COMS W4111-002/V002 (Spring 2023) Introduction to Databases

</span>

Homework 3b: MongoDB and Neo4j Common Tasks

Overview

- There are three parts to HW 3:
    - `3a` : Written questions
    - `3b:` A set of common practical tasks.
    - `3c:` Programming and Non-Programming track specific tasks.
- This notebook explains part `3b,` which is a prerequisite for part `3c.`
- In this task, you will process files to load data into both Neo4j and MongoDB in the cloud.
- This builds on the previous notebook's setup tasks that we published.

Data Loading

Read Files

- The project data a data directory. The directory contains JSON data.
- The next few cells read the JSON information and load into Python lists.

In [1]:

```
import json
```

In [2]:

```
def load_json(fn):
    result = None
    with open(fn, "r") as in_file:
        result = json.load(in_file)
    return result
```

In [3]:

```
characters = load_json("./data/characters.json")
```

In [4]:

```
characters[0:2]
```

Out[4]:

```
[{'characterName': 'Addam Marbrand',
  'characterLink': '/character/ch0305333/',
  'actorName': 'B.J. Hogg',
  'actorLink': '/name/nm0389698/'},
 {'characterName': 'Aegon Targaryen',
  'houseName': 'Targaryen',
  'royal': True,
  'parents': ['Elia Martell', 'Rhaegar Targaryen'],
  'siblings': ['Rhaenys Targaryen', 'Jon Snow'],
  'killedBy': ['Gregor Clegane']}]
```

In [5]:

```
episodes = load_json("./data/episodes.json")
```

In [6]:

```
episodes[0:1]
```

Out[6]:

```
[{'seasonNum': 1,
  'episodeNum': 1,
  'episodeTitle': 'Winter Is Coming',
  'episodeLink': '/title/tt1480055/',
  'episodeAirDate': '2011-04-17',
  'episodeDescription': "Jon Arryn, the Hand of the King, is dead. King Robert Baratheon
plans to ask his oldest friend, Eddard Stark, to take Jon's place. Across the sea, Viser
ys Targaryen plans to wed his sister to a nomadic warlord in exchange for an army.",
  'openingSequenceLocations': ["King's Landing",
   'Winterfell',
   'The Wall',
   'Pentos'],
  'scenes': [{'sceneStart': '0:00:40',
    'sceneEnd': '0:01:45',
    'location': 'The Wall',
    'subLocation': 'Castle Black',
    'characters': [{'name': 'Gared'},
     {'name': 'Waymar Royce'},
     {'name': 'Will'}]},
   {'sceneStart': '0:01:45',
    'sceneEnd': '0:03:24',
    'location': 'North of the Wall',
    'subLocation': 'The Haunted Forest',
    'characters': [{'name': 'Gared'},
     {'name': 'Waymar Royce'},
     {'name': 'Will'}]},
   {'sceneStart': '0:03:24',
    'sceneEnd': '0:03:31',
    'location': 'North of the Wall',
    'subLocation': 'The Haunted Forest',
    'characters': [{'name': 'Will'},
     {'name': 'Wight Wildling Girl', 'alive': False}]},
   {'sceneStart': '0:03:31',
    'sceneEnd': '0:03:38',
    'location': 'North of the Wall',
    'subLocation': 'The Haunted Forest',
    'characters': [{'name': 'Will'}]},
   {'sceneStart': '0:03:38',
    'sceneEnd': '0:03:44',
    'location': 'North of the Wall',
    'subLocation': 'The Haunted Forest',
 'characters': []},
{'sceneStart': '0:03:44',
 'sceneEnd': '0:05:36',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Gared'},
  {'name': 'Waymar Royce'},
  {'name': 'Will'}]},
{'sceneStart': '0:05:36',
 'sceneEnd': '0:05:41',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Will'}]},
{'sceneStart': '0:05:41',
 'sceneEnd': '0:05:48',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Gared'}, {'name': 'Waymar Royce'}]},
{'sceneStart': '0:05:48',
 'sceneEnd': '0:05:58',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Will'}]},
{'sceneStart': '0:05:58',
 'sceneEnd': '0:06:21',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Gared'},
  {'name': 'Waymar Royce',
   'alive': False,
   'mannerOfDeath': 'Back stab',
   'killedBy': ['White Walker']},
  {'name': 'White Walker'}]},
{'sceneStart': '0:06:21',
 'sceneEnd': '0:06:39',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Will'}]},
{'sceneStart': '0:06:39',
 'sceneEnd': '0:06:49',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Will'}, {'name': 'Wight Wildling Girl'}]},
{'sceneStart': '0:06:49',
 'sceneEnd': '0:07:45',
 'location': 'North of the Wall',
 'subLocation': 'The Haunted Forest',
 'characters': [{'name': 'Will'},
  {'name': 'Gared',
   'alive': False,
   'mannerOfDeath': 'Decapitation',
   'killedBy': ['White Walker']},
  {'name': 'White Walker'}]},
{'sceneStart': '0:09:27',
 'sceneEnd': '0:12:38',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Will'},
  {'name': 'Jon Snow'},
  {'name': 'Bran Stark'},
  {'name': 'Robb Stark'},
  {'name': 'Eddard Stark'},
  {'name': 'Catelyn Stark'},
  {'name': 'Rickon Stark'},
  {'name': 'Sansa Stark'},
  {'name': 'Arya Stark'},
  {'name': 'Theon Greyjoy'},
  {'name': 'Rodrik Cassel'},
  {'name': 'Septa Mordane'}]},
{'sceneStart': '0:12:38',
 'sceneEnd': '0:15:41',
 'location': 'The North',
 'subLocation': 'Outside Winterfell',
 'characters': [{'name': 'Will',
   'alive': False,
   'mannerOfDeath': 'Decapitation',
   'killedBy': ['Eddard Stark']},
  {'name': 'Eddard Stark', 'weapon': [{'action': 'has', 'name': 'Ice'}]},
  {'name': 'Robb Stark'},
  {'name': 'Theon Greyjoy'},
  {'name': 'Bran Stark'},
  {'name': 'Rodrik Cassel'},
  {'name': 'Jory Cassel'},
  {'name': 'Jon Snow'}]},
{'sceneStart': '0:15:41',
 'sceneEnd': '0:18:44',
 'location': 'The North',
 'subLocation': 'Outside Winterfell',
 'characters': [{'name': 'Jon Snow'},
  {'name': 'Eddard Stark', 'weapon': [{'action': 'has', 'name': 'Ice'}]},
  {'name': 'Theon Greyjoy'},
  {'name': 'Bran Stark'},
  {'name': 'Robb Stark'},
  {'name': 'Rodrik Cassel'},
  {'name': 'Jory Cassel'},
  {'name': 'Grey Wind'},
  {'name': 'Lady'},
  {'name': 'Nymeria'},
  {'name': 'Summer'},
  {'name': 'Shaggydog'},
  {'name': 'Ghost'}]},
{'sceneStart': '0:18:44',
 'sceneEnd': '0:20:45',
 'location': 'The Crownlands',
 'subLocation': "King's Landing",
 'characters': [{'name': 'Jon Arryn',
   'alive': False,
   'title': 'Hand',
   'mannerOfDeath': 'Poison',
   'killedBy': ['Lysa Arryn']},
  {'name': 'Cersei Lannister'},
  {'name': 'Jaime Lannister'}]},
{'sceneStart': '0:20:45',
 'sceneEnd': '0:22:43',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Catelyn Stark'},
  {'name': 'Eddard Stark', 'weapon': [{'action': 'has', 'name': 'Ice'}]}]},
{'sceneStart': '0:22:43',
 'sceneEnd': '0:23:09',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Maester Luwin'}]},
{'sceneStart': '0:23:09',
 'sceneEnd': '0:23:39',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Robb Stark'},
  {'name': 'Theon Greyjoy'},
  {'name': 'Jon Snow'}]},
{'sceneStart': '0:23:39',
 'sceneEnd': '0:29:16',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Bran Stark'},
  {'name': 'Catelyn Stark'},
  {'name': 'Maester Luwin'},
  {'name': 'Summer'},
  {'name': 'Arya Stark'},
  {'name': 'Joffrey Baratheon',
   'weapon': [{'action': 'has', 'name': "Lion's Tooth"}]},
  {'name': 'Sandor Clegane'},
  {'name': 'Rickon Stark'},
  {'name': 'Eddard Stark'},
  {'name': 'Sansa Stark'},
  {'name': 'Theon Greyjoy'},
  {'name': 'Jon Snow'},
  {'name': 'Jory Cassel'},
  {'name': 'Hodor'},
  {'name': 'Rodrik Cassel'},
  {'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Cersei Lannister'},
  {'name': 'Jaime Lannister'},
  {'name': 'Myrcella Baratheon'},
  {'name': 'Tommen Baratheon'}]},
{'sceneStart': '0:29:16',
 'sceneEnd': '0:30:47',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Eddard Stark'}]},
{'sceneStart': '0:30:47',
 'sceneEnd': '0:32:57',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Jaime Lannister'},
  {'name': 'Tyrion Lannister',
   'sex': {'with': ['Ros'], 'type': 'paid', 'when': 'present'}},
  {'name': 'Ros',
   'sex': {'with': ['Tyrion Lannister'],
    'type': 'paid',
    'when': 'present'}}]},
{'sceneStart': '0:32:57',
 'sceneEnd': '0:33:45',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Eddard Stark'}]},
{'sceneStart': '0:33:45',
 'sceneEnd': '0:36:17',
 'location': 'Pentos',
 'characters': [{'name': 'Daenerys Targaryen'},
  {'name': 'Viserys Targaryen'}]},
{'sceneStart': '0:36:17',
 'sceneEnd': '0:38:23',
 'location': 'Pentos',
 'characters': [{'name': 'Daenerys Targaryen'},
  {'name': 'Viserys Targaryen'},
  {'name': 'Illyrio Mopatis'},
  {'name': 'Khal Drogo', 'title': 'Khal'},
  {'name': 'Qotho'}]},
{'sceneStart': '0:38:23',
 'sceneEnd': '0:40:05',
 'location': 'Pentos',
 'characters': [{'name': 'Viserys Targaryen'},
  {'name': 'Illyrio Mopatis'},
  {'name': 'Daenerys Targaryen'}]},
{'sceneStart': '0:40:05',
 'sceneEnd': '0:40:56',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Sansa Stark'}]},
{'sceneStart': '0:40:56',
 'sceneEnd': '0:41:19',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Catelyn Stark'},
  {'name': 'Cersei Lannister'},
  {'name': 'Rodrik Cassel'}]},
{'sceneStart': '0:41:19',
 'sceneEnd': '0:44:14',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Jon Snow'},
  {'name': 'Benjen Stark'},
  {'name': 'Tyrion Lannister'}]},
{'sceneStart': '0:44:14',
 'sceneEnd': '0:47:28',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Rodrik Cassel'},
  {'name': 'Benjen Stark'},
  {'name': 'Eddard Stark'},
  {'name': 'Robb Stark'},
  {'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Catelyn Stark'},
  {'name': 'Cersei Lannister'},
  {'name': 'Sansa Stark'},
  {'name': 'Joffrey Baratheon'},
  {'name': 'Jaime Lannister'},
  {'name': 'Arya Stark'},
  {'name': 'Theon Greyjoy'}]},
{'sceneStart': '0:47:28',
 'sceneEnd': '0:50:29',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Eddard Stark'},
  {'name': 'Catelyn Stark'},
  {'name': 'Maester Luwin'}]},
{'sceneStart': '0:50:29',
 'sceneEnd': '0:56:05',
 'location': 'Pentos',
 'characters': [{'name': 'Khal Drogo',
   'title': 'Khal',
   'married': {'to': 'Daenerys Targaryen',
    'when': 'present',
    'type': 'arranged',
    'consummated': True}},
  {'name': 'Daenerys Targaryen',
   'title': 'Khaleesi',
   'married': {'to': 'Khal Drogo',
    'when': 'present',
    'type': 'arranged',
    'consummated': True}},
  {'name': 'Viserys Targaryen'},
  {'name': 'Illyrio Mopatis'},
  {'name': 'Qotho'},
  {'name': 'Jorah Mormont'},
  {'name': 'Drogon', 'born': False},
  {'name': 'Rhaegal', 'born': False},
  {'name': 'Viserion', 'born': False}]},
{'sceneStart': '0:56:05',
 'sceneEnd': '0:57:48',
 'location': 'Pentos',
 'characters': [{'name': 'Khal Drogo',
   'title': 'Khal',
   'sex': {'with': ['Daenerys Targaryen'],
    'when': 'present',
    'type': 'rape'}},
  {'name': 'Daenerys Targaryen',
   'title': 'Khaleesi',
   'sex': {'with': ['Khal Drogo'], 'when': 'present', 'type': 'rape'}}]},
{'sceneStart': '0:57:48',
 'sceneEnd': '0:59:06',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Tyrion Lannister'},
  {'name': 'Sandor Clegane'},
  {'name': 'Theon Greyjoy'},
  {'name': 'Eddard Stark'},
  {'name': 'Benjen Stark'},
  {'name': 'Robb Stark'},
  {'name': 'Robert Baratheon', 'title': 'King'},
  {'name': 'Bran Stark'},
  {'name': 'Summer'}]},
{'sceneStart': '0:59:06',
 'sceneEnd': '1:00:57',
 'location': 'The North',
 'subLocation': 'Winterfell',
 'characters': [{'name': 'Bran Stark'},
  {'name': 'Summer'},
  {'name': 'Jaime Lannister',
   'sex': {'with': ['Cersei Lannister'],
    'when': 'present',
    'type': 'love'}},
  {'name': 'Cersei Lannister',
   'sex': {'with': ['Jaime Lannister'],
    'when': 'present',
    'type': 'love'}}]}]}]
```

In [7]:

```
locations = load_json("./data/locations.json")
```

In [8]:

```
locations[0:2]
```

Out[8]:

```
[{'location': 'North of the Wall',
  'subLocation': ['The Lands of Always Winter',
   'Cave Outside Wildling Camp',
   'Wildling Camp',
   'Frostfang Mountains',
   'The Three-Eyed Raven',
   'Outside the Three-Eyed Raven',
   'Fist of the First Men',
   'Wildlings March South',
   'Nightswatch March South',
   'Hardhome',
   "Craster's Keep",
   'The Haunted Forest',
   'South to the Wall',
   'Destroyed Cabin',
   'Near Nightfort',
   'The Wall']},
 {'location': 'The Wall',
  'subLocation': ['Nightfort',
   'Eastwatch',
   'Top of the Wall',
   'Castle Black',
   'Outside Castle Black',
   "Mole's Town",
   'The Gift']}]
```

In [9]:

```
characters_groups = load_json("./data/characters-groups.json")
```

In [10]:

```
characters_groups[0:1]
```

Out[10]:

```
[{'name': 'Stark',
  'characters': ['Arya Stark',
   'Benjen Stark',
   'Bran Stark',
   'Catelyn Stark',
   'Eddard Stark',
   'Ghost',
   'Grey Wind',
   'Jon Snow',
   'Lady',
   'Nymeria',
   'Rickon Stark',
   'Robb Stark',
   'Sansa Stark',
   'Shaggydog',
   'Summer']}]
```

Assignment 1: Load MongoDB

Load Information

- You must load the data into collections in MongoDB.
- The following code uses the helper files from setup to get a connection to MongoDB.

In [11]:

```
import got_mongo
```

In [12]:

```
from pymongo import MongoClient
```

In [13]:

```
client = MongoClient("mongodb+srv://de2418:Edy990127@aaaa.1raes.mongodb.net/GoT?retryWr
```

In [14]:

```
list(client.list_databases())
```

Out[14]:

```
[{'name': 'GoT', 'sizeOnDisk': 581632, 'empty': False},
 {'name': 'admin', 'sizeOnDisk': 344064, 'empty': False},
 {'name': 'local', 'sizeOnDisk': 7151984640, 'empty': False}]
```

- In the following cells, write Python code that loads the information into:
  - Database: `GoT`
  - Collections: `characters, episodes, locations, groups.`
- Put your code to load in the cells below. Later, there are some tests.

In [15]:

```
# Load characters
client['GoT']['characters'].drop()
for c in characters:
    client['GoT']['characters'].insert_one(c)
```

In [16]:

```
# Load episodes
client['GoT']['episodes'].drop()
for c in episodes:
    client['GoT']['episodes'].insert_one(c)
```

In [17]:

```
# Load Locations
client['GoT']['locations'].drop()
for c in locations:
    client['GoT']['locations'].insert_one(c)
```

In [18]:

```
# Load character groups.
client['GoT']['groups'].drop()
for c in characters_groups:
    client['GoT']['groups'].insert_one(c)
```

Interesting Tests

- If you have loaded the data character, the following queries should produce interesting results.

Characters in Episodes

In [19]:

```
def get_characters_per_episode(seasonNum, episodeNum):
    client = MongoClient("mongodb+srv://de2418:Edy990127@aaaa.1raes.mongodb.net/GoT?ret
    result = client['GoT']['episodes'].aggregate([
        {
            '$match': {
                'episodeNum': seasonNum,
                'seasonNum': episodeNum
            }
        }, {
            '$project': {
                'characters': '$scenes.characters'
            }
        }, {
            '$unwind': {
                'path': '$characters',
                'preserveNullAndEmptyArrays': False
            }
        }, {
            '$unwind': {
                'path': '$characters',
                'preserveNullAndEmptyArrays': False
            }
        }, {
            '$project': {
                'characterName': '$characters.name',
                '_id': 0
            }
        }
    ])
    return result
```

In [20]:

```
result = get_characters_per_episode(2, 1)
result = list(result)
```

In [21]:

```
result[0:2]
```

Out[21]:

```
[{'characterName': 'Jorah Mormont'}, {'characterName': 'Daenerys Targaryen'}]
```

In [22]:

```
result = [l['characterName'] for l in result]
result = set(result)
```

In [23]:

```
result
```

Out[23]:

```
{'Arya Stark',
 'Benjen Stark',
 'Bran Stark',
 'Catelyn Stark',
 'Catspaw Assassin',
 'Cersei Lannister',
 'Daenerys Targaryen',
 'Doreah',
 'Drogon',
 'Eddard Stark',
 'Ghost',
 'Ilyn Payne',
 'Irri',
 'Jaime Lannister',
 'Jhiqui',
 'Joffrey Baratheon',
 'Jon Snow',
 'Jorah Mormont',
 'Jory Cassel',
 'Khal Drogo',
 'Lady',
 'Maester Luwin',
 'Mycah',
 'Myrcella Baratheon',
 'Nymeria',
 'Rast',
 'Rhaegal',
 'Robb Stark',
 'Robert Baratheon',
 'Rodrik Cassel',
 'Sandor Clegane',
 'Sansa Stark',
 'Summer',
 'Theon Greyjoy',
 'Tommen Baratheon',
 'Tyrion Lannister',
 'Viserion',
 'Viserys Targaryen'}
```

Find All the Actors who Played a Character in a Group

In [24]:

```
def get_actor_for_group(group_name):

    result = client['GoT']['groups'].aggregate([
        {
            '$match': {
                "name": group_name
            }
        },
        {
            '$unwind': {
                'path': '$characters',
                'preserveNullAndEmptyArrays': True
            }
        }, {
            '$project': {
                'groupName': '$name',
                'characterName': '$characters'
            }
        }, {
            '$lookup': {
                'from': 'characters',
                'localField': 'characterName',
                'foreignField': 'characterName',
                'as': 'character'
            }
        }, {
            '$unwind': {
                'path': '$character'
            }
        }, {
            '$project': {
                'characterName': 1,
                'groupName': 1,
                'actorName': '$character.actorName',
                'actorLink': {
                    '$concat': [
                        'https://www.imdb.com', '$character.actorLink'
                    ]
                }
            }
        }
    ])
    return result
```

In [25]:

```
nights_watch_actors = get_actor_for_group("Night's Watch")
```

In [26]:

```
result = list(nights_watch_actors)
result
```

Out[26]:

```
[{'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Alliser Thorne',
  'actorName': 'Owen Teale',
  'actorLink': 'https://www.imdb.com/name/nm0853583/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Eddison Tollett',
  'actorName': 'Ben Crompton',
  'actorLink': 'https://www.imdb.com/name/nm0174005/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Grenn',
  'actorName': 'Mark Stanley',
  'actorLink': 'https://www.imdb.com/name/nm4440305/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Jeor Mormont',
  'actorName': 'James Cosmo',
  'actorLink': 'https://www.imdb.com/name/nm0181920/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Karl Tanner',
  'actorName': 'Burn Gorman',
  'actorLink': 'https://www.imdb.com/name/nm1218607/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Maester Aemon',
  'actorName': 'Peter Vaughan',
  'actorLink': 'https://www.imdb.com/name/nm0891092/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Olly',
  'actorName': "Brenock O'Connor",
  'actorLink': 'https://www.imdb.com/name/nm5282143/'},
{'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Othell Yarwyck',
  'actorName': 'Brian Fortune',
  'actorLink': 'https://www.imdb.com/name/nm1987270/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Pypar',
  'actorName': 'Josef Altin',
  'actorLink': 'https://www.imdb.com/name/nm1086981/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Qhorin Halfhand',
  'actorName': 'Simon Armstrong',
  'actorLink': 'https://www.imdb.com/name/nm1409802/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Rast',
  'actorName': 'Luke Barnes',
  'actorLink': 'https://www.imdb.com/name/nm4430828/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Samwell Tarly',
  'actorName': 'John Bradley',
  'actorLink': 'https://www.imdb.com/name/nm4263213/'},
 {'_id': ObjectId('625e15e7bfd4b732a108d77b'),
  'groupName': "Night's Watch",
  'characterName': 'Yoren',
  'actorName': 'Francis Magee',
  'actorLink': 'https://www.imdb.com/name/nm0535837/'}]
```

Assignment 2: Load Neo4j

Load Data

- You loaded information about characters and groups into MongoDB.

- You also have to data loaded into a Python data structure from prior steps in this notebook.

- Your task is to:
  - Create a node in your Neo4j database for each group.
  - Create a node in your Neo4j database for each character in a group. If a character is in multiple groups, you should only create the node once.
  - Create a relationship `MEMBER_OF` from the character node to the group node for each member in the group.

- The code below retests your ability to connect to Neo4j.

In [28]:

```
from py2neo import data, Graph, NodeMatcher, Node, Relationship, RelationshipMatcher
```

In [29]:

```
g = Graph('neo4j+s://13f30c6c.databases.neo4j.io:7687',
          auth = ("neo4j", "isZCQgcAMZ9UusAH_w-hoqAtB7TsjJnXiEwrvbuCbGA"))
```

In [30]:

```
q = "match (p:Person) where p.name=$name return p"
res = g.run(q, name = "Tom Hanks")
print(type(res))
for r in res:
    print(type(r))
    print(r)
    print("Labels = ", r['p'].labels)
    print("Properties = ", dict(r['p']))
```

```
<class 'py2neo.cypher.Cursor'>
```

- Use the cells below to load Neo4j.

- In the next section, we will run some tests that will help you understand if you created the information correctly.

In [48]:

```
# Put your loading code here.
# Looking at the code in got_test_neo.py may help you.
#
```

In [32]:

```
group_names = [g['name'] for g in characters_groups]
group_names
```

Out[32]:

```
['Stark',
 'Targaryen',
 'Baratheon',
 'Lannister',
 "Night's Watch",
 'Dothraki',
 'Greyjoy',
 'Tyrell',
 'Wildlings',
 'Martell',
 'Frey',
 'Tully',
 'White Walkers',
 'Include']
```

In [33]:

```
cypher_q = "create (g:GoT_Group {groupName: $g_name}) return g"
```

In [34]:

```
g1 = group_names[0]
```

In [35]:

```
for i in range(len(group_names)):
    result = g.run(cypher_q, g_name = group_names[i])
```

In [36]:

```
ccs = []
for x in characters_groups:
    gn = x['name']
    for c in x['characters']:
        ccs.append([gn,c])
```

In [37]:

```
cypher_q2 = "create (g:GoT_Character {characterName: $c_name}) return g"
```

In [38]:

```
c1 = ccs[0][1]
```

In [39]:

```
for i in range(len(ccs)):
    g.run(cypher_q2, c_name = ccs[i][1])
```

In [40]:

```
cypher_q3 = """
    match (c:GoT_Character {characterName: $c_name}), (g:GoT_Group {groupName: $g_name}
    create (c)-[:MEMBER_OF]->(g)
"""
```

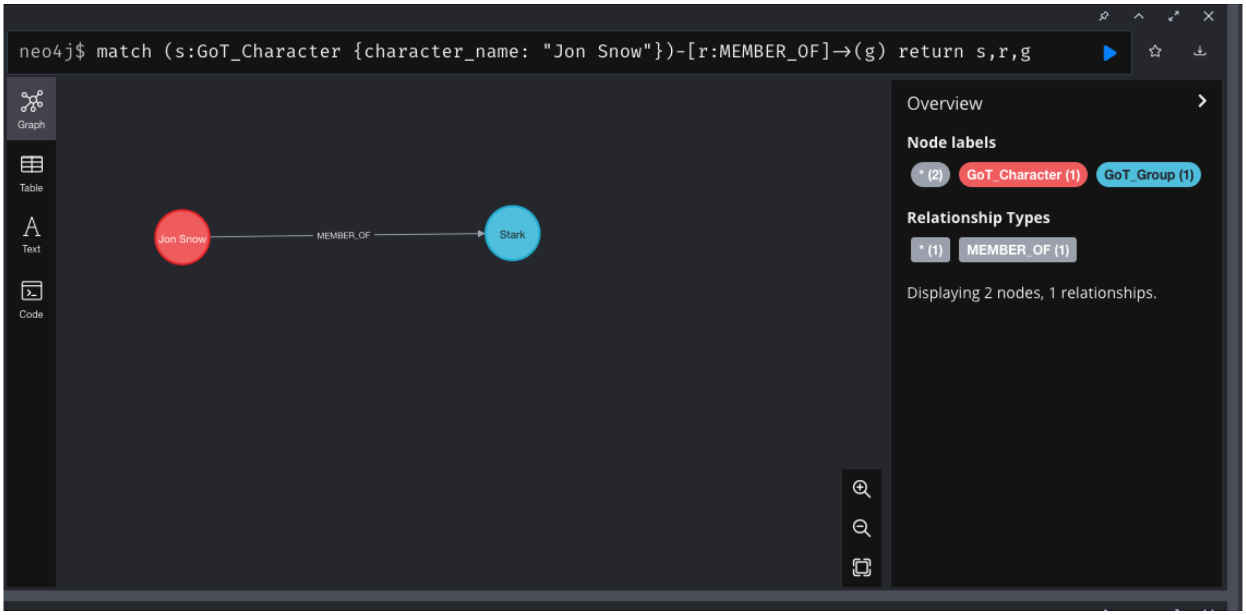
In [41]:

```
for i in range(len(ccs)):
    g.run(cypher_q3, c_name = ccs[i][1], g_name = ccs[i][0])
```

Test Load

- Running the following query in the query explorer for Neo4j should produce the following.

```
match (s:GoT_Character {character_name: "Jon Snow"})-[r:MEMBER_OF]->(g)
    return s,r,g
```



```
neo4j$ match (s:GoT_Character {character_name: "Jon Snow"})-[r:MEMBER_OF]→(g) return s,r,g
```

Overview

Node labels

* (2) GoT_Character (1) GoT_Group (1)

Relationship Types

* (1) MEMBER_OF (1)

Displaying 2 nodes, 1 relationships.

- You can test this using ... ...

In [42]:

```
cypher_q = """
    match
        (c:GoT_Character {characterName: 'Jon Snow'})-[r:MEMBER_OF]->(g)
    return c,r,g
```

```
"""
res = g.run(cypher_q, None)
```

In [43]:

```
for r in res:
    print(r[0])
    print(r[1])
    print(r[2])
```

```
(_4:GoT_Character {characterName: 'Jon Snow'})
(_4)-[:MEMBER_OF {}]->(_151)
(_151:GoT_Group {groupName: 'Stark'})
```

- The following query and code will produce a data frame of groups and characters

In [44]:

```
cypher_q = """
    match
        (c:GoT_Character)-[r:MEMBER_OF]->(g)
    return c.characterName, g.groupName
"""
res = g.run(cypher_q, None)
```

In [45]:

```
import pandas as pd
```

In [46]:

```
df = pd.DataFrame(res)
```

In [47]:

```
df.columns = ['character_name', 'group_name']
df
```

Out[47]:

	character_name	group_name
0	Catelyn Stark	Stark
1	Eddard Stark	Stark
2	Ghost	Stark
3	Grey Wind	Stark
4	Jon Snow	Stark
...	...	...
151	Yezzan zo Qaggaz	Include
152	Yohn Royce	Include
153	Arya Stark	Stark
154	Benjen Stark	Stark
155	Bran Stark	Stark

156 rows × 2 columns