Department of CSE & CSE(AI&ML)
Lab - 9
NEO4J

What is Neo4j?

Neo4j is a graph database management system designed to store, query, and manipulate data in the form of nodes, relationships, and properties. Unlike traditional relational databases that use tables, Neo4j uses a graph model, which is ideal for handling highly connected data.

Graph Database Basics

Graph Component	Description	Example
Node	Represents an entity	Person, Movie, Product
Relationship	Connects nodes, has a type and direction	(:Person)-[:LIKES]->(:Movie)
Property	Key-value pairs on nodes or relationships	name: "Alice", age: 30

Core Data Model in Neo4j

Component	Purpose	Example
Node	Entity/object in the system	(:Person {name: 'Alice'})
Relationship	A connection between two nodes, always directed	(:Person)-[:FRIEND_WITH]->(:Person)
Property	Key-value pair stored in a node or relationship	age: 25, since: 2015
Label	Defines the type/class of a node	:Person, :Movie, :Product
Relationship Type	Describes the relationship's meaning	FRIEND_WITH, LIKES, WORKS_AT

Common Symbols in Cypher

Symbol	Meaning
( )	Node
[ ]	Relationship
:	Label (for nodes) or type (for relationships)
{}	Properties
-> or <-	Direction of relationship

In Neo4j, a node represents an entity or object, like a person, product, city, or movie.

- It can have labels (types or categories)
- It can have properties (key-value data)

CREATE (nodeAlias:Label {property1: value1, property2: value2, ...})

CREATE (p:Person)

Creates a node with label Person, but no data yet.

Create a node with properties:

CREATE (p:Person {name: "Alice", age: 30, city: "Bangalore"})

Creates a Person node with three properties.

Create multiple nodes at once:

```
CREATE
  (a:Person {name: "Alice"}),
  (b:Person {name: "Bob"}),
  (c:Movie {title: "Inception", year: 2010})
```

Create node with multiple labels:

CREATE (p:Person:Employee {name: "Charlie", dept: "HR"})
Now the node is both Person and Employee.

Best Practices

- Use CamelCase or PascalCase for labels like :Person, :Movie
- Use snake_case or lowercase for property keys like name, birth_year
- Enclose string values in double quotes " " or single quotes ' '

PROBLEM STATEMENT

You are building a University Social Graph to track students, professors, courses, and friendships. The goal is to:

- Model academic and social relationships
- Perform Cypher queries for analysis
- Visualize how entities interact in a university setting

Part A: Create Nodes

1. Create Student Nodes with name, age and major. We have Alice, 21, CSE & Bob, 22, ECE and Charlie, 20, CSE.

2. Create Professor Nodes, Dr. Smith for the CSE department and Dr. Jones for the ECE department

3. Create Course Nodes (CODE: CS101, DATA STRUCTURES & CODE: EC202, DIGITAL SYSTEMS)

Part B: Create Relationships

4. Show that Alice has enrolled into “CS101” and Bob has enrolled into “EC202”

5. Show that professor Dr. Smith teaches “CS101” and Dr. Jones teaches “EC202”

6. Create friendship between Alice and Charlie

Part C: Query the Graph

7. List All Students

8. Find Courses Taught by Dr. Smith

9. Find Friends of Charlie

10. List All Students in the Same Course

11. Find Professors Who Teach Alice’s Courses

12. Find Students Who Are Friends and Enrolled in the Same Course

13. Find Courses with More Than One Student Enrolled

Bonus Query (Advanced)

14. Count how many students each professor teaches.

Assignment Deliverables:

- Submit screenshots of Cypher queries and results
