Kùzu

A Graph Database Management System for Python Graph Data Science

https://kuzudb.com

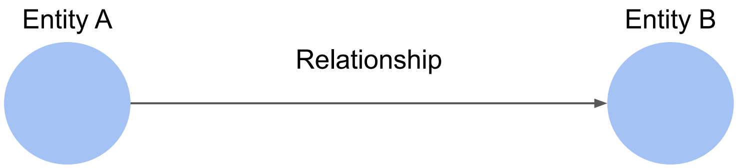
FOSDEM 2024

Outline

- What are graphs?
- When do you need graph modeling?
- Features of a competent graph database management system
- Kùzu’s vision
  - As a GDBMS
  - As the go-to solution for graph data science
- Walkthrough: How Kùzu makes graph data science easier

What are graphs/networks?

Abstract representation of entities and relationships



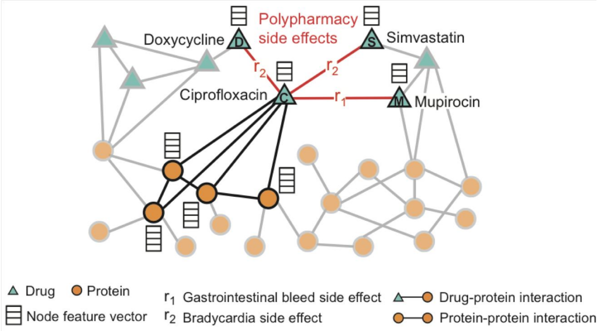
Entity A

Relationship

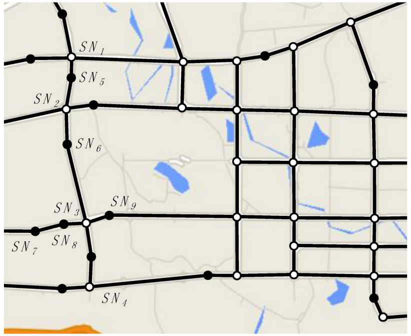
Entity B

Graphs: Natural ways to represent data

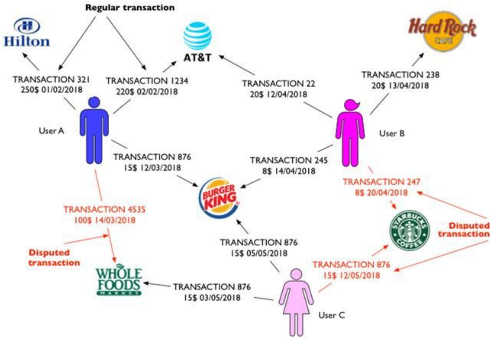
Social network


Polypharmacy side effects
Doxycycline
D
Simvastatin
S
Ciprofloxacin
C
Mupirocin
M
r_2
r_2
r_1
△ Drug
● Protein
Node feature vector
r_1 Gastrointestinal bleed side effect
r_2 Bradycardia side effect
Drug-protein interaction
Protein-protein interaction

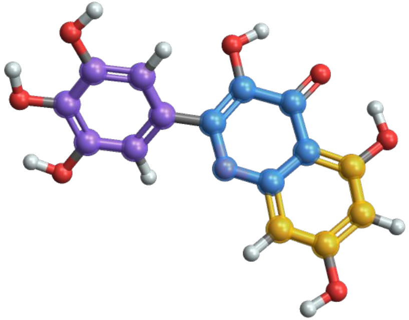
Drug interactions



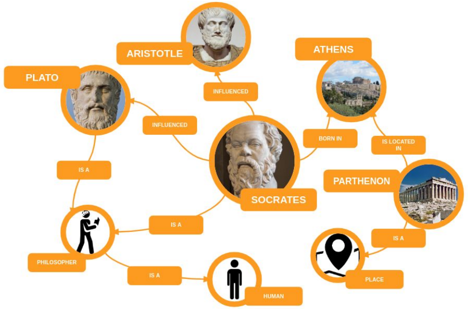
Traffic networks


Regular transaction
Hilton
AT&T
Hard Rock CAFE
TRANSACTION 321 250$ 01/02/2018
TRANSACTION 1234 220$ 02/02/2018
TRANSACTION 22 20$ 12/04/2018
TRANSACTION 238 20$ 13/04/2018
User A
User B
TRANSACTION 876 15$ 12/03/2018
TRANSACTION 245 8$ 14/04/2018
BURGER KING
TRANSACTION 247 8$ 20/04/2018
TRANSACTION 4535 100$ 14/03/2018
STARBUCKS COFFEE
Disputed transaction
Disputed transaction
TRANSACTION 876 15$ 05/05/2018
TRANSACTION 876 15$ 12/05/2018
WHOLE FOODS MARKET
TRANSACTION 876 15$ 03/05/2018
User C

Transaction graph



Molecular networks

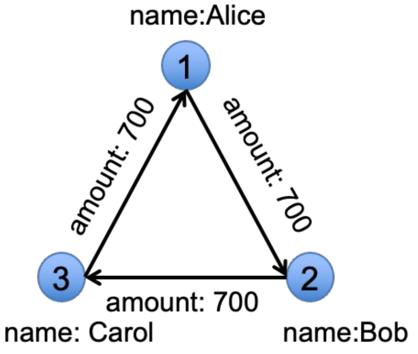


Knowledge graphs

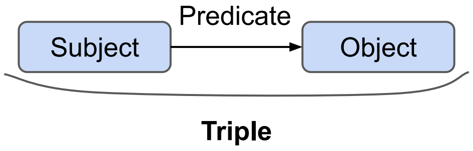
Graph DBMS: Overview

Data Models

Labelled Property Graph



RDF



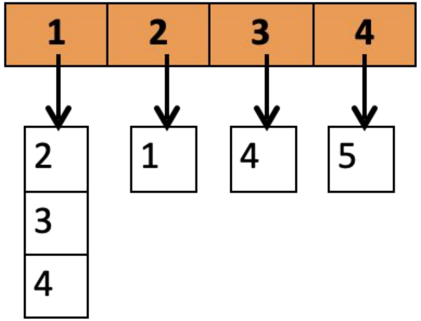
Query Language

High level query language designed for graphs

```
MATCH (a:account)-[:Transfer*]->(b:account)
WHERE a.name=Bob
RETURN b.name
```

Systems

Graph-specific storage structures, indices, operators



…
Shortest Paths
…

When do you need graph modeling?

Recursive joins, path-finding and identifying patterns

“Give me all direct or indirect possible sources of money flow into Alice’s account from Canada.”

```
MATCH (a:Account)-[:Transfer*]->(b:Account)
WHERE a.location=“Canada” AND b.owner=“Alice”
RETURN *
```

Recursive SQL? Hard!

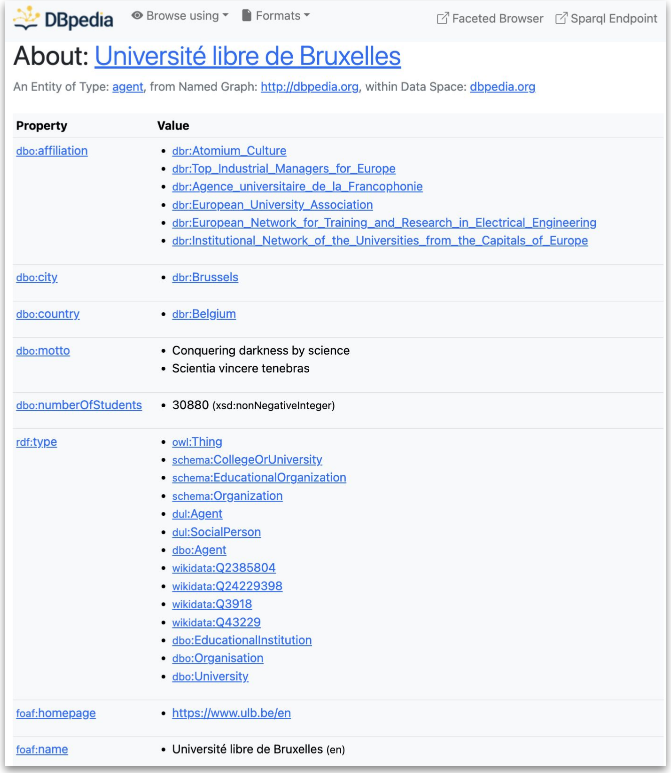
“Give me shortest path of money flow into Alice’s account from Canada.”

```
MATCH (a:Account)-[:Transfer* SHORTEST]->(b:Account)
WHERE a.location=“Canada” AND b.owner=“Alice”
RETURN *
```

*Recursive SQL? **Super** Hard!*

When do you need graph modeling?

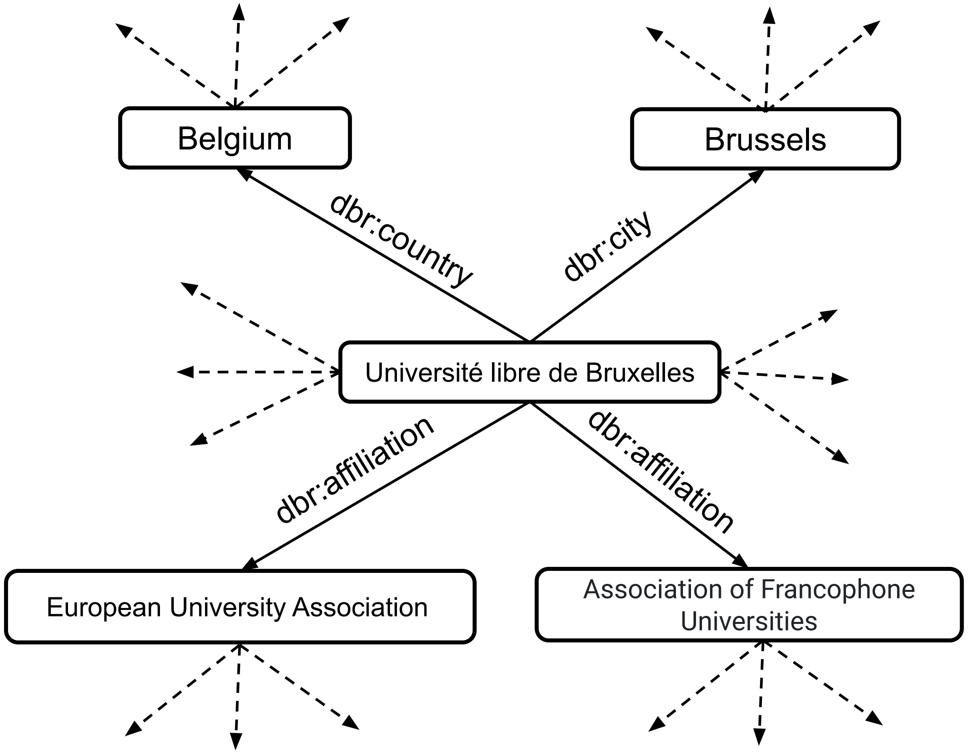
Very heterogeneous datasets to construct “knowledge graphs”



About: Université libre de Bruxelles

An Entity of Type: agent, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

Property	Value
dbo:affiliation	dbr:Atomium_Culture dbr:Top_Industrial_Managers_for_Europe dbr:Agence_universitaire_de_la_Francophonie dbr:European_University_Association dbr:European_Network_for_Training_and_Research_in_Electrical_Engineering dbr:Institutional_Network_of_the_Universities_from_the_Capitals_of_Europe
dbo:city	dbr:Brussels
dbo:country	dbr:Belgium
dbo:motto	Conquering darkness by science Scientia vincere tenebras
dbo:numberOfStudents	30880 (xsd:nonNegativeInteger)
rdf:type	owl:Thing schema:CollegeOrUniversity schema:EducationalOrganization schema:Organization dul:Agent dul:SocialPerson dbo:Agent wikidata:Q2385804 wikidata:Q24229398 wikidata:Q3918 wikidata:Q43229 dbo:EducationalInstitution dbo:Organisation dbo:University
foaf:homepage	https://www.ulb.be/en
foaf:name	Université libre de Bruxelles (en)



Belgium
Brussels
dbr:country
dbr:city
Université libre de Bruxelles
dbr:affiliation
dbr:affiliation
European University Association
Association of Francophone Universities

Feature set of a competent graph database

What every competent GDBMS should do:

- Pre-defined/pointer-based joins: joins are defined between nodes through edges
- Many-to-many growing joins
- Heterogeneous datasets (e.g., knowledge graphs)
- Recursive join queries

```
MATCH (a:Account)-[:Transfer*]->(b:Account)
```

- Schema querying

```
MATCH (a:Account)-[e1]->(b:Account)-[e2]->(c:Account)
WHERE type(e1) != type(e2)
RETURN *
```

Blog post: https://kuzudb.com/blog/what-every-gdbms-should-do-and-vision.html

Kùzu Vision: GDBMS

Kùzu aims to represent the state-of-the-art of how graphs should be stored, indexed and queried

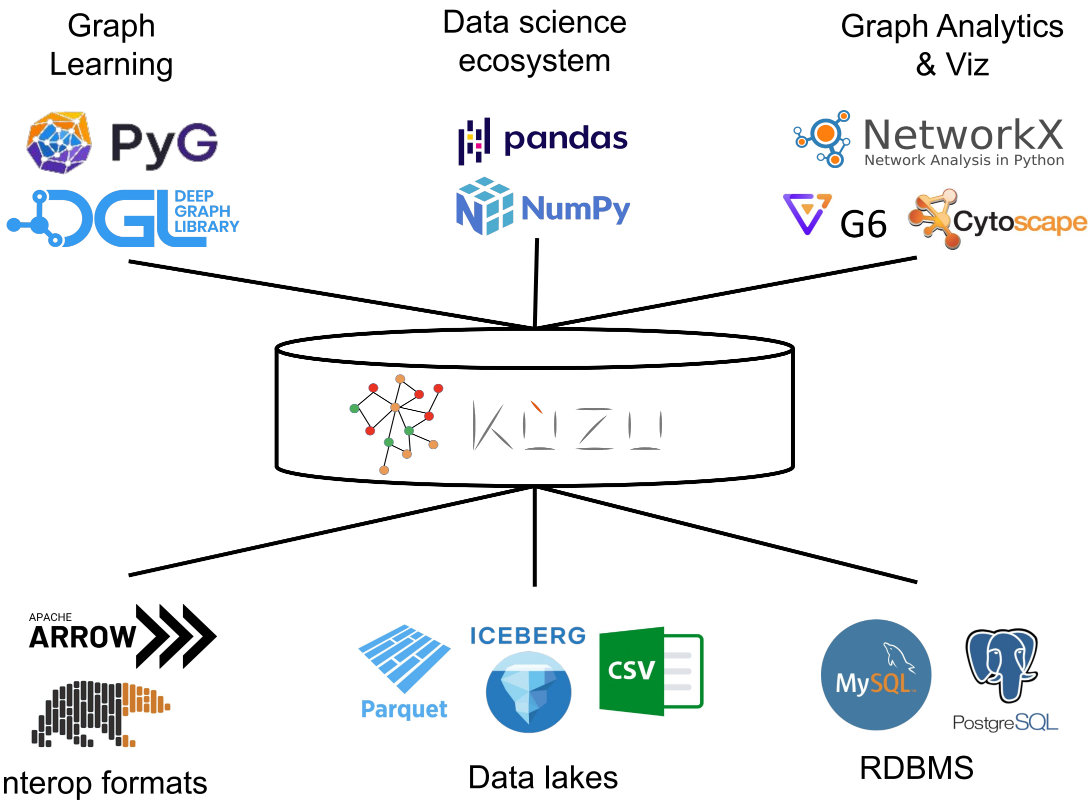
- **Highly scalable** to several TBs of data
- **Very fast** query speeds
- **Property graphs** + **RDF** support via **Cypher** query language
- **Easy to use** & embeddable (like DuckDB/SQLite, but for graphs)

UNIVERSITY OF WATERLOO | DSG Data Systems Group → Kùzu Inc.

Kùzu is based on many years of research at University of Waterloo.
It's now being developed at a spin-off company called Kùzu Inc.

Kùzu Vision: Graph Data Science

To be the go-to backend for graph modeling and data science



Graph Learning

Data science ecosystem

Graph Analytics & Viz

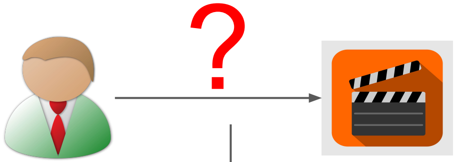
Interop formats

Data lakes

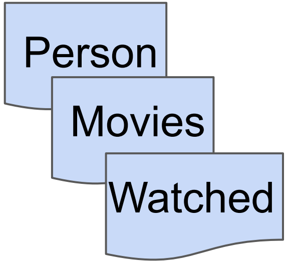
RDBMS

How Kùzu makes Graph Data Science easier

- 2 data sources w/ people, their friends & movies they watch
- Goal: Movie recommender system
- Approach: Graph Neural Network (GNN)-based link prediction

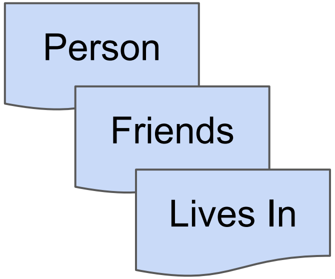


Dataset 1



name	age	watched
Alice	25	m1
Bob	20	m2
Carl	30	m3

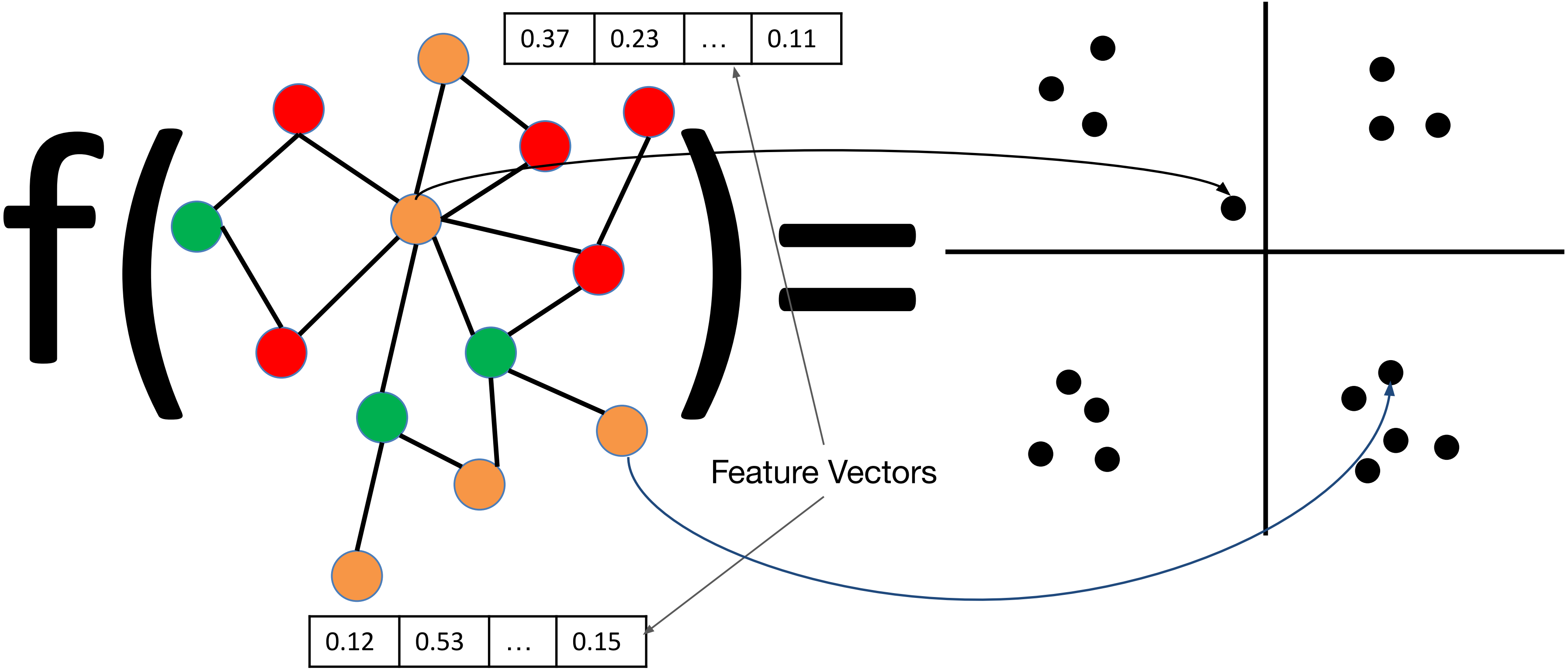
Dataset 2



person	friend	LivesIn
Alice	Bob	Toronto
Bob	Carl	Waterloo
Carl		Waterloo

A very brief overview of graph ML

- Graph ML goal: Embed nodes into a vector space
- Benefits: Models incorporate the structure of a graph based on its relationships

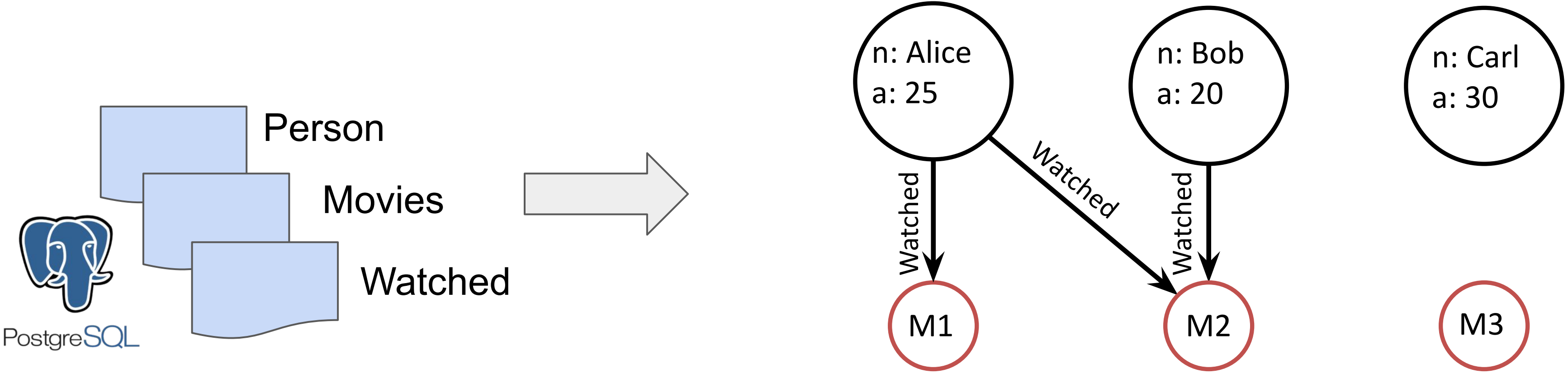


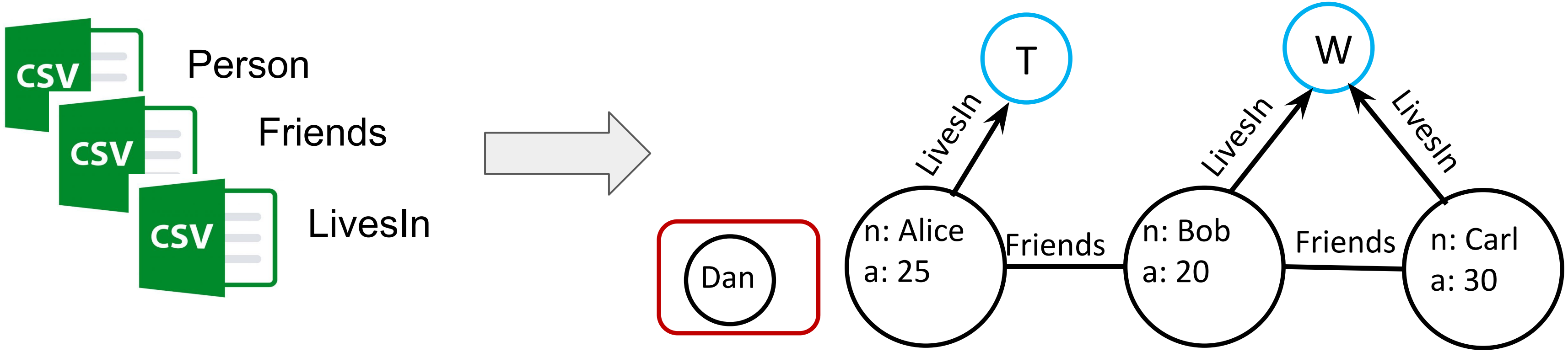
0.37	0.23	…	0.11

0.12	0.53	…	0.15

How Kùzu makes Graph Data Science easier

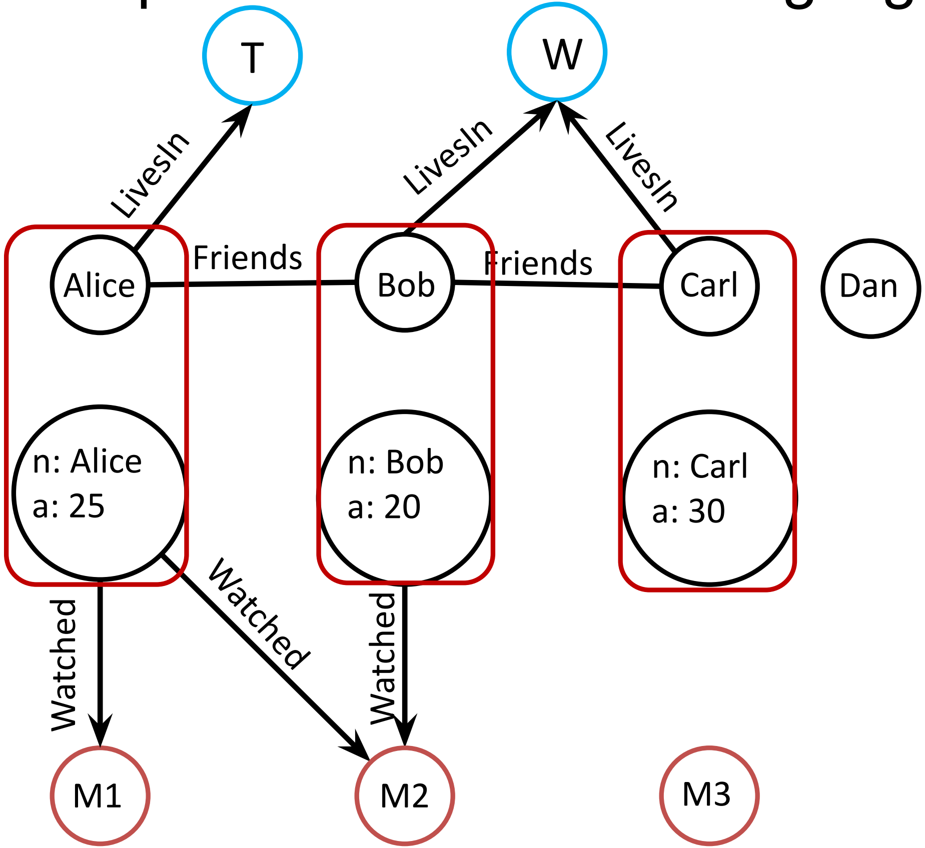
Data might exist in multiple sources

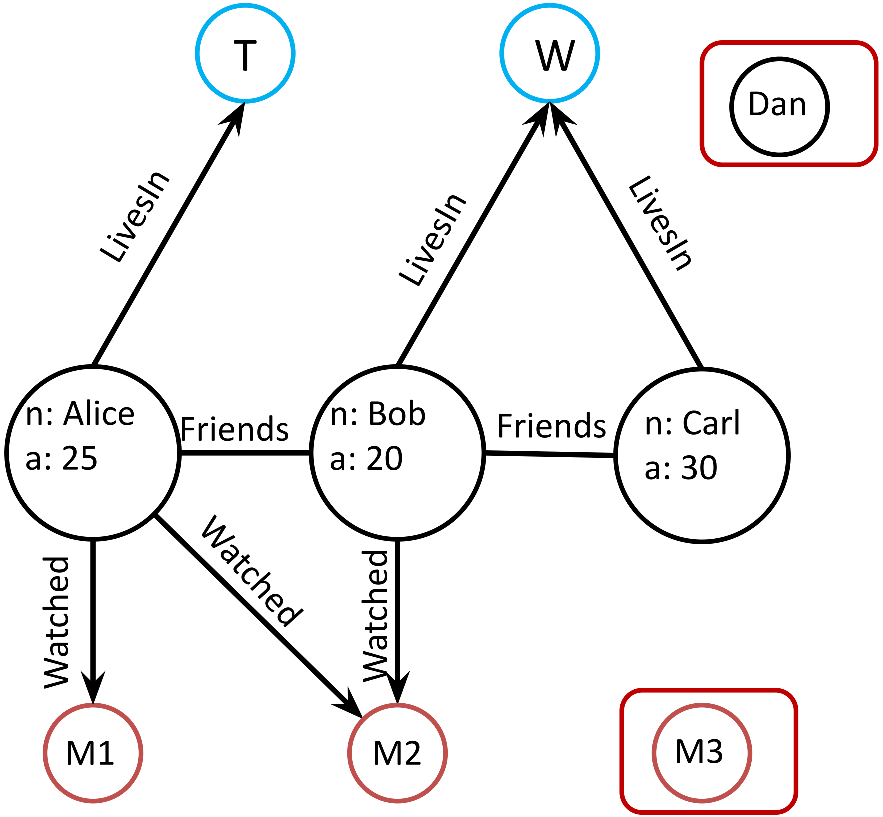




How Kùzu makes Graph Data Science easier

Deduplication + remove dangling nodes





How Kùzu makes Graph Data Science easier

Steps in graph learning pipeline with Kùzu

1. Install
2. Load data to a graph
3. Deduplication
4. Remove dangling nodes
5. Export to PyG
6. Train GNN
7. Make predictions

```
pip install kuzu
```

```
import kuzu
…
conn.execute(‘CREATE NODE TABLE person …’)

conn.execute(‘COPY person FROM …’)
…
```

How Kùzu makes Graph Data Science easier

Steps in graph learning pipeline with Kùzu

1. Install
2. Load data to a graph
3. Deduplication
4. Remove dangling nodes
5. Export to PyG
6. Train GNN
7. Make predictions

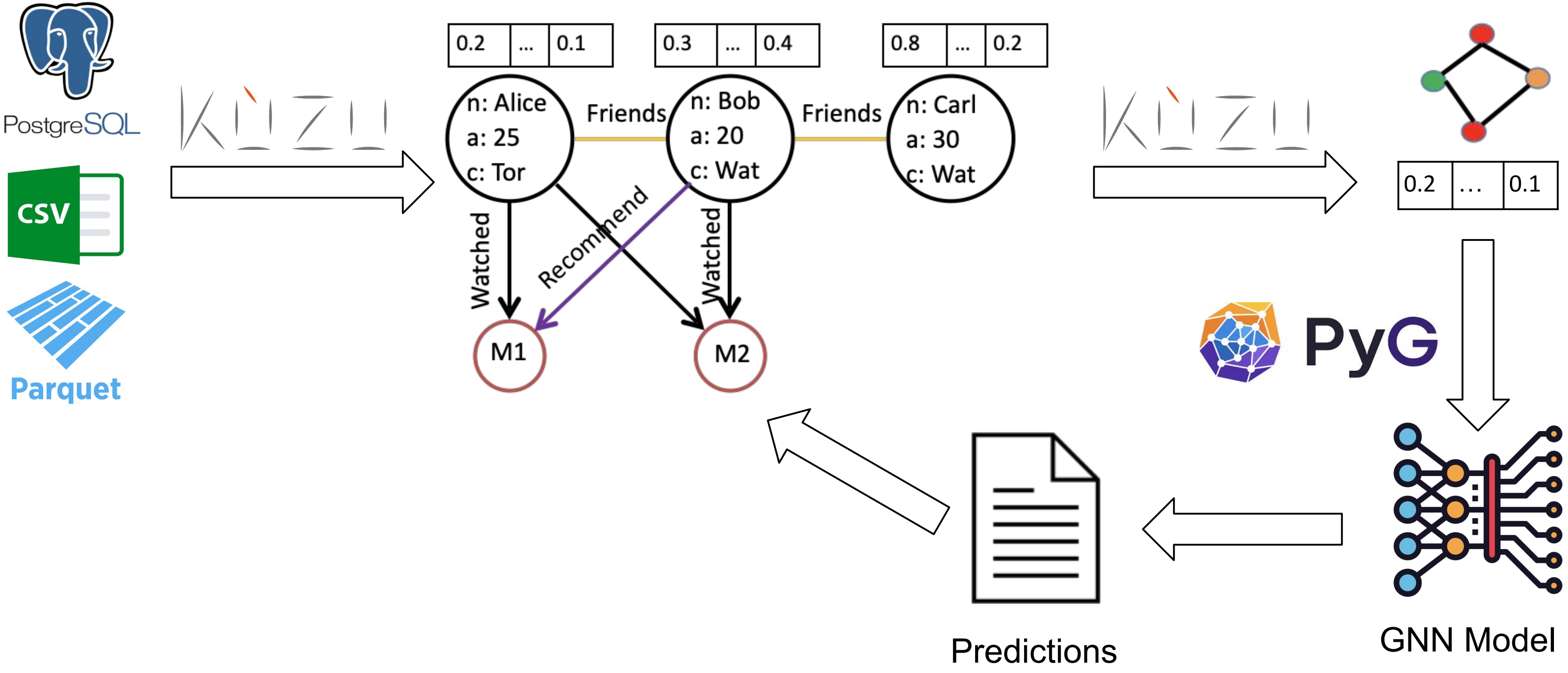
```
conn.execute(“
MATCH (a:User), (b:User2)-[e:LivesIn]->(c:City)
WHERE a.name = b.name
CREATE (a)-[:LivesIn]->(c)
“)
```

```
conn.execute(“
MATCH (a) WHERE NOT EXISTS (a)-[]->(b)
DELETE a
“)
```

Once data and features exist are loaded to a graph, work in PyG as normal

How Kùzu makes Graph Data Science easier

Graph learning pipeline



Predictions

GNN Model

Takeaways

- Kùzu is an in-process, analytical graph database system
  - “Like DuckDB/SQLite, but for graphs”
- Highly scalable: optimized for multi-core parallelism
- Integrated with the PyData ecosystem: numpy, pyarrow, NetworkX, PyTorch
- Support for property graphs and RDF graphs via Cypher query language
- Embeddable and easy to use from within your application

Embeddable with Rich Bindings

Kùzu is **open source** (MIT licensed)
Give it a try and star us on GitHub!

Join our Discord!

github.com/kuzudb/kuzu

@kuzudb

https://kuzudb.com/blog/