Meta-Property Graphs: Extending Property Graphs with Metadata Awareness and Reification

DBDBD 2024

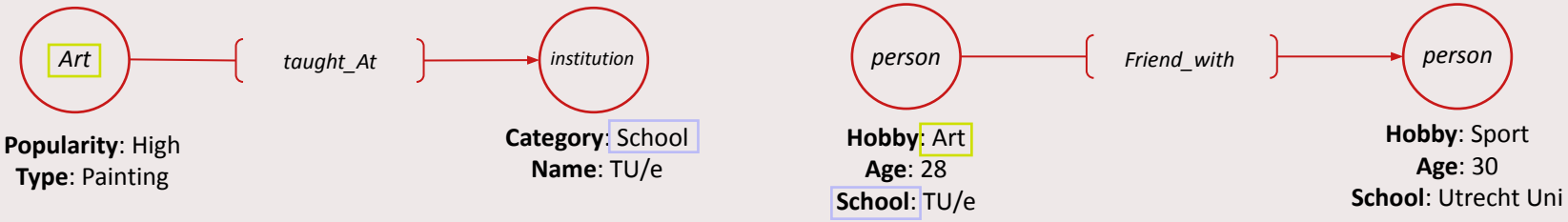
Sepehr Sadoughi, Nikolay Yakovets, George Fletcher

Mathematics and Computer Science, Database group

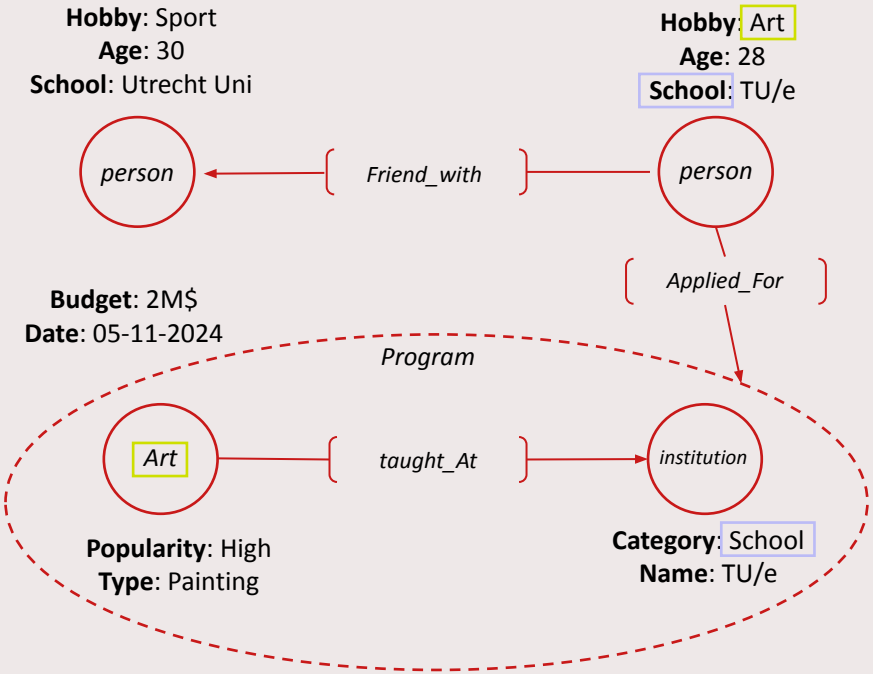
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Motivation (problem definition)

- **Graph databases** are emerging. And now we have the **GQL ISO standard**.
- Need for support **Heterogeneity** between data and meta-data in Modern data engineering applications
  - e.g.: (1) when **integrating** data sources for building a **knowledge graph,** one source data value may correspond to another source label or attribute key. (2) **Auditing** an existing KG it might require an **aggregation** of different data objects for annotation.



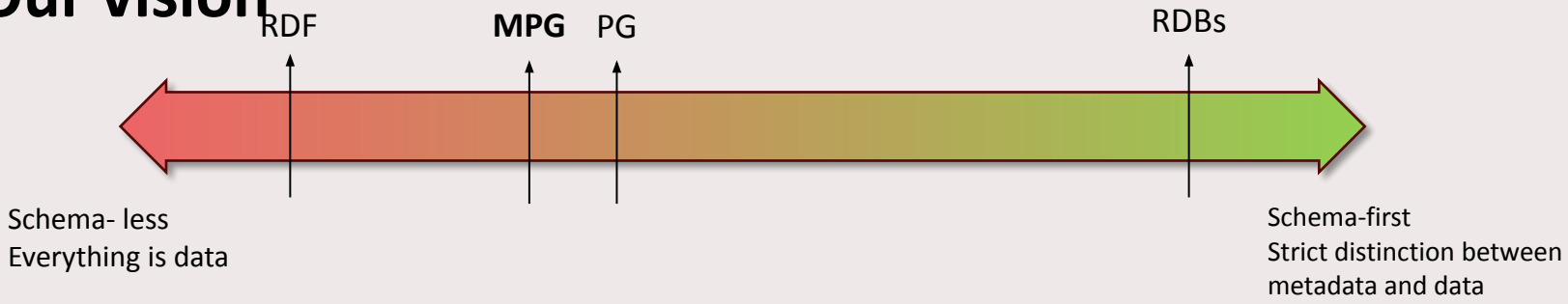
Motivation (problem definition)



PG – GQL ISO standard

Pros	Cons
Schema-flexible	Strict distinction between metadata and data
Support properties and labels over both nodes and edges	No support for reification

Our vision



Contribution 1: Meta-Property Graph Model

- Treat properties and label sets as **first-class citizen** and data objects.
- Enabling **promotion and demotion** of Meta-data (i.e. property keys and label sets) to data.
- Extending property graph data model to support **reification** as the second form of metadata.

Contribution 2: MetaGPML

- A **backward compatible** extension of GPML query language to support the Meta-property Graph features.

Meta-Property Graph Model

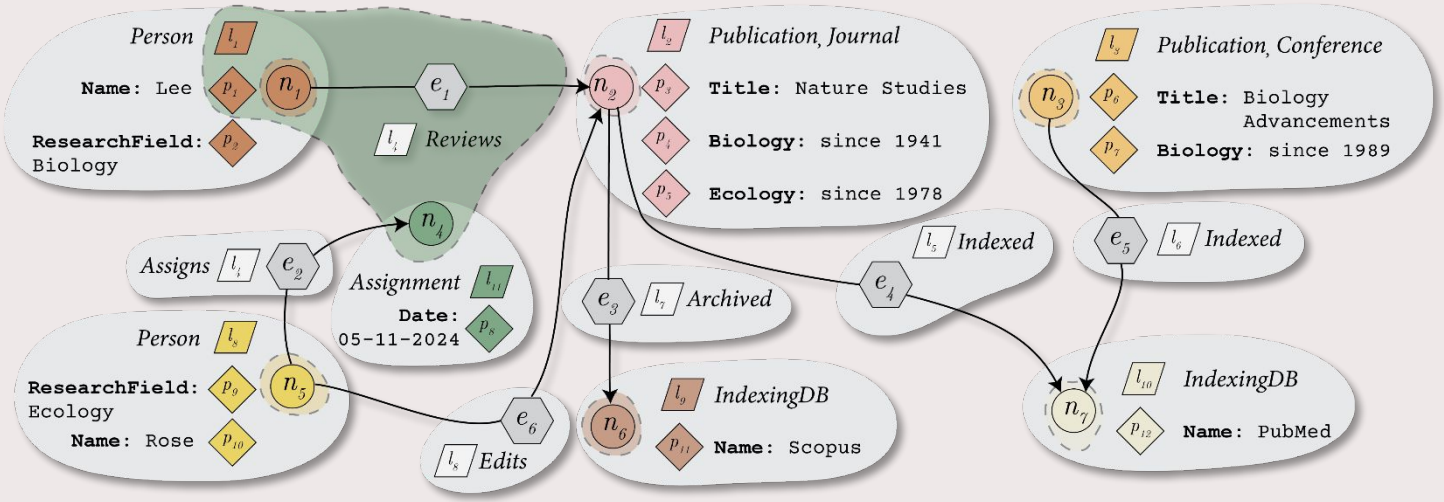


Fig 1: MPG example scenario

Meta-GPML to go

Nodes	
`(x:l)`	Node x with label l in it's label set
`(x:?y)`	Node x with label set y
`(x:?y::π)`	Node x with label set y and reified pattern π
`(x:?y::π).z`	Node x with label set y and reified pattern π and property z
`(x:l::π).z`	Node x with label l and reified pattern π and property z

Edges	
`-[x]->`	Edge x
`-[x:l]->`	Edge x with label l
`-[x:?y]->`	Edge x with label set y
`-[x:?y].z->`	Edge x with label set y and property z
`-[x:l].z->`	Edge x with label l and property z

Properties	
`{x}`	Property x
`KEY(x)`	Key of property x
`VAL(x)`	Value of property x

Labels	
`|x|`	Label x
`c ELEMENTOF x`	Check if c exist in label x

TU/e

How it works

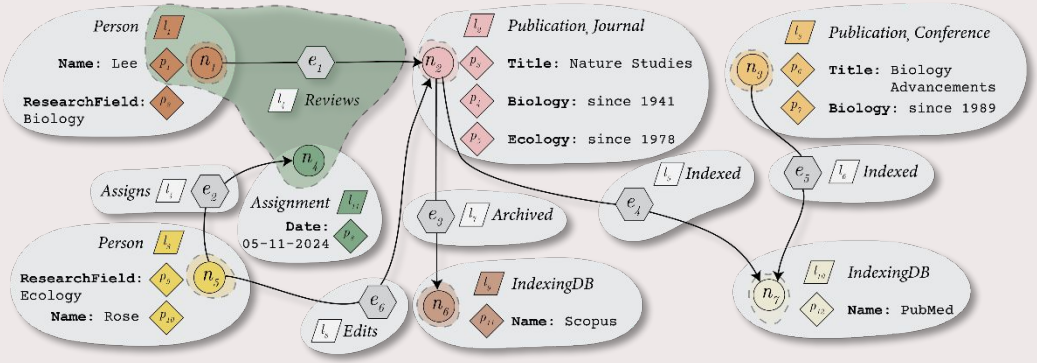


Fig 1: MPG example scenario

Q1: Which label sets contain the label `Publication`?

```
MATCH |l|
WHERE "Publication" ELEMENTOF l
RETURN l AS "Publication_Co_Tags"
```

Publication_Co_Tags
{"Publication", "Journal"}
{"Publication", "conference"}

Q2: What are the values of `Name` properties?

```
MATCH {p}
WHERE KEY(p) = "Name"
RETURN VAL(p) AS "Names"
```

Names
Lee
Scopus
Rose
PubMed

How it works

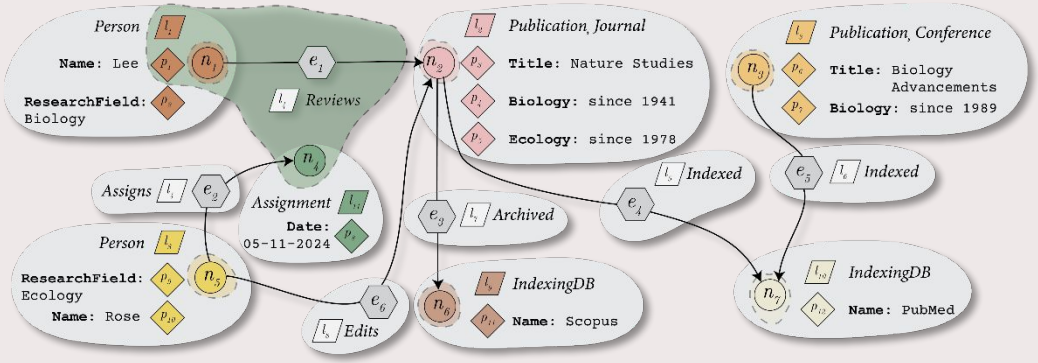


Fig 1: MPG example scenario

Q3: Which relationships does each `Publication` have with each `Indexing_DB`?

```
MATCH (x:Publication)-[:?y]->(z:Indexing_DB)
RETURN x.Title AS "Title", y AS z.Name
```

{ (Title → ”Nature Studies”, Scopus → ”Archived”) ,
(Title → ”Nature Studies”, PubMed → ”Indexed”) ,
(Title → ”Biology Advancements”, PubMed → ”Indexed”)}

Q4: Who assigned `Lee` as a reviewer and when?

```
MATCH (x:Person)-[:assigns]->
(y::(z:Person)-[:reviews]->())
WHERE z.Name = "Lee"
RETURN z.Name AS "reviewer name",
y.Date AS "Date",
x.Name AS "Assigning editor"
```

Reviewer name	Date	Assigning editor
Lee	05-11-2024	Rose

Next steps towards the MPG vision

Thank you!

Open research challenges:

- Physical representations and indexing strategies for MPGs
- Query evaluation and optimization solutions for the novel capabilities of MetaGPML
- Extending the GQL and SQL/PGQ languages with MetaGPML
- Schema and constraint languages for MPG, building upon the PG-Schema framework
- Educational and training resources for students and professionals in working with MPGs and MetaGPML

Sepehr Sadoughi – PhD candidate
s.sadoughi@tue.nl