

Graph Databases

-Santosh K Kasetty

Database:

- It is organized collection of data.
- Collection of schemas, queries, tables and other objects.
- It is formally referred as a set of related data and the way it is organized.
- Access to these data is generally provided by a database management system.

Types of Databases:

Databases are mainly classified based on the way they organize the data.

- Text databases
- Document databases
- Relational databases – collection of data items organized as tables. Data model used in this can be called as relational data model.

Graph database:

- Data here is organized as graph data model.
- A graph generally consists of nodes and relationships between the nodes.
- In a graph data model-
 - we have labelled nodes with properties
 - named relations with properties
 - we may use hypergraphs sometimes.

Types:

- We have many graph databases such as-

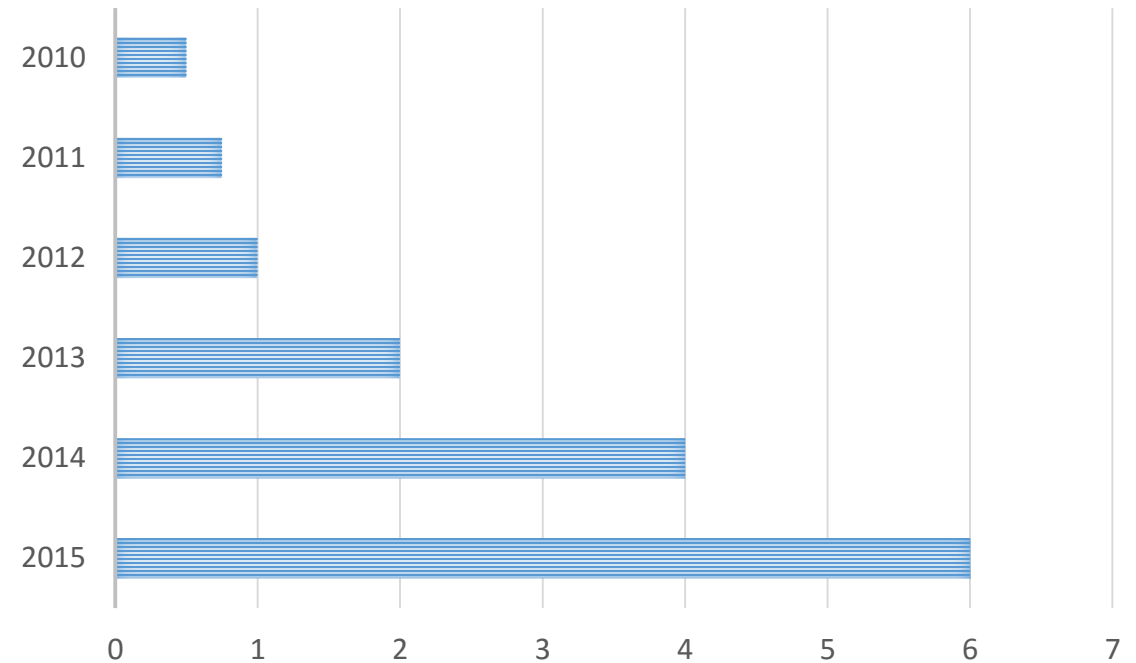
1. Neo4j
2. Graph DB
3. Vertex DB
4. Power graph
5. Hypergraph DB

Neo4j uses a simple but expressive language called cypher that is purpose-built for traversing data relationships.

Why not Relational database?

- Relational databases are most widely used now-a-days but they have some disadvantages too.

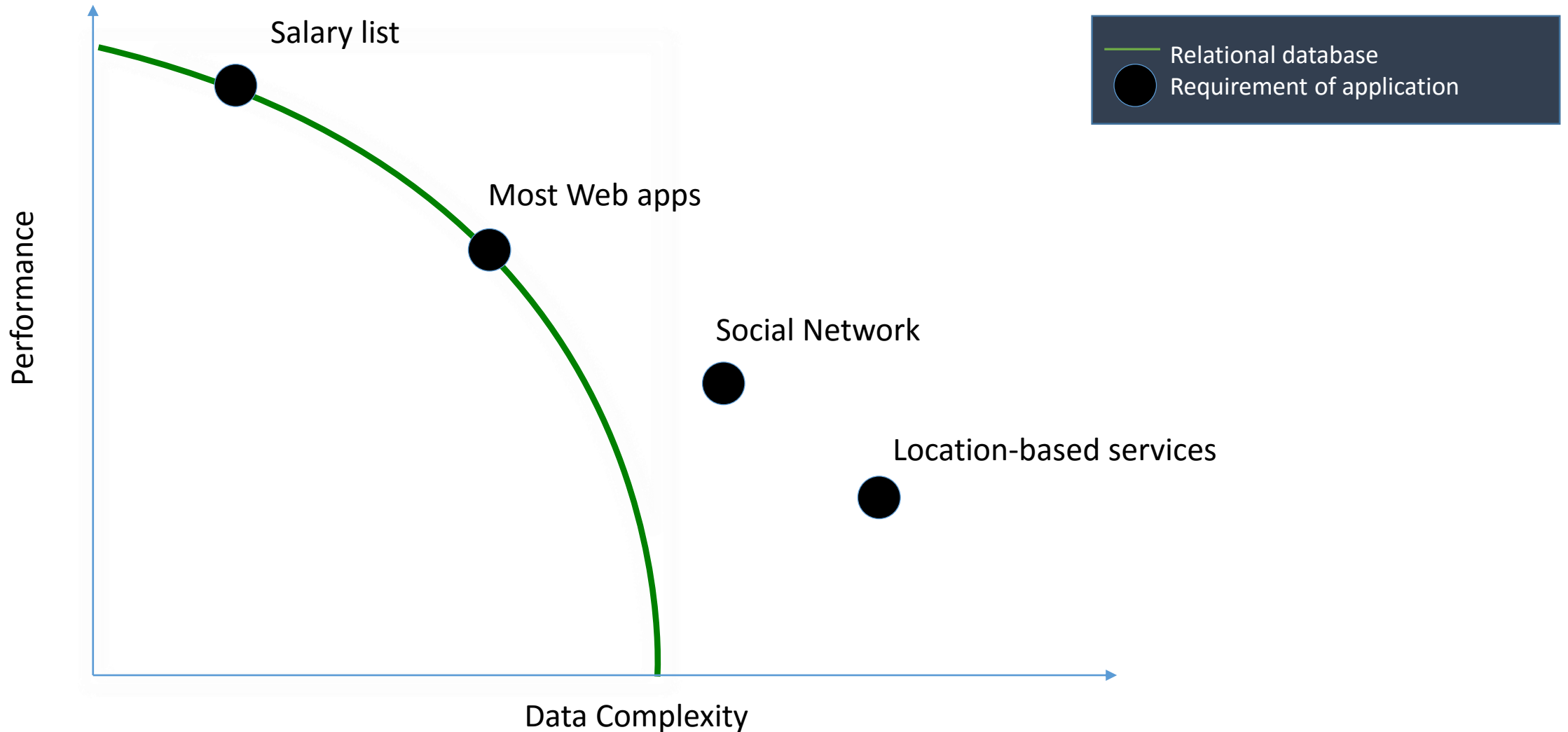
- Increasing data size



Highly connected data:

- Connectedness – with the increase in the connectivity of the data, the solution to a certain problem becomes complex with the use of relational databases.

Relational database performance



Speed matters...

- Even relational databases can solve the problems solved by using graph databases but speed matters
- With the increase in size of the data and join tables, query time increases using relational databases
- While in graph databases the size of the data do not affect the processing time of the query.
- This is because, in graph models, what we imagine for a particular problem is as similar as conceptual model.

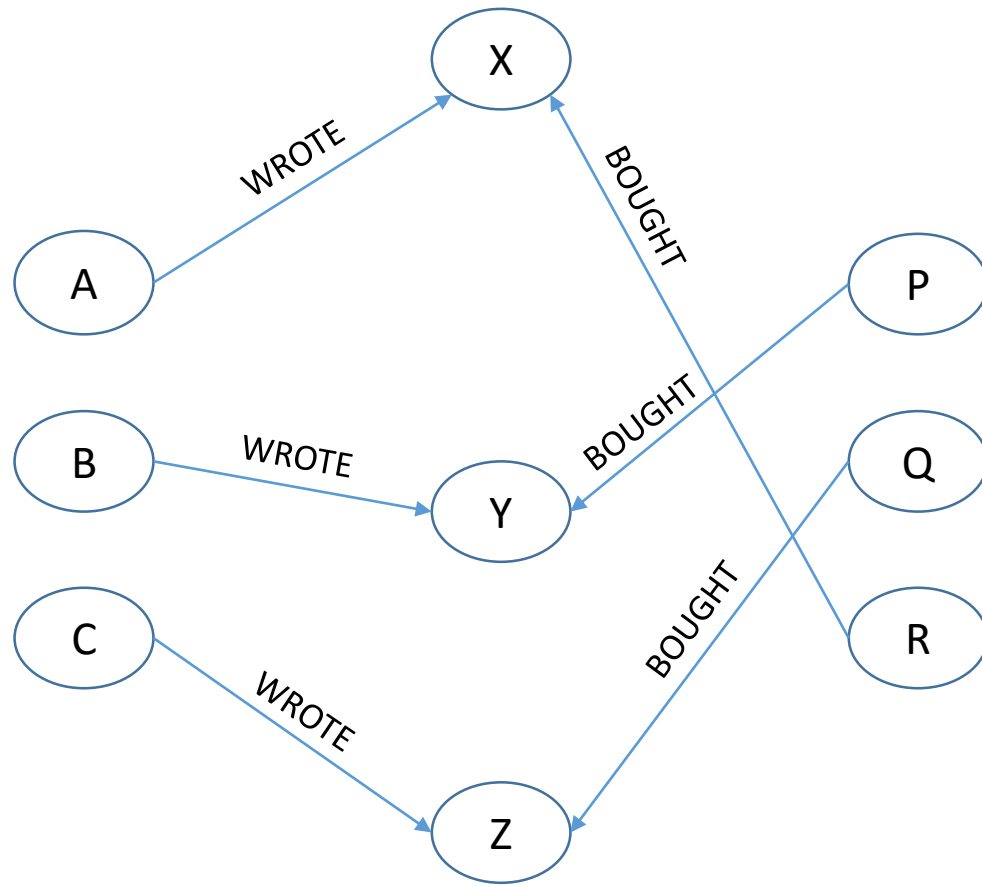
Converting relational model to a graph model:

- Tables become nodes.
- Join tables become relationships.
- Table name as a node label
- Columns of a table turn into properties.
- Foreign keys into relationships.

For Example:

Author	Book
A	X
B	Y
C	Z

Book	Person
Y	P
Z	Q
X	R



Graph database queries:

- Unlike relational SQL queries, graph database queries are straightforward to write and understand.
- cypher queries are much simpler than SQL queries. In fact, a long SQL query can frequently be compressed to many fewer lines in Cypher.
- By using Neo4j and Cypher, mismatch between the conceptual and physical models of their database is reduced.
- (sql) — [INTO] → (cypher)

Recommendation engine

- In a relational database, it requires multiple tables to model a simple recommendation engine.
- And also need of many join tables is required to get solution to a particular query.
- Where as, a graph model can easily be sketched.
- Now, lets just take an example of the product recommendations purchased by people who bought the same product..

Sql queries vs database queries:

- Database query-

Match (u: customer)-[bought]→(p: product) ←[bought]-(peer: customer)-[bought]→(reco: product)

Where NOT (u)-[bought]→(reco)

RETURN reco as Recommendation, count * as frequency

ORDER BY Frequency DESC LIMIT 5;

- Sql query-

- ?

Implementing Dijkstra's algorithm:

- Match (from: location{ location name: x}), (to: location{ location name: y}),
 path= shortestPath((from)—[connected to]→(to))
 RETURN path;
- Similarly we can find all the shortest paths from one location to another location using keyword allShortestPaths

Cont..

- If you want to order by the length (number of hops) of the paths in descending order :
- Match (from: location{ location name: x}), (to: location{ location name: y}),
paths= allShortestPaths((from)—[connected to]→(to))
RETURN paths;
ORDER BY length(paths) DESC

Update:

- A node in graph databases can have multiple properties.
- If a property value changes and if in case you need to update the index, its not enough to just update the new value but you need to remove the old value as well.
- SET clause is used to add a property
- REMOVE clause is used to remove a property of a particular node or relationship.

Delete a node with all its relationships:

- When you have a highly connected data and if in case you need to delete all the data connected to a particular entity or node-
- It becomes complex in relational databases
- But in case of graph databases
- DETACH DELETE is the keyword used
- For example:
- Match (n: "name")
- DETACH DELETE n;

THANK YOU