Graph Theory

Dr. Irfan Yousuf
Department of Computer Science (New Campus)
UET, Lahore

(Lecture # 27; May 03, 2023)

Outline

- Dominating Set in Graphs
- Graph Database

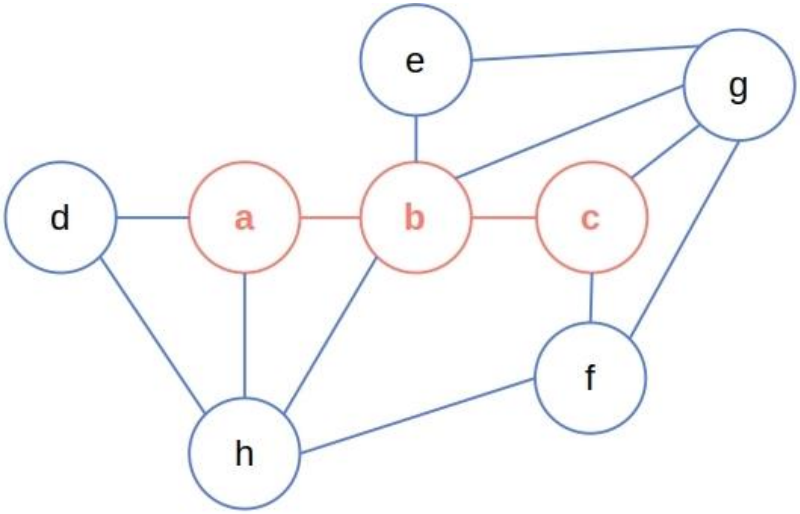
Dominating Set

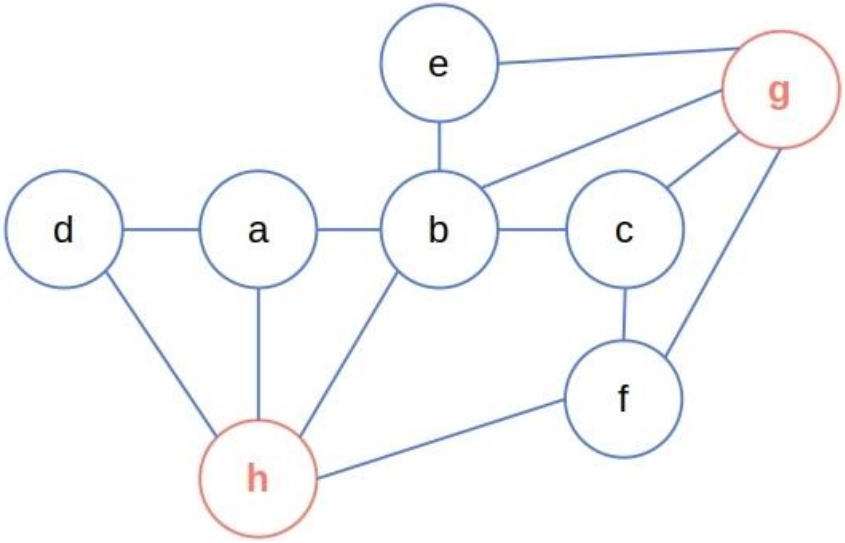
- A dominating set D of the graph G=(V, E) is any subset of V such that any node out of D is adjacent to at least one node from it.

Formally, $D \subseteq V$ is a dominating set of G if:

$$(\forall v \in V \setminus D)(\exists u \in D)(u \text{ and } v \text{ are neighbors})$$

Dominating Set





Domination Number

The domination number $\gamma(G)$ of a graph G is the cardinality of its smallest dominating set:

$$\gamma(G) = \min \{|S| : S \subseteq V \text{ and } S \text{ is a dominating set of } G\}$$

Dominating Set: Application

- Network Protocols / Routing in Ad-hoc networks
- Compiler
- Social Networks
- Flow-Networks

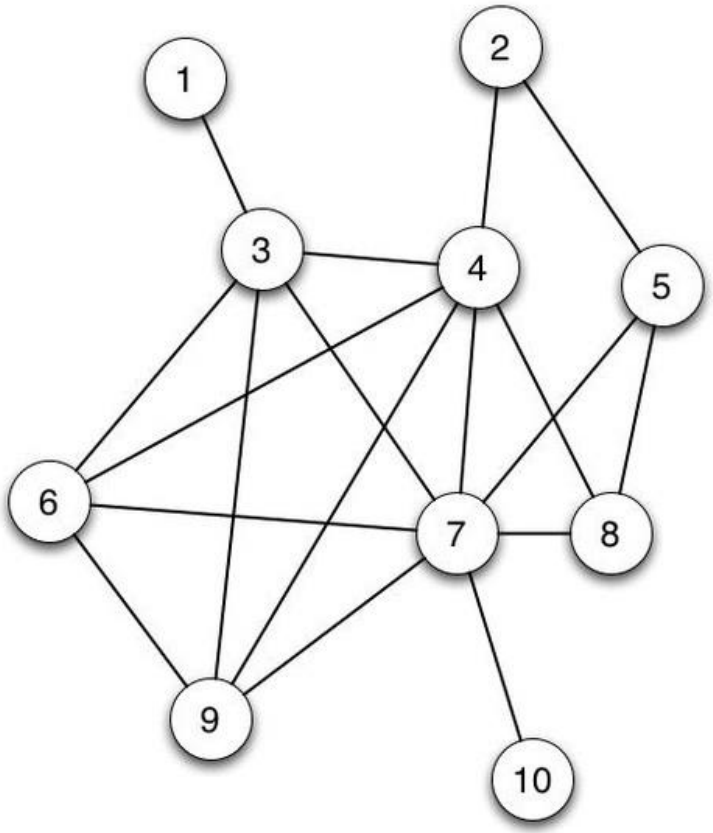
Dominating Set Problem

- The dominating set problem concerns testing whether $\gamma(G) \leq K$ for a given graph G and input K.
- It is a classical NP-complete decision problem in computational complexity theory.
- Therefore, it is believed that there may be no efficient algorithm that can compute $\gamma(G)$ for all graphs G.
- However, there are efficient approximation algorithms, as well as efficient exact algorithms for certain graph classes.

A Naïve Algorithm

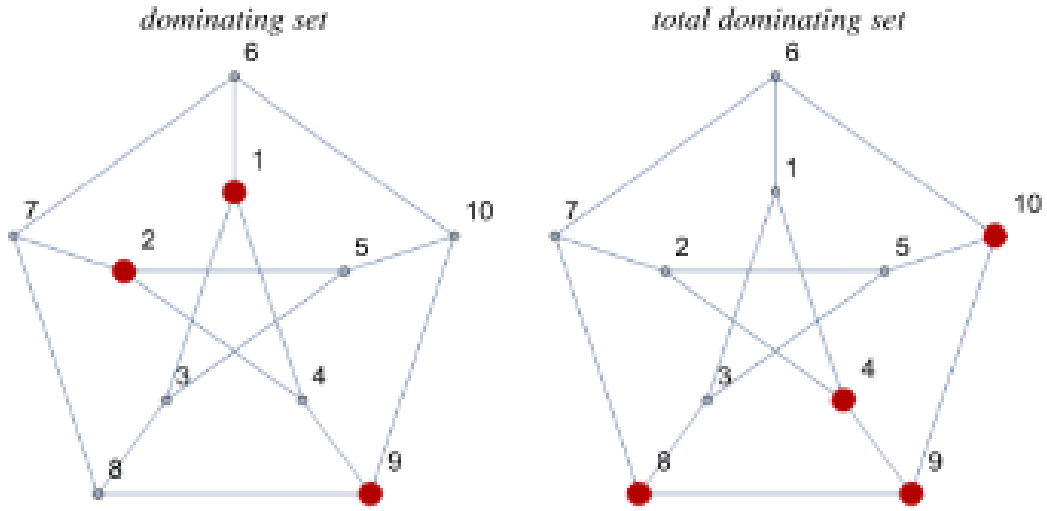
- First, we have to initialize a set ‘S’ as empty
- Take any edge ‘e’ of the graph connecting the vertices (say A and B)
- Add one vertex between A and B (let say A) to our set S
- Delete all the edges in the graph connected to A
- Go back to step 2 and repeat, if some edge is still left in the graph
- The final set S is a Dominant Set of the graph

A Naïve Algorithm: Example



Total Dominating Set

- A total dominating set (or strongly-dominating set) is a set of vertices such that all vertices in the graph, including the vertices in the dominating set themselves, have a neighbor in the dominating set.



K-Dominating Set

- A k-dominating set is a set of vertices such that each vertex not in the set has at least k neighbors in the set.

Special Topics in Graph Theory

Graph Databases

- Data model is represented by nodes and relationships
- Uses graph structures to semantically represent objects and relationships
- Relationships are first class citizens and can have properties on their own
- Allows simple and fast retrieval of complex hierarchical structures
- Directly relates data items in the store to allow data to be linked together

Why Graph Databases?

"Relational database is not good for relationship data" - JOIN

- Friend links on a social network

- "People who bought this also bought..." Amazon-style recommendation

Building Blocks

- Nodes
- Relationships
- Properties
- Labels

Nodes

- Nodes represent entities and complex types
- Nodes can contain properties
- Each node can have different properties

Relationships

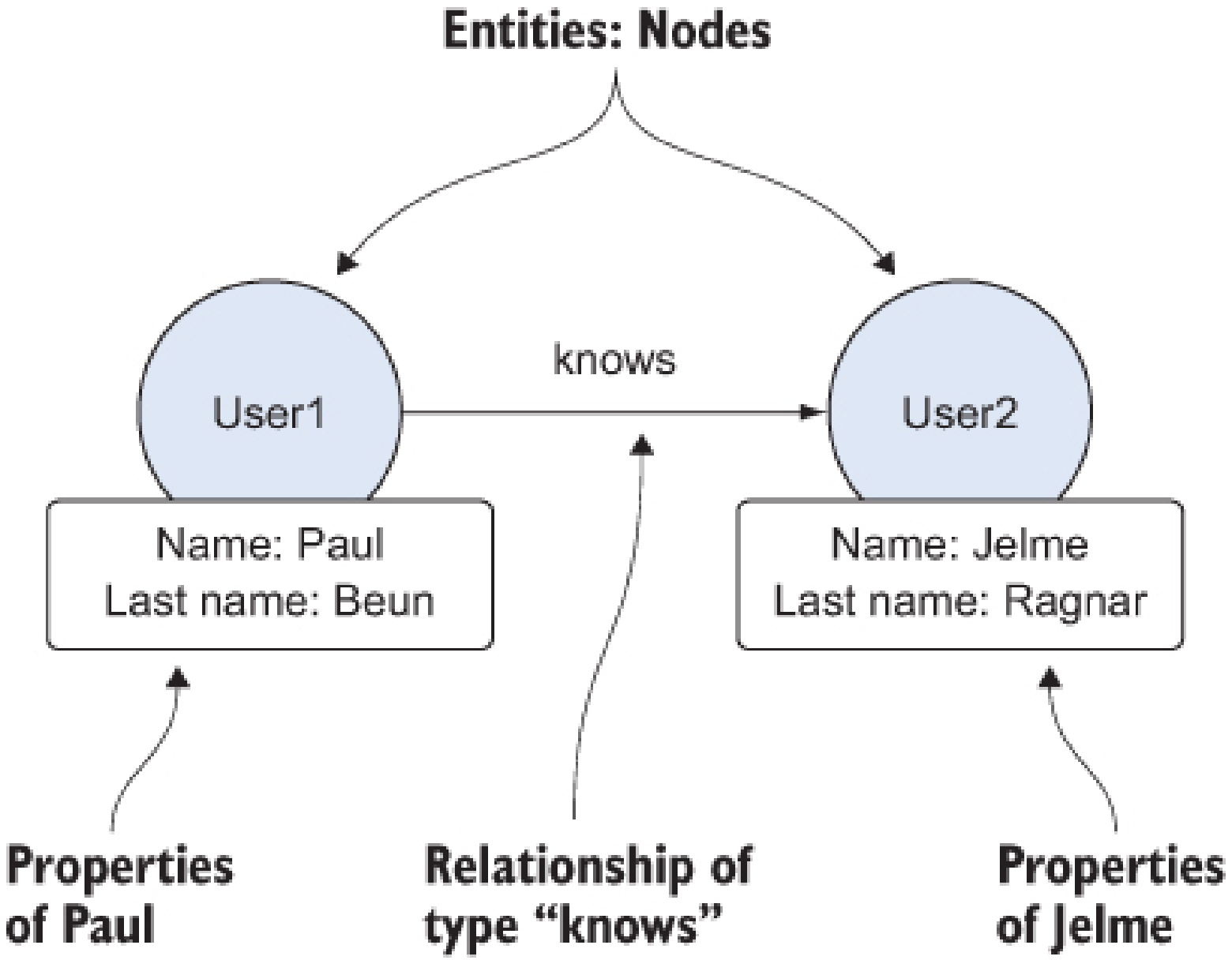
- Every relationship has a name and direction
- Relationships can contain properties, which can further clarify the relationship
- Must have a start and end node

Properties

- Key value pairs used for nodes and relationships
- Adds metadata to your nodes and relationships
- Entity attributes
- Relationship qualities

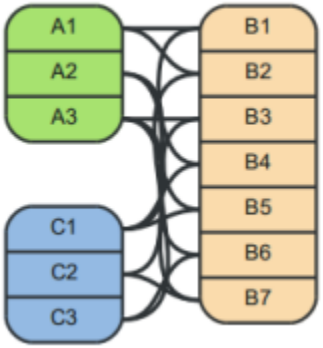
Labels

- Used to represent objects in your domain (e.g. user, person, movie)
- With labels, you can group nodes
- Allows us to create indexes and constraints with groups of nodes

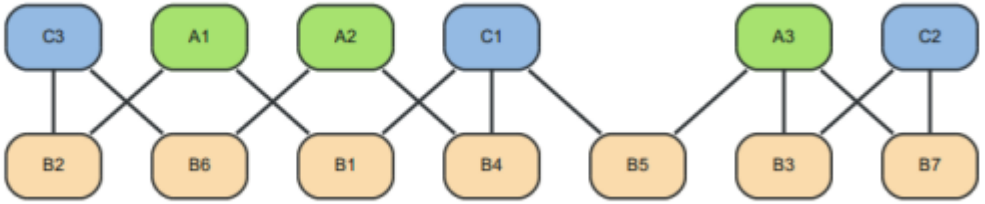

Entities: Nodes
User1
knows
User2
Name: Paul
Last name: Beun
Name: Jelme
Last name: Ragnar
Properties of Paul
Relationship of type “knows”
Properties of Jelme

Graph Database

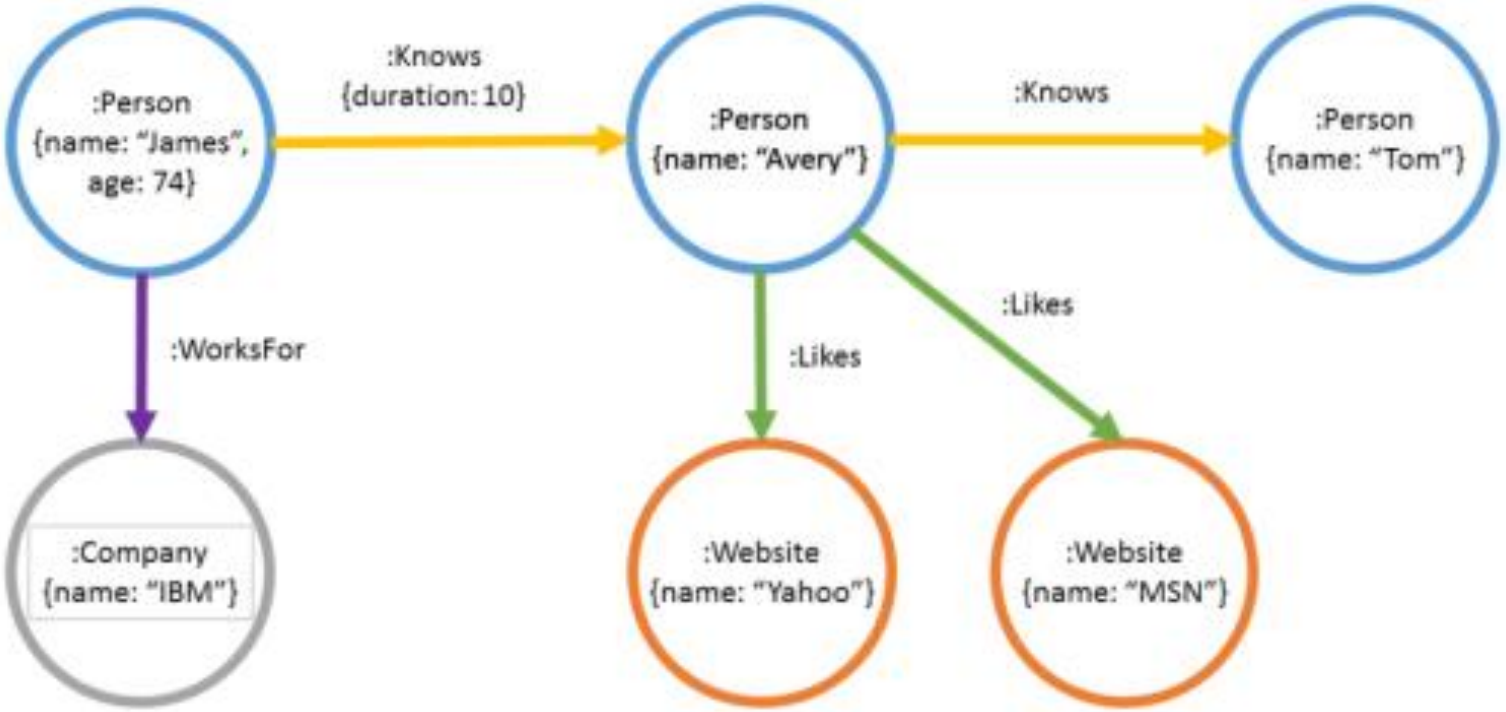
RDBMS



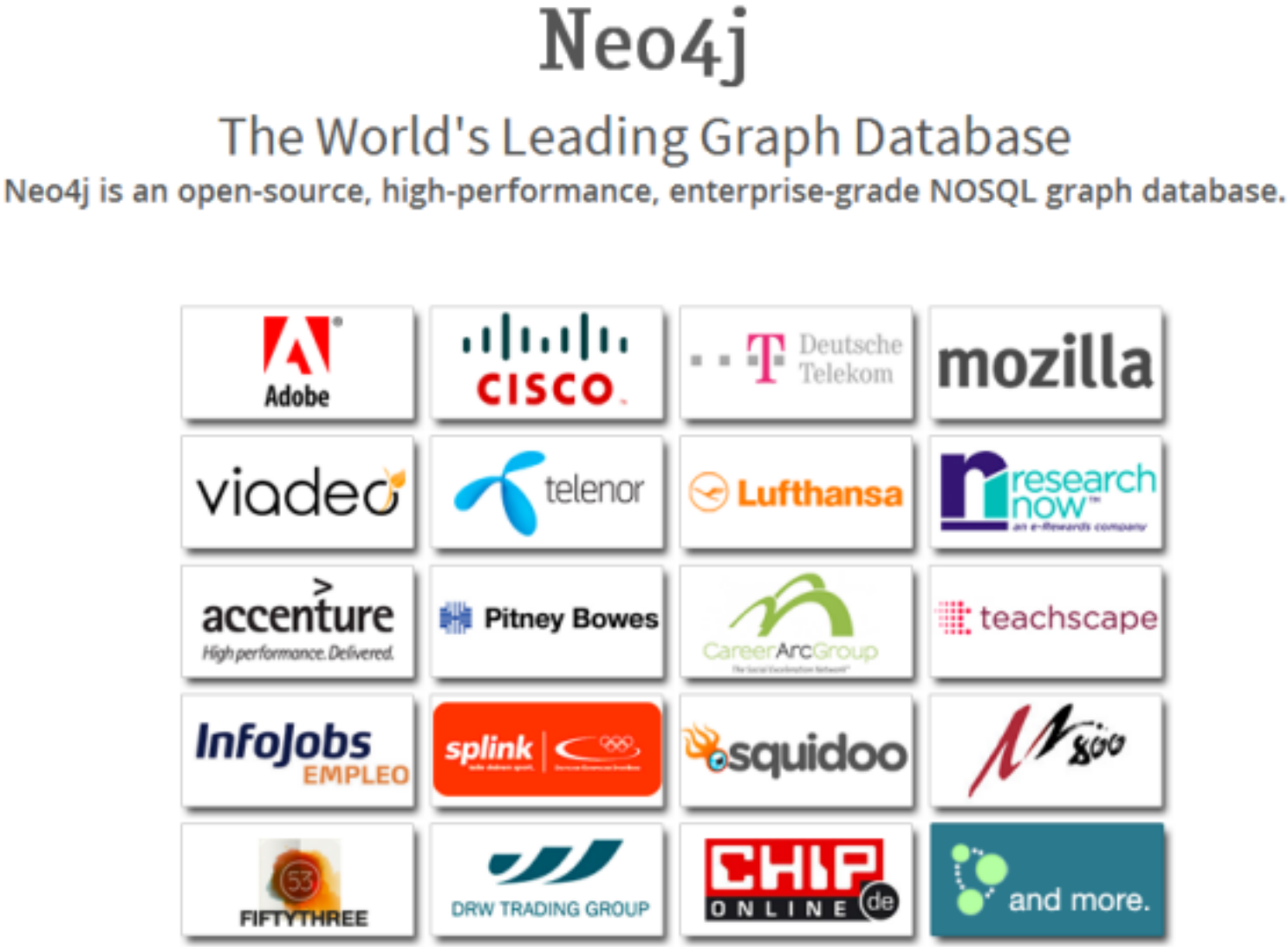
Graph Database



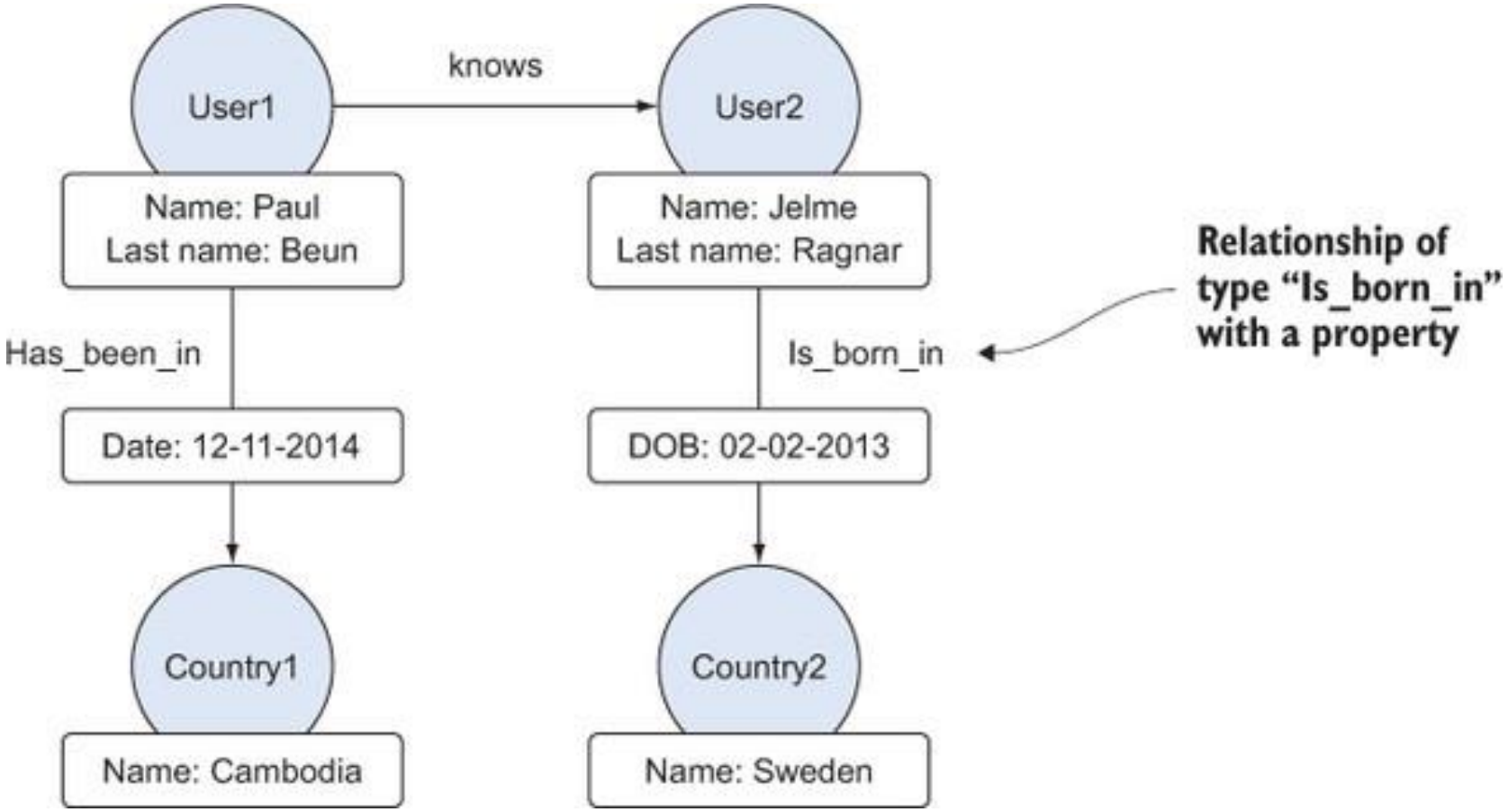
Graph Database



Graph Database



Graph Database



Graph Database

A starting node—In this case the node with name property “Paul”

A traversal path—In this case a path starting at node Paul and going to Cambodia

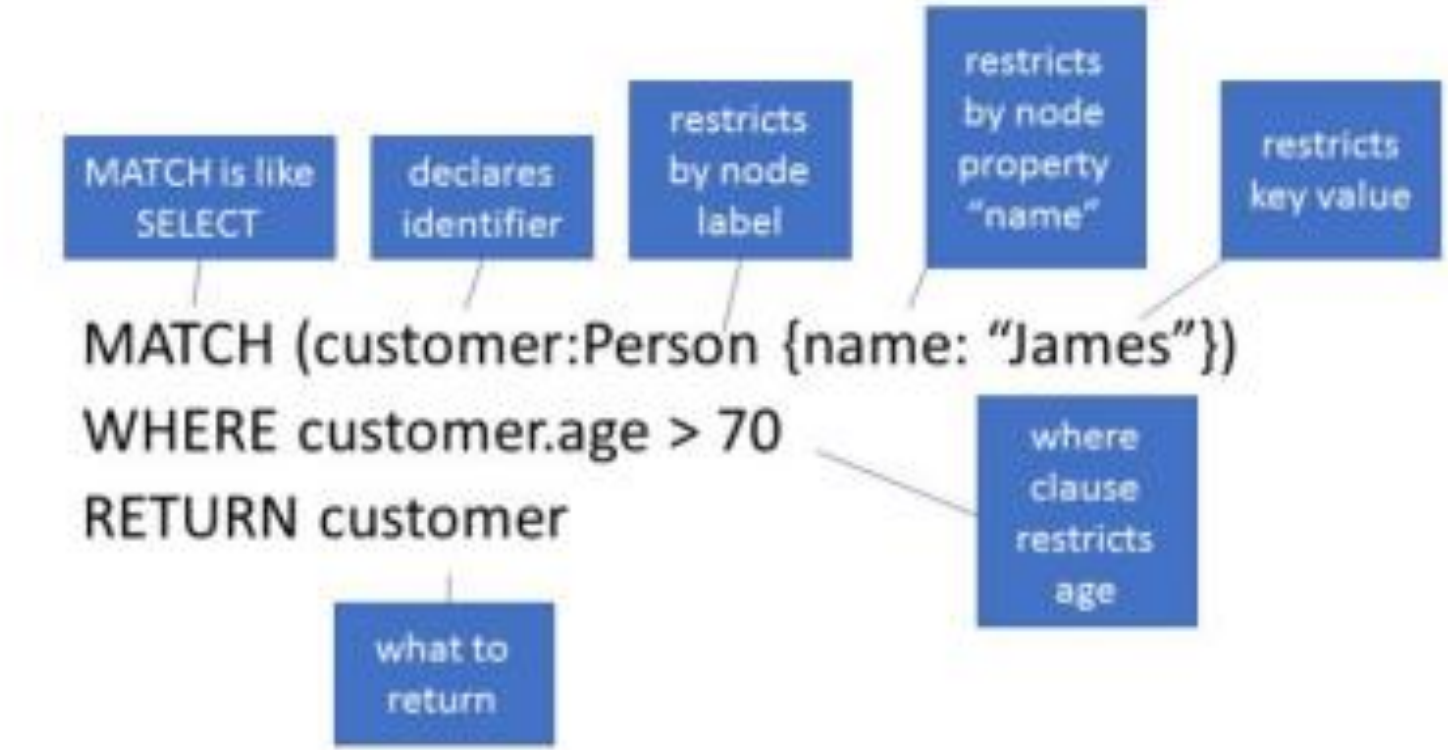
End node—Country node with name property “Cambodia”

Graph Database

Cypher: Neo4j's query language

- Declarative query language based on SQL
- Matches patterns in graph to
  - retrieve, add, update, delete data
- Manages indexes/constraints

Cypher


MATCH is like SELECT
declares identifier
restricts by node label
restricts by node property "name"
restricts key value
MATCH (customer:Person {name: "James"})
WHERE customer.age > 70
RETURN customer
where clause restricts age
what to return

Cypher

```
MATCH (customer:Person {name: "James"}) – [:Knows] -> (friend:Person)
        –[:Likes]->(:Website {name: "Yahoo"})
RETURN friend
```

Graph Database

Graph Databases: Pros and Cons

- Pros:
  - Powerful data model, as general as RDBMS
  - Connected data locally indexed
  - Easy to query
- Cons
  - Sharding (lots of people working on this)
    - Scales UP reasonably well
  - Requires rewiring your brain

Summary

- Graph Database