# Data Centric Web Applications

## Lab 7 Neo4j Relationships

### Part 1

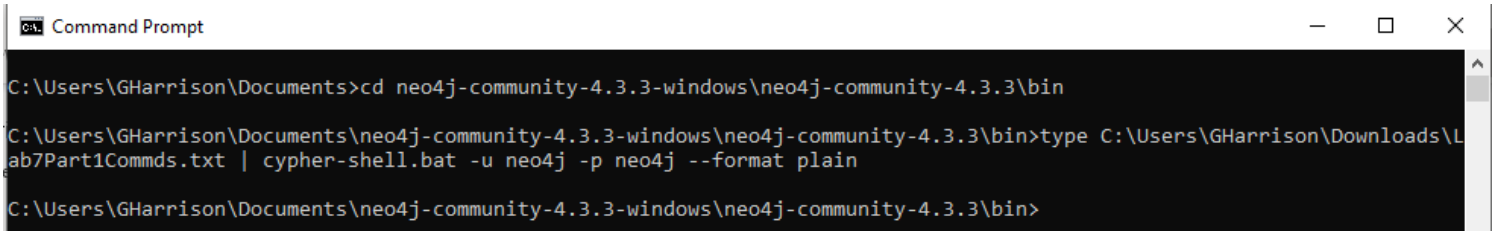
- Get *Lab7Part1Commds.txt* from and run the following command:

```
type Path_to_Lab7Part1Commands.txt | cypher-shell.bat -u neo4j -p neo4j --format plain
```

from the `bin` folder of your Neo4j installation.
E.g. Assuming:

- *Lab7Part1Commds.txt* was downloaded to `C:\Users\GHarrison\Downloads`
- Neo4j installation is at `C:\Users\GHarrison\Documents\neo4j-community-4.3.3-windows\neo4j-community-4.3.3\`
- Neo4j username is `neo4j`
- Neo4j password is `neo4j`

The following should be run, and if no errors are reported the database will be set up.
NOTE: This will delete everything from your current database (as specified in `neo4j.conf`).




- There will be 5 `COUNTY`, 11 `TOWN`, and 15 `PERSON` nodes in the database.

- Create the following relationships (some of which have a property called `countyTown`) between the nodes specified below:

| TOWN | RELATIONSHIP | COUNTY |
|---|---|---|
| Galway | PART_OF {countyTown:true} | Galway |
| Tuam | PART_OF | Galway |
| Clifden | PART_OF | Galway |
| Carrick-on-Shannon | PART_OF {countyTown:true} | Leitrim |
| Manorhamilton | PART_OF | Leitrim |
| Castlebar | PART_OF {countyTown:true} | Mayo |
| Ballina | PART_OF | Mayo |
| Roscommon | PART_OF {countyTown:true} | Roscommon |
| Castlerea | PART_OF | Rocommon |
| Sligo | PART_OF {countyTown:true} | Sligo |
| Collooney | PART_OF | Sligo |

- Create the following relationships (some of which have a property called `since`) between the nodes specified below:

| PERSON | RELATIONSHIP | TOWN |
|---|---|---|
| Tom | LIVES_IN | Galway |
| Sean | LIVES_IN since:2010 | Galway |
| Bob | LIVES_IN | Galway |
| Mary | LIVES_IN since:2018 | Clifden |
| Alice | LIVES_IN since:2010 | Clifden |
| Pat | LIVES_IN since:1959 | Carrick-on-Shannon |
| Alan | LIVES_IN | Carrick-on-Shannon |
| Bill | LIVES_IN | Manorhamilton |
| Yvonne | LIVES_IN | Castlebar |
| Walter | LIVES_IN | Ballina |
| Colin | LIVES_IN | Roscommon |
| Brendan | LIVES_IN since:2013 | Castlerea |
| Susan | LIVES_IN | Castlerea |
| Lucy | LIVES_IN | Sligo |
| Michael | LIVES_IN | Sligo |

- Show the PERSONs who live in Galway TOWN.

```
MATCH(p:PERSON)-[r:LIVES_IN]->(q:TOWN{name:"Galway"}
```

- Show the age of the oldest PERSONs who lives in Carrick-on-Shannon.

```
MATCH(p:PERSON)-[r:LIVES_IN]->(q:TOWN{name:"Carrick-on-Shannon
```

- Show the average age of males who live in Roscommon COUNTY.

```
MATCH(p:PERSON{sex:"M"})-[r:LIVES_IN]->(q:TOWN)-[:PART_OF]->(s:COUNTY{name:"Roscommon
```

- Show the number of males who live in Galway COUNTY.

```
MATCH(p:PERSON{sex:"M"})-[r:LIVES_IN]->(q:TOWN)-[:PART_OF]->(s:COUNTY{name:"Galway"
```

- Show the name and population of the COUNTY where Lucy lives.

```
MATCH(p:PERSON{name:"Lucy"})-[:LIVES_IN]->()-[:PART_OF]->(q:COUNTY
```

- Show the COUNTY name , TOWN name and PERSON name where the person has lived in the town since the year 2010.

```
MATCH(p:PERSON)-[r:LIVES_IN]->(q:TOWN)-[:PART_OF]->(s:COUNTY)WHEREr.since>201
```

- Show the COUNTY name and the TOWN name of all towns with a population of less than 5000.

```
MATCH(p:TOWN)-[:PART_OF]->(q:COUNTY)WHEREp.pop<5000
```

- For people living in towns since 2011 or later, show the person's name (as *Name*), how long they've been living in the town (as *Since*), and the name of the town (as *Town*), in chronological order.

```
MATCH(p:PERSON)-[r:LIVES_IN]->(q:TOWN)WHEREr.since>2010RETUF
```

- Show the total population of the towns in county Galway (as *County_Galway_Pop*),

```
MATCH(p:TOWN)-[r:PART_OF]->(q:COUNTY{name:"Galway"
```

- Show the county name (as *County*), the towns in the county (as *Towns*) and the number of towns in the county (as *Num_Towns*).
  E.g.

| "County" | "Towns" | "Num_Towns" |
|---|---|---|
| "Galway" | ["Clifden","Tuam","Galway"] | 3 |

```
MATCH(p:TOWN)-[r:PART_OF]->(q:COUNTY{name:"Galway"})RETURNq.nameASCoun
```

## Part 2

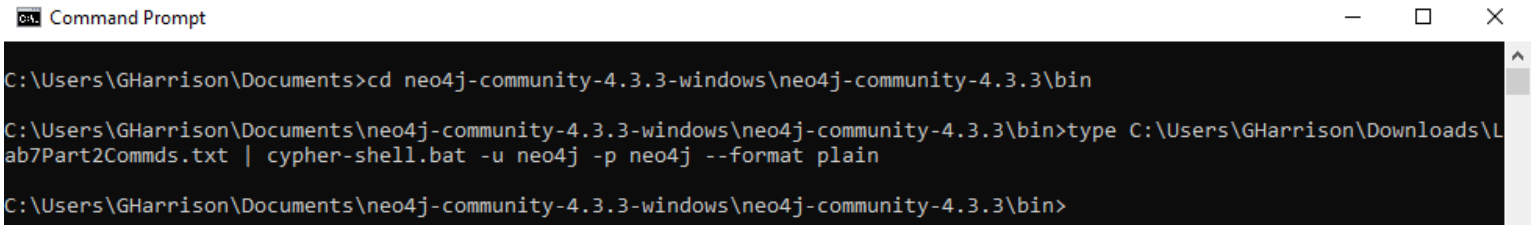
- Get *Lab7Part2Commds.txt* from and run the following command:

```
type Path_to_Lab7Part2Commands.txt | cypher-shell.bat -u neo4j -p neo4j --format plain
```

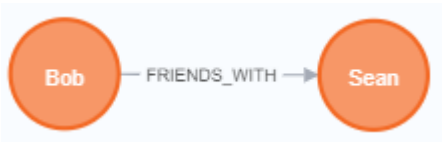
from the `bin` folder of your Neo4j installation.
E.g. Assuming:

- *Lab7Part2Commds.txt* was downloaded to `C:\Users\GHarrison\Downloads`
- Neo4j installation is at `C:\Users\GHarrison\Documents\neo4j-community-4.3.3-windows\neo4j-community-4.3.3\`
- Neo4j username is `neo4j`
- Neo4j password is `neo4j`

The following should be run, and if no errors are reported the database will be set up.
NOTE: This will delete everything from your current database (as specified in `neo4j.conf`).




**NOTE:** In this database, the FRIENDS_WITH relationship can be read in either direction. In this example, “Bob” is FRIENDS_WITH “Sean”, however it can be taken that “Sean” is also FRIENDS_WITH “Bob”.



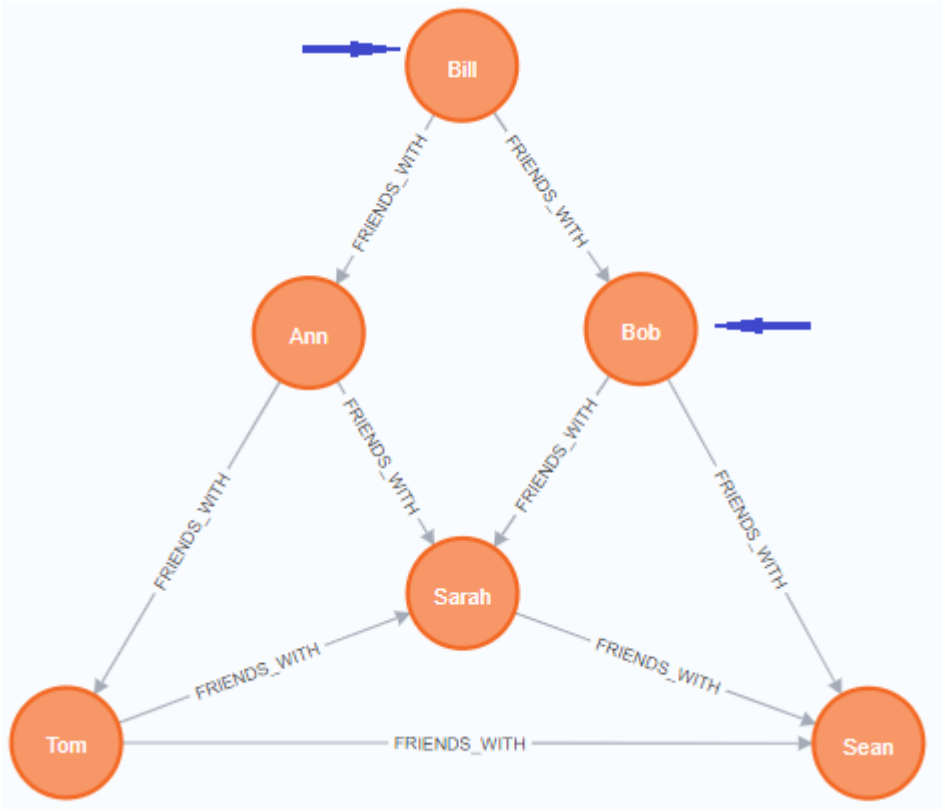
- Show the names of Bill's hobbies.

```
MATCH(n:Person{name:"Bill"})-[o:LIKES]->(p:Hobby)
```

- Show the names of hobbies people who live in Galway like (as *Galway_Hobbies*) in alphabetical order.

```
MATCH(m:County{name:"Galway"})<-[:LIVES_IN]-(n:Person)-[:LIKES]->(p:Hobby
```

- Show all friends-of-friends of Tom.
  A friend-of-a-friend (FOAF) is someone whom your friend is friends with, but not you. In the example below, Bill and Bob are FOAFs of Tom.
  Ann is a friend of Tom, and her friend is Sarah. So, Sarah would be a FOAF of Tom, but as Tom is already friends with her, she's not FOAF of Tom. Similarly, for Ann and Sean.




```
MATCH(p:Person{name:"Tom"})-[:FRIENDS_WITH*2]-(o
```

- Show the unique hobbies that people who live in Westmeath like (as *Westmeath_Hobbies*).

```
MATCH(c:County{name:"Westmeath"})<-[:LIVES_IN]-(p:Person)-[:LIKES]->(h:Hobb
```

- Show the number of people who like relaxation hobbies (as *Relaxation*).

```
MATCH(p:Person)-[:LIKES]->(h:Hobby{type:"relaxation"
```

- Show a heading called *Likes_Basketball* that returns true if Sarah LIKES basketball, or false if Sarah doesn't like basketball.

```
MATCH(p:Person{name:"Sarah"})-[:LIKES]->(h:Hobby{name:"Basketball
```