

# What is a graph database?

NOSQL CONCEPTS



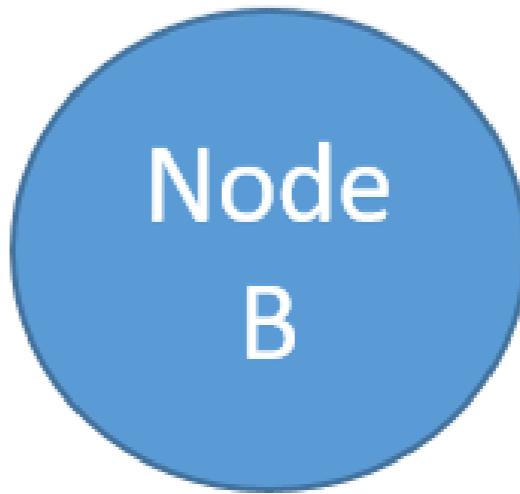
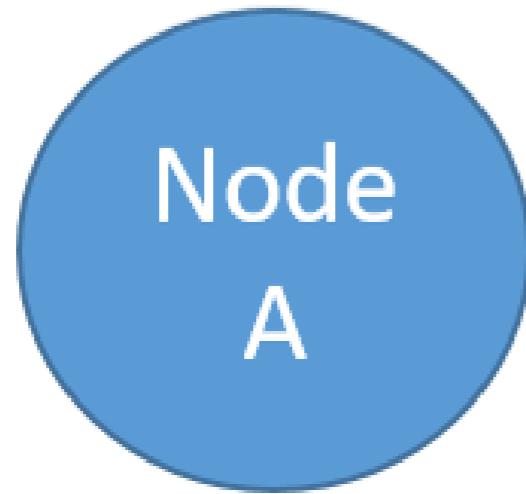
**Miriam Antona**  
Software engineer

# Graph databases - overview

- Treat **data** and its **relationships** with the same importance
- Based on **graph theory**
  - branch of mathematics
  - studies graphs for modeling the relationships between objects

# Graph databases - parts of a graph

- Vertices/nodes
- Edges



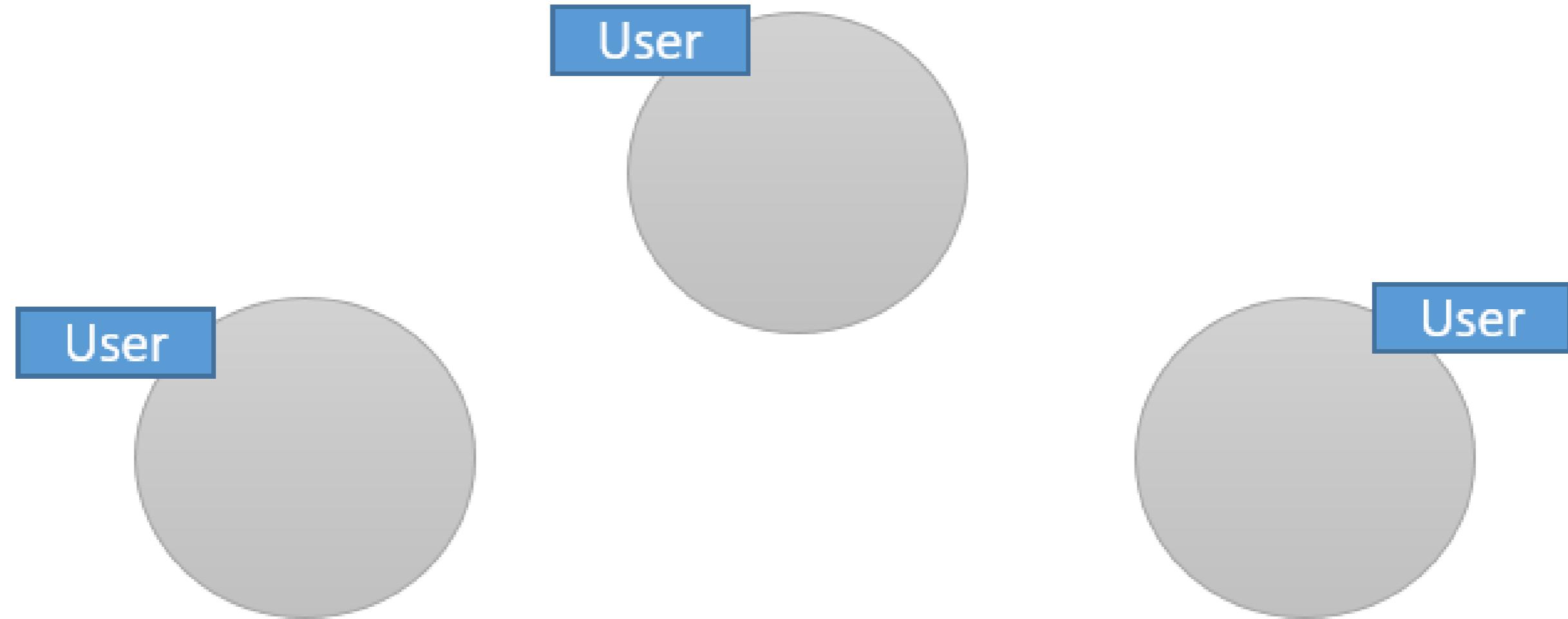
# Graph databases - parts of a graph

- Vertices/nodes
- Edges/links/arcs



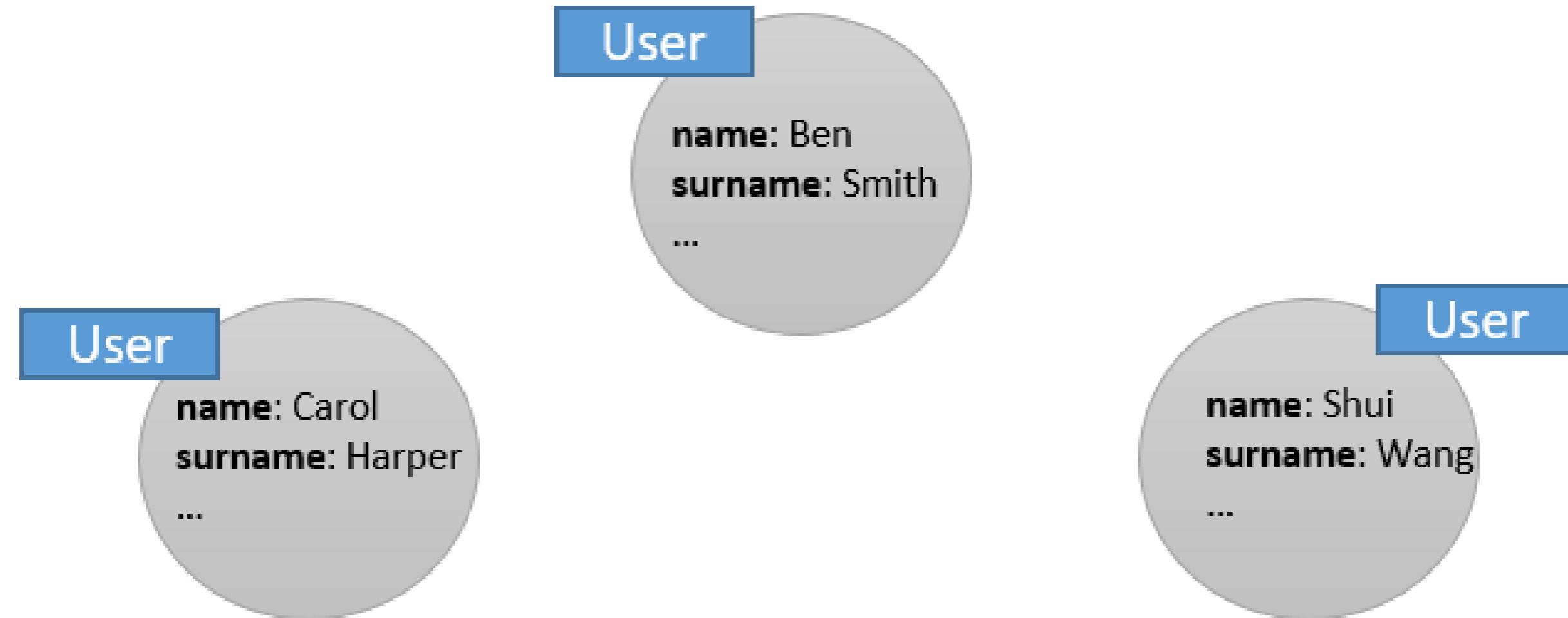
# Graph databases - nodes

- Represent **entities** (users, cities, airports, employees...)
- Have **properties**



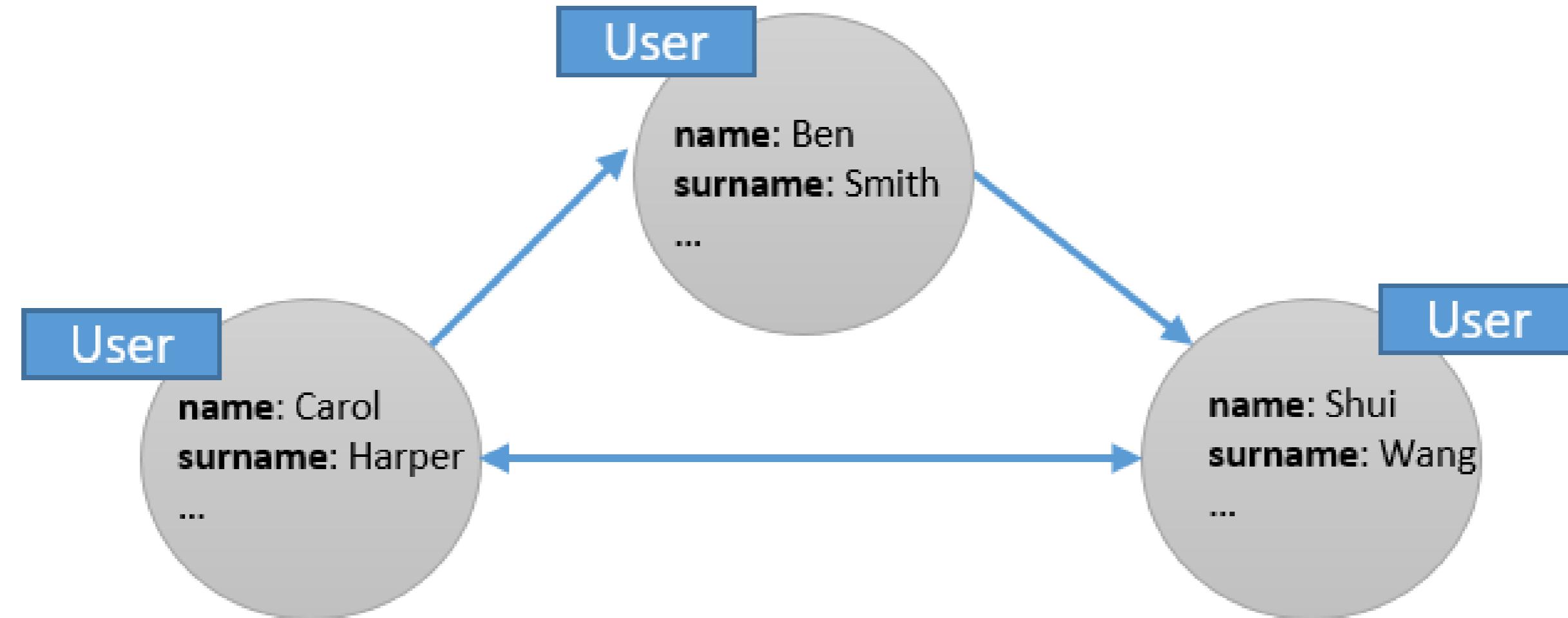
# Graph databases - nodes

- Represent **entities** (users, cities, airports, employees...)
- Have **properties**



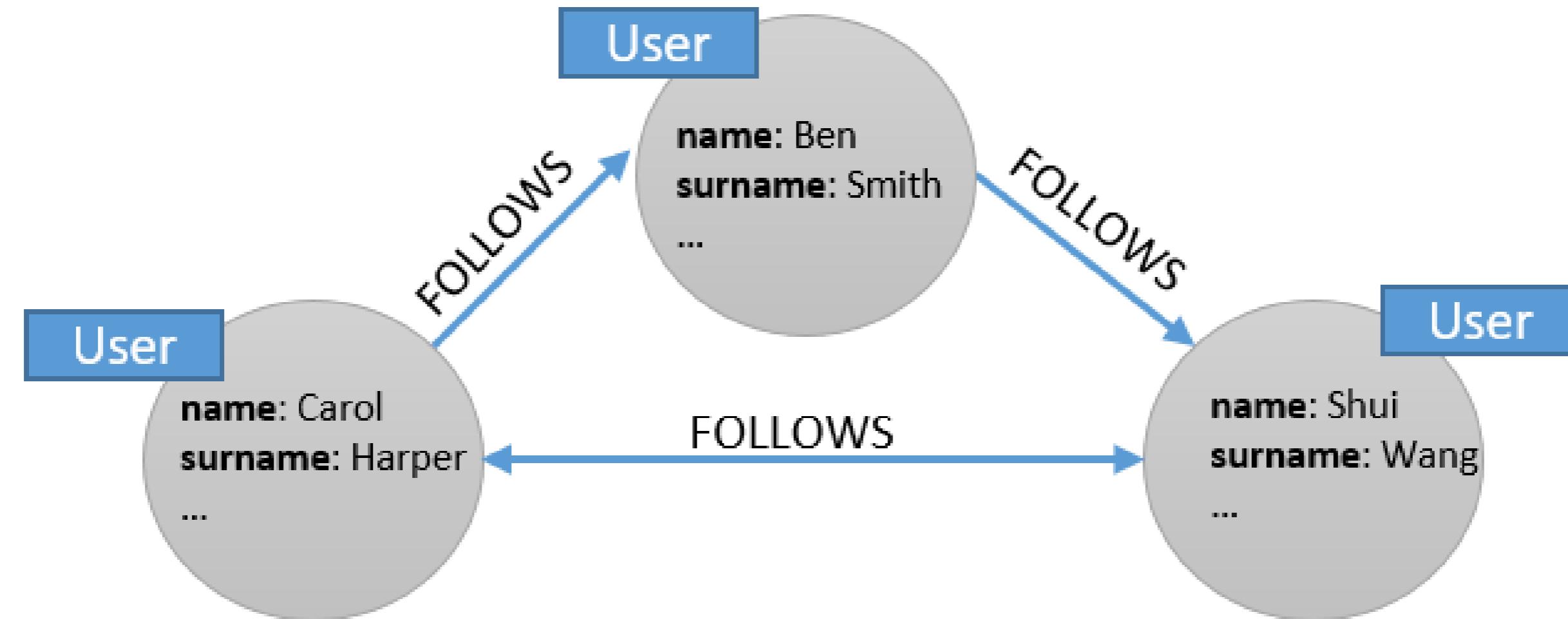
# Graph databases - edges

- Connect the nodes
- Define the **relationships** between the nodes



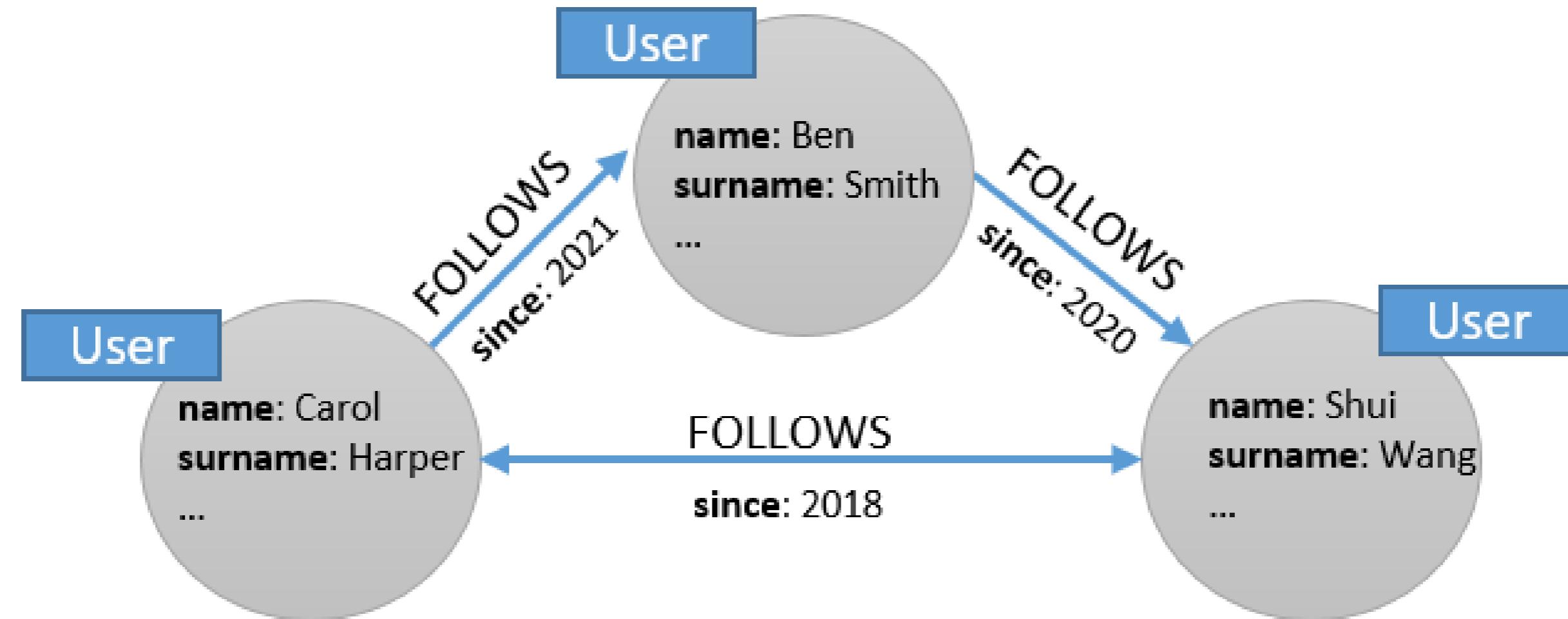
# Graph databases - edges

- Connect the nodes
- Define the **relationships** between the nodes



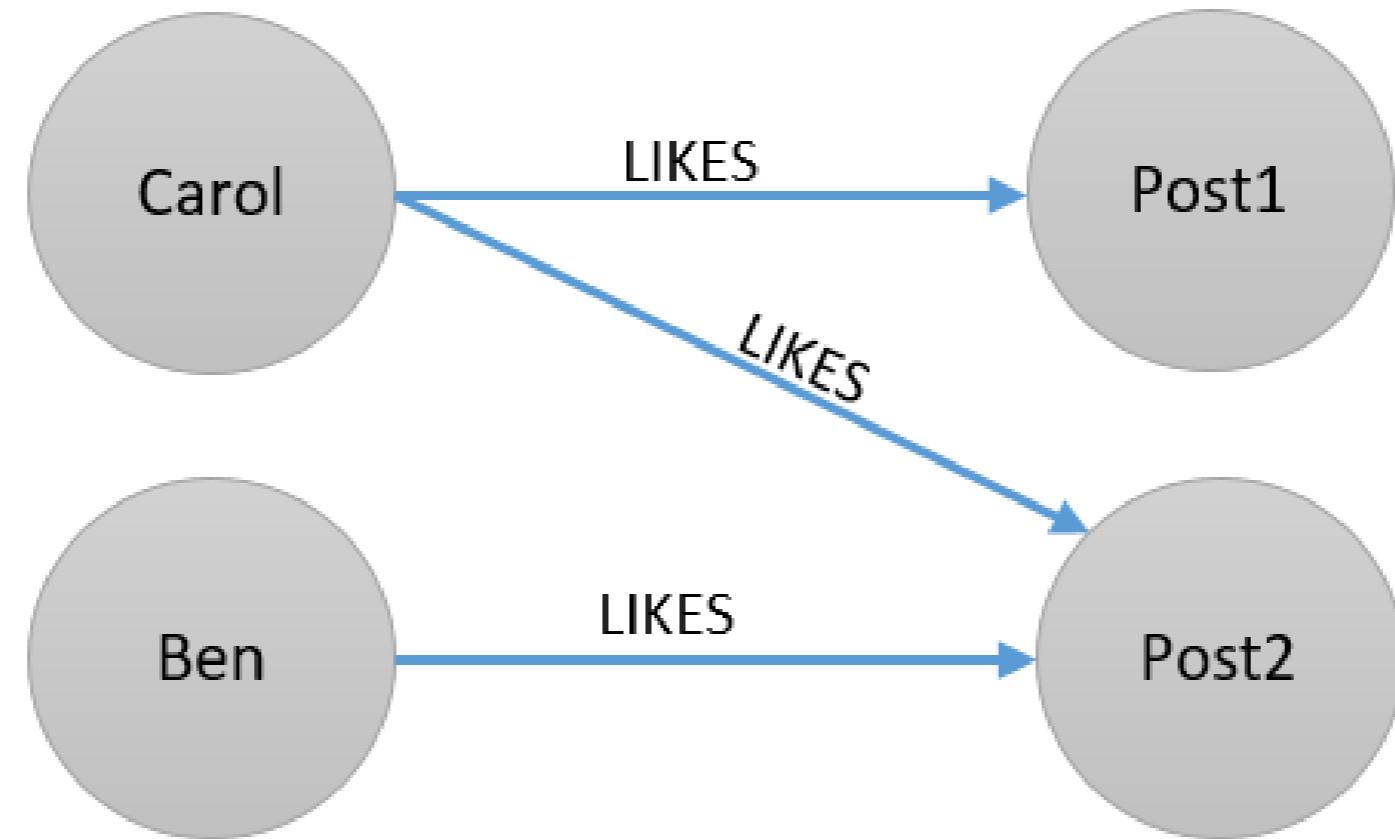
# Graph databases - edges

- Connect the nodes
- Define the **relationships** between the nodes



# Graph databases - types of edges

- Directed:
  - Specific direction



# Graph databases - types of edges

- Undirected:
  - No direction
  - The relationship is mutual



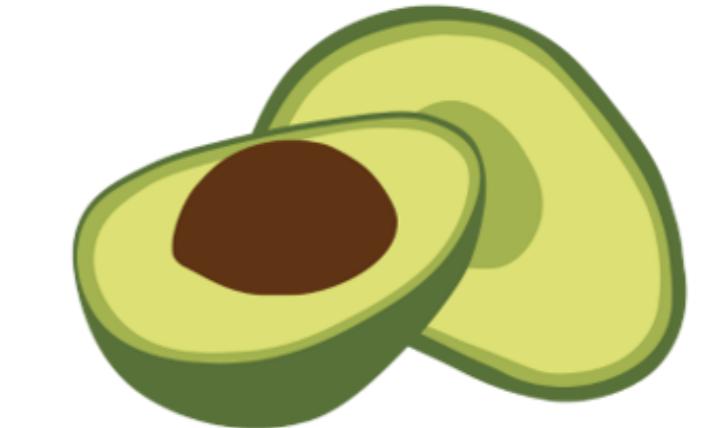
# Graph databases - queries

- **Traversing** the graph
- Examples:
  - Get all the users that Ben follows
  - Get when Carol started following Shui
  - Get the shortest path from one city to another
- **Path:** set of nodes and edges across a graph
- **Query languages:** Cypher, Gremlin, etc.

# Popular graph databases



Azure Cosmos DB



ArangoDB

# **Let's practice!**

**NOSQL CONCEPTS**

# Advantages and limitations of graph databases

NOSQL CONCEPTS



**Miriam Antona**  
Software engineer

# Advantages - flexibility

- Can **change** as applications and industries change
- **Don't need** to define the final **final structure** in advance
- Can **add/delete nodes, properties, and edges**

# Advantages - performance

- Doesn't need to perform **joins**
  - Joining can be potentially **time-consuming**
- Follow edges from node to node
  - **Simpler and faster**

# Advantages - easy representation of the data

- **Similar** structure to **human thinking**
  - Graph modeling very **intuitive**
- **Easily visualized**
- **Facilitates understanding**

# Advantages - horizontal scalability

- It is **possible**
- **More difficult** than in other NoSQL databases
  - Graphs are connected
  - Need to be distributed across multiple machines

# Limitations

- Entity properties with **extremely large values**
  - BLOBs (Binary Large Objects): multimedia objects
  - CLOBs (Character Large Objects): collections of character data
  - Graph databases **won't perform well**
  - **Bad practice**
  - Use **another database** to store that information
- **Significant change** for developers
  - New data modeling mindset
  - Learn Cypher, Gremlin...

# **Let's practice!**

**NOSQL CONCEPTS**

# When to use graph databases

NOSQL CONCEPTS



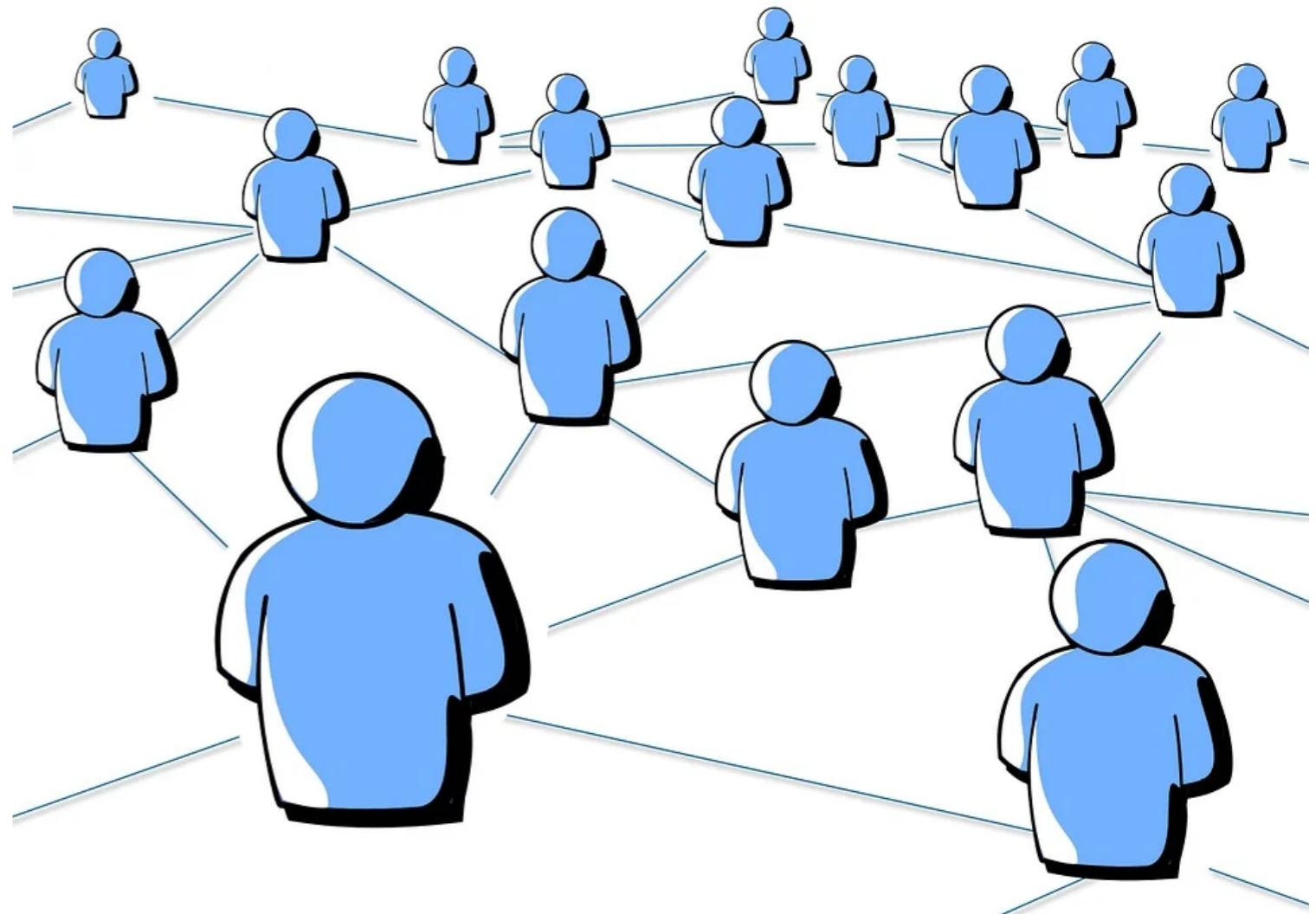
**Miriam Antona**  
Software engineer

# Suitable cases - General information

- Highly **connected** data

# Suitable cases - Social graphs

- **Social networks**
  - Users and their relationships
  - Posts
  - Likes
  - ...
- **Employees of a company**
  - Projects
  - Relationships between the employees



# Suitable cases - Infectious diseases

- Model:
  - Interactions between people
  - Contact events
  - Exposures
- Help to understand **transmission chains**



# Suitable cases - Location services

- Locations and distances
- Optimize the routes for navigation applications
- Recommendations for nearby points of interest:
  - restaurants
  - cinemas
  - hospitals
  - etc.



# Suitable cases - Fraud detection

- **Stop fraud in real-time:**
  - Credit card fraud
  - E-commerce fraud
  - Money laundering
- **Model:** individuals, credit cards, phone numbers, devices, IP addresses...
- **Uncover suspicious patterns:**
  - multiple users coming from the same IP
  - ...



# Suitable cases - Real-time recommendations

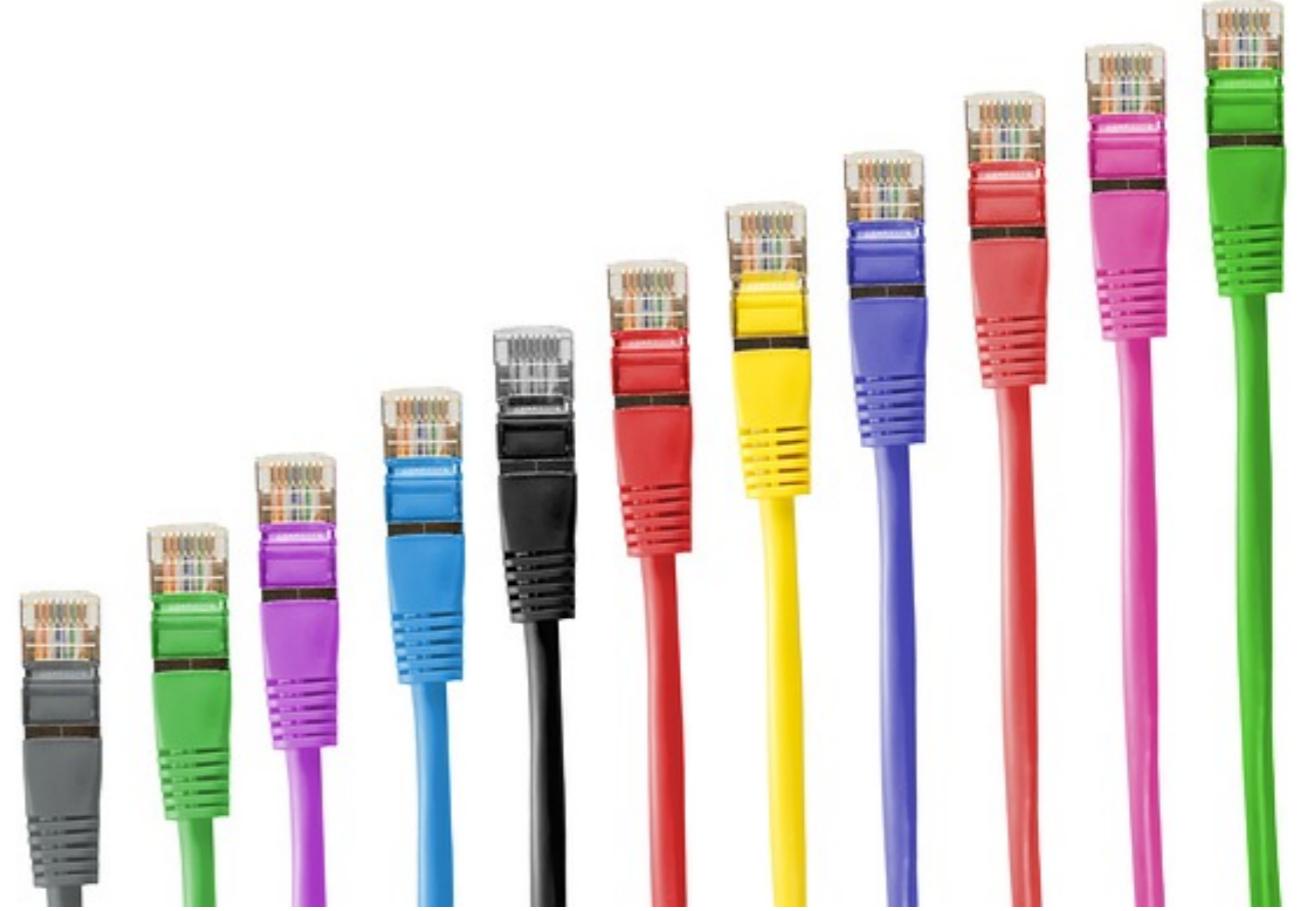
- Store buying history and browsing behavior to **recommend an item**
- Offer an item to a friend
- Show unwatched movies based on similar users

You may also like



# Suitable cases - Networks

- Storage of relationships between **networks** and **infrastructure** elements
  - physical machines
  - virtual machines
  - applications
  - routers
  - switches
  - ...
- Alert in real-time about potential **design flaws**



# Unsuitable cases

- **Disconnected data**
- **Relationships** between the data are **not important**
- Applications that only perform **general searches** without a specific starting point
  - Are not optimized for those queries
- **Properties** that contain extremely **large values** (BLOBs, CLOBs...)

# **Let's practice!**

**NOSQL CONCEPTS**

# Neo4j case study

## NOSQL CONCEPTS



**Miriam Antona**  
Software engineer

# Neo4j - overview

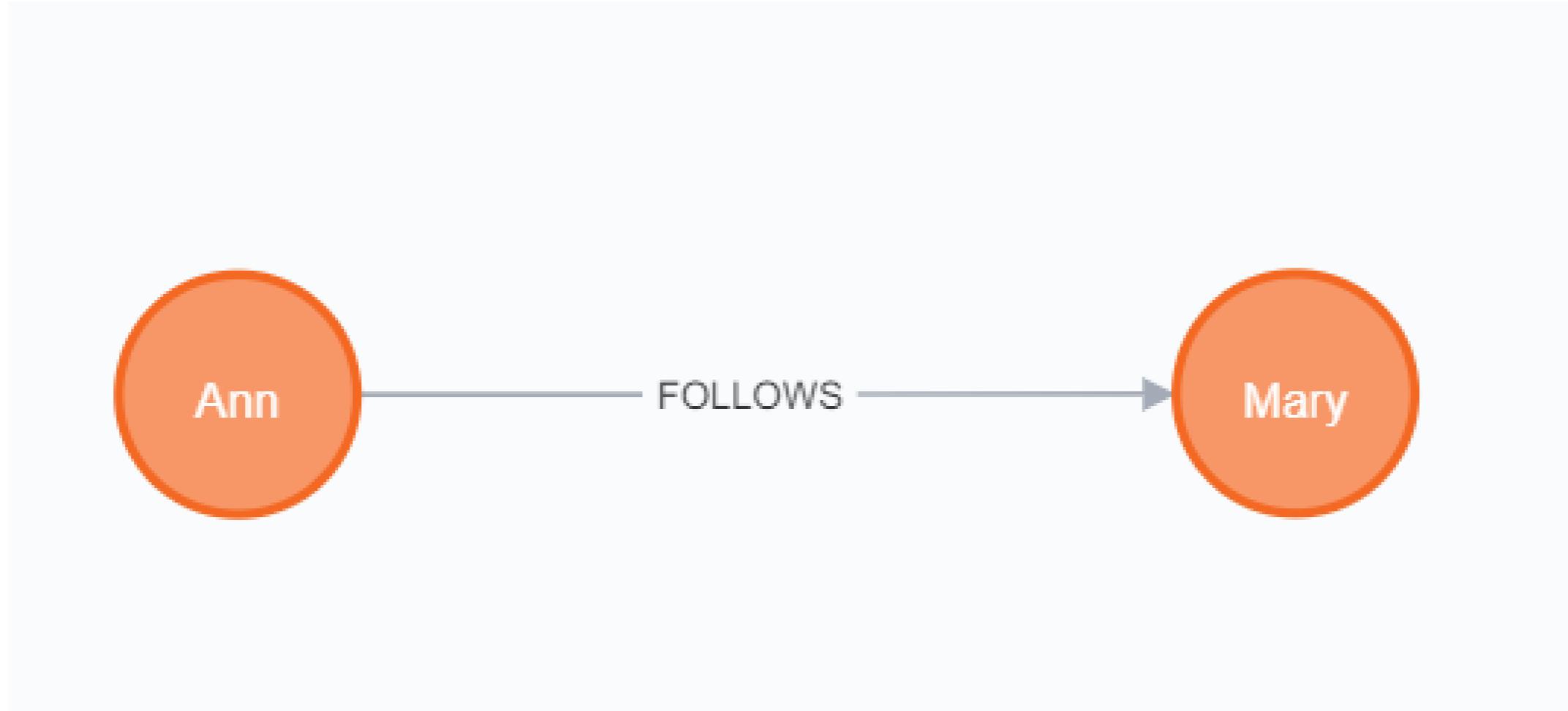
- Leader in **graph database** technology
- 420+ employees
- Reveals and predicts how people, processes, and systems are **interrelated**



# Neo4j graph platform

- Suite of applications and tools
  - Interact and use graph data

# Neo4j graph platform - Graph database

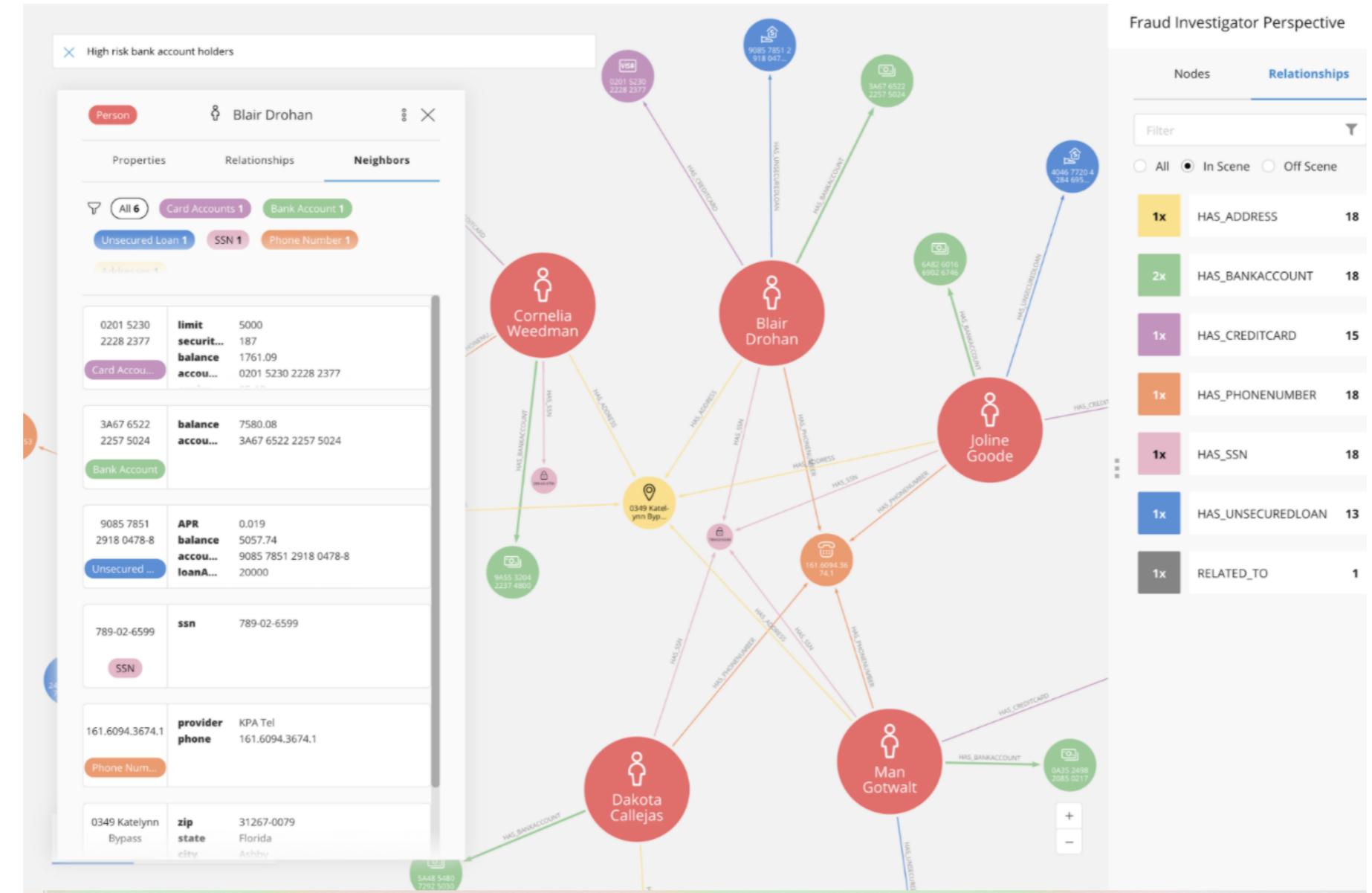


- **Store and connect the data**

# Neo4j graph platform - Graph data science library

- Used for executing **graph algorithms**:
  - Path finding
  - Centrality
  - Community detection
  - Similarity
  - Link prediction
  - Node embeddings
  - Node classification

# Neo4j graph platform - Bloom



- View and analyze data without any programming knowledge

# Neo4j graph platform - Cypher

- Powerful **query language**
- Inspired by SQL
- **Save and get** data from the graph database
- Multiple queries within the **same transaction**

```
CREATE (:User {name: 'Ann'})
```

# Neo4j graph platform - Connectors and Integrations

- Facilitate the use of Neo4j in existing architectures
  - Neo4j Connector for Apache Spark
  - Neo4j Connector for Apache Kafka
  - Neo4j Connector for BI
  - Neo4j Labs Integrations

# Neo4j graph platform - Developer tools

- Neo4j Desktop
  - Local Neo4j databases
- Neo4j Browser
  - Interact with the graphs using a browser
- Neo4j Sandbox
  - Learn about Neo4j, test personal ideas...

# Neo4j graph platform - Aura

- Cloud Database Service
- Run in the cloud without managing the infrastructure



# Neo4j - drivers

- **Official drivers:** C#, Java, Python, JavaScript, Spring...
- **Community drivers:** R, PHP, Ruby...

# Neo4j - popular uses

- Analytics and artificial intelligence
- Fraud detection
- Real-time recommendations
- Knowledge graphs
- Life sciences
- Telecommunications
- ...

# Neo4j - customers



Vanguard®



Adobe



v o l v o



# Gousto - overview

- Recipe box company
- Delivers over a million meals each month
- 400+ staff



# Gousto - problem and solution

- **Problem:** More choice implied difficulty to navigate
- **Solution:** Use Neo4j
  - Internal **recommendation system** with recipes
    - subscriber's previous interactions with the menu
    - information on upcoming recipes

# Gousto - results

- 30% increase in the number of customers that select recommended recipes
- Better cost control

# Gousto - results

- 30% increase in the number of customers that select recommended recipes
- Better cost control

<sup>1</sup> <https://neo4j.com/case-studies/>

# **Let's practice!**

**NOSQL CONCEPTS**

# Congratulations!

## NOSQL CONCEPTS



**Miriam Antona**  
Software engineer

# Chapter 1

- Differences between **NoSQL** and **relational databases**
- Main concepts of **key-value** databases
- **Advantages**
  - simplicity
  - horizontal scalability
  - flexibility
- **Limitations**
  - no complex queries
  - search just by key

# Chapter 1

- **Suitable** cases
  - user-session information
  - user profiles and user preferences
  - shopping carts
  - ...
- **Unsuitable** cases
  - search a key based on its value
- **Case study:** Editoo and Redis
  - reduction in downtime
  - higher performance

# Chapter 2

- Basics of **document** databases
- **Advantages**
  - flexibility
  - horizontal scalability
  - intuitive for developers
- **Limitations**
  - care about the code
  - care about redundant data

# Chapter 2

- **Suitable cases**
  - catalogs
  - event logging
  - user profiles
  - etc.
- **Unsuitable cases**
  - very structured data
- **Case study:** Shutterfly and MongoDB
  - improved its performance

# Chapter 3

- Basics of **column family** databases
- **Advantages**
  - flexibility
  - speed
  - horizontal scalability
- **Limitations**
  - no multirow transactions
  - no joins
  - no subqueries

# Chapter 3

- **Suitable** scenarios
  - event logging
  - Content Management Systems
  - time-series data
- **Unsuitable** scenarios
  - need to change the queries very often
  - need complex queries and joins
  - don't deal with large amounts of data
- **Case study:** Bigmate and Apache Cassandra
  - can handle concurrent millions of operations

# Chapter 4

- Main concepts of **graph** databases
- **Advantages**
  - flexibility
  - high performance
  - horizontal scalability
  - easiness of data representation
- **Limitations**
  - extremely large objects (BLOBs, CLOBs...)
  - significant change for developers

# Chapter 4

- **Suitable** situations
  - social graphs
  - infectious diseases
  - fraud detection
  - location services
  - etc.
- **Unsuitable** situations
  - disconnected data
  - unimportant relationships between data
  - etc.

# Chapter 4

- **Case study:** Gousto and Neo4j
  - better cost control
  - increased number of customers that select recommended recipes

# **Thank you!**

**NOSQL CONCEPTS**