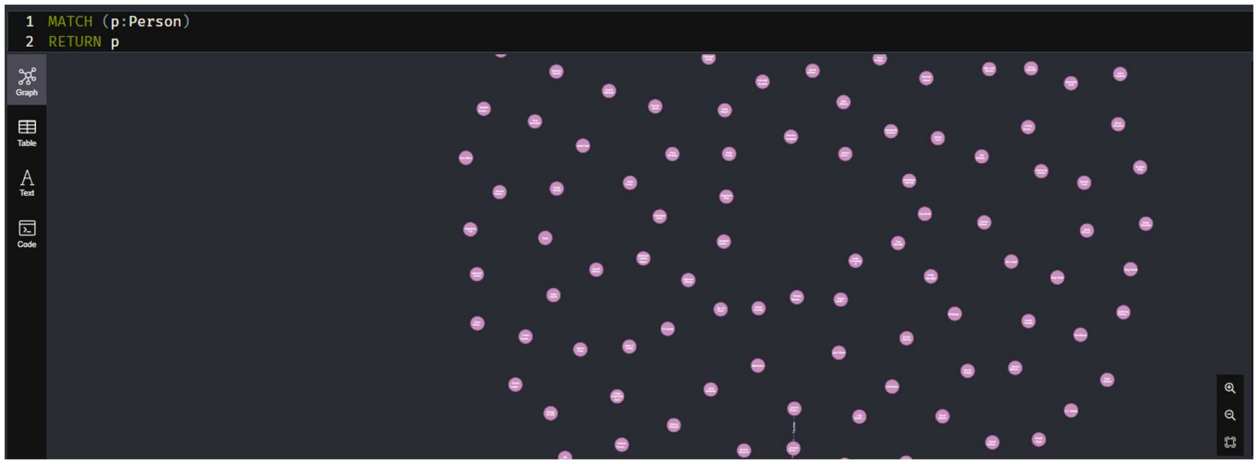
Week 9 Coding Practice

Student: Dipak Bange

Practice on Movie Dataset(sandbox)

1. Retrieve all Person nodes
   MATCH (p:Person)
   RETURN p



2. Retrieve Person nodes that have a born property value of 1970
   MATCH (p:Person {born : 1970})
   RETURN p

3. Retrieve a movie releases in 2003 and tagline "Free your mind"
   MATCH (m:Movie {released : 2003, tagline : 'Free your mind'})
   RETURN m

```
1 MATCH (m:Movie {released : 2003, tagline : 'Free your mind'})
2 RETURN m
```

4. Return property values as a table for a person born in 1965 with two columns name and year
   MATCH (p: Person {born : 1965})
   RETURN p.name, p.born

```
1 MATCH (p: Person {born : 1965})
2 RETURN p.name, p.born
```

	p.name	p.born
1	"Lana Wachowski"	1965
2	"Tom Tykwer"	1965
3	"John C. Reilly"	1965

Started streaming 3 records after 2 ms and completed after 4 ms.

5. Add aliases column headings NAME and YEAR for the query #4
   MATCH (p:Person {born : 1965})
   RETURN p.name as NAME, p.born AS YEAR

```
1 MATCH (p:Person {born : 1965})
2 RETURN p.name as NAME, p.born AS YEAR
```

	NAME	YEAR
1	"Lana Wachowski"	1965
2	"Tom Tykwer"	1965
3	"John C. Reilly"	1965

6. Add aliases column headings NAME OF PERSON and YEAR BORN for the query #4
   MATCH (p: Person {born: 1965})
   RETURN p.name AS ‘NAME OF PERSON’, p.born AS ‘YEAR BORN’

```
1 MATCH (p: Person {born: 1965})
2 RETURN p.name AS `NAME OF PERSON`,  p.born AS `YEAR BORN`
```

	NAME OF PERSON	YEAR BORN
1	"Lana Wachowski"	1965
2	"Tom Tykwer"	1965
3	"John C. Reilly"	1965

7. Examine relationship schema in Movie database + make a screenshot
   CALL db.schema.visualization()

```
neo4j$ CALL db.schema.visualization()
```

nodes

```
[
  {
    "identity": -2,
    "labels": [
      "Movie"
    ],
    "properties": {
      "name": "Movie",
      "indexes": [
        "released"
      ],
      "constraints": [
        "Constraint( id=13, name='constraint_3044d997', type='UNIQUENESS', schema=(:Movie {title}), ownedIndex=9 )"
      ]
    },
    "elementId": "-2"
```

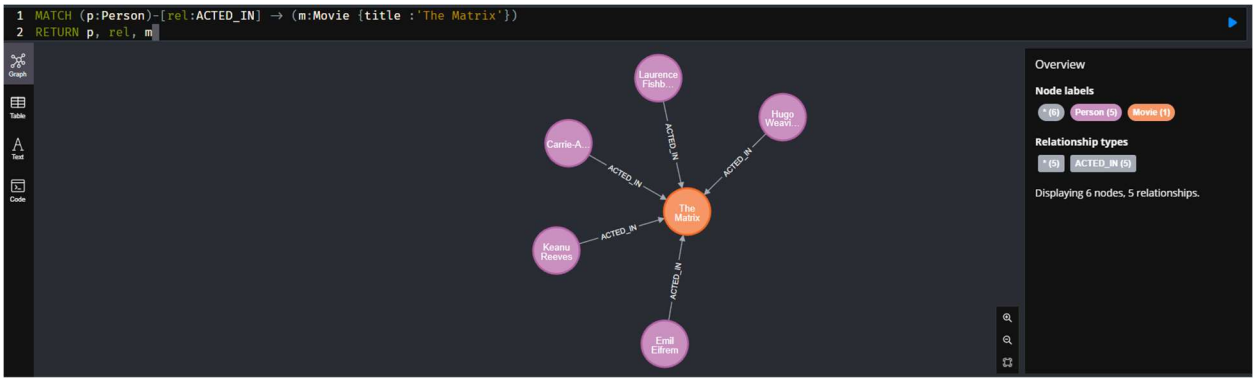
relationships

```
[
  {
    "identity": -1,
    "start": -1,
    "end": -2,
    "type": "ACTED_IN",
    "properties": {

    },
    "elementId": "-1",
    "startNodeElementId": "-1",
    "endNodeElementId": "-2"
  }
  {
    "identity": -6,
```

8. Find a person who acted in the movie "the Matrix" and
   return person, relation, and movie
   MATCH (p:Person)-[rel:ACTED_IN] -> (m:Movie {title : 'The Matrix'})
   RETURN p, rel, m

```
1 MATCH (p:Person)-[rel:ACTED_IN] → (m:Movie {title :'The Matrix'})
2 RETURN p, rel, m
```



9. Retrieve all movies that are connected to Tom Hanks. Note two alternative ways
   of writing the same query.
   MATCH (m: Movie) <-- (p:Person {name : 'Tom Hanks'})
   RETURN m.title
   //or
   MATCH (p : Person {name : 'Tom Hanks'}) --> (m : Movie)
   RETURN m.title

```
MATCH (p : Person {name : 'Tom Hanks'}) ⟶ (m : Movie)
RETURN m.title
```

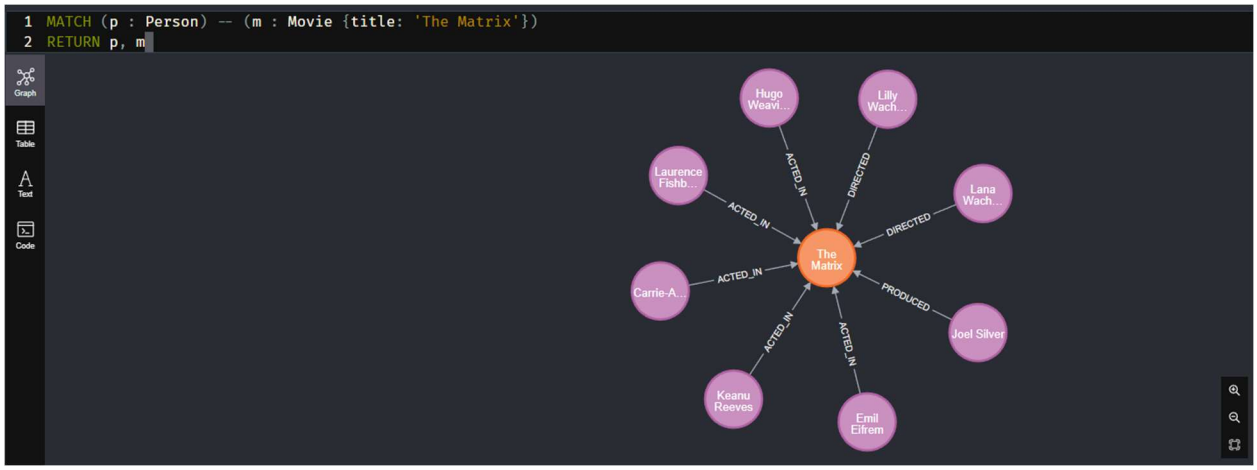
	m.title
1	"Apollo 13"
2	"You've Got Mail"
3	"A League of Their Own"
4	"Joe Versus the Volcano"
5	"That Thing You Do"
6	"The Da Vinci Code"

10. Return all relationships for the title "The Matrix"
    MATCH (p : Person) – [rel] -> (m : Movie {title : 'The Matrix'})
    RETURN p.name, type(rel)

```
MATCH (p : Person) - [rel] → (m : Movie {title : 'The Matrix'})
RETURN p.name, type(rel)
```

	p.name	type(rel)
1	"Emil Eifrem"	"ACTED_IN"
2	"Joel Silver"	"PRODUCED"
3	"Lana Wachowski"	"DIRECTED"
4	"Lilly Wachowski"	"DIRECTED"
5	"Hugo Weaving"	"ACTED_IN"
6	"Laurence Fishburne"	"ACTED_IN"

11. Use anonymous relationship for the title "The Matrix"
    MATCH (p : Person) – (m : Movie {'The Matrix'})
    RETURN p, m



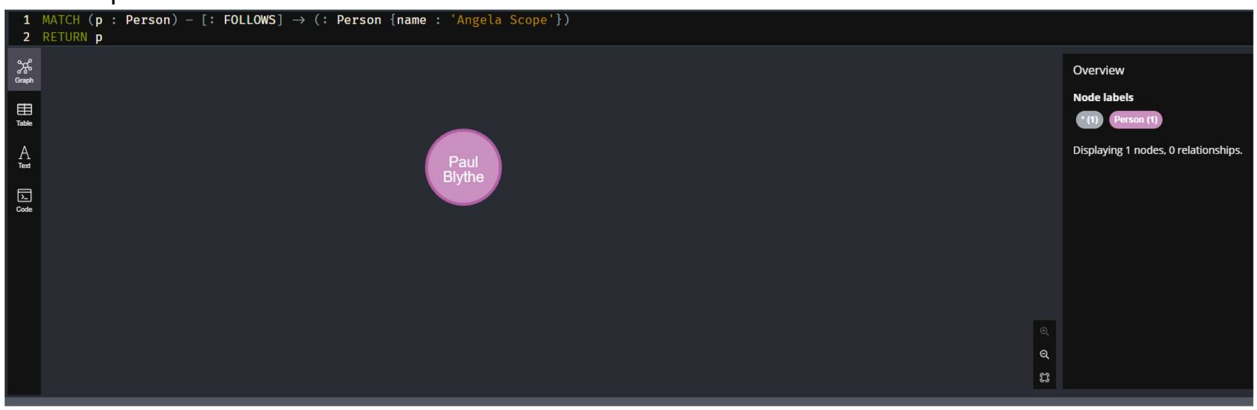
12. Find who reviewed "Cloud Atlas" and return name and rating

    MATCH (p : Person)- [r : REVIEWED] -> (m : Movie {title : 'Cloud Atlas'})

13. Find all nodes (Person) who follow Angela Scope. Note: you can leave empty variable (:Person) if you do not need to return it

    MATCH (p : Person) – [: FOLLOWS] -> (: Person {name : 'Angela Scope'})

    RETURN p



14. Find a person whom Angela Scope follows. Note the change of direction.

    MATCH (p : Person) <- [:FOLLOWS] – (: Person {name : 'Angela Scope'})

    RETURN p


1 MATCH (p : Person) ← [:FOLLOWS] – (: Person {name : 'Angela Scope'})
2 RETURN p
Graph
Table
Text
Code
Jessica Thom…
Overview
Node labels
* (1)
Person (1)
Displaying 1 nodes, 0 relationships.