USN : 1RVU23CSE394
NAME : SAHANAA M S

Ex No: 7 Date: 15/10/2025	Exploring and Querying a Graph Database Using Neo4j

Objective:

To gain hands-on experience with Neo4j, a NoSQL graph database, by creating, updating, deleting, and analyzing data using nodes, relationships, and properties. This lab demonstrates how graph databases model complex relationships naturally and how Cypher queries can be used for data exploration, recommendation, and analysis.

Outcomes:

- Understand the concept of nodes, relationships, and properties in graph databases.
- Design and implement a graph data model for an e-commerce platform.
- Execute CREATE, READ, UPDATE, and DELETE (CRUD) operations using Cypher.
- Perform analytical queries to derive insights such as product recommendations, user behavior, and category-wise revenue.
- Visualize graph connections and understand the benefits of Neo4j for highly connected data.

Materials:

Software: Neo4j Desktop
Dataset / Case: ShopSmart E-commerce Platform
Entities: Users, Products, Categories, Brands, Reviews
Relationships: BOUGHT, VIEWED, RATED, BELONGS_TO, MADE_BY, SIMILAR_TO, FRIENDS_WITH, REVIEWS

Lab Procedure:

Stage 1: Graph Database Setup

- Launch Neo4j Desktop and create a new project named ShopSmartGraph.
- Start a new database and open the Neo4j Browser.
- Use the following commands to view schema details:
  SHOW DATABASES;
  CALL db.labels();
  CALL db.relationshipTypes();
  CALL db.propertyKeys();
  CALL db.schema.visualization();

USN : 1RVU23CSE394
NAME : SAHANAA M S

Stage 2: Creating the Graph (CREATE Operations)

- Use Cypher queries to create nodes and relationships such as BOUGHT, VIEWED, BELONGS_TO, MADE_BY, RATED, FRIENDS_WITH, and SIMILAR_TO.

Stage 3: Querying the Graph (READ Operations)

- Perform read queries to analyze user behavior and product data using MATCH and RETURN statements.

Stage 4: Updating the Graph (UPDATE Operations)

- Modify existing data using SET and MERGE operations.

Stage 5: Deleting Data (DELETE Operations)

- Remove nodes and relationships using DELETE or DETACH DELETE.

Stage 6: Analytical / Complex Queries

- Use Cypher aggregations and filters to perform deeper analysis, such as top users, product ratings, and revenue by category.

Stage 7: Visualization

- Visualize the full graph using MATCH (n)-[r]->(m) RETURN n, r, m;

Results:

Successfully created and visualized an e-commerce graph model in Neo4j.
Executed CRUD and analytical queries using Cypher.
Derived insights on top users, product performance, and category-wise revenue.
Observed how graph databases simplify traversal-based queries compared to SQL.

Conclusion:

This experiment demonstrated how Neo4j graph databases provide a natural and efficient way to store and query highly connected data such as e-commerce relationships. By performing create, read, update, delete, and analytical operations, we learned how to model real-world systems using nodes and relationships and how to extract meaningful insights using Cypher.

GitHub Link: