

NoSQL 2
Neo4J

Neo4J

- Graph Oriented Database
- Good for managing data with a rich set of relationships

Neo4J Terminology

- Node
- Properties
- Labels
- Relationships

Neo4J Query Language

Cypher

Create nodes

- CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
- CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
- CREATE (AndyW:Person {name:'Andy Wachowski', born:1967})
- CREATE (TomH:Person {name:'Tom Hanks', born:1956})

Neo4J Query Language

Cypher

Create Relationships

- `CREATE (Keanu)-[:ACTED_IN {roles:['Neo']}]>(TheMatrix)`
- `CREATE (AndyW)-[:DIRECTED]>(TheMatrix)`
- `CREATE (TomH)-[:ACTED_IN {roles:['Joe Fox']}]>(YouveGotMail)`

Neo4J Query Language

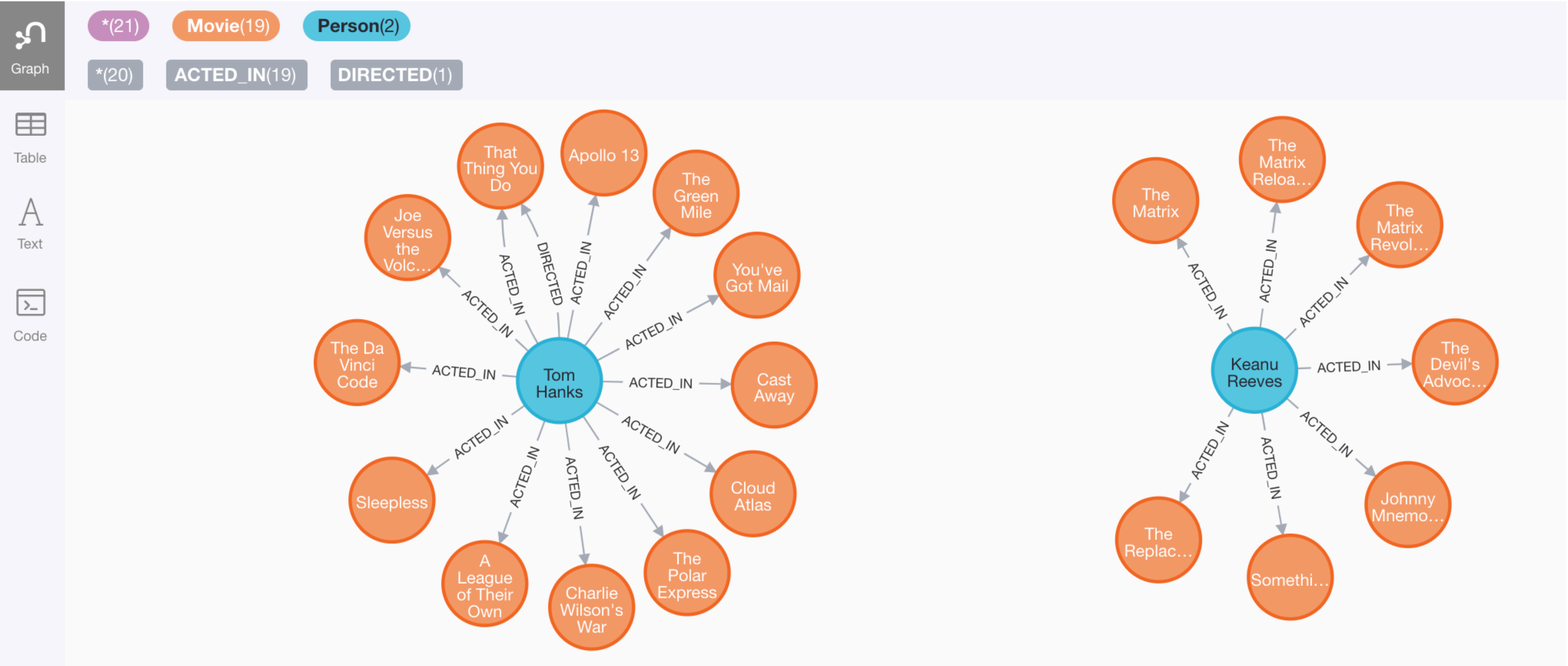
Cypher

Queries

- MATCH (movie:Movie) RETURN movie
- MATCH (person:Person) RETURN person
- MATCH (person:Person) RETURN person.name
- MATCH (m:Movie)-[:ACTED_IN]-(a:Person) WHERE a.name = 'Tom Hanks' Return m.title as movie

MATCH (m:Movie)<-[:ACTED_IN]-(a:Person) WHERE a.name = 'Tom Hanks' or a.name = 'Keanu Reeves'
Return m, a

```
neo4j$ MATCH (m:Movie)←[:ACTED_IN]-(a:Person) WHERE a.name = 'Tom Hanks' or a.name = 'Keanu ...
```



Displaying 21 nodes, 20 relationships.