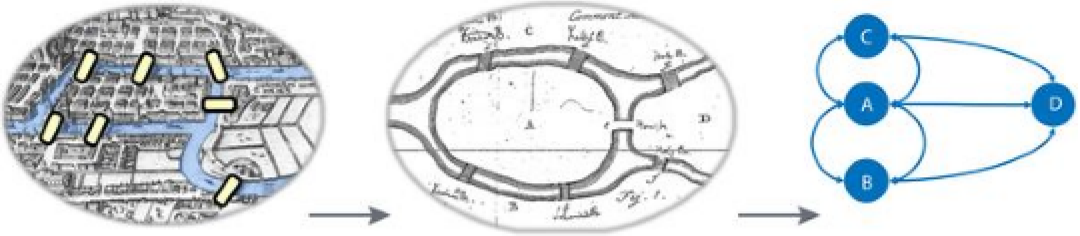
Introduction

What are graphs?

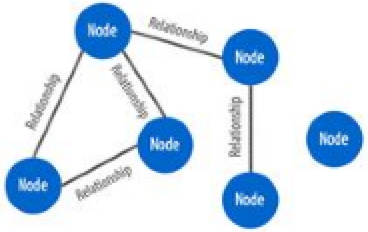
- 1736, Leonhard Euler.
- Is it possible to visit all four areas of a city connected by 7 bridges, by crossing each bridge only once?

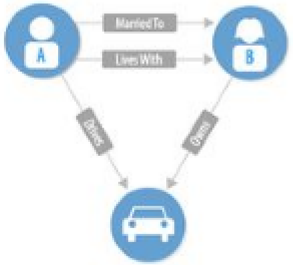


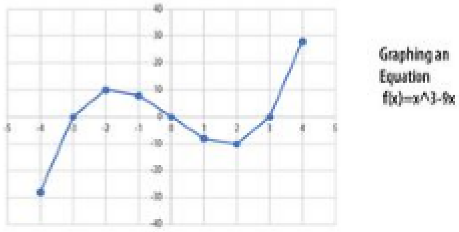
Formal definition

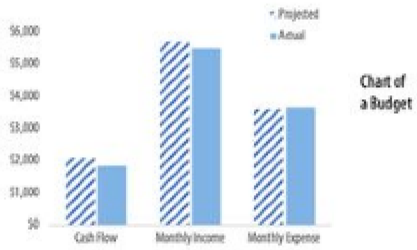
- Collection of *vertices* (*nodes*) and *edges* (*relationships*).
- Vertices represent entities of the real world, and edges the relationship between them.

Graph vs charts









Examples

- Modeling dynamic environments from financial markets to IT services.
- Forecasting the spread of epidemics as well as rippling service delays and outages.
- Finding predictive features for machine learning to combat financial crimes.
- Uncovering patterns for personalized experiences and recommendations.

The most valuable graphs

- **Facebook:** discrete information about people is important, but *relationships* among them (the *social graph*) is more.
- **Google:** Store and process discrete documents is fine, the *web graph* that encodes relationships among them is where the value is.

Graph analytics and algorithms

- Use relationships between nodes to infer the organization and dynamics of complex systems.
- **Four main families**:
  - Pathfinding and graph search.
  - Centrality.
  - Community detection.
  - Link prediction.

Local vs global properties

- **Local:** Graph queries that consider specific parts of the graph, and description of interactions in the surrounding subgraph.
- **Global:** Graph queries or processing that sheds light on the overall nature of the network.
- Some cases lie in between (e.g. transaction analysis) but had been divided due to technology limitations.

OLTP vs OLAP

- **Online transaction processing (OLTP)** operations are typically short activities like booking a ticket, crediting an account, booking a sale.
- **Online analytical processing (OLAP)** facilitates more complex queries and analysis over historical data.
- **HTAP:** Hybrid transactional and analytical processing.

Data Storage

Storing connected data

- Mechanical tapes.
- **Relational databases.**
    - Excellent option for storing tabular data.
    - Multiple data sources and their relationships (PK, FK).
    - *But* not very easy to handle relationships between entities.

Relational DBs downsides

- Heavily normalized schema leads to small join tables.
- Expensive joins are needed for some queries (purchase history).
- *Which customers bought this product?* is expensive, and *which customers buying this product also bought that product?* is even worse!

Data revolution

As more data is stored, better storage methods are needed.

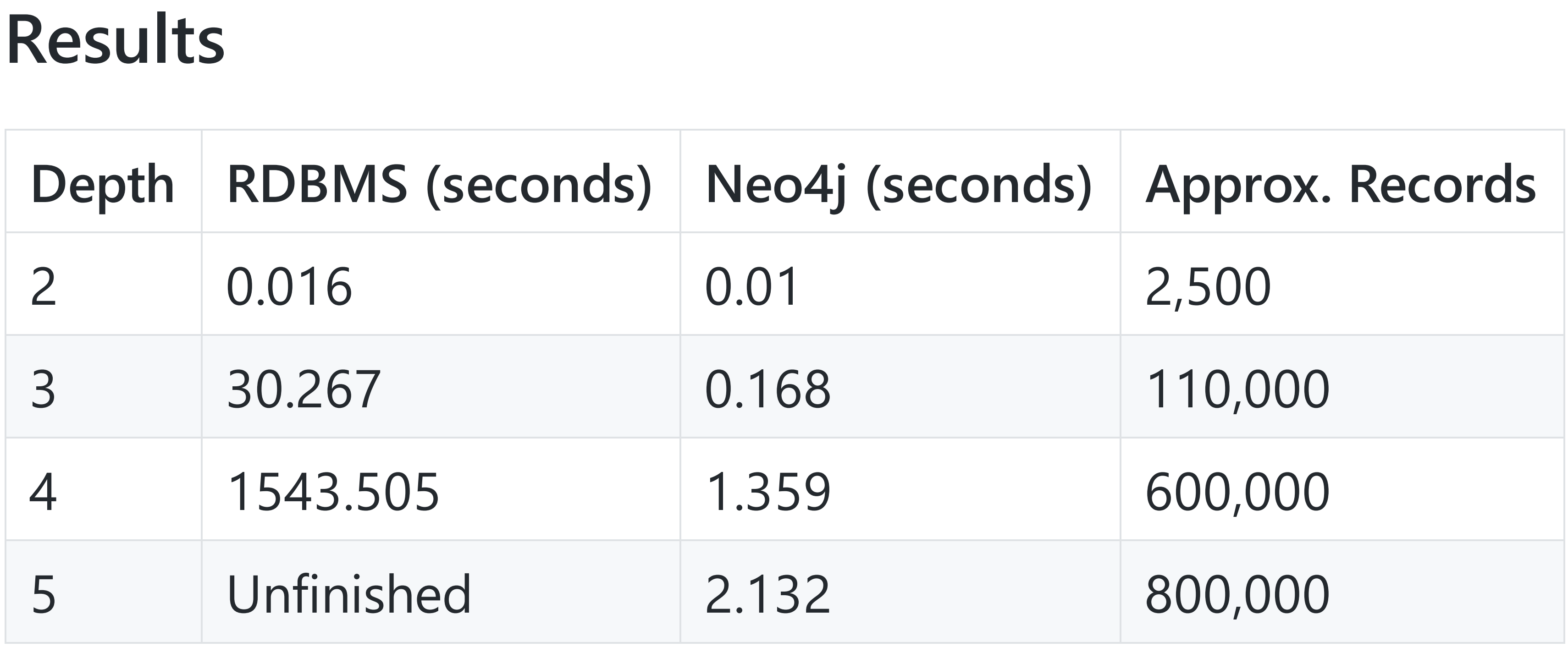
- **Performance:** improve indices.
- **Developer Experience:** document databases a partial solution.

Graph databases improve performance *and* developer experience!

Performance boost?

Partner and Vukotic's experiment:

- Social network with 1 million people.
- Each of them with around 50 friends.
- **Goal:** Find *friends of friends* on the database (up to depth 5).

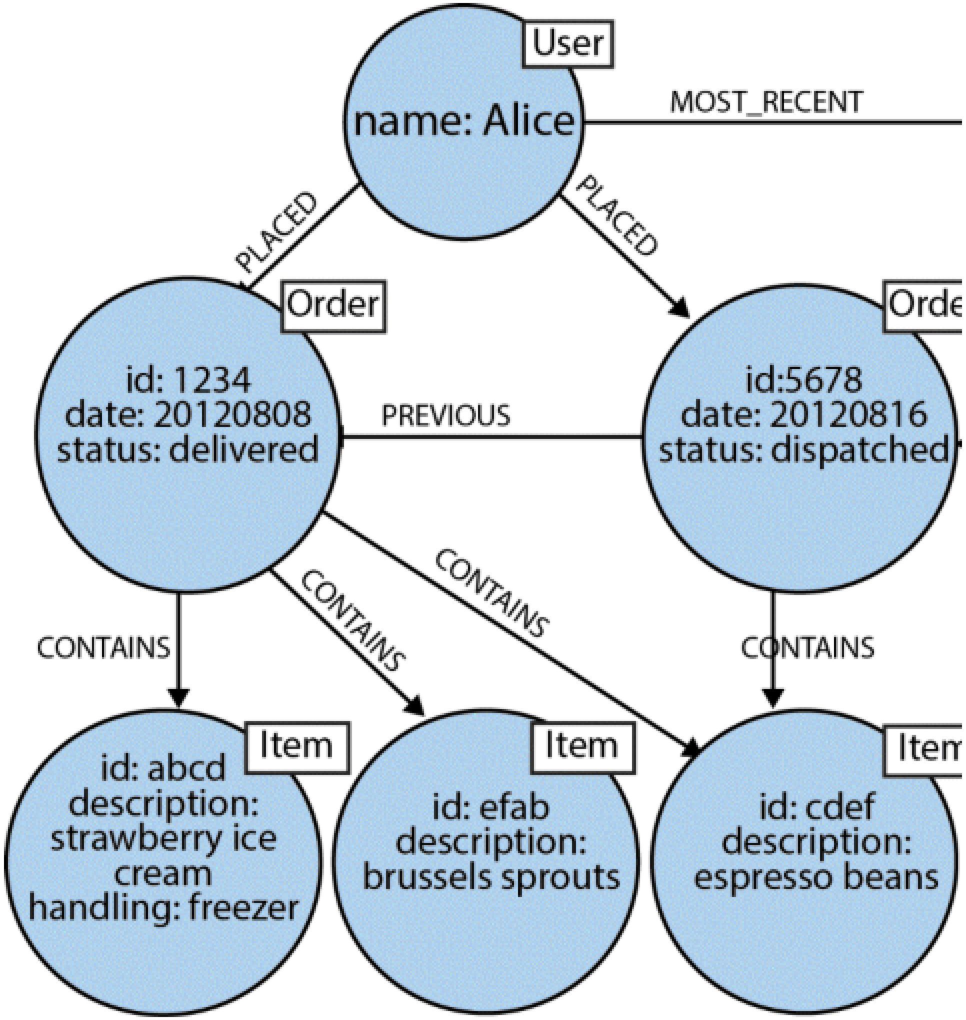


Results

Depth	RDBMS (seconds)	Neo4j (seconds)	Approx. Records
2	0.016	0.01	2,500
3	30.267	0.168	110,000
4	1543.505	1.359	600,000
5	Unfinished	2.132	800,000

Another example: online retail

- Quickly retrieve a user purchase history (without joins).



Graph platforms and processing

Apache Spark

- Support for various data science workflows.
- Great choice when:
  - algorithms are parallelizable.
  - analysis can be run offline in batch mode.
  - graph analysis is on data which is not transformed into a graph format.
  - infrequent use of graph algorithms.
  - team can (and want to) code their own algorithms.

Neo4j Graph Platform

- Tightly integrated graph database + algorithm-centric processing optimized for graphs.
- Great choice when:
  - algorithms are performance-sensitive.
  - analysis/results are integrated with transactional workloads.
  - integration with graph visualization platforms.
  - team prefers prepackaged and supported algorithms.

NetworkX

- Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
- Great choice when:
  - all the Python advantages needed: fast prototyping, multi-platform, easy to teach.
  - develop more sophisticated analysis than with Neo4j.
  - no mood to deal with Neo4j's lack of documentation.
  - local, in-memory graph toolkit.

The veredict?

- Many organizations use both Neo4j and Spark for graph processing.
- Spark can do high-level filtering and preprocessing of massive datasets.
- Neo4j can do more specific processing and integration with graph-based applications.
- NetworkX is a great tool for prototyping and doing more complicated analysis.