# 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.

```python
In [78]: import json
```

```python
In [79]: def load_json(fn):
    result = None
    with open(fn, "r") as in_file:
        result = json.load(in_file)
    return result
```

```
In [80]: characters = load_json("./data/characters.json")
```

```
In [81]: characters[0:2]
```

```
Out[81]: [{'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 [82]: episodes = load_json("./data/episodes.json")
```

```
In [83]: episodes[0:1]
```

```
Out[83]:  [{'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. A
          cross the sea, Viserys 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 [84]: `locations = load_json("./data/locations.json")`

In [85]: `locations[0:2]`

```
Out[85]: [{'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 [86]: characters_groups = load_json("./data/characters-groups.json")
```

```
In [87]: characters_groups[0:1]
```

```
Out[87]: [{'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 [30]: !pip install pymongo
```

```
Requirement already satisfied: pymongo in /Users/anyadevgan/opt/anaconda3/lib/
python3.9/site-packages (4.1.1)
```

In [48]:
```python
from pymongo import MongoClient
```

In [49]:
```python
client = \
    MongoClient("mongodb+srv://ad3706:sas41179@cluster0.fzxwm.mongodb.net/GoTR?
```

In [50]:
```python
list(client.list_databases())
```

Out[50]:
```
[{'name': 'GoTR', 'sizeOnDisk': 491520, 'empty': False},
 {'name': 'admin', 'sizeOnDisk': 344064, 'empty': False},
 {'name': 'local', 'sizeOnDisk': 8927674368, 'empty': False}]
```

In [34]:
```python
#import got_mongo #REMOVED AS TOLD IN RECITATION
```

In [35]:
```python
#client = got_mongo.get_client() #REMOVED AS TOLD IN RECITATION
```

- 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 [36]:
```python
# Load characters
client['GoTR']['characters'].drop()
```

In [37]:
```python
for c in characters:
    client['GoTR']['characters'].insert_one(c)
```

In [38]:
```python
client['GoTR']['episodes'].drop()
```

In [39]:
```python
# Load episodes
for e in episodes:
    client['GoTR']['episodes'].insert_one(e)
```

In [40]:
```python
client['GoTR']['locations'].drop()
```

In [41]:
```python
# Load locations
for l in locations:
    client['GoTR']['locations'].insert_one(l)
```

In [42]:
```python
client['GoTR']['groups'].drop()
```

In [43]:
```python
# Load character groups.
for g in characters_groups:
    client['GoTR']['groups'].insert_one(g)
```

## Interesting Tests

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

## Characters in Episodes

```
In [44]:  #client = got_mongo.get_client()
```

```
In [51]:  def get_characters_per_episode(seasonNum, episodeNum):
              #client = got_mongo.get_client()
              result = client['GoTR']['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 [52]:  result = get_characters_per_episode(2, 1)
          result = list(result)
```

```
In [53]:  result[0:2]
```

```
Out[53]:  [{'characterName': 'Jorah Mormont'}, {'characterName': 'Daenerys Targaryen'}]
```

```
In [54]:  result = [l['characterName'] for l in result]
          result = set(result) #converting into a set gets rid of duplicates
```

```
In [55]:  result
```

Out[55]: {'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

```python
In [57]: def get_actor_for_group(group_name):
             #client = got_mongo.get_client()

             result = client['GoTR']['groups'].aggregate([
                 {
                     '$match': {
                         "name": group_name
                     }
                 },
                 {
                     '$unwind': {
                         'path': '$characters',
                         'preserveNullAndEmptyArrays': True
                     }
                 }, {
                     '$project': {
                         'groupName': '$name',
                         'characterName': '$characters'
```

```
                }
            }, {
                '$lookup': { #use lookup for join operation
                    'from': 'characters',
                    'localField': 'characterName',
                    'foreignField': 'characterName',
                    'as': 'character'
                }
            }, {
                '$unwind': {
                    'path': '$character'
                }
            }, {
                '$project': {
                    'characterName': 1,
                    'groupName': 1,
                    'actorName': '$character.actorName',
                    'actorLink': {
                        '$concat': [
                            'https://www.imdb.com', '$character.actorLink'
                        ]
                    }
                }
            }
        ])
        return result
```

In [58]:
```
nights_watch_actors = get_actor_for_group("Night's Watch")
```

In [59]:
```
result = list(nights_watch_actors)
result
```

```
Out[59]: [{'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Alliser Thorne',
           'actorName': 'Owen Teale',
           'actorLink': 'https://www.imdb.com/name/nm0853583/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Eddison Tollett',
           'actorName': 'Ben Crompton',
           'actorLink': 'https://www.imdb.com/name/nm0174005/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Grenn',
           'actorName': 'Mark Stanley',
           'actorLink': 'https://www.imdb.com/name/nm4440305/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Jeor Mormont',
           'actorName': 'James Cosmo',
           'actorLink': 'https://www.imdb.com/name/nm0181920/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Karl Tanner',
           'actorName': 'Burn Gorman',
           'actorLink': 'https://www.imdb.com/name/nm1218607/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Maester Aemon',
           'actorName': 'Peter Vaughan',
           'actorLink': 'https://www.imdb.com/name/nm0891092/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Olly',
           'actorName': "Brenock O'Connor",
           'actorLink': 'https://www.imdb.com/name/nm5282143/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Othell Yarwyck',
           'actorName': 'Brian Fortune',
           'actorLink': 'https://www.imdb.com/name/nm1987270/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Pypar',
           'actorName': 'Josef Altin',
           'actorLink': 'https://www.imdb.com/name/nm1086981/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Qhorin Halfhand',
           'actorName': 'Simon Armstrong',
           'actorLink': 'https://www.imdb.com/name/nm1409802/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Rast',
           'actorName': 'Luke Barnes',
           'actorLink': 'https://www.imdb.com/name/nm4430828/'},
          {'_id': ObjectId('625c4893954cd83cf48322d3'),
           'groupName': "Night's Watch",
           'characterName': 'Samwell Tarly',
           'actorName': 'John Bradley',
           'actorLink': 'https://www.imdb.com/name/nm4263213/'},
```

```
{'_id': ObjectId('625c4893954cd83cf48322d3'),
 '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.

In [108…    `characters_groups`

```
Out[108]:  [{'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']},
            {'name': 'Targaryen',
             'characters': ['Daenerys Targaryen',
              'Drogon',
              'Rhaegal',
              'Viserion',
              'Viserys Targaryen']},
            {'name': 'Baratheon',
             'characters': ['Joffrey Baratheon',
              'Myrcella Baratheon',
              'Renly Baratheon',
              'Robert Baratheon',
              'Selyse Baratheon',
              'Shireen Baratheon',
              'Stannis Baratheon',
              'Tommen Baratheon',
              'Gendry']},
            {'name': 'Lannister',
             'characters': ['Cersei Lannister',
              'Jaime Lannister',
              'Kevan Lannister',
              'Lancel Lannister',
              'Tyrion Lannister',
              'Tywin Lannister']},
            {'name': "Night's Watch",
             'characters': ['Alliser Thorne',
              'Eddison Tollett',
              'Grenn',
              'Jeor Mormont',
              'Karl Tanner',
              'Maester Aemon',
              'Olly',
              'Othell Yarwyck',
              'Pypar',
              'Qhorin Halfhand',
              'Rast',
              'Samwell Tarly',
              'Yoren']},
            {'name': 'Dothraki',
             'characters': ['Doreah', 'Irri', 'Khal Drogo', 'Rakharo', 'Qhono']},
            {'name': 'Greyjoy',
             'characters': ['Balon Greyjoy',
              'Euron Greyjoy',
              'Theon Greyjoy',
              'Yara Greyjoy']},
```

```
{'name': 'Tyrell',
 'characters': ['Loras Tyrell',
  'Mace Tyrell',
  'Margaery Tyrell',
  'Olenna Tyrell']},
{'name': 'Wildlings',
 'characters': ['Baby Sam',
  'Craster',
  'Gilly',
  'Mag the Mighty',
  'Mance Rayder',
  'Osha',
  'Rattleshirt',
  'Thenn Warg',
  'Tormund Giantsbane',
  'Wun Wun',
  'Ygritte']},
{'name': 'Martell',
 'characters': ['Doran Martell',
  'Ellaria Sand',
  'Nymeria Sand',
  'Obara Sand',
  'Oberyn Martell',
  'Trystane Martell',
  'Tyene Sand']},
{'name': 'Frey',
 'characters': ['Black Walder Rivers', 'Lothar Frey', 'Walder Frey']},
{'name': 'Tully', 'characters': ['Brynden Tully', 'Edmure Tully']},
{'name': 'White Walkers',
 'characters': ['Mag the Mighty Wight',
  'The Night King',
  'White Walker',
  'Wight Wildling Girl',
  'Wun Wun Wight']},
{'name': 'Include',
 'characters': ['Archmaester Ebrose',
  'Barristan Selmy',
  'Beric Dondarrion',
  'Brienne of Tarth',
  'Bronn',
  'Daario Naharis',
  'Davos Seaworth',
  'Gendry',
  'Grand Maester Pycelle',
  'Greatjon Umber',
  'Gregor Clegane',
  'Greizhen mo Ullhor',
  'Grey Worm',
  'Hallyne',
  'High Sparrow',
  'Hizdahr zo Loraq',
  'Hodor',
  'Hot Pie',
  'Janos Slynt',
  "Jaqen H'ghar",
  'Jojen Reed',
  'Jorah Mormont',
  'Jory Cassel',
  'Kraznys mo Nakloz',
  'Lady Crane',
```

```
            'Leaf',
            'Locke',
            'Lommy Greenhands',
            'Lord Varys',
            'Lyanna Mormont',
            'Lysa Arryn',
            'Maester Luwin',
            'Matthos Seaworth',
            'Meera Reed',
            'Melisandre',
            'Mirri Maz Duur',
            'Missandei',
            'Olyvar',
            'Petyr Baelish',
            'Podrick Payne',
            'Prendahl na Ghezn',
            'Pyat Pree',
            'Qyburn',
            'Ramsay Snow',
            'Razdal mo Eraz',
            'Rickard Karstark',
            'Robin Arryn',
            'Rodrik Cassel',
            'Roose Bolton',
            'Ros',
            'Salladhor Saan',
            'Sandor Clegane',
            'Septa Mordane',
            'Septa Unella',
            'Shae',
            'Silk King',
            'Smalljon Umber',
            'Syrio Forel',
            'Talisa Maegyr',
            'The Waif',
            'Thoros of Myr',
            'Three-Eyed Raven',
            'Xaro Xhoan Daxos',
            'Yezzan zo Qaggaz',
            'Yohn Royce']}]
```

In [109…  ```
          from py2neo import data, Graph, NodeMatcher, Node, Relationship, RelationshipMa
          ```

In [110…  ```
          g = Graph(
              "neo4j+s://4606c119.databases.neo4j.io:7687",
              auth=("neo4j", "wbiRMNlI3Vu3EhOzEiCrEYbbjtQIz-xWk442i_zhvO0")
          )
          ```

- 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 [111]... `#import got_test_neo #REMOVED AS IN RECITATION`

In [112]... `#g = got_test_neo.get_graph() #REMOVED AS IN RECITATION`

In [113]...
```python
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'>
<class 'py2neo.cypher.Record'>
Node('Person', born=1956, name='Tom Hanks')
Labels =  :Person
Properties =  {'born': 1956, 'name': 'Tom Hanks'}
```

- 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 [114]...
```python
# Put your loading code here.
# Looking at the code in got_test_neo.py may help you.
#
```

In [169]...
```python
#Create a node for each group
group_names = [g['name'] for g in characters_groups]
group_names
```

Out[169]:
```
['Stark',
 'Targaryen',
 'Baratheon',
 'Lannister',
 "Night's Watch",
 'Dothraki',
 'Greyjoy',
 'Tyrell',
 'Wildlings',
 'Martell',
 'Frey',
 'Tully',
 'White Walkers',
 'Include']
```

In [170]... `cypher_q1 = "create (g:GoTR_Group {group_name: $g_name}) return g"`

In [171]...
```python
for tmp in group_names:
    result = g.run(cypher_q1, g_name=tmp)
```

In [172]... `result`

Out[172]:                                                                **g**

(_238:GoTR_Group {group_name: 'Include'})

In [173…
```python
#Create a node for each character in a group
chars = set()
for x in characters_groups:
    for c in x['characters']:
        chars.add(c)
```

In [174…
```python
chars
```

```
Out[174]:  {'Alliser Thorne',
            'Archmaester Ebrose',
            'Arya Stark',
            'Baby Sam',
            'Balon Greyjoy',
            'Barristan Selmy',
            'Benjen Stark',
            'Beric Dondarrion',
            'Black Walder Rivers',
            'Bran Stark',
            'Brienne of Tarth',
            'Bronn',
            'Brynden Tully',
            'Catelyn Stark',
            'Cersei Lannister',
            'Craster',
            'Daario Naharis',
            'Daenerys Targaryen',
            'Davos Seaworth',
            'Doran Martell',
            'Doreah',
            'Drogon',
            'Eddard Stark',
            'Eddison Tollett',
            'Edmure Tully',
            'Ellaria Sand',
            'Euron Greyjoy',
            'Gendry',
            'Ghost',
            'Gilly',
            'Grand Maester Pycelle',
            'Greatjon Umber',
            'Gregor Clegane',
            'Greizhen mo Ullhor',
            'Grenn',
            'Grey Wind',
            'Grey Worm',
            'Hallyne',
            'High Sparrow',
            'Hizdahr zo Loraq',
            'Hodor',
            'Hot Pie',
            'Irri',
            'Jaime Lannister',
            'Janos Slynt',
            "Jaqen H'ghar",
            'Jeor Mormont',
            'Joffrey Baratheon',
            'Jojen Reed',
            'Jon Snow',
            'Jorah Mormont',
            'Jory Cassel',
            'Karl Tanner',
            'Kevan Lannister',
            'Khal Drogo',
            'Kraznys mo Nakloz',
            'Lady',
            'Lady Crane',
            'Lancel Lannister',
            'Leaf',
```

```
'Locke',
'Lommy Greenhands',
'Loras Tyrell',
'Lord Varys',
'Lothar Frey',
'Lyanna Mormont',
'Lysa Arryn',
'Mace Tyrell',
'Maester Aemon',
'Maester Luwin',
'Mag the Mighty',
'Mag the Mighty Wight',
'Mance Rayder',
'Margaery Tyrell',
'Matthos Seaworth',
'Meera Reed',
'Melisandre',
'Mirri Maz Duur',
'Missandei',
'Myrcella Baratheon',
'Nymeria',
'Nymeria Sand',
'Obara Sand',
'Oberyn Martell',
'Olenna Tyrell',
'Olly',
'Olyvar',
'Osha',
'Othell Yarwyck',
'Petyr Baelish',
'Podrick Payne',
'Prendahl na Ghezn',
'Pyat Pree',
'Pypar',
'Qhono',
'Qhorin Halfhand',
'Qyburn',
'Rakharo',
'Ramsay Snow',
'Rast',
'Rattleshirt',
'Razdal mo Eraz',
'Renly Baratheon',
'Rhaegal',
'Rickard Karstark',
'Rickon Stark',
'Robb Stark',
'Robert Baratheon',
'Robin Arryn',
'Rodrik Cassel',
'Roose Bolton',
'Ros',
'Salladhor Saan',
'Samwell Tarly',
'Sandor Clegane',
'Sansa Stark',
'Selyse Baratheon',
'Septa Mordane',
'Septa Unella',
'Shae',
```

```
                    'Shaggydog',
                    'Shireen Baratheon',
                    'Silk King',
                    'Smalljon Umber',
                    'Stannis Baratheon',
                    'Summer',
                    'Syrio Forel',
                    'Talisa Maegyr',
                    'The Night King',
                    'The Waif',
                    'Thenn Warg',
                    'Theon Greyjoy',
                    'Thoros of Myr',
                    'Three-Eyed Raven',
                    'Tommen Baratheon',
                    'Tormund Giantsbane',
                    'Trystane Martell',
                    'Tyene Sand',
                    'Tyrion Lannister',
                    'Tywin Lannister',
                    'Viserion',
                    'Viserys Targaryen',
                    'Walder Frey',
                    'White Walker',
                    'Wight Wildling Girl',
                    'Wun Wun',
                    'Wun Wun Wight',
                    'Xaro Xhoan Daxos',
                    'Yara Greyjoy',
                    'Yezzan zo Qaggaz',
                    'Ygritte',
                    'Yohn Royce',
                    'Yoren'}
```

In [176… 
```python
cypher_q2 = "create (g:GoTR_Character {character_name: $c_name}) return g"
```

In [177…
```python
for tmp in chars:
    result = g.run(cypher_q2, c_name=tmp)
```

In [178…
```python
#Create relationships

#run 1 neo4j query that:
    #a) for x in characters_groups
    #b) you set chars = x['characters']
    #c) you set name = x['name']
    #d) for c in chars:
        #neo4j query that creates relationship between c and name
```

In [179…
```python
cypher_q3 = """
    match (c:GoTR_Character {character_name: $c_name}), (g:GoTR_Group {group_na
    create (c)-[:MEMBER_OF]->(g)
"""
```

In [180…
```python
for x in characters_groups:
    c = x['characters']
    gn = x['name']
    for tmp in c:
        result = g.run(cypher_q3, c_name = tmp, g_name = gn)
```

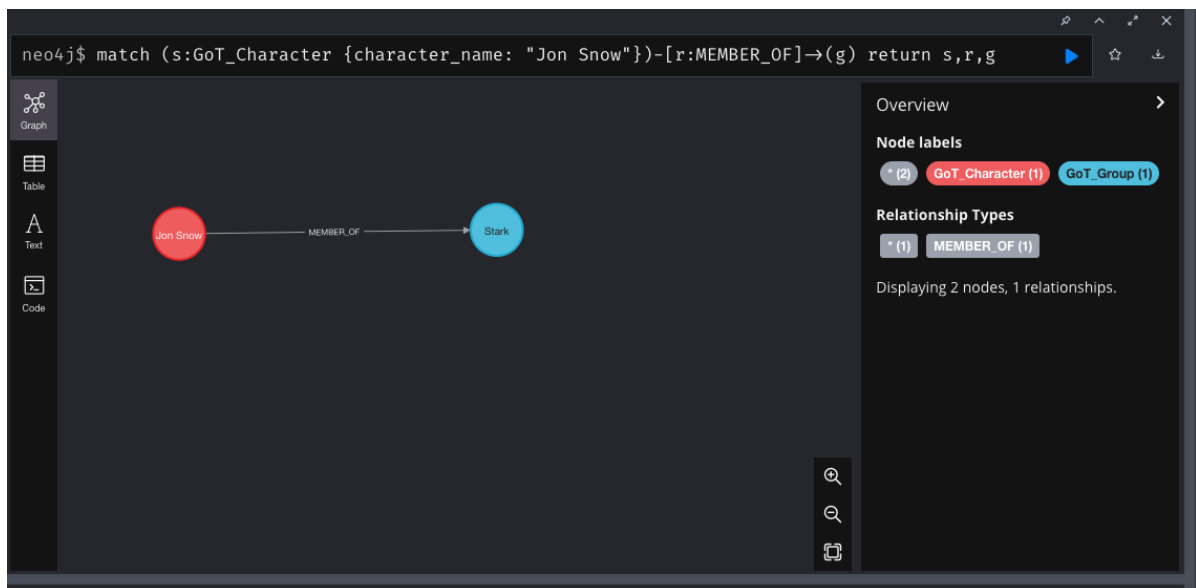In [181…    `result`

Out[181]:   (No data)

## 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
  ```



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

In [182…
```python
cypher_q = """
    match
        (c:GoTR_Character {character_name: 'Jon Snow'})-[r:MEMBER_OF]->(g)
    return c,r,g
"""
res = g.run(cypher_q, None)
```

In [183…
```python
for r in res:
    print(r[0])
    print(r[1])
    print(r[2])
```

```
(_386:GoTR_Character {character_name: 'Jon Snow'})
(_386)-[:MEMBER_OF {}]->(_225)
(_225:GoTR_Group {group_name: 'Stark'})
```

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

```
In [184…    cypher_q = """
                match
                    (c:GoTR_Character)-[r:MEMBER_OF]->(g)
                return c.character_name, g.group_name
            """
            res = g.run(cypher_q, None)
```

```
In [185…    import pandas as pd
```

```
In [186…    df = pd.DataFrame(res)
```

```
In [187…    df.columns = ['character_name', 'group_name']
            df
```

Out[187]:

|     | character_name | group_name |
| --- | --- | --- |
| 0 | Jorah Mormont | Include |
| 1 | Mace Tyrell | Tyrell |
| 2 | Doran Martell | Martell |
| 3 | Olly | Night's Watch |
| 4 | Joffrey Baratheon | Baratheon |
| ... | ... | ... |
| 149 | Gilly | Wildlings |
| 150 | Grand Maester Pycelle | Include |
| 151 | Lyanna Mormont | Include |
| 152 | Shaggydog | Stark |
| 153 | Catelyn Stark | Stark |

154 rows × 2 columns

```
In [ ]:
```