

Database Management Systems

Graph Databases

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

July, 2021



1 Basics

2 Property Graph Model

3 Neo4j

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Graph databases are part of the NoSQL databases created to address the limitations of the existing relational databases.

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Graph databases are part of the NoSQL databases created to address the limitations of the existing relational databases.

Querying relationships within a graph database is fast because they are perpetually stored within the database itself.

Property Graph Model

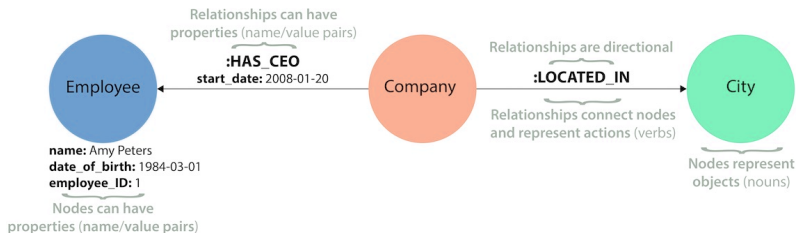
Property Graph Model is a key concept in Graph Databases. In this model, data is organized as nodes, relationships, and properties (data stored on the nodes or relationships).

Property Graph Model

Property Graph Model is a key concept in Graph Databases. In this model, data is organized as nodes, relationships, and properties (data stored on the nodes or relationships).

- Nodes are the entities in the graph. They can hold any number of attributes (key-value pairs) called properties. Nodes can be tagged with labels, representing their different roles in your domain. Node labels may also serve to attach metadata (such as index or constraint information) to certain nodes.
- Relationships provide directed, named, semantically-relevant connections between a pair of node entities. A relationship always has a direction, a type, a start node, and an end node. Like nodes, relationships can also have properties. In most cases, relationships have quantitative properties (e.g., weights, costs, ratings, time intervals, etc.).

Property Graph Model – An example



Neo4j

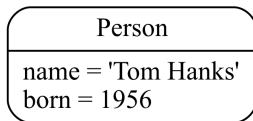
Neo4j is an open-source, NoSQL based, native graph database that provides an ACID-compliant transactional back-end for various applications.

Neo4j efficiently implements the Property Graph Model down to the physical level (i.e., the data is stored exactly as you connect it), and the database uses pointers to navigate and traverse the graph.

Property graphs in Neo4j – Nodes

Nodes are often used to represent entities. The simplest possible graph is a single node.

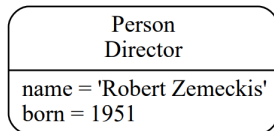
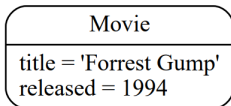
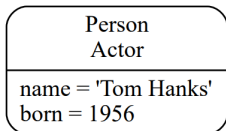
The following property graph consists of a single node.



Property graphs in Neo4j – Labels

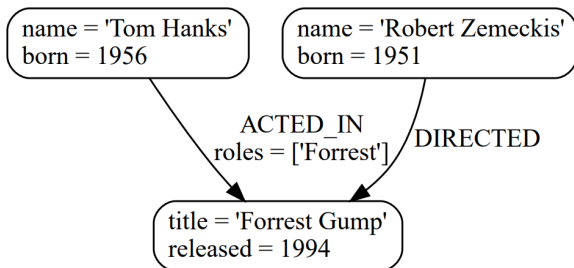
Labels are used to shape the domain by grouping nodes into sets where all nodes that have a certain label belongs to the same set. A node can have zero to many labels.

In the following example, by including additional labels to the nodes having the labels `Person` (one possible way of describing the data), we express different dimensions of the data.



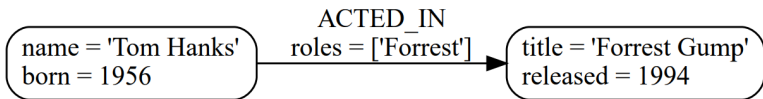
Property graphs in Neo4j – Relationships

A relationship connects two nodes. Relationships organize nodes into structures, allowing a graph to resemble a list, a tree, a map, or a compound entity – any of which may be combined into yet more complex, richly inter-connected structures.

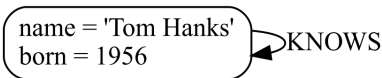


Property graphs in Neo4j – Relationship types

A relationship must have exactly one relationship type. Relationships always have a direction. However, you only have to pay attention to the direction where it is useful. This means that there is no need to add duplicate relationships in the opposite direction unless it is needed in order to properly describe your use case.



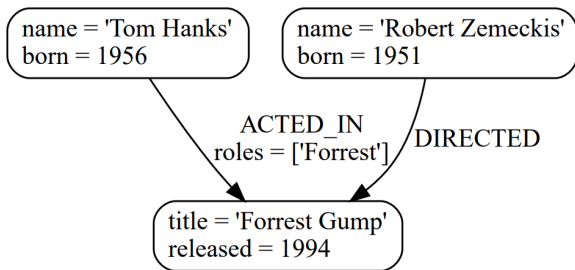
Note that, a node can have relationships to itself as shown below.



Property graphs in Neo4j – Properties

Properties are name-value pairs that are used to add qualities to nodes and relationships.

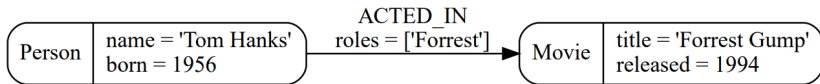
In the following example, we have used the properties `name` and `born` on `Person` nodes, `title` and `released` on `Movie` nodes, and the property `roles` on the `:ACTED_IN` relationship.



Property graphs in Neo4j – Paths

A traversal (visiting nodes by following relationships) is how you query a graph in order to find answers to questions. The traversal result is generally returned as a path. Note that, the shortest possible path has length zero and it contains a single node and no relationships.

For finding out which movies Tom Hanks acted in the example shown earlier, the traversal would start from the 'Tom Hanks' node, follow any `:ACTED_IN` relationships connected to the node, and end up with 'Forrest Gump' as the result as shown below.



Property graphs in Neo4j – Naming conventions

Graph Entity	Recommended Style
Node label	Camel case, beginning with an uppercase character
Relationship type	Upper case, using underscore to separate words
Property	Lower camel case, beginning with a lower-case character

More on Neo4j

Look into the Neo4j documentation:

<https://neo4j.com/docs>