Neo4j

https://neo4j.com/

Introduction : Graph Database

- A database with an explicit graph structure
- with nodes, edges, and properties to represent and store data
- Each node knows its adjacent nodes
- Thus provides index-free adjacency
- Graph databases are schemaless
- ***Native / built-in*** support to represent relationships
- ACID-compliant transactional DB
- Accessible from Java API , the ***Cypher query language***

What is a Graph?

- An abstract representation of a set of objects where some pairs are connected by links.

id	**Object (Vertex, Node)**
→	**Link (Edge, Arc, Relationship)**

Graph Database

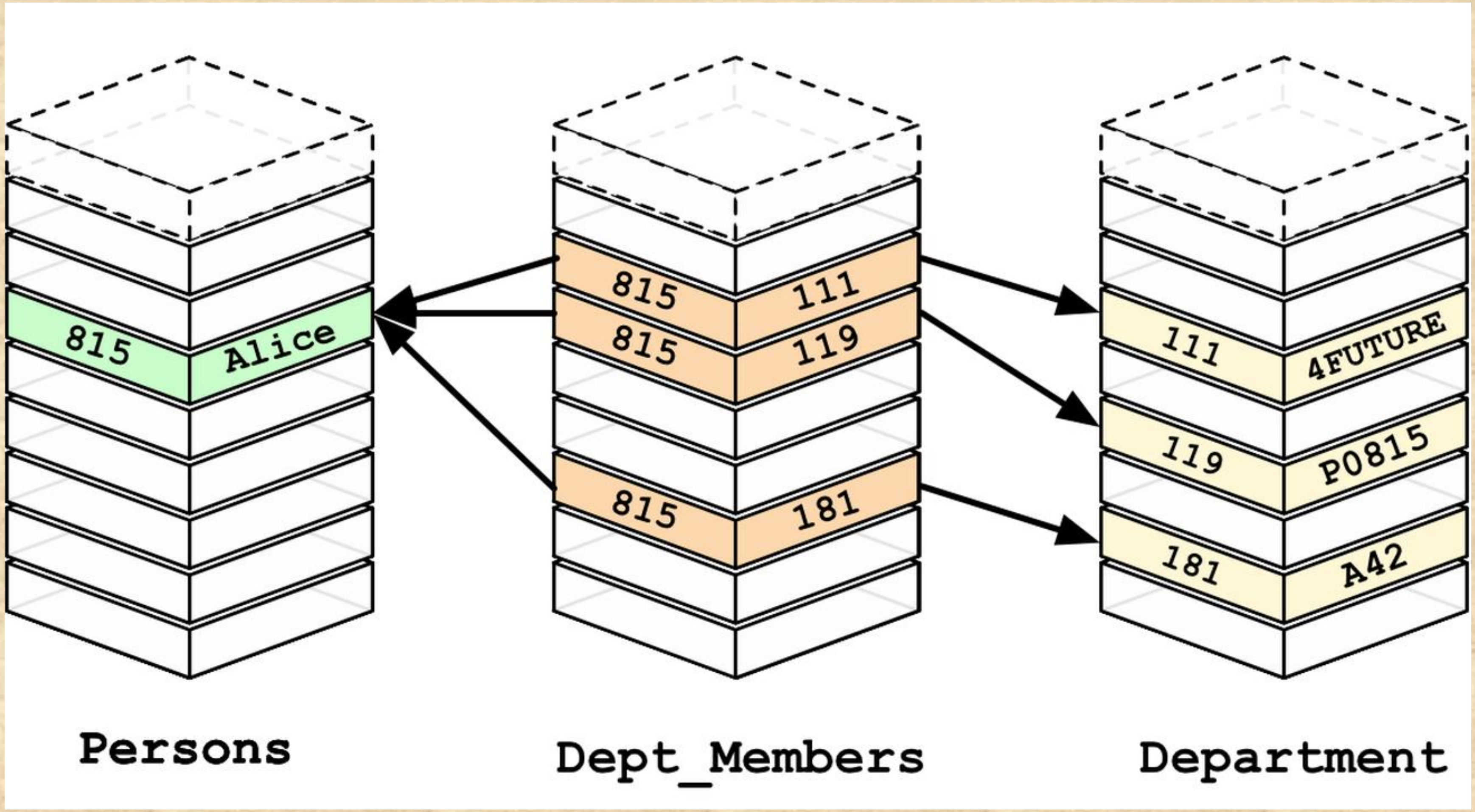
Data Model :

- ➢ **Nodes and**
- ➢ **Relationships**

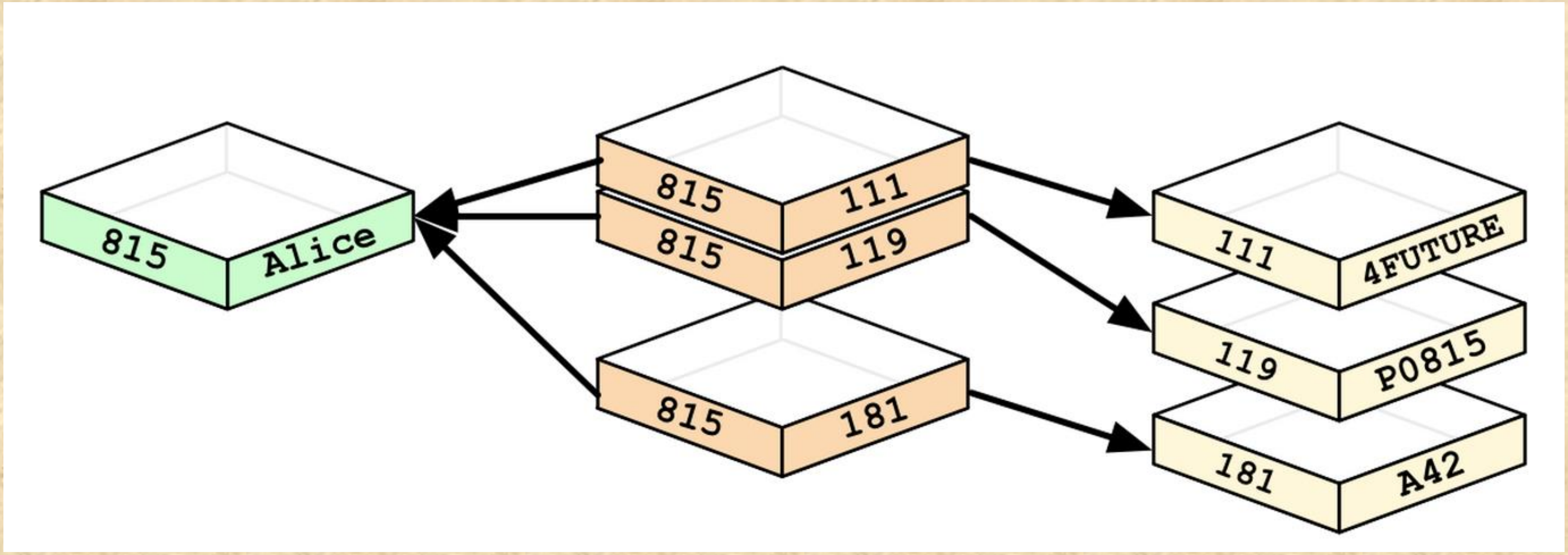
Graph DB vs Relational DB

Graph	Relational
1. Graphs	1. Tables
2. Nodes	2. Rows
3. Properties & its Values	3. Columns & Data
4. Relationships	4. Constraints
5. Traversal	5. Joins

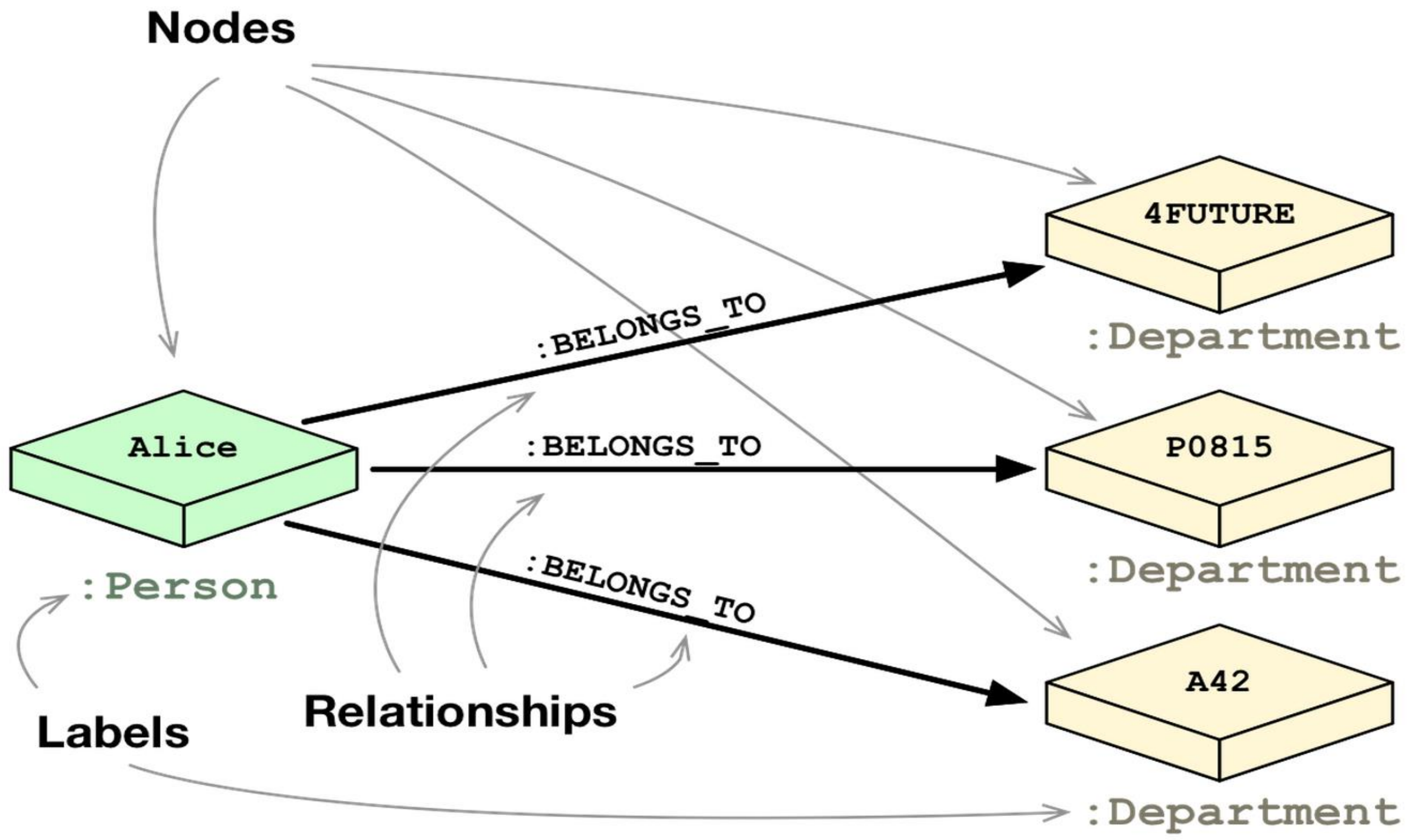
Relational Databases



Graph Databases



Graph Database

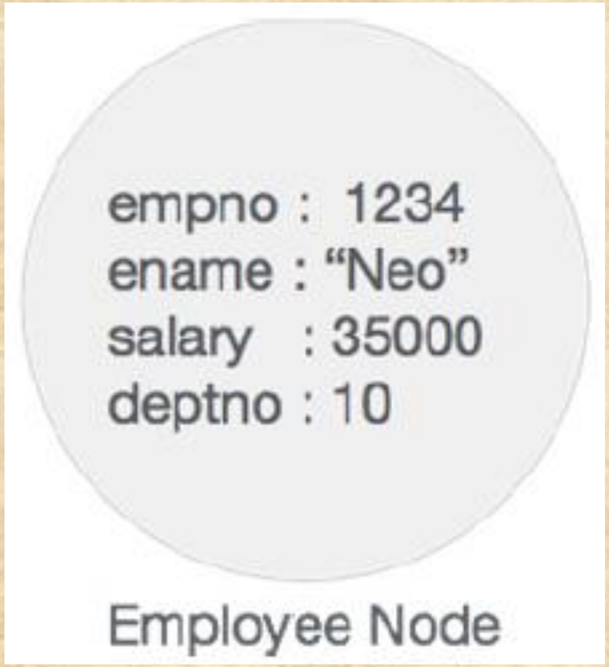


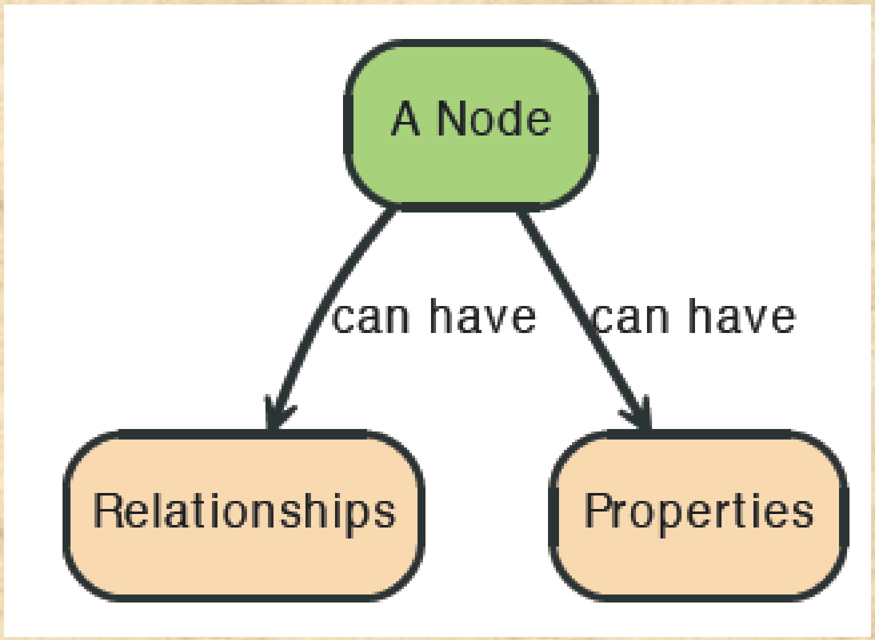
Mapping

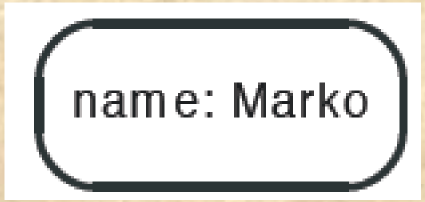
- Each entity table is represented by a label on nodes
- Each row in a entity table is a node
- Columns on those tables become node properties.
- Join tables are transformed into relationships, columns on those tables become relationship properties

Node in Neo4j : Fundamental unit

contains properties with key-value pairs

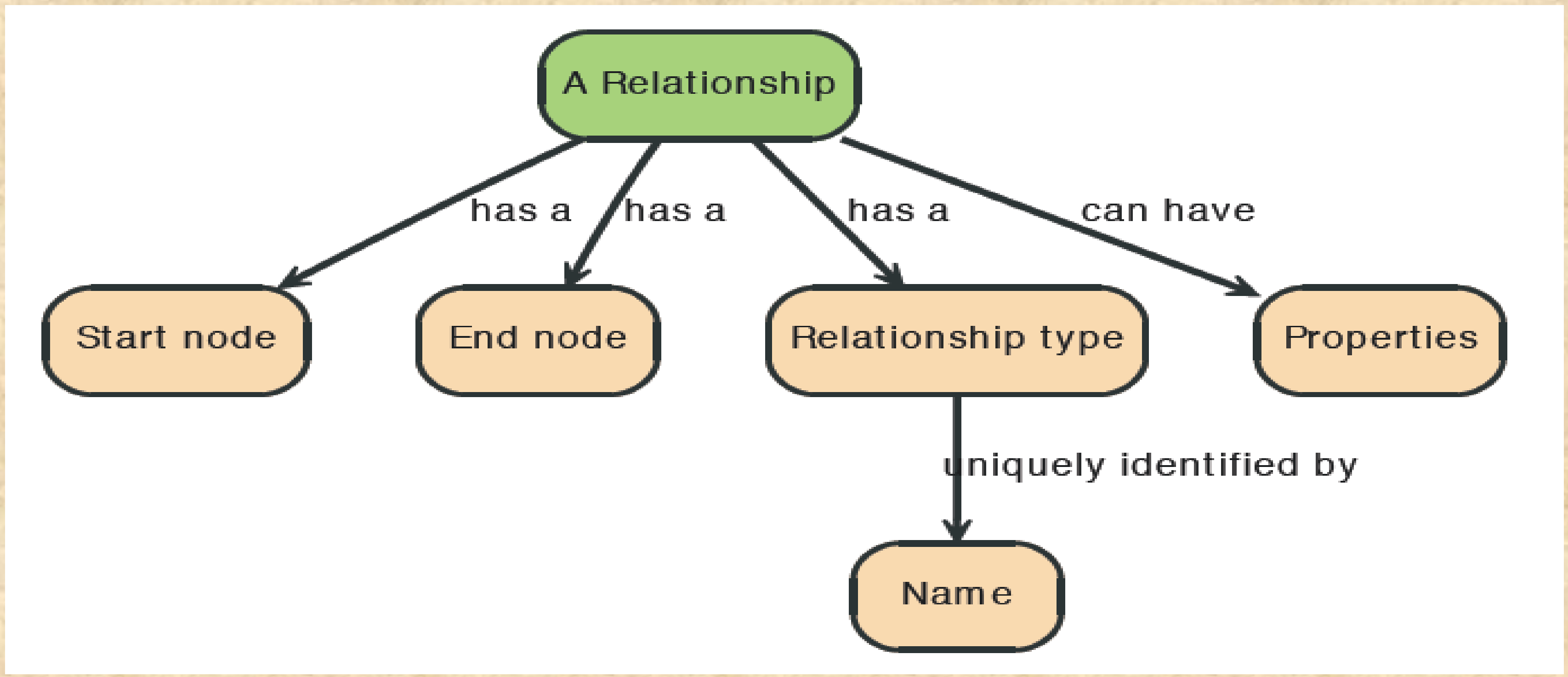




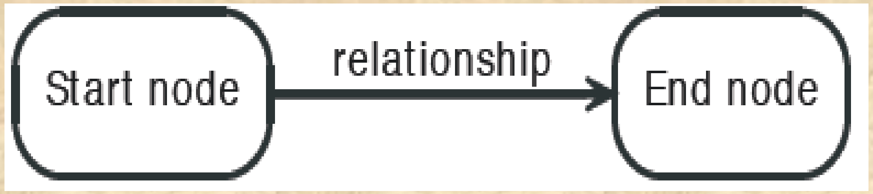


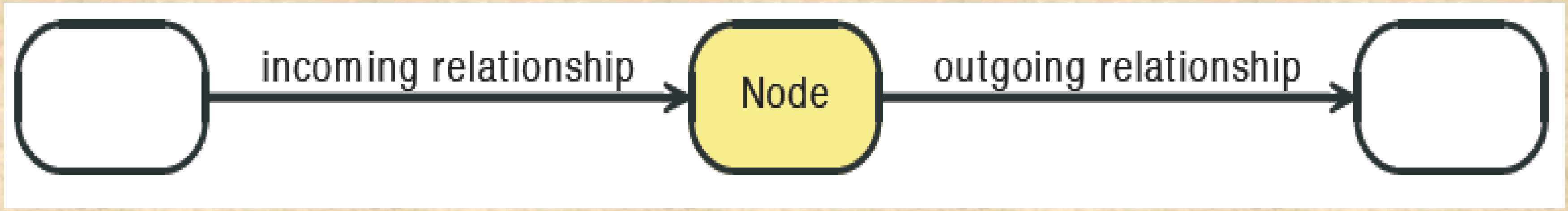
Relationships in Neo4j

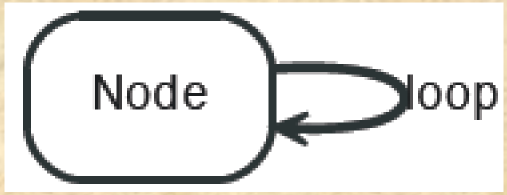
- Relationships between nodes are a key part of Neo4j.



Relationships in Neo4j







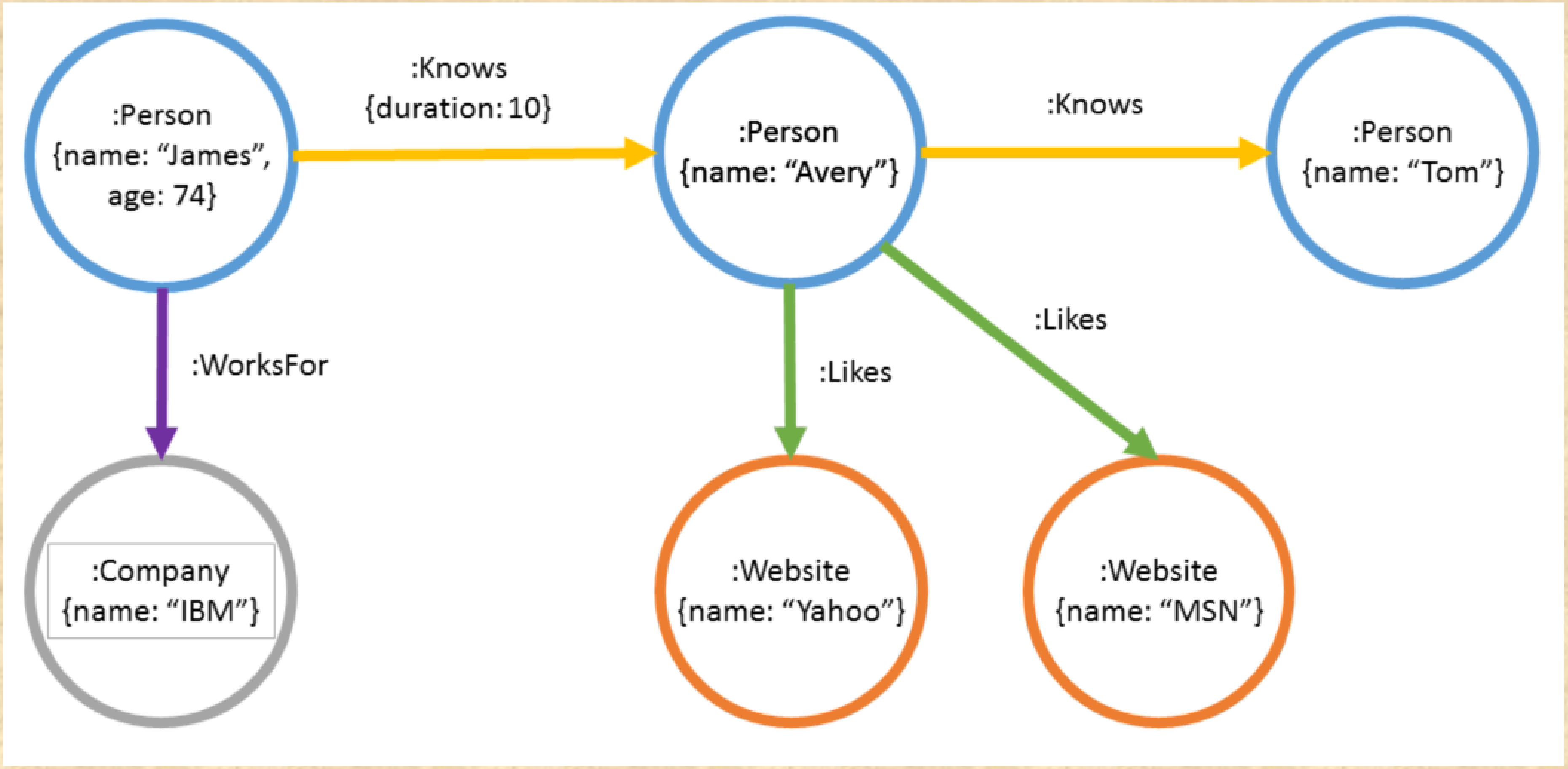
Properties

- Both nodes and relationships can have properties.
- Properties are key-value pairs where the key is a string.
- Property values can be either a primitive or an array of one primitive type.
- For example String, int and int[] values are valid for properties.

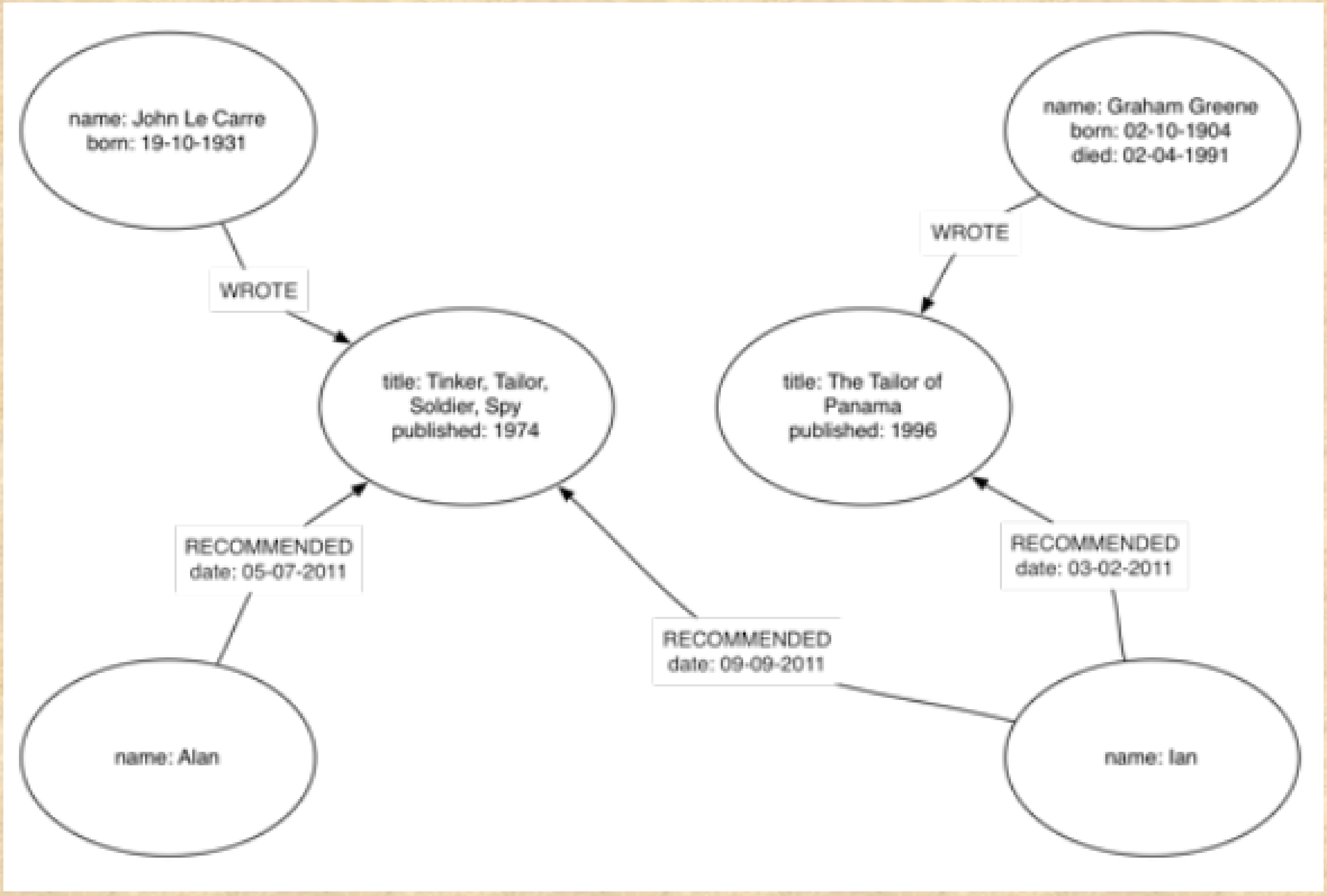
Supported data type in Neo4j

- **Number**, an abstract type, which has the subtypes **Integer** and **Float**
- **String**
- **Boolean**
- The spatial type **Point**
- Temporal types: **Date**, **Time**, **LocalTime**, **DateTime**, **LocalDateTime** and **Duration**

Example

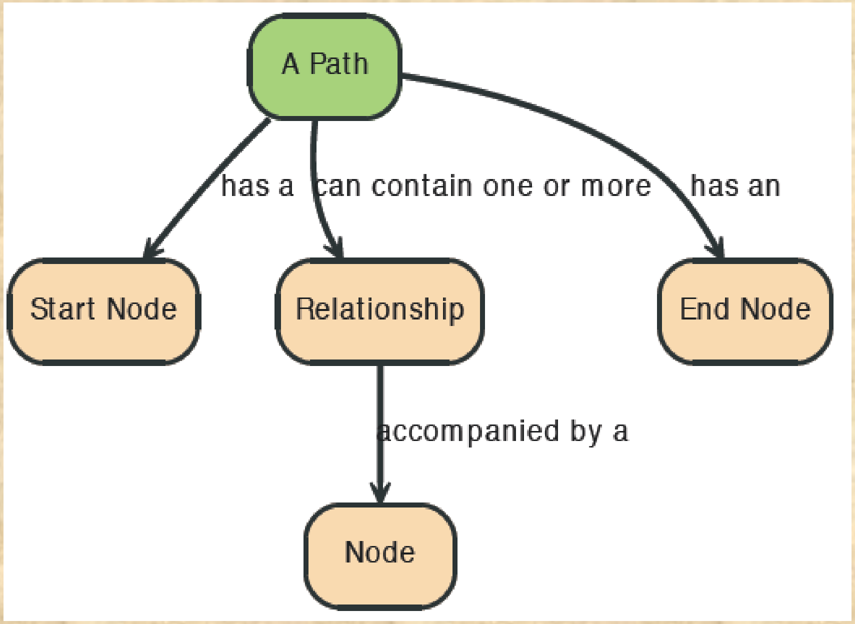

:Person
{name: "James",
age: 74}
:Knows
{duration: 10}
:Person
{name: "Avery"}
:Knows
:Person
{name: "Tom"}
:WorksFor
:Likes
:Likes
:Company
{name: "IBM"}
:Website
{name: "Yahoo"}
:Website
{name: "MSN"}

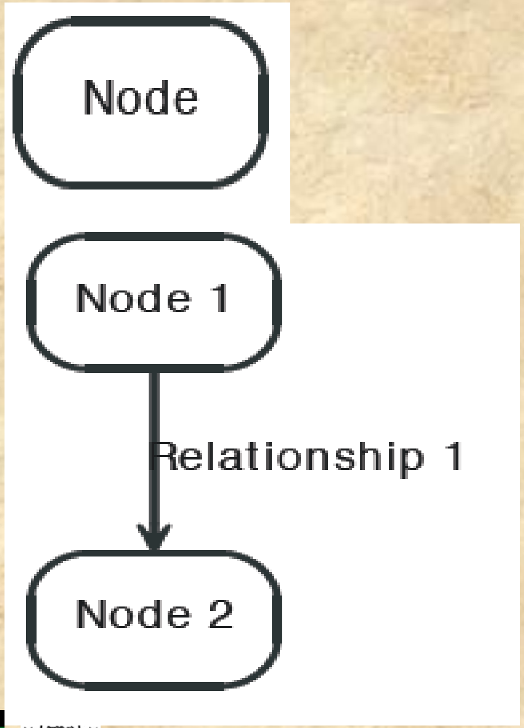
Example


name: John Le Carre
born: 19-10-1931
name: Graham Greene
born: 02-10-1904
died: 02-04-1991
WROTE
WROTE
title: Tinker, Tailor, Soldier, Spy
published: 1974
title: The Tailor of Panama
published: 1996
RECOMMENDED
date: 05-07-2011
RECOMMENDED
date: 03-02-2011
RECOMMENDED
date: 09-09-2011
name: Alan
name: Ian

Paths in Neo4j – Traversing : Query

- A path is one or more nodes with connecting relationships, typically retrieved as a query or traversal result.





Powered by

Adobe

cisco

Deutsche Telekom

mozilla

Walmart

viadeo

telenor

Lufthansa

research now

accenture
High performance. Delivered.

Pitney Bowes

CareerArcGroup

teachscape

Linked in

InfoJobs
EMPLEO

splink

squidoo

FIFTYTHREE

DRW TRADING GROUP

CHIP ONLINE

and more.

hp

Google™

PayPal™

ebay

Supported platforms

springdataNeo4j

REST://

ERLANG

Microsoft .net

JavaScript

Gremlin $G = (V, E)$

php

nodeJS

Certified for IBM DB2 e-business software

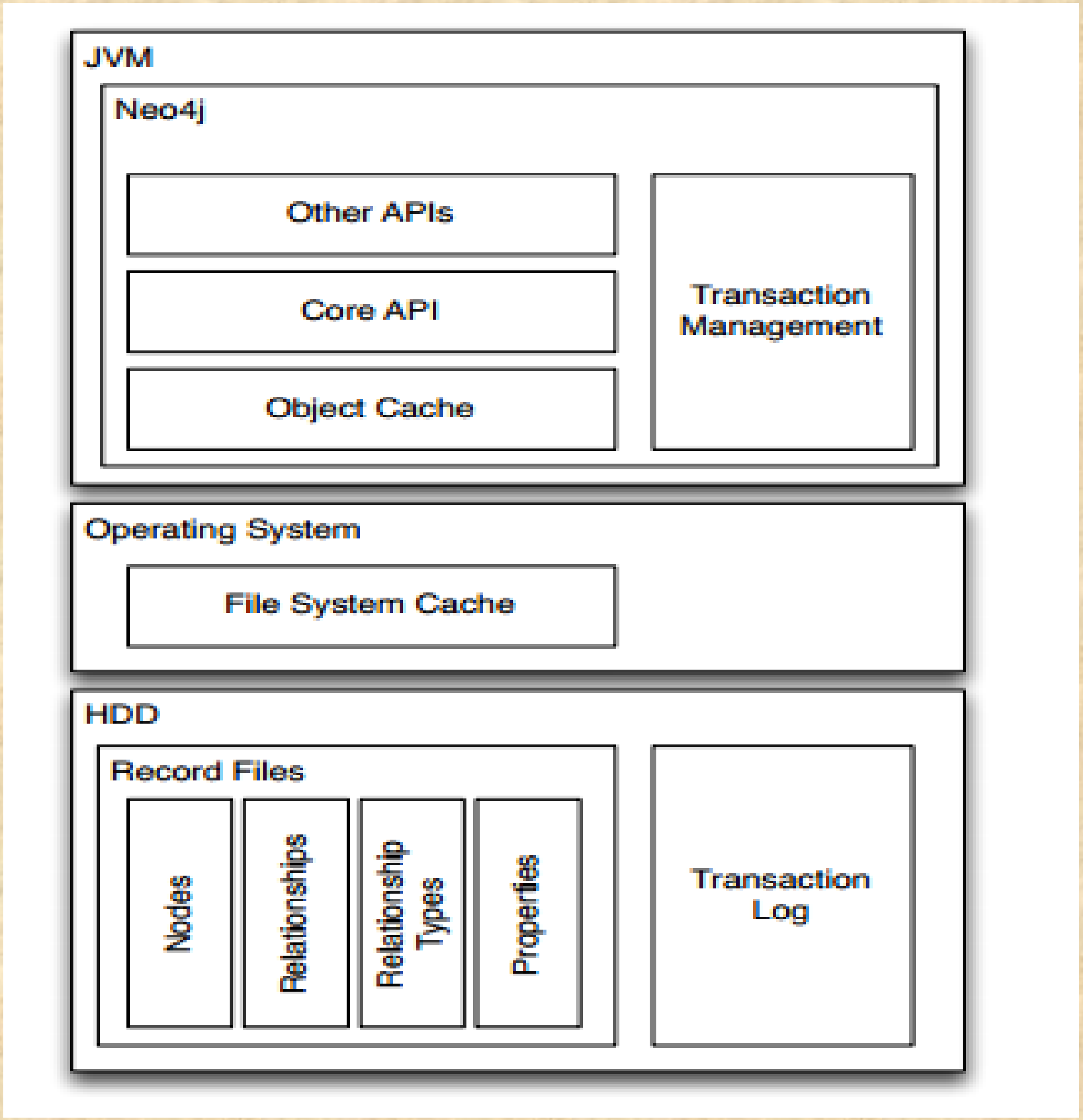
When to use Graph Databases ?

- Complex data
- Densely--connected, semi--structured domains
  - – Lots of join tables? Connectedness
  - – Lots of sparse tables? Semi--structure
- Data Model Volatility
- Easy to evolve
- Join Complexity and Performance
- Millions of "joins" per second
- Consistent query times as dataset grows

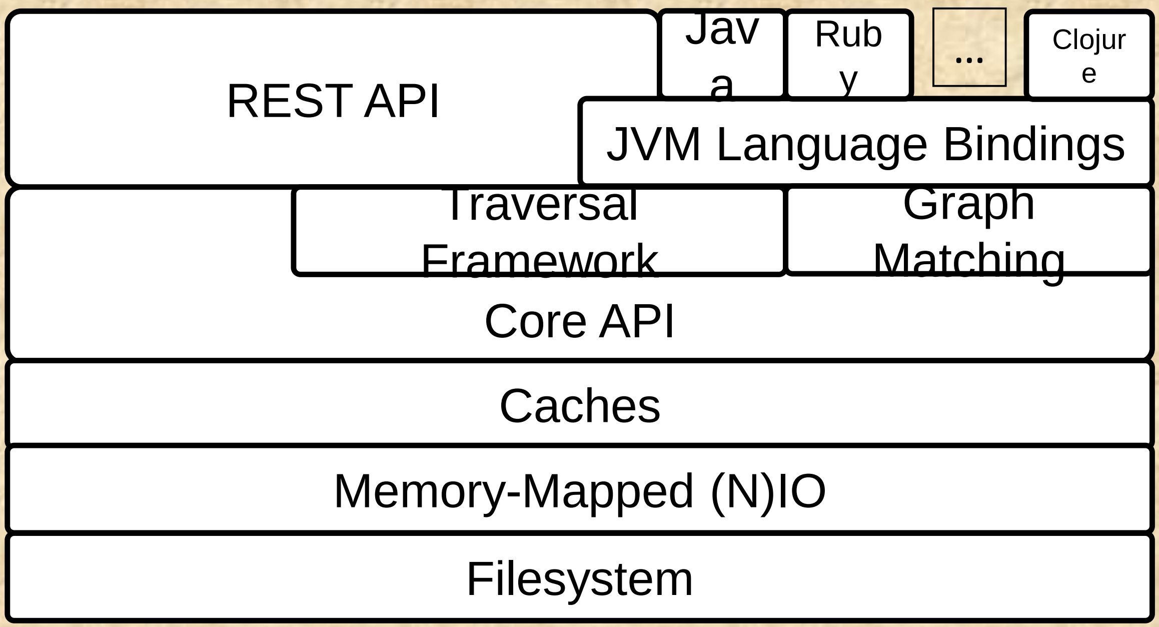
Target applications of graphs database

- Recommendations
- Business intelligence
- Social computing
- Geospatial
- Systems management
- Web of things
- Genealogy
- Time series data
- Product catalogue
- Web analytics
- Scientific computing (especially bioinformatics)
- Indexing your *slow* RDBMS
- And much more!

Neo4j Software Architecture



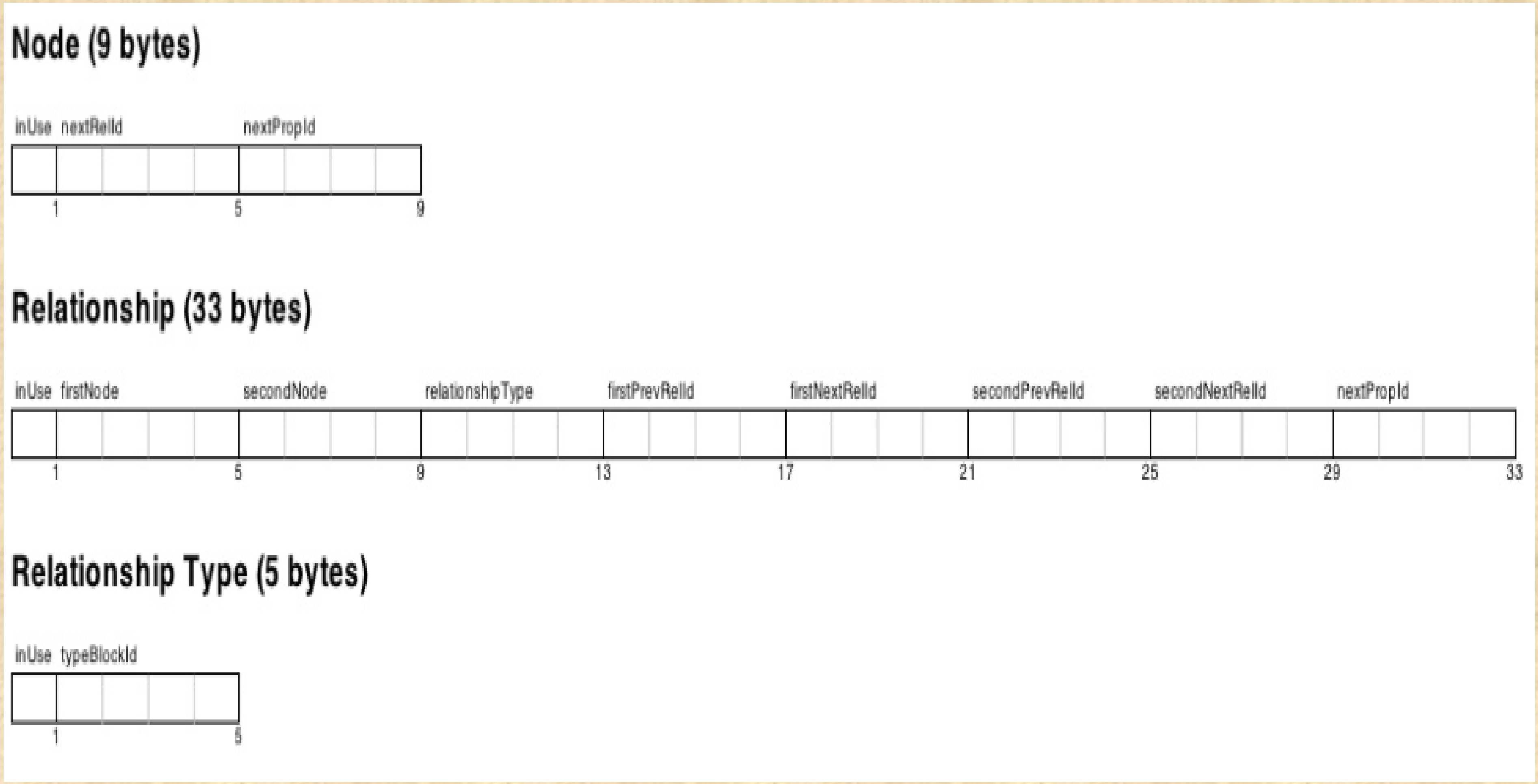
Neo4j Logical Architecture



Storage File Organization

- Neo4j stores graph data in a number of different store files.
- Each store file contains the data for a specific part of the graph e.g. **nodes**, **relationships**, **properties**

Store File Formats



Node store

- Size:9 bytes
  - **First byte**:in-use flag
  - **Next 4 bytes**:ID of first relationship
  - **Last 4 bytes**:ID of first property of node

Relationship store

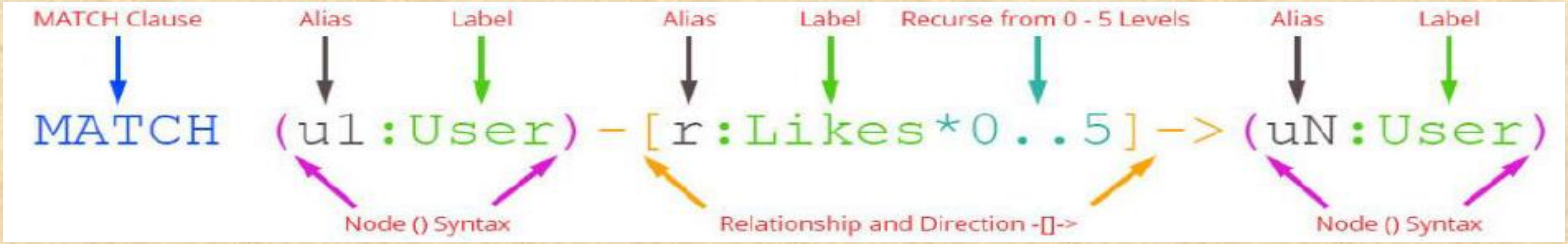
- Size:33 bytes
    - **First byte**:In use flag
    - **Next 8 bytes**:IDs of the nodes at the start and end of the relationship
    - **4 bytes**:Pointer to the relationship type
    - **16 bytes**:pointers for the next and previous relationship records for each of the start and end nodes
    - **4 bytes**:next property id

Cypher Query Langauge (CQL)

- SQL has been the de facto language for RDBMS
- Cypher is a declarative language that serves the same purpose as SQL
- Uses ASCII-Art to represent patterns
- Nodes are surrounded with parentheses
- Use arbitrary variables to refer to nodes Variable scope restricted to single statement
- Case Sensitive – standard naming convention
- https://neo4j.com/docs/developer-manual/current/cypher/syntax/naming/

Cypher Query : Syntax

- Relationships are specified using an arrow (- ->) between nodes
- Square bracket inside arrow for specification
  - Relationships - 1 type
  - Nodes - 0 or more labels
- Cypher allows patterns to be assigned to variables that increase modularity and reduce repetition



```
MATCH (u1:User)-[r:Likes*0..5]->(uN:User)
```

Cypher Query Clauses

- Minimum/simplest query consist of a **MATCH** clause followed by a **RETURN** clause.

```
MATCH (a:Person {name:'Jim'})-[:KNOWS]->(b)-[:KNOWS]-
>(c), (a)-[:KNOWS]->(c)

RETURN b, c
```

- **WHERE :** Provides criteria for filtering pattern matching results.
- **CREATE and CREATE UNIQUE :** Create nodes and relationships.
- **MERGE :** Ensures that the supplied pattern exists in the graph, either by reusing existing nodes and relationships that match the supplied predicates, or by creating new nodes and relationships

Cypher Query Clauses …

- **DELETE/REMOVE :** Removes nodes, relationships, and properties.
- **SET :** Sets property values and labels
- **ORDER BY :** Sorts results as part of a **RETURN**
- **SKIP LIMIT :** Skip results at the top and limit the number of results
- **FOREACH :** Performs an updating action for each element in a list.
- **UNION :** Merges results from two or more queries.
- **WITH :** Chains subsequent query parts and forwards results from one to the next. Similar to piping commands in Unix.
- For more detail https://neo4j.com/docs/cypher-refcard/current/?ref=beginners-ebook

Getting started

- Multiple ways to start
  - Neo4j Sandboxes (cloud containers)
  - VMs (VirtualBox - Windows & Mac)
  - VMs (Linux - VirtualBox or KVM)
  - **Desktop installation**
  - **Server – single instance**
  - **Clustering** – Enterprise

Neo4j Database Server Installation on Windows Machine

- Pre-requisite : JDK 8.0
- Visit the Neo4j official site using https://neo4j.com/try-neo4j/
- Download Neo4J Community Server Edition
- Download setup/run or **zip file/extract**
- Place the extracted files in a permanent home on your server, for example **D:\neo4j\**. The top level directory is referred to as **NEO4J_HOME**.

Installation …

- To run Neo4j as a console application, use:

  <NEO4J_HOME>\bin\neo4j console

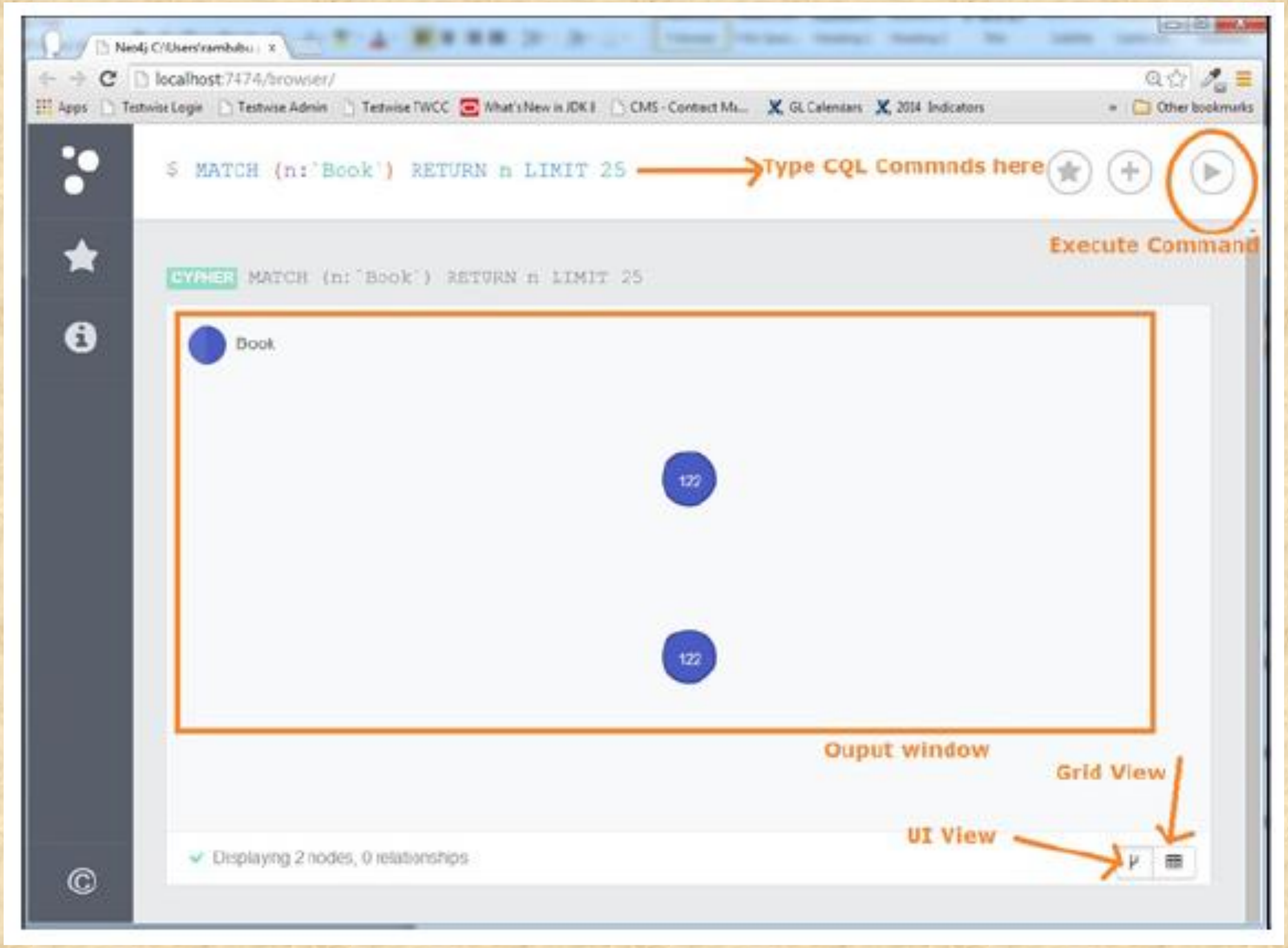
- To install Neo4j as a service use:

  <NEO4J_HOME>\bin\neo4j install-service

- Start Neo4J browser : **http://localhost:7474**
- Connect using the **username 'neo4j'** with default **password 'neo4j'**. You'll then be prompted to change the password.
- **$** :play movie graph
- **$** :play northwind graph

Neo4j Data Browser

- Open using URL **http://localhost:7474/browser/**



```
$ MATCH (n:`Book`) RETURN n LIMIT 25
```

CYPHER MATCH (n:`Book`) RETURN n LIMIT 25

Displaying 2 nodes, 0 relationships

Remote Access to Data Browser

- Open the `neo4j.conf` file in an editor
- Add following entries in **HTTP Connector** section
- **dbms.connector.http.type=HTTP**
- **dbms.connector.http.enabled=true**
- **dbms.connector.http.address=0.0.0.0:7474**
- **Instead of 0.0.0.0 , put here actual IP**
- **E.g. 10.10.7.101**
- **dbms.connector.http.address=10.10.7.101:7474**

Application Developments

- Connecting through programming languages
- Neo4j officially supported drivers
  - Java
  - Javascript
  - **C#**
  - **Python**

Neo4j for C#.NET Developers

- PM> Install-Package **Neo4j.Driver-4.2.0**
- Neo4j Community Drivers
- **Neo4jClient :** A .NET client for Neo4j, which makes it easy to write Cypher queries in C# with IntelliSense
- GitHub Link : https://github.com/DotNet4Neo4j/neo4jclient

Neo4j Python Driver

- Find out / download the latest version of the driver at **https://pypi.python.org/pypi/neo4j-driver**
- Or Install the latest version of the driver if you are online :

```
pip install neo4j
```