PRACTICAL-12

AIM:

To perform Graph Path and Connectivity analytics and implement basic queries after loading data using Neo4j.

THEORY:

Neo4j:

• Neo4j is a graph database management system developed by Neo4j, Inc. Described by its developers as an ACID-compliant transactional database with native graph storage and processing, Neo4j is available in a GPL3-licensed open-source "community edition", with online backup and high availability extensions licensed under a closed source commercial license.

• Neo also licenses Neo4j with these extensions under closed-source commercial terms.

• Neo4j is implemented in Java and accessible from software written in other languages using the Cypher query language through a transactional HTTP endpoint, or through the binary "Bolt" protocol.

• In Neo4j, everything is stored in the form of an edge, node, or attribute. Each node and edge can have any number of attributes. Both nodes and edges can be labelled. Labels can be used to narrow searches. As of version 2.0, indexing was added to Cypher with the introduction of schemas. Previously, indexes were supported separately from Cypher.

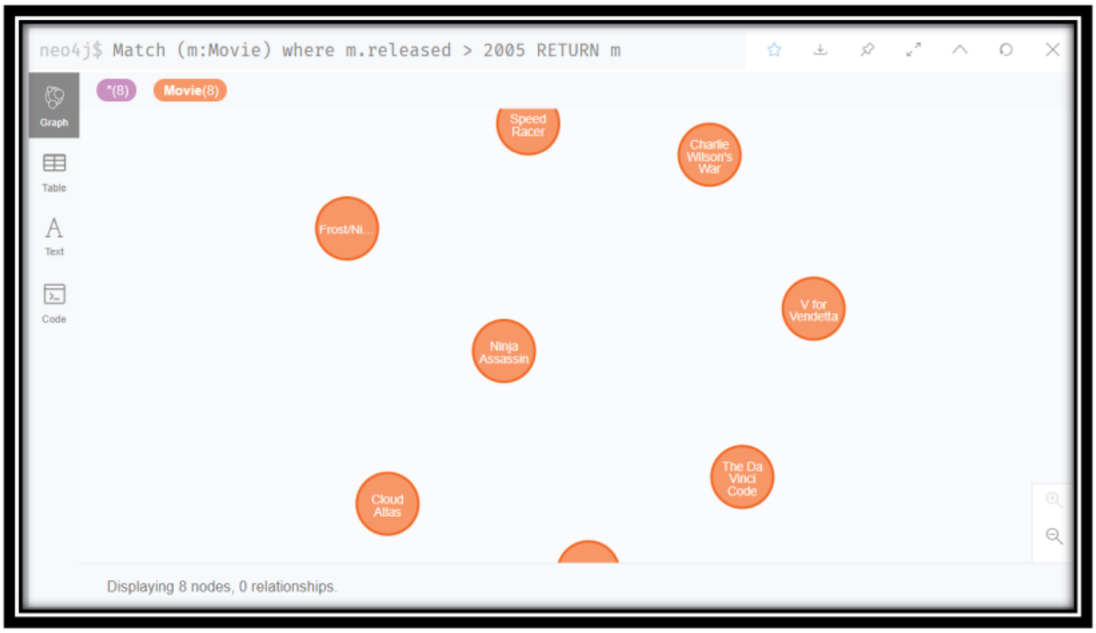
• Neo4j is developed by Neo4j, Inc., based in the San Francisco Bay Area, United States, and also in Malmö, Sweden. The Neo4j, Inc. board of directors consists of Michael Treskow (Eight Roads), Emmanuel Lang (Greenbridge), Christian Jepsen, Denise Persson (CMO of Snowflake), David Klein (One Peak), and Emil Eifrem (CEO of Neo4j).

• Neo4j comes in 2 editions: Community and Enterprise. It is dual-licensed: GPL v3 and a commercial license. The Community Edition is free but is limited to running on one node only due to the lack of clustering and is without hot backups.

• The Enterprise Edition unlocks these limitations, allowing for clustering, hot backups, and monitoring. The Enterprise Edition is available under a closed-source Commercial license.

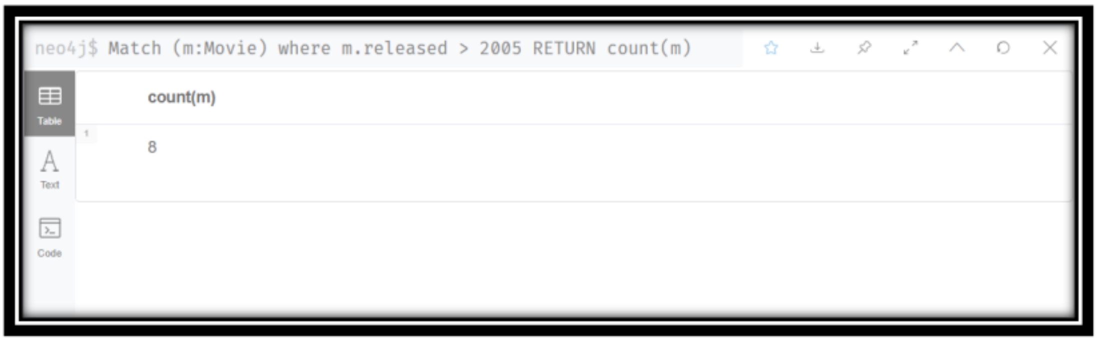
CODE:

Neo4j:



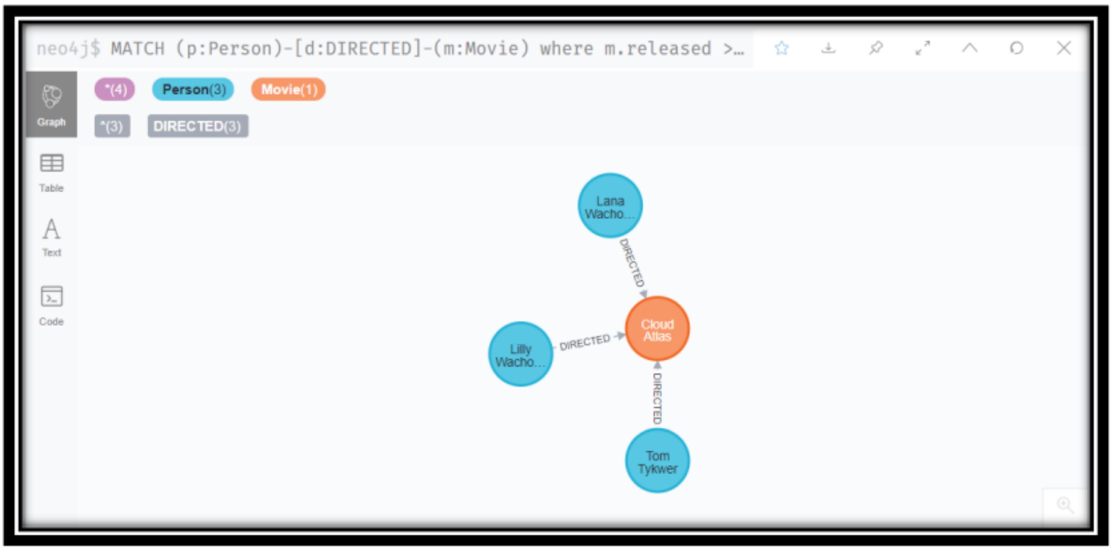
```
neo4j$ Match (m:Movie) where m.released > 2005 RETURN m
```

Displaying 8 nodes, 0 relationships.

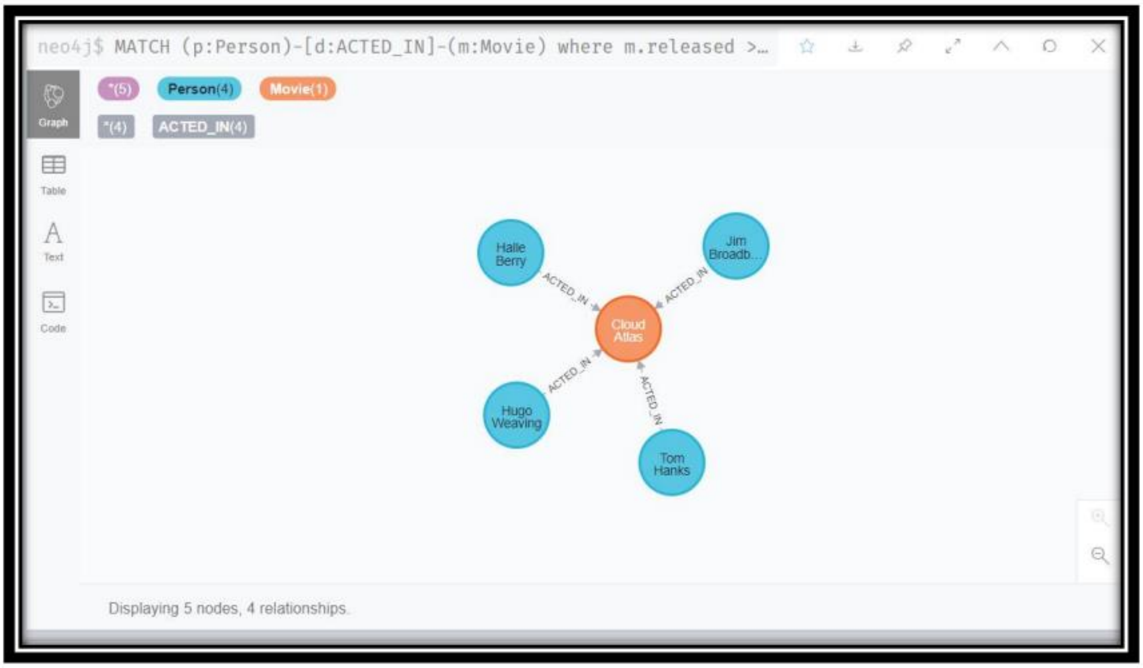


```
neo4j$ Match (m:Movie) where m.released > 2005 RETURN count(m)
```

count(m)
8

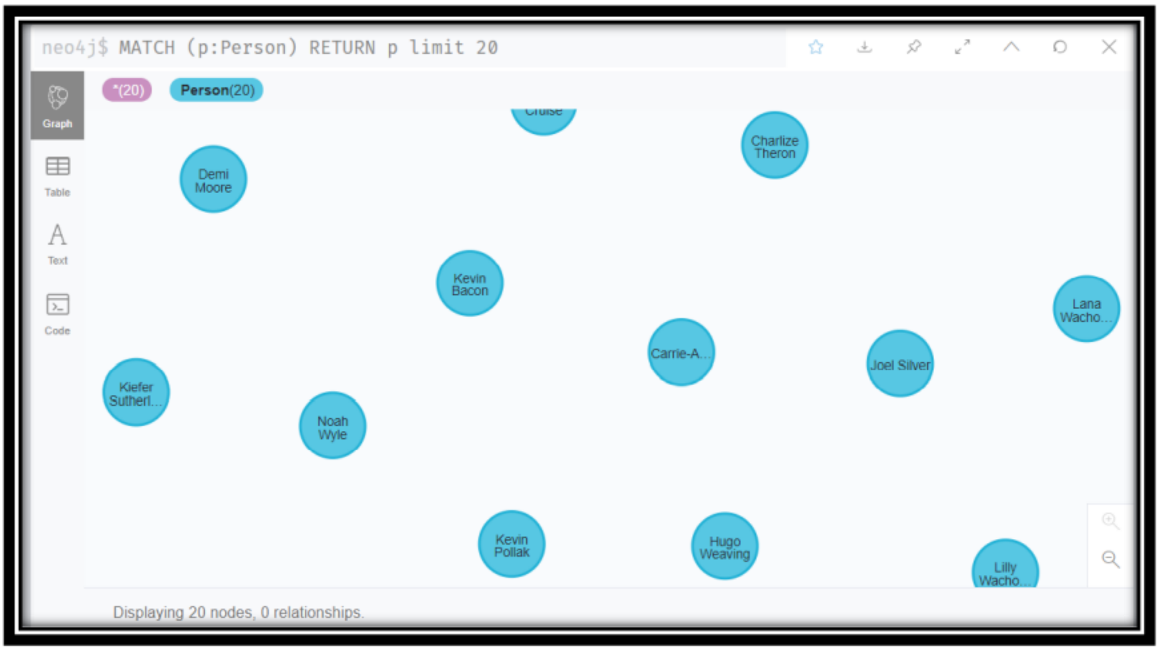


```
neo4j$ MATCH (p:Person)-[d:DIRECTED]-(m:Movie) where m.released >…
```



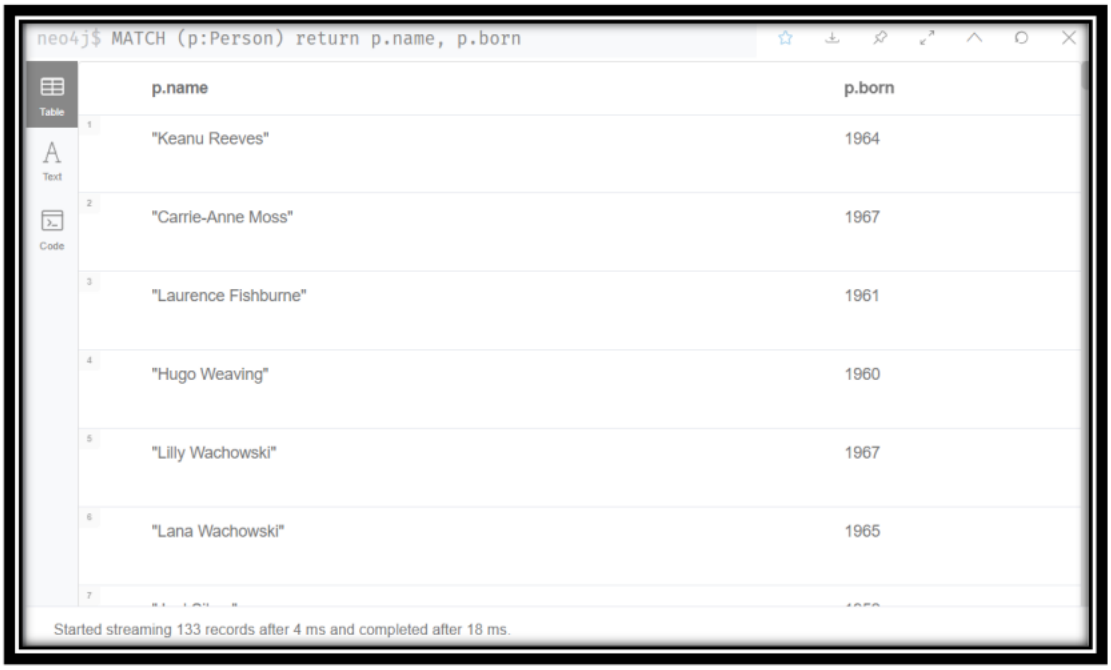
```
neo4j$ MATCH (p:Person)-[d:ACTED_IN]-(m:Movie) where m.released >…
```

Displaying 5 nodes, 4 relationships.



```
neo4j$ MATCH (p:Person) RETURN p limit 20
```

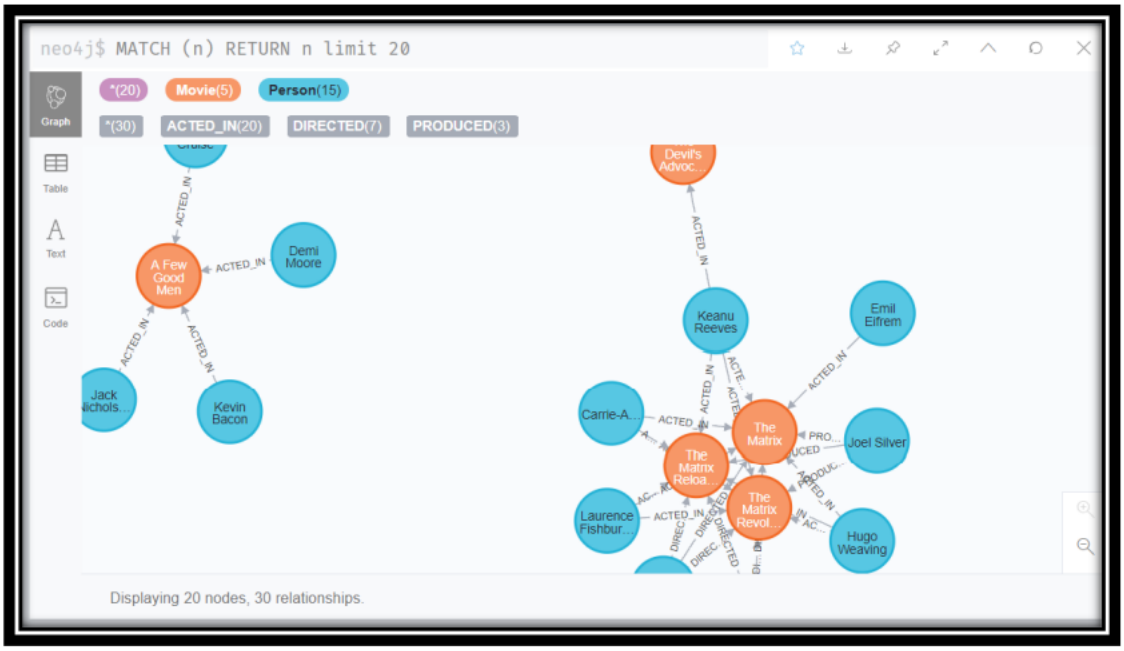
Displaying 20 nodes, 0 relationships.



```
neo4j$ MATCH (p:Person) return p.name, p.born
```

	p.name	p.born
1	"Keanu Reeves"	1964
2	"Carrie-Anne Moss"	1967
3	"Laurence Fishburne"	1961
4	"Hugo Weaving"	1960
5	"Lilly Wachowski"	1967
6	"Lana Wachowski"	1965

Started streaming 133 records after 4 ms and completed after 18 ms.



```
neo4j$ MATCH (n) RETURN n limit 20
```

Displaying 20 nodes, 30 relationships.

CONCLUSION:

In this practical, we learnt about neo4j and used different type of queries on sandbox on movie dataset.