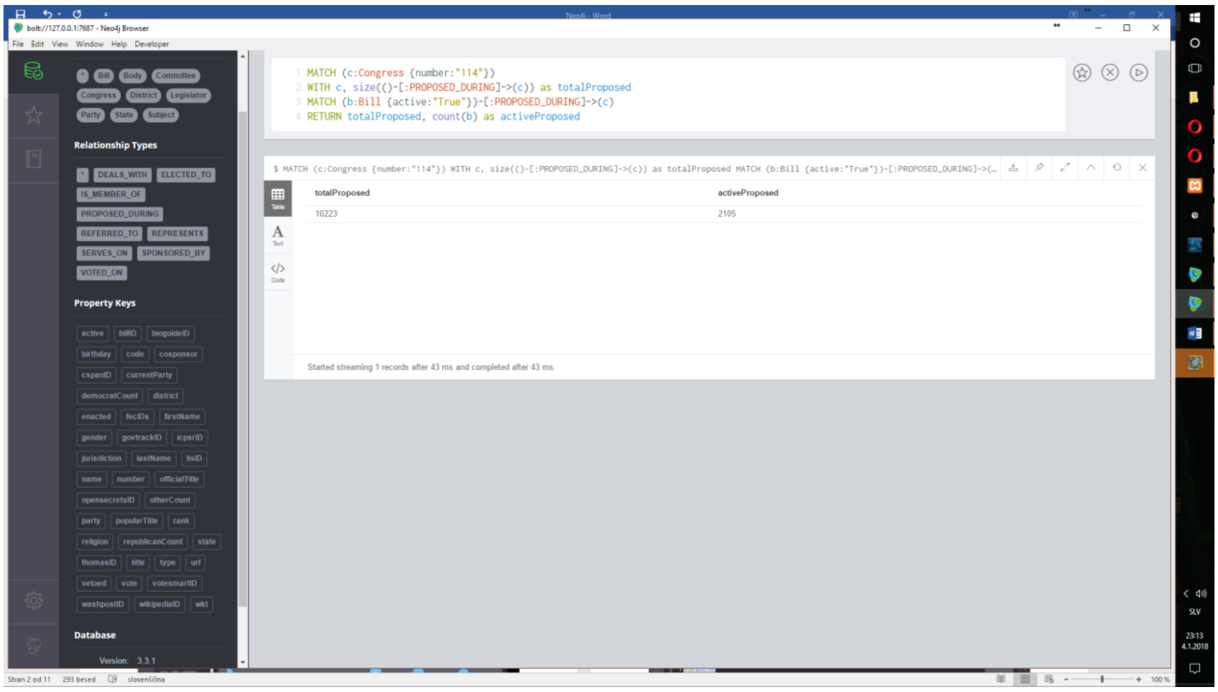
4)

MATCH (c:Congress {number:"114"})

WITH c, size(()-[:PROPOSED_DURING]->(c)) as totalProposed

MATCH (b:Bill {active:"True"})-[:PROPOSED_DURING]->(c)

RETURN totalProposed, count(b) as activeProposed



```
MATCH (c:Congress {number:"114"})
WITH c, size(()-[:PROPOSED_DURING]->(c)) as totalProposed
MATCH (b:Bill {active:"True"})-[:PROPOSED_DURING]->(c)
RETURN totalProposed, count(b) as activeProposed
```

```
$ MATCH (c:Congress {number:"114"}) WITH c, size(()-[:PROPOSED_DURING]->(c)) as totalProposed MATCH (b:Bill {active:"True"})-[:PROPOSED_DURING]->(…
```

totalProposed	activeProposed
10223	2105

Started streaming 1 records after 43 ms and completed after 43 ms.

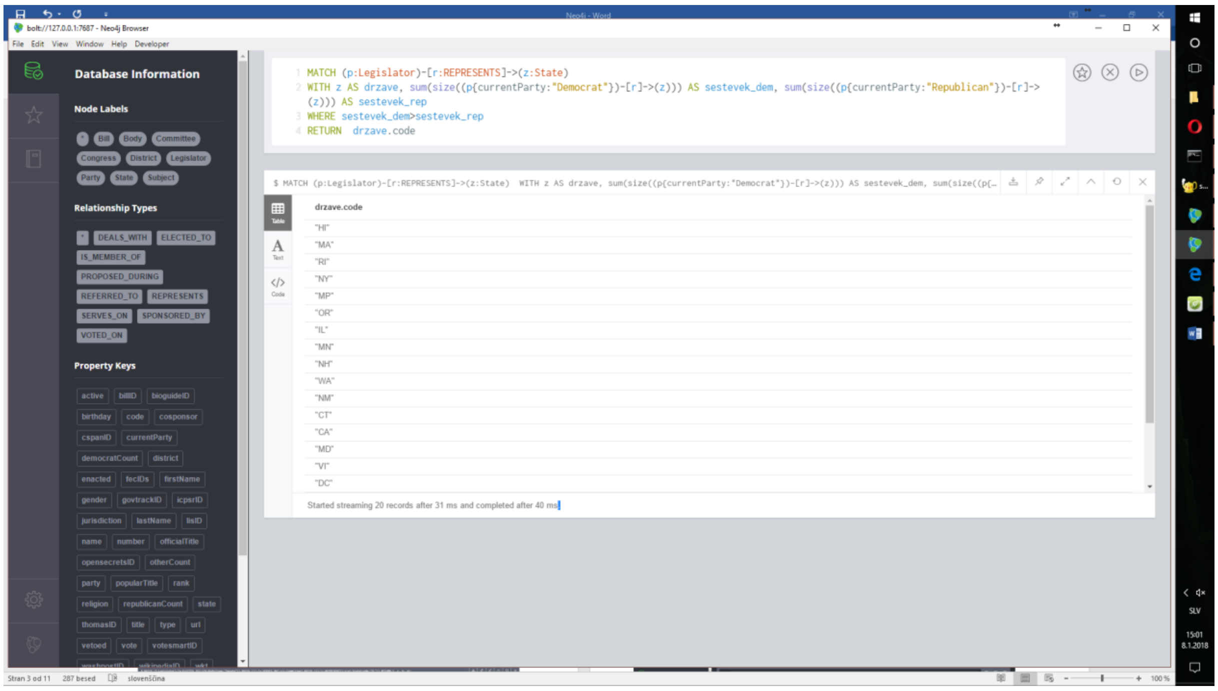
5)

MATCH (p:Legislator)-[r:REPRESENTS]->(z:State)

WITH z AS drzave, sum(size((p{currentParty:"Democrat"})-[r]->(z))) AS sestevek_dem, sum(size((p{currentParty:"Republican"})-[r]->(z))) AS sestevek_rep

WHERE sestevek_dem>sestevek_rep

RETURN drzave.code



```
MATCH (p:Legislator)-[r:REPRESENTS]->(z:State)
WITH z AS drzave, sum(size((p{currentParty:"Democrat"})-[r]->(z))) AS sestevek_dem, sum(size((p{currentParty:"Republican"})-[r]->
(z))) AS sestevek_rep
WHERE sestevek_dem>sestevek_rep
RETURN  drzave.code
```

```
drzave.code
"HI"
"MA"
"RI"
"NY"
"MP"
"OR"
"IL"
"MN"
"NH"
"WA"
"NM"
"CT"
"CA"
"MD"
"VI"
"DC"
Started streaming 20 records after 31 ms and completed after 40 ms
```

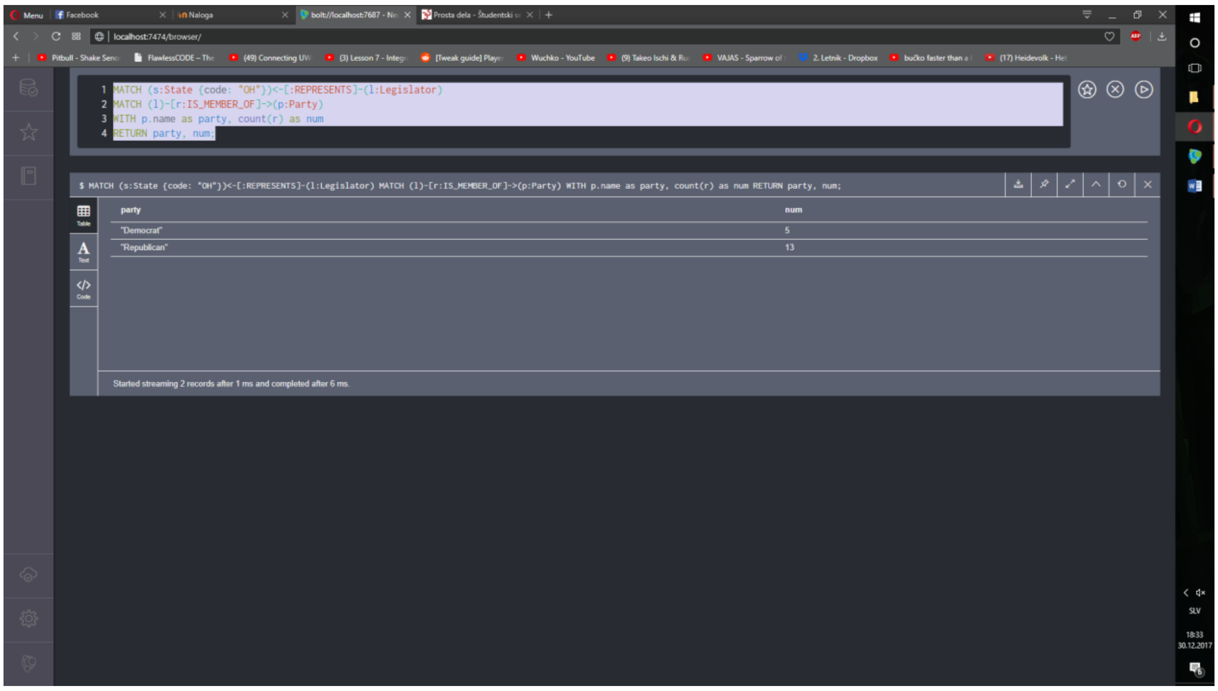
6.

MATCH (s:State {code: "OH"})<-[:REPRESENTS]-(l:Legislator)

MATCH (l)-[r:IS_MEMBER_OF]->(p:Party)

WITH p.name as party, count(r) as num

RETURN party, num;



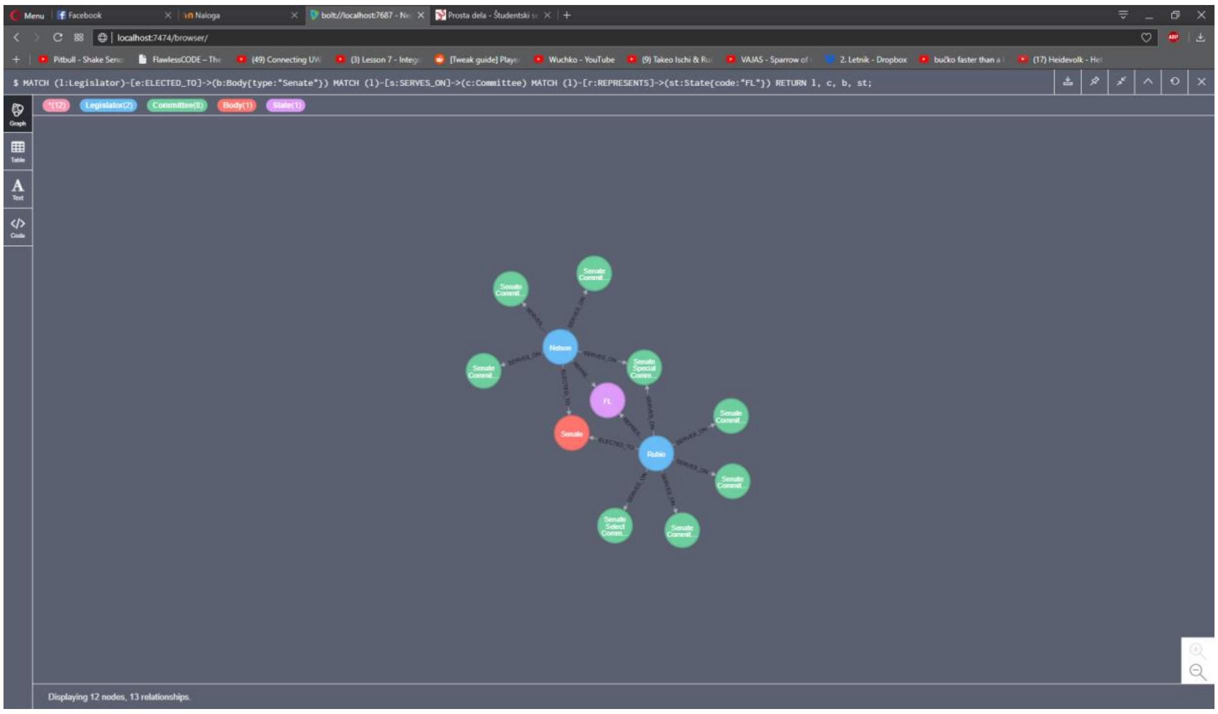
```
MATCH (s:State {code: "OH"})<-[:REPRESENTS]-(l:Legislator)
MATCH (l)-[r:IS_MEMBER_OF]->(p:Party)
WITH p.name as party, count(r) as num
RETURN party, num;
```

```
$ MATCH (s:State {code: "OH"})<-[:REPRESENTS]-(l:Legislator) MATCH (l)-[r:IS_MEMBER_OF]->(p:Party) WITH p.name as party, count(r) as num RETURN party, num;
```

party	num
"Democrat"	5
"Republican"	13

Started streaming 2 records after 1 ms and completed after 6 ms.

7.)



```
$ MATCH (l:Legislator)-[e:ELECTED_TO]->(b:Body{type:"Senate"}) MATCH (l)-[s:SERVES_ON]->(c:Committee) MATCH (l)-[r:REPRESENTS]->(st:State{code:"FL"}) RETURN l, c, b, st;
```

Displaying 12 nodes, 13 relationships.

```
MATCH (l:Legislator)-[e:ELECTED_TO]->(b:Body{type:"Senate"})

MATCH (l)-[s:SERVES_ON]->(c:Committee)

MATCH (l)-[r:REPRESENTS]->(st:State{code:"FL"})

RETURN l, c, b, st;
```

8.)

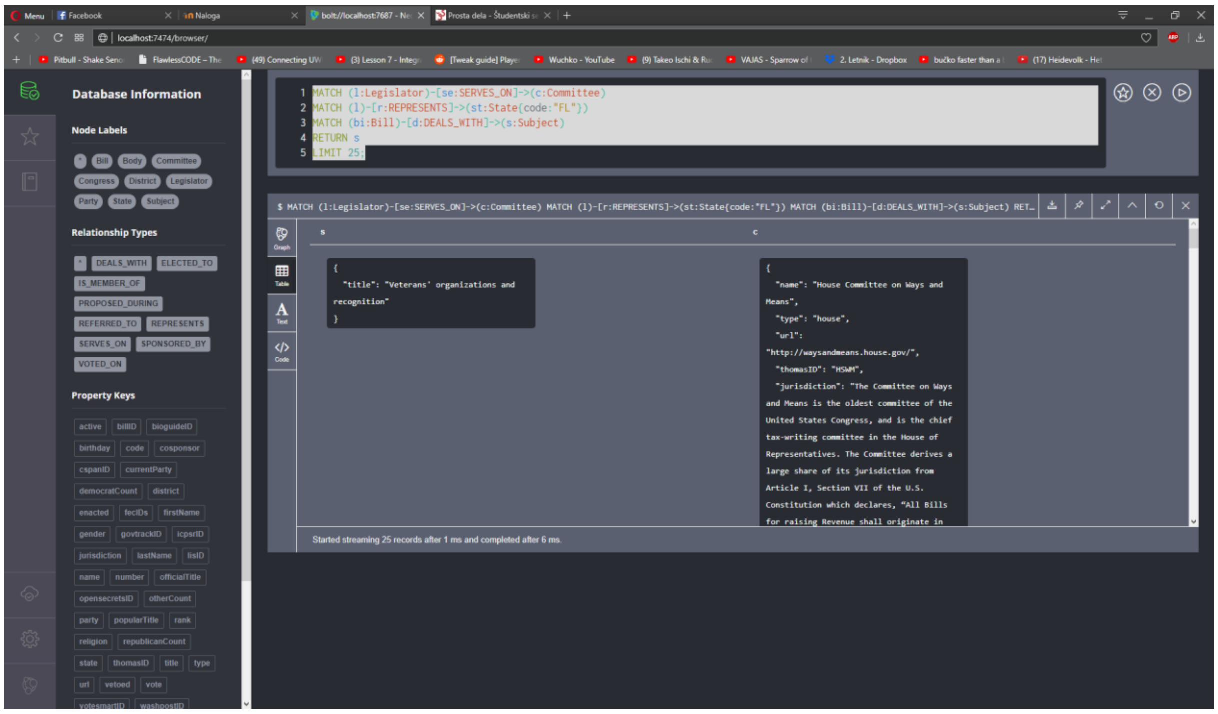
```
MATCH (l:Legislator)-[se:SERVES_ON]->(c:Committee)

MATCH (l)-[r:REPRESENTS]->(st:State{code:"FL"})

MATCH (bi:Bill)-[d:DEALS_WITH]->(s:Subject)

RETURN s

LIMIT 25;
```



```
MATCH (l:Legislator)-[se:SERVES_ON]->(c:Committee)
MATCH (l)-[r:REPRESENTS]->(st:State{code:"FL"})
MATCH (bi:Bill)-[d:DEALS_WITH]->(s:Subject)
RETURN s
LIMIT 25;
```

```
$ MATCH (l:Legislator)-[se:SERVES_ON]->(c:Committee) MATCH (l)-[r:REPRESENTS]->(st:State{code:"FL"}) MATCH (bi:Bill)-[d:DEALS_WITH]->(s:Subject) RET…
```

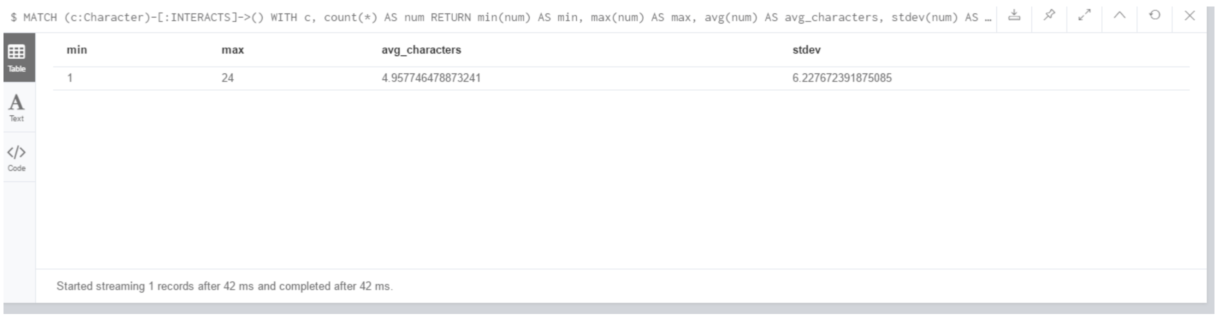
```
s
{
  "title": "Veterans' organizations and recognition"
}

c
{
  "name": "House Committee on Ways and Means",
  "type": "house",
  "url": "http://waysandmeans.house.gov/",
  "thomasID": "HSWM",
  "jurisdiction": "The Committee on Ways and Means is the oldest committee of the United States Congress, and is the chief tax-writing committee in the House of Representatives. The Committee derives a large share of its jurisdiction from Article I, Section VII of the U.S. Constitution which declares, "All Bills for raising Revenue shall originate in
```

Started streaming 25 records after 1 ms and completed after 6 ms.

GAME OF THRONES

1)
```
MATCH (c:Character)-[:INTERACTS]->()
WITH c, count(*) AS num
RETURN min(num) AS min, max(num) AS max, avg(num) AS
avg_characters, stdev(num) AS stdev
```

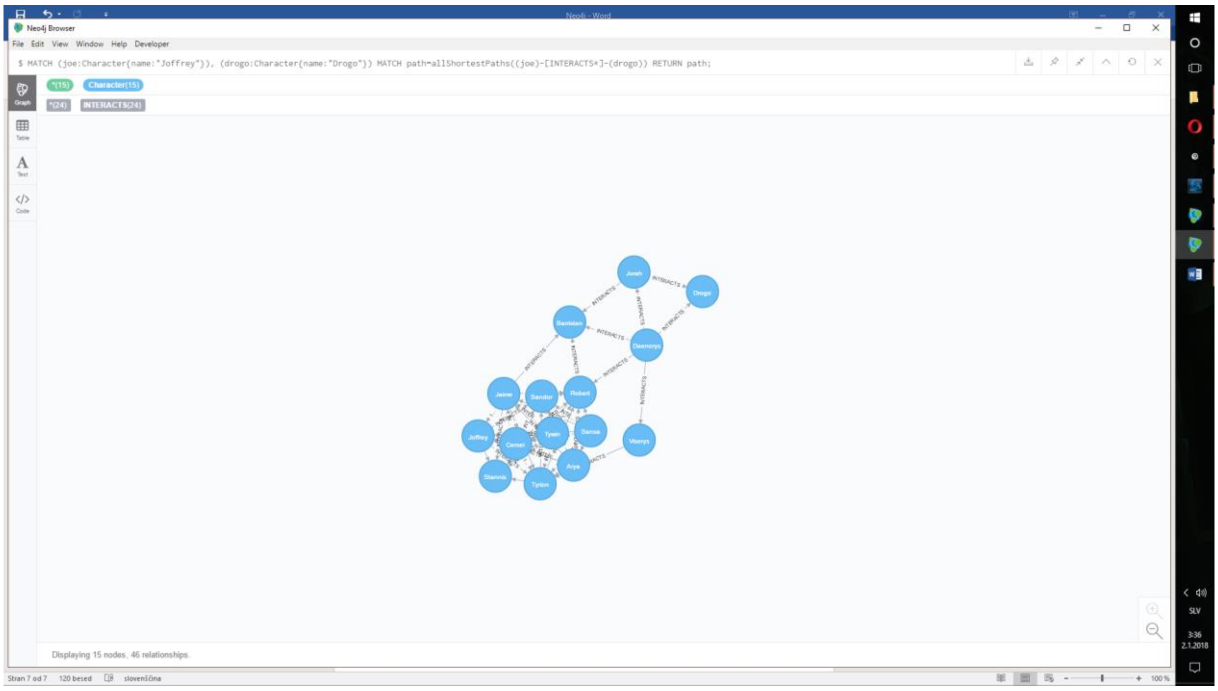


```
$ MATCH (c:Character)-[:INTERACTS]->() WITH c, count(*) AS num RETURN min(num) AS min, max(num) AS max, avg(num) AS avg_characters, stdev(num) AS …
```

min	max	avg_characters	stdev
1	24	4.957746478873241	6.227672391875085

Started streaming 1 records after 42 ms and completed after 42 ms.

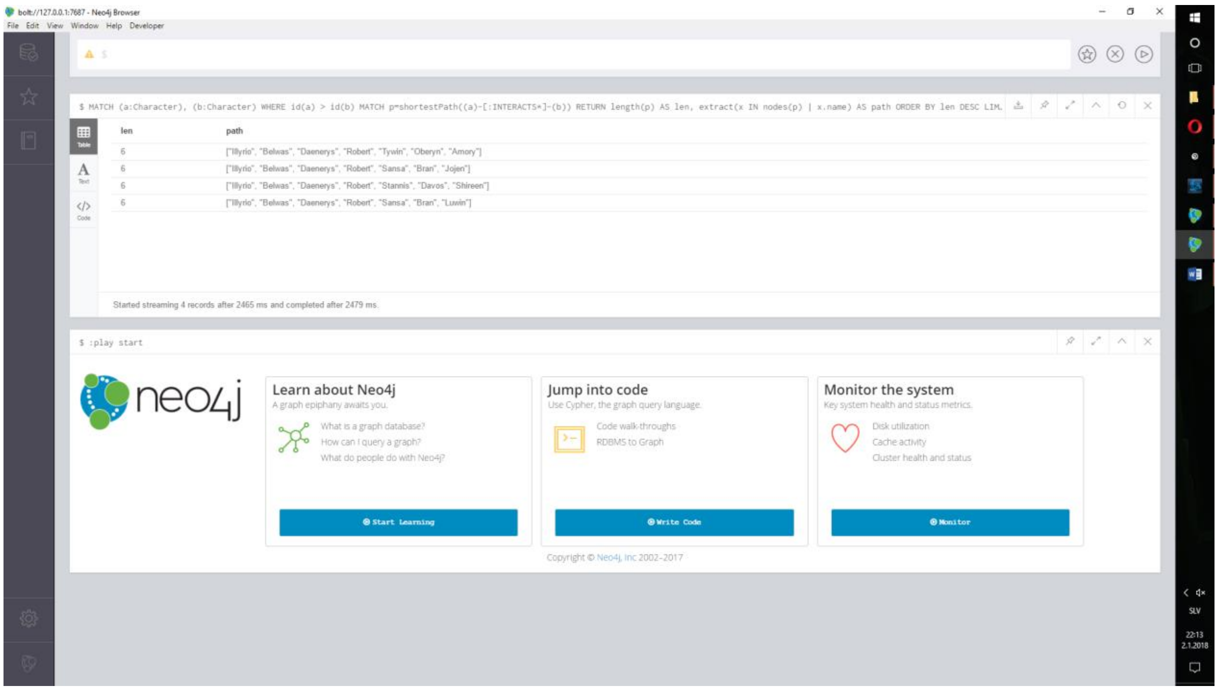
2)
```
MATCH (joe:Character{name:"Joffrey"}), (drogo:Character{name:"Drogo"})
MATCH path=allShortestPaths((joe)-[INTERACTS*]-(drogo))
RETURN path;
```



```
$ MATCH (joe:Character{name:"Joffrey"}), (drogo:Character{name:"Drogo"}) MATCH path=allShortestPaths((joe)-[INTERACTS*]-(drogo)) RETURN path;
```

Displaying 15 nodes, 46 relationships.

3.
```
MATCH (a:Character), (b:Character) WHERE id(a) > id(b)
MATCH p=shortestPath((a)-[:INTERACTS*]-(b))
RETURN length(p) AS len, extract(x IN nodes(p) | x.name) AS path
ORDER BY len DESC LIMIT 4
```



```
$ MATCH (a:Character), (b:Character) WHERE id(a) > id(b) MATCH p=shortestPath((a)-[:INTERACTS*]-(b)) RETURN length(p) AS len, extract(x IN nodes(p) | x.name) AS path ORDER BY len DESC LIM
```

len	path
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Tywin", "Oberyn", "Amory"]
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Sansa", "Bran", "Jojen"]
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Stannis", "Davos", "Shireen"]
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Sansa", "Bran", "Luwin"]

Started streaming 4 records after 2465 ms and completed after 2479 ms.

```
$ :play start
```

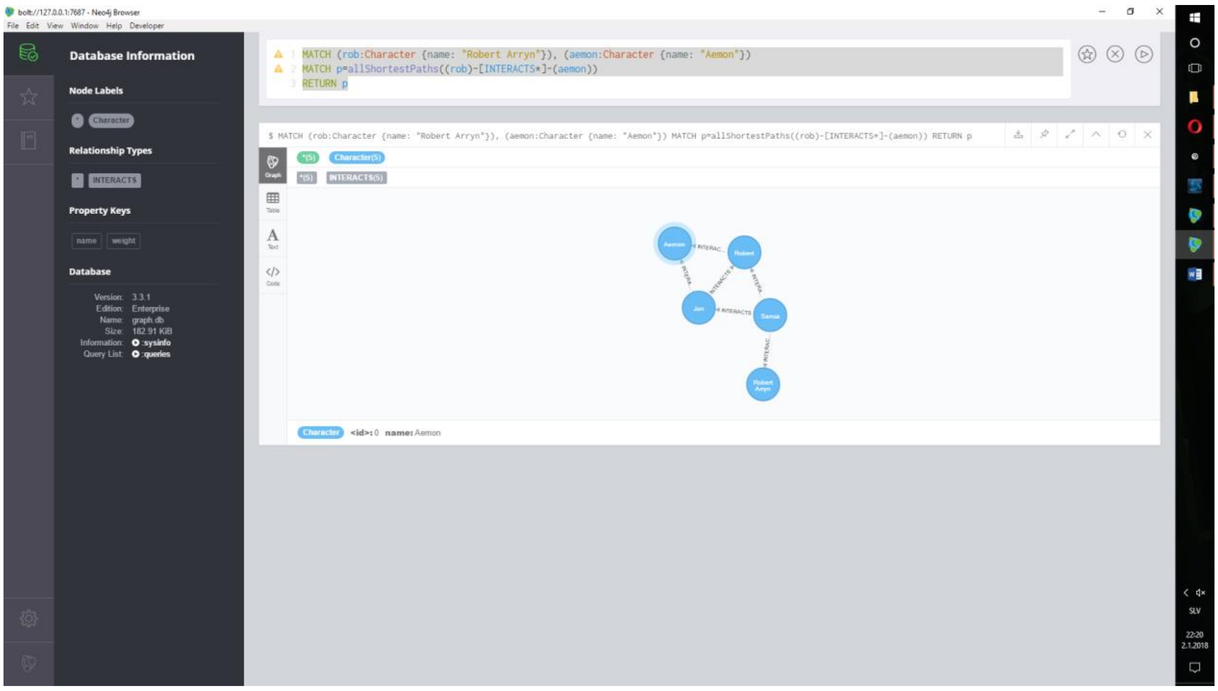
bolt://127.0.0.1:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor
Copyright © Neo4j, Inc 2002–2017
22:13
2.1.2018

4.

MATCH (rob:Character {name: "Robert Arryn"}), (aemon:Character {name: "Aemon"})

MATCH p=allShortestPaths((rob)-[INTERACTS*]-(aemon))

RETURN p

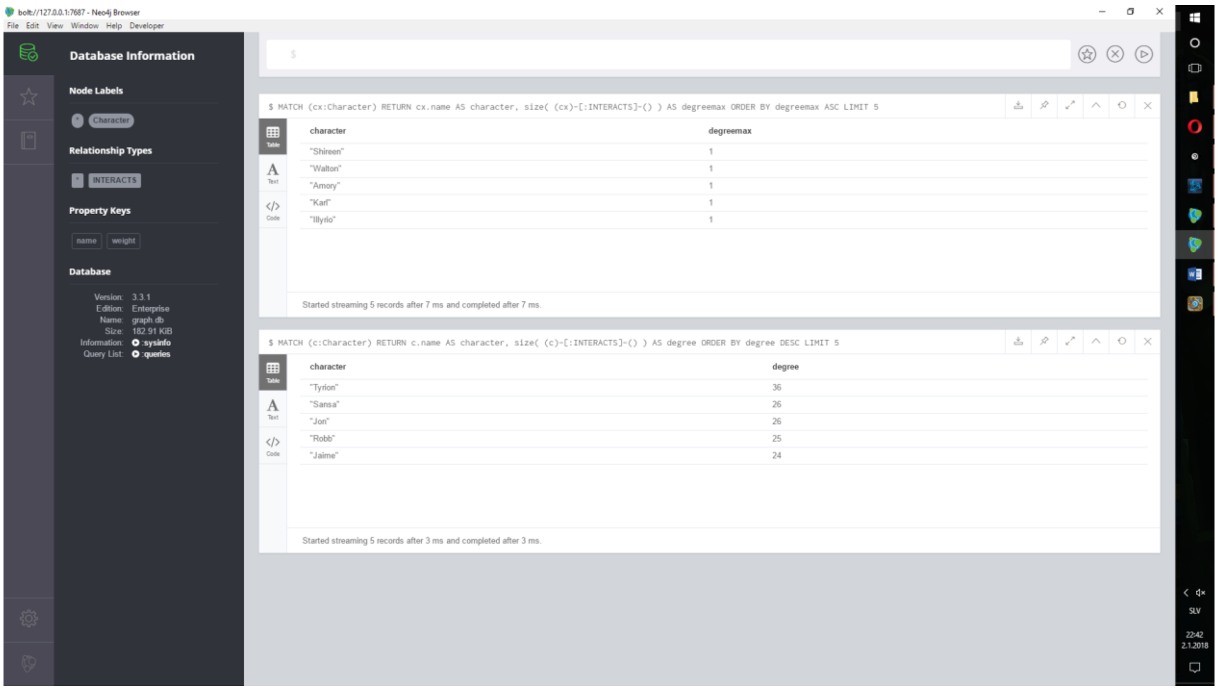


```
MATCH (rob:Character {name: "Robert Arryn"}), (aemon:Character {name: "Aemon"})
MATCH p=allShortestPaths((rob)-[INTERACTS*]-(aemon))
RETURN p
```

```
$ MATCH (rob:Character {name: "Robert Arryn"}), (aemon:Character {name: "Aemon"}) MATCH p=allShortestPaths((rob)-[INTERACTS*]-(aemon)) RETURN p
```

Character <id>: 0 name: Aemon

5.



```
$ MATCH (cx:Character) RETURN cx.name AS character, size( (cx)-[:INTERACTS]-() ) AS degreemax ORDER BY degreemax ASC LIMIT 5
```

character	degreemax
"Shireen"	1
"Walton"	1
"Amory"	1
"Karl"	1
"Illyrio"	1

Started streaming 5 records after 7 ms and completed after 7 ms.

```
$ MATCH (c:Character) RETURN c.name AS character, size( (c)-[:INTERACTS]-() ) AS degree ORDER BY degree DESC LIMIT 5
```

character	degree
"Tyrion"	36
"Sansa"	26
"Jon"	26
"Robb"	25
"Jaime"	24

Started streaming 5 records after 3 ms and completed after 3 ms.

Database Information

Node Labels: Character

Relationship Types: INTERACTS

Property Keys: name, weight

Database

Version: 3.3.1
Edition: Enterprise
Name: graph.db
Size: 182.91 KiB
Information: :sysinfo
Query List: :queries

bolt://127.0.0.1:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code
SLV
22:42
2.1.2018

MATCH (cx:Character)

RETURN cx.name AS character, size((cx)-[:INTERACTS]-()) AS degreemax ORDER BY degreemax ASC LIMIT 5

MATCH (c:Character)

RETURN c.name AS character, size((c)-[:INTERACTS]-()) AS degree ORDER BY degree DESC LIMIT 5

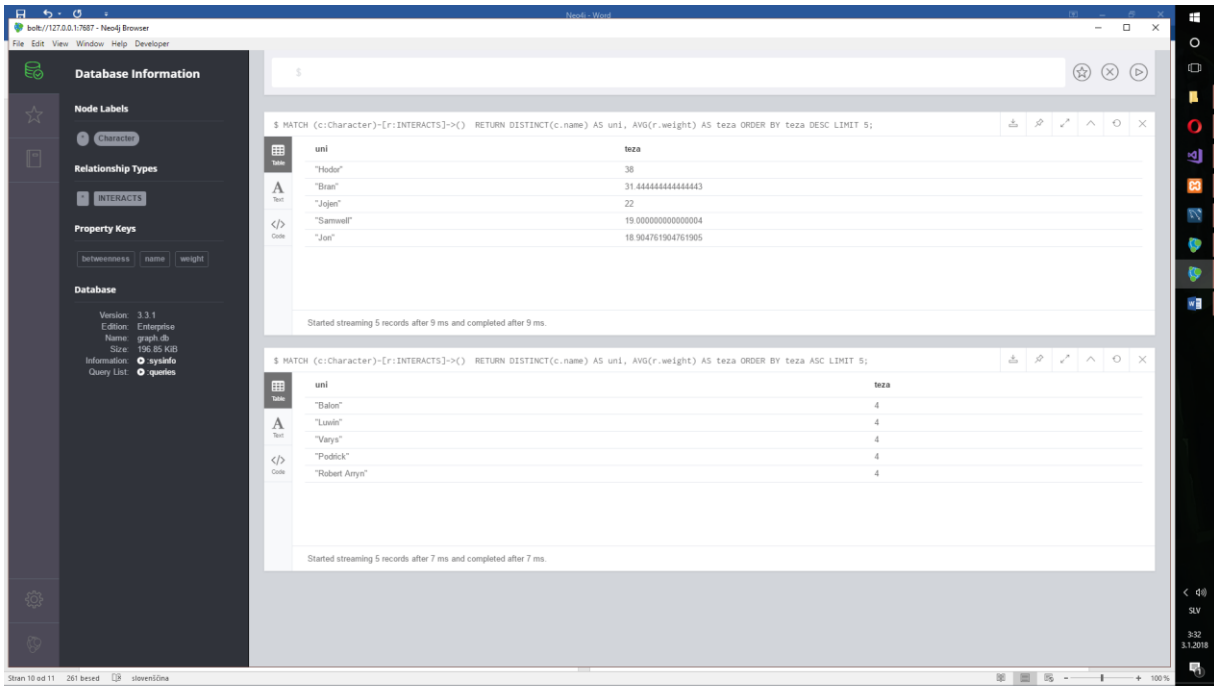
6.

LAHKO TUDI MATCH (C:CHARACTER)-[R:INTERACTS]-(), RAZLIČNI REZULTATI!

MATCH (c:Character)-[r:INTERACTS]->()

RETURN DISTINCT(c.name) AS peder, AVG(r.weight) AS teza

ORDER BY teza ASC LIMIT 5;



Database

Version:	3.3.1
Edition:	Enterprise
Name:	graph.db
Size:	196.85 KiB
Information:	:sysinfo
Query List:	:queries

```
$ MATCH (c:Character)-[r:INTERACTS]->()  RETURN DISTINCT(c.name) AS uni, AVG(r.weight) AS teza ORDER BY teza DESC LIMIT 5;
```

uni	teza
"Hodor"	38
"Bran"	31.444444444444443
"Jojen"	22
"Samwell"	19.000000000000004
"Jon"	18.904761904761905

Started streaming 5 records after 9 ms and completed after 9 ms.

```
$ MATCH (c:Character)-[r:INTERACTS]->()  RETURN DISTINCT(c.name) AS uni, AVG(r.weight) AS teza ORDER BY teza ASC LIMIT 5;
```

uni	teza
"Balon"	4
"Luwin"	4
"Varys"	4
"Podrick"	4
"Robert Arryn"	4

Started streaming 5 records after 7 ms and completed after 7 ms.

MATCH (c:Character)-[r:INTERACTS]->()

RETURN DISTINCT(c.name) AS uni, AVG(r.weight) AS teza

ORDER BY teza DESC LIMIT 5;

BETWEEN

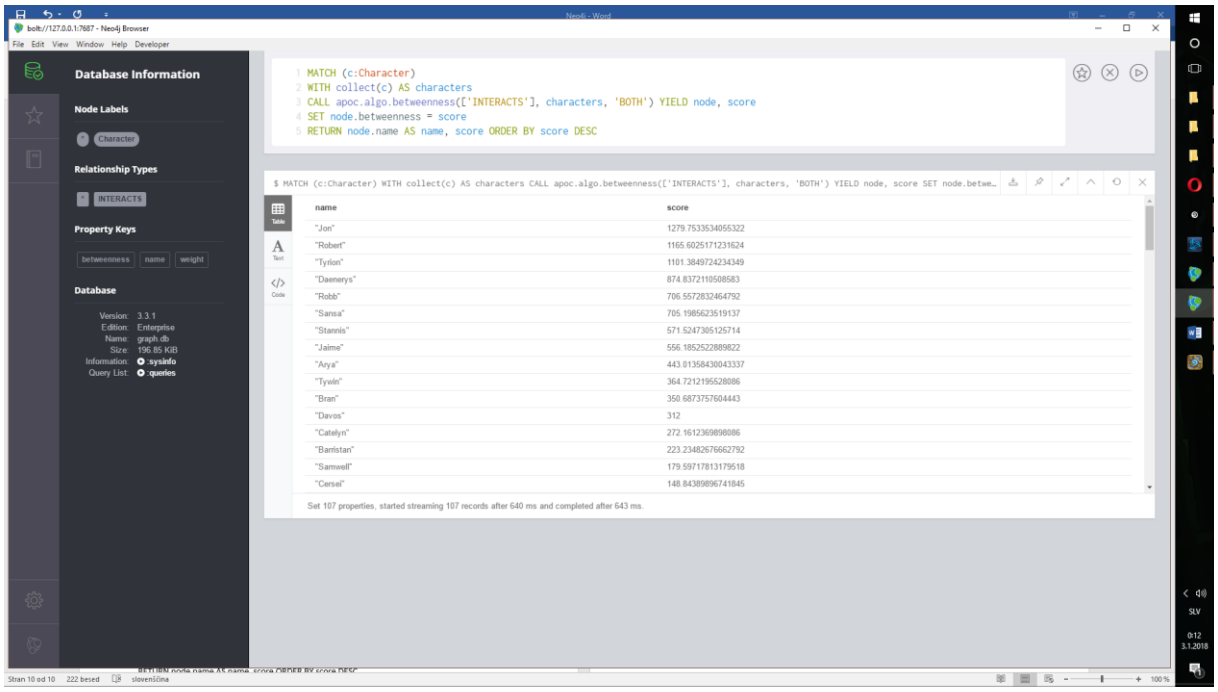
MATCH (c:Character)

WITH collect(c) AS characters

CALL apoc.algo.betweenness(['INTERACTS'], characters, 'BOTH') YIELD node, score

SET node.betweenness = score

RETURN node.name AS name, score ORDER BY score DESC



```
MATCH (c:Character)
WITH collect(c) AS characters
CALL apoc.algo.betweenness(['INTERACTS'], characters, 'BOTH') YIELD node, score
SET node.betweenness = score
RETURN node.name AS name, score ORDER BY score DESC
```

name	score
"Jon"	1279.7533534055322
"Robert"	1165.6025171231624
"Tyrion"	1101.3849724234349
"Daenerys"	874.8372110508583
"Robb"	706.5572832464792
"Sansa"	705.1985623519137
"Stannis"	571.5247305125714
"Jaime"	556.1852522889822
"Arya"	443.01358430043337
"Tywin"	364.7212195528086
"Bran"	350.6873757604443
"Davos"	312
"Catelyn"	272.1612369898086
"Barristan"	223.23482676662792
"Samwell"	179.59717813179518
"Cersei"	148.84389896741845

Set 107 properties, started streaming 107 records after 640 ms and completed after 643 ms.

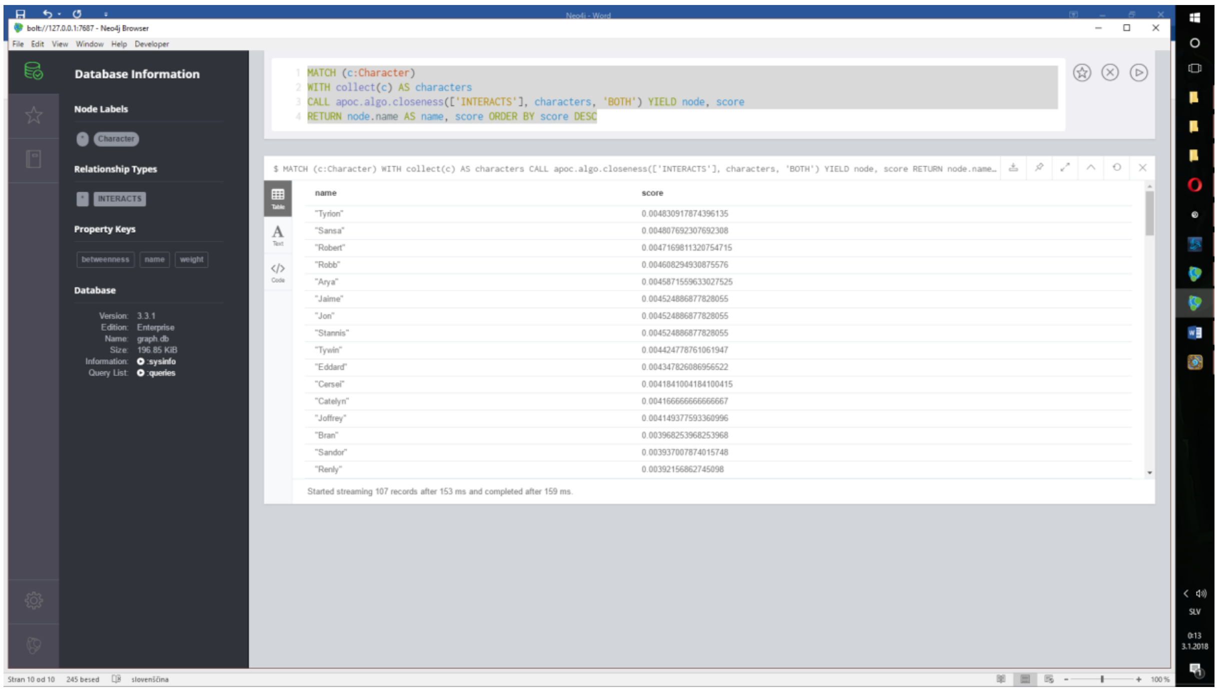
CLOSENESS

MATCH (c:Character)

WITH collect(c) AS characters

CALL apoc.algo.closeness(['INTERACTS'], characters, 'BOTH') YIELD node, score

RETURN node.name AS name, score ORDER BY score DESC



```
MATCH (c:Character)
WITH collect(c) AS characters
CALL apoc.algo.closeness(['INTERACTS'], characters, 'BOTH') YIELD node, score
RETURN node.name AS name, score ORDER BY score DESC
```

name	score
"Tyrion"	0.004830917874396135
"Sansa"	0.004807692307692308
"Robert"	0.0047169811320754715
"Robb"	0.004608294930875576
"Arya"	0.0045871559633027525
"Jaime"	0.004524886877828055
"Jon"	0.004524886877828055
"Stannis"	0.004524886877828055
"Tywin"	0.004424778761061947
"Eddard"	0.004347826086956522
"Cersei"	0.0041841004184100415
"Catelyn"	0.004166666666666667
"Joffrey"	0.004149377593360996
"Bran"	0.003968253968253968
"Sandor"	0.003937007874015748
"Renly"	0.00392156862745098

Started streaming 107 records after 153 ms and completed after 159 ms.