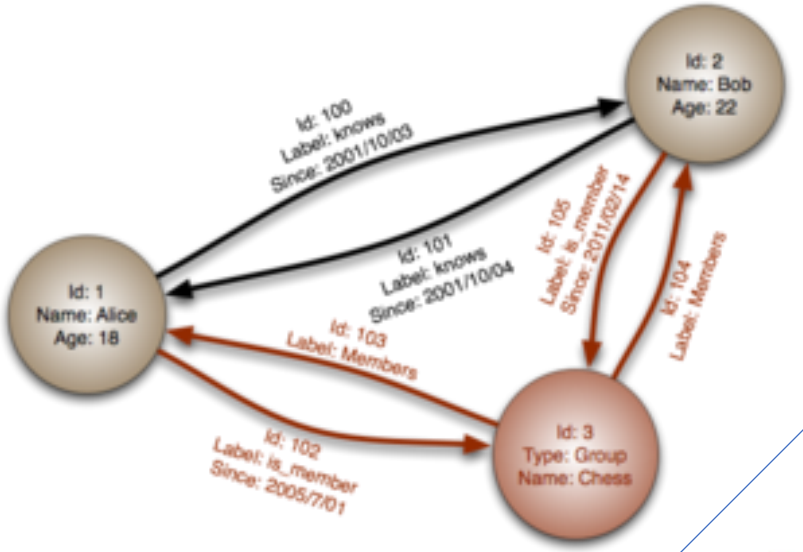
Graph Databases

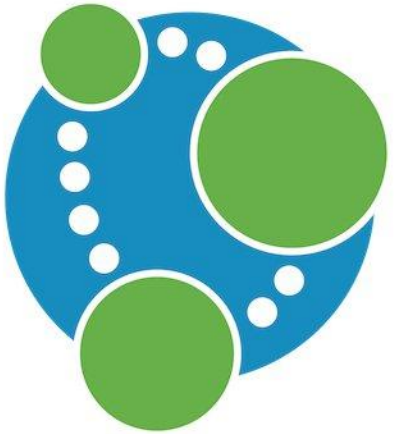
Michael Enudi

Journey through the world of databases and data engineering

Scope

- Introduction to Graph Databases.
- Movielens in a Graph
- Neo4J
- Cypher
- Performing CRUD in Neo4J
- Data Analysis in Neo4J
- Graph Databases: Wrap Up.



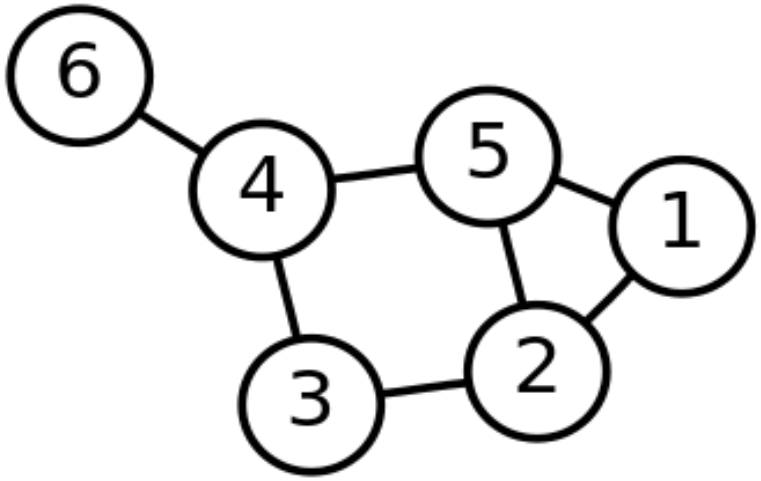


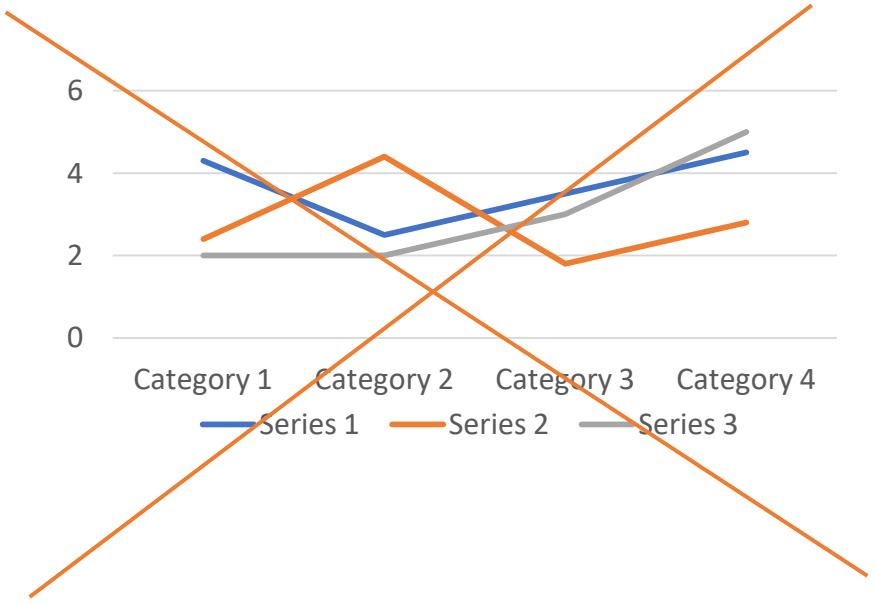
What are Graphs??

An abstract representation of a set of objects and their relationships

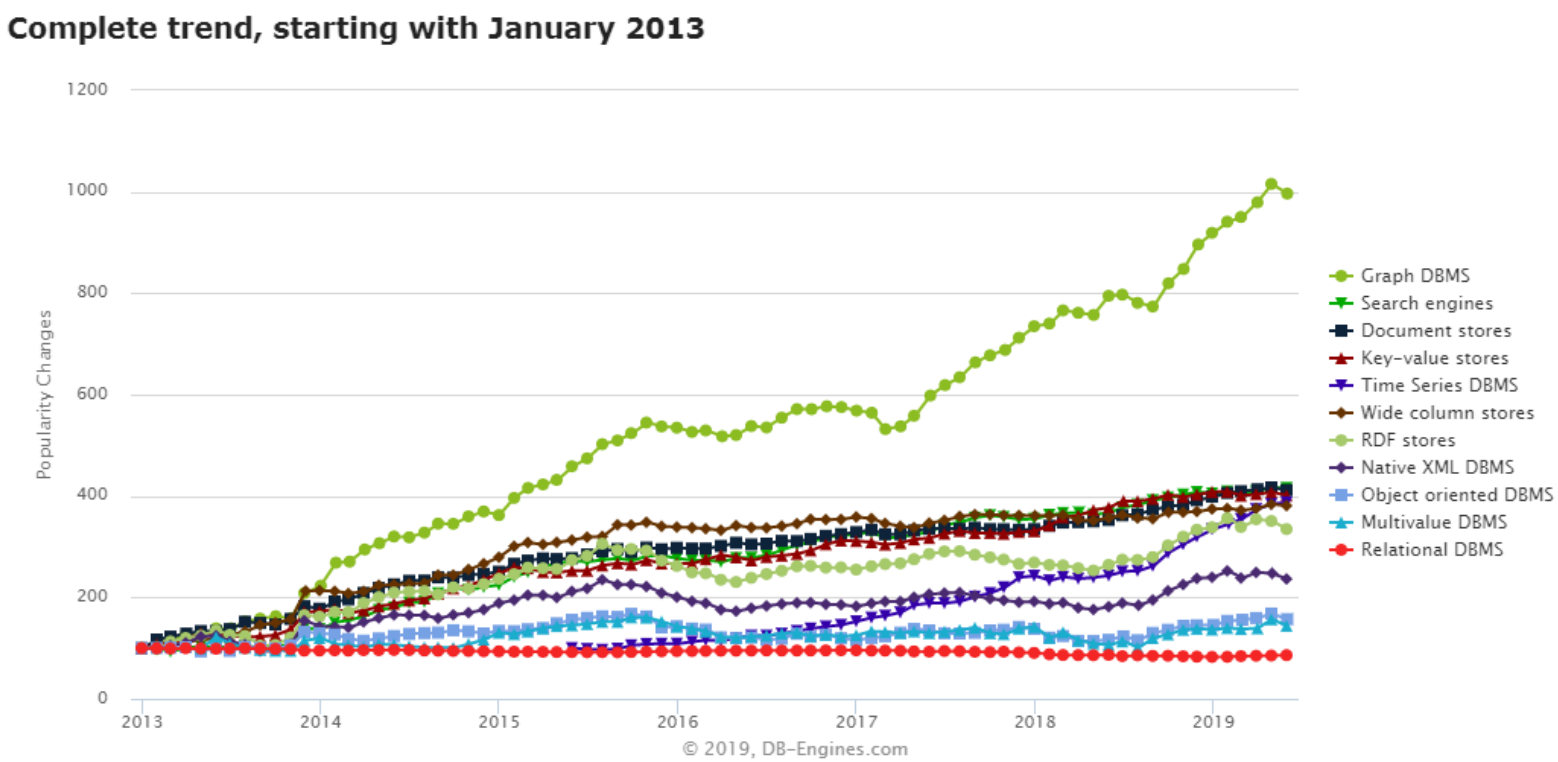
It is based off an already established subject in mathematics called graph theory.

The objects or entities are called Vertex/Vertices or Node/Nodes while the relationship are called Edge/Edges or Relationship/Relationships.

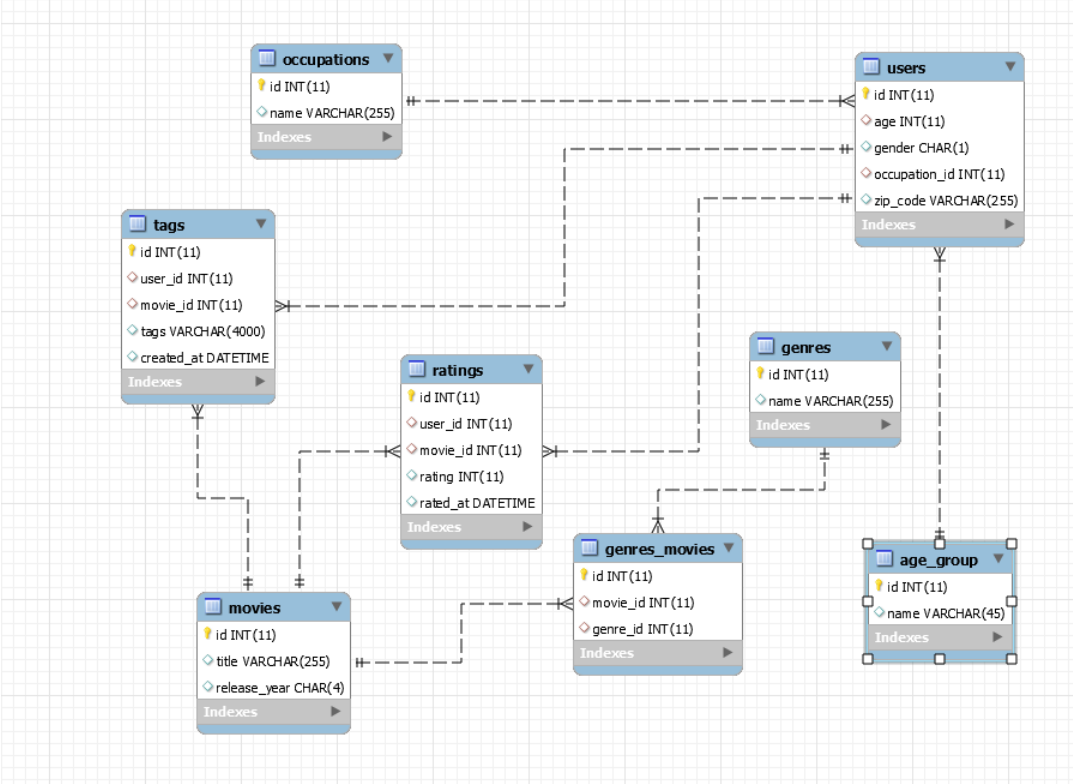




Graph databases are the family of NoSQL databases that are optimized to perform CRUD and analytical operations or data modelled as graphs



Why are RDBMS/other NoSQL DBs not enough



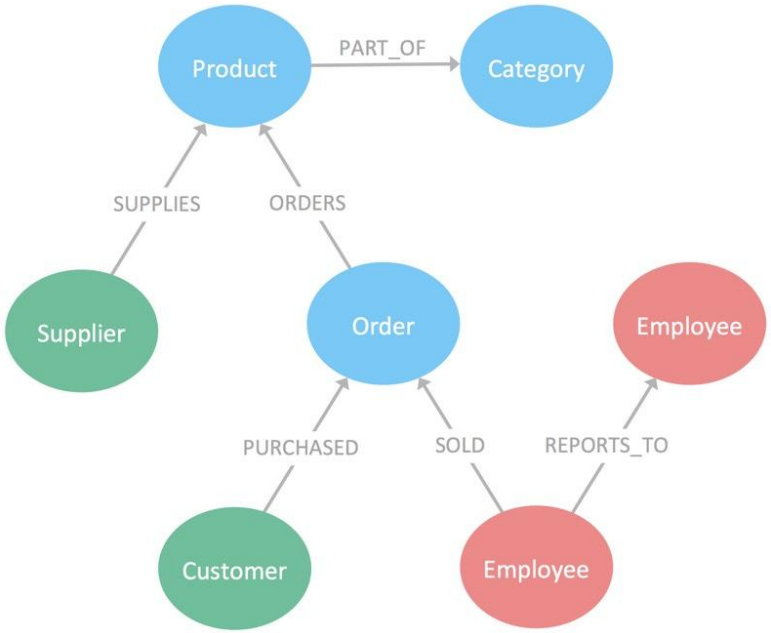
- Relational databases (MySQL, PostgresSQL)
- Document-oriented databases (MongoDB, CouchDB)
- Key-value stores (Memcache, Redis)
- Columnar (Bigtable-like) databases (HBase, Cassandra)

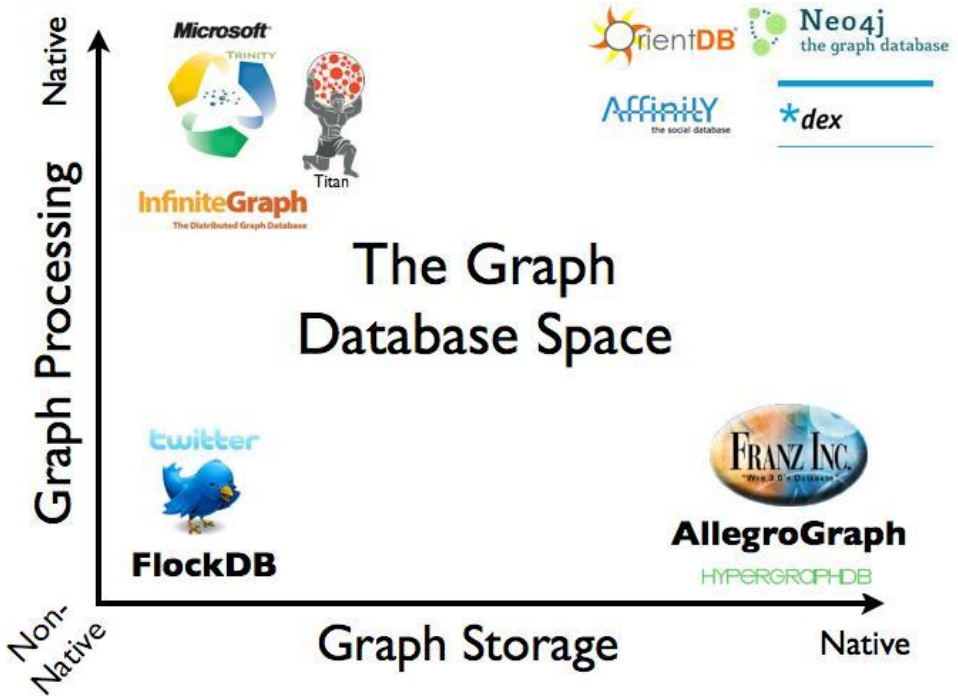
- ❑ Relationships were an after-thought.
- ❑ Using various implementation of linking or relationship definitions (eg. ObjRef in MongoDB or Foreign-keys in RDBMS) will perform badly when performing large graph transversal.

Graph DBMS

They

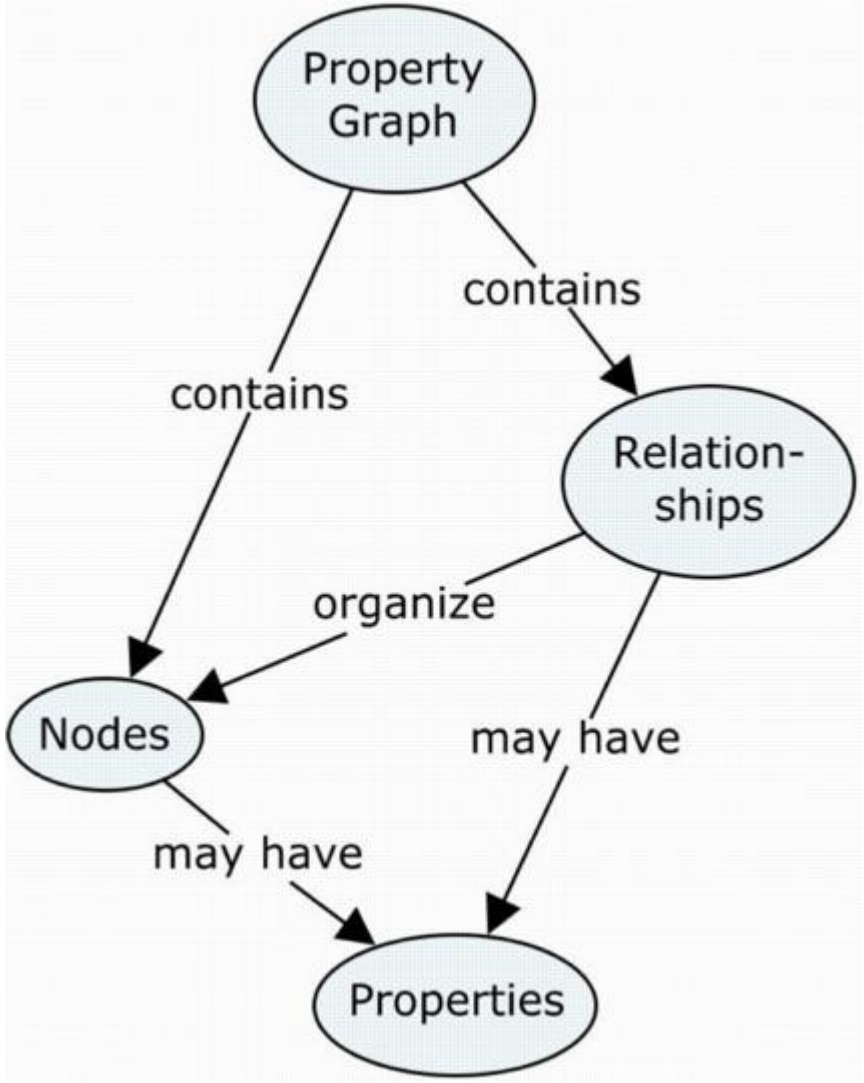
- Storage
- Progressing platform





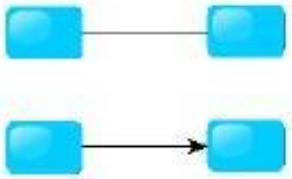
Graph Model

1. Nodes are used to represent entities or actors.
2. Relationships denotes the connection between nodes or actors. The word “connection” should be used in the context of the domain. It could be an action that describes what nodes do or could be an adjective that qualifies the nodes or a grouping for that node.
3. The context of a relationship determines if it should be a directed or undirected relationship.
4. Nodes and relationships can contain properties.
5. Sometimes, nodes can refer to themselves – self loop.
6. A node can also not have relationships.

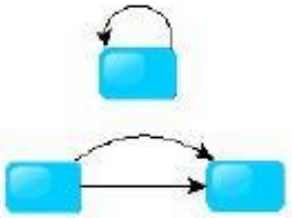


Different Kinds of Graphs

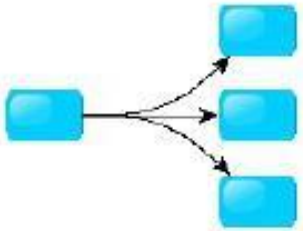
- Undirected Graph
- Directed Graph



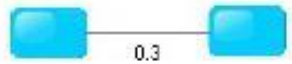
- Pseudo Graph
- Multi Graph



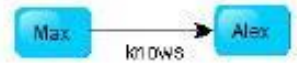
- Hyper Graph



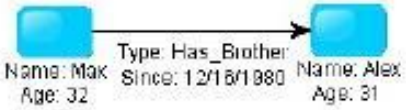
- Weighted Graph



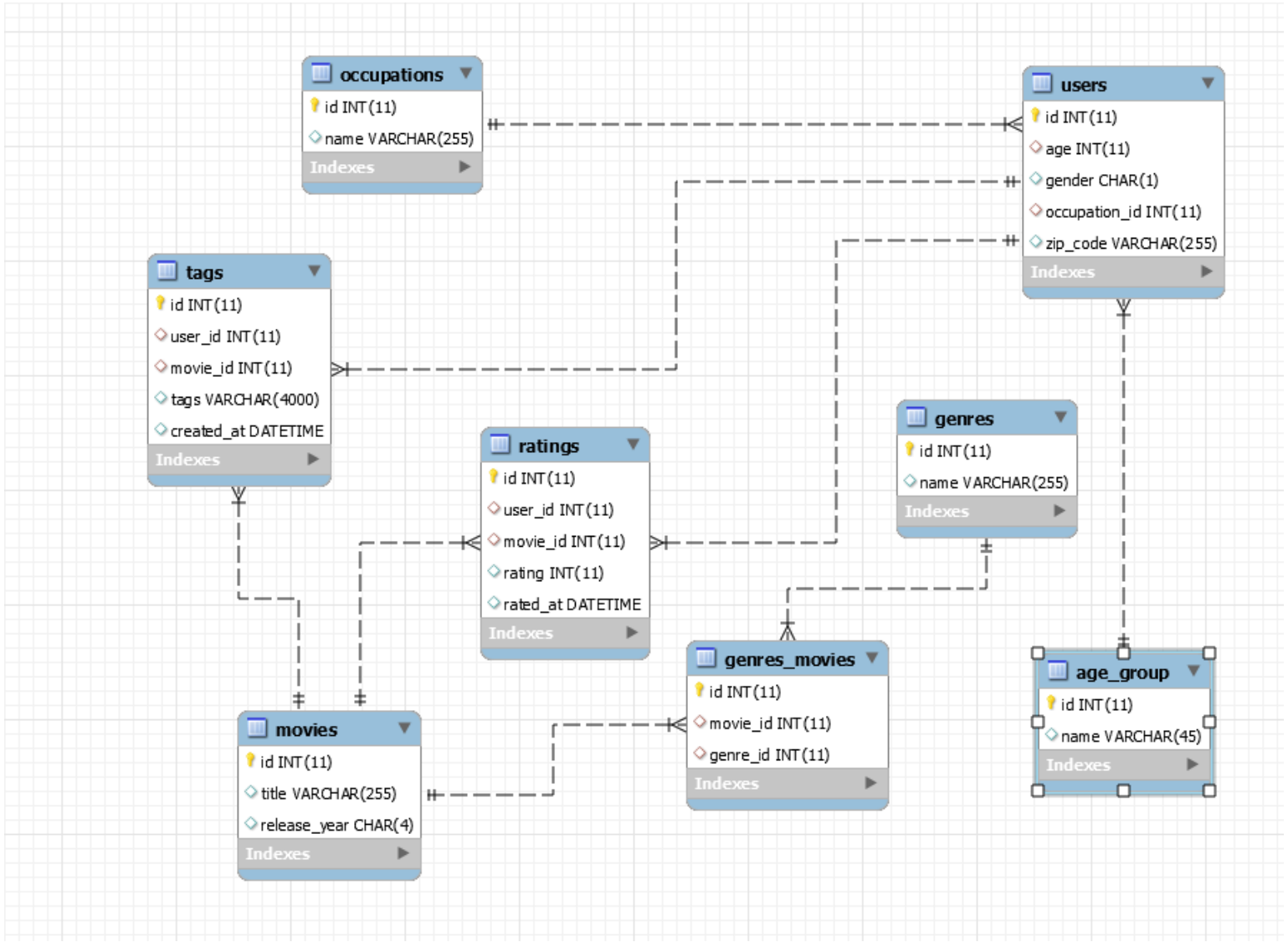
- Labeled Graph



- Property Graph



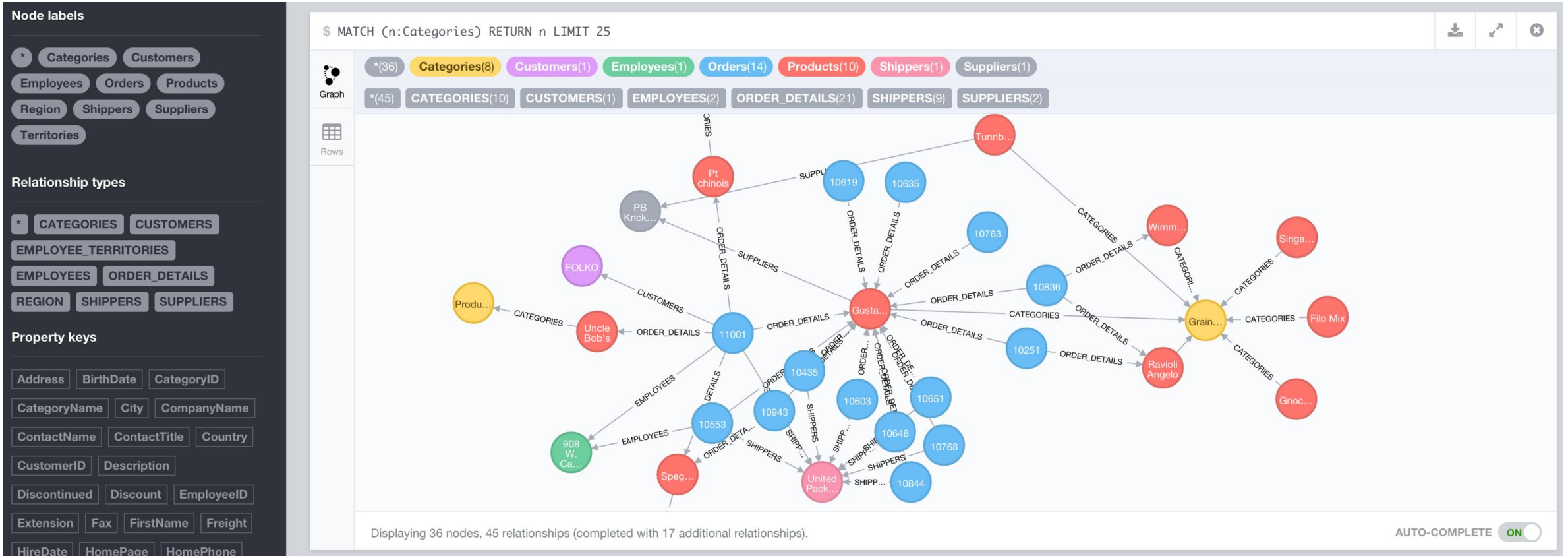
Movielens


occupations
id INT(11)
name VARCHAR(255)
Indexes
users
id INT(11)
age INT(11)
gender CHAR(1)
occupation_id INT(11)
zip_code VARCHAR(255)
Indexes
tags
id INT(11)
user_id INT(11)
movie_id INT(11)
tags VARCHAR(4000)
created_at DATETIME
Indexes
ratings
id INT(11)
user_id INT(11)
movie_id INT(11)
rating INT(11)
rated_at DATETIME
Indexes
genres
id INT(11)
name VARCHAR(255)
Indexes
genres_movies
id INT(11)
movie_id INT(11)
genre_id INT(11)
Indexes
age_group
id INT(11)
name VARCHAR(45)
Indexes
movies
id INT(11)
title VARCHAR(255)
release_year CHAR(4)
Indexes

Neo4J

- A Native Graph Database with Lucene Index
- Labelled, Property, multi-graph
- ACID-compliant
- Well over 64 Billion Nodes, 64 Billion Relationships
- High Availability (with Enterprise Edition)
- Embedded Server
- Bolt protocol or REST API
- Developed by Neo Technology
- Current version is 3.5.8
- Nodes and relationships are labelled property objects
- High profile users include Walmart, Cisco and eBay, NASA

Neo4J Web Console



Cypher

Cypher is an expressive (yet compact) graph database query language.

```
(emil)<-[:KNOWS]-(jim)-[:KNOWS]->(ian)-[:KNOWS]->(emil)
```

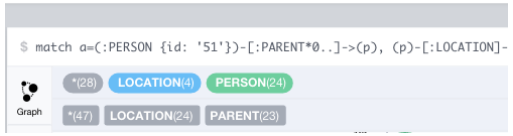
Like SQL, cypher is made up of clauses, statements, functions and expressions.

Examples of clauses include MATCH, RETURN, WHERE, CREATE, CREATE UNIQUE, MERGE, DELETE, SET, FOREACH, UNION, WITH, etc.

```
MATCH (movie:Movie)
WHERE coalesce(movie.genres, "-") <> "-"
WITH SPLIT(movie.genres, "|") as parts, movie as m
UNWIND parts as x
MATCH (g: Genre {name: x})
MERGE (m)-[:IS_A]->(g)
REMOVE m.genres;
```

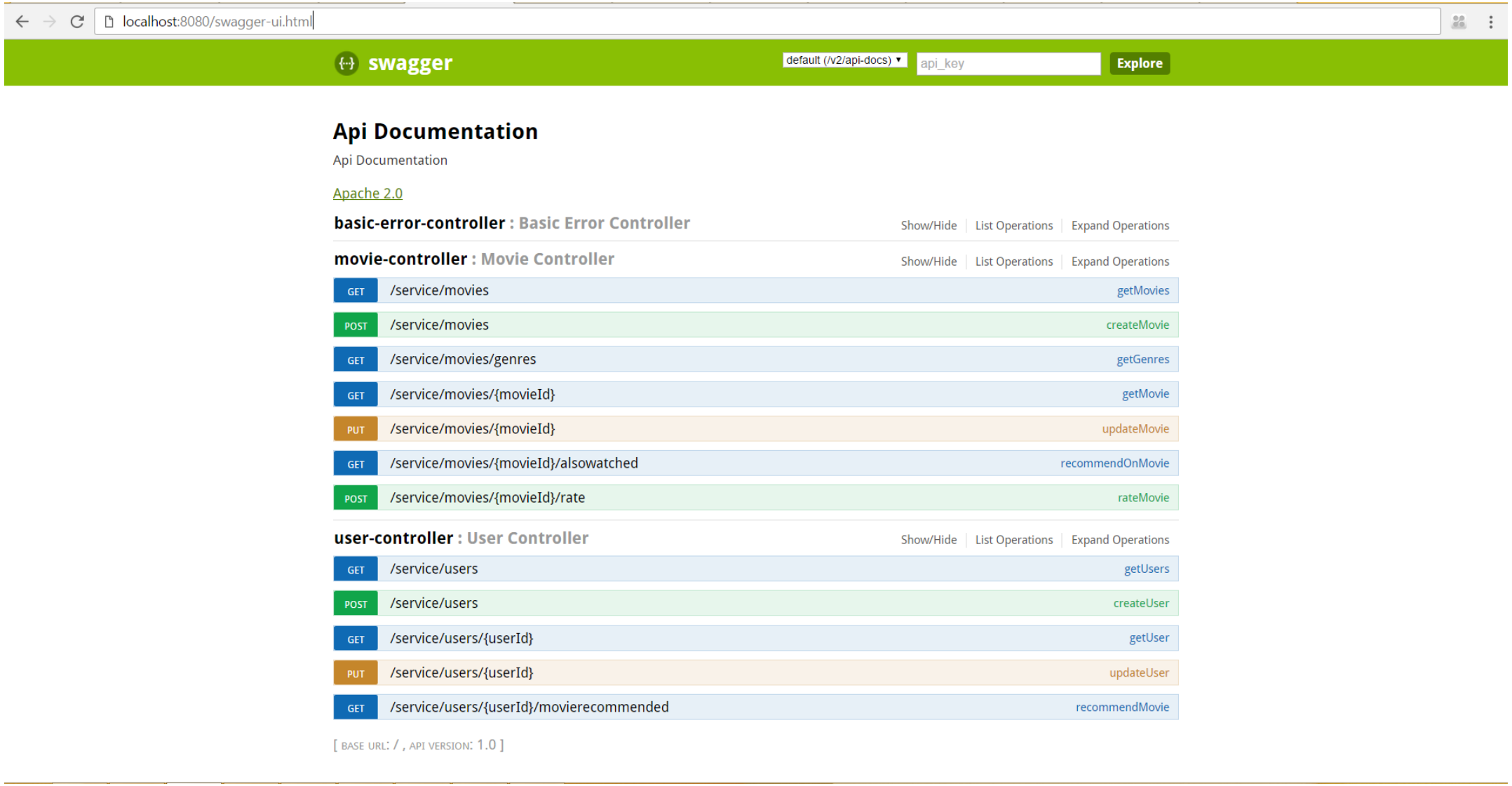
Cypher

```
match a=(:PERSON {id: '51'})-[:PARENT*0..]->(p),
    (p)-[:LOCATION]->(l)
where all(n in nodes(a)[0..-1] where n.gender=0)
return p, l
```



Clause	Description
CREATE	creates nodes and relationship
RETURN	returns projected nodes, relationship or their properties to the user.
MATCH	provides a predicate that Neo4J should evaluate as true. It is mostly used to retrieve a starting node or nodes in a Cypher query.
SET	writes or updates the property of a node or relationship.
REMOVE	removes a node or relationship property
MERGE	Provides a construct to create a node or relationship if it does not exists or do something else if it does
WITH	Provides a way to pipeline results from one part of a cypher query to another part. It is projection without sending the results to the end user
WHERE	filters records that that has been returned from a prior pattern matching
FOREACH	iterates and executes an operation on each item in a collection
DELETE	deletes a node or relationship

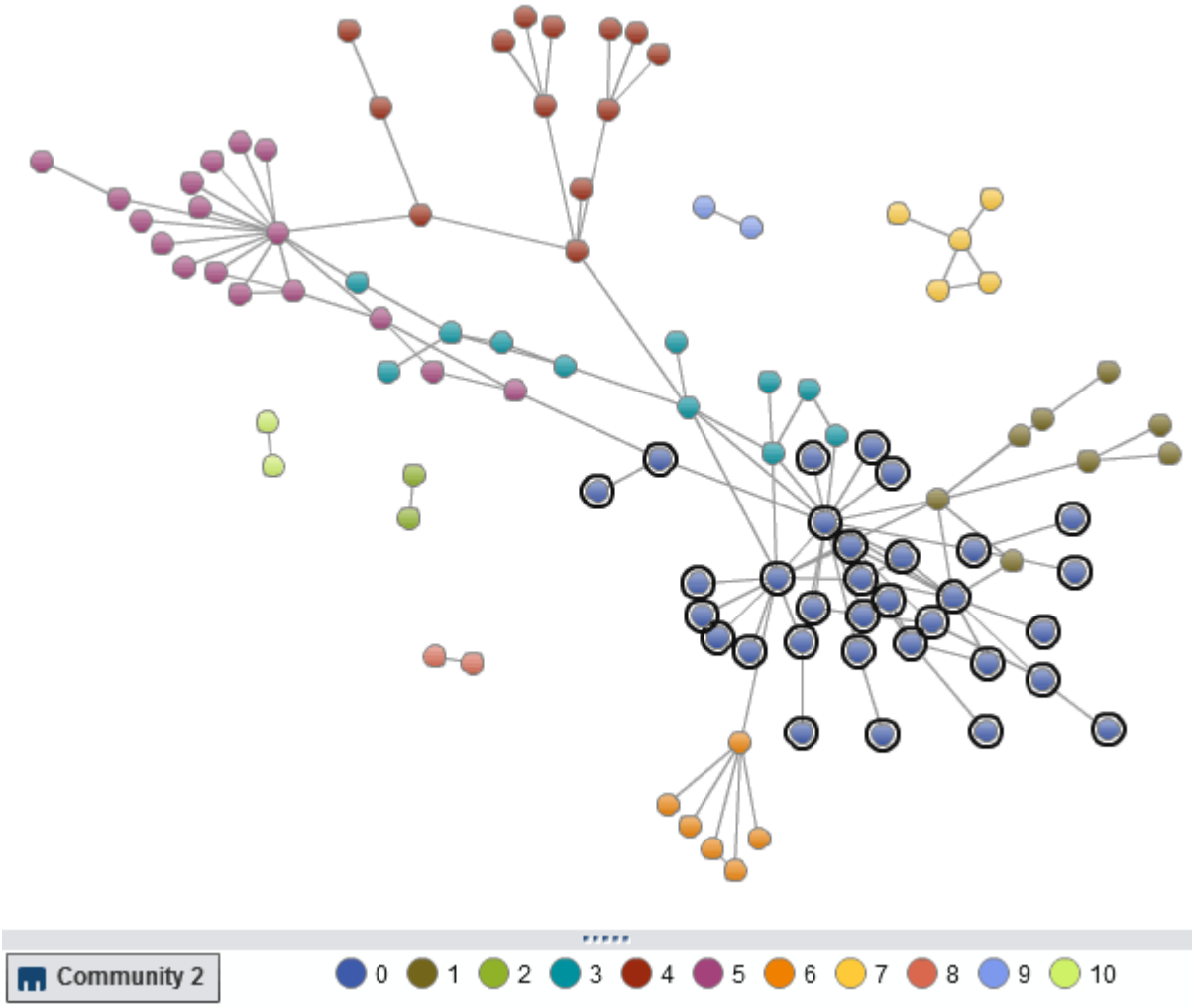
Movielens Restful Web Service



Graph Analytics

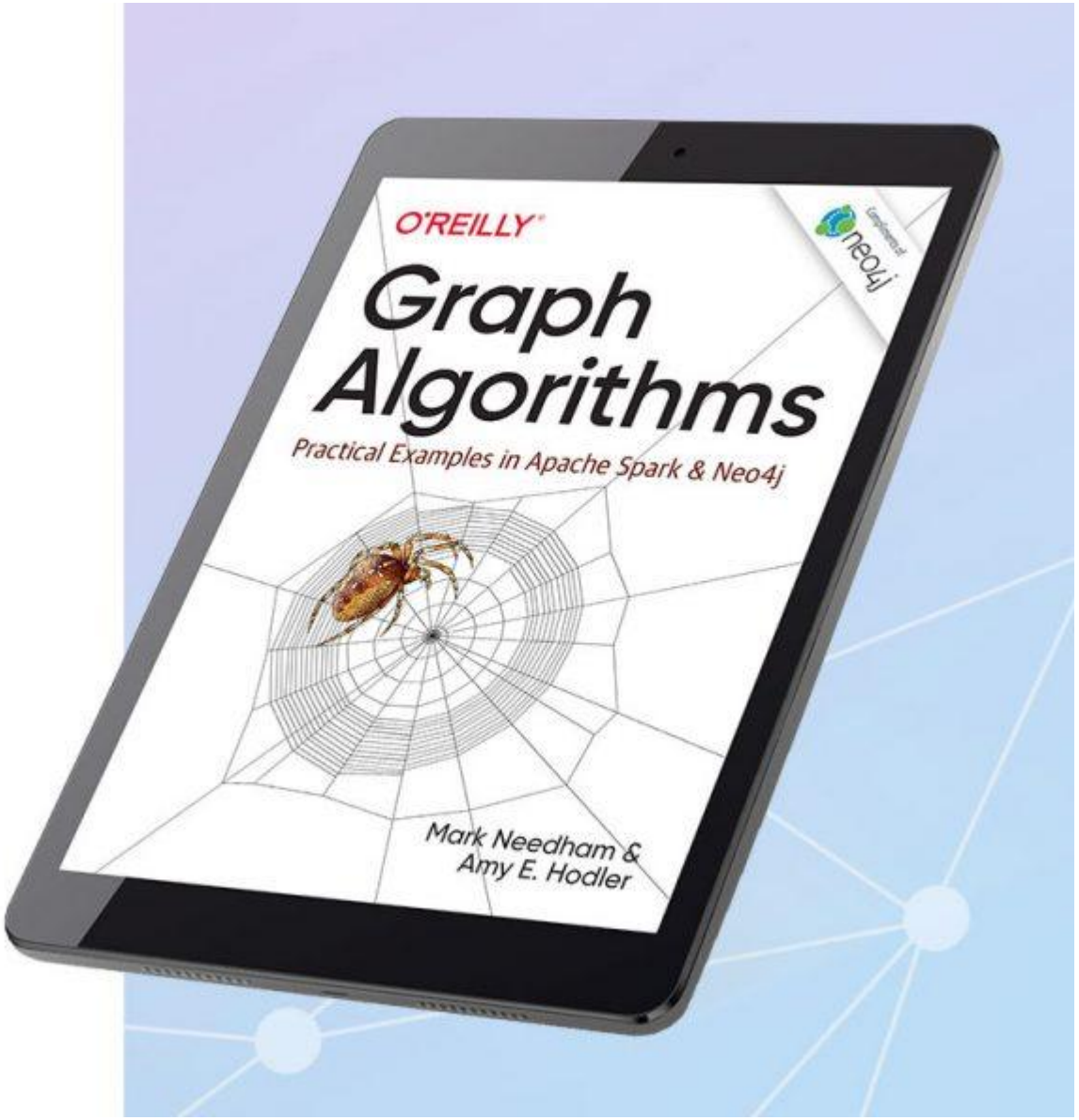
Using graph algorithms to explain, discover patterns, measure strengths and direction in a graphs.

- Search
- Path-finding
- Community detection
- Propagation
- Centrality
- Similarity



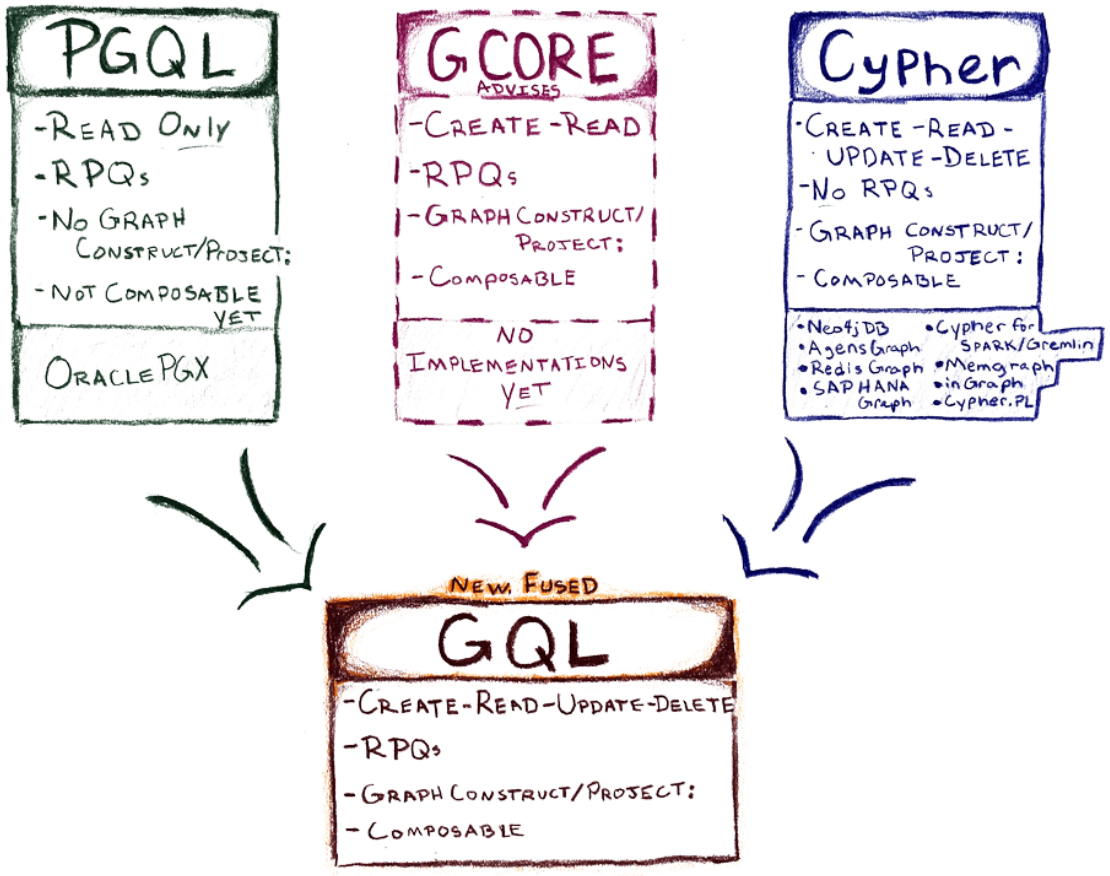
Graph Algorithms

Examples in Cypher and Spark's GraphFrames



Graph Database Wrap-up

A unified graph query language



openCypher

And support openCypher

neo4j

structr

neo4j

Spark

PURETHINK

ORACLE

databricks

+ableau

LINKURIOUS

graphgrid

Information Analysis Incorporated

Tom Sawyer Software

GRAPHENEDB

xnlogic

GraphAware

Graph Story

ThoughtWorks

Cypher for Spark

Neo4J and Big data pipeline

- Neo4j-Spark-Connector
- Neo4j MongDB connector
- Neo4J Cassandra connector
- Neo4j Elastic search connector
- …..

- ✓ Sometimes, a some task are better processed outside the databases because of either the complexity or we require integration with data from other sources.
- ✓ Neo4J has connectors to load and store data to big data processing platforms like Apache Spark.
- ✓ It can be used as a store for final or intermediate result of graph or even non-graph processing frameworks.
- ✓ Also, Neo4J can be integrated to other NoSQL database to change the model for a totally different goal and perspective.