7BDIN006W
Big Data Theory and Practice

Lecture 9

Big Data and NoSQL Databases.

Introduction to Neo4j

UNIVERSITY OF WESTMINSTER

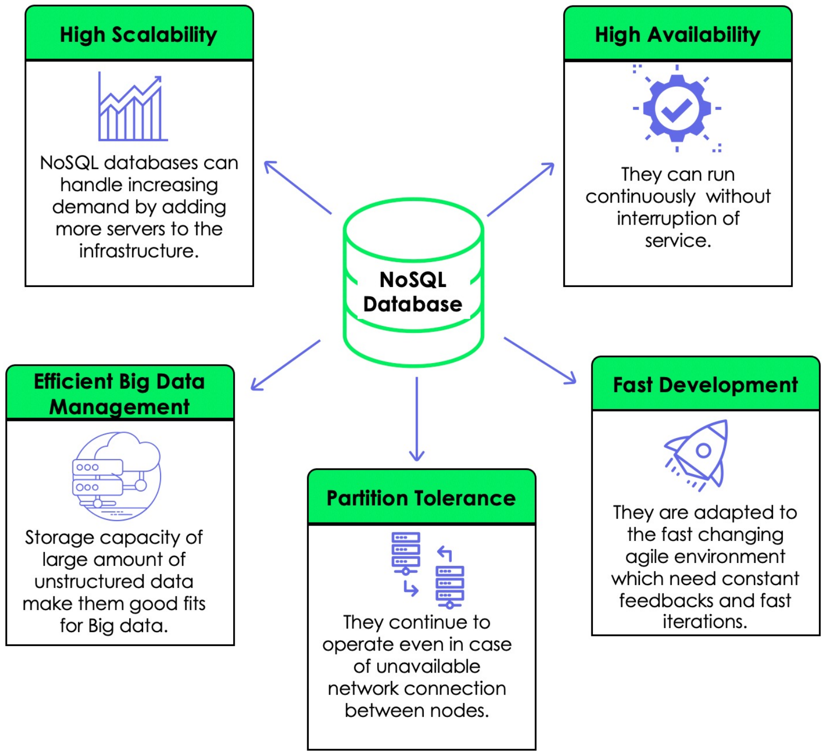
No-SQL Databases in Big Data

AEROSPIKE
cassandra
Clustrix
Couchbase
amazon DynamoDB
APACHE HBASE
IBM Cloudant
MarkLogic
memsql Speed. Scale. Simplicity.
mongoDB
NUODB
ORACLE NOSQL DATABASE
riak
splice MACHINE
TRANSLATTICE
VOLTDB

Background: Traditional relational databases (RDBMS) struggle with the scale, variety, and complexity of Big Data. This limitation led to the emergence of No-SQL databases, designed to handle large volumes of unstructured and semi-structured data efficiently.

Definition: No-SQL ("Not Only SQL") databases are non-tabular databases and store data differently than relational tables. They provide a mechanism for storing and retrieving data modelled in means other than the tabular relations used in relational databases.

Key Features of NoSQL Databases



NoSQL Database

High Scalability

NoSQL databases can handle increasing demand by adding more servers to the infrastructure.

High Availability

They can run continuously without interruption of service.

Efficient Big Data Management

Storage capacity of large amount of unstructured data make them good fits for Big data.

Partition Tolerance

They continue to operate even in case of unavailable network connection between nodes.

Fast Development

They are adapted to the fast changing agile environment which need constant feedbacks and fast iterations.

Schema Flexibility

Dynamic, schema-less data models allow for varied and evolving data structures.

Scalability

Efficient horizontal scaling by distributing data across multiple servers.

Diverse Data Models

Supports multiple types, including key-value, document, column-family, and graph databases.

Performance

Optimized for high-speed operations and specific data access patterns.

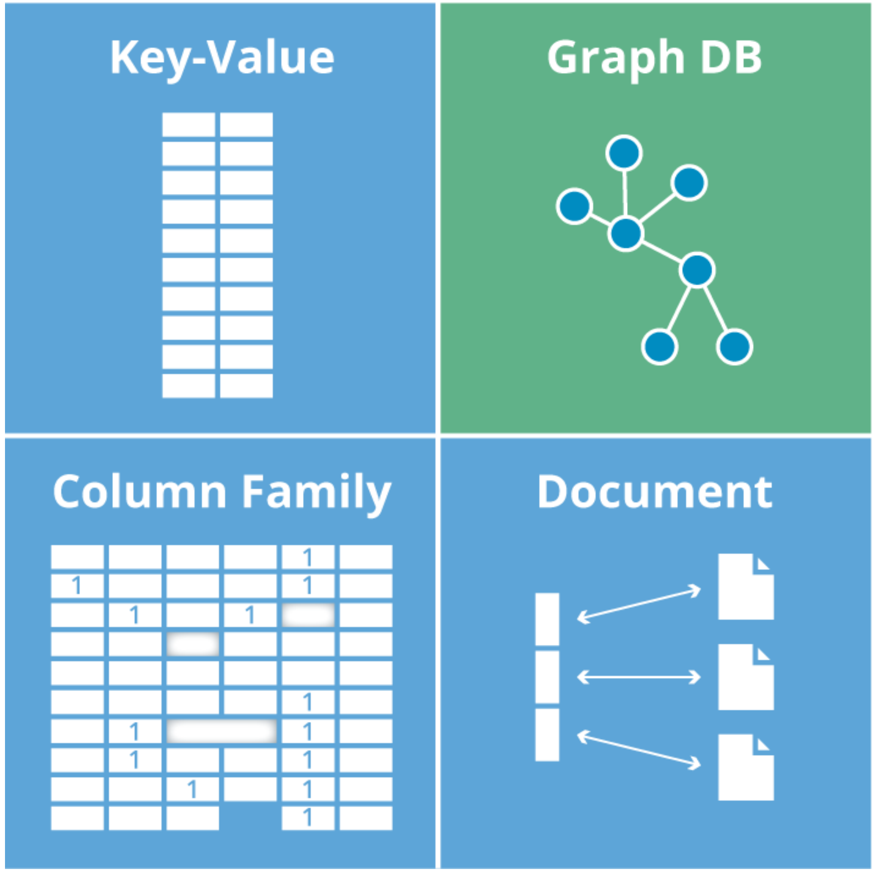
High Availability and Fault Tolerance

Built-in data replication and sharding for resilience and load balancing.

Big Data and Real-Time Analytics

Ideal for large-scale data handling and fast data processing.

4 Types of No-SQL Databases



Key-Value Stores

Example: Redis, DynamoDB.

Characteristics: Simplest No-SQL databases, storing data as a collection of key-value pairs. Ideal for scenarios requiring high-speed read/write access to data.

Use Cases: Session stores, user profiles, and configuration settings.

Document Databases

Example: MongoDB, CouchDB.

Characteristics: Store data in documents (typically JSON, BSON, XML). More flexible than key-value stores, allowing nested structures.

Use Cases: Content management systems, e-commerce applications, and real-time analytics.

Column-Family Stores

Example: Cassandra, HBase.

Characteristics: Store data in columns instead of rows, suitable for querying large datasets. Offer high scalability and performance.

Use Cases: Time-series data, event logging, and IoT applications.

Graph Databases

Example: Neo4j, Amazon Neptune.

Characteristics: Represent data in graph structures with nodes, edges, and properties. Optimized for handling complex relationships.

Use Cases: Social networks, fraud detection, and recommendation engines.

Neo4j

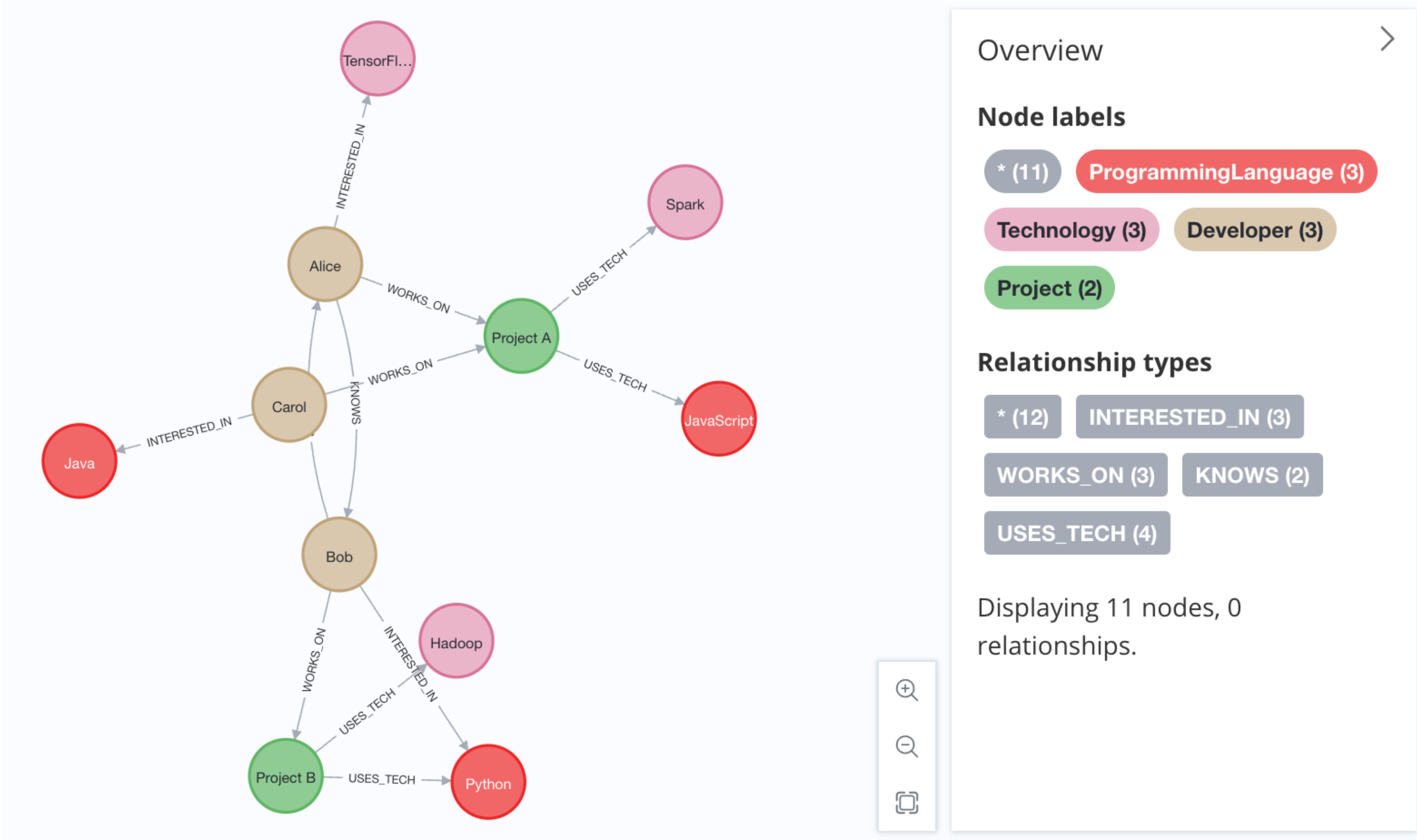
Neo4j is an open-source No-SQL graph database implemented in Java. It's designed to handle data relationships efficiently and is known for its high performance in traversing complex networks.

Unlike relational databases, Neo4j stores data in nodes and relationships, which directly map to real-world entities and their interconnections. This structure is inherently more suited for associative data sets.

Key Features

- **Cypher Query Language**: Neo4j uses its query language, Cypher, designed for easy querying of graph data. Cypher syntax is intuitive and expressive, making complex queries more readable and straightforward to write.
- **ACID Transactions**: Ensures full ACID (Atomicity, Consistency, Isolation, Durability) compliance, guaranteeing reliable transaction processing.
- **High Availability and Scalability**: Supports clustering for high availability and offers capabilities for horizontal scalability.

Example: Simple Developers Graph



Overview

Node labels

* (11) ProgrammingLanguage (3) Technology (3) Developer (3) Project (2)

Relationship types

* (12) INTERESTED_IN (3) WORKS_ON (3) KNOWS (2) USES_TECH (4)

Displaying 11 nodes, 0 relationships.

Neo4j Demonstration

Let’s consider
a simplified Developers Database…