Neo4j is one of the popular Graph Databases and Cypher Query Language (CQL). Neo4j is written in Java Language.

Neo4j is a NoSQL database. It is highly scalable and schema-free. It's world most popular graph database management system. Neo4j was developed by Neo technology and called an ACID (Atomicity, consistency, isolation & durability)-compliant transactional database with native graph storage and processing.

The graph model doesn't usually require a predefined schema. So there is no need to create the database structure before you load the data (like you do in a relational database). In Neo4j, the data is the structure. Neo4j is a "schema-optional" DBMS.

In Neo4j, no need to set up primary key/foreign key constraints to predetermine which fields can have a relationship, and to which data. You just have to define the relationships between the nodes you need.

RDBMS Vs Graph Database

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **RDBMS** | **Graph Database** |
| 1 | Tables | Graphs |
| 2 | Rows | Nodes |
| 3 | Columns and Data | Properties and its values |
| 4 | Constraints | Relationships |
| 5 | Joins | Traversal |

# Features of Neo4j:

**Flexible Schema:**Neo4j follows a data model called graph model. The graph contains nodes and the nodes are connected to each other. Nodes and relationships store data in key-value pairs known as properties.

**ACID Property:**Neo4j supports full ACID properties (Atomicity, Consistency, Isolation and Durability).

**Scalability:**Neo4j facilitates you to scale the database by increasing the number of reads/writes, and the volume without affecting the data integrity and the speed of query processing.

**Reliability:**Neo4j provides replication for data safety and reliability.

**Cypher Query Language:**Neo4j provides a powerful declarative query language called Cypher Query language. It is used to create and retrieve relations between data without using the complex queries like Joins.

**Built-in Web applications:**Neo4j also provides a built-in Neo4j browser web application which can be used to create and retrieve your graph data.

**GraphDB:**Neo4j follows Property Graph Data Model.

## General Features:

* It supports UNIQUE constraints.
* It uses Native graph storage with Native GPE(Graph Processing Engine).
* It supports exporting of query data to JSON and XLS format.
* **It provides REST API to be accessed by any Programming Language like Java, Spring, Scala etc.**
* It provides Java Script to be accessed by any UI MVC Framework like Node JS.
* **It supports two kinds of Java API: Cypher API and Native Java API to develop Java applications.**

# Advantages of Neo4j:

**Highly scalable:** Neo4j is highly scalable. It provides a simple, powerful and flexible data model which can be changed according to applications and uses. It provides:

* Higher vertical scaling.
* Improved operational characteristics at scale.
* Higher concurrency.
* Simplified tuning.

**Schema-free:** Neo4j is schema-free like other NoSQL databases.

**High availability:** Neo4j provides high availability for large enterprise real-time applications with transactional guarantees.

**Real-time data analysis:** Neo4j provides results based on real-time data.

**Easy representation:** Neo4j provides a very easy way to represent connected and semi-structured data.

**Fast Execution:** Neo4j is fast because more connected data is very easy to retrieve and navigate.

**Easy retrieval:** Neo4j facilitates you not only represent but also easily retrieve **(traverse/navigate)** connected data faster other databases comparatively.

**Cypher Query language:** Neo4j provides CQL **(Cypher Query Language)** a declarative query language to represent the graph visually, using ASCII-art syntax. The commands of this language is very easy to learn and human readable.

**No Join:** Neo4j doesn't require complex Joins to retrieve connected/related data as it is very easy to retrieve its adjacent node or relationship details without Joins or Indexes because it is a graph database and all nodes are already connected.

## What is Graph database

A graph database is a database which is used to model the data in the form of graph. It store any kind of data using:

* Nodes
* Relationships
* Properties

**Nodes:** Nodes are the records/data in graph databases. Data is stored as properties and properties are simple name/value pairs.

Nodes can be grouped together by applying a Label to each member. A node can have zero or more labels. Labels do not have any properties. Storing data in Neo4j is similar to add more records in other databases.

**Relationships:** It is used to connect nodes. It specifies how the nodes are related.

* Relationships always have direction.
* Relationships always have a type.
* Relationships form patterns of data.

**Properties:** Properties are named data values.

**Spring Boot Data Neo4j:**

**Product.java**

@Node  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class Product {  
 @Id  
 private Integer id;  
 private String name;  
 private int qty;  
 private double price;  
}

**application.properties**

spring.neo4j.uri=bolt://localhost:7687  
spring.neo4j.authentication.username=neo4j  
spring.neo4j.authentication.password=Praneetha#1  
server.port=9092

**Customer.java**

@Node  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
@Builder  
public class Customer {  
 @Id  
 private Integer custId;  
 private String name;  
 private String[] address;  
 @Relationship(type = "Purchase",direction = Relationship.Direction.*OUTGOING*)  
 private List<Product> products;  
}