Objectives

Graph Databases

- Introduction to GraphDB
- Neo4j

NoSQL Databases

What is it

- Not Only NoSQL
- Non Relational Database system
- Relaxation in ACID properties (Atomicity, Consistency, Isolation, Duarbility)
- Uses CAP theorem (Consistency, Availability, Partition Tolerance)

Types

- Document databases
- Key-value stores
- Column-oriented databases
- Graph databases

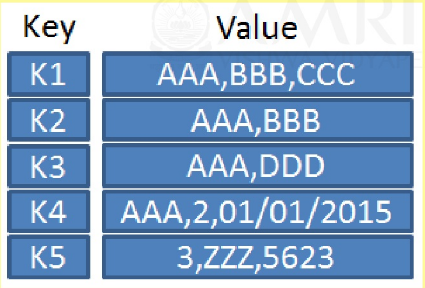
Graph Databases

What is it

- A database with an explicit graph structure
- Each node knows its adjacent nodes
- Plus an Index for lookups

Key-Value Stores

- Simplest form of database management systems.
- They store pairs of keys and values as well as retrieve values when a key is known.



Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Examples

(twitter.com) Tweet id ⇒ information about tweet

(amazon.com) Item number ⇒ information about it

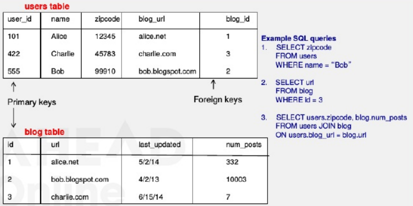
(kayak.com) Flight number ⇒ information about flight, e.g., availability

(yourbank.com) Account number ⇒ information about it

Key-Value Stores..

- It's a dictionary datastructure.
  - Insert, lookup, and delete by key
  - E.g., hash table, binary tree
- But distributed
- Key-Value stores reuse many techniques from DHTs

Examples



users table

user_id	name	zipcode	blog_url	blog_id
101	Alice	12345	alice.net	1
422	Charlie	45783	charlie.com	3
555	Bob	99910	bob.blogspot.com	2

Primary keys
Foreign keys

blog table

id	url	last_updated	num_posts
1	alice.net	5/2/14	332
2	bob.blogspot.com	4/2/13	10003
3	charlie.com	6/15/14	7

Example SQL queries

```
1.  SELECT zipcode
    FROM users
    WHERE name = "Bob"

2.  SELECT url
    FROM blog
    WHERE id = 3

3.  SELECT users.zipcode, blog.num_posts
    FROM users JOIN blog
    ON users.blog_url = blog.url
```

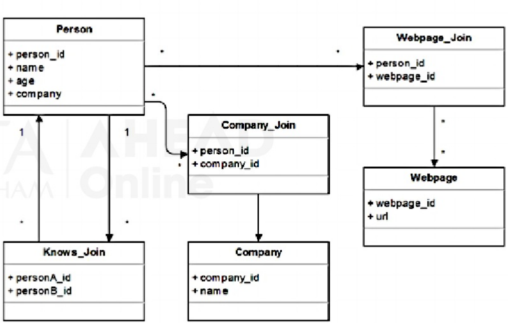
Tradtional RDBMS..

- Data stored in tables and schema-based, i.e., structured tables
- Primary key that is unique within that table
- Queried using SQL and supports joins

Graph Databases ..

RDBMS Vs GraphDB

- Graph Operations are hard to implement in relational database
- Excution of graph operations are inefficient in RDBMS
- Any meaningful work that requires traversals would require the relational database to execute a lot of joins



Person
+ person_id
+ name
+ age
+ company

Webpage_Join
+ person_id
+ webpage_id

Company_Join
+ person_id
+ company_id

Webpage
+ webpage_id
+ url

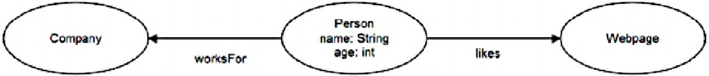
Knows_Join
+ personA_id
+ personB_id

Company
+ company_id
+ name

Graph Databases ..

RDBMS Vs GraphDB

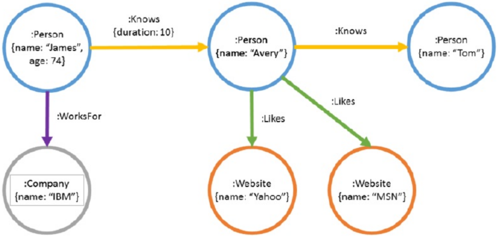
- The circles represent nodes, and the solid lines represent relationships.
- We need only create a new edge from the Person node to the Company node.
- Do not have to execute joins for each edge traversal



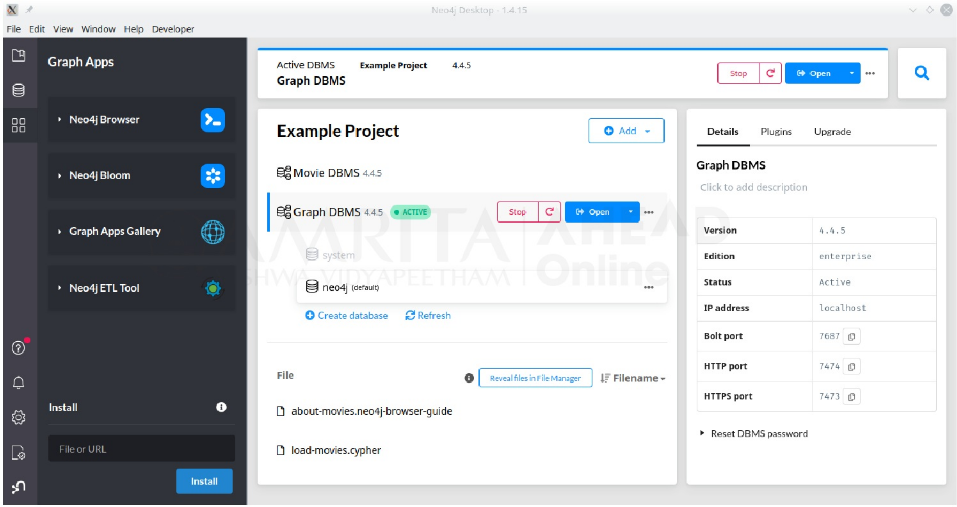
Neo4j Graph Database

Graph Model

- The Neo4j graph is composed of nodes and edges, with an unlimited number of edges between nodes.
- Nodes and edges can have properties, which are key-value pairs.
- They can also be given labels, which define the type of each node or edge.
- We can add additional constraints to the schema like uniqueness.

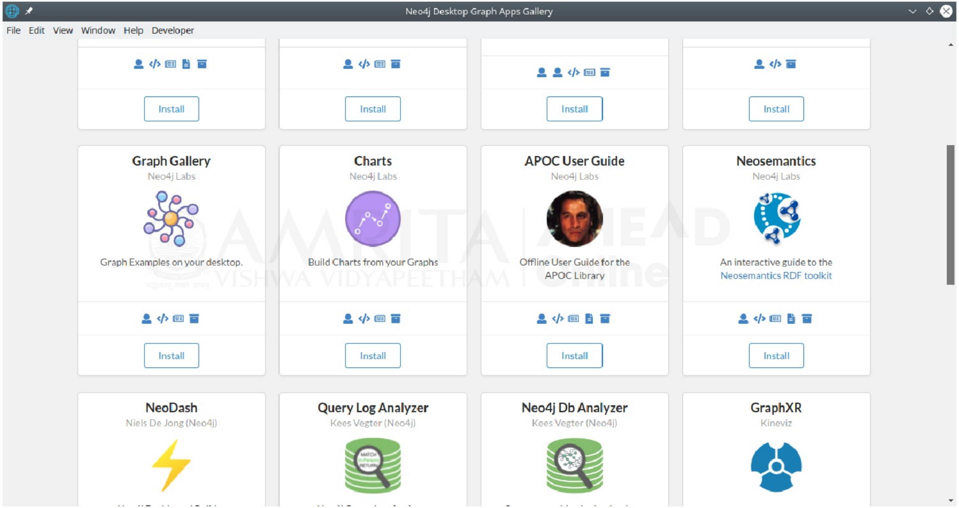


NEO4J DESKTOP



Version	4.4.5
Edition	enterprise
Status	Active
IP address	localhost
Bolt port	7687
HTTP port	7474
HTTPS port	7473

Neo4j Desktop: Graph Apps



Graph Gallery
Neo4j Labs
Graph Examples on your desktop.

Charts
Neo4j Labs
Build Charts from your Graphs

APOC User Guide
Neo4j Labs
Offline User Guide for the APOC Library

Neosemantics
Neo4j Labs
An interactive guide to the Neosemantics RDF toolkit

NeoDash
Niels De Jong (Neo4j)

Query Log Analyzer
Kees Vegter (Neo4j)

Neo4j Db Analyzer
Kees Vegter (Neo4j)

GraphXR
Kineviz

Neo4j-NetworkX

```
// from local server
from neo4j import GraphDatabase
driver = GraphDatabase.driver("bolt://localhost:7687",
          auth=("user", "password"))
```

Neo4j Options

- Neo4j Desktop
- Neo4j Sandbox
- Neo4j Aura
- Neo4j Enterprise

```
import nxneo4j as nx2
G1 = nx2.DiGraph(driver)
G2 = nx2.Graph(driver)
```

```
// from neo4j sandbox
from neo4j import GraphDatabase, basic_auth
driver = GraphDatabase.driver(
    "bolt://54.174.242.100:36186",
    auth=basic_auth("neo4j", "invention-airship-gunnery"))
```

NEO4J-NETWORKX ..

```
G1.add_node(1)
G1.add_nodes_from([2, 3, 4])
G1.add_edge([1, 2])
```

```
G2.add_edges_from([(’A’, ’C’),(’B’, ’A’),(’B’, ’C’), (’B’, ’D’),
(’B’, ’E’), (’C’, ’E’),(’D’,’B’), (’E’, ’C’), (’E’, ’D’) ])
```

IN BUILT DATASETS IN NEO4J

- Game of Throne- G.load_got()
- Twitter- G.load_twitter()
- European Roads- G.load_euroads()

```
nx2.pagerank(G)
nx2.betweenness_centrality(G)
nx2.closeness_centrality(G)
nx2.clustering(G)
nx2.list_connected_components(G)
```

SUMMARY

GRAPH DATABASES

- Neo4j & Examples

References

- Miller, J. J.
  Graph database applications and concepts with neo4j.
  In *Proceedings of the southern association for information systems conference, Atlanta, GA, USA* (2013), vol. 2324.

- Needham, M.
  `https://medium.com/neo4j/`
  `experimental-a-networkx-esque-api-for-neo4j-graph-algorithms-4002baac45`

- Prad Nelluru, Bharat Naik, E. L.
  Graph databases
  `www.cs.utexas.edu/users/dsb/cs386d/Projects14/GraphDB.pdf`.