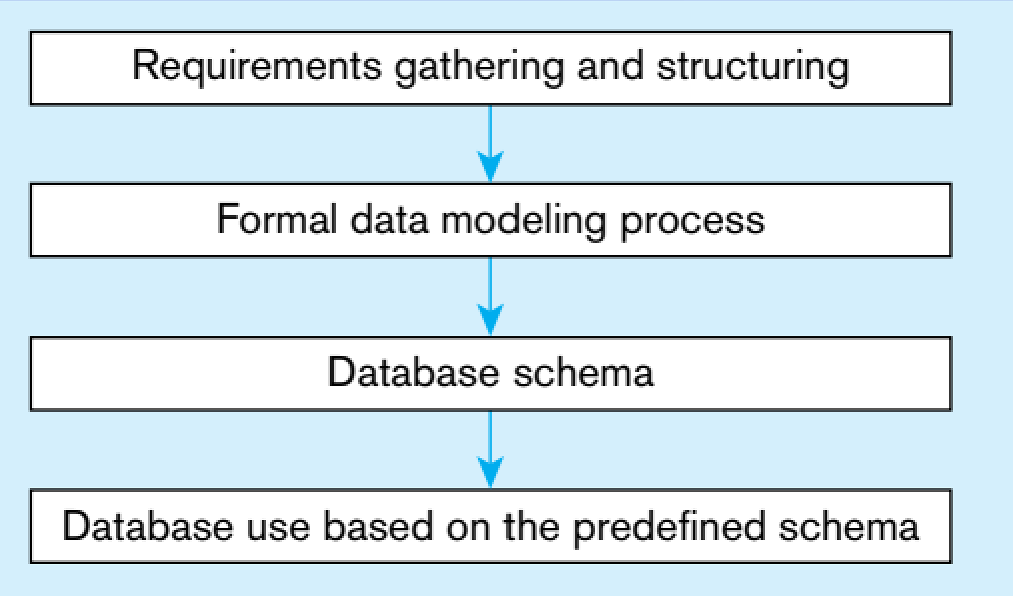
Big Data Ecosystem - Database

Dr. Deepak Saxena, SME IIT Jodhpur

Relational vs Big Data Approach

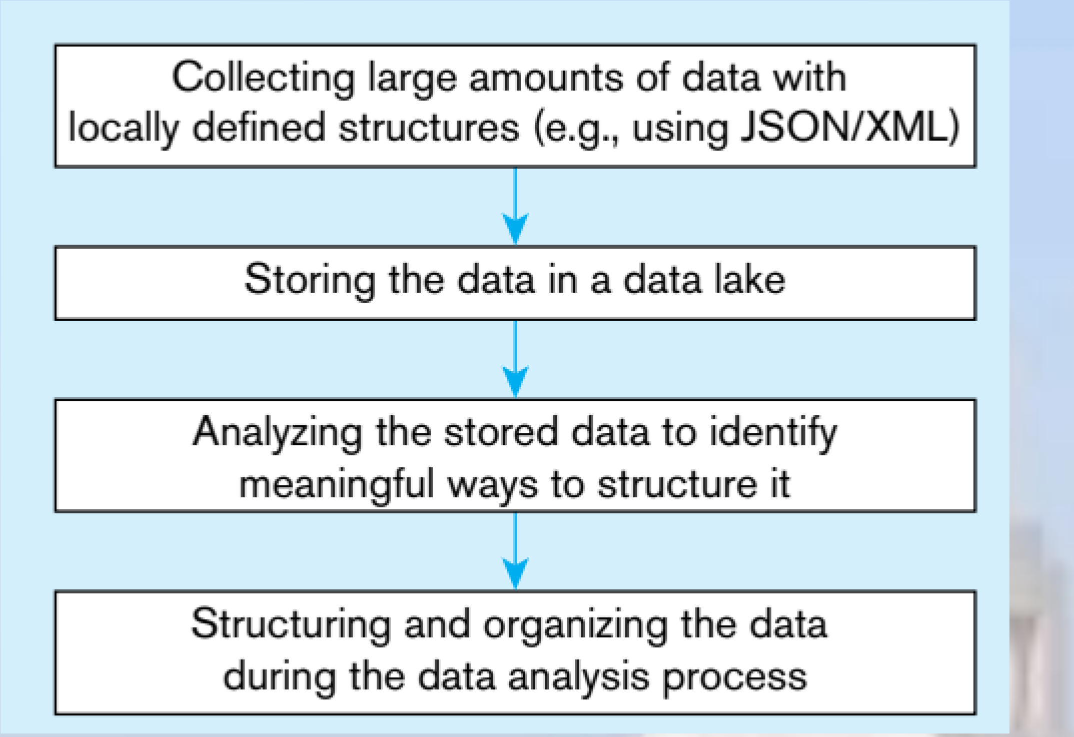


Requirements gathering and structuring

Formal data modeling process

Database schema

Database use based on the predefined schema



Collecting large amounts of data with locally defined structures (e.g., using JSON/XML)

Storing the data in a data lake

Analyzing the stored data to identify meaningful ways to structure it

Structuring and organizing the data during the data analysis process

Schema on Write
SQL (Structured Query Language)
Suitable for Data Warehouse

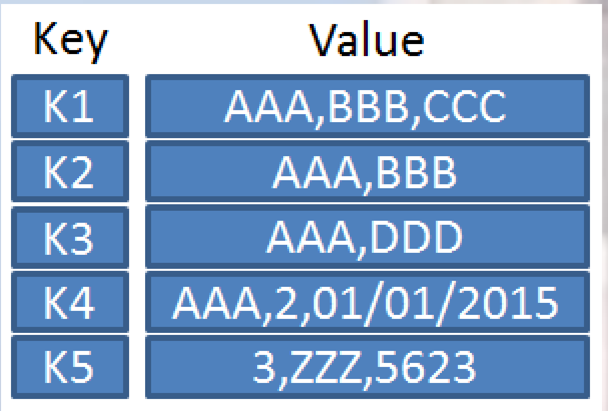
Schema on Read
NoSQL (Not Only SQL)
Suitable for Data Lake

Classification of NoSQL Database Systems

- Key-Value Stores
- Document Stores
- Wide-column Stores
- Graph-oriented databases

Key-Value Stores

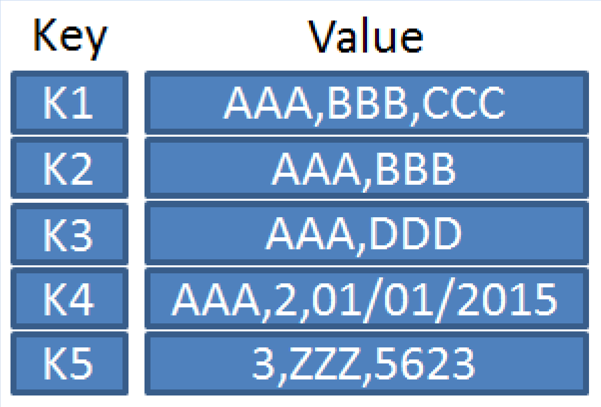
- A key-value store database maintains a structure that allows it to store and access “values” based on a “key”.
- The “key” is typically a string, with or without specific meaning.
- If some part of the “value” needs to be changed, the entire collection will need to be updated.
- Software solutions: REDIS, Amazon DynamoDB
- When to use key value database?
  - Handling Large Volume of Small and Continuous Reads and Writes
  - Storing Basic Information
  - Applications with Infrequent Updates and Simple Queries
  - Key-Value Databases for Volatile Data



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

Key-Value Stores

- Use Cases
  - Session management on a large scale.
  - Using cache (in-memory database) to accelerate application responses.
  - Storing personal data on specific users.
  - Product recommendations, storing personalized lists.
  - Managing each player’s session in massive multiplayer online games.



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

Key Value Databases

Advantages

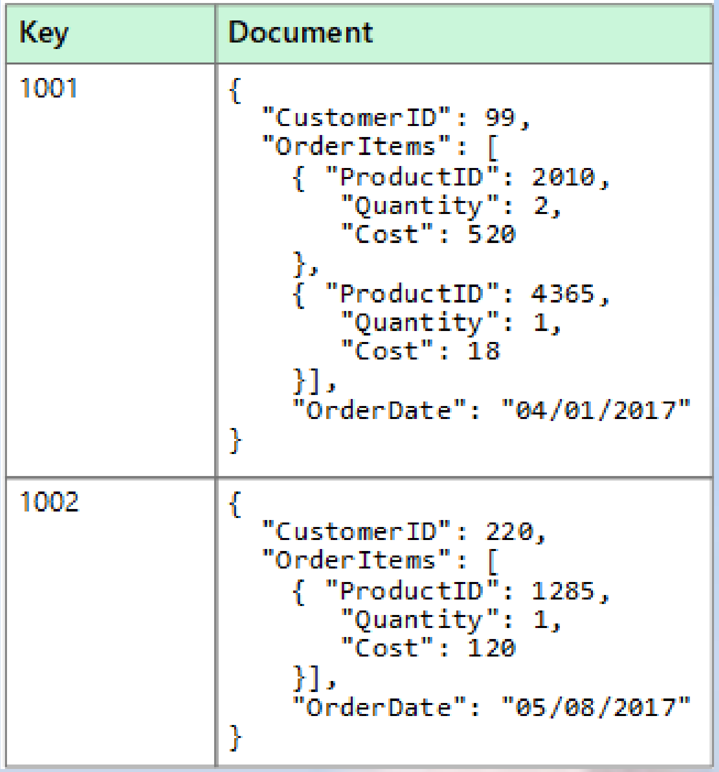
- Simplicity
- Speed
- Scalability
- Easy to move

Disadvantages

- Simplicity
- No query language
- Values can't be filtered

Document Stores

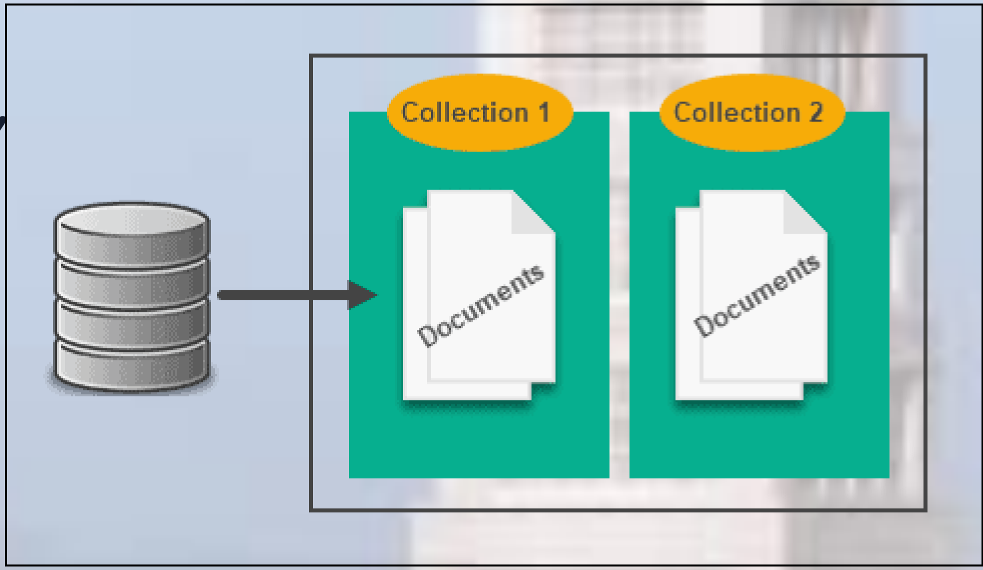
- A document in this context is a structured set of data formatted using a standard such as JSON, BSON, or XML.
- The key difference between key-value stores and document stores is that a document store has the capability of accessing and modifying the contents of a specific document based on its structure.
- The “documents” may have a hierarchical structure, and they *do not* typically reference each other.
- Software Solutions: MongoDB, Amazon DocumentDB



Key	Document

1001	{
	"CustomerID": 99,
	"OrderItems": [
	{ "ProductID": 2010,
	"Quantity": 2,
	"Cost": 520
	},
	{ "ProductID": 4365,
	"Quantity": 1,
	"Cost": 18
	}],
	"OrderDate": "04/01/2017"
	}

1002	{
	"CustomerID": 220,
	"OrderItems": [
	{ "ProductID": 1285,
	"Quantity": 1,
	"Cost": 120
	}],
	"OrderDate": "05/08/2017"
	}



Document Stores: Use Cases

- Customer data management and personalization
- Internet of Things (IoT) and time-series data
- Product catalogs and content management
- Payment processing
- Mobile apps
- Operational analytics
- Real-time analytics

Document Stores

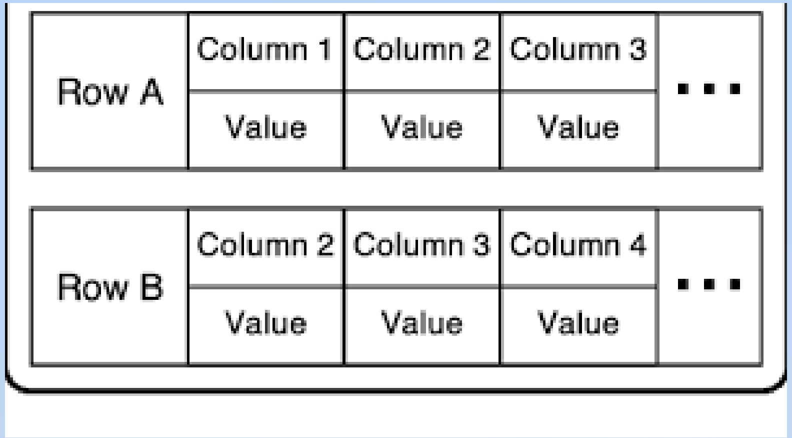
Advantages

- Schema-less
- Faster creation and maintenance
- No foreign keys
- Open formats
- Built-in versioning

Disadvantages

- Consistency-Check Limitations
- Security

Wide-column Stores

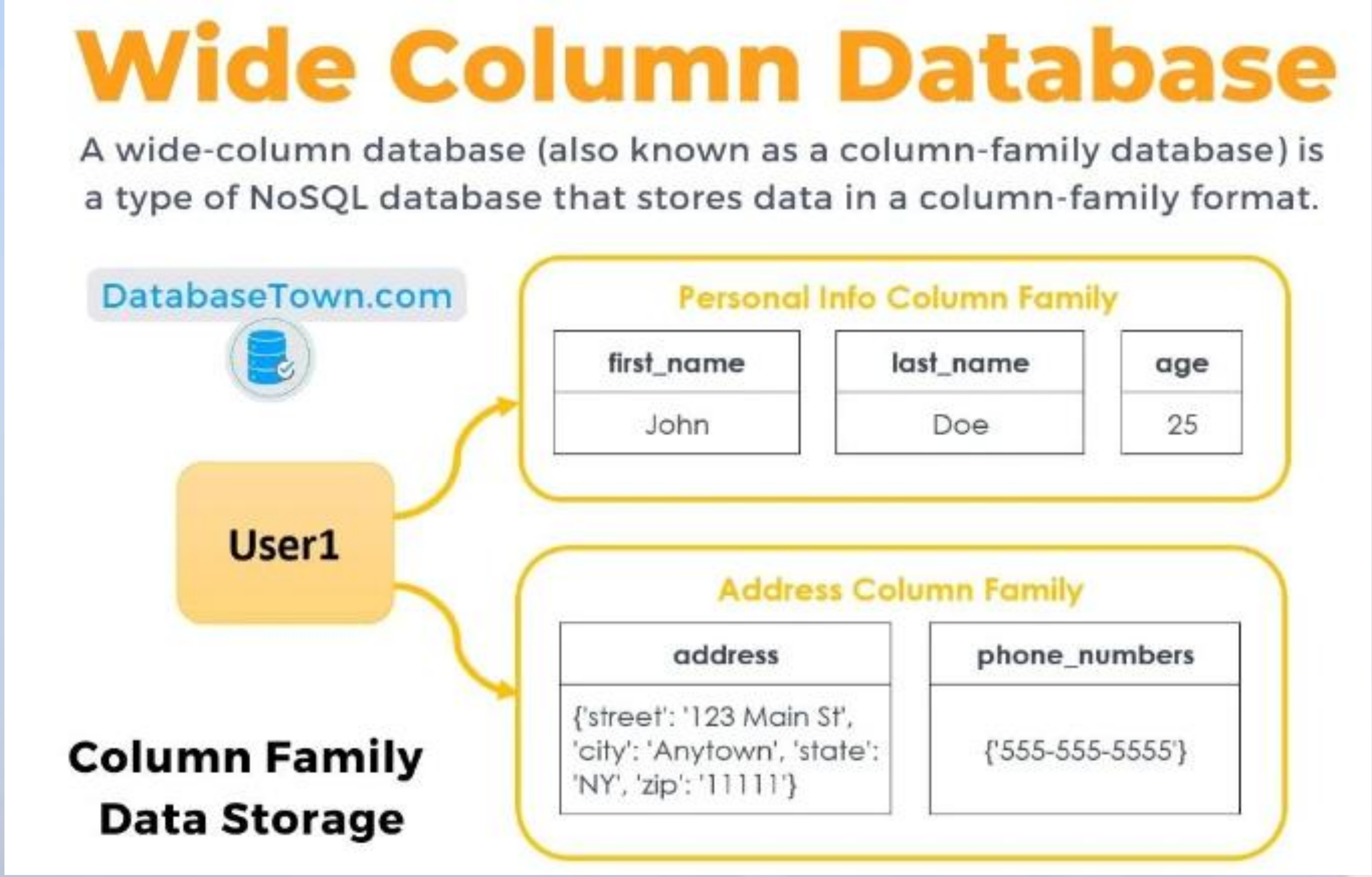


Row A	Column 1	Column 2	Column 3	...
	Value	Value	Value	

Row B	Column 2	Column 3	Column 4	...
	Value	Value	Value	

- Consist of rows and columns, and their characteristic feature is the distribution of data based on both key values (records) and columns, using “column groups” or “column families” to indicate which columns are best to be stored together.
- They allow each row to have a different column structure (there are no constraints defined by shared schema), and the length of the rows varies.
- A column is only written if there is a data element for it.
- Software solutions: Apache Cassandra, BigTable, ScyllaDB

Wide-column Stores



Wide Column Database

A wide-column database (also known as a column-family database) is a type of NoSQL database that stores data in a column-family format.

DatabaseTown.com

User1

Personal Info Column Family

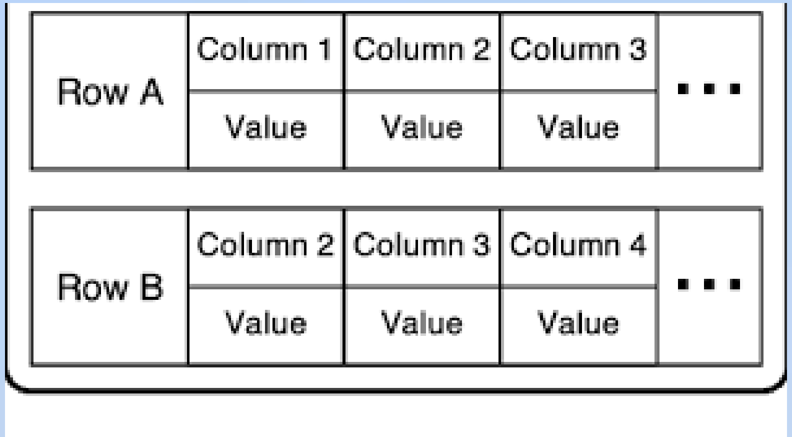
first_name	last_name	age
John	Doe	25

Address Column Family

address	phone_numbers
{'street': '123 Main St', 'city': 'Anytown', 'state': 'NY', 'zip': '11111'}	{'555-555-5555'}

Column Family Data Storage

Wide-column Stores: Use Cases



Row A	Column 1	Column 2	Column 3	. . .
	Value	Value	Value	

Row B	Column 2	Column 3	Column 4	. . .
	Value	Value	Value	

- Log data
- IoT (Internet of Things) sensor data
- Time-series data, such as temperature monitoring or financial trading data
- Attribute-based data, such as user preferences or equipment features
- Real-time analytics
- High throughput data such as gaming or e-commerce

Wide-column Stores

Advantages

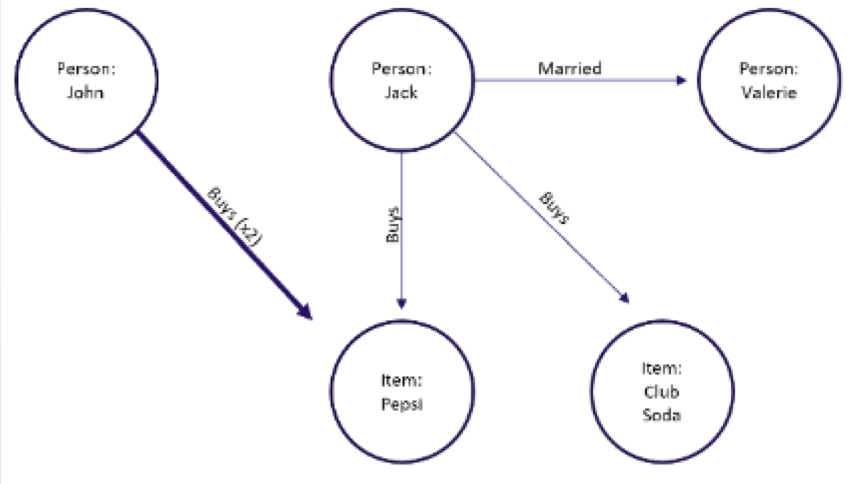
- High performance
- Flexible and efficient data model
- Scalability
- Distributed Systems
- Handling high write throughput

Disadvantages

- Limited querying capabilities
- Limited data modelling
- Limited support for advanced features
- Data Migration

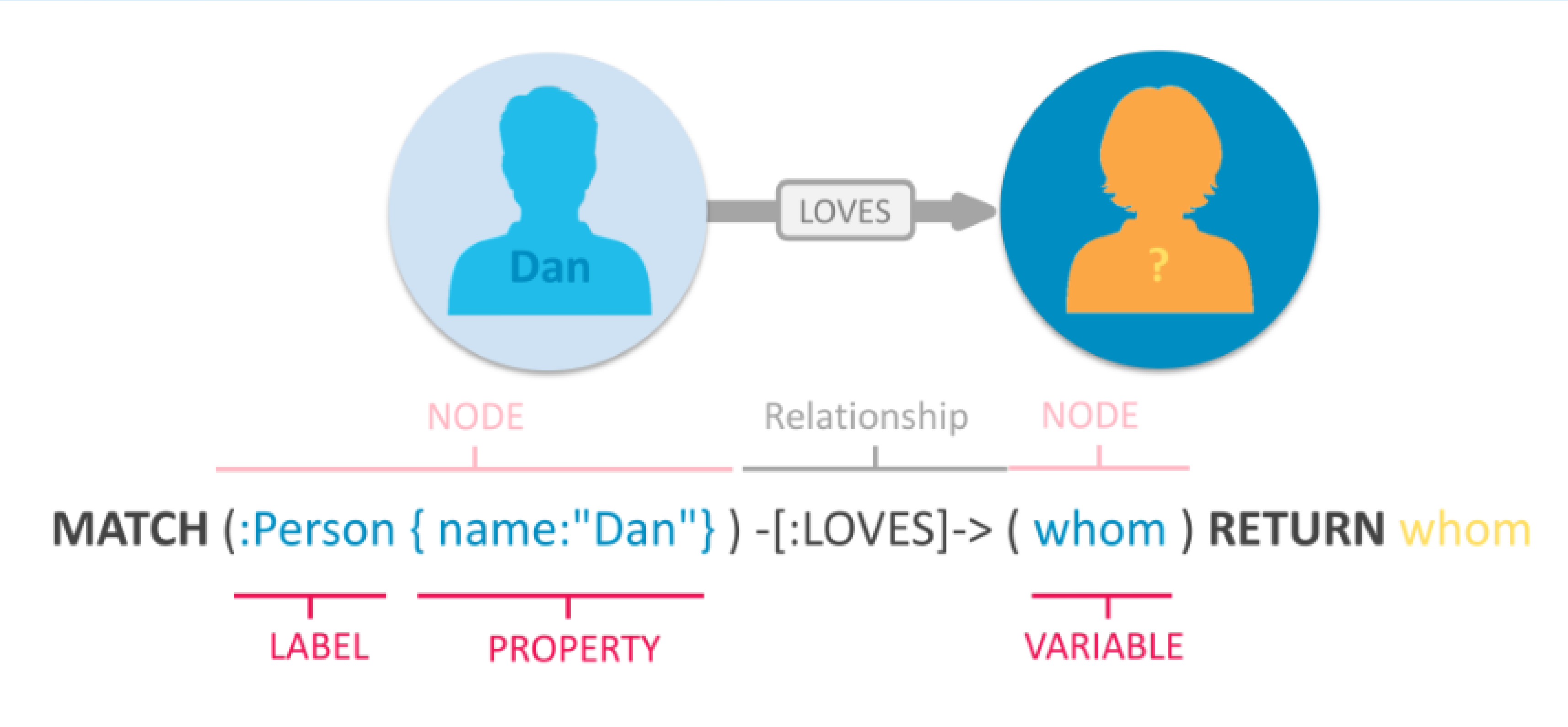
Graph-oriented databases

- Data in a graph-oriented database is stored in nodes with properties (named attribute values), and the connections between the nodes represent relationships between the real-world instances.
- The collections of attributes associated with each node may vary.
- Relationships may also have attributes associated with them.
- Software solutions: Neo4J, Oracle's Graph Database



The property graph model in Neo4J

In Neo4j, information is organized as nodes, relationships, and properties.



```
MATCH (:Person { name:"Dan"} ) -[:LOVES]-> ( whom ) RETURN whom
```

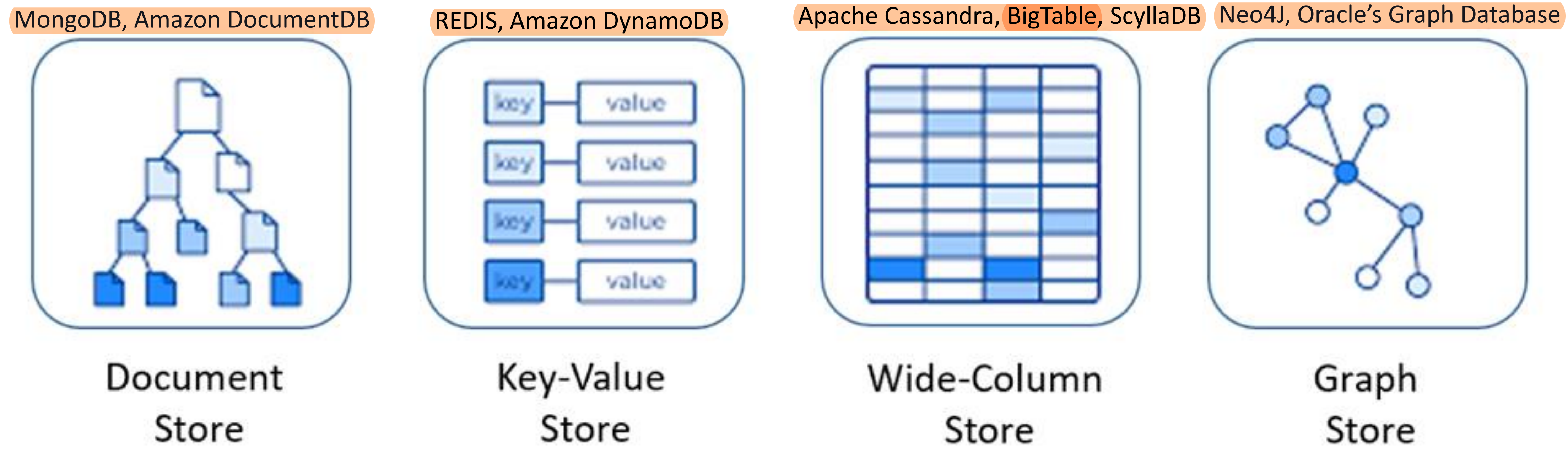
The property graph model in Neo4J

- ***Nodes*** are the entities in the graph.
  - Nodes can be tagged with labels, representing their different roles in your domain. (For example, Person).
  - Nodes can hold any number of key-value pairs, or properties. (For example, name)
  - Node labels may also attach metadata (such as index or constraint information) to certain nodes
- ***Relationships*** provide directed, named, connections between two node entities (e.g., Person LOVES Person).
  - Relationships always have a direction, a type, a start node, and an end node, and they can have properties, just like nodes.
  - Nodes can have any number or type of relationships without sacrificing performance.
  - Although relationships are always directed, they can be navigated efficiently in any direction.

Graph-oriented databases

- Graph databases allow:
  - Navigate deep hierarchies,
  - Find hidden connections between distant items, and
  - Discover inter-relationships between items.
- Use Cases
  - Fraud detection
  - Detection of money laundering
  - Bill of materials
  - Cybersecurity
  - Contact tracing
  - Product recommendations
  - AI (Feature Engineering, Neural Networks)

To summarize



MongoDB, Amazon DocumentDB

REDIS, Amazon DynamoDB

Apache Cassandra, BigTable, ScyllaDB

Neo4J, Oracle's Graph Database

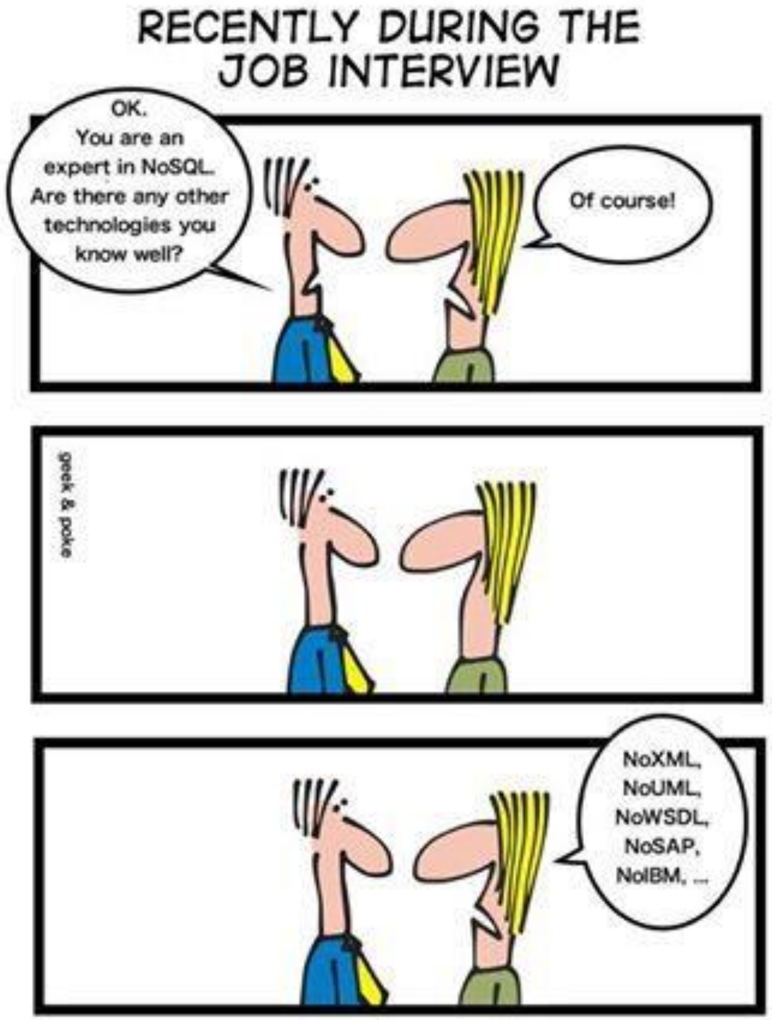
Document Store

Key-Value Store

Wide-Column Store

Graph Store

What not to claim in your job interview…


RECENTLY DURING THE JOB INTERVIEW
OK. You are an expert in NoSQL. Are there any other technologies you know well?
Of course!
geek & poke
NoXML, NoUML, NoWSDL, NoSAP, NoIBM, ...