Knowledge Graphs & Exploration: How to Find Your Way in the Data Wilderness

Matteo Lissandrini

https://people.cs.aau.dk/~matteo/

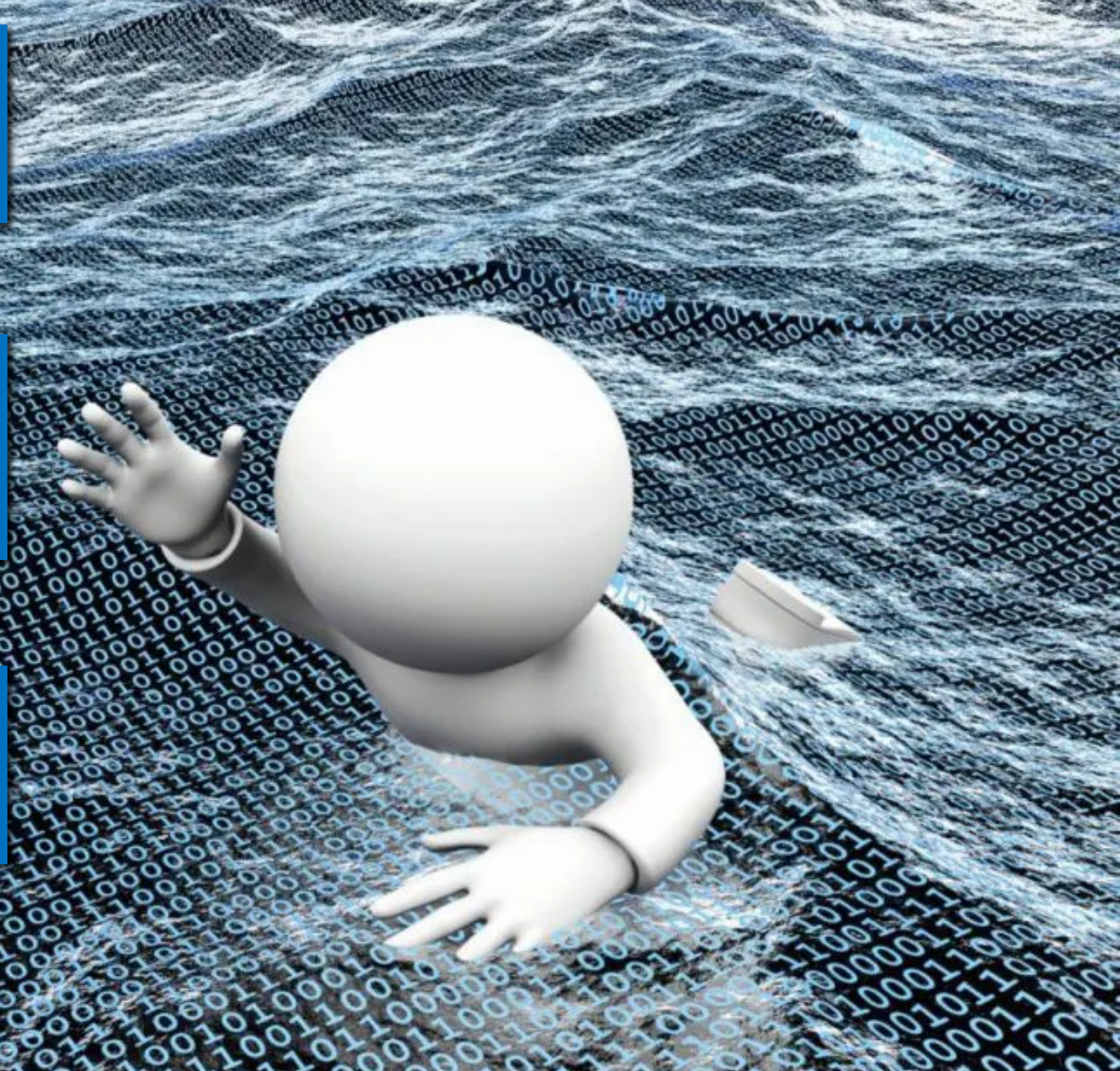
AALBORG UNIVERSITY


BIG DATA

Knowledge Graphs
(integrating heterogenous data)

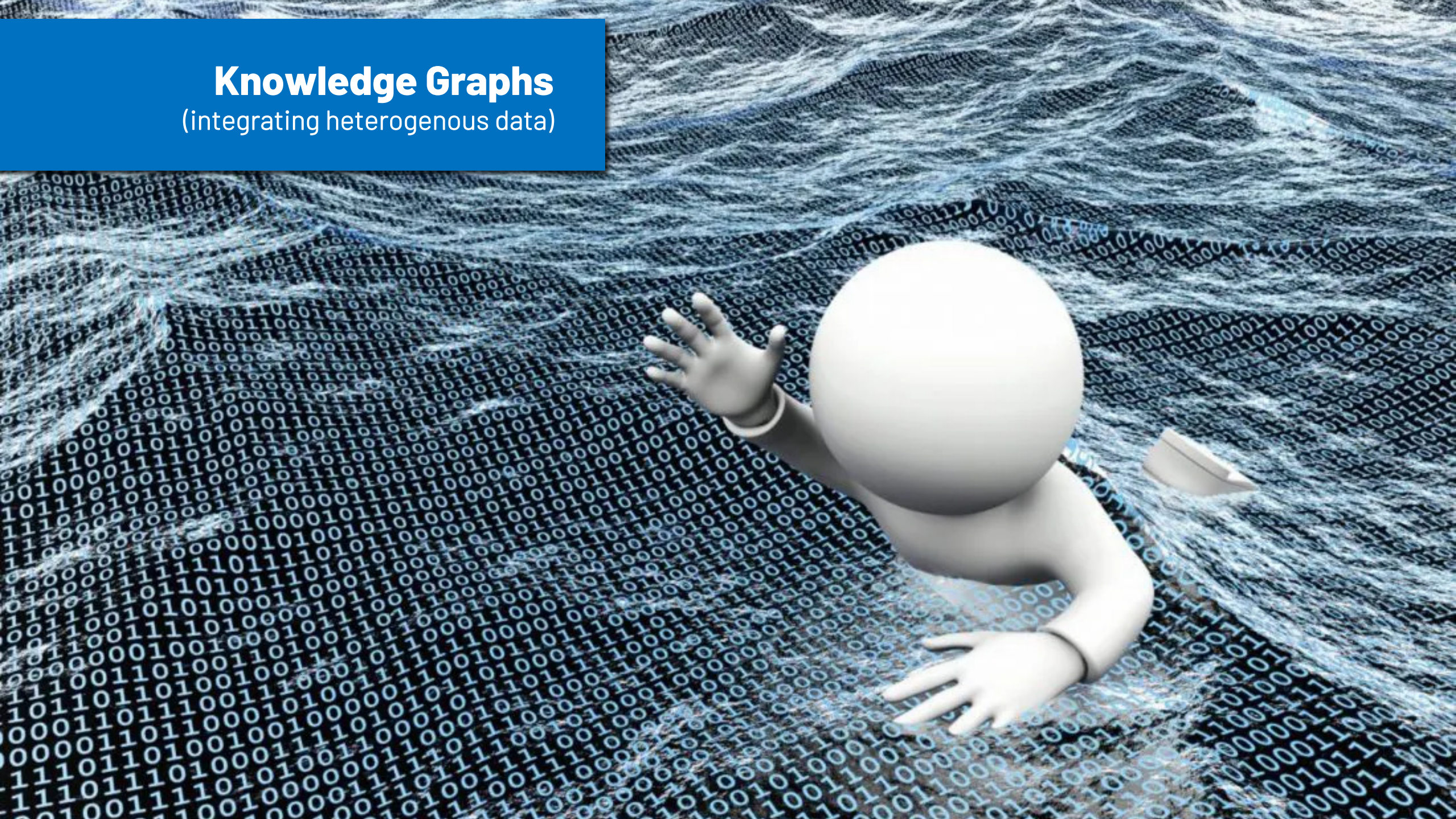
KG Search & Exploration
(exploring heterogenous data)

KG Exploration Systems
(systems to analyze heterogenous data)

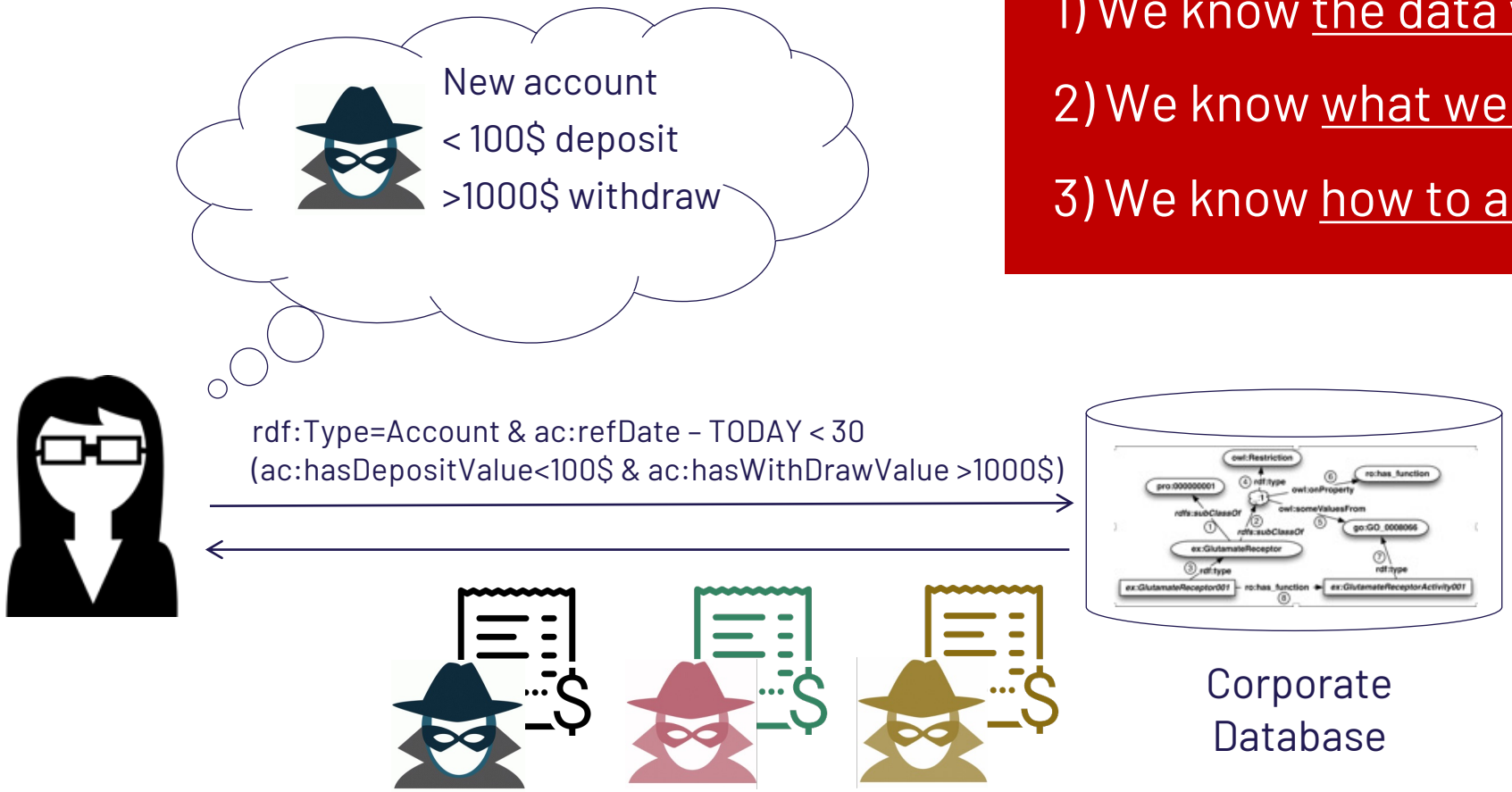


Knowledge Graphs

(integrating heterogenous data)

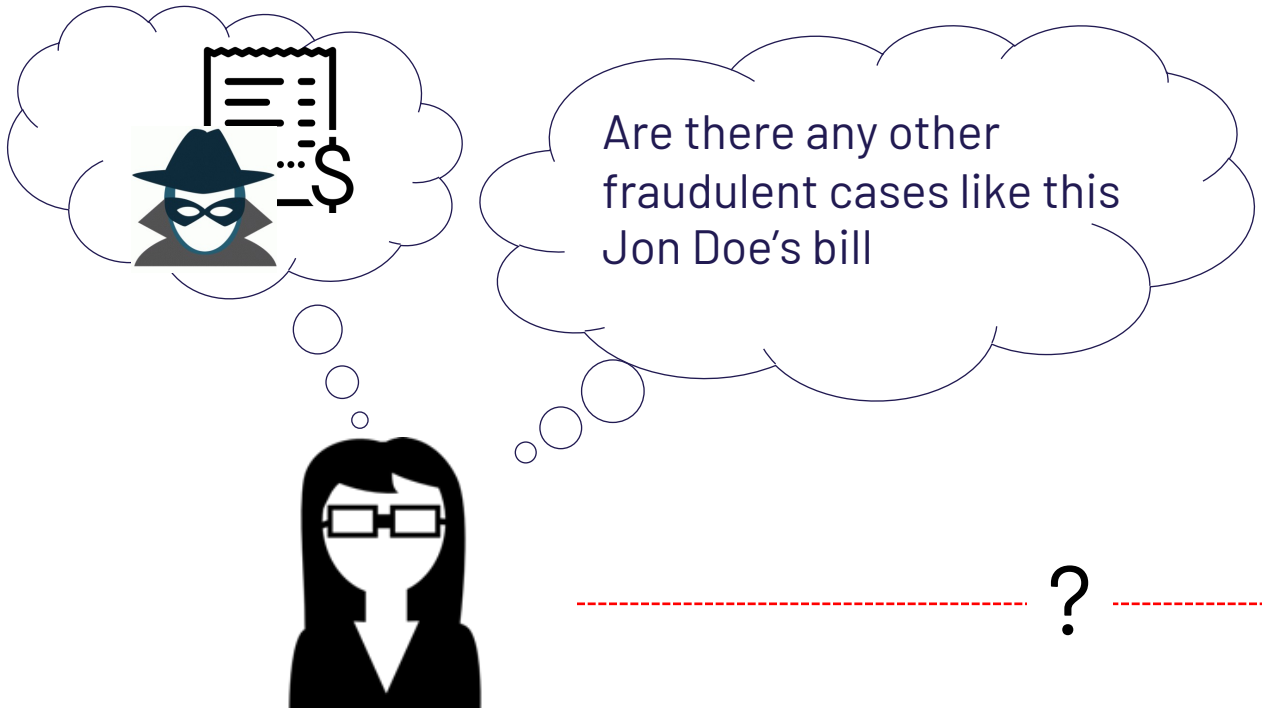


Traditional Data Management Use-case



1) We know the data we have
2) We know what we are looking for
3) We know how to ask for it

Modern Data Management Use-case



1) Not sure about the data we have

2) Not clear what we are looking for

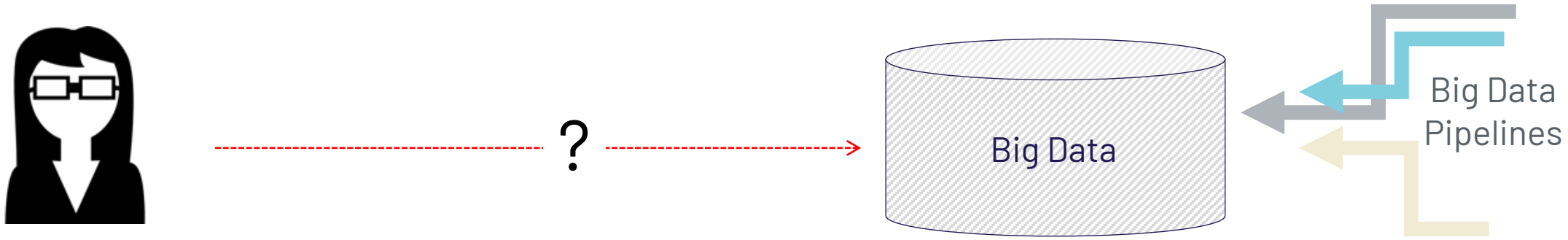
How can we understand the data we can access?
How can we describe what we are looking for?

The (Big) Data Integration Problem

The Vs of Big Data

« ...capture, store, and process the semi-structured and unstructured **(variety)** data generated with high speed (velocity), and huge in size (volume)... »

Not sure about the data we have



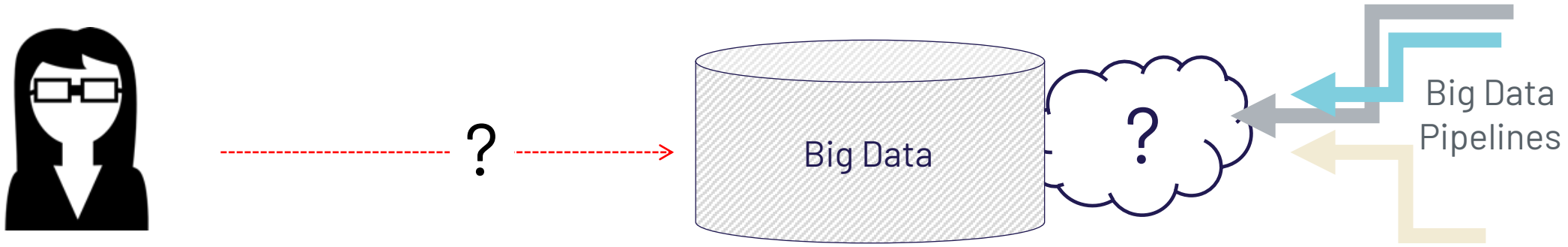
Lissandrini et al. 2022

The (Big) Data Integration Problem

The Vs of Big Data

« ...capture, store, and process the semi-structured and unstructured (variety) data generated with high speed (velocity), and huge in size (volume)... »

Not sure about the data we have



How can we integrate, model, and make sense of large volumes of data produced by many different sources ?

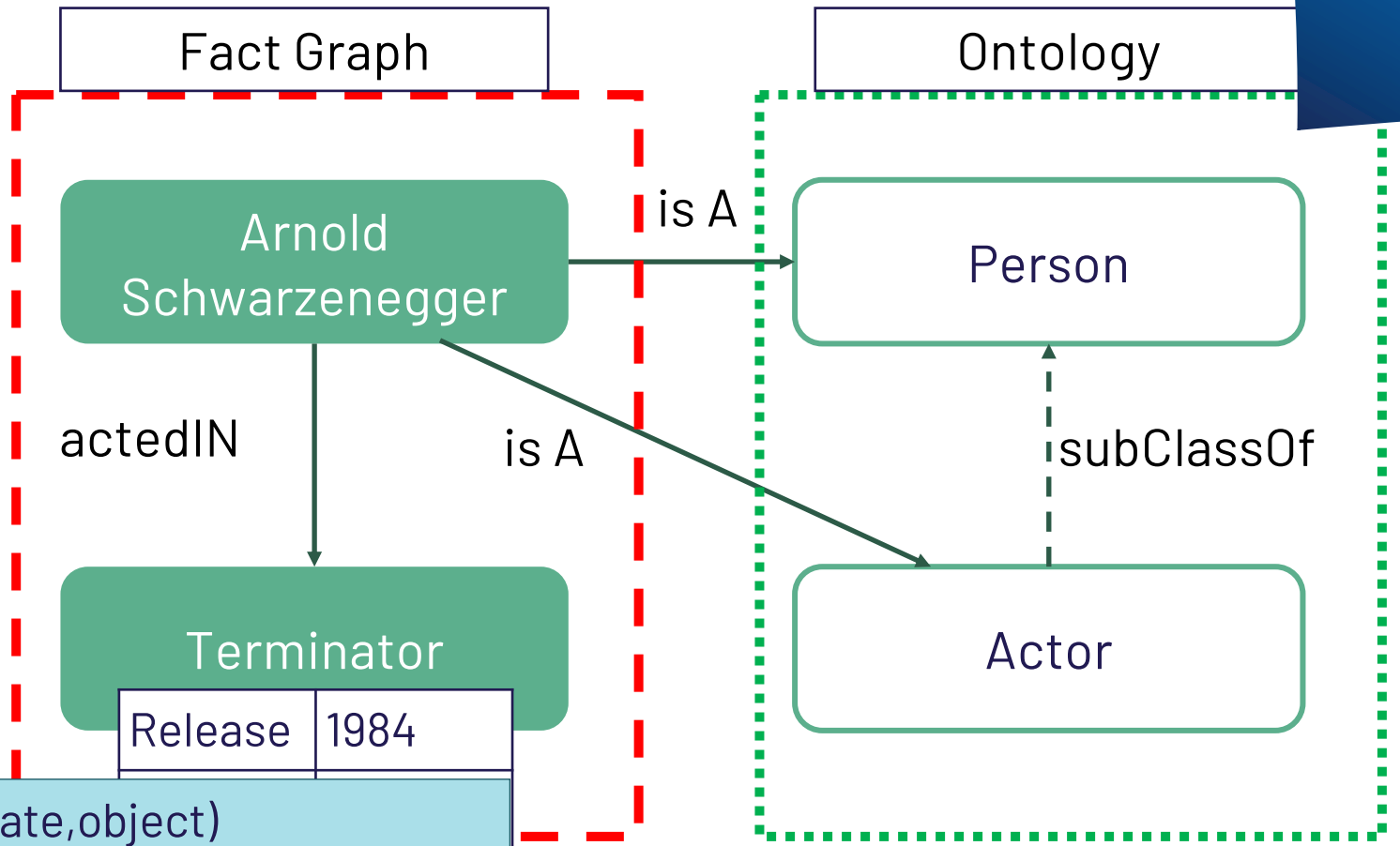
Lissandrini et al. 2022

Graphs & Knowledge Graphs

Connected Data

"A graph is a graph is a graph"

The Structure evolves & adapts to the Data



Edge-labelled Multigraphs

G: ⟨V, E, L, ℓ⟩

Attributes:

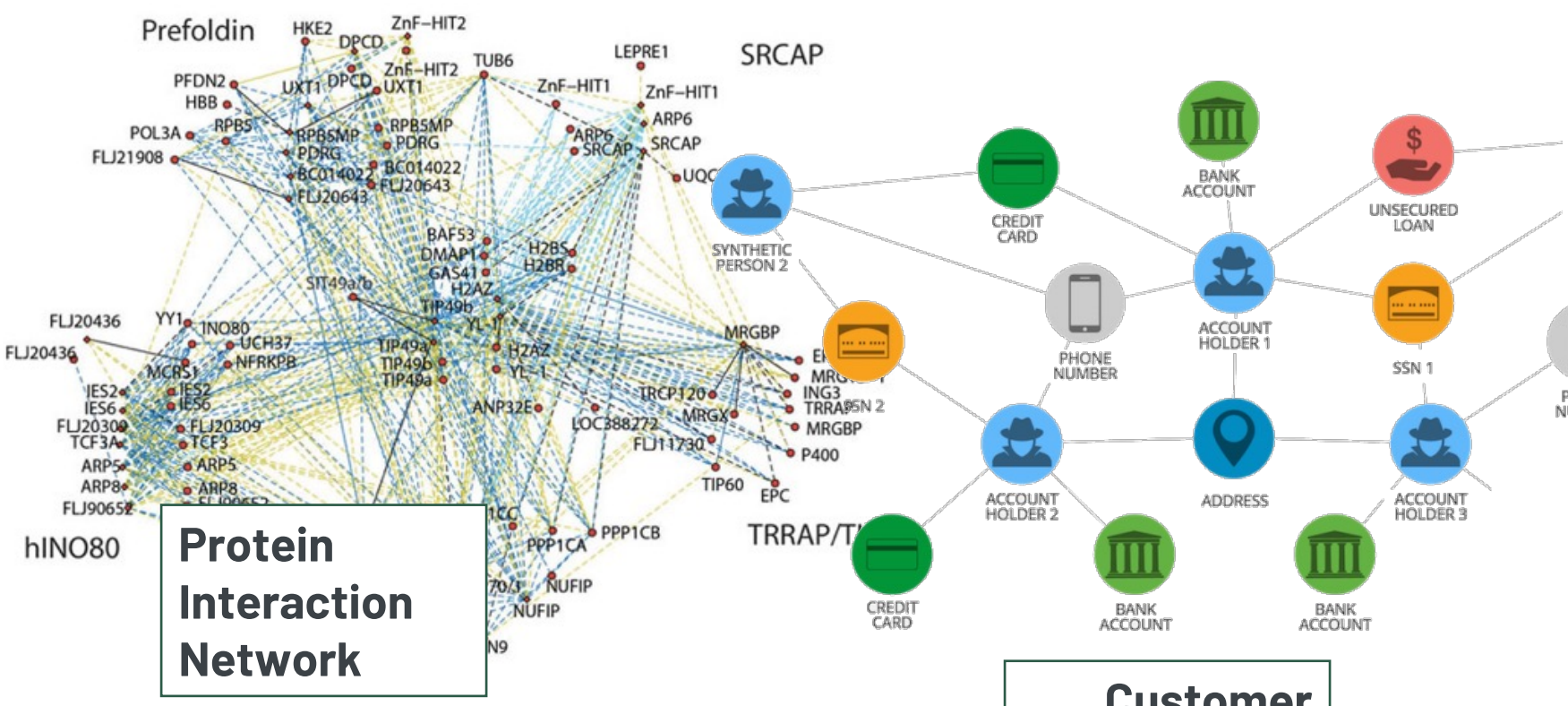
V/E:<key,value>

RDF (subject,predicate,object)

(Arnold_Schwarzenegger,isA,Person)
(Actor, subClassOf, Person)
(Arnold_Schwarzenegger, actedIn, Terminator)

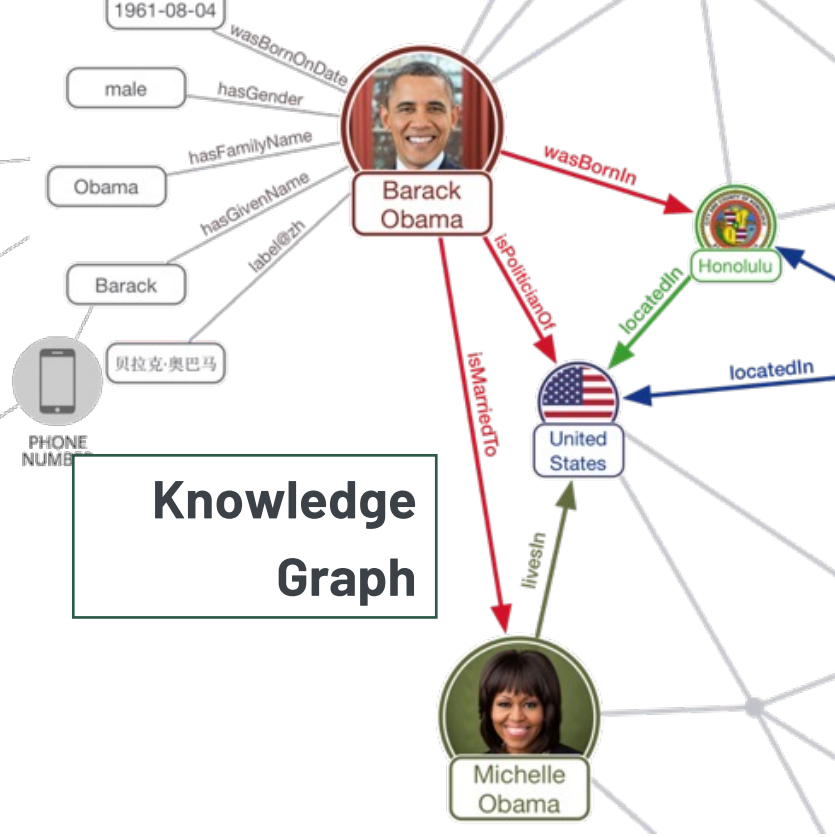
The structure of the graph is as important as the data values
Connections = Information

Lissandrini et al. 2022



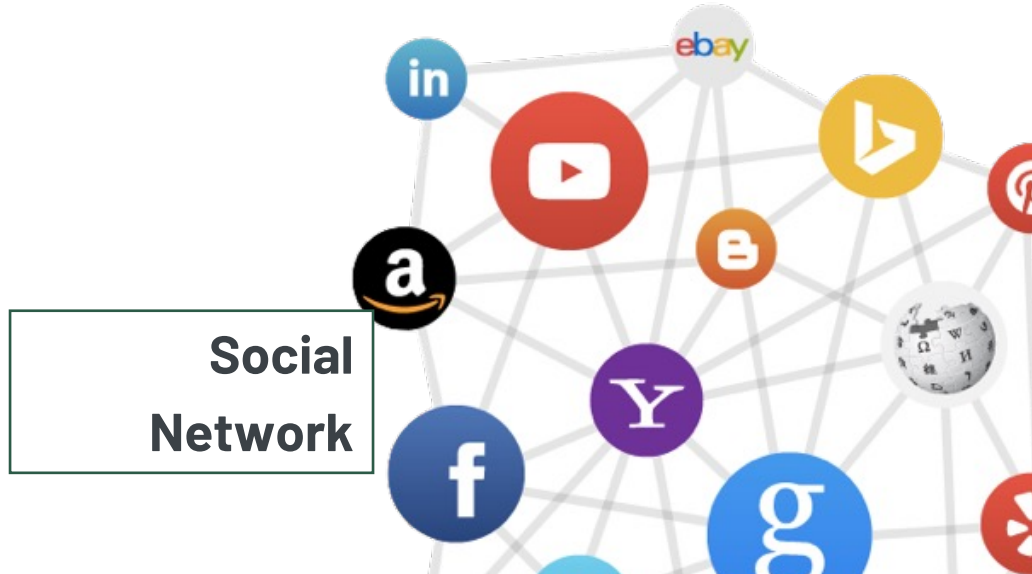
Protein Interaction Network

Customer Network



Knowledge Graph

Graphs *are* Everywhere



Social Network

Lissandrini et al. 2022

The Growing Role of Graphs & Knowledge Graphs

COMMUNICATIONS OF THE ACM

Home / Magazine Archive / August 2019 (Vol. 62, No. 8) / Industry-Scale Knowledge Graphs: Lessons and

PRACTICE

Industry-Scale Knowledge Graphs: Lessons and Challenges

By Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, Jamie Taylor
Communications of the ACM, August 2019, Vol. 62 No. 8, Pages 36-43
10.1145/3331166
Comments

Credit: Adempercem / Stutterstock

Knowledge graphs are critical to many enterprises today
provide the structured data and factual knowledge that
many products and make them more intelligent and "ma

In general, a knowledge graph describes objects of inter
connections between them. For example, a knowledge g
have nodes for a movie, the actors in this movie, the dire
so on. Each node may have properties such as an actor's
and age. There may be nodes for multiple movies involv
particular actor. The user can then traverse the knowled
to collect information on all the movies in which the act
appeared or, if applicable, directed.

Many practical implementations impose constraints on
in knowledge graphs by defining a *schema* or *ontology*. For example, a link from a movie to its direct
connect an object of type Movie to an object of type Person. In some cases the links themselves might have
their own properties: a link connecting an actor and a movie might have the name of the specific role the actor
played. Similarly, a link connecting a politician with a specific role in government might have the time period

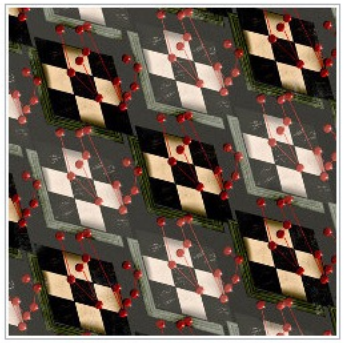
COMMUNICATIONS OF THE ACM

Home / Magazine Archive / September 2021 (Vol. 64, No. 9) / The Future Is Big Graphs: A Community View on Graph... /

CONTRIBUTED ARTICLES

The Future Is Big Graphs: A Community View on Graph Processing Systems

By Sherif Sakr, Angela Bonifati, Hannes Voigt, Alexandru Iosup, Khaled Ammar,
Arenas, Maciej Besta, Peter A. Boncz, Khuzaima Daudjee, Emanuele Della Valle,
Haslhofer, Tim Hegeman, Jan Hidders, Katja Hose, Adriana Iamnitchi, Vasiliki Ka
Özsu, Eric Peukert, Stefan Plantikow, Mohamed Ragab, Matei R. Ripeanu, Semih
Juan F. Sequeda, Joshua Shinavier
Communications of the ACM, September 2021, Vol. 64 No. 9, Pages 62-71
10.1145/3434642
Comments



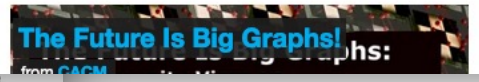
Credit: Alli Torban

Graphs are, by nature, 'unifyi
interconnectedness to repres
real- and digital-world phenc
consumers of graph instance
these abstractions, future pro
and systems. What needs to
graph processing to continue

Back to Top

Key Insights

- Graphs ar
enabling r
graph proc
every dom
- Diverse w
languages
suitable a
metrics wi
processin
decade.



COMMUNICATIONS OF THE ACM

Home / Magazine Archive / March 2021 (Vol. 64, No. 3) / Knowledge Graphs / Full Text

REVIEW ARTICLES

Knowledge Graphs

By Claudio Gutierrez, Juan F. Sequeda
Communications of the ACM, March 2021, Vol. 64 No. 3, Pages 96-104
10.1145/3418294
Comments

"Those who cannot remember the past are condemned to repeat it."
—George Santayana

Back to Top

Key Insights

- Data was traditionally considered a material object, tied to bits, with no semantics per se. Knowledge was traditionally conceived as the immaterial object, living only in people's minds and language. The destinies of data and knowledge became bound together, becoming almost inseparable, by the emergence of digital computing in

Knowledge Graph Adoption

Lissandrini et al. 2022

The entries of data sources used to construct the KG ***are continuously changing****...*

[...]

Self-serve data onboarding: Low-effort ***onboarding of new data sources*** *is important to ensure consistent growth of the KG.*

Saga: A Platform for Continuous Construction and Serving of Knowledge At Scale

Ihab F. Ilyas, Theodoros Rekatsinas, Vishnu Konda
Jeffrey Pound, Xiaoguang Qi, Mohamed Soliman
Apple

ABSTRACT

We introduce Saga, a next-generation knowledge construction and serving platform for powering knowledge-based applications at industrial scale. Saga follows a hybrid batch-incremental design to continuously integrate billions of facts about real-world entities and construct a central knowledge graph that supports multiple production use cases with diverse requirements around data freshness, accuracy, and availability. In this paper, we discuss the unique challenges associated with knowledge graph construction at industrial scale, and review the main components of Saga and how they address these challenges. Finally, we share lessons-learned from a wide array of production use cases powered by Saga.

CCS CONCEPTS

• **Computer systems organization** → *Neural networks*; **Data flow architectures**; **Special purpose systems**; • **Information systems** → **Deduplication**; **Extraction, transformation and loading**; **Data cleaning**; **Entity resolution**.

KEYWORDS

knowledge graphs, knowledge graph construction, entity resolution, entity linking

ACM Reference Format:
Ihab F. Ilyas, Theodoros Rekatsinas, Vishnu Konda, Jeffrey Pound, Xiaoguang Qi, Mohamed Soliman. 2022. Saga: A Platform for Continuous Construction and Serving of Knowledge At Scale. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3514221.3526049

1 INTRODUCTION

Accurate and up-to-date knowledge about real-world entities is needed in many applications. Search and assistant services require open-domain knowledge to power question answering. Other applications need rich entity data to render entity-centric experiences. Many internal applications in machine learning need training data sets with information on entities and their relationships. All of these applications require a broad range of knowledge that is accurate and continuously updated with facts about entities.

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

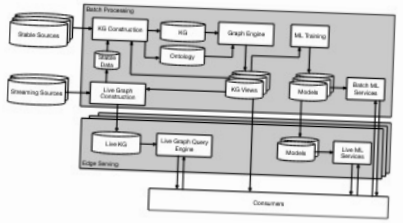


Figure 1: Overview of the Saga knowledge platform.

Constructing a central knowledge graph (KG) that can serve these needs is a challenging problem, and developing a KG construction and serving solution that can be shared across applications has obvious benefits. This paper describes our effort in building a next-generation knowledge platform for continuously integrating billions of facts about real-world entities and powering experiences across a variety of production use cases.

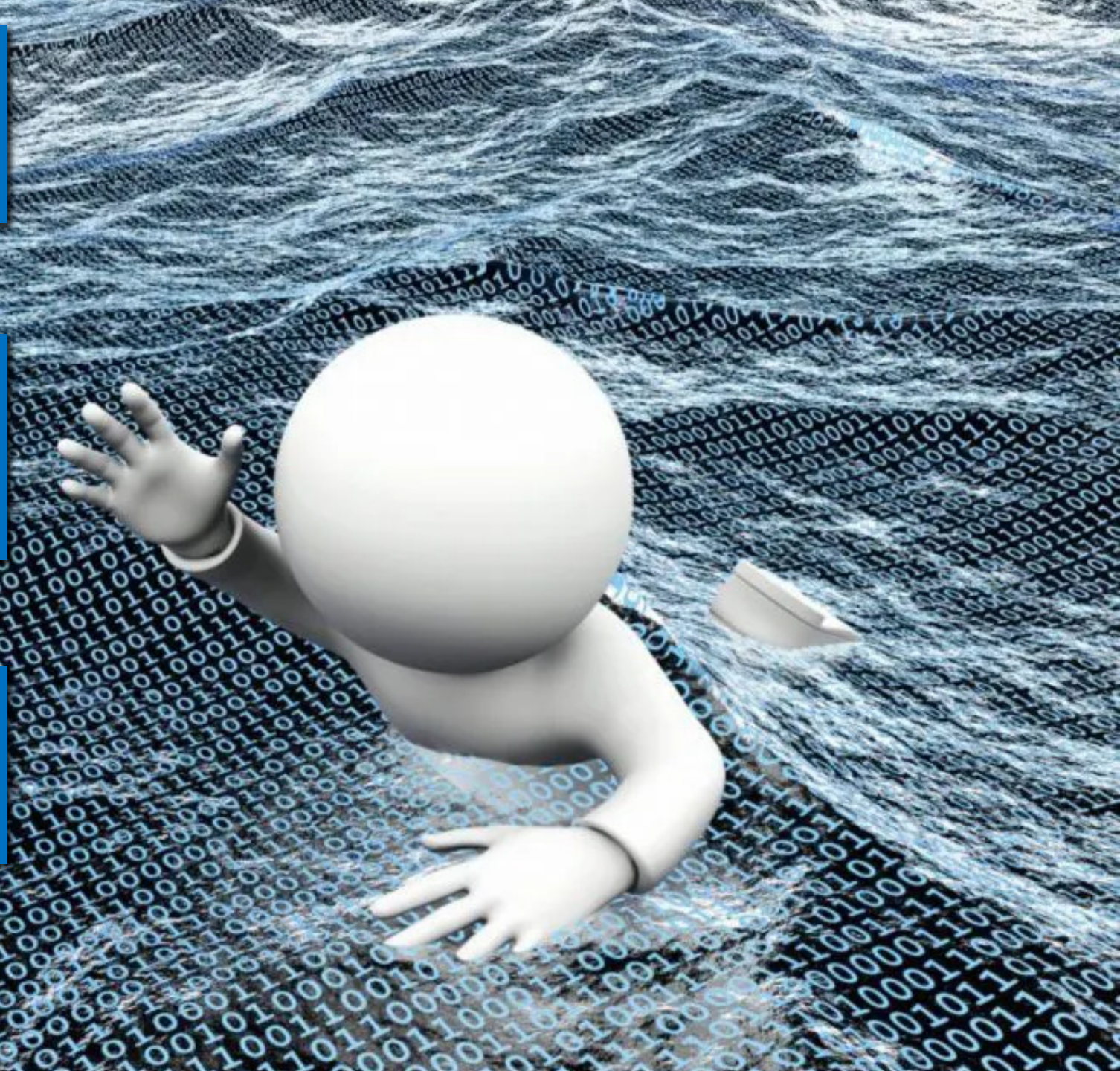
Knowledge can be represented as a graph with edges encoding *facts* amongst *entities* (nodes) [61]. Information about entities is obtained by integrating data from multiple structured databases and data records that are extracted from unstructured data [19]. The process of cleaning, integrating, and fusing this data into an accurate and canonical representation for each entity is referred to as *knowledge graph construction* [80]. Continuous construction and serving of knowledge plays a critical role as access to up-to-date and trustworthy information is key to user engagement. The entries of data sources used to construct the KG are continuously changing: new entities can appear, entities might be deleted, and facts about existing entities can change at different frequencies. Moreover, the set of input sources can be dynamic. Changes to licensing agreements or privacy and trustworthiness requirements can affect the set of admissible data sources during KG construction. Such data feeds impose unique requirements and challenges that a knowledge platform needs to handle:

(1) *Hybrid batch and stream construction*: Knowledge construction requires operating on data sources over heterogeneous domains. The update rates and freshness requirements can differ across sources. Updates from streaming sources with game scores need to be reflected in the KG within seconds but sources that focus on verticals such as songs can provide batch updates with millions of entries on a daily basis. Any platform for constructing and serving knowl-

Knowledge Graphs
(integrating heterogenous data)

KG Search & Exploration
(exploring heterogenous data)

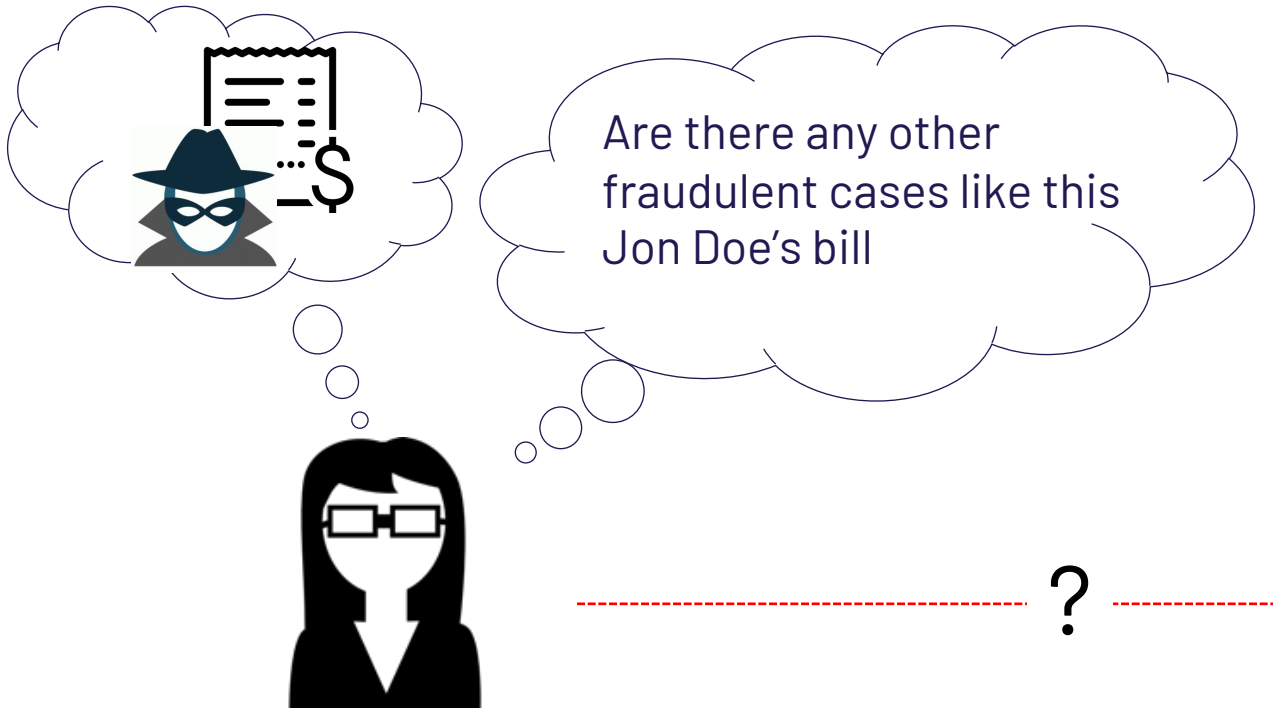
KG Exploration Systems
(systems to analyze heterogenous data)



KG Search & Exploration

(exploring heterogenous data)

Modern Data Management Use-case



1) Not sure about the data we have

2) Not clear what we are looking for

How can we understand the data we can access?
How can we describe what we are looking for?

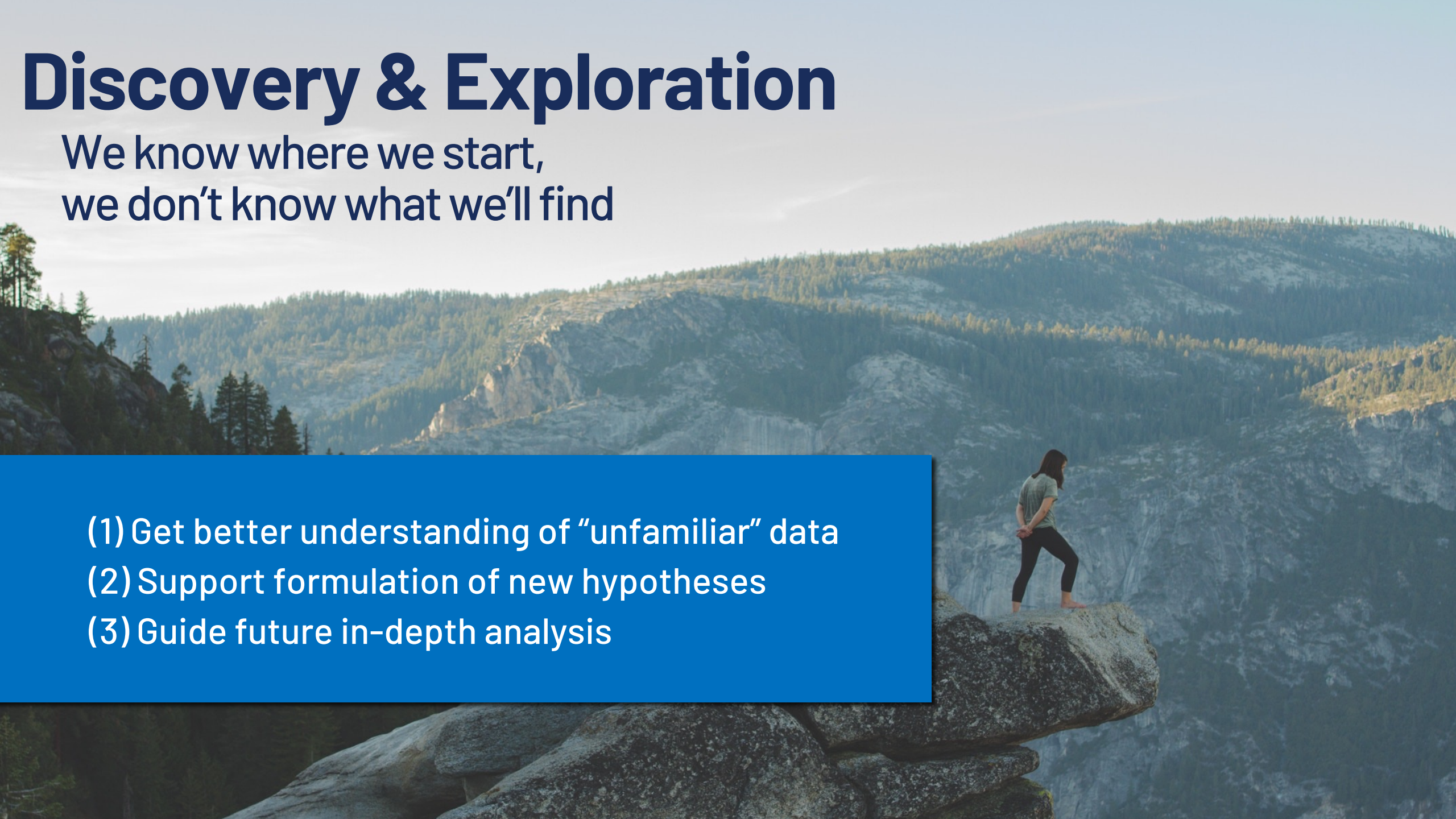
Discovery & Exploration

We know where we start, we don't know what we'll find

The process of identifying relevant datasets and the gradual understanding of their contents

Discovery & Exploration

We know where we start,
we don't know what we'll find



(1) Get better understanding of "unfamiliar" data
(2) Support formulation of new hypotheses
(3) Guide future in-depth analysis

Data Exploration Needs

What information do we have about accounts ? Where are they stored? How many accounts are on record ?

Summarization &Profiling

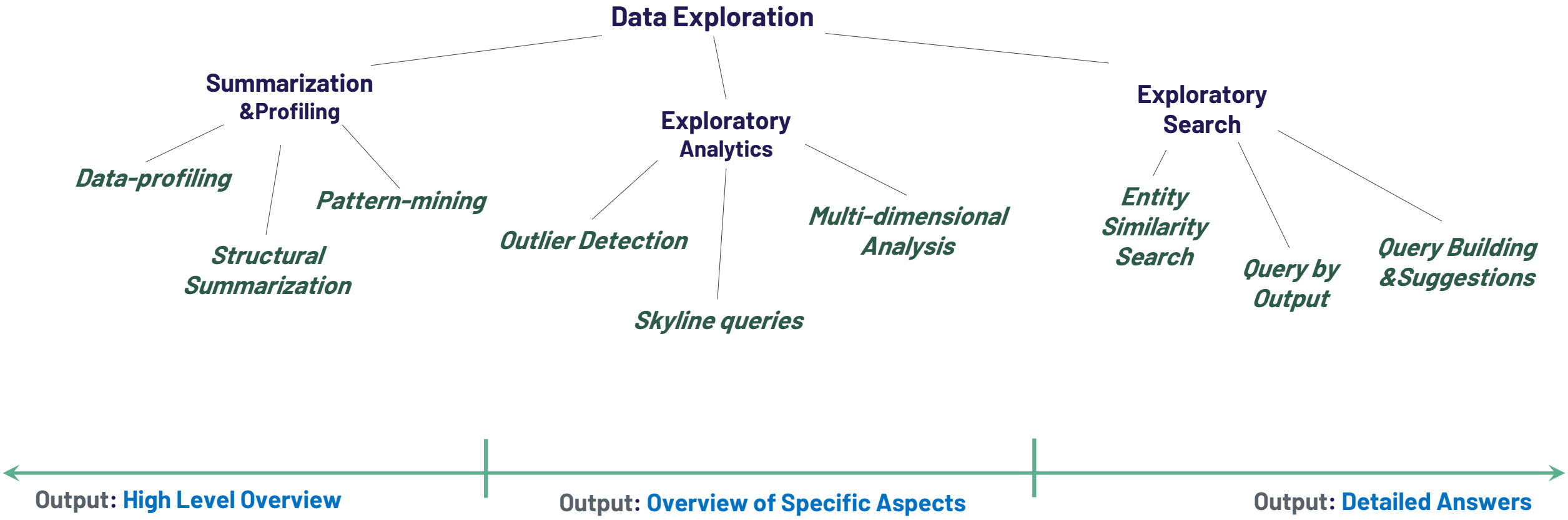
What accounts have abnormal amounts of withdrawals? What is the average account lifespan?

Exploratory Analytics

Are there fraudulent accounts similar to this?

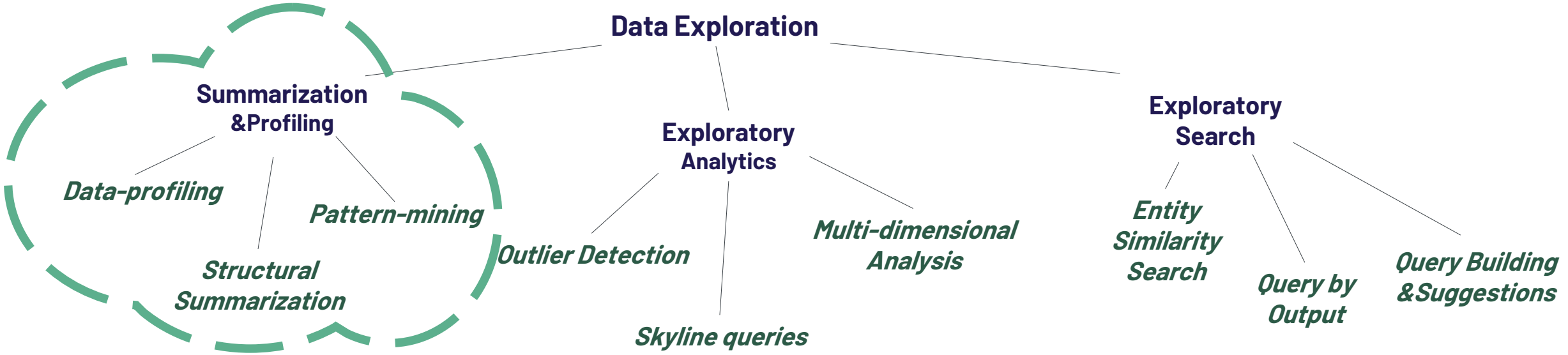
Exploratory Search

Data Exploration Methods



Lissandrini et al. 2022

Data Exploration Methods



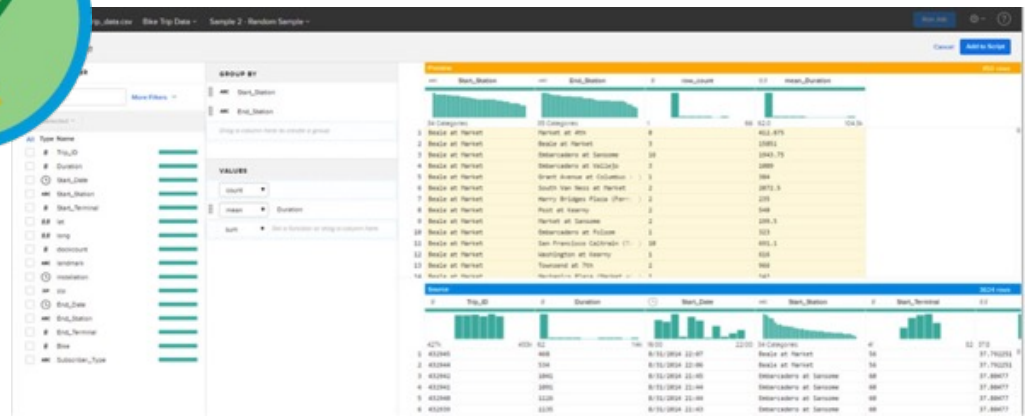
No Interaction and Personalization
Requires: **No Domain Knowledge**
Output: **High Level Overview**

Medium-High Interactivity
Requires: **High-level Information Need**
Output: **Overview of Specific Aspects**

High Interactivity and Personalization
Requires: **Detailed Sample or Query Intent**
Output: **Detailed Answers**

Lissandrini et al. 2022

Data Preparation & Visualization software



Trifacta: data preparation

OpenRefine: data preparation and cleanup

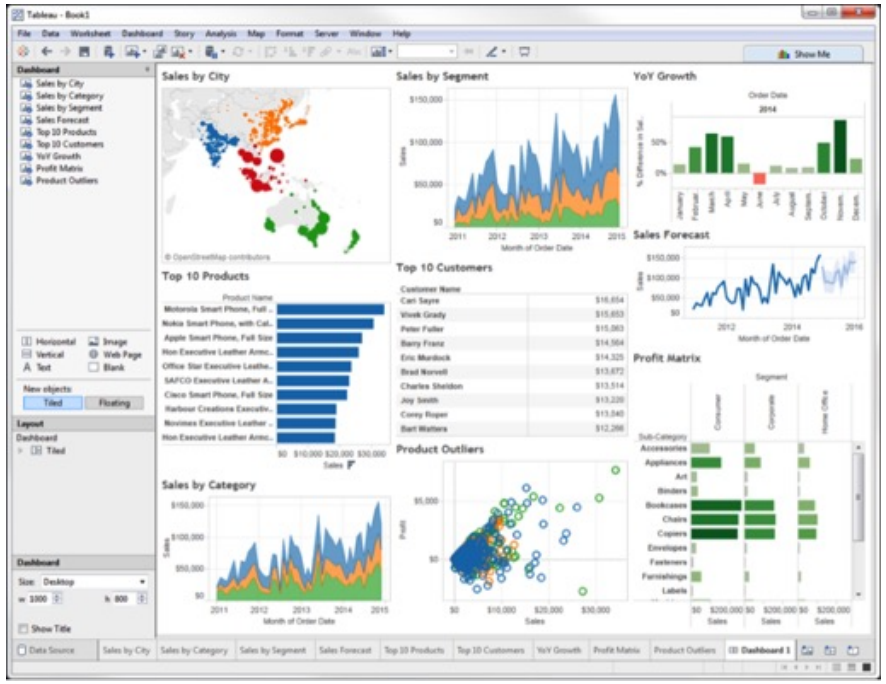


Tableau: visual analysis and statistics

They provide an “overview” but how can we “dig into” the data ?

KG Profiling

Heiko Paulheim [2017]
http://ceur-ws.org/Vol-1927/paper1.pdf

Obtain a basic understanding of the contents of a KG

1. How many instances? How many classes?
2. What's the vocabulary (predicates/attributes)
3. Are there big-hubs? Are there disconnected islands?

Table 1: Global Properties of the Knowledge Graphs compared in this paper

	DBpedia	YAGO	Wikidata	OpenCyc	NELL
Version	2016-04	YAGO3	2016-08-01	2016-09-05	08m.995
# instances	5,109,890	5,130,031	17,581,152	118,125	1,974,297
# axioms	397,831,457	1,435,808,056	1,633,309,138	2,413,894	3,402,971
avg. indegree	13.52	17.44	9.83	10.03	5.33
avg. outdegree	47.55	101.86	41.25	9.23	1.25
# classes	754	576,331	30,765	116,822	290
# relations	3,555	93,659	11,053	165	1,334
Releases	biyearly	> 1 year	live	> 1 year	1-2 days

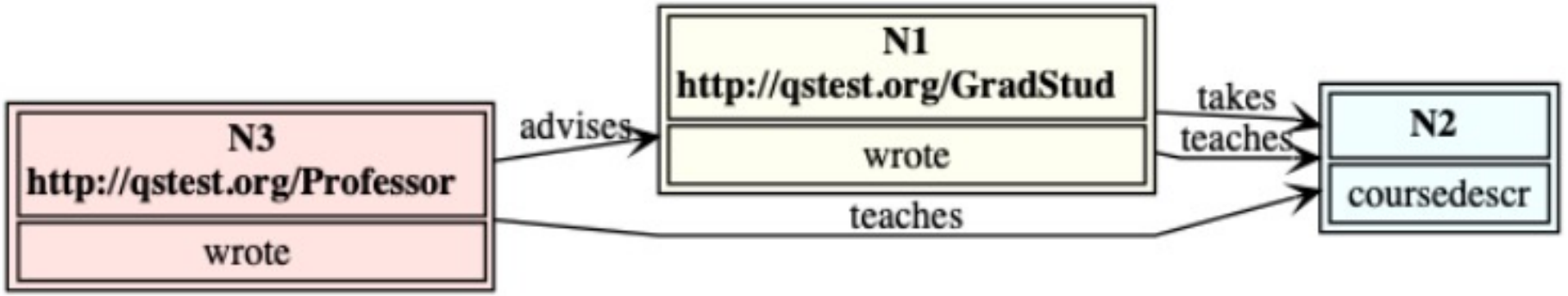
Lissandrini et al. 2022

KG Summarization & Pattern Mining

Surveys:
Kellou-Menouer et al [2022]
Čebirić et al [2019]

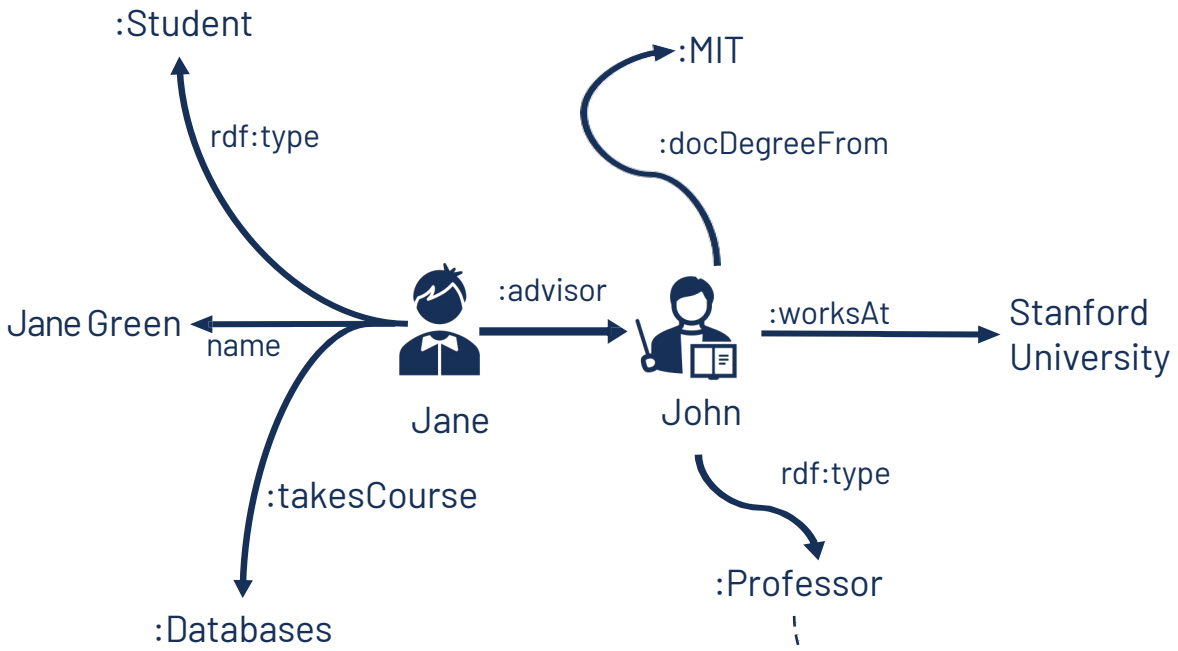
Extract overall structural information

1. How are classes connected?
2. Which predicates and attributes are shared by entities of this type?
3. What is the prevalence of connections across nodes with these properties?



KG Schema Extraction

Extract "the schema" of entity types



Validating Shapes

```
:ProfessorShape
   a                  sh:NodeShape ;
   sh:targetClass ex:Professor ;

   sh:property    [ sh:NodeKind sh:IRI ;
                    sh:path     ex:docDegreeFrom ;
                    sh:maxCount        1 ;
                    sh:minCount        1 ;   ] ;


   sh:property    [ sh:NodeKind sh:IRI ;
                    sh:path     ex:teacherOf ;
                    sh:maxCount      1 ;
                    sh:minCount      1 ;] ;

   sh:property    [ sh:NodeKind sh:IRI ;
                    sh:path     ex:worksAt ;
                    sh:node        :University;
                    sh:maxCount      1 ; ] .
```

Professor SHACL Shape

Under Submission

KG Schema Extraction (results)

Extract "the schema" of entity types & their connections

KG Statistics

	DBpedia	LUBM	YAGO-4	Wdt15	Wdt21
# of triples	52 M	91 M	210 M	290 M	1.926 B
# of objects	19 M	12 M	126 M	64 M	617 M
# of subjects	15 M	10 M	5 M	40 M	196 M
# of literals	28 M	5.5 M	111 M	40 M	904 M
# of instances	5 M	1 M	17 M	3 M	91 M
# of classes	427	22	8,902	13,227	82,693
# of properties	1,323	20	153	4,906	9,017
Size in GBs	6.6	15.66	28.59	42	234

Shapes Extracted

	NS	PS	Non-Literal PSc	Literal PSc
	COUNT	COUNT/AVG	COUNT/AVG	COUNT/AVG
LUBM	23	164 / 7.1	323 / 3.0	57 /1.0
DBpedia	426	11,916 / 27.9	38,454 / 6.9	5,335 /1.0
YAGO-4	8,897	76,765 / 8.6	315,413 / 14.5	50,708 / 1.0
Wdt15	13,227	202,085 / 15.2	114,890 / 3.0	106,599 / 1.0
Wdt21	82,651	2,051,538 / 24.8	3,765,953 / 5.6	1,113,856 / 1.0

Node Types

Property Types

Extracted Constraints

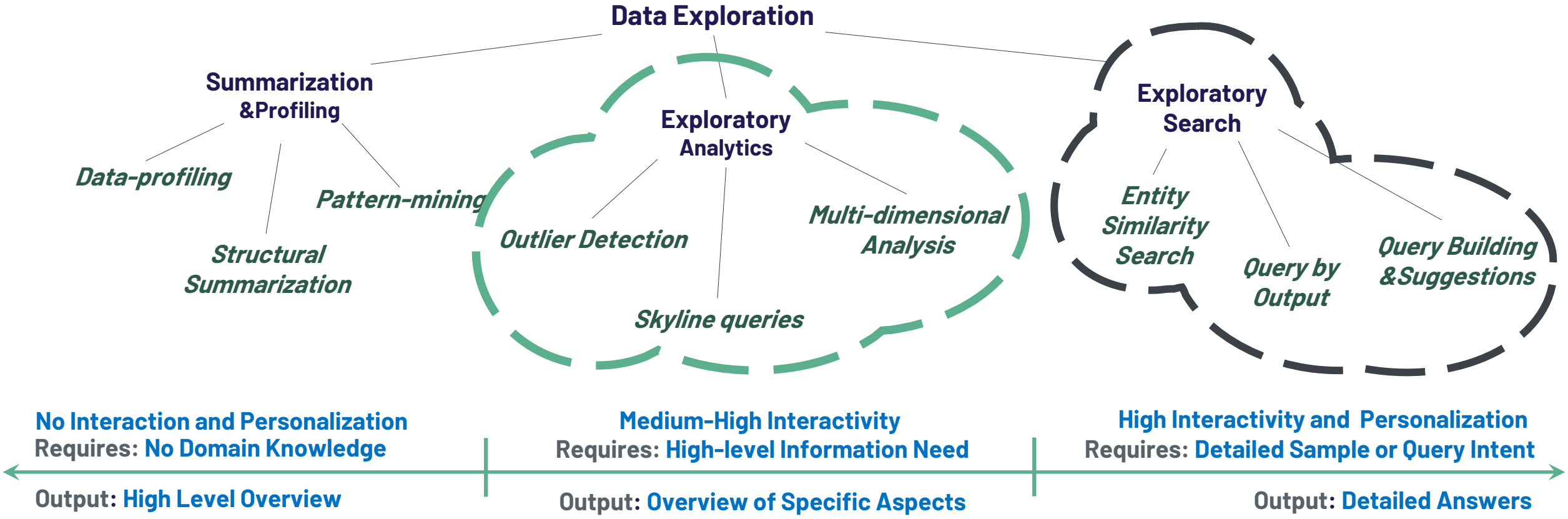
Table 3: Running Time (T) in minutes (m) and hours (h) along with Memory (M) consumption in GB and timeout 🕑.

		DBpedia		LUBM		YAGO-4		Wdt15		Wdt21	
		T	M	T	M	T	M	T	M	T	M
F	SheXer	26 m	18	58 m	33	1.9 h	24	3.2 h	59	-	Out_M
	QSE-Exact	3 m	16	8 m	16	23 m	16	16 m	50	2.5 h	235
	QSE-Approx	**1 m**	10	**2 m**	10	**13 m**	10	**13 m**	16	**1.3 h**	32
Q	SheXer	9 h	65	15 h	140	🕑	-	13 h	180	🕑	-
	QSE-Exact	34 m	16	47 m	16	2.4 h	16	1.2 h	16	🕑	-
	QSE-Approx	**16 m**	6	**3 m**	7	**39 m**	16	**49 m**	16	**5.7 h**	64

Good News:

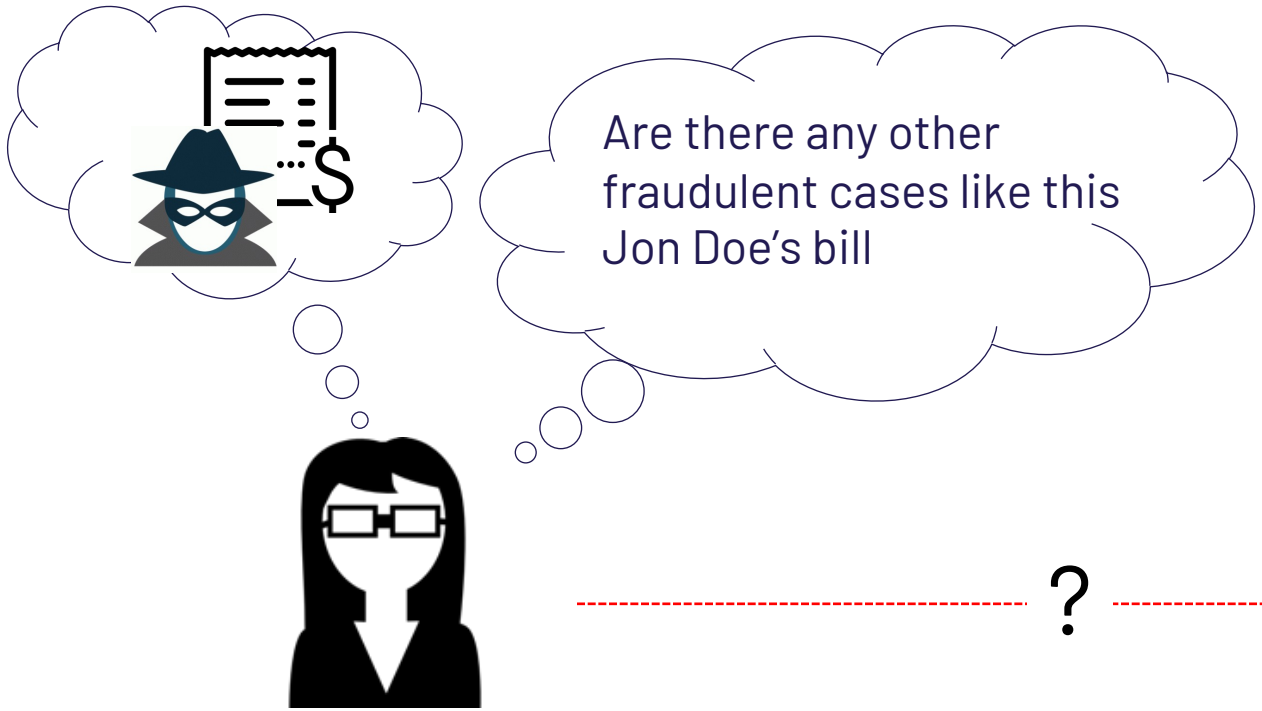
Scalable Approximate Methods!

Data Exploration Methods



Lissandrini et al. 2022

Modern Data Management Use-case



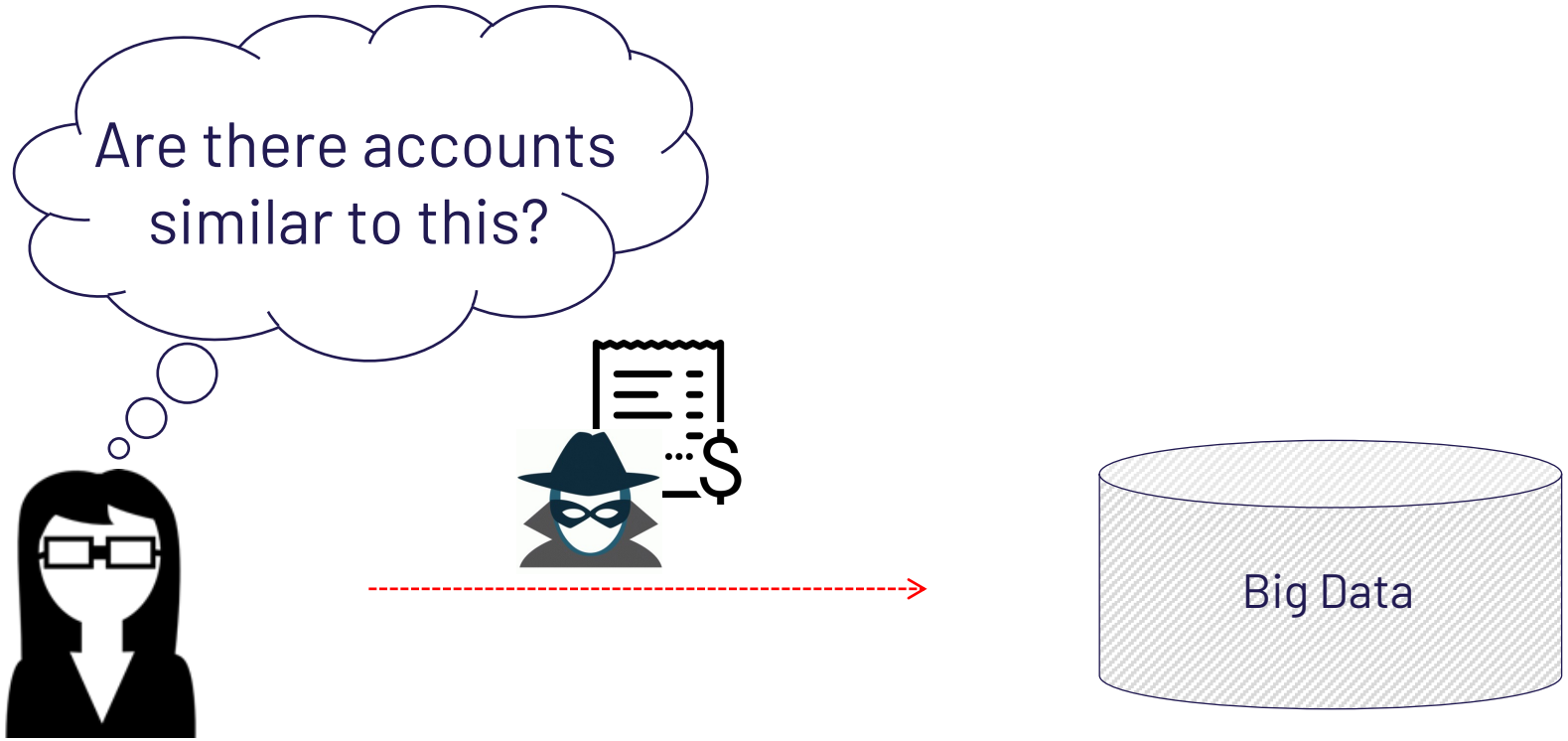
1) Not sure about the data we have

2) Not clear what we are looking for

How can we describe what we are looking for?

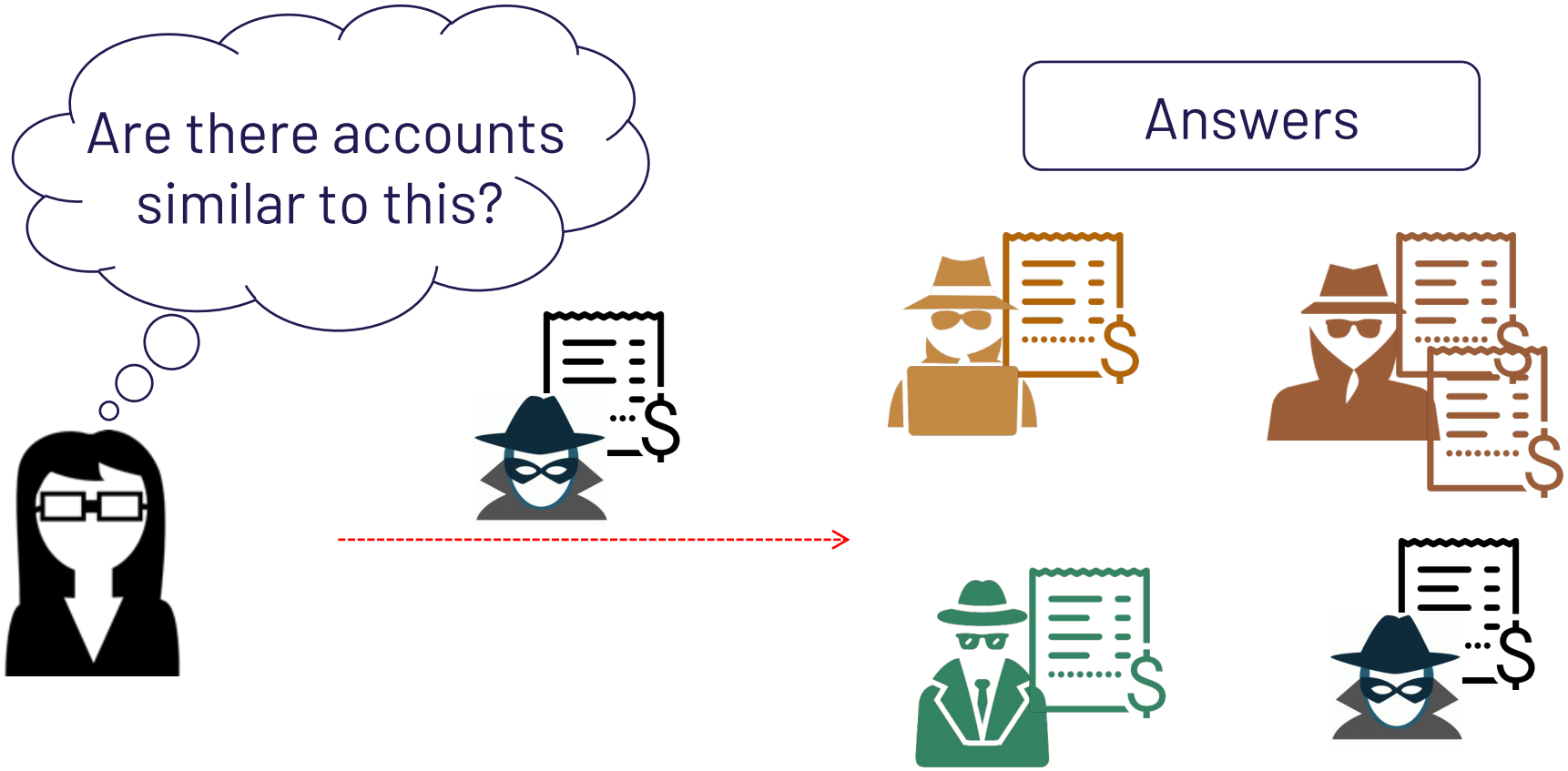
Lissandrini et al. 2022

Examples “as” Exploratory Methods



Example is always more efficacious than precept
Samuel Johnson, Rasselas (1759)

Examples “as” Exploratory Methods



Example is always more efficacious than precept

Samuel Johnson, Rasselas (1759)

Lissandrini et al. 2022

Similarities are the key . . .

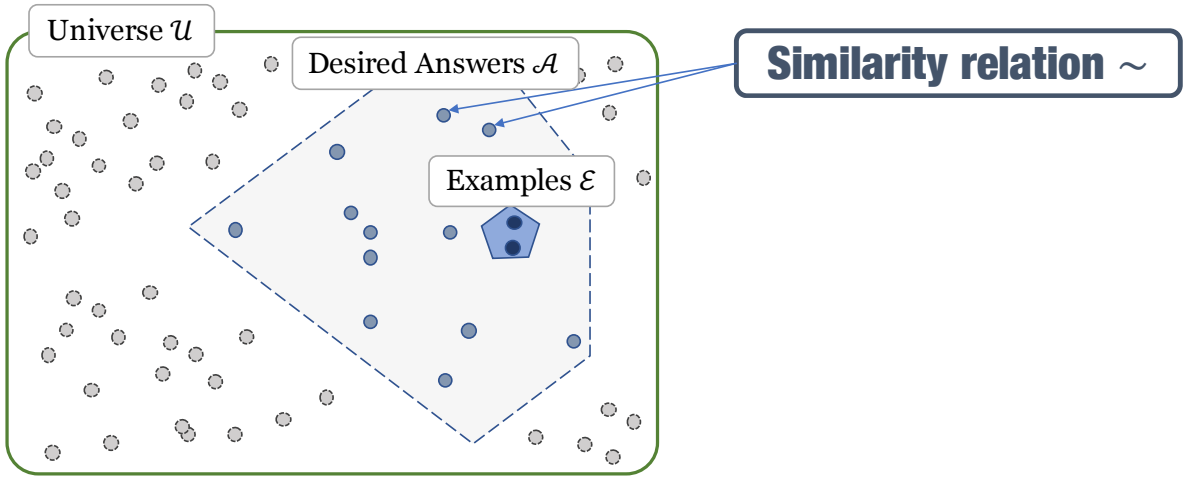
If we knew how similar each item is with respect to any other for each user, we would know the answer

Lissandrini et al. 2022

The Example-based problem

Given

a set of examples $\mathcal{E}$ from a universe $\mathcal{U}$



Find

a similarity " $\sim$ "

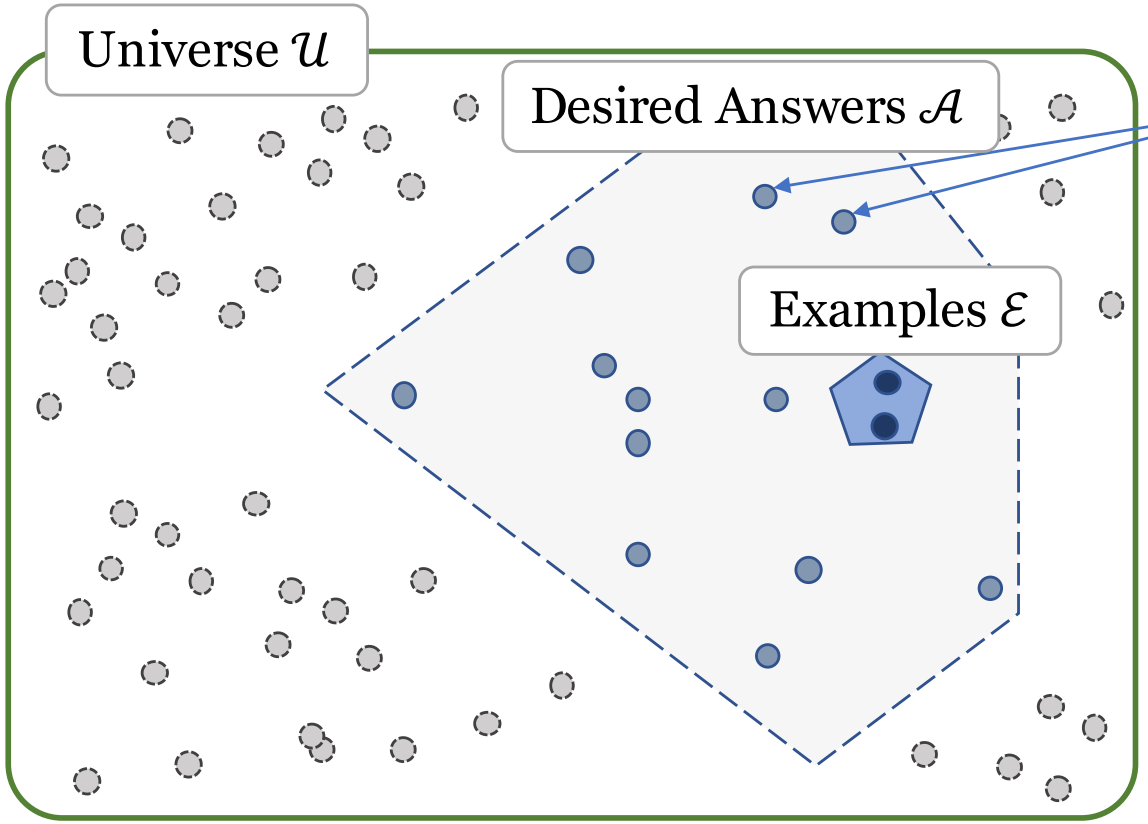
such that

1. When $\mathcal{E}$ is part of the answers $\mathcal{A}$ (partially or totally)
2. The answers in $\mathcal{A}$ are the most similar to the examples in $\mathcal{E}$ according to " $\sim$ "

What similarity " $\sim$ " should we use ?
How do we identify " $\sim$ " (for each user) ?

Lissandrini et al. 2022

Example-based methods

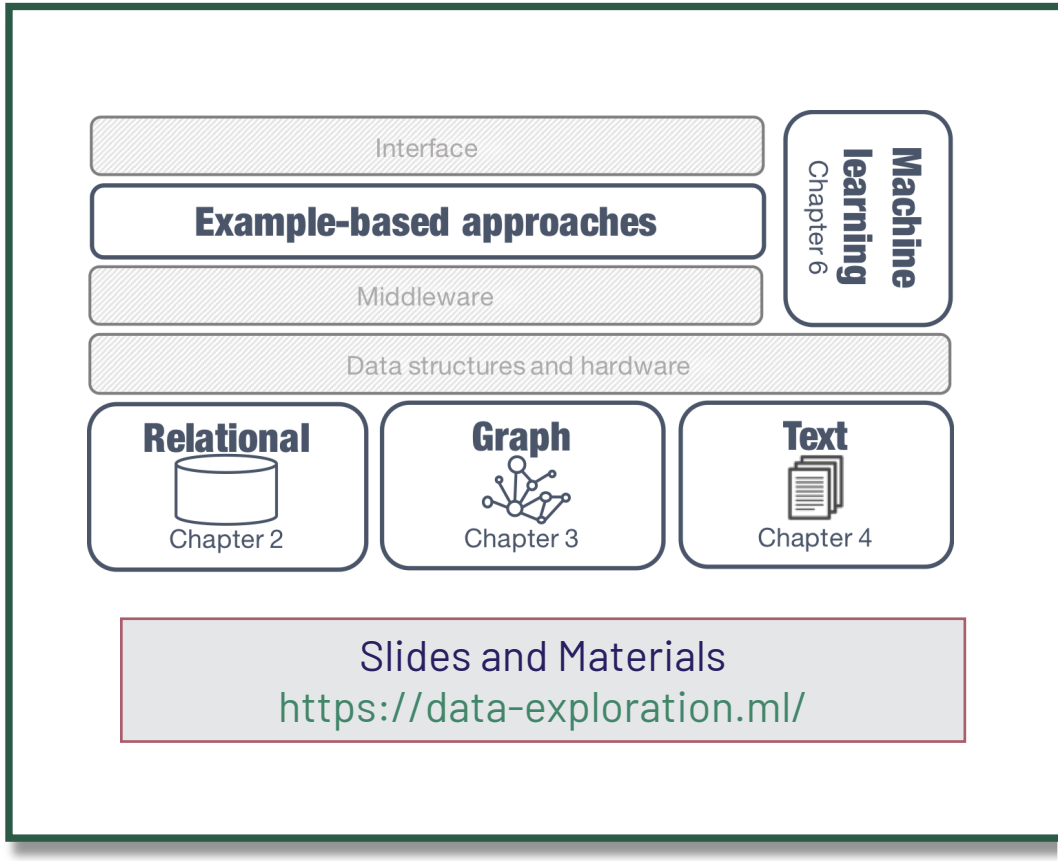


Similarity relation $\sim$

Implicit (Unknown)	Explicit (Known)
Query Reverse Engineering	**Structural Similarity**
Rule Discovery	**Proximity Search**
Relation Extraction	**Content Matching**

Lissandrini et al. 2022

Example-Based Exploration



Slides and Materials
https://data-exploration.ml/

VLDB 2017 Munich · Germany

SIGMOD/PODS AMSTERDAM 2019

SIGIR 2019

ESWC 2020

Graph Exemplar Queries

Search for Structures

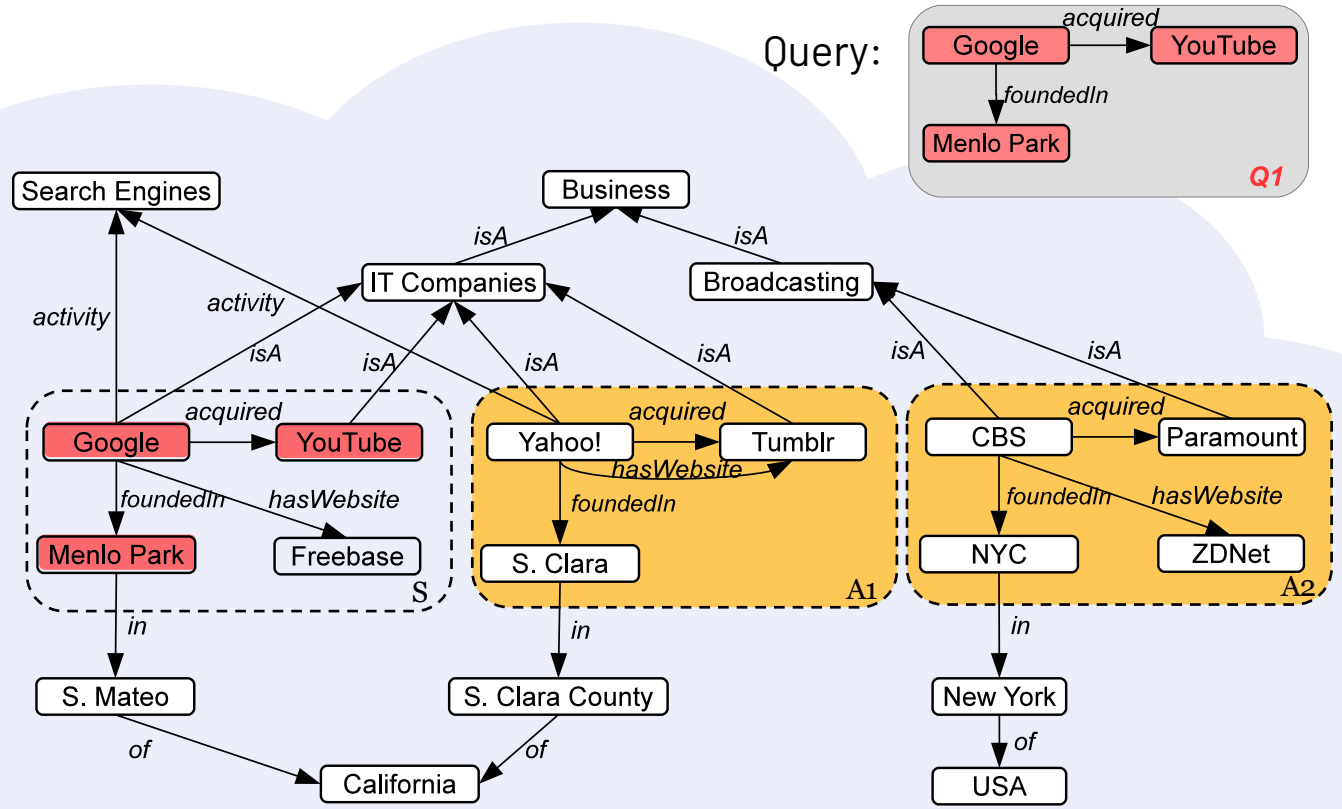
Mottin et al. [2016]

Model: Knowledge Graph

Query: Example Structure

Similarity: Isomorphism/Simulation

Output: A set of Sub-Graphs



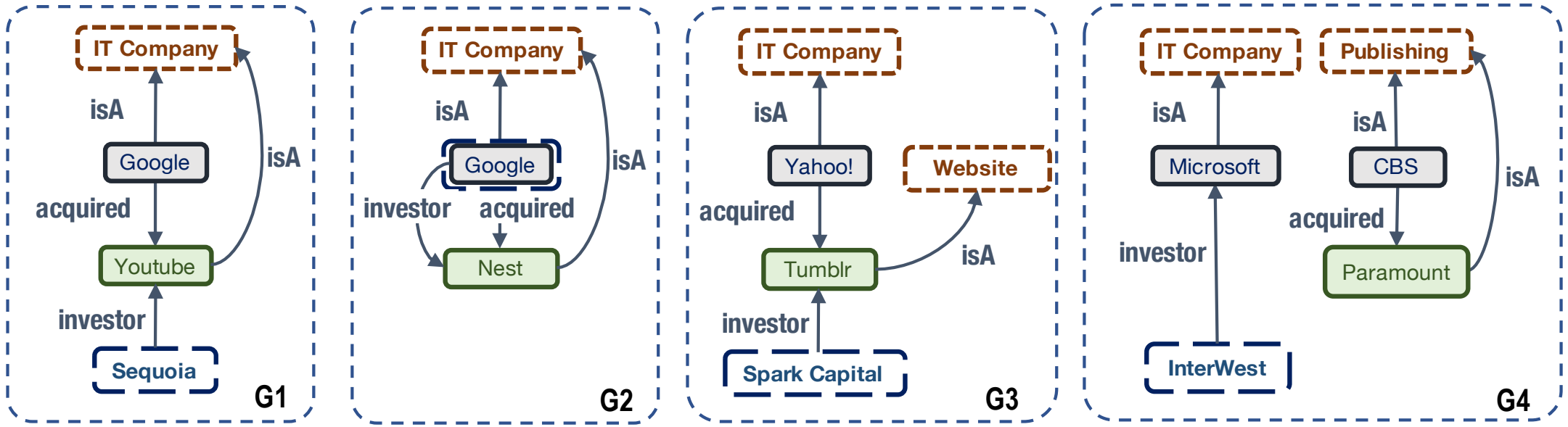
Knowledge Graph

Case: Rich Schema ⟶ Find complex structures

Graph Isomorphism vs. Simulation Variants

Structural Congruence/Similarity

Isomorphism requires an bijective function
Simulation requires only a surjective relation
Preserves only Parent → Child relationships



Example of Simulating (G1~ {G2,G3,G4}) and Strong-simulating Graphs (G1≈G2)

Strong Simulation preserves close connectivity

Strong simulation: Capturing topology in graph pattern matching – Shuai Ma et al., 2014

Computing Exemplar Queries (i)

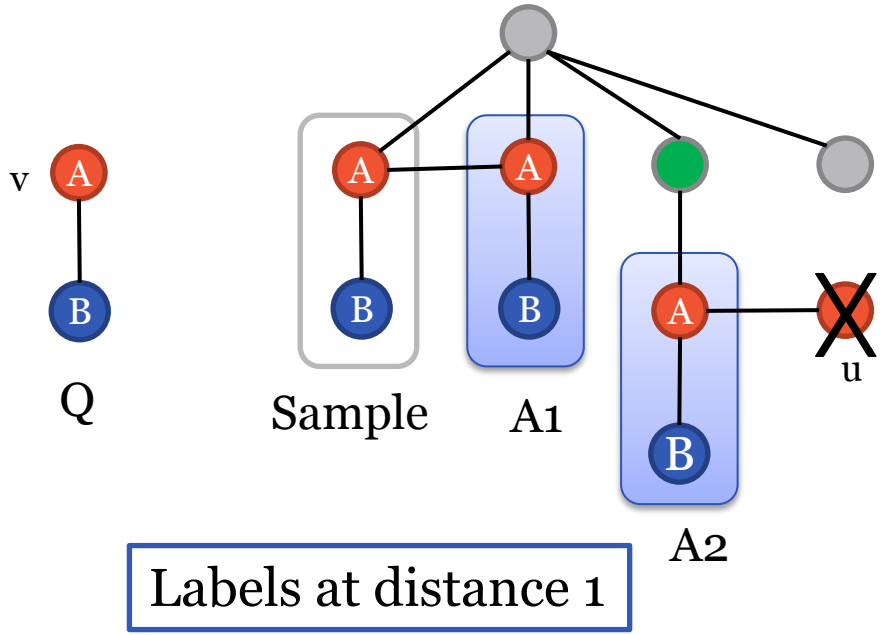
Fast Structure Matching

Mottin et al. [2016]

NP-complete (subgraph isomorphism)

$O(|V|^4)$ (simulation)

Reduce Search Space:
Removes nodes that cannot be part of a solution



Exact Pruning technique:

- Compute the neighbor labels of each node

$$W_{n,a,i} = \{n_1 | l(n_1, n_2) = a \vee \in N_{i-1}(n)\}$$

- Prune nodes not matching query nodes neighborhood labels
- Apply iteratively on the query nodes

neighborhood (v) = {(B,1)}
$\notin$
neighborhood (u) = {(A,1)}

No Match

Computing Exemplar Queries (ii)

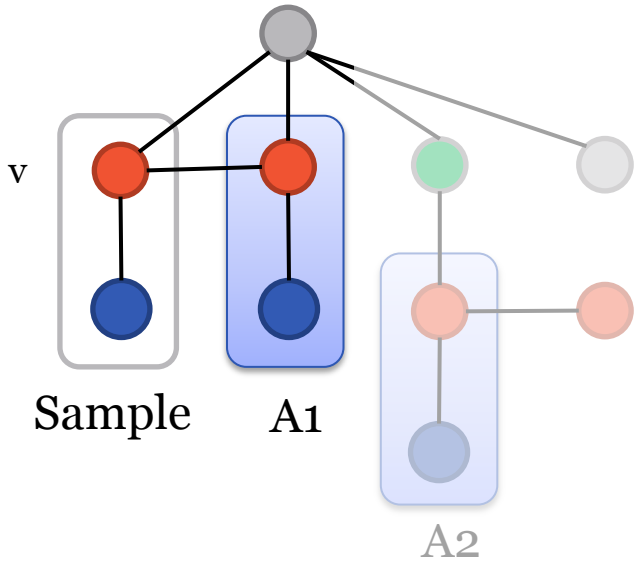
Prune Irrelevant Answers

Mottin et al. [2016]

NP-complete (subgraph isomorphism)

$O(|V|^4)$ (simulation)

Reduce Search Space:
Removes nodes that are likely to be less relevant



Approximation:

- Nodes closed to the sample are more important
- Use Personalized PageRank with a weighted matrix

$$\boldsymbol{v} = (1-c)A\boldsymbol{v} + c\boldsymbol{p}$$

- Weight edges: frequency of the edge-label

$$I(e_{ij}^{\ell}) = I(\ell) = \log \tfrac{1}{P(\ell)} = -\log P(\ell)$$

$$P(\ell) = \frac{|E^{\ell}|}{|E|}$$

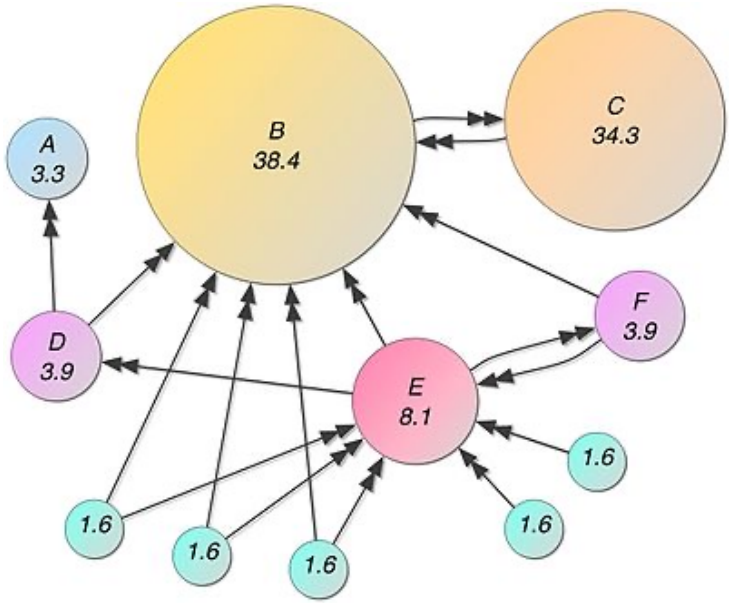
Lissandrini et al. 2022

https://en.wikipedia.org/wiki/PageRank

SublinearAlgorithms for Personalized PageRank, with Applications – Ashish Goel et al

Proximal Relevance

Establish Relatedness via Connectivity



Global Page Rank

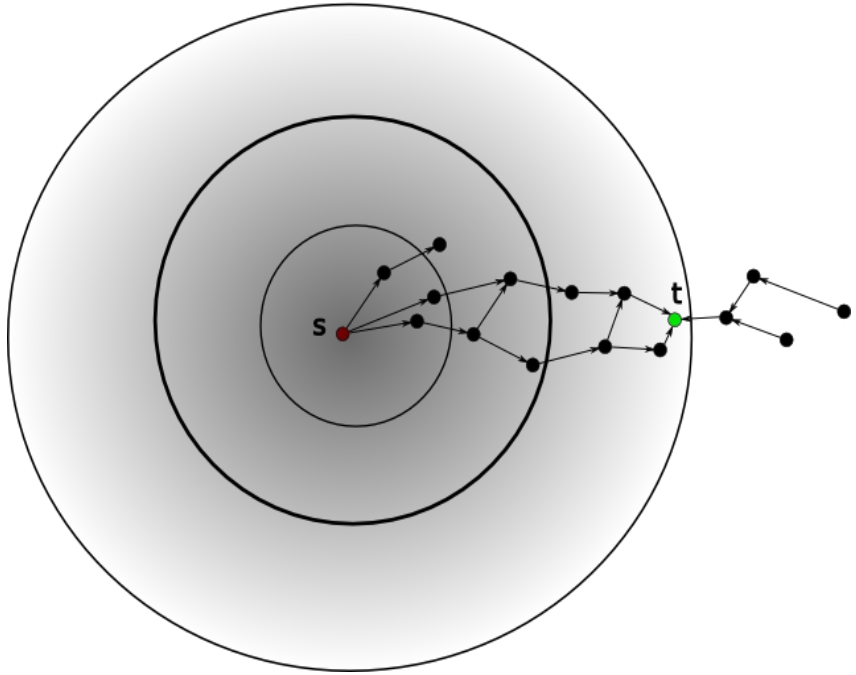
Starting from a random node, traversing randomly, **random restart point** anywhere in the graph

Personalized Page Rank

- Start from seed nodes, i.e. the documents D_{rel}
- Navigate towards locally connected nodes

Example based Exploration implies locality

CHALLENGE:
Identify meaningful transition probabilities



Personalized Page Rank

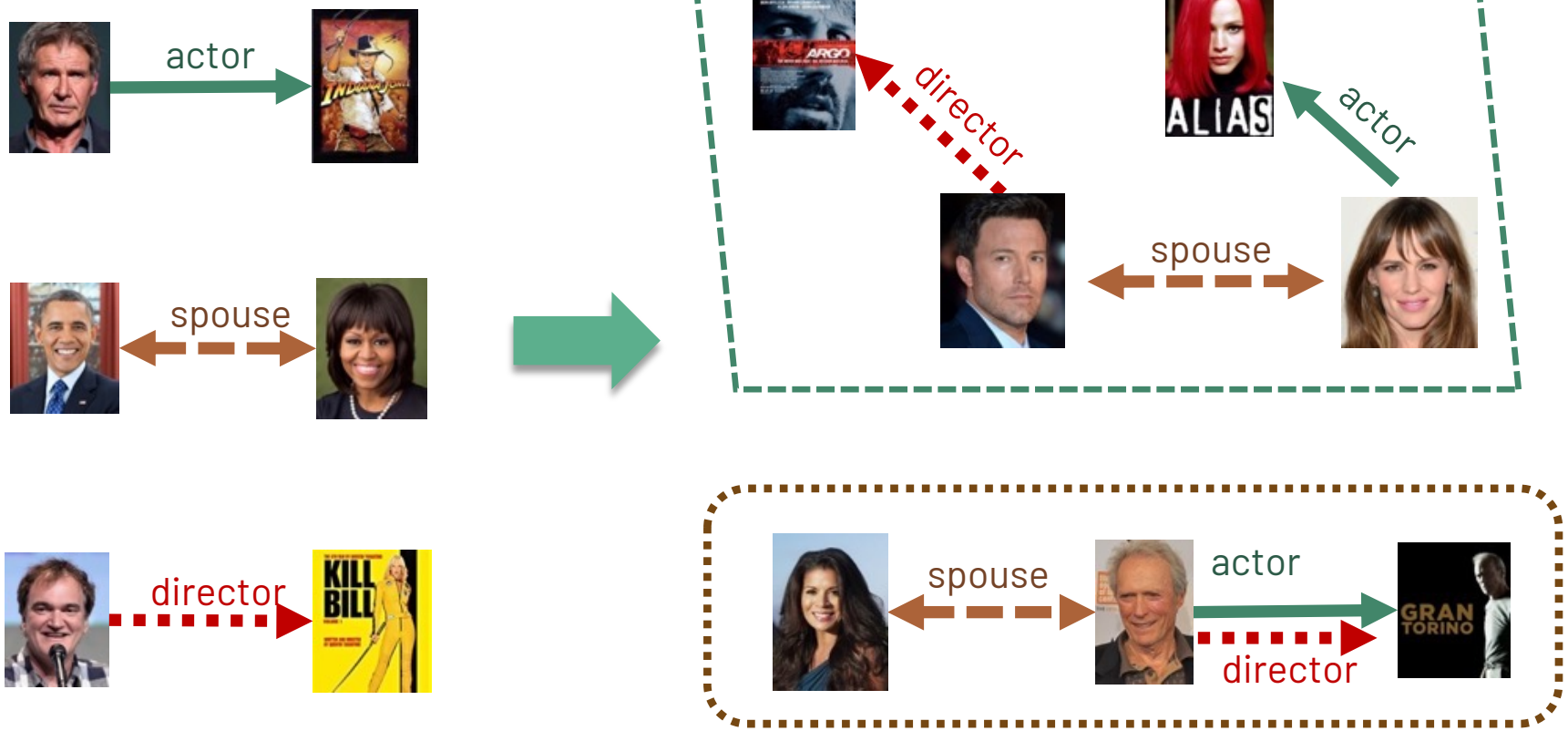
Starting from a **limited set of nodes**, traversing randomly, restart point is one in **the initial set.**
Bound not to travel too far

Lissandrini et al. 2022

Search with Multiple Examples

Combining partial answers

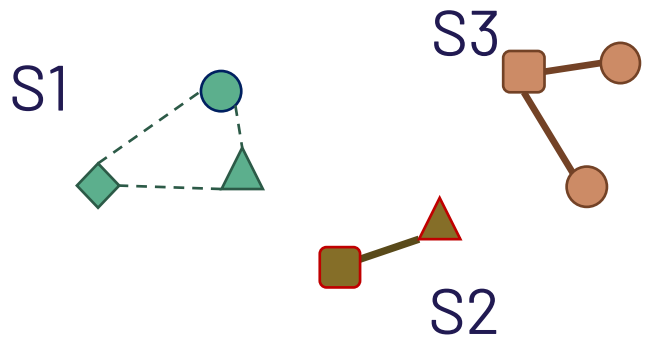
Lissandrini et al. [2018]



- Multiple Simple Examples
- Each Example describes an Aspect
- Results are Combinations of aspects
- Results have possibly Multiple Structures

Case: Unknown Structures ⟶ Find Complex Connections with Simpler Components

Search Framework



Multi-exemplar Answering

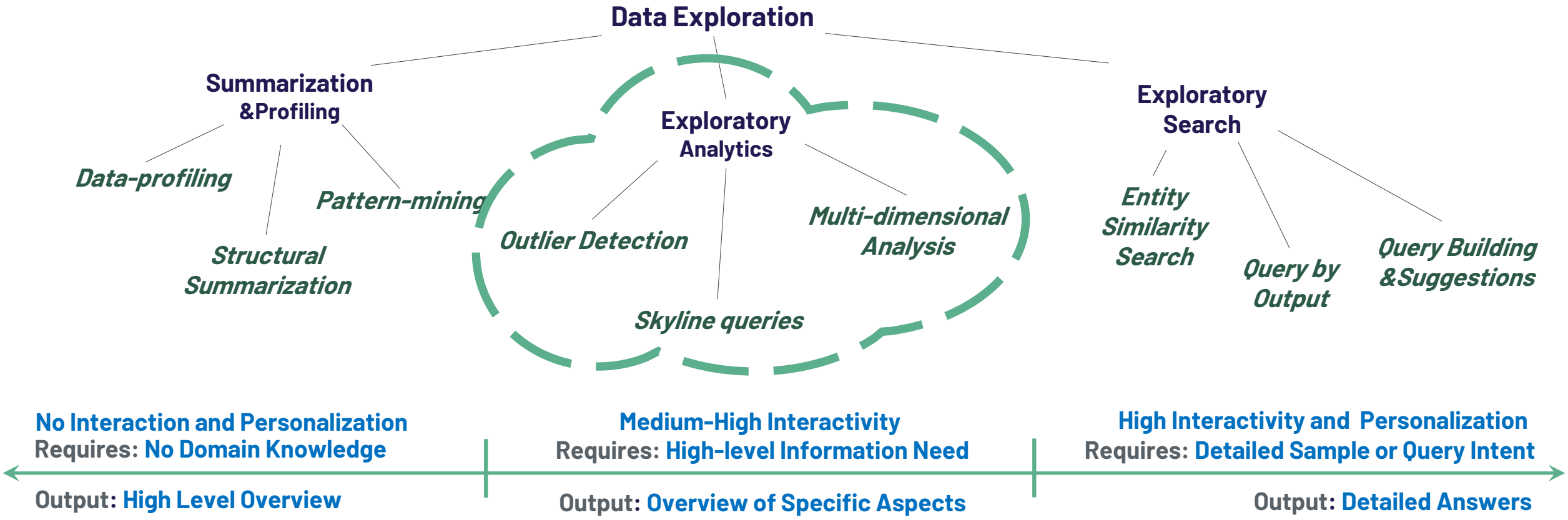
Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, \ldots, s_m \rangle$
Output: Answers $\mathcal{A}$

1: $\mathcal{G} \leftarrow \textsc{Partial}(G, \mathcal{S})$
2: $\mathcal{A} \leftarrow \textsc{Search}(\mathcal{G}, \mathcal{S})$
3: **return** $\mathcal{A}$

Lissandrini et al. [ICDE'2018]

Exploit **Localized Search**

Data Exploration Methods



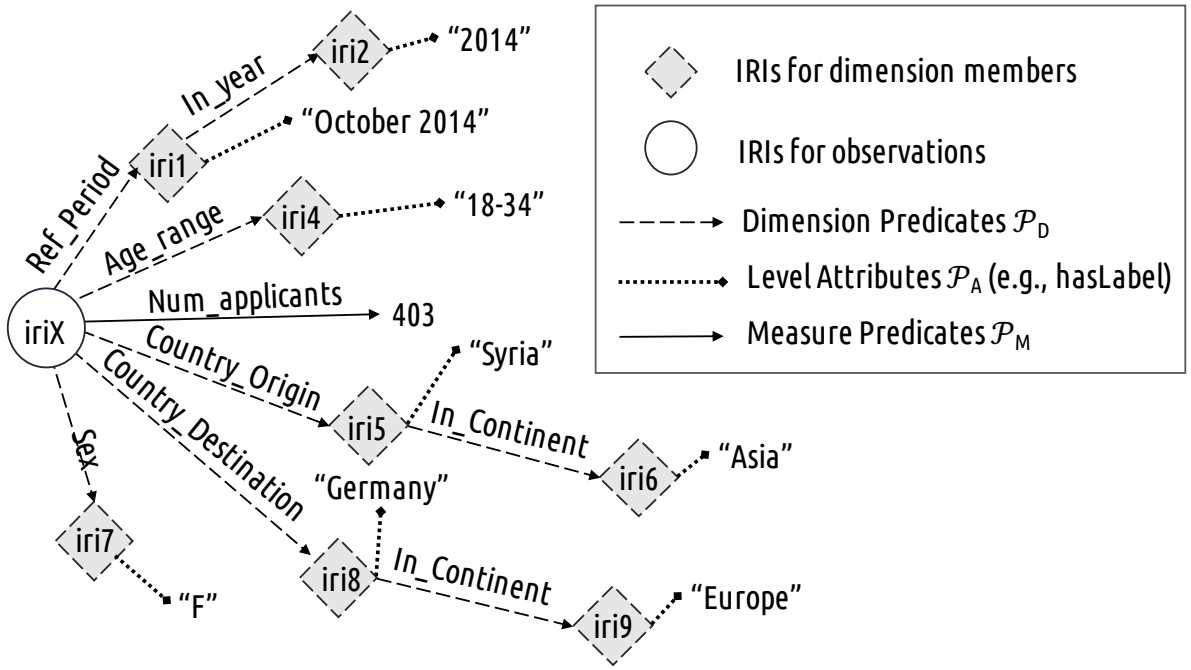
Lissandrini et al. 2022

ReOLAP: Reverse Engineering OLAP Queries on Knowledge Graphs

Lissandrini et al. [EDBT'2023]

Focus: Entities with statistical data

Goal: Enable Analytical Queries



From user **example**: `<Europe, 2014>`

Produce SPARQL analytical query that

1) Maps examples to **dimension members**
2) Produces path queries **to traverse hierarchies**
3) Includes **aggregates of measures**

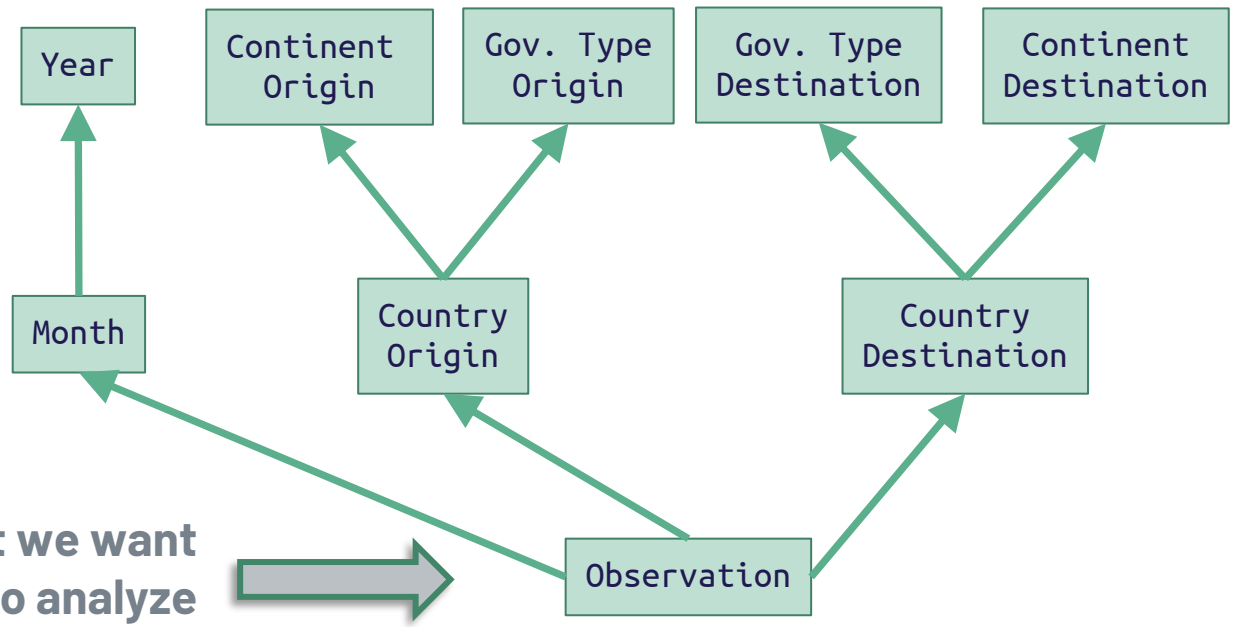
ReOLAP: Reverse Engineering OLAP Queries on Knowledge Graphs

Lissandrini et al. [EDBT'2023]

From user **example**: <Europe, 2014>

Produce SPARQL analytical query that

1) Maps examples to **dimension members**
2) Produces path queries **to traverse hierarchies**
3) Includes **aggregates of measures**



This is what we want to analyze

```
SELECT ?continent ?year (sum(xsd:float(?obsValue)) as ?sumobsValue)
  FROM . . .
 WHERE {
          ?hq :CountryDestination / :ContinentDestination ?continent.
          ?hq :inMonth / :inYear ?year.
          ?hq :Value ?obsValue
        }
GROUP BY ?continent ?year
```

SPARQL

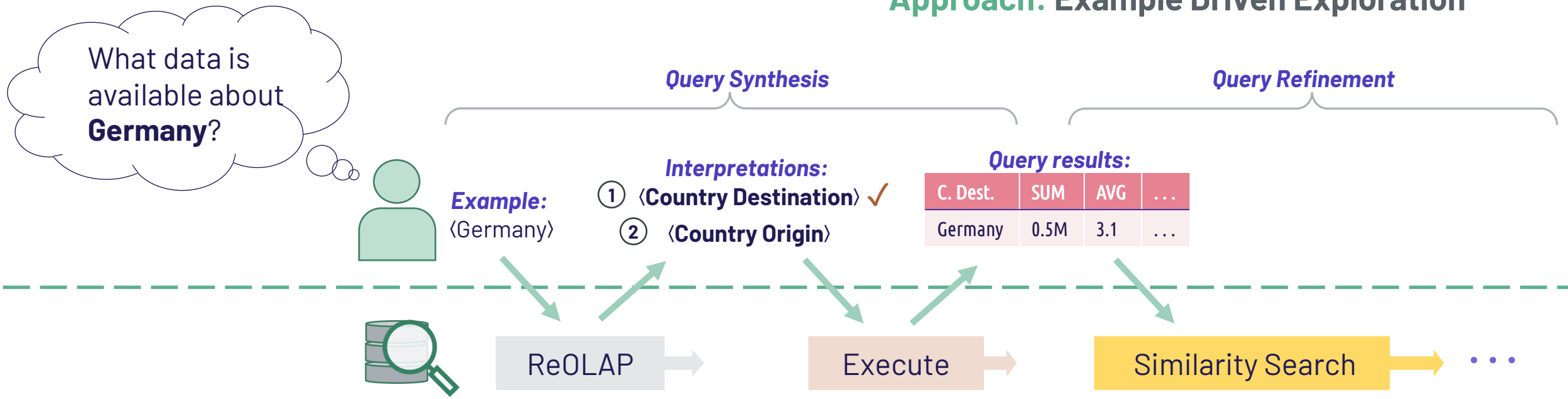
Note: This is one of the 2 possible interpretations

ReOLAP: Exploration Workflow (i)

Iterative & Interactive Exploration Process

Lissandrini et al. [EDBT'2023]

GOAL: Remove user need for writing queries
Approach: Example Driven Exploration



Exploration is an iterative process, where the result of a query leads to more advanced questions to answer...

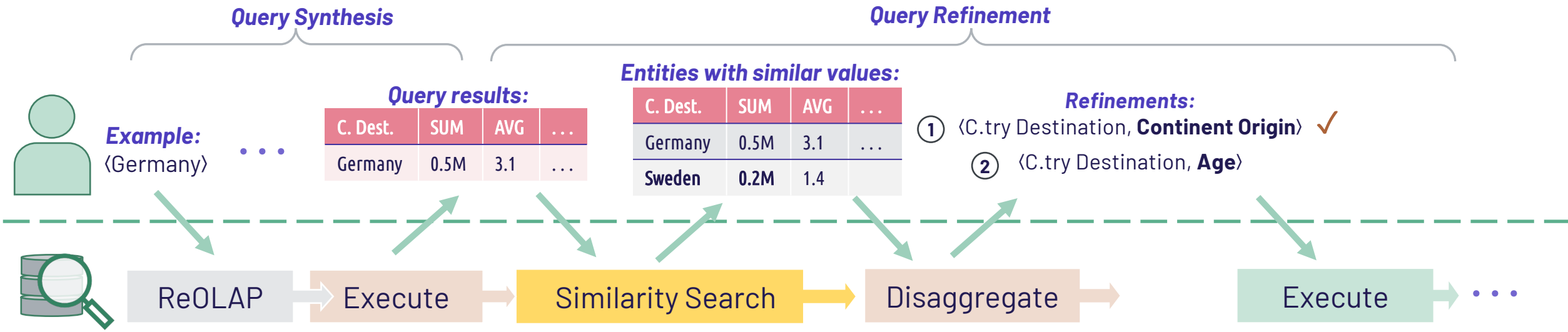
1) Not sure **about the data** we have
2) Not clear **what we are looking for**
3) Not sure **where to look for it**

ReOLAP: Exploration Workflow (ii)

Iterative & Interactive Exploration Process

Lissandrini et al. [EDBT'2023]

GOAL: Remove user need for writing queries
Approach: Example Driven Exploration

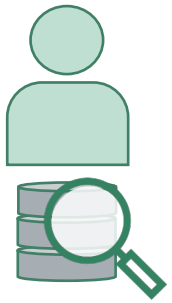


ReOLAP: Exploration Operators

Iterative & Interactive Exploration Process

Lissandrini et al. [EDBT'2023]

GOAL: Remove user need for writing queries
Approach: Example Driven Exploration



Query Synthesis *Query Refinement*

ReOLAP → Execute → Similarity Search → Disaggregate → Execute → • • •

We need operators equivalent to: DRILL-DOWN, SLICE, DICE

From Example: ⟨Germany⟩ obtain Interpretation

ReOLAP → From an **example tuple** to a **SPARQL query** Identify **dimensions, aggregates,** and **group by**

From Entity&Measure ⟨Germany, SUM:0.5M⟩ identify similar entities

Similarity Search → From an **example tuple with values,** identify **entities** that present the most similar values in the current aggregation level

From Entity&Measure ⟨Germany, SUM:0.5M⟩ obtain disaggregation

Disaggregate → From the current **Aggregation Level** Propose possible decompositions, e.g., introduce dimensions

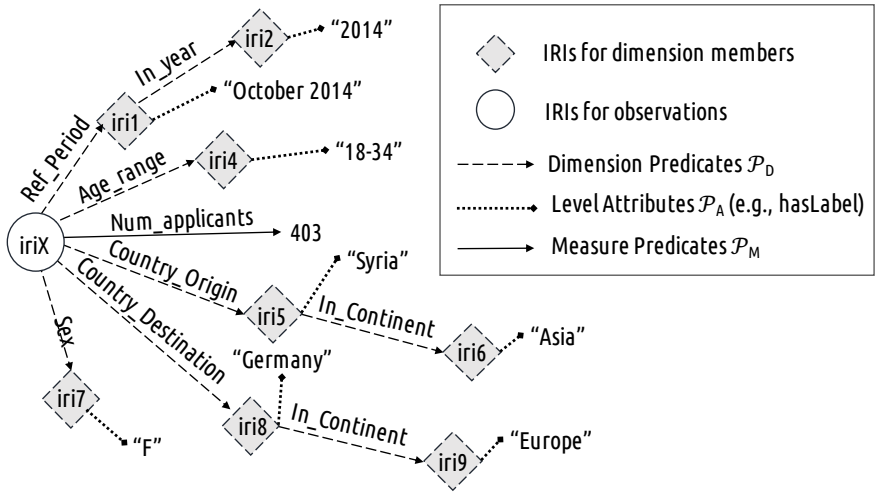
Reduce large set of tuples ⟨Germany, Syria⟩ , ⟨France, Syria⟩, ...

Subset Results → When a query returns **a large number** of tuples, identify **filter conditions to reduce to a subset (top-k subset, same percentile)**

ReOLAP: Reducing the Search Space

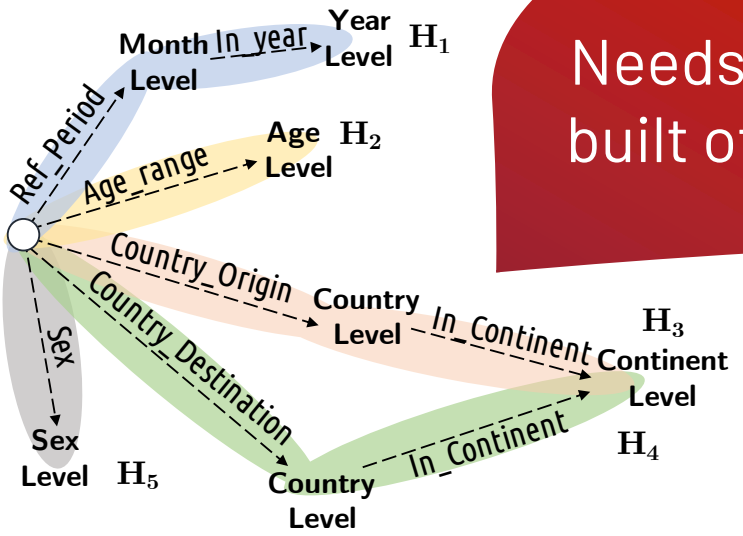
Lissandrini et al. [EDBT'2023]

The query needs to analyze "observations" connected to dimensions and measures:



Virtual Schema Graph
Obtained through Graph Traversal

Queries are obtained traversing the Virtual Schema Graph



Needs to be built offline!

Virtual Schema Graph: A Structural Graph Summary

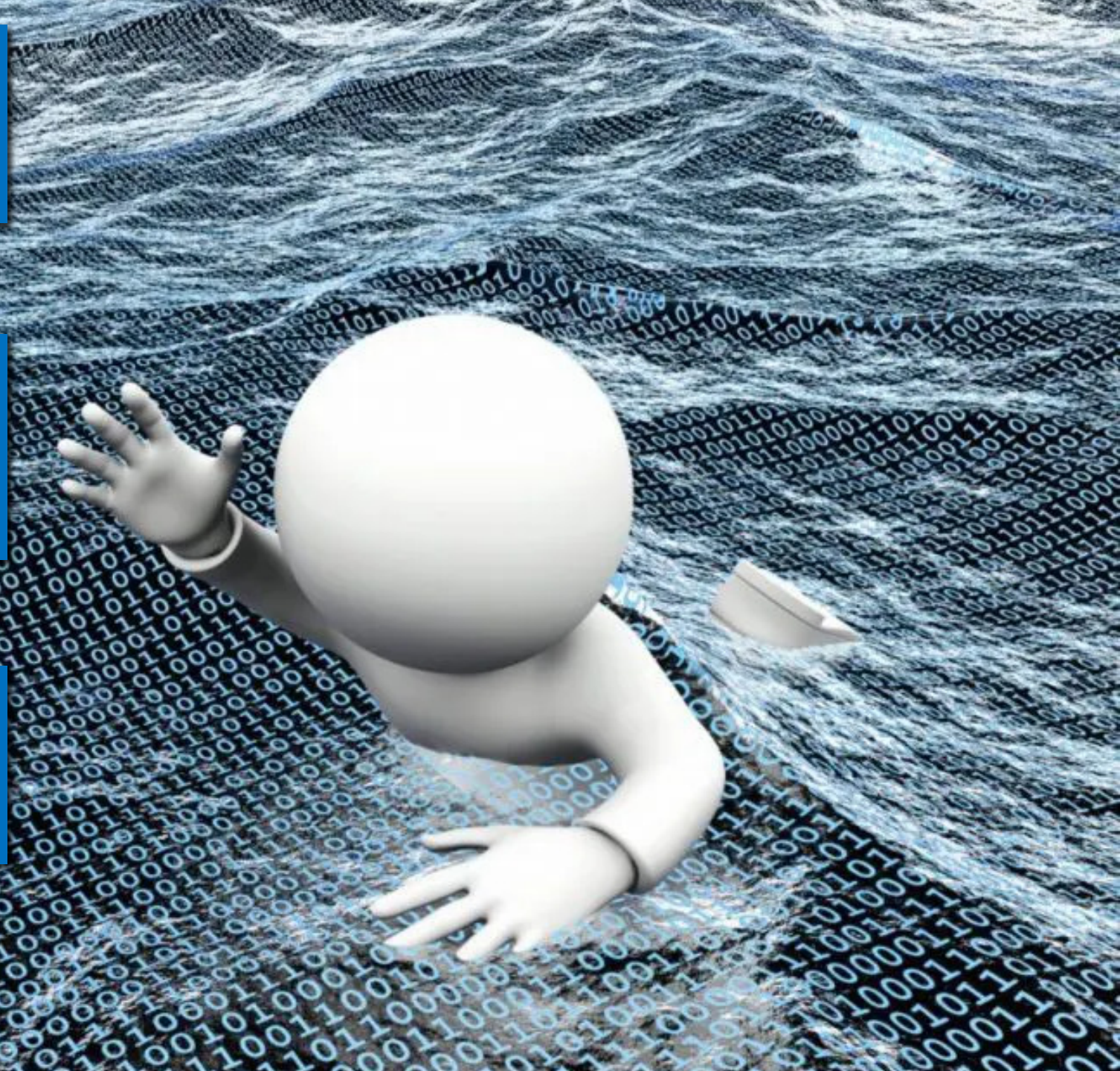
Data Structure representing the Analytical Schema: A Summary of the Search Space

Analytical Schema Describes: Dimensions, Hierarchies, & Measures

Knowledge Graphs
(integrating heterogenous data)

KG Search & Exploration
(exploring heterogenous data)

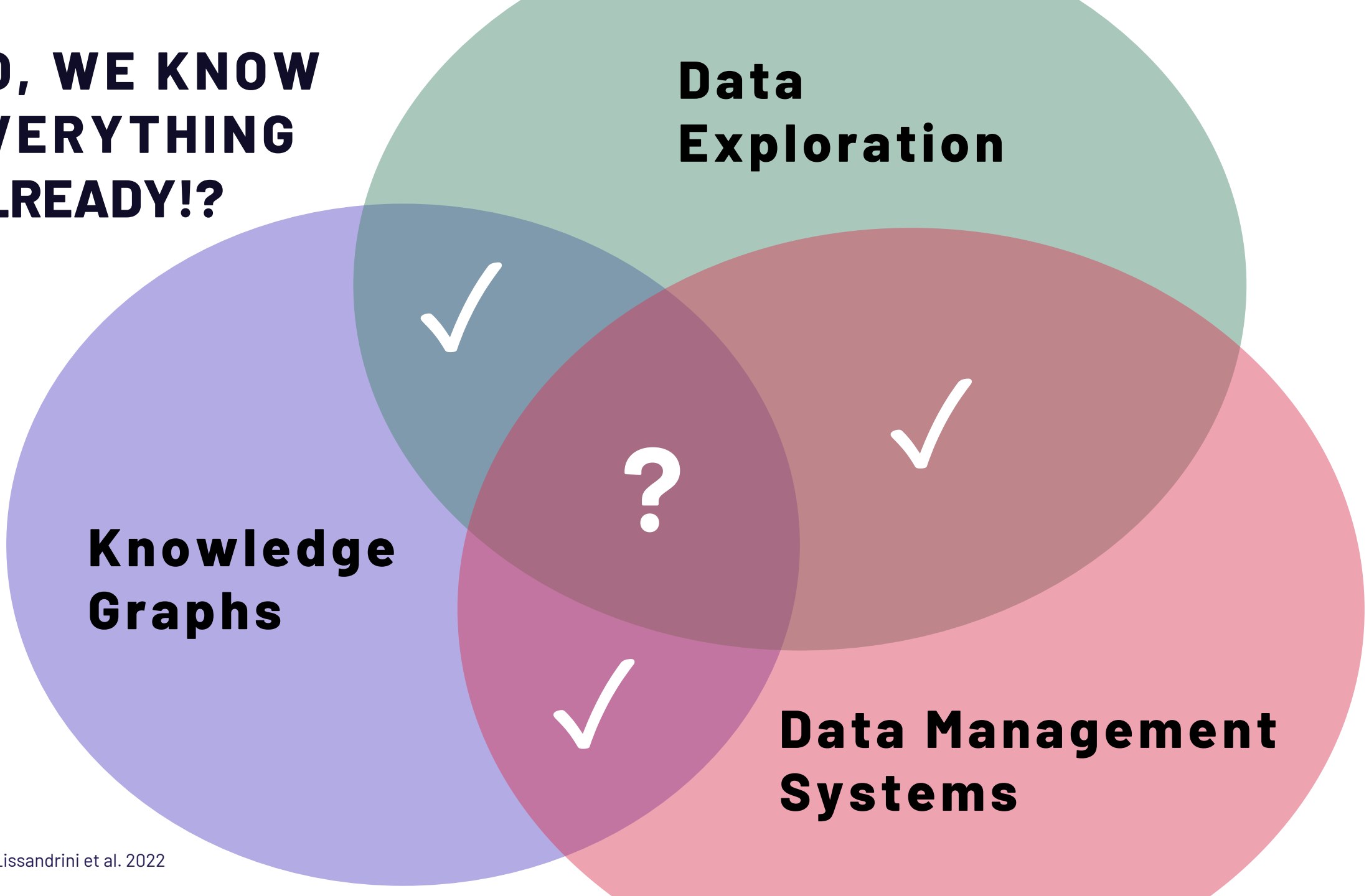
KG Exploration Systems
(systems to analyze heterogenous data)



KG Exploration Systems

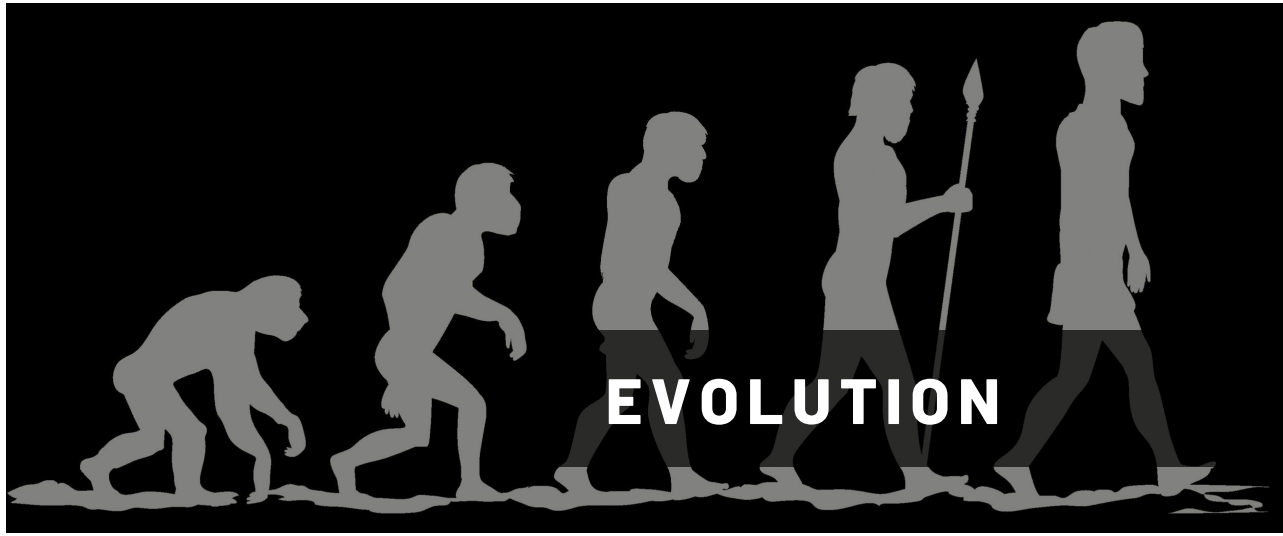
(systems to analyze heterogenous data)

SO, WE KNOW EVERYTHING ALREADY!?



Lissandrini et al. 2022

KG Data Management Challenges



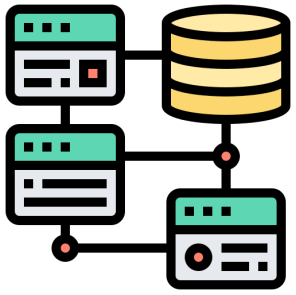
Lissandrini et al. 2022

What About Existing Systems?

- Exploit **regular** or small schemas
- Study **simple** queries
- Inefficient on long **path** queries
- Limited **scalability** on KG wit
  large number of **relati**

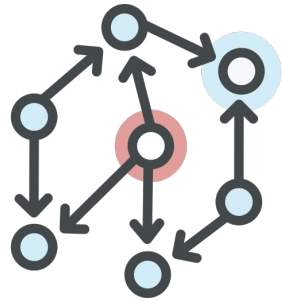
- Optimized for **point-wise** queries
  (i.e., nodes bors)
- for **complex path**
  queries
- Lack of **benchmark** for KG **exploration**

No system for KG exploration



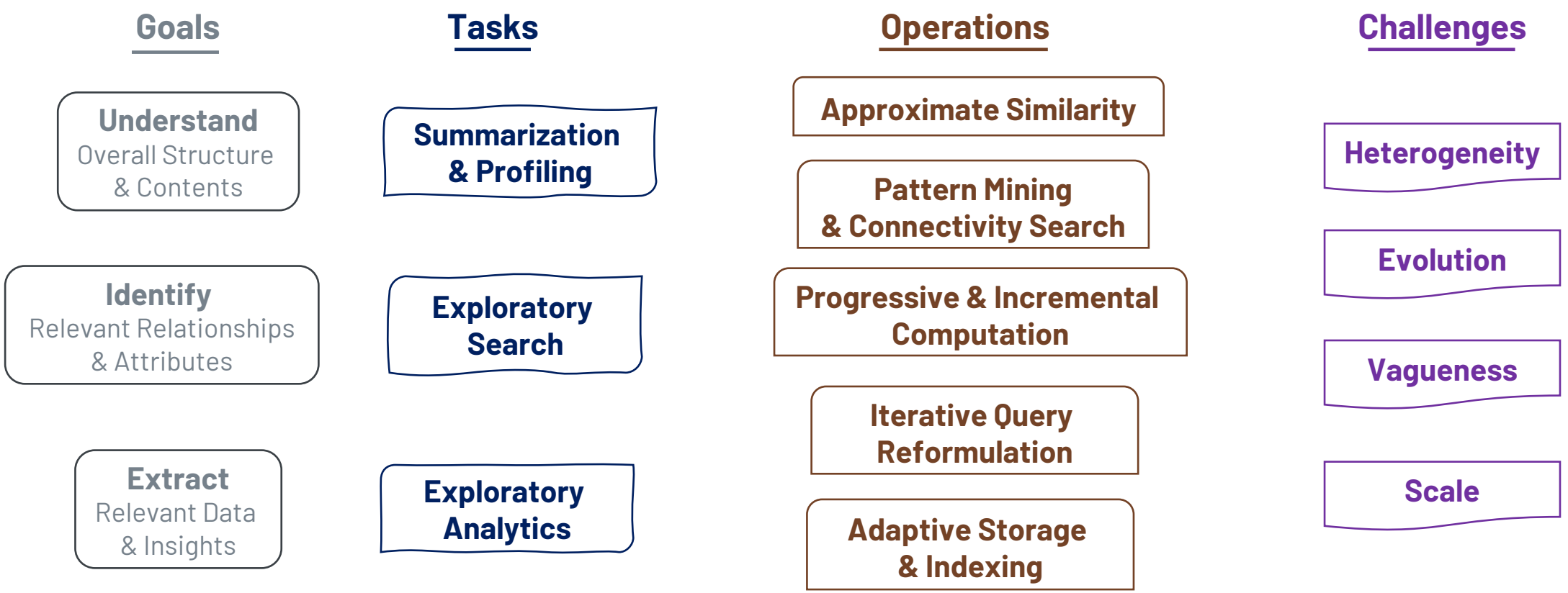
Relational DBMS

Graph DBMS



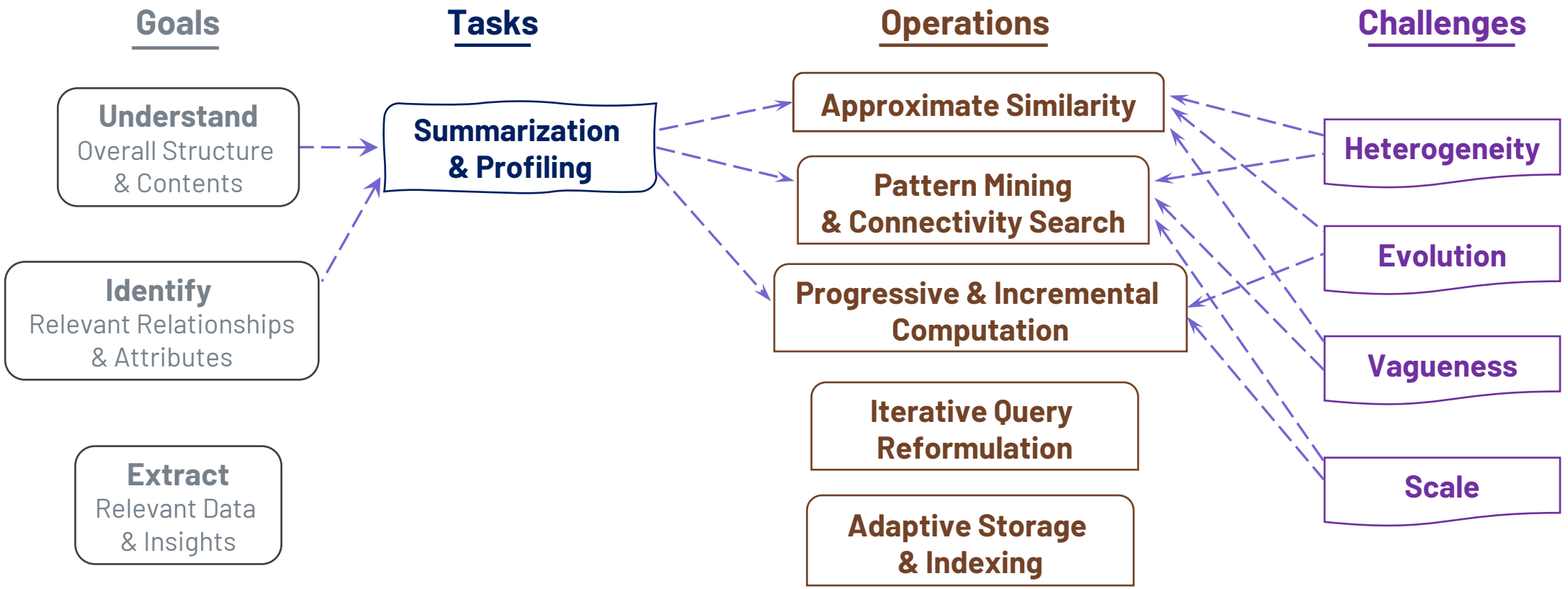
Lissandrini et al. 2022

Overview: Goals / Tasks / Operations / Challenges



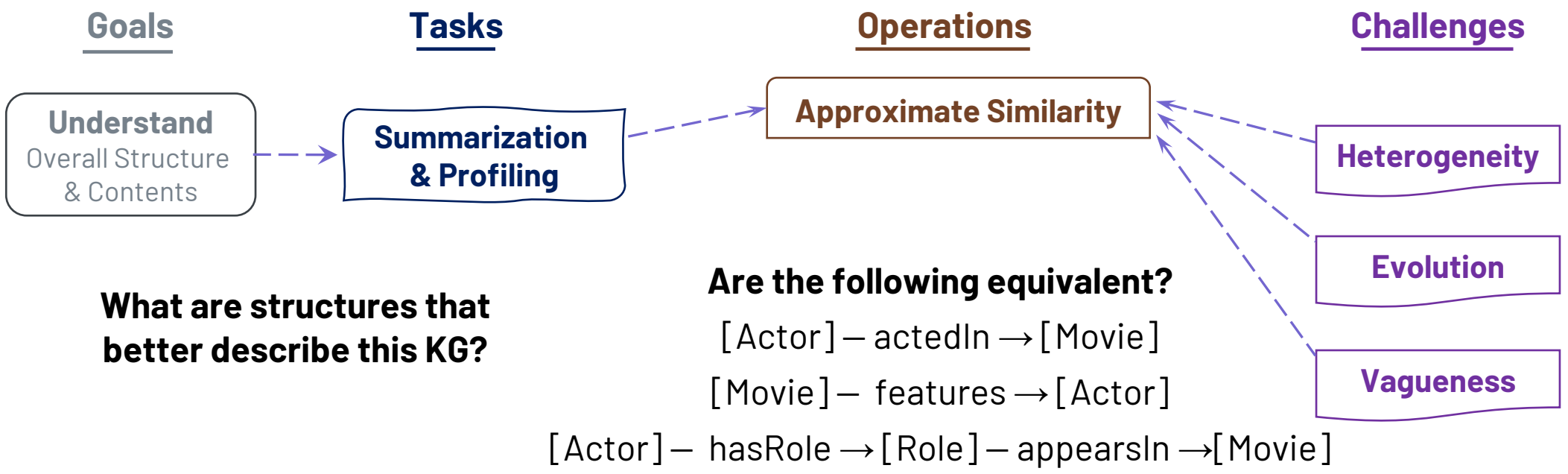
Lissandrini et al. 2022

Summarization & Profiling

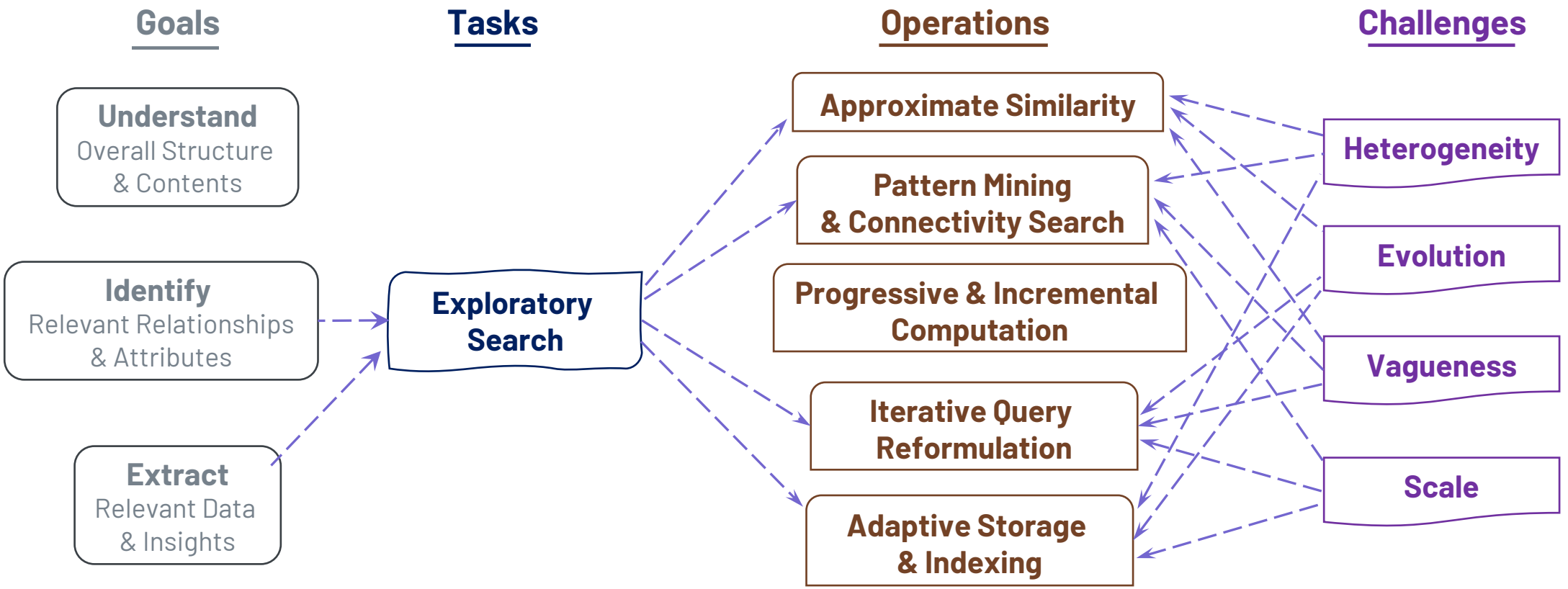


Lissandrini et al. 2022

An Example

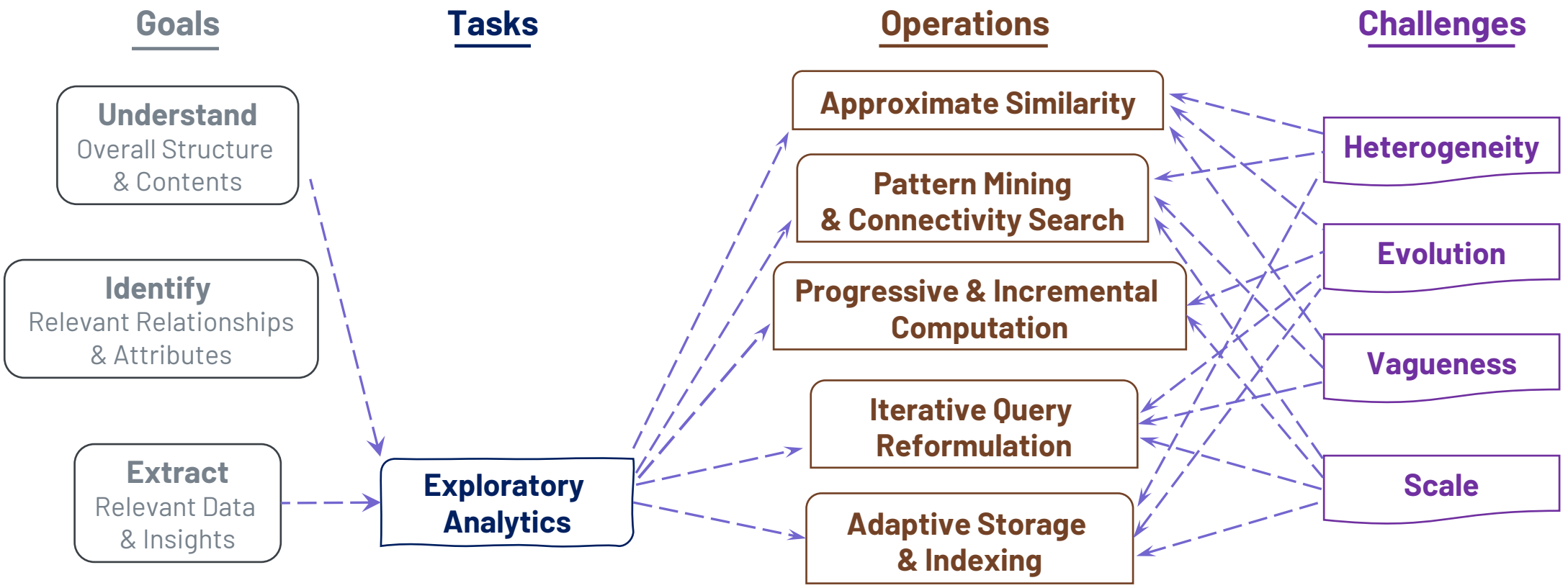

Goals
Tasks
Operations
Challenges
Understand
Overall Structure
& Contents
Summarization
& Profiling
Approximate Similarity
Heterogeneity
Evolution
Vagueness
What are structures that better describe this KG?
Are the following equivalent?
[Actor] – actedIn → [Movie]
[Movie] – features → [Actor]
[Actor] – hasRole → [Role] – appearsIn → [Movie]

Exploratory Search



Lissandrini et al. 2022

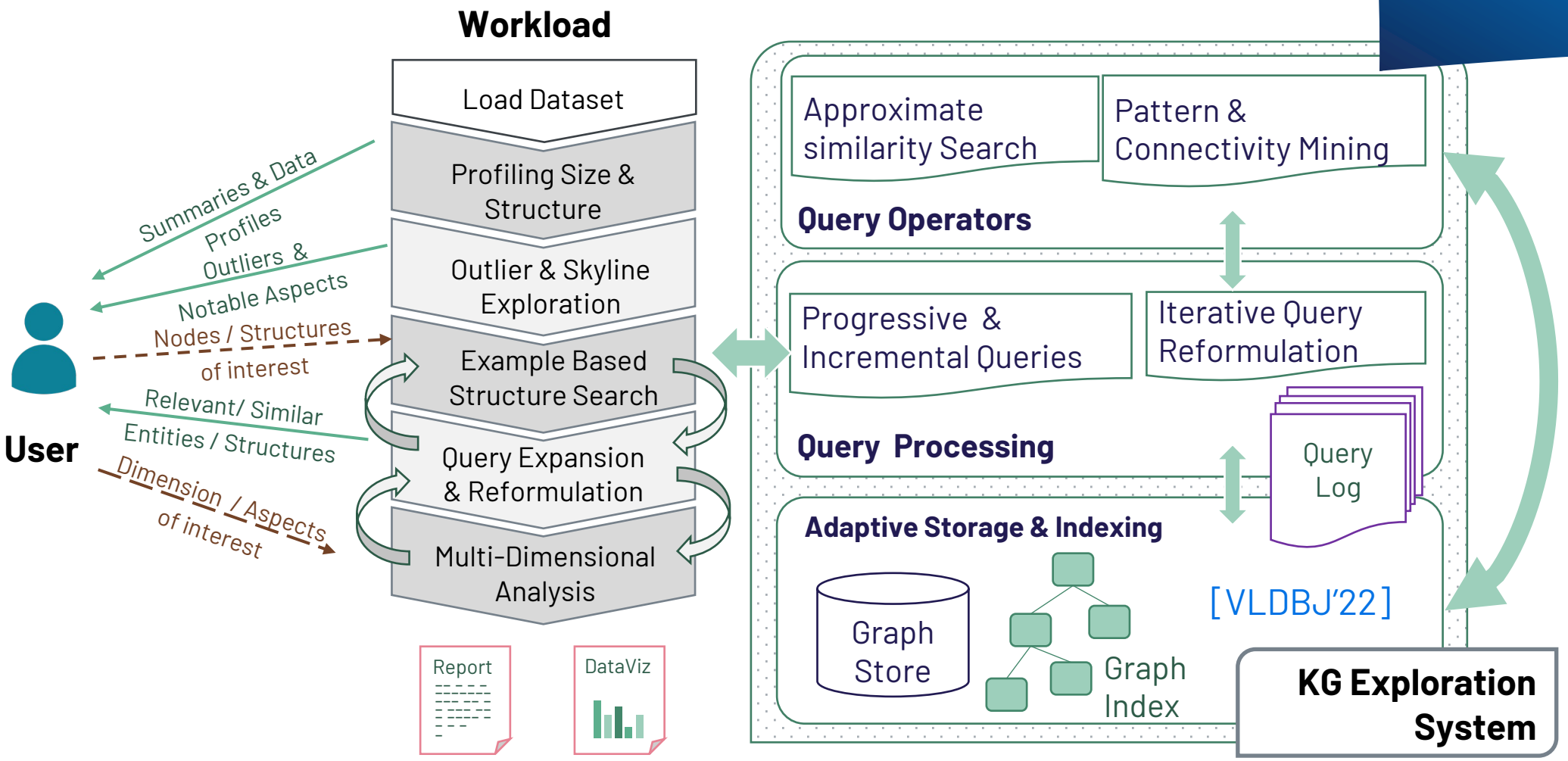
Exploratory Analytics



Lissandrini et al. 2022

A KG Exploration System

Thanks!
Questions?



Sagi, T., Lissandrini, M., Pedersen, T.B., and Hose K.
"**A design space for RDF data representations**"
The **VLDB Journal** 31, 347–373 (2022).

Understanding RDF Data Representations in Triplestores

Matteo Lissandrini,
Tomer Sagi, Torben Bach Pedersen, Katja Hose

AALBORG UNIVERSITY
DENMARK

GOALS

How to** (efficiently?) **store a Knowledge Graph

How to model the space of alternative data representations for KGs

How to evaluate alternative storage options

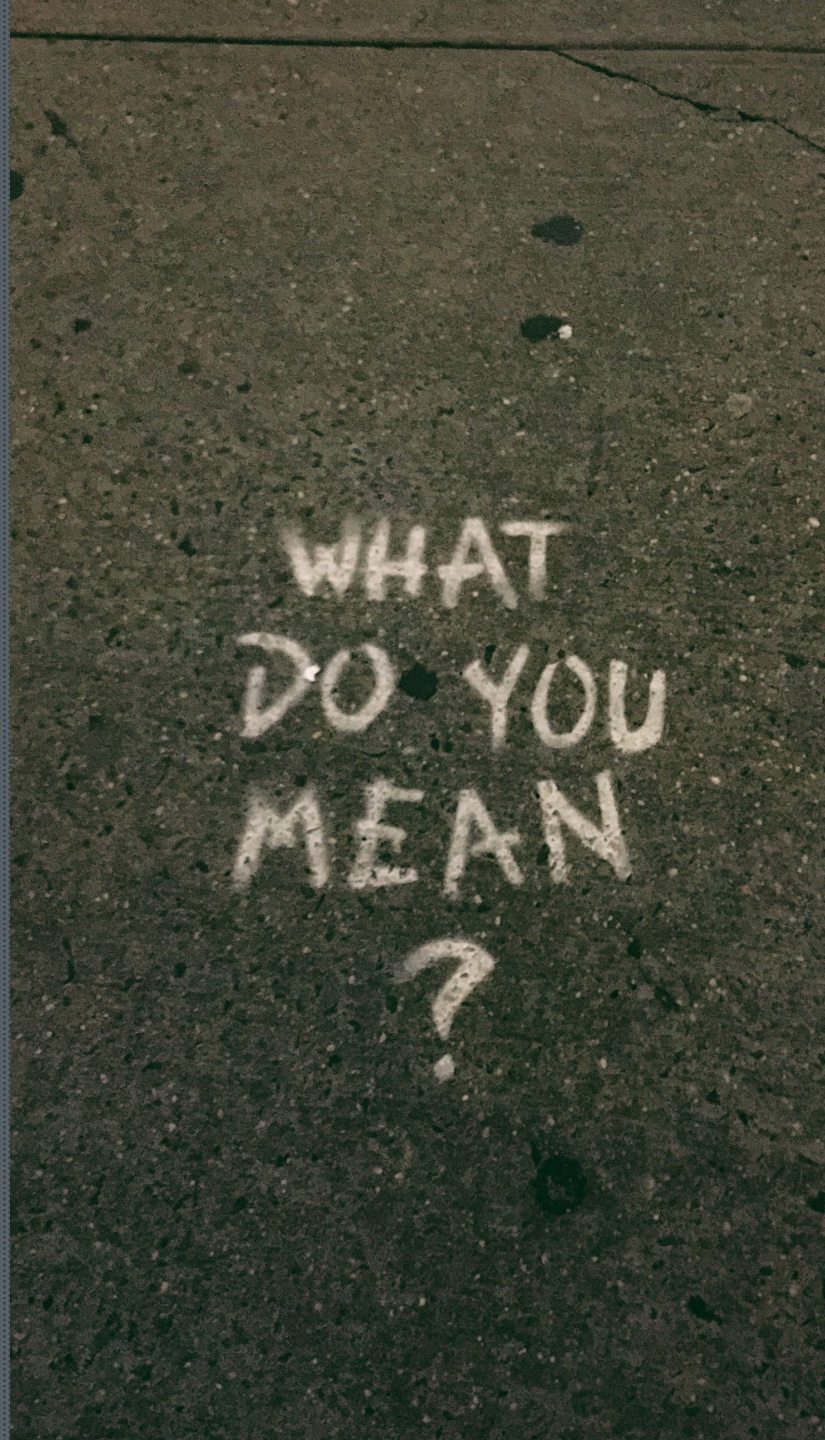
DISCLAIMER

Things to keep in mind

Intuition ≫ Technical Details

Summary of a 27pp survey with 23 systems (both prototypes & commercial)

Generate discussion of applicability beyond Triplestores



Knowledge Graphs

Heterogeneous Data & Heterogeneous Connections

Multiple node / edge types

WikiData/Dbpedia has 9K/1.3K edge types

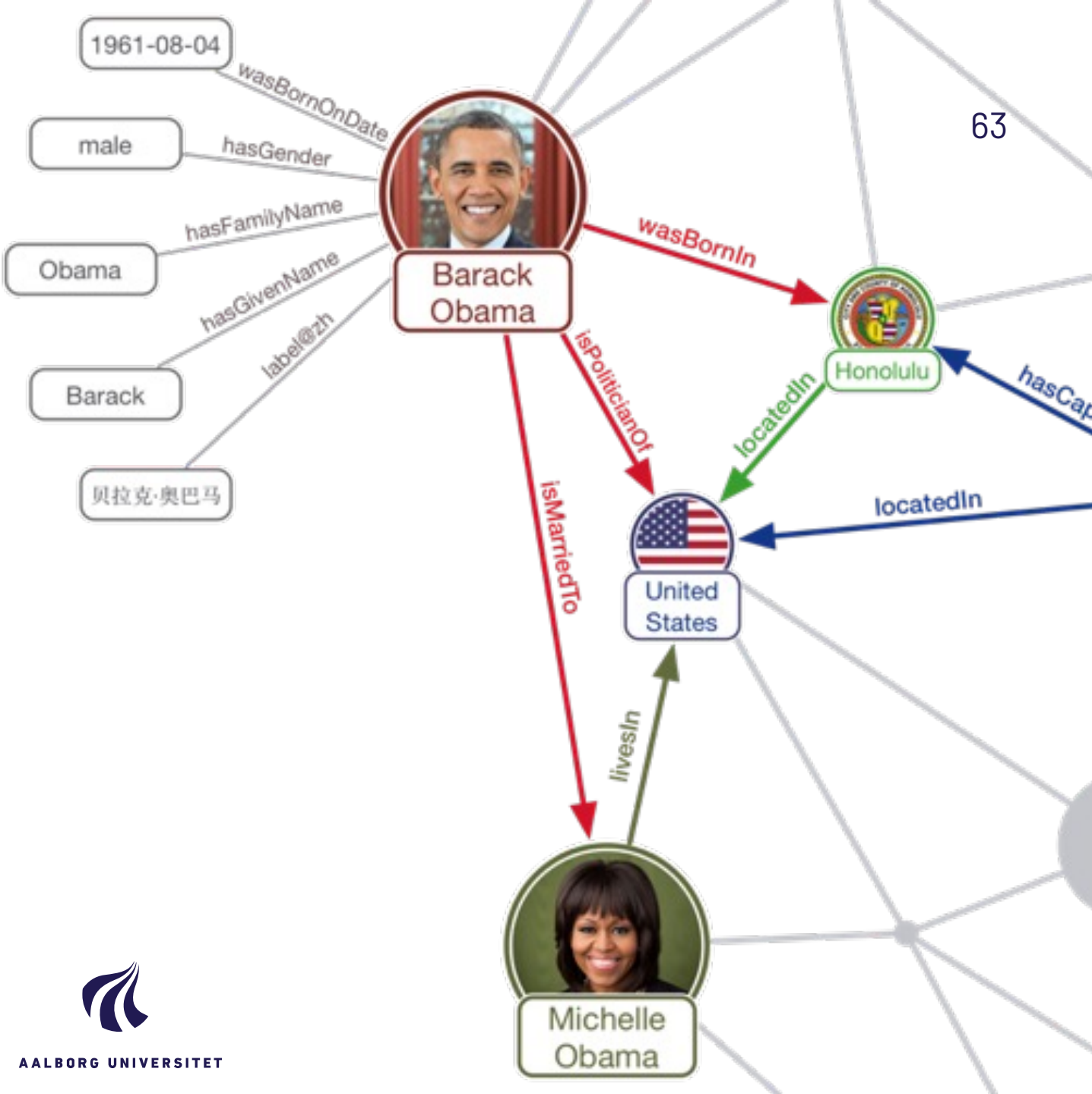
Literals with Data Types & Annotations

@en , ^^xsd:integer

Taxonomic data is part of the graph

WikiData/DBpedia has 82K/0.4K classes

Incomplete/Inconsistent Schema



AALBORG UNIVERSITET

Triplestores

Representing a KG as a Collection of Facts

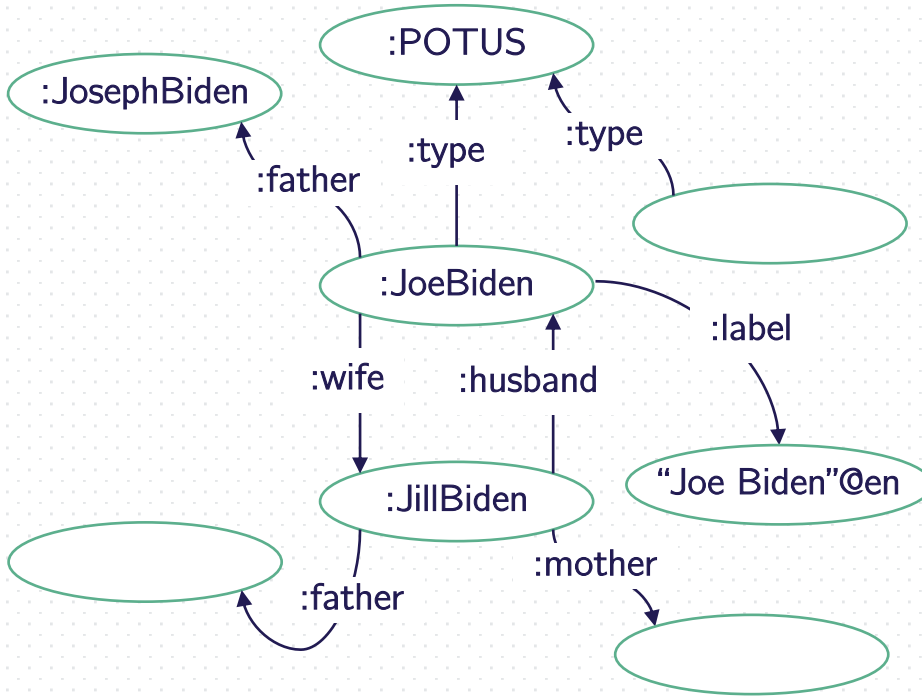
the RDF Model

- **Nodes** are either:
  - **Entities** (resources identified by IRI)
  - **Literals** (values as strings, integers, dates)
  - **Blank** Nodes (special kind of nodes without IRI)

- **Edges are statements**
  (Subject, Predicate, Object)

That's a triple!

- Edge types are resources




```
@prefix : <http://www.example.kg/> .
:JoeBiden :label "Joe Biden"@en .
:JoeBiden :type  :POTUS .
:JoeBiden :wife :JillBiden .
:JillBiden :husband :JoeBiden .
```

Lissandrini et al. 2022

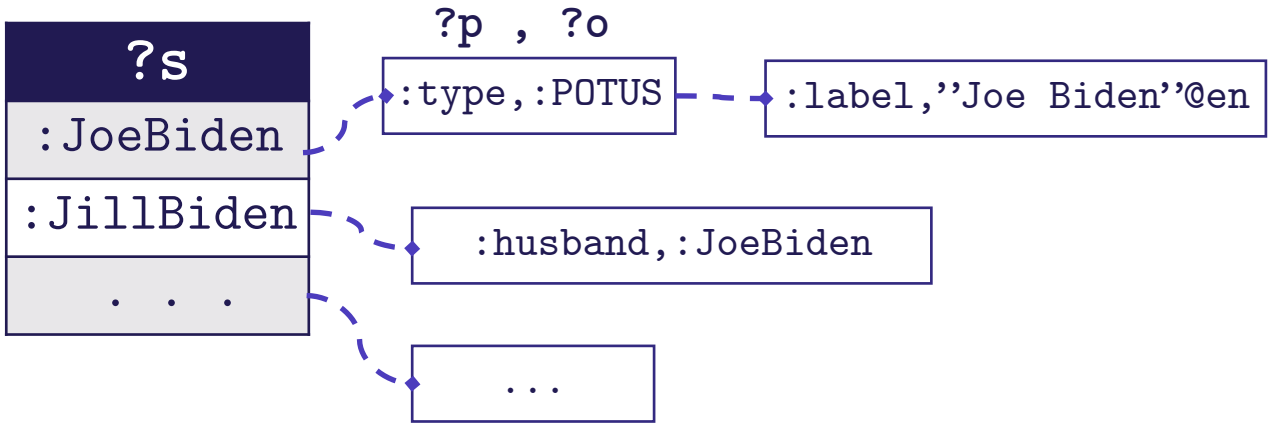
How to store a Triple?

2 random options

List of triples

?s	?p	?o
:JoeBiden	:label	“JoeBiden”@en
:JoeBiden	:type	:POTUS
:JillBiden	:husband	:JoeBiden
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Hash table



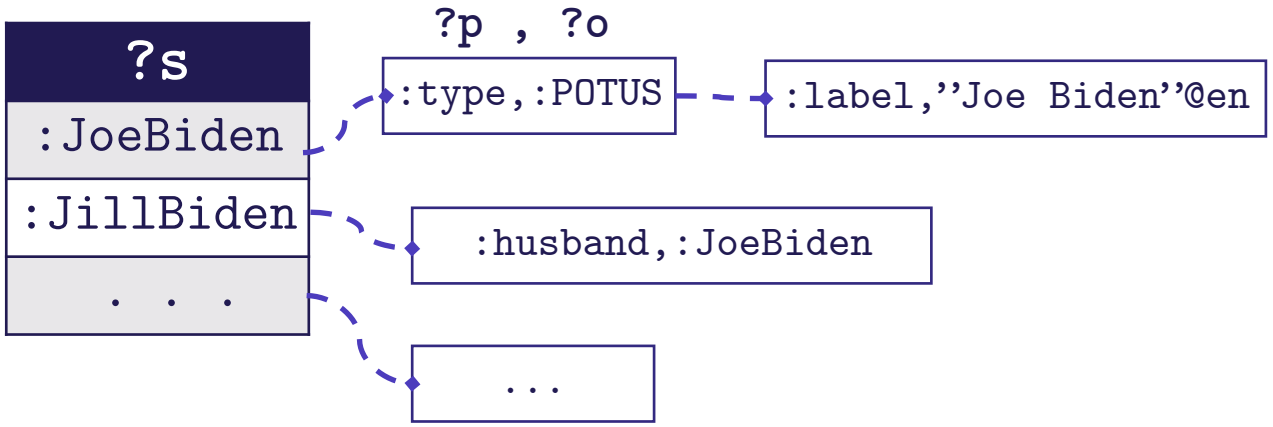
How to store a Triple?

a 3rd option

SORTED List of triples

?s	?p	?o
:JillBiden	:husband	:JoeBiden
:JoeBiden	:label	‘‘JoeBiden’’@en
:JoeBiden	:type	:POTUS
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Hash table



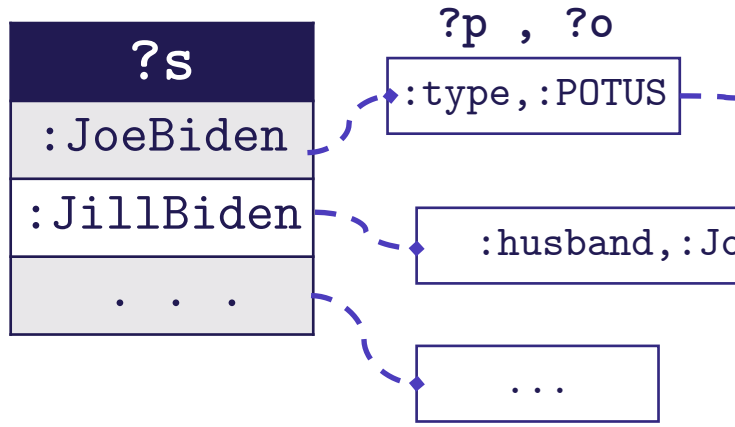
How to store a Triple?

a 4th & 5th option

Property Table

?s	:husband	:label	:type	:wife
:JillBiden	:JoeBiden	-	-	-
:JoeBiden	-	"JoeBiden"@en	:POTUS	:JillBiden
. . .	. . .	. . .	. . .	. . .

Hash table



Property Tables

:husband

?s	?o
:JillBiden	:JoeBiden
. . .	. . .

:label

?s	?o
:JoeBiden	"JoeBiden"@en
. . .	. . .

There is more !!!

:type

:wife

What's Best ?

It Depends!

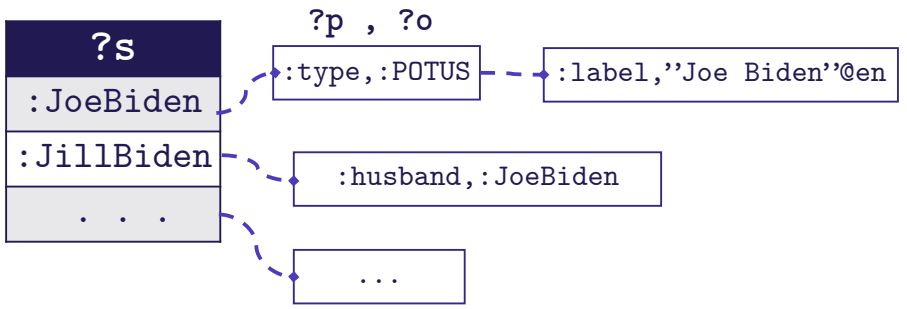
List of triples

?s	?p	?o
:JoeBiden	:label	‘‘JoeBiden’’@en
:JoeBiden	:type	:POTUS
:JillBiden	:husband	:JoeBiden
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Property Table

?s	:husband	:label	:type	:wife
:JillBiden	:JoeBiden	-	-	-
:JoeBiden	-	‘‘JoeBiden’’@en	:POTUS	:JillBiden
. . .	. . .	. . .	. . .	. . .

Hash table



***SORTED* List of triples**

?s	?p	?o
:JillBiden	:husband	:JoeBiden
:JoeBiden	:label	‘‘JoeBiden’’@en
:JoeBiden	:type	:POTUS
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Property Tables

:husband

?s	?o
:JillBiden	:JoeBiden
. . .	. . .

:label

?s	?o
:JoeBiden	‘‘JoeBiden’’@en
. . .	. . .

:type

?s	?o
:JoeBiden	:POTUS
. . .	. . .

:wife

?s	?o
:JoeBiden	:JillBiden
. . .	. . .

Querying a Triplestore

SPARQL & Graph patterns

- **Triple Pattern**

  A statement with variables

  (?f :label ?n) ⟶ ?f & ?n **are variables**

- **Basic Graph Pattern**

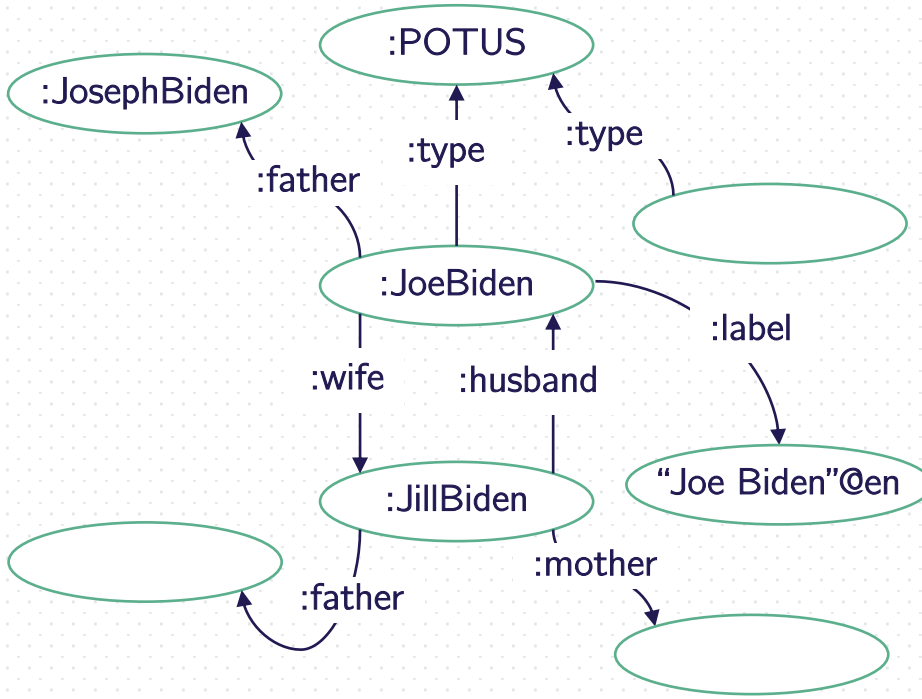
  A set of triple patterns that need to be satisfied

  (?f :label ?n) & (?f :type ?t)

- **Property Paths**

  RegEx like expression in place of triple patterns

  (?f :father+/:type/:subclass* :President)




```
SELECT ?f ?n
WHERE {
      ?f :father / :type :POTUS .
      ?f :label ?n .
}
```

Querying a Triplestore

The Space of Access Patterns

Presence of **CONSTANTS:**

(:JoeBiden :label ?n) **VS.** (?s :label ?n)

Nature of **TRAVERSAL:**

(?s :type / :subclass ?t) **VS.** ?s :type+ / :subclass* ?t

Nature of **FILTERS:**

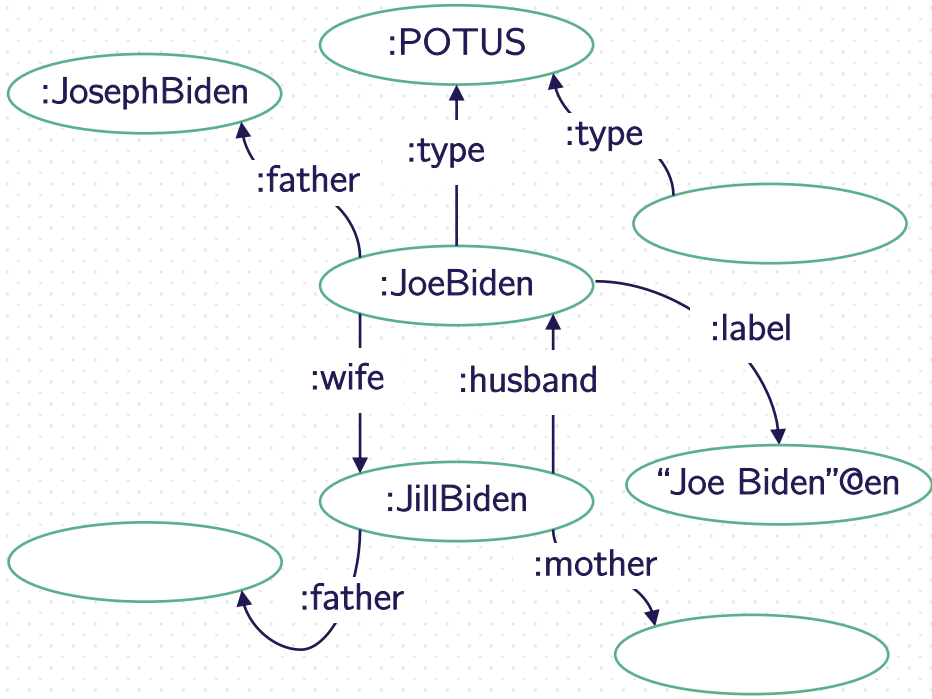
FILTER(?n >= “J”) **VS.** FILTER(isLiteral(?n))

Join on **PIVOT:**

Binary on Same Position **VS.** Arbitrary N-way

RETURN values**:**

All Values **VS.** Existence **VS.** Aggregates




```
SELECT ?f ?n
WHERE {
      ?f :father / :type :POTUS .
      ?f :label ?n .
}
```

What's Best ?

```
SELECT ?o
WHERE {
    :JoeBiden :type ?o .
}
```

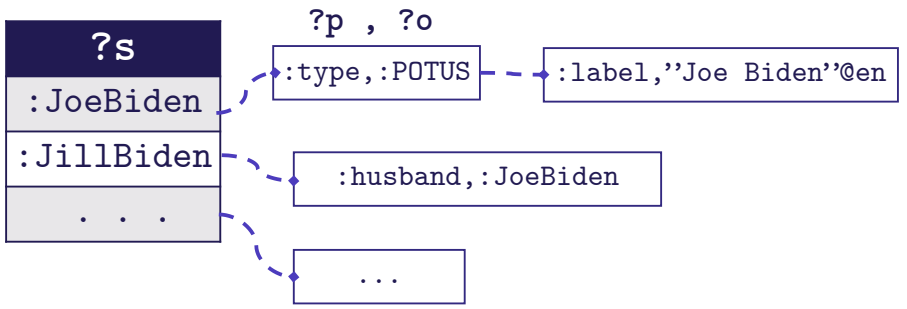
List of triples

?s	?p	?o
:JoeBiden	:label	''JoeBiden''@en
:JoeBiden	:type	:POTUS
:JillBiden	:husband	:JoeBiden
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Property Table

?s	:husband	:label	:type	:wife
:JillBiden	:JoeBiden	-	-	-
:JoeBiden	-	''JoeBiden''@en	:POTUS	:JillBiden
. . .	. . .	. . .	. . .	. . .

Hash table



***SORTED* List of triples**

?s	?p	?o
:JillBiden	:husband	:JoeBiden
:JoeBiden	:label	''JoeBiden''@en
:JoeBiden	:type	:POTUS
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

Property Tables

:husband

?s	?o
:JillBiden	:JoeBiden
. . .	. . .

:label

?s	?o
:JoeBiden	''JoeBiden''@en
. . .	. . .

:type

?s	?o
:JoeBiden	:POTUS
. . .	. . .

:wife

?s	?o
:JoeBiden	:JillBiden
. . .	. . .

Compatibility

Representation can be "**compatible**" to an **access pattern in 3 ways:**

- **seek compatible:** the first result can be retrieved in a single random access
- **sequence compatible:** the remaining results can be computed without random access
- **selection compatible:** if no excess results are retrieved by the computation

```
SELECT ?o
WHERE {
    :JoeBiden :type ?o .

}
```

O(|E|)

List of triples

?s	?p	?o
:JoeBiden	:label	"JoeBiden"@en
:JoeBiden	:type	:POTUS
:JillBiden	:husband	:JoeBiden
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

O(log(|E|))

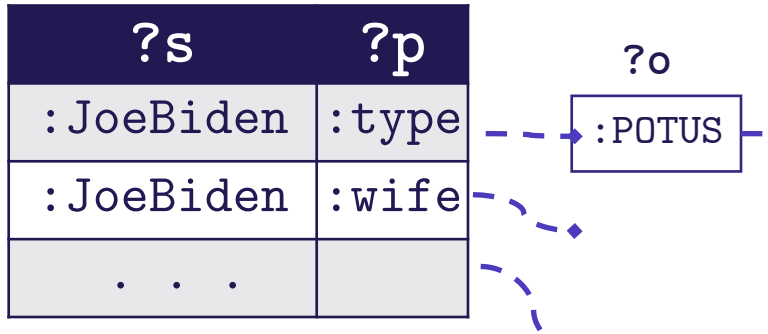
SORTED List of triples

?s	?p	?o
:JillBiden	:husband	:JoeBiden
:JoeBiden	:label	"JoeBiden"@en
:JoeBiden	:type	:POTUS
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

O(log(|N|) +|L|)

Property Table

?s	:husband	:label	:type	:wife
:JillBiden	:JoeBiden	-	-	-
:JoeBiden	-	"JoeBiden"@en	:POTUS	:JillBiden
. . .	. . .	. . .	. . .	. . .



Design Choice 1

Subdivision

reduce search space by fragmenting data

What's Best ?

Representation can be "**compatible**" to an **access pattern in 3 ways:**

- **seek compatible:** the first result can be retrieved in a single random access
- **sequence compatible:** the remaining results can be computed without random access
- **selection compatible:** if no excess results are retrieved by the computation

```
SELECT ?w
WHERE {
      ?w :label: ?l .

}
```

O(|E|)

O(|E|)

List of triples

?s	?p	?o
:JoeBiden	:label	"JoeBiden"@en
:JoeBiden	:type	:POTUS
:JillBiden	:husband	:JoeBiden
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

O(|E|)

O(log(|E|))

SORTED List of triples

?s	?p	?o
:JillBiden	:husband	:JoeBiden
:JoeBiden	:label	"JoeBiden"@en
:JoeBiden	:type	:POTUS
:JoeBiden	:wife	:JillBiden
. . .	. . .	. . .

O(|N|)

O(log(|N|) +|L|)

Property Table

?s	:husband	:label	:type	:wife
:JillBiden	:JoeBiden	-	-	-
:JoeBiden	-	"JoeBiden"@en	:POTUS	:JillBiden
. . .	. . .	. . .	. . .	. . .

O(|E_ℓ|)

O(|E_ℓ|)

Property Tables

:husband

?s	?o
:JillBiden	:JoeBiden
. . .	. . .

:label

?s	?o
:JoeBiden	"JoeBiden"@en
. . .	. . .

:type

?s	?o
:JoeBiden	:POTUS
. . .	. . .

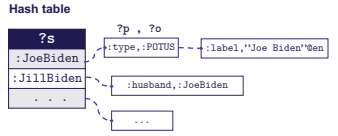
:wife

?s	?o
:JoeBiden	:JillBiden
. . .	. . .

O(|E|)

O(|E_n|)

Hash table



Design Choice 1

Subdivision

reduce search space by fragmenting data

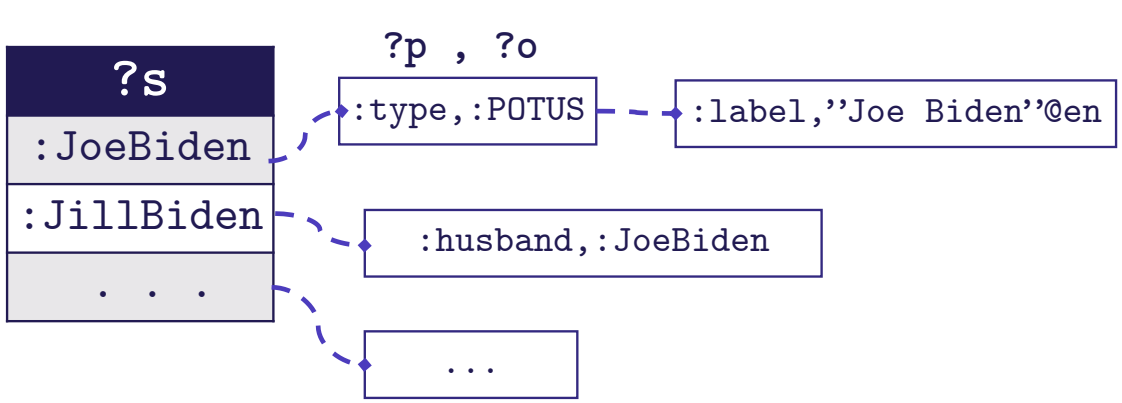
Multiple Representations

$O(|E|)$

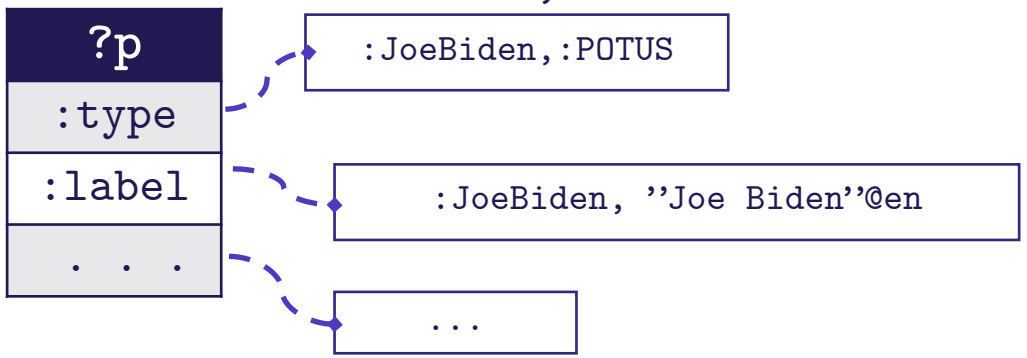
$O(|E_n|)$

```
SELECT ?w
WHERE {
      ?w :label: ?l .
      ?w :type ?t .
}
```

Hash table



$O(|E_\ell|)$



Design Choice 2

Redundancy

add copied of data with compatible data access methods

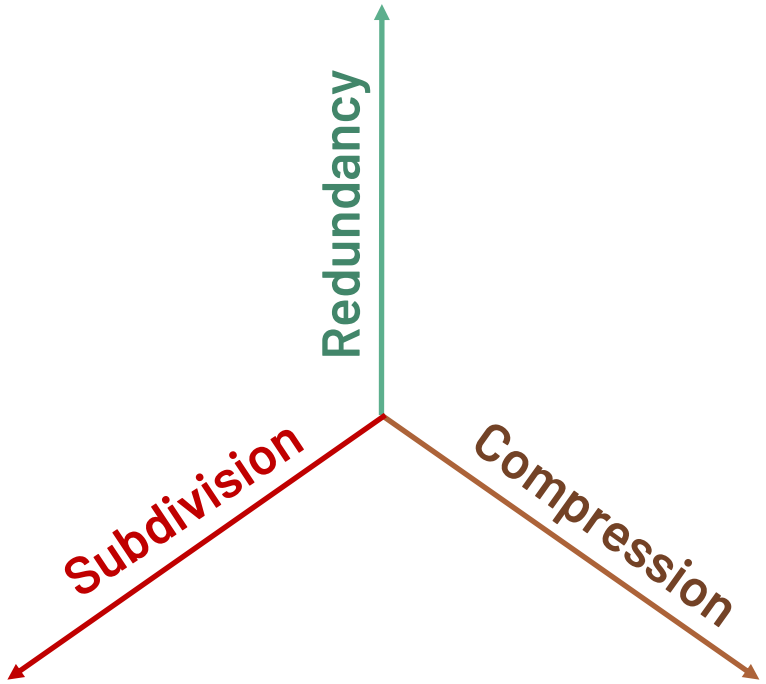
The SCR Design Space

Design Choice 3

Compression

compress data representation to reduce bytes transferred

e.g., replace IRIs with IDs



A design space for RDF data representations 355

Table 2 Summary of data representation design space axes

Axis	Minimal Extreme	Maximal Extreme	Positive Effect	Negative Effect
Subdivision	Unsorted file	Pointers between every related data item	↓ # unneeded data items	↑ # random seeks
Compression	No compression	Compress all subdivisions in all representations	↓ # read bytes	↑ Decompression cost
Redundancy	Single representation	One representation for each possible BGP	↓ # random seeks	↑ maintenance cost, storage cost, query optimization time

The SCR Design Space

The **SCR** design space analysis provides:

- detailed analysis of ***access patterns***
- ***cost model*** *to evaluate the* ***compatibility*** *of different data representations*
- *Analysis of* ***prevalence of access patterns*** *in existing workloads*
- ***opportunity for identifying unexplored solutions*** *(automatically?)*

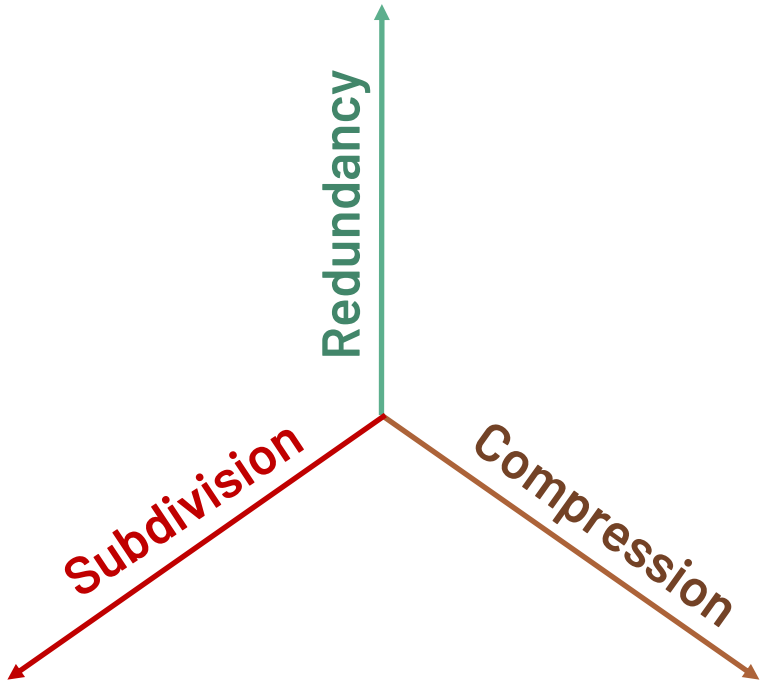


Table 16 Prevalence of access patterns in popular RDF benchmarks

		Constants							Filter			Traversal						
Set	#	S	P	O	SP	PO	SO	SPO	[ ]	[ )	Sp*	S·O	O·S	SkO	OkS	SP*O	OP*S	All
BioBench	22	0.00	0.77	0.00	0.14	0.86	0.00	0.00	0.00	0.00	0.14	0.77	0.86	0.64	0.36	0.00	0.00	0.86
WikiData	46	0.00	0.93	0.02	0.00	0.83	0.00	0.00	0.11	0.15	0.11	0.89	0.80	0.22	0.26	0.04	0.50	0.43
ComplexQ	3443	0.00	0.88	0.00	0.34	0.96	0.00	0.00	0.00	0.02	0.98	1.00	0.96	0.38	0.40	0.00	0.00	0.0*
SWDF	64030	0.24	0.23	0.01	0.31	0.03	0.0*	0.10	0.00	0.00	0.0*	0.56	0.05	0.21	0.0*	0.00	0.00	0.23
DBpedia	169721	0.05	0.69	0.03	0.19	0.71	0.0*	0.0*	0.0*	0.01	0.64	0.87	0.74	0.56	0.07	0.0*	0.0*	0.78

Lissandrini et al. 2022