Introduction to Graph Databases

Table of Contents

About this module

The Neo4j Graph Platform enables developers to create applications that are best architected as graph-powered systems that are built upon the rich connectedness of data.

At the end of this module, you be able to:

- Describe what a graph database is.
- Describe some common use cases for using a graph database.
- Model a real-world scenario as a graph.

The evolution of graph databases

Today's business and user requirements demand applications that connect more and more of the world's data, yet still expect high levels of performance and data reliability. Many applications of the future will be built using graph databases like Neo4j.

In this video, you will learn how the need for graph databases has evolved.

What Is a graph database?

A graph database is an online database management system with Create, Read, Update and Delete (CRUD) operations working on a graph data model. Graph databases are generally built for use with online transaction processing (OLTP) systems. Accordingly, they are normally optimized for transactional performance, and engineered with transactional integrity and operational availability in mind.

Unlike other databases, relationships take first priority in graph databases. This means your application doesn't have to infer data connections using foreign keys or out-of-band processing, such as MapReduce.

By assembling the simple abstractions of nodes and relationships into connected structures, graph databases enable us to build sophisticated models that map closely to our problem domain.

The case for graph databases

The biggest value that graphs bring to the development stack is their ability to store relationships and connections as first-class entities.

For instance, the early adopters of graph technology reimagined their businesses around the value of data relationships. These companies have now become industry leaders: LinkedIn, Google, Facebook and PayPal.

As pioneers in graph technology, each of these enterprises had to build their own graph database from scratch. Fortunately for today's developers, that's no longer the case, as graph database technology is now available off the shelf.

What is a graph?

A graph is composed of two elements: **nodes** and **relationships**.

Each node represents an entity (a person, place, thing, category or other piece of data), and each relationship represents how two nodes are connected. For example, the two nodes *Person* and *Location*, might have the relationship *LIVES_AT* pointing from a *Person* node to *Location* nod

This general-purpose structure allows you to model all kinds of scenarios: from a system of roads, to a network of devices, to a population's medical history, or anything else defined by relationships.

With Neo4j, nodes can have **labels** that are used to define types for nodes. For example, a *Location* node is a node with the label *Location*. That same node can also have a label, *Residence*. Another *Location* node can also have a label, *Business*. A label can be used to group nodes of the same type. For example, you may want to retrieve all of the *Business* nodes.

The Neo4j database is a property graph. You can add **properties** to nodes and relationships to further enrich the graph model. This enables you to closely allign data and connections in the graph to your real-world application. For example, a *Person* node might have a property, *name* and a *Location* node might have a property, *address*.

How does Neo4j support the property graph model?

- Neo4j is a **Database** - use it to reliably **store information** and **find it later**.
- Neo4j's data model is a **Graph**, in particular a **Property Graph**.
- **Cypher** is Neo4j's graph query language (**SQL for graphs!**).
- Cypher is a declarative query language: it describes **what** you are interested in, not **how** it is acquired.
- Cypher is meant to be very **readable** and **expressive**.

Check your understanding

Question 1

What elements make up a graph?

Select the corrrect answers.

- [ ] tuples
- [ ] nodes
- [ ] documents
- [ ] relationships

Question 2

Suppose that you want to create a graph to model customers, products, what products a customer buys, and what products a customer rated. You have created nodes in the graph to represent the customers and products. In this graph, what relationships would you define?

Select the correct answers.

- [ ] BOUGHT
- [ ] IS_A_CUSTOMER
- [ ] IS_A_PRODUCT
- [ ] RATED

Question 3

What query language is used with a Neo4j Database?

Select the correct answer.

- [ ] SQL
- [ ] CQL
- [ ] Cypher
- [ ] OPath

Summary

You should now be able to:

- Describe what a graph database is.
- Describe some common use cases for using a graph database.
- Model a real-world scenario as a graph.