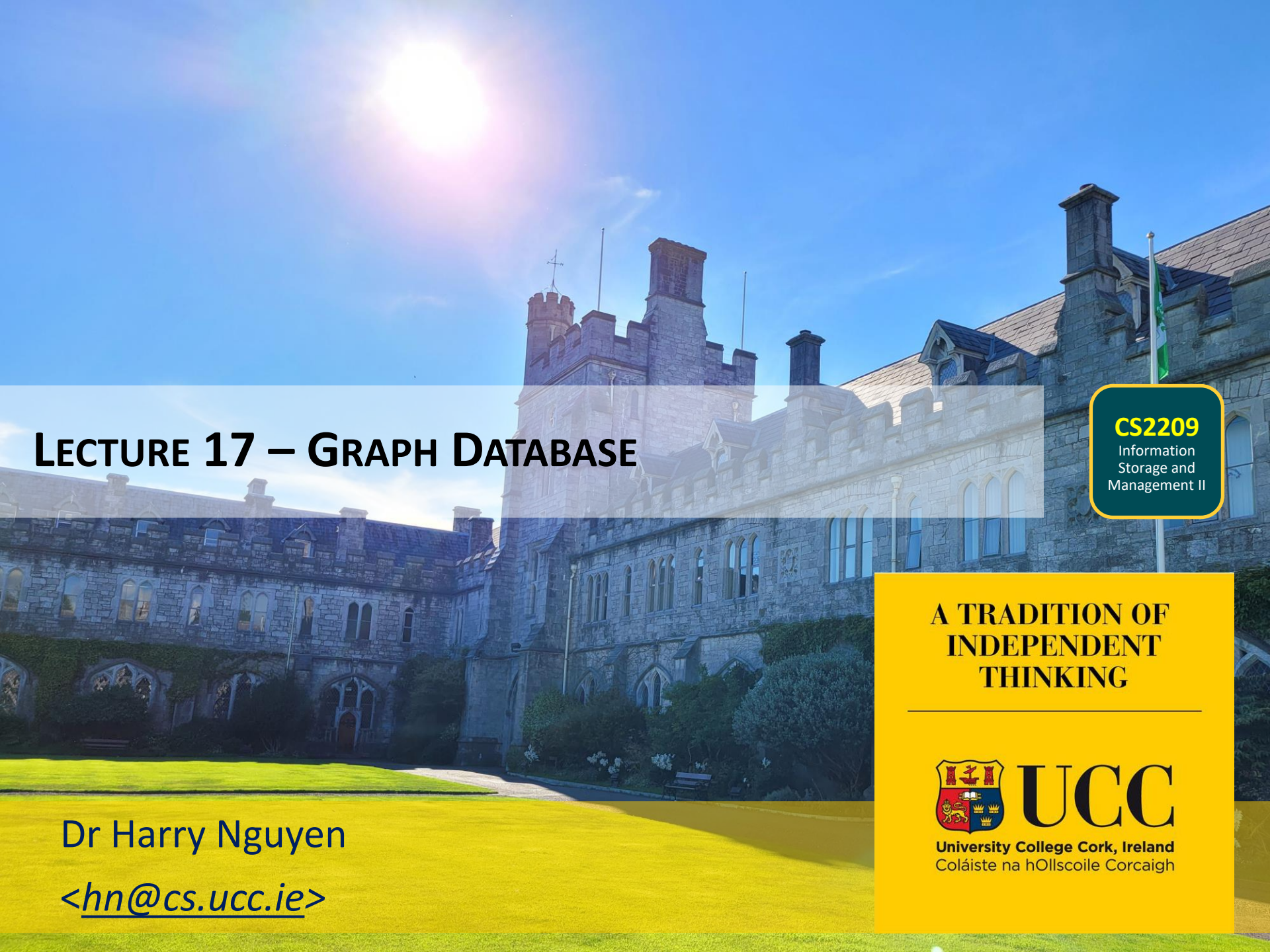


LECTURE 17 – GRAPH DATABASE

CS2209
Information Storage and Management II

A TRADITION OF INDEPENDENT THINKING

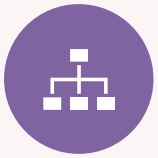
UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Dr Harry Nguyen
<hn@cs.ucc.ie>

What We Saw Last Lecture

Graphs

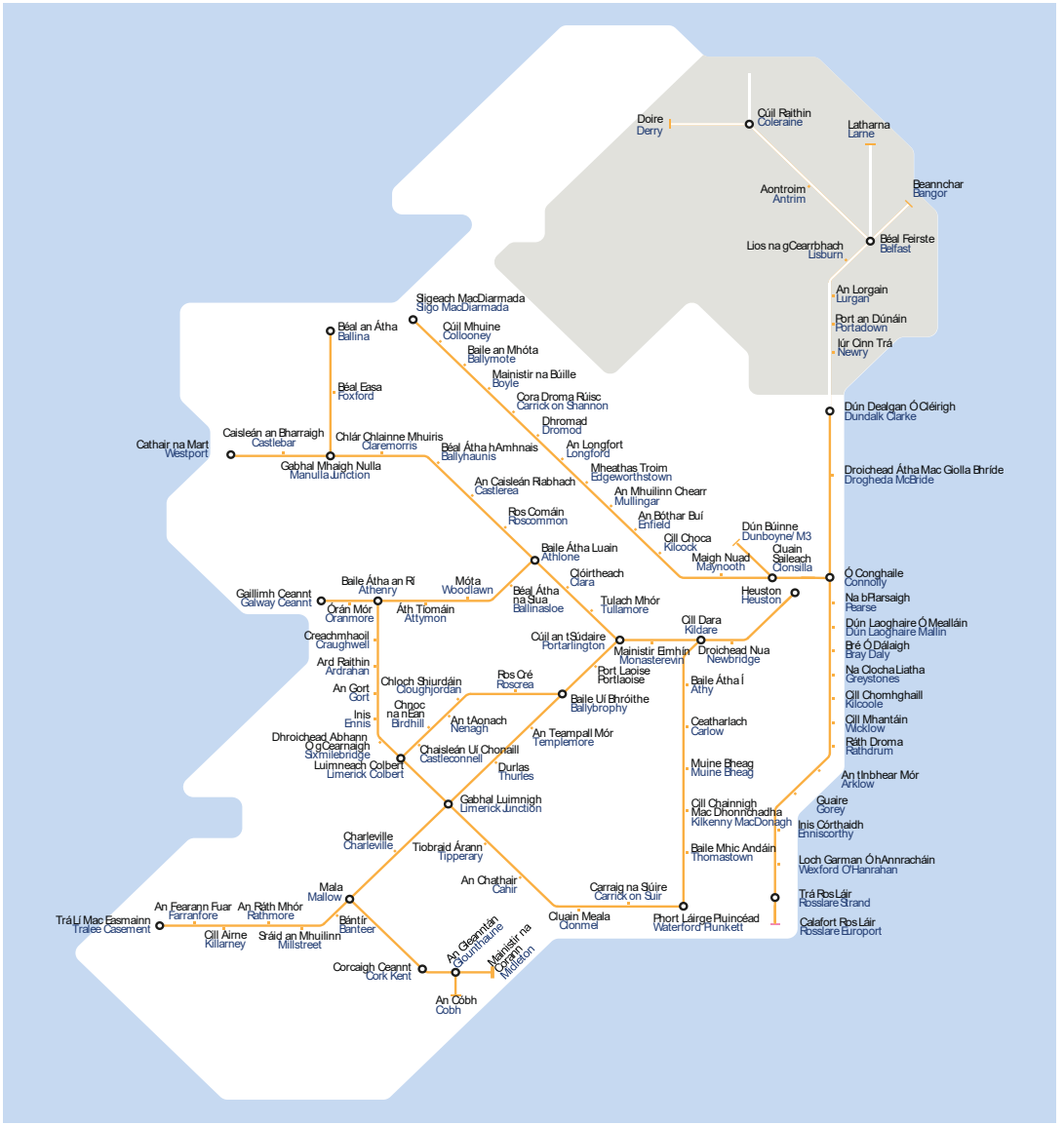
Graph Representations



Graph Database Definition

GRAPHS AS MODELS

- Networks:
  - Transportation
  - Roadmaps
  - Computers
  - Electrical
  - …



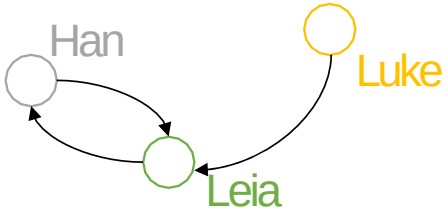
GRAPH

- A graph is a formalism for representing relationships among items. One way to represent graphs:

- A graph **G = (V,E)**
  - A set of vertices, also known as nodes

    $V = \{v_1, v_2, \ldots, v_n\}$
  - A set of edges

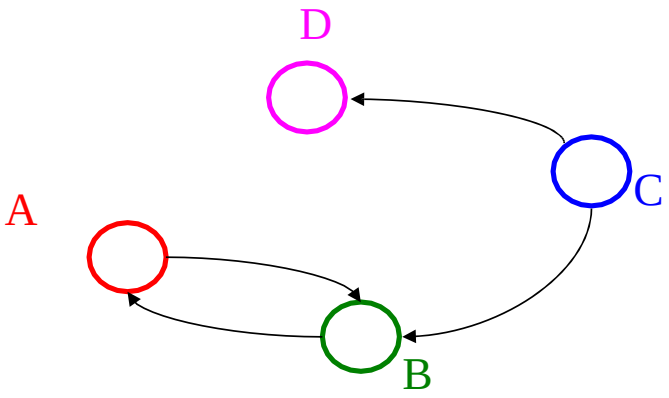
    $E = \{e_1, e_2, \ldots, e_m\}$
    - Each edge **e_i** is a pair of vertices **(v_j, v_k)**
    - An edge "connects" the vertices
    - It can also be represented as **$v_j v_k$**




```
V = {Han,Leia,Luke}
E = {(Luke,Leia),
     (Han,Leia),
     (Leia,Han)}
```

GRAPH EXAMPLE

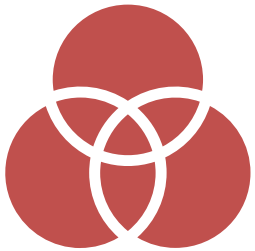
- How are the sets V and E for this graph?



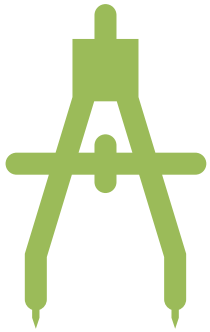

```
V = {A, B, C, D}

E = {(C, B),
     (A, B),
     (B, A),
     (C, D)}
```

REPRESENTATIONS



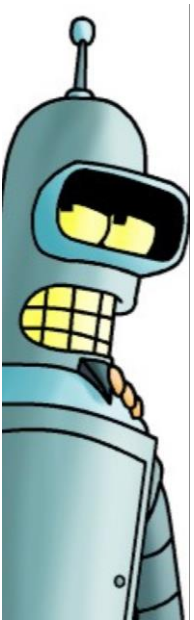
Incidence Matrix:
Depicts the incidents of an edge with a vertex

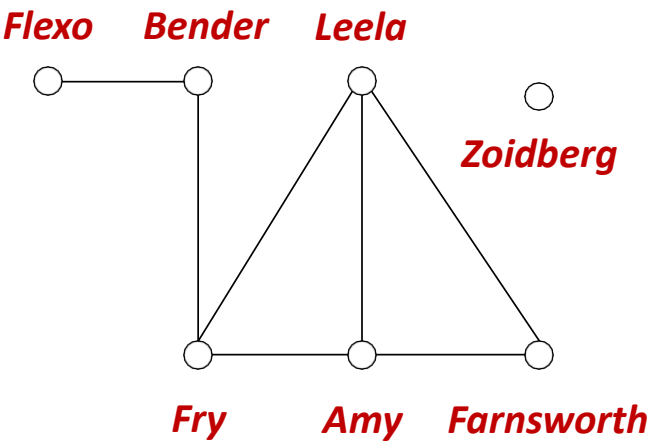


Adjacency Matrix/List:
Depicts the connections between two vertices

Social Networks

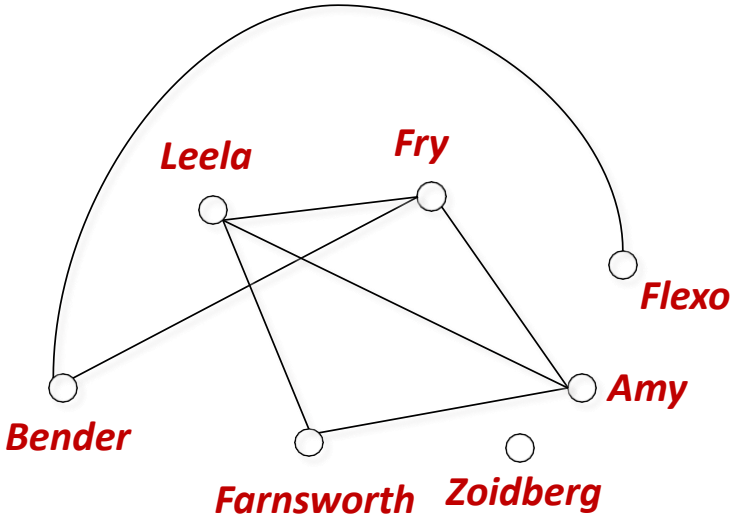
- Modelling Social Networks with graphs, the vertices just happen to have people's names
- Such a graph could represent friendships (or any kind of relationship)

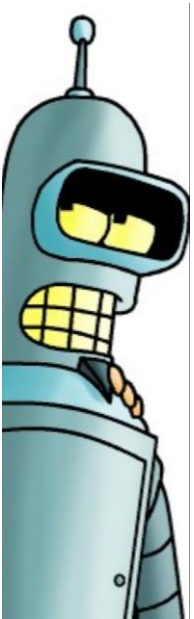




Social Networks

- Now check out the graph below.
- What can we say about it in comparison to the previous figure?





Moral of the Story

One graph may be drawn in (infinitely) many ways, but it always provides us with the same information

Graphs are a structure for describing relationships between objects

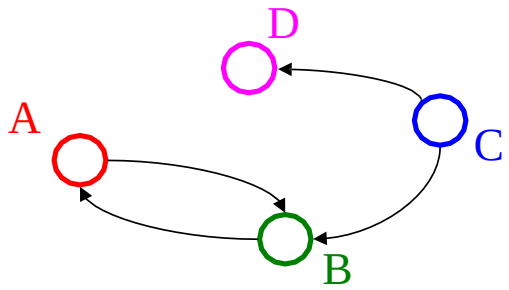
The vertices denote the objects and the edges represent the relationship

Directed Graphs: Degree

In-degree of a vertex: number of in-bound edges, i.e., edges where the vertex is the destination

Out-degree of a vertex: number of out-bound edges, i.e., edges where the vertex is the source

Degree of a vertex: Sum of in-degree and out-degree

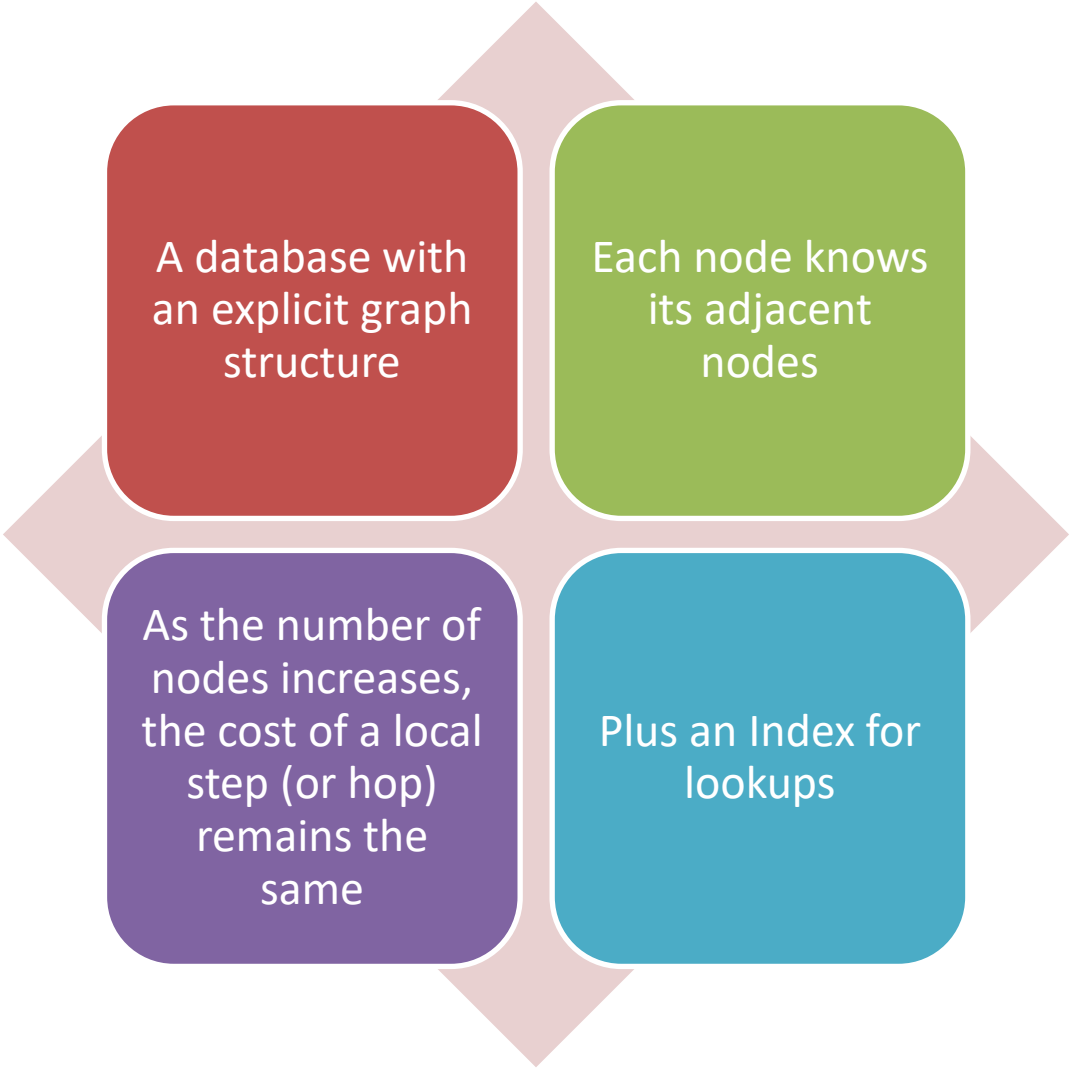


Number of Edges

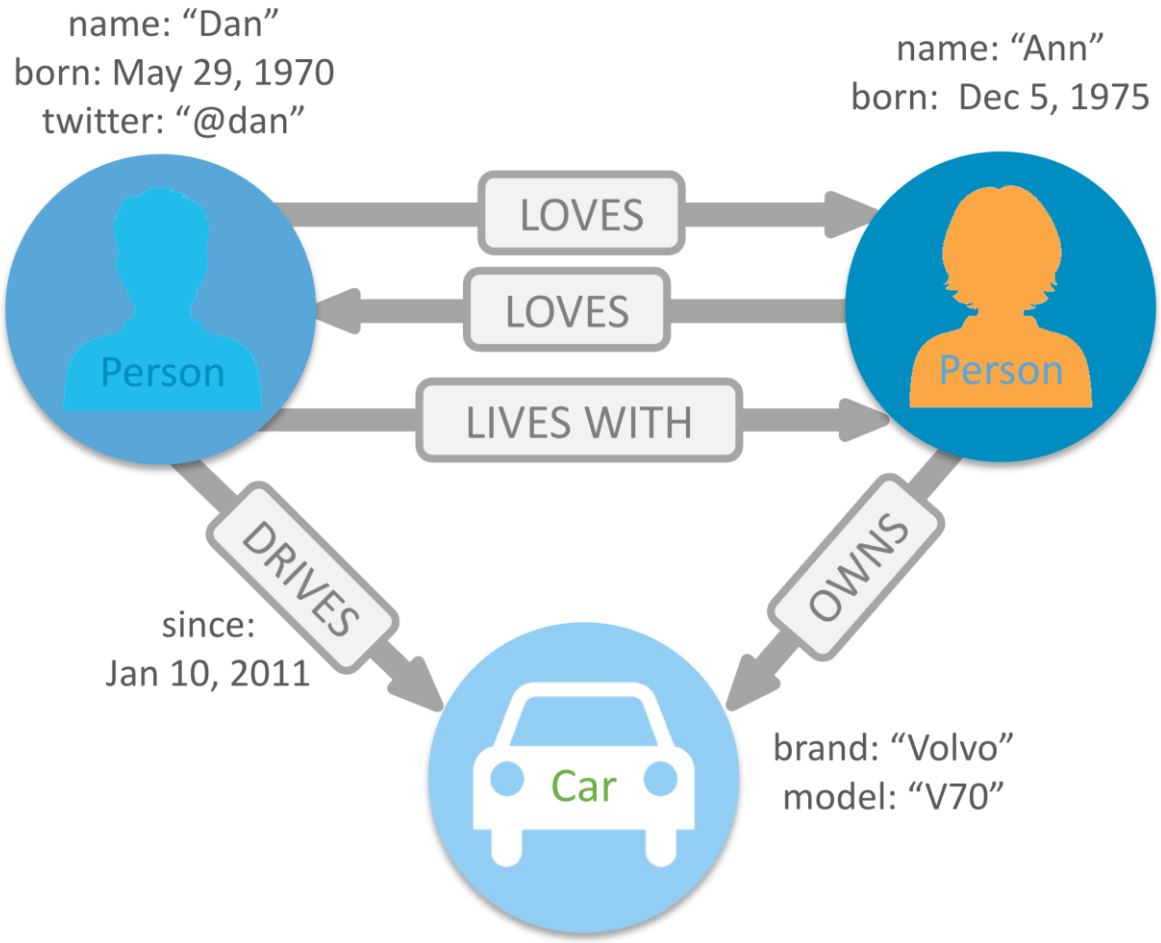
For a graph **G = (V,E)**

- **|V|** is the number of vertices
- **|E|** is the number of edges (assuming no self-loops)
  - Minimum? 0
  - Maximum for directed? $|V|*(|V| - 1) \in O(|V|^2)$
  - Maximum for undirected? $(|V|*(|V| - 1)) / 2 \in O(|V|^2)$
  - A node can have a degree / in-degree / out-degree of zero!

What is a Graph Database?



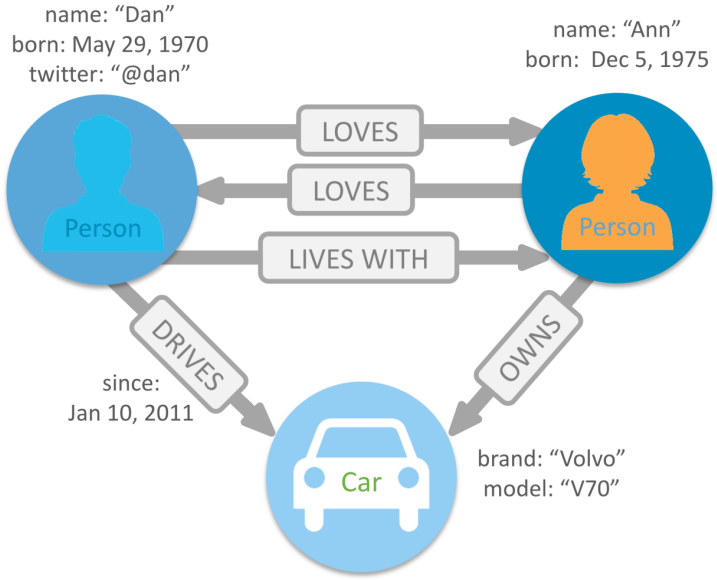
GRAPH DATA



From Neo4J documentation

GRAPH DATA

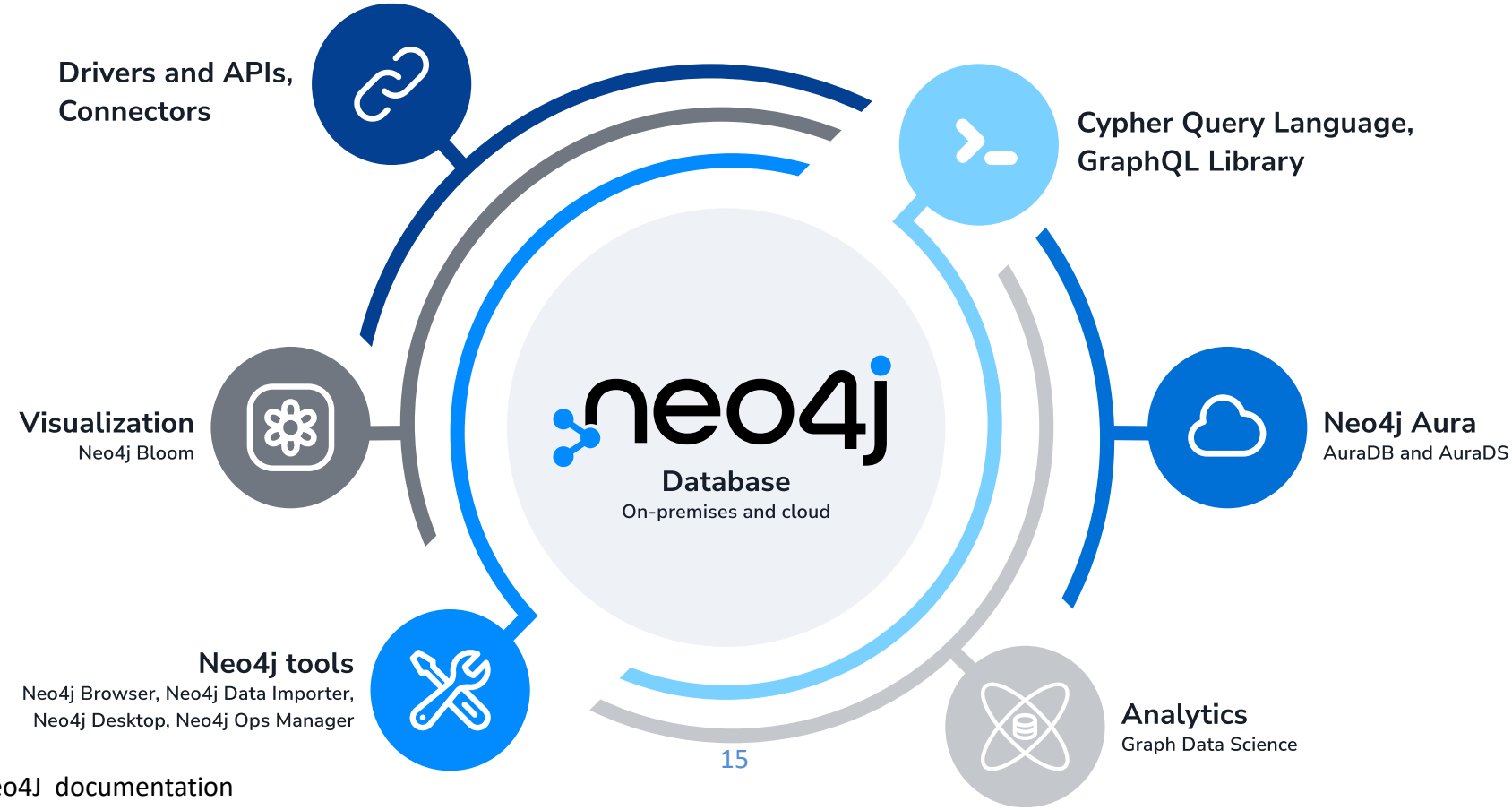
- **Nodes** describe entities (discrete objects) of a domain. They can have zero or more labels to define (classify) what kind of nodes they are.
- **Relationships** describe a connection between a source node and a target node. They are directional and always have a direction (one direction). They also have one **type** to describe the type of relationship they are.
- Nodes and relationships can have **properties** (key-value pairs), which further describe them.



What is Neo4j?

CS2209 Information Storage and Management II

- A Graph Database
- Property Graph
- *Full ACID: (atomicity, consistency, isolation, durability)*
- 32 Billion Nodes, 32 Billion Relationships, 64 Billion Properties
- Embedded Server
- REST API



From Neo4J documentation

NEO4J – TYPICAL USAGES

Highly connected data (social networks)

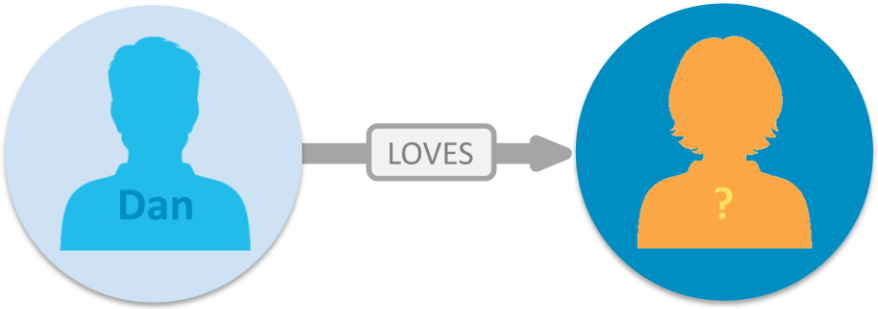
Recommendations (e-commerce)

Path Finding

A* (Least Cost path)

Data First Schema (bottom-up, but you still need to design)

CYPHER QUERY



NODE Relationship NODE

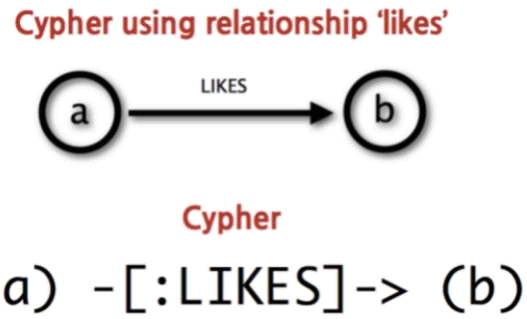
```
MATCH (:Person { name:"Dan"} ) -[:LOVES]-> ( whom ) RETURN whom
```

LABEL PROPERTY VARIABLE

From Neo4J documentation

The Structure of a Cypher Query

- Nodes are surrounded with parentheses which look like circles, e.g. (a)
- A relationship is basically an arrow --> between two nodes with additional information placed in square brackets inside of the arrow



- A query is comprised of several distinct clauses, like:
  - MATCH: The graph pattern to match. This is the most common way to get data from the graph
  - WHERE: Not a clause in its own right, but rather part of MATCH, OPTIONAL MATCH and WITH. Adds constraints to a pattern or filters the intermediate result passing through WITH
  - RETURN: What to return

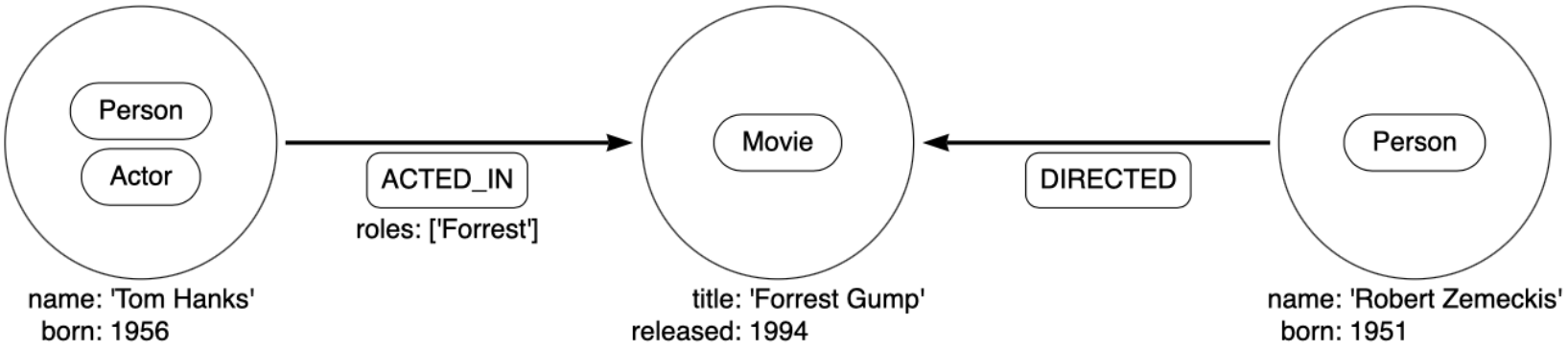
```
MATCH (john {name: 'John'})-[:friend]->()-[:friend]->(fof) RETURN john.name, fof.name
```

WRITING CYPHER QUERIES

- Node labels, relationship types and property names are case-sensitive in Cypher

CREATE	Creates nodes with labels and properties or more complex structures
MERGE	Matches existing or creates new nodes and patterns. This is especially useful together with uniqueness constraints.
DELETE	Deletes nodes, relationships, or paths. Nodes can only be deleted when they have no other relationships still existing
DETACH DELETE	Deletes nodes and all their relationships
SET	Sets values to properties and add labels on nodes
REMOVE	Removes properties and labels on nodes
ORDER BY	A sub-clause that specifies that the output should be sorted and how

GRAPH DATABASE - EXAMPLE




```
CREATE (:Person:Actor {name: 'Tom Hanks', born: 1956})-
[:ACTED_IN {roles: ['Forrest']}]->(:Movie {title:
'Forrest Gump'})<-[:DIRECTED]-(:Person {name: 'Robert
Zemeckis', born: 1951})
```

SUMMARY

- Graph Data
- Neo4J
- Cypher Query