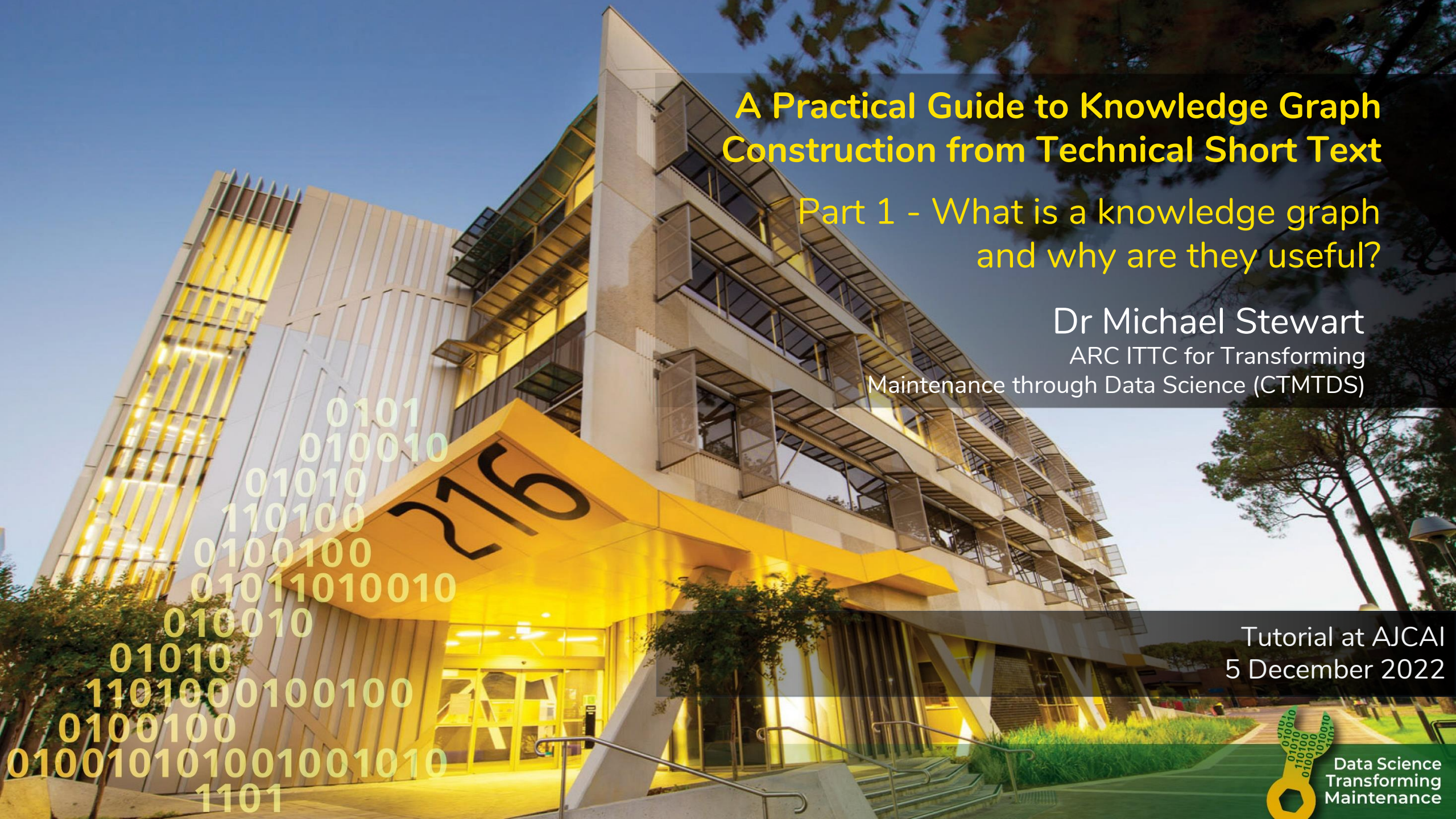


A Practical Guide to Knowledge Graph Construction from Technical Short Text

Part 1 - What is a knowledge graph and why are they useful?

Dr Michael Stewart
ARC ITTC for Transforming Maintenance through Data Science (CTMTDS)

Tutorial at AJCAI
5 December 2022

Data Science Transforming Maintenance

Outline – Part 1

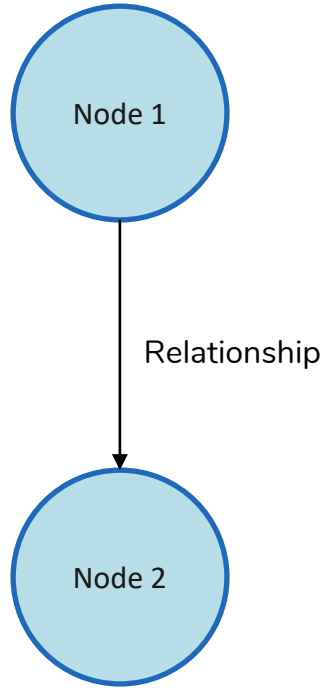
- Intro to graph databases
- Property graphs & intro to Neo4j
- Why graph databases?
- From graph databases to knowledge graphs

Outline – Part 1

- Intro to graph databases
- Property graphs & intro to Neo4j
- Why graph databases?
- From graph databases to knowledge graphs

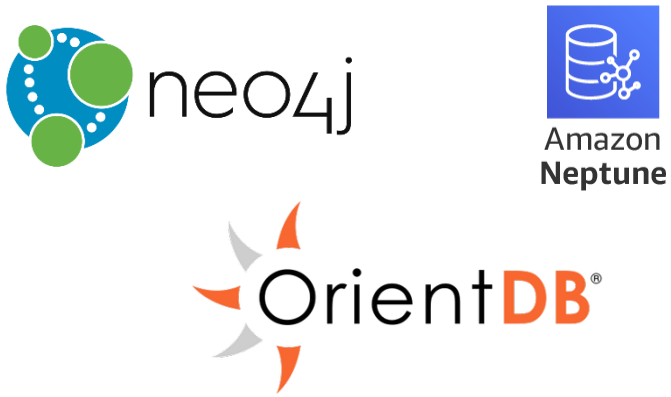
What is a graph?

- A graph is a collection of **vertices** and **edges**, also known as **nodes** and **relationships**. Edges can be **directed** or **undirected.**
- We can model all sorts of scenarios using graphs – social networks, natural language, scientific papers, etc.
- While graph databases have been around for many years, they have gained popularity in recent years thanks to the advent of **knowledge graphs**.



High-level view of the Graph Space

Graph Databases



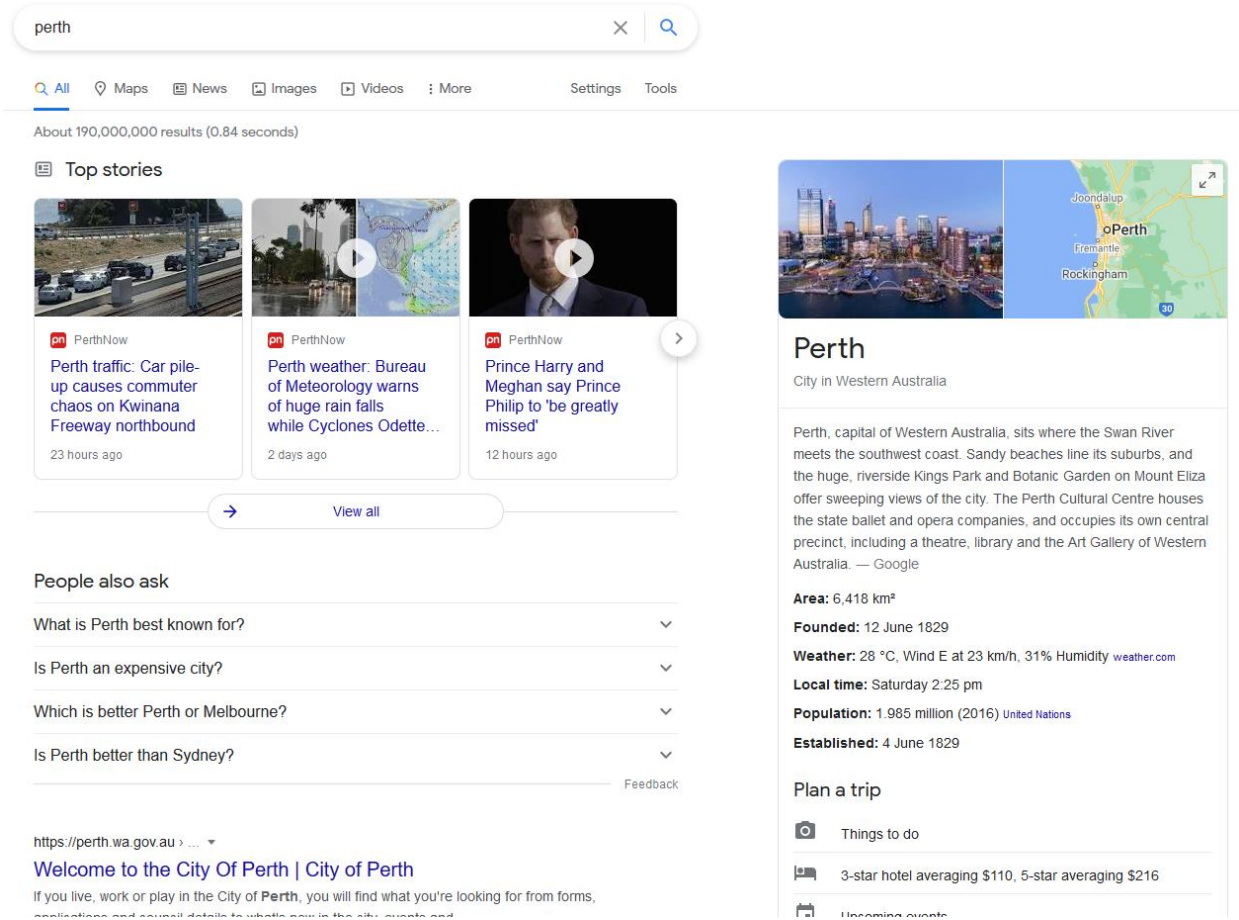
Technologies used primarily for **transactional** online graph persistence, typically accessed directly in **real time** from an application.

Graph Compute Engines

Technologies used primarily for **offline graph analytics**, performed as a series of batch steps.

Examples of graphs in use today

Google's search functionality is made possible via a knowledge graph.



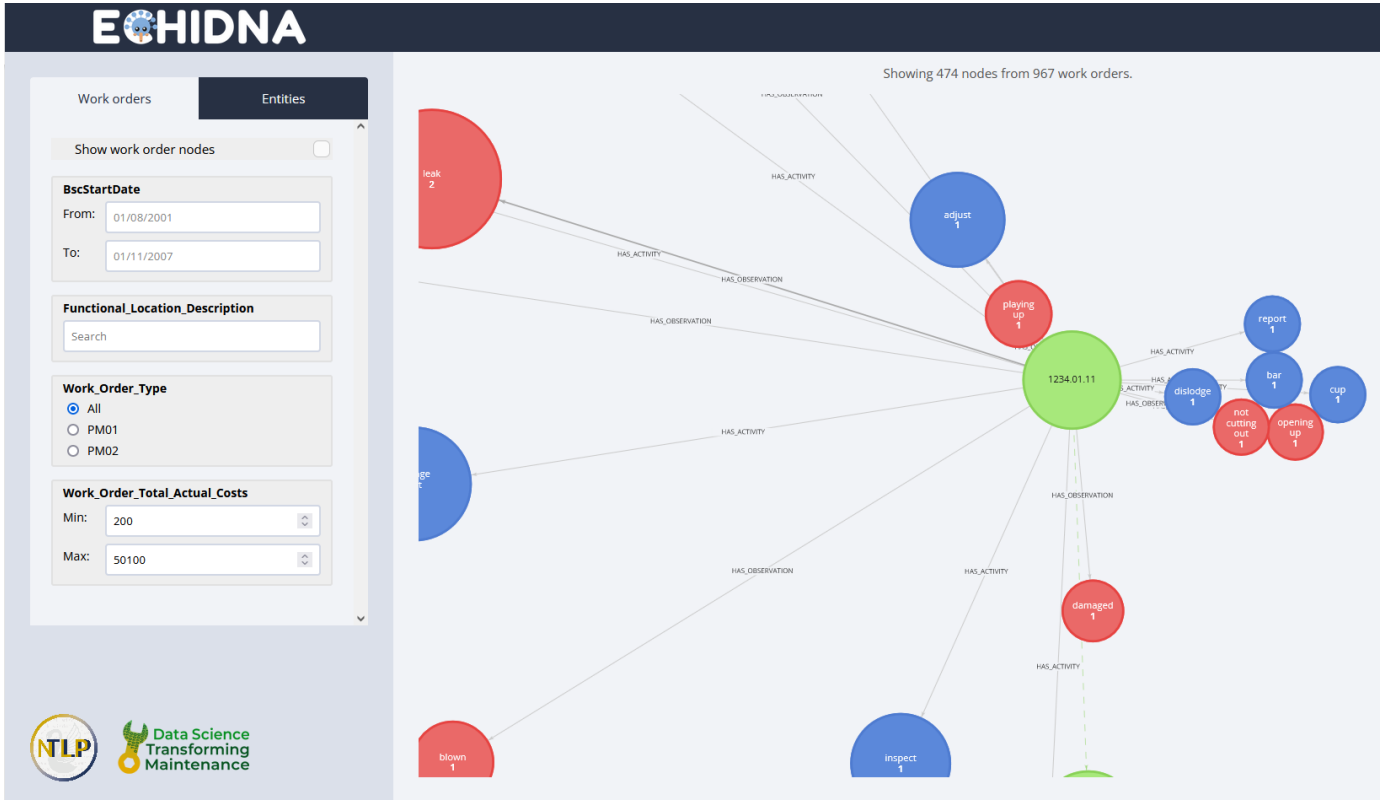
Examples of graphs in use today

Connected Papers: a graph-based tool for finding scientific papers

https://www.connectedpapers.com/

Examples of graphs in use today

Echidna: A graph-based tool for visualising maintenance work orders



https://nlp-tlp.org/echidna/

Companies using Graph Databases

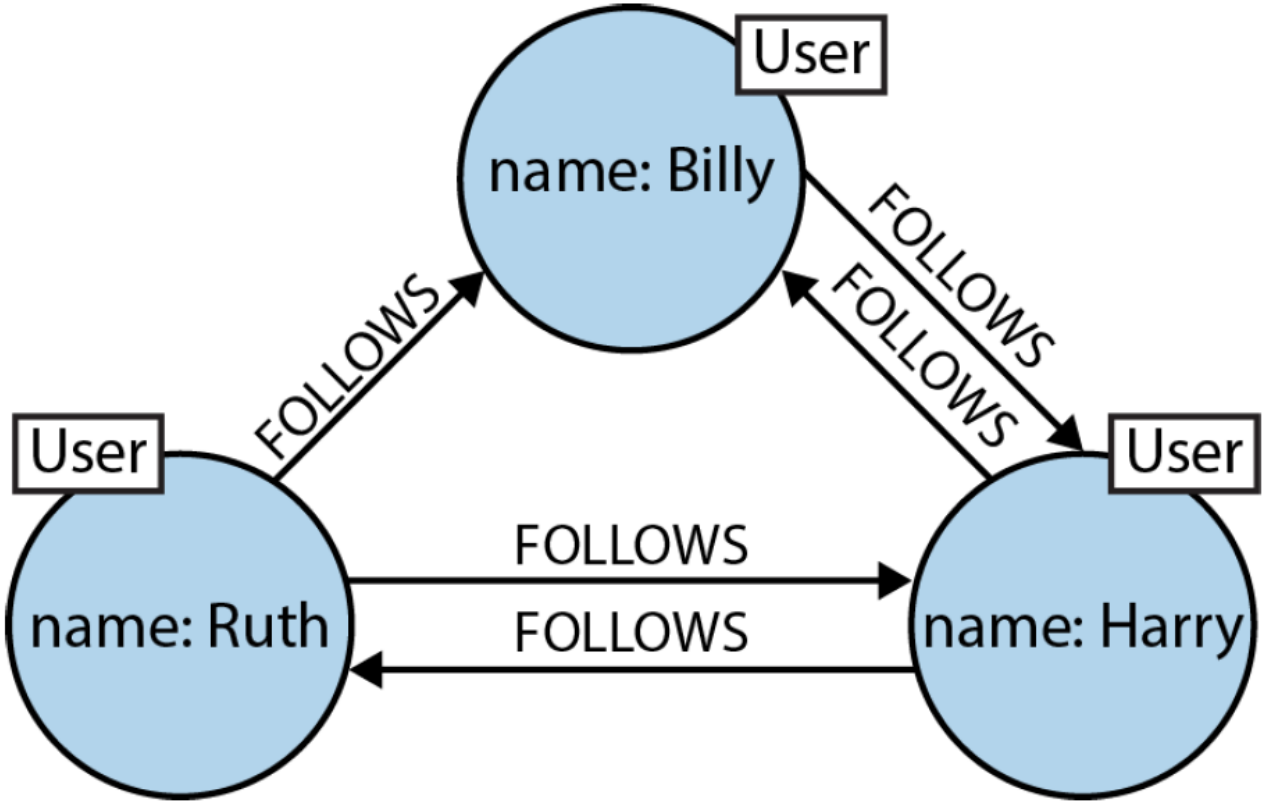
Outline – Part 1

- Intro to graph databases
- Property Graphs & Intro to Neo4j
- Why graph databases?
- From graph databases to knowledge graphs

Property Graph Model

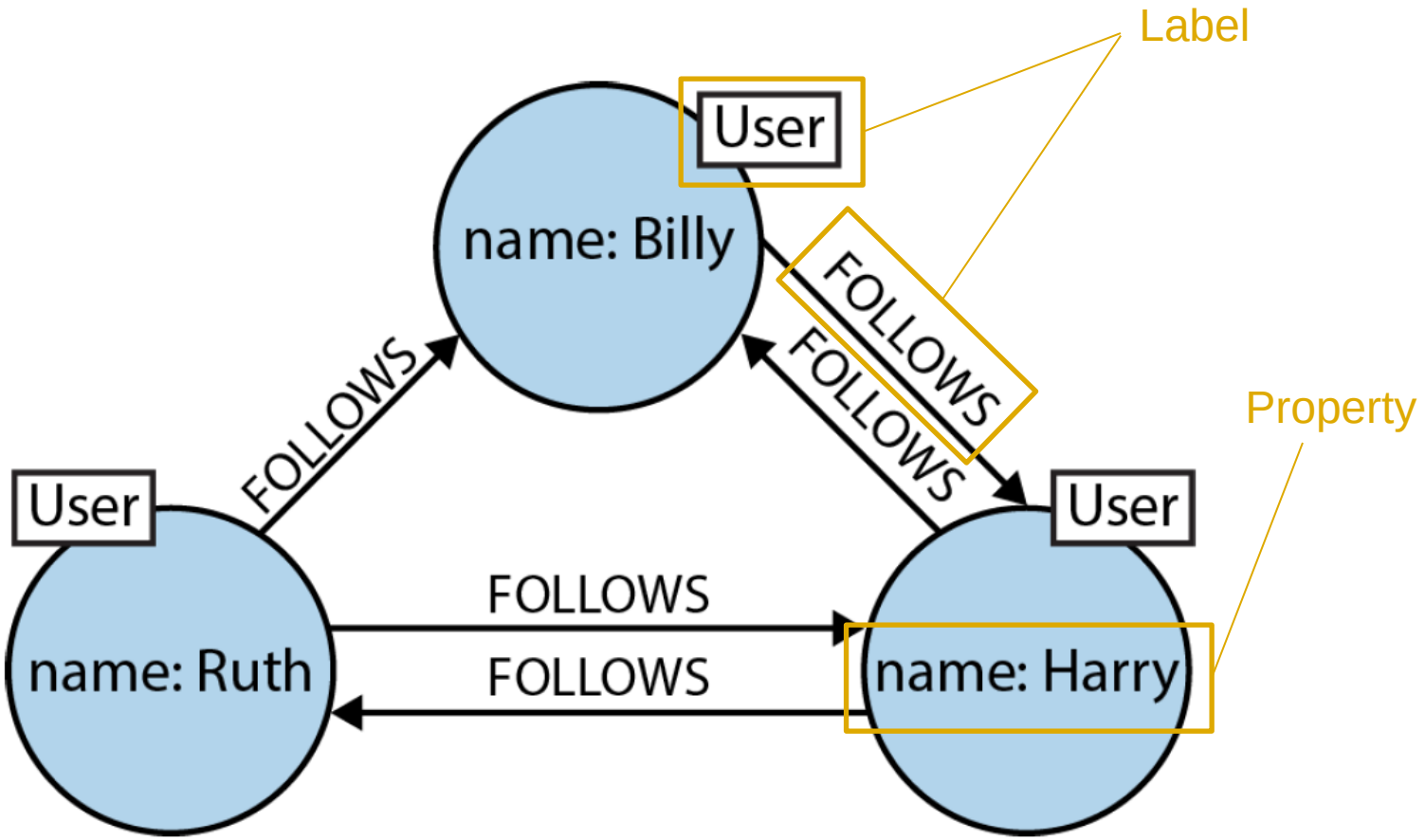
- The most common form of graph model is the **property graph model**, whereby:
  - The graph contains **nodes** and **relationships**.
  - A node may have zero or more **properties** (key-value pairs).
  - Nodes can be labelled with one or more **labels**.
  - Relationships can be **named** and **directed**, and always have a start and end node.
  - Relationships can also contain properties.

Simple Property Graph – Social Network



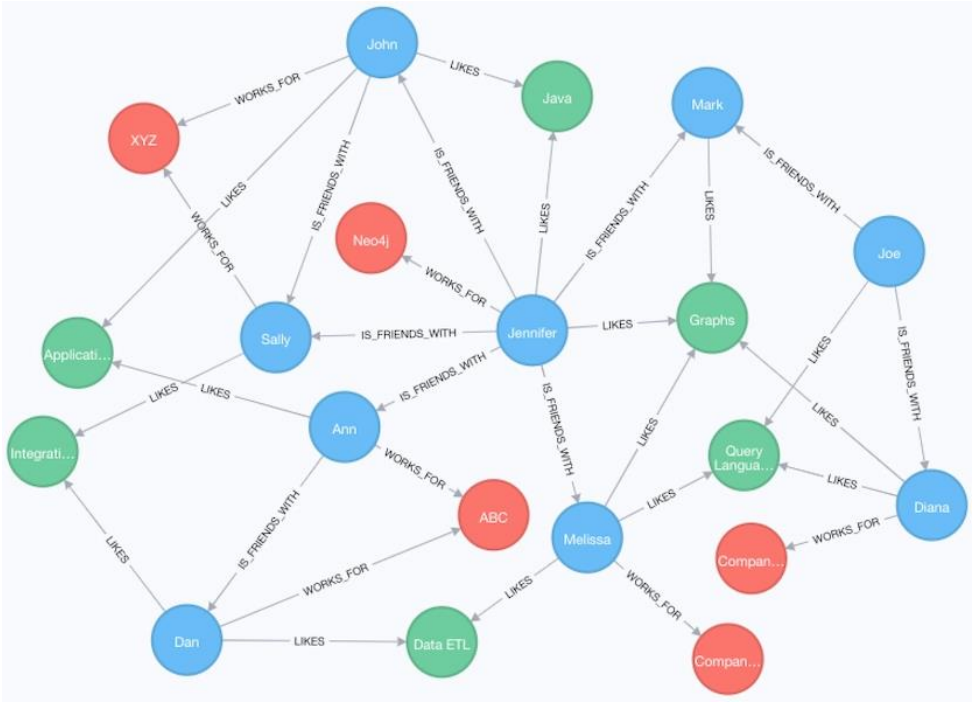
Source: Graph Databases by Robinson, Webber & Eifrem

Simple Property Graph – Social Network



*No relationship properties are shown here. An example might be the date in which the person followed another person.

Demo – Neo4j



Data Science Transforming Maintenance

Outline – Part 1

- Intro to graph databases
- Property graphs & intro to Neo4j
- Why graph databases?
- From graph databases to knowledge graphs

Why graph databases?

Graph databases are the best tools for both for representing the rich and varied **relationships** between things and for **recognizing patterns** based upon these relationships.

Edges are “first order citizens” just like the vertices.

Strengths of Graph Databases

- **Performance:** Queries over highly connected data are considerably faster than relational database queries.
- **Flexibility**: Graphs are **naturally additive**, meaning we can add new nodes, relationships, labels, properties etc on the fly without affecting existing queries.
- **Agility**: Graph databases are schema free, and are quick to set up and run.

Graphs vs Relational Tables

- Graph databases excel when compared to relational tables when **relationships** in the data are important.
- They are also much better at handling **unstructured data**, such as entities appearing in text.
- They are considerably **more flexible**.
- They are often more **space efficient** because they do not need to store “null” values.

Relational Database Example

User

UserID	User	Email	Address
1	Alice	alice@example.org	1 Duck St, …
2	Bob	bob@example.org	1 Duck St, …
3	…	…	37 Rabbit Lane…
4	Zach	zach@example.org	49 Rabbit Lane…

Product

ProductID	Description	Handling
321	Strawberry ice cream	freezer
765	Potatoes	null
…	…	…
987	Dried spaghetti	null

Order

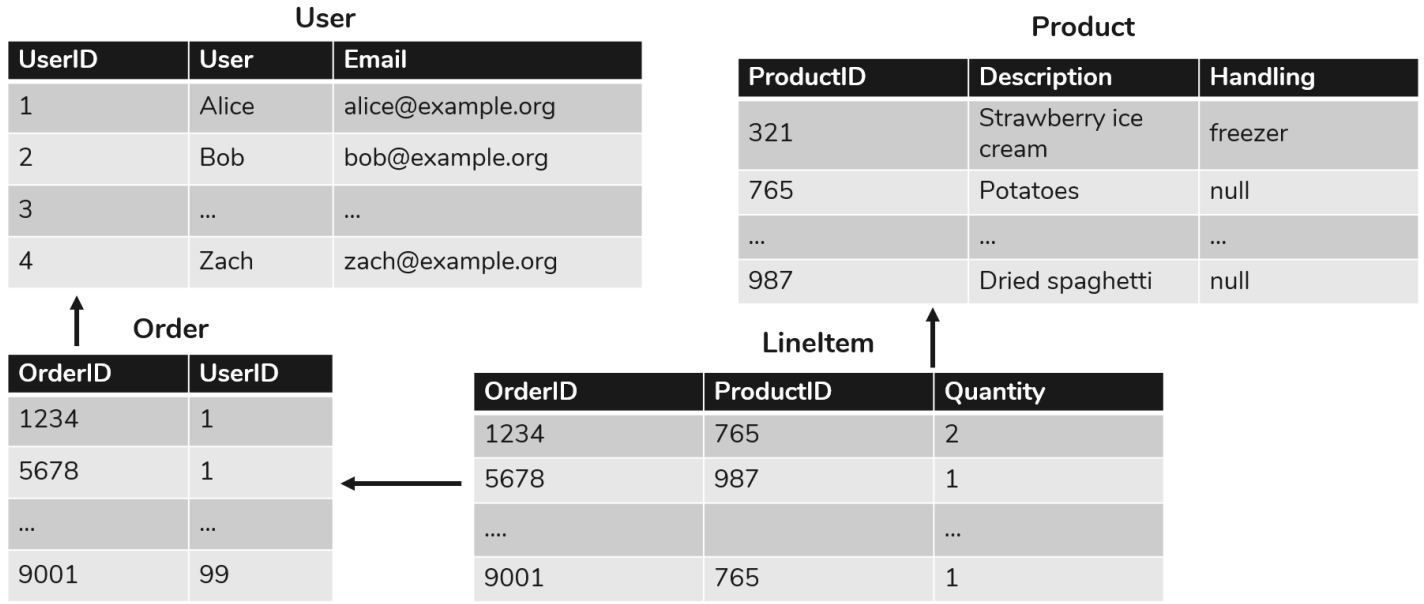
OrderID	UserID
1234	1
5678	1
…	…
9001	99

LineItem

OrderID	ProductID	Quantity
1234	765	2
5678	987	1
….		…
9001	765	1

Relational Database Example

- "Which items did a customer buy?"
- "Which customers bought this product?"
- "Which customers buying *this* product also bought *that* product?"



User

UserID	User	Email
1	Alice	alice@example.org
2	Bob	bob@example.org
3	...	...
4	Zach	zach@example.org

Product

ProductID	Description	Handling
321	Strawberry ice cream	freezer
765	Potatoes	null
...	...	...
987	Dried spaghetti	null

Order

OrderID	UserID
1234	1
5678	1
...	...
9001	99

LineItem

OrderID	ProductID	Quantity
1234	765	2
5678	987	1
....		...
9001	765	1

Relational Database Example

- RDBMs were originally designed to **codify** paper forms and tabular structures.
- Queries across multiple tables are
  - inefficient yet doable
  - prohibitively slow
- RDBMs schemas are **inflexible**, and can't keep up with dynamic and uncertain variables.

User

UserID	User	Email
1	Alice	alice@example.org
2	Bob	bob@example.org
3	...	...
4	Zach	zach@example.org

Product

ProductID	Description	Handling
321	Strawberry ice cream	freezer
765	Potatoes	null
...	...	...
987	Dried spaghetti	null

Order

OrderID	UserID
1234	1
5678	1
...	...
9001	99

LineItem

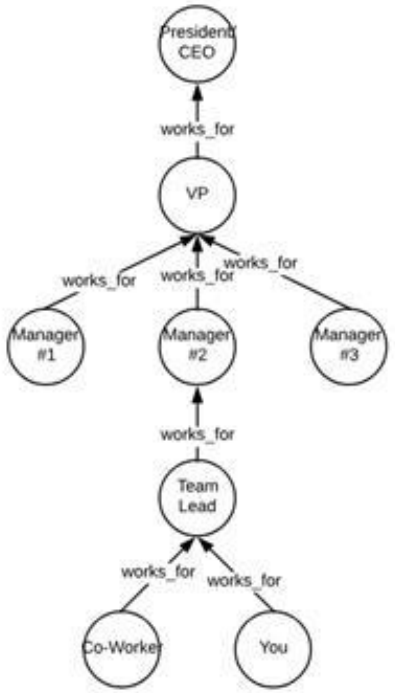
OrderID	ProductID	Quantity
1234	765	2
5678	987	1
....		...
9001	765	1

"I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail".
- Abraham Maslow (The Psychology of Science, 1966)

Recursive Queries

Given a list of employees and managers in a company, how we would determine a person's reporting hierarchy?

```
g.V().
  repeat(
    out('works_for')
  ).path().next()
```




```
WITH RECURSIVE org AS (
     SELECT employee_id,
            manager_employee_id,
            employee_name,
            1 AS level
     FROM org_chart
  UNION
     SELECT m.employee_id,
            e.manager_employee_id,
            e.employee_name,
            m.level+1 AS level
     FROM org_chart AS e
       INNER JOIN org AS m ON
e.manager_employee_id = m.employee_id
  )

SELECT employee_id,
manager_employee_id, employee_name,
FROM org
ORDER BY level ASC;
```

Outline – Part 1

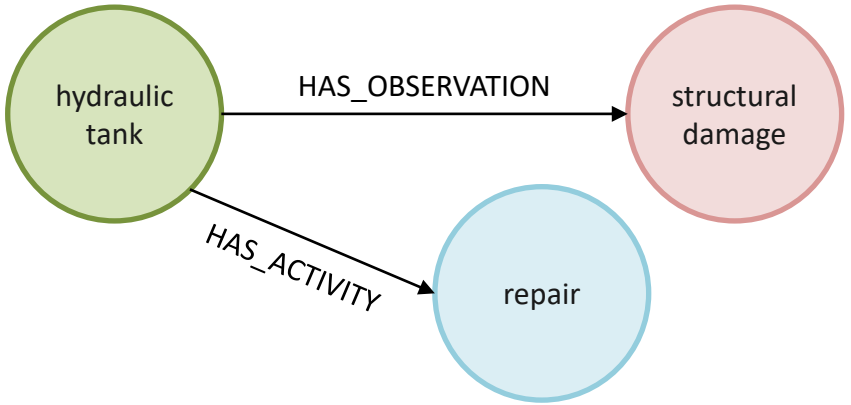
- Intro to graph databases
- Property graphs & intro to Neo4j
- Why graph databases?
- From graph databases to knowledge graphs

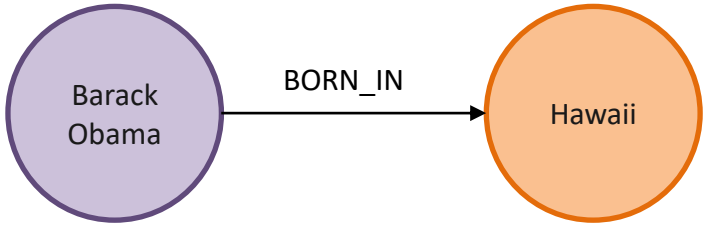
“The search for information takes 14-30 percent of the engineers’ time.”

Deloitte, “Wisdom of Enterprise Knowledge Graphs”

Knowledge Graphs

- A knowledge graph is a type of graph database that captures information about **entities**, **objects**, **events** or **concepts**.
- It is comprised of *triples,* i.e. facts in the form <entity_1, relation, entity_2>.
- Knowledge graphs are often built from unstructured text and extended to incorporate **structured information** from a range of data sources.





What makes knowledge graphs so useful?

Natural format for capturing knowledge in unstructured text

- Over **70% of data** in organisations is **unstructured**, and therefore inaccessible.
- **Noisy, unstructured text** is present in many domains, e.g. maintenance work orders, doctor's notes, safety records, and so on.
- For example, consider **maintenance work orders:**

```
replace seal on pump

fix a/c too hot
```

- Graph databases provide the means to **unlock the knowledge** captured within unstructured data and combine this with knowledge held within the structured data.

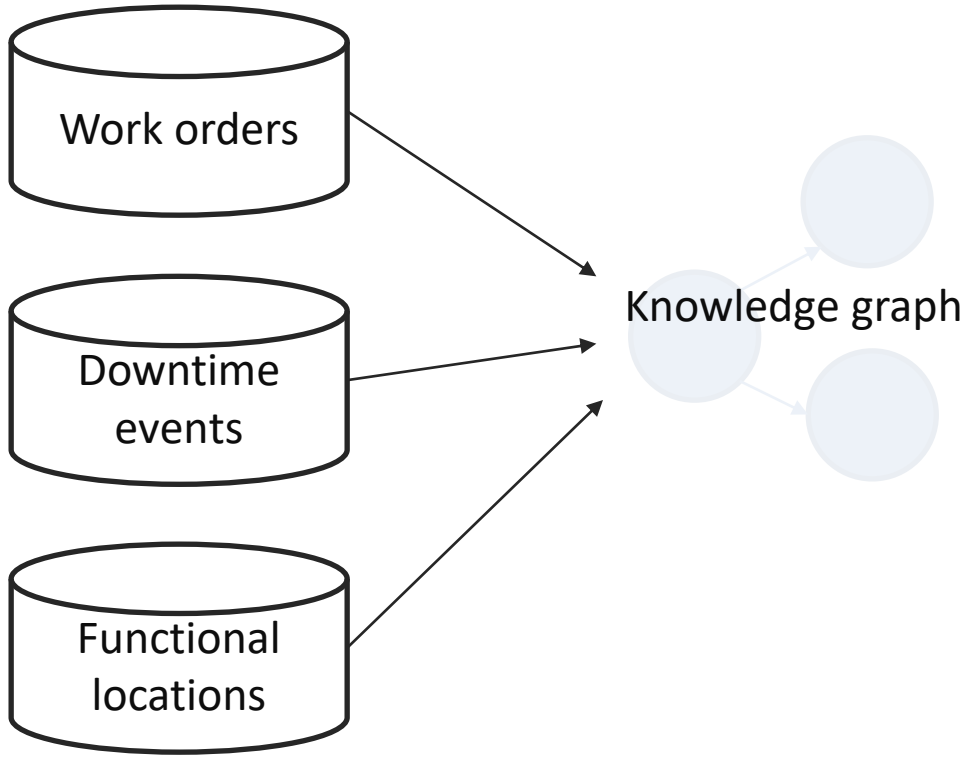
What makes knowledge graphs so useful?
Data integration

- Most companies maintain a wide range of different databases containing valuable knowledge.
- This data is often **unconnected** due to differences in formats and structure.
- Each separate database only captures a fragment of knowledge held within a company.
- Knowledge graphs provide the facility to integrate the data into one complex graph.

Photo found at Unsplash.com

What makes knowledge graphs so useful?
Data integration

Example – Maintenance Domain

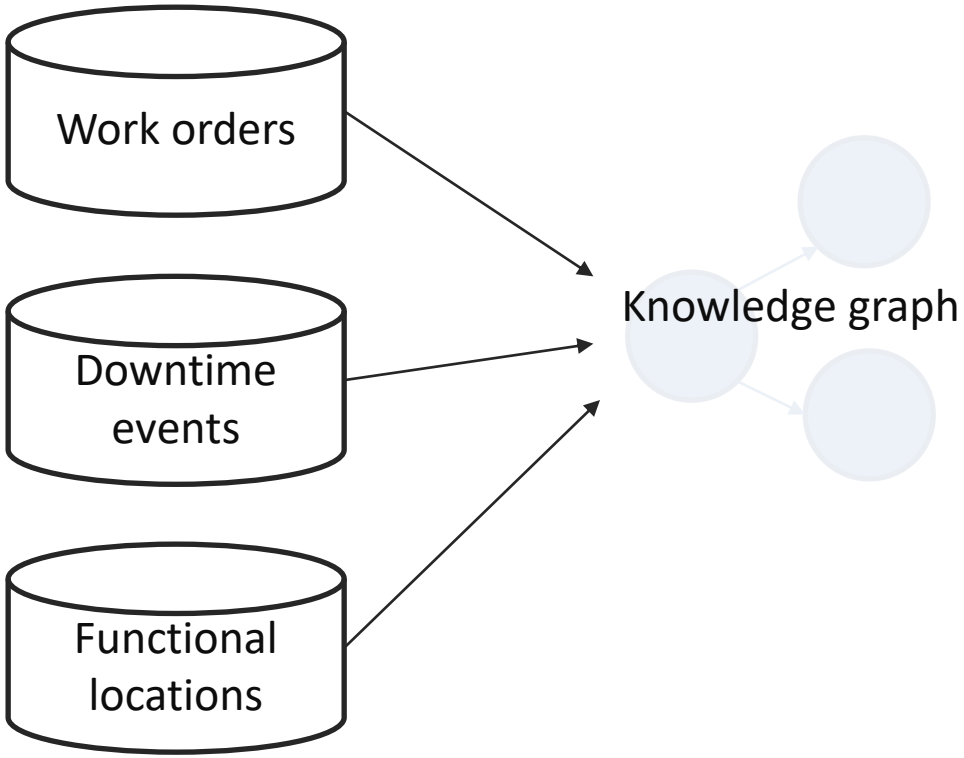


What makes knowledge graphs so useful?
Facilitating decision making

- Knowledge graphs facilitate **complex decision making** supported by collective knowledge from a range of domains, including unstructured data.
- They go hand in hand with machine learning, facilitating a range of machine learning opportunities such as **question answering**, **recommendation systems**, **supply chain management**.
- They are also excellent for increasing **data accessibility**, providing domain experts with the means to quickly ask questions across many different datasets.

What makes knowledge graphs so useful?
Facilitating decision making

Example – Maintenance Domain



Query

"Which pumps have cost more than $50,000 due to downtime in the past 6 months?"

Result

```
Pump 1234.0.1
Pump 1234.0.4
```

Examples of real-world knowledge graphs

- There are many real-world KGs available, for example:
  - **Wikidata**: The large-scale (700m + triples) KG behind Wikipedia.
  - **Freebase:** Massive (3b+ triples) KG used by Google.
  - **YAGO**: A huge semantic knowledge base derived from Wikipedia, Wordnet, and Geonames.
  - **Semantic Scholar**: A large KG of scientific literature.

There is an excellent list of real-world knowledge graphs available at https://github.com/totogo/awesome-knowledge-graph.

What makes a knowledge graph a “knowledge” graph?

- There is not yet a consensus on a formal definition of a knowledge graph.
- Wu et al.[1] describe a KG as having three essential components:
  - **Concepts** (an entity, attribute, or a fact)
  - **Relations**
  - **Background knowledge** about concepts and relations
- **Background knowledge** differentiates Knowledge Graphs from text graphs or data graphs.

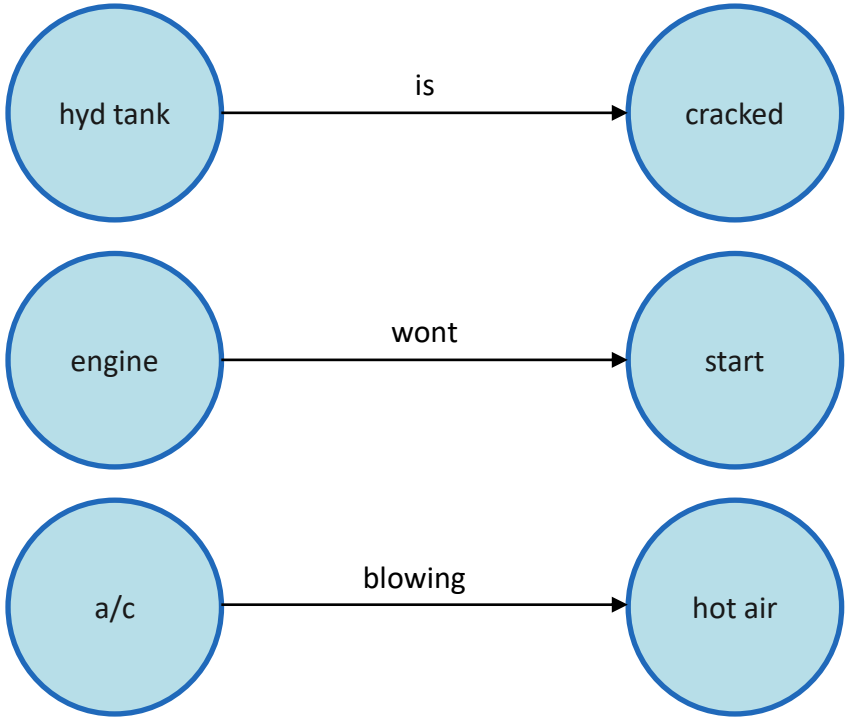
Wu, X., Wu, J., Fu, X., Li, J., Zhou, P., & Jiang, X. (2019, November). Automatic knowledge graph construction: A report on the 2019 ICDM/ICBK contest. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 1540-1545). IEEE

Text graph vs data graph vs knowledge graph

Work order
repair hyd tank is cracked
engine wont start
a/c blowing hot air
engin u/s

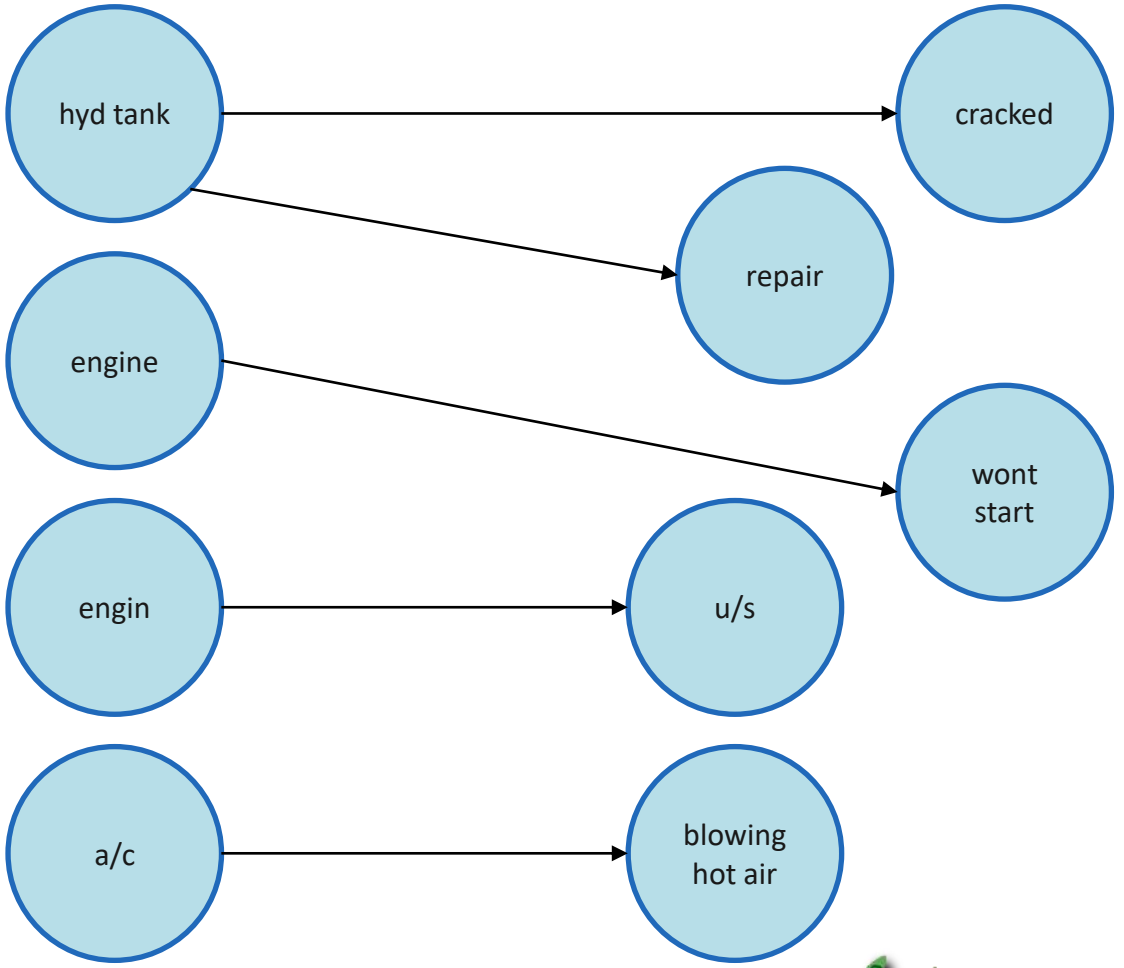
Text graph vs data graph vs knowledge graph

Work order
repair hyd tank is cracked
engine wont start
a/c blowing hot air
engin u/s



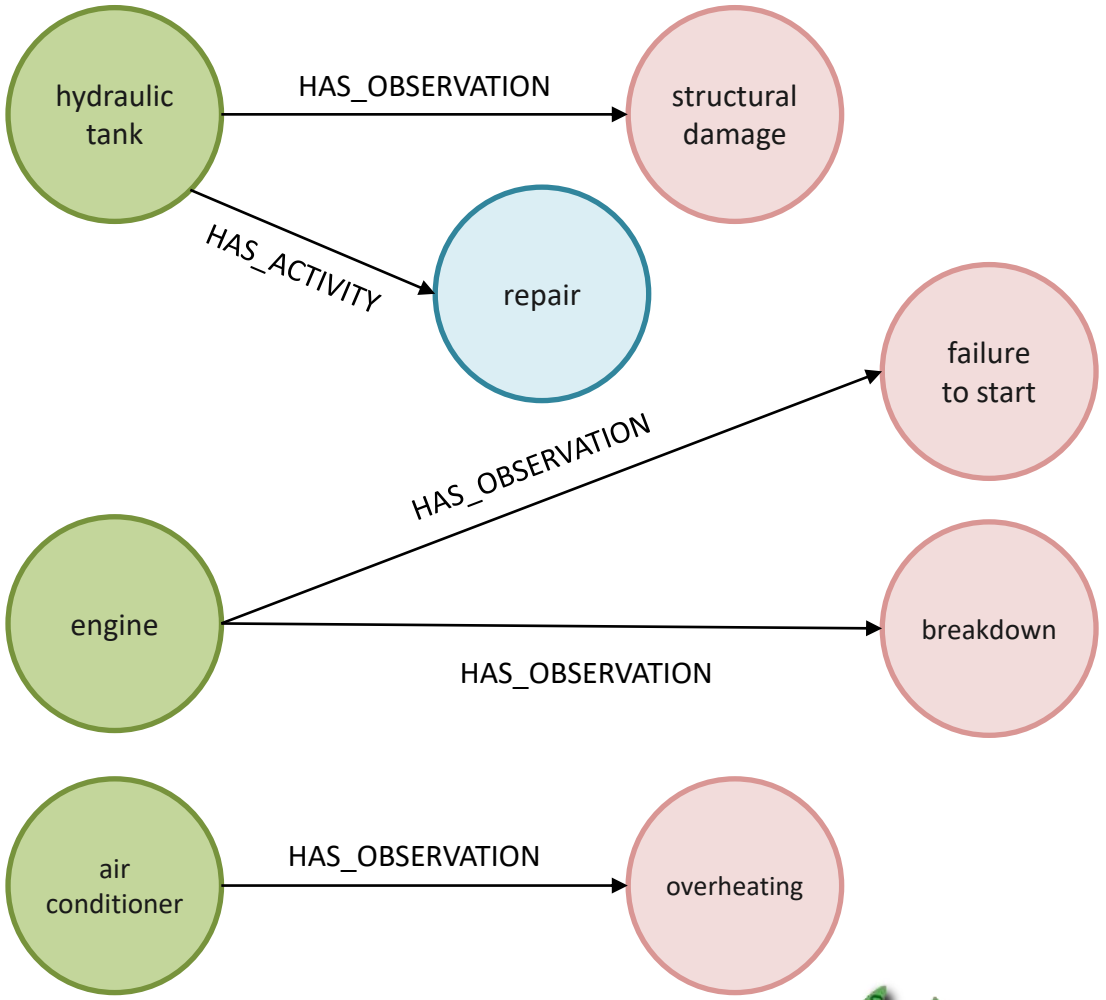
Text graph vs data graph vs knowledge graph

Work order
repair hyd tank is cracked
engine wont start
a/c blowing hot air
engin u/s



Text graph vs data graph vs knowledge graph

Work order
repair hyd tank is cracked
engine wont start
a/c blowing hot air
engin u/s



Knowledge Construction from Text (KGC)

- To build knowledge graphs from unstructured text we must employ **Natural Language Processing** (NLP) or **Technical Language Processing** (TLP) techniques, which we will demonstrate in the following session.
- Most approaches to knowledge graph construction from text are pipeline-based and include **three core components**:
  - Entity extraction
  - Relation extraction
  - Entity linking
- **Lexical normalisation** (i.e. text cleaning) is also an important technique for technical language.

Conclusion – Part 1

- Knowledge graphs are a powerful tool as they are able to **unlock knowledge** captured within the vast **unstructured text** present in many organisations.
- Graphs excel when dealing with **highly connected data**.
- They are also the perfect tool for **bringing data together** from across a range of areas in a business.

Conclusion – Part 1

- In the next session we will demonstrate the process of **Knowledge Graph Construction** via a Jupyter notebook walkthrough.
- We will introduce the key concepts behind **information extraction** (i.e. constructing knowledge graphs from text) – **lexical normalisation, entity recognition,** and **relation extraction.**
- In the final session we will look at how the knowledge graph can be **queried** in Neo4j in order to easily access important knowledge captured within the graph.

Questions

Email: michael.stewart@uwa.edu.au

CTMTDS
https://maintenance.org.au

UWA NLP-TLP Group
https://nlp-tlp.org

Echidna – Demo
https://nlp-tlp.org/echidna

Redcoat – Demo
https://nlp-tlp.org/redcoat