

# Chapter 1

## Graph database

The DBSM chosen for the graph database is Neo4j, it was use to manege the social part of the site and also to keep track of the reviews made for the feed and suggestion functionality.

The entities used in the database are:

- User
- TopCritic
- Movie

The nodes themselves do not store a lot of data, we decided to keep the bare minimum by having the following attributes:

- for the User and TopCritic:
  - id
  - name
- for Movie:
  - id
  - title

The relationship present are:

- User -[:FOLLOWS]-> TopCritic
- User -[:REVIEWED]-> Movie
- TopCritic -[:REVIEWED]-> Movie

In particular the *REVIEWED* relationship contains the following information:

- content
- date
- freshness

Below is present a snapshot of the graph database taken from Neo4j.

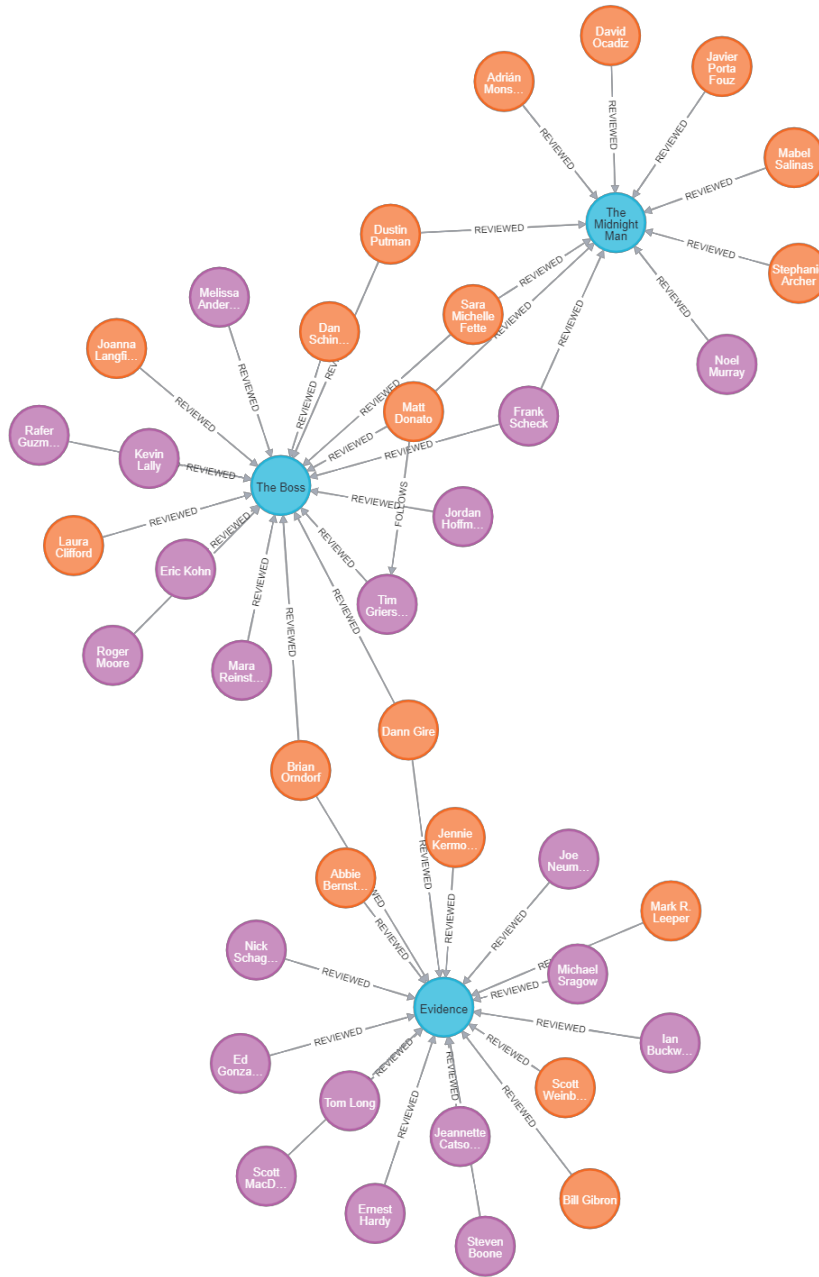


Figure 1.1: graph DB

As previously stated in the Feasibility study section 2 , the graph database design was heavily influenced by the document one, it was in fact generated by python script (Appendix A .6) that starts from the movie collection and, after generating the nodes for the *Movie* entity, does the same for the user by dividing them between normal user and top critic. It then generates the *REVIEWED* relationship based upon the data stored in the document database and finally it generates the *FOLLOWS* relationship randomly

# 1 Index

The *id* attributes is common to all nodes and it stores a string which is also used as id by the document database in the *\_id* field, this choice was made so that we could have a way to identify the same object across the different databases with only one string from the application. Due to the OOP language used to build the latter the data was manipulated with classes containing the id of the object so, to improved the performance of the graph database we decided to use the *id* attribute as a index.

Below the information in a json format from the Neo4j DBSM about the new index (some attributes were omitted for space)

```
1  [
2    {
3      "id": 3,
4      "name": "movie_id_index",
5      "type": "RANGE",
6      "entityType": "NODE",
7      "labelsOrTypes": [
8        "Movie"
9      ],
10     "properties": [
11       "id"
12     ]
13   },
14   {
15     "id": 5,
16     "name": "topcritic_id_index",
17     "type": "RANGE",
18     "entityType": "NODE",
19     "labelsOrTypes": [
20       "TopCritic"
21     ],
22     "properties": [
23       "id"
24     ]
25   },
26   {
27     "id": 4,
28     "name": "user_id_index",
29     "type": "RANGE",
30     "entityType": "NODE",
31     "labelsOrTypes": [
32       "User"
33     ],
34     "properties": [
35       "id"
36     ]
37   }
38 ]
```