

МИХАИЛ ГАЛКИН

**ОБРАБОТКА И ОПТИМИЗАЦИЯ
SPARQL-ЗАПРОСОВ К
СЕМАНТИЧЕСКИМ RDF ХРАНИЛИЩАМ**

ABOUT

- ▶ Double-degree PhD student at the University of Bonn and ITMO University
 - ▶ Smart Data Analytics Department @ Uni Bonn
- ▶ Researcher at Fraunhofer IAIS
 - ▶ Enterprise Information Systems Department @ IAIS
- ▶ 19 publications (14 conference + 3 workshops + 2 journals)
 - ▶ Joint task force on semantic data management:
Maria-Esther Vidal, Ioanna Lytra, Maribel Acosta, Kemele M. Endris,
Mohammed N. Mami

OUTLINE

- ▶ Intro to RDF + SPARQL
- ▶ Motivation: Semantic Data Integration
- ▶ MULDER: A Federated Query Engine
- ▶ SMJoin: A Multi-way Join Operator for SPARQL
- ▶ SJoin: A Semantic Join Operator
- ▶ MSimJoin: A Multi-way Semantic Join Operator

RDF 101

RDF 101

- ▶ Resource Description Framework - a graph-based model with formal semantics encoded in vocabularies

RDF 101

- ▶ Resource Description Framework - a graph-based model with formal semantics encoded in vocabularies

<subject>

<predicate>

<object>

RDF 101

- ▶ Resource Description Framework - a graph-based model with formal semantics encoded in vocabularies

<subject>



MSU

<predicate>



isA

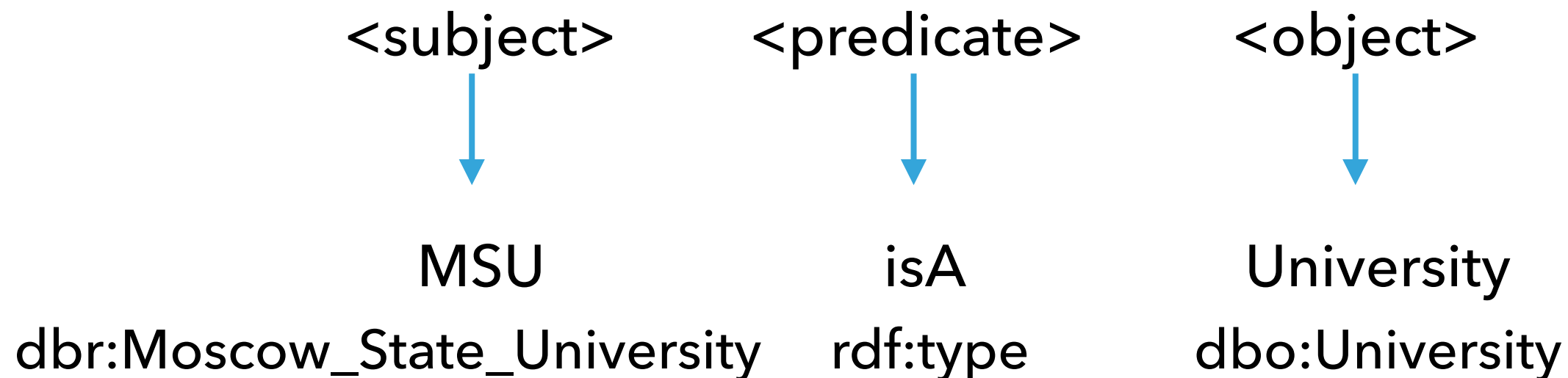
<object>



University

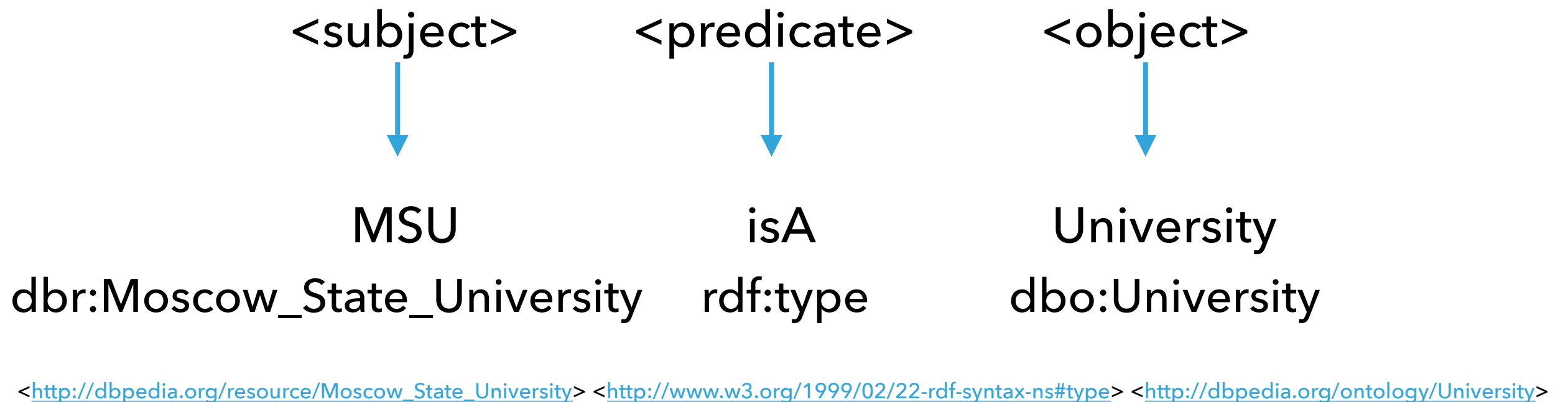
RDF 101

- ▶ Resource Description Framework - a graph-based model with formal semantics encoded in vocabularies



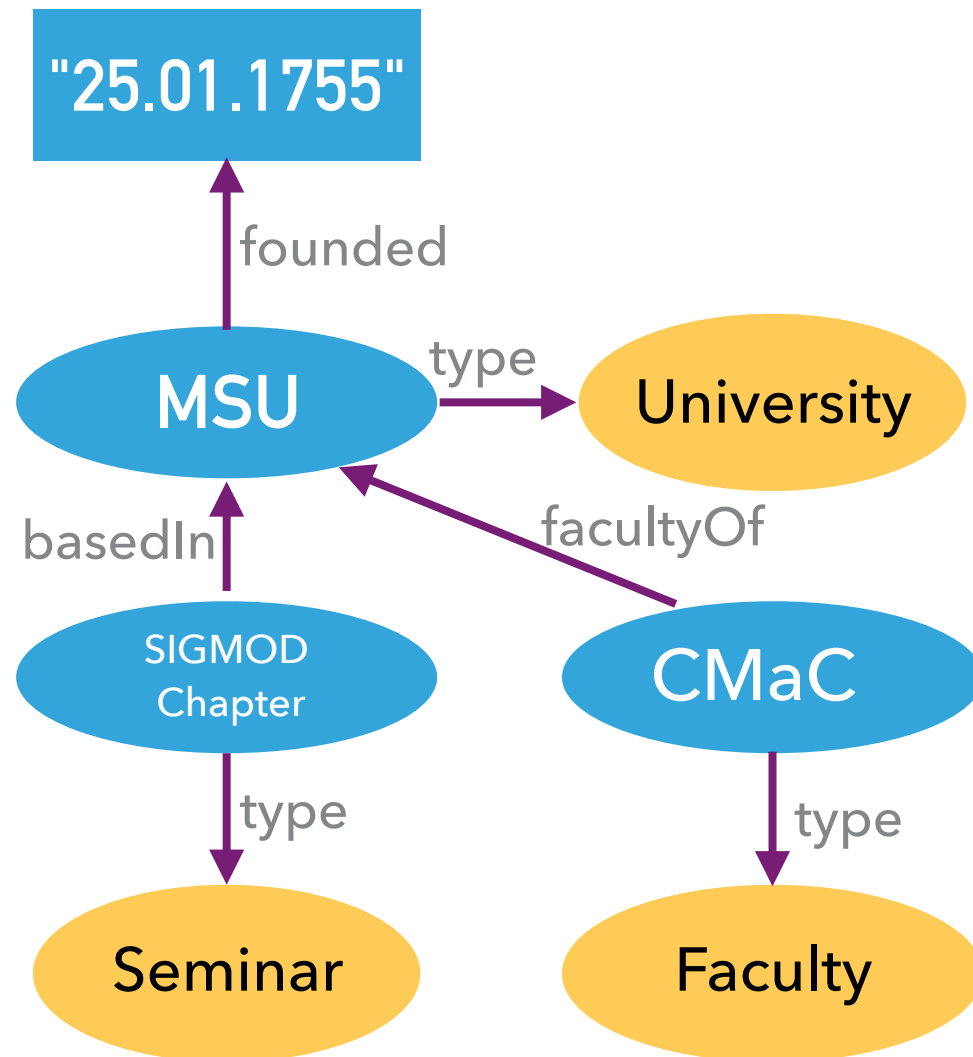
RDF 101

- ▶ Resource Description Framework - a graph-based model with formal semantics encoded in vocabularies



RDF 101 - RDFS & OWL SEMANTICS

RDF 101 - RDFS & OWL SEMANTICS

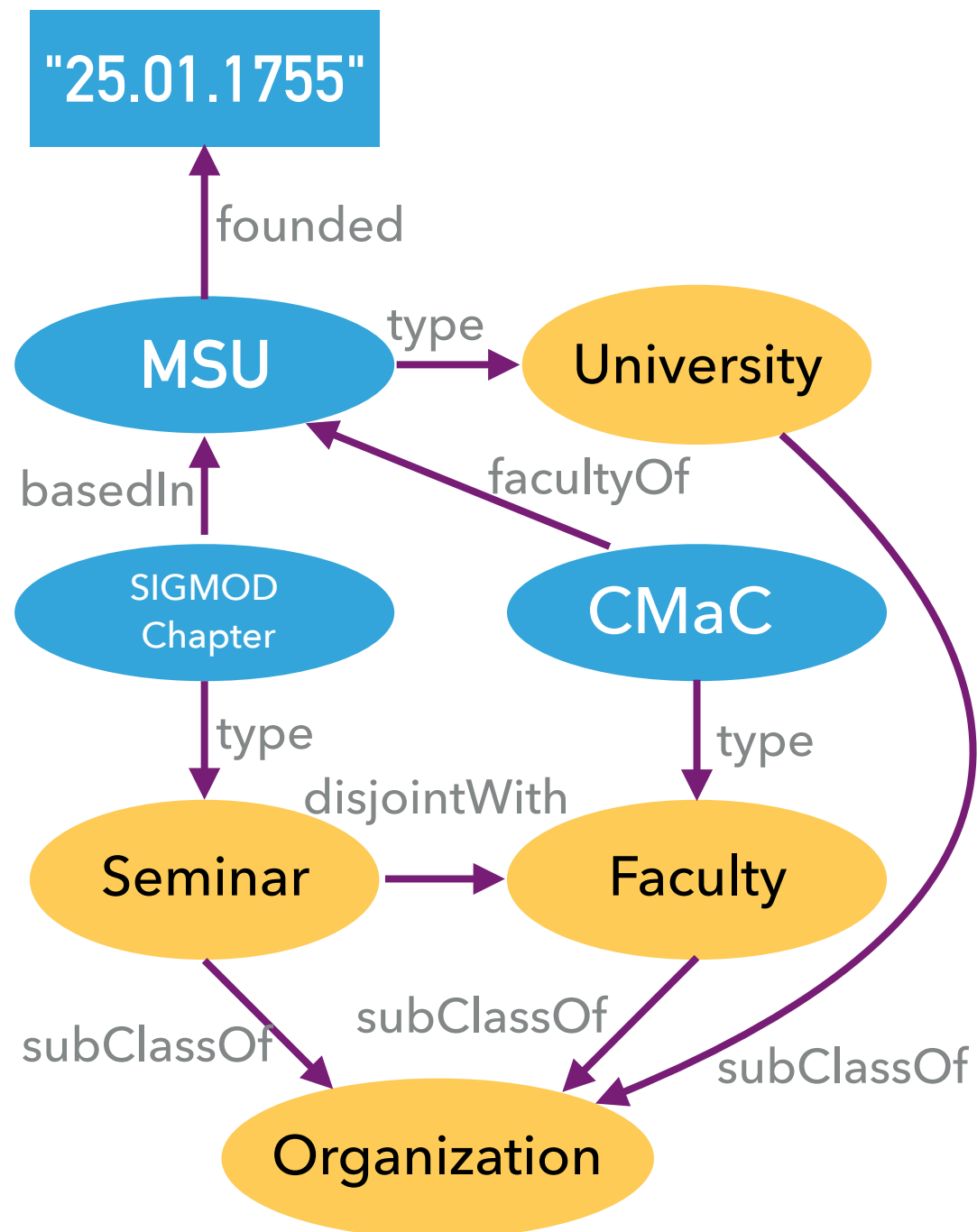


MSU type University;
founded "25.01.1755".

SIGMOD_Chapter type Seminar;
basedIn MSU.

CMaC type Faculty;
facultyOf MSU.

RDF 101 - RDFS & OWL SEMANTICS



MSU type University;
founded "25.01.1755".

SIGMOD_Chapter type Seminar;
basedIn MSU.

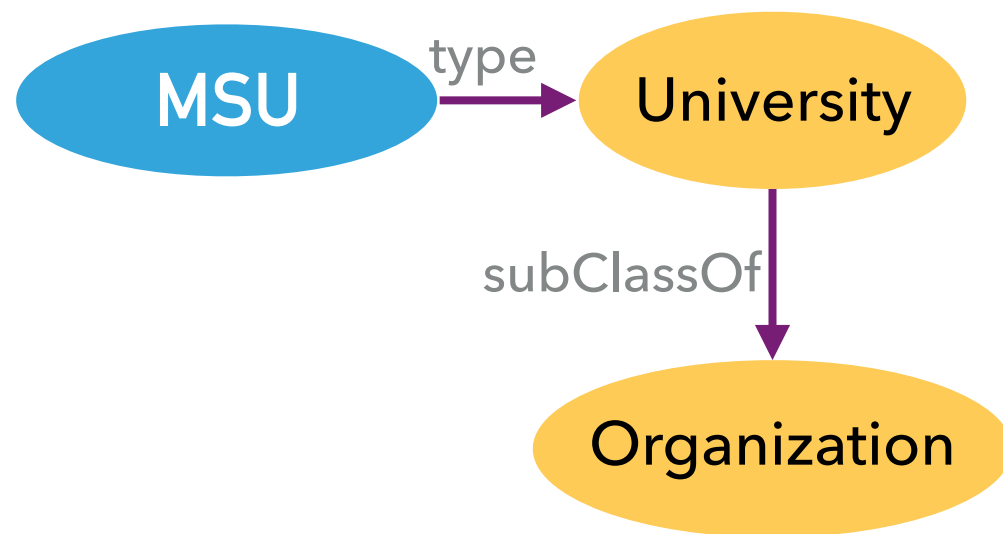
CMaC type Faculty;
facultyOf MSU.

Seminar rdfs:subClassOf Organization.
Faculty rdfs:subClassOf Organization.
University rdfs:subClassOf Organization.

Seminar owl:disjointWith Faculty.

REASONING

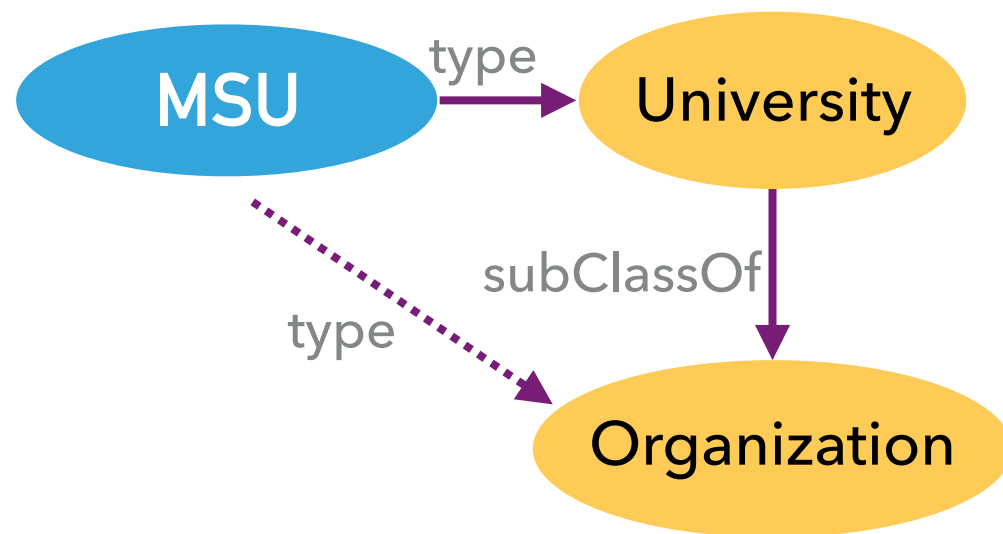
REASONING



MSU type University;
 founded "25.01.1755".

University rdfs:subClassOf Organization.

REASONING

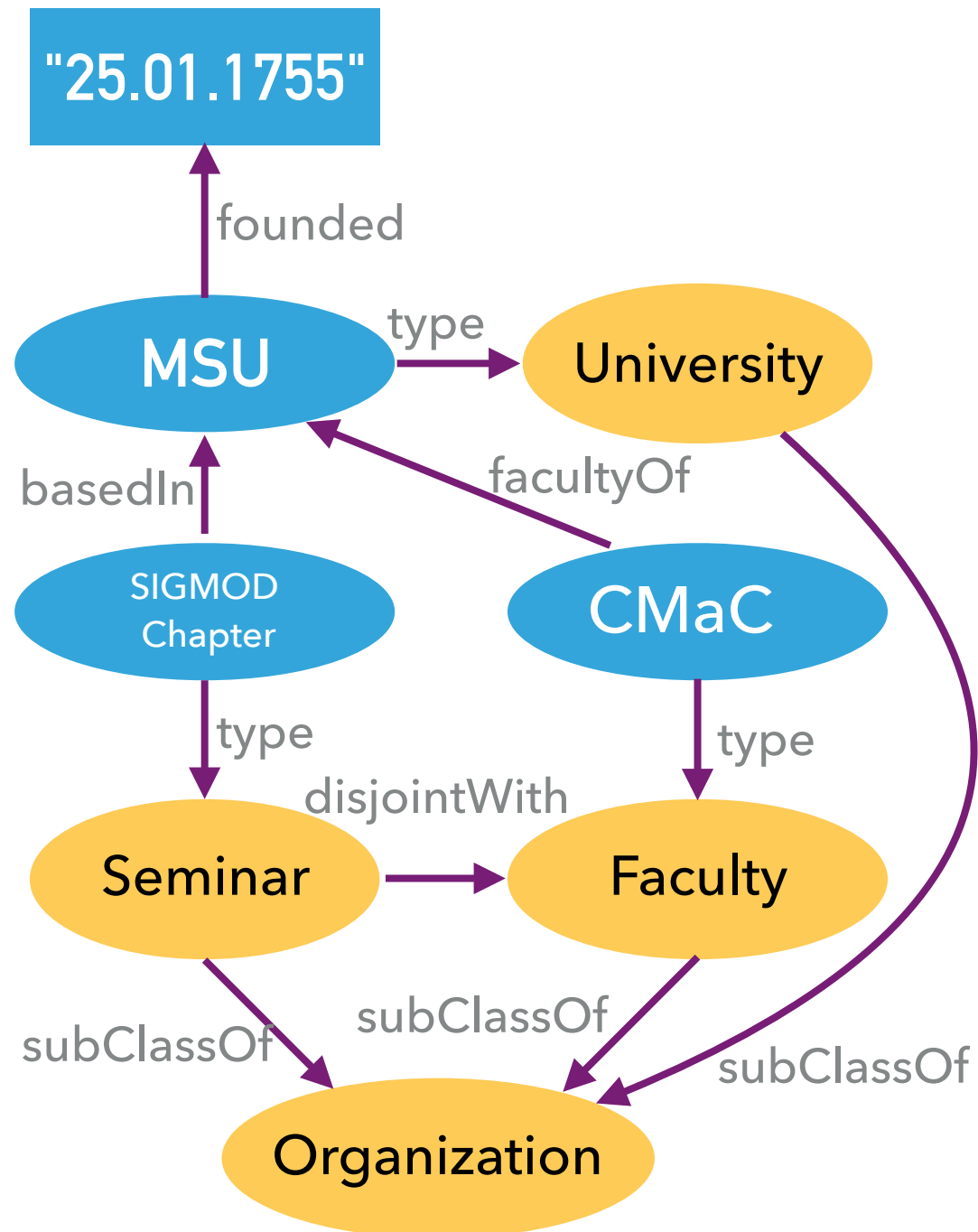


MSU type University;
founded "25.01.1755".

University rdfs:subClassOf Organization.

MSU type Organization.

SPARQL QUERIES

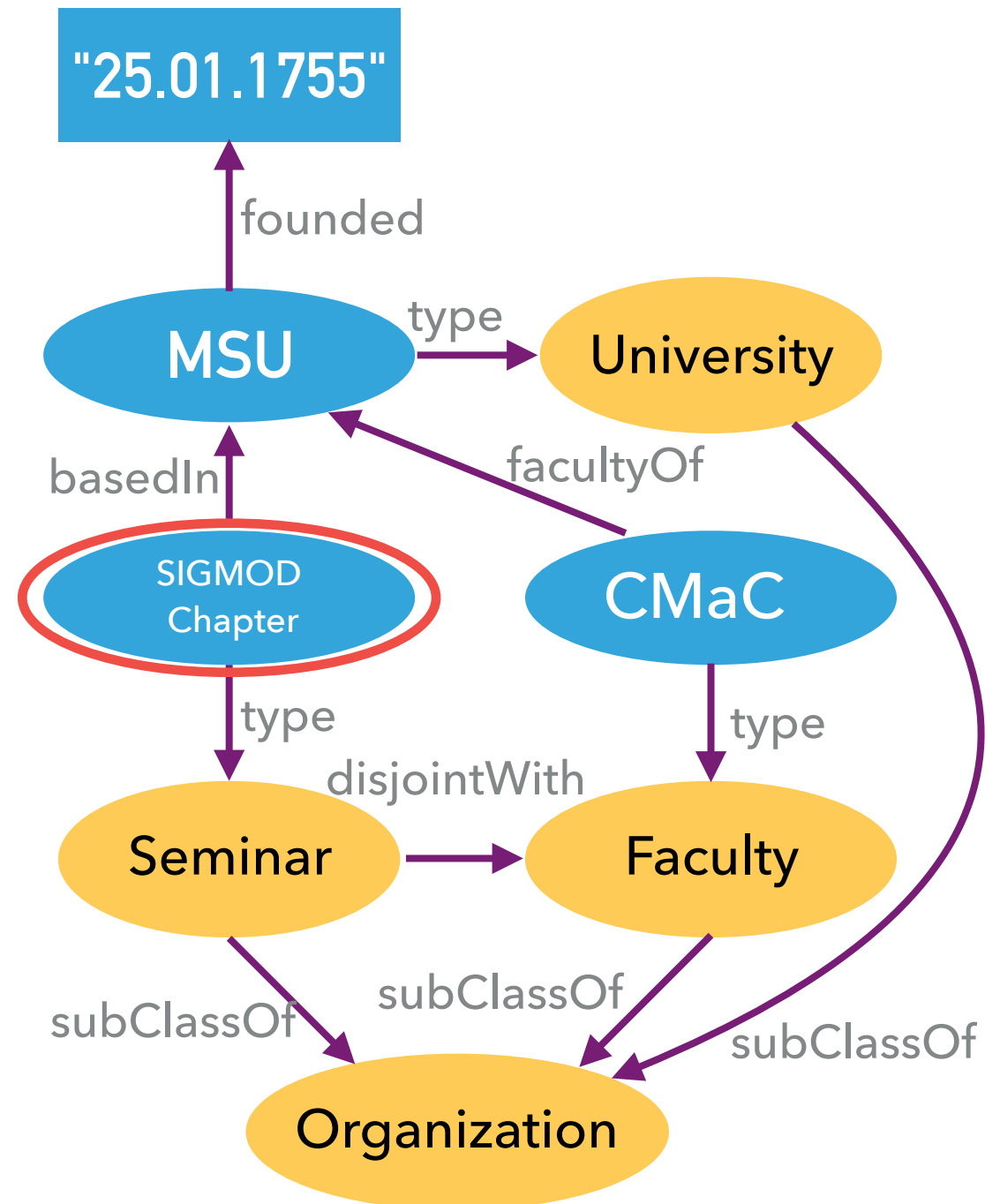


```

SELECT ?date WHERE {
  ?seminar type Seminar .
  ?seminar basedIn ?uni .
  ?uni founded ?date .
}

```

SPARQL QUERIES



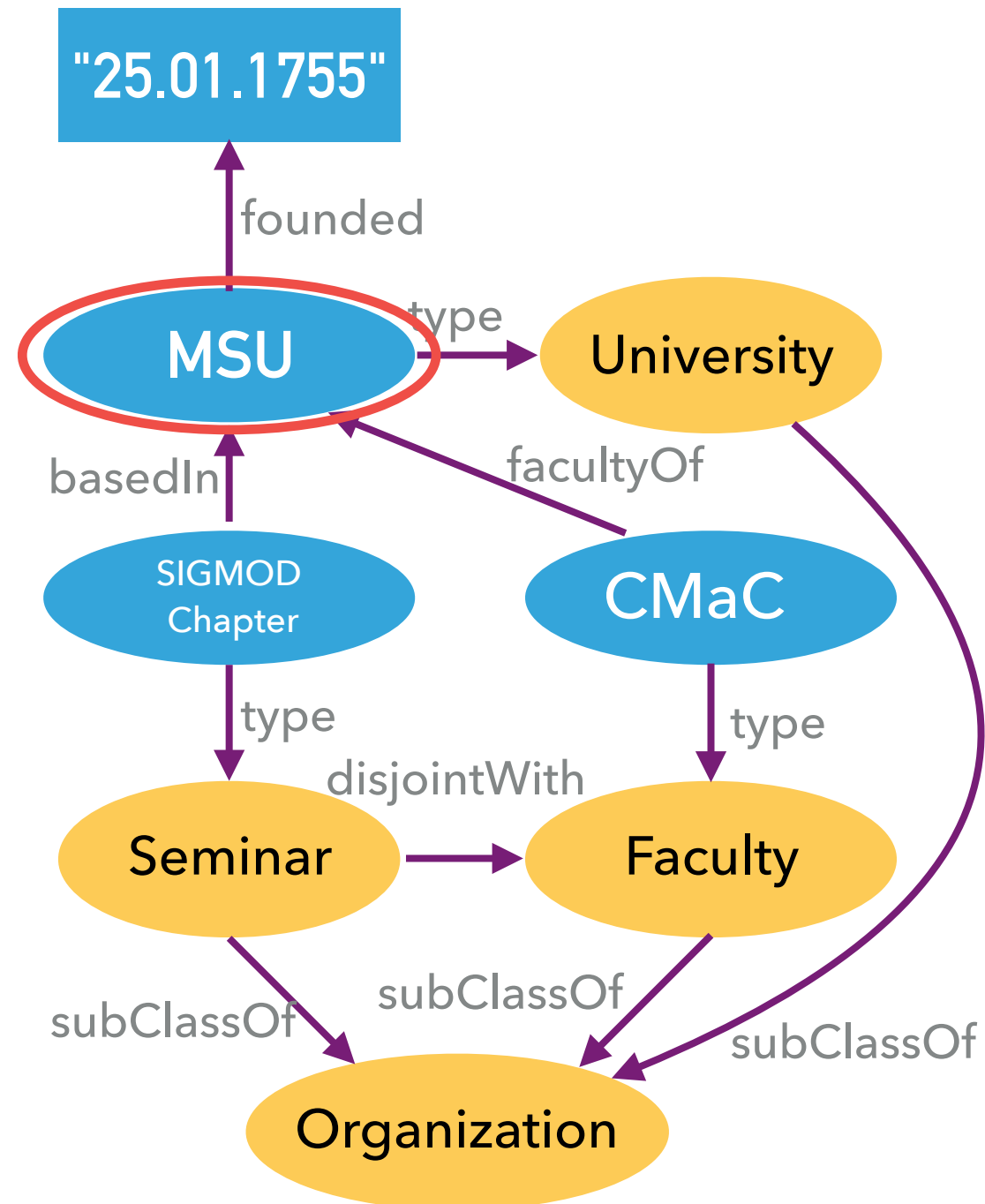
SELECT ?date WHERE {

?seminar type Seminar .

?seminar basedIn ?uni .

?uni founded ?date . }

SPARQL QUERIES

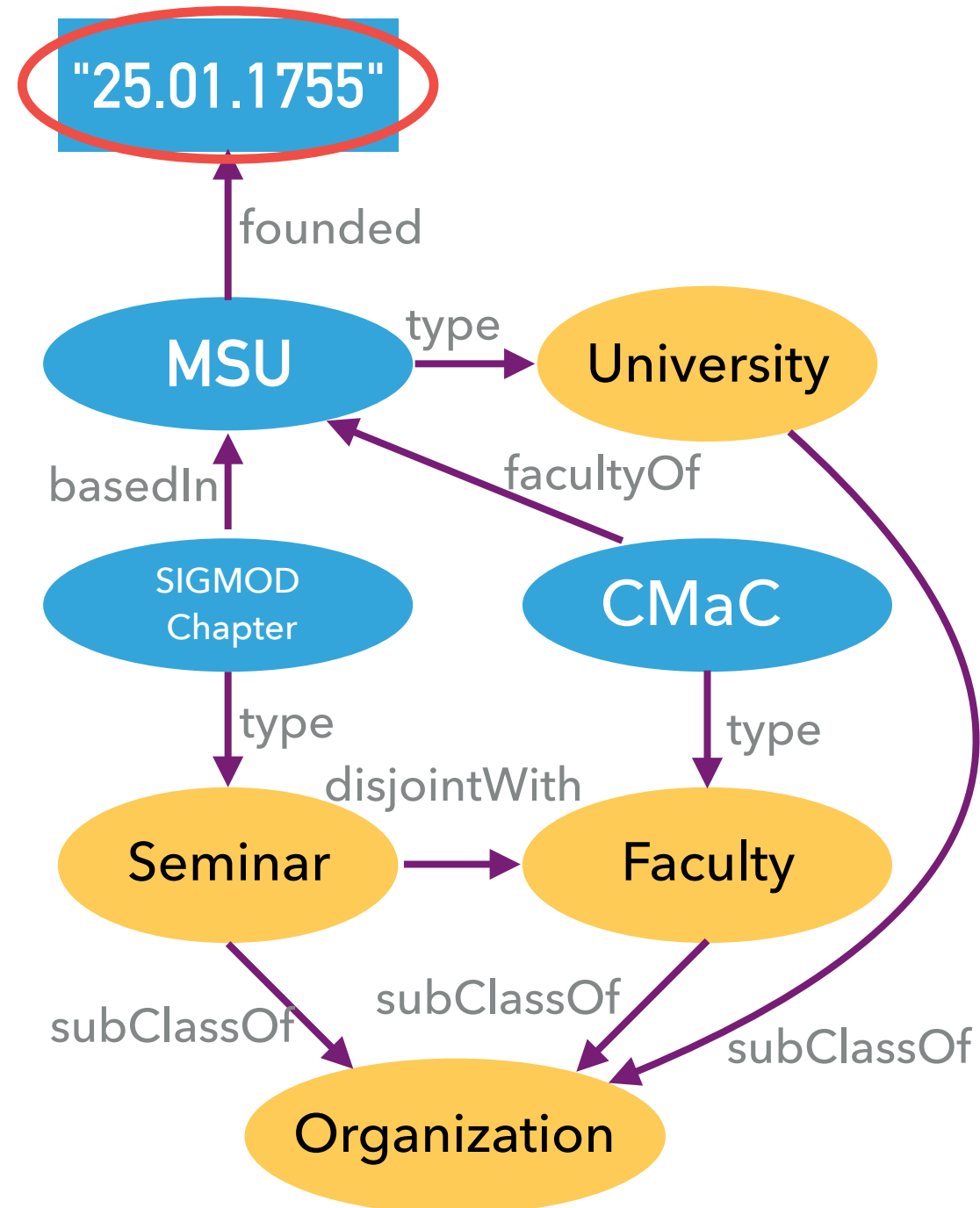


```

SELECT ?date WHERE {
  ?seminar type Seminar .
  ?seminar basedIn ?uni .
  ?uni founded ?date .
}

```

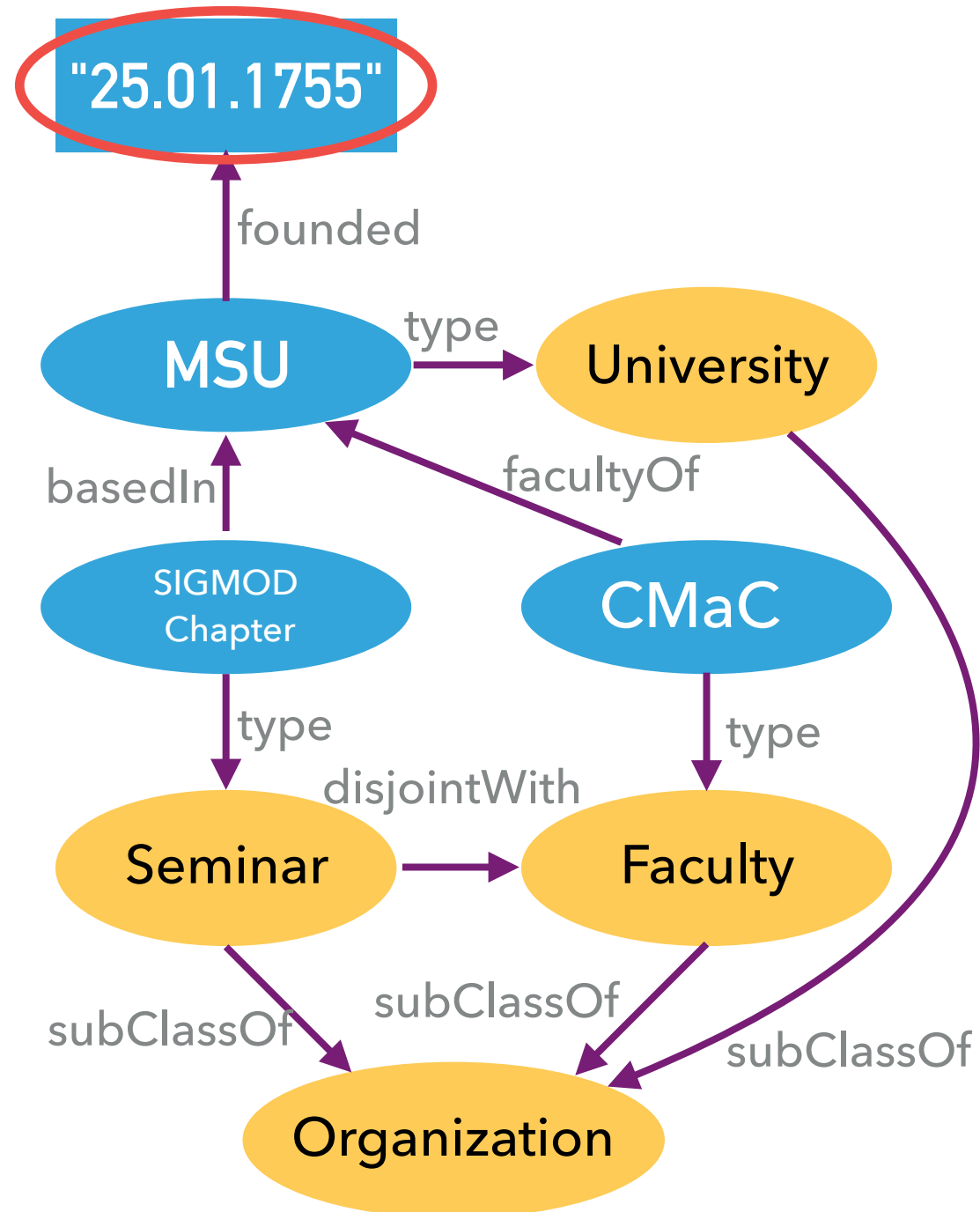
SPARQL QUERIES



```

SELECT ?date WHERE {
  ?seminar type Seminar .
  ?seminar basedIn ?uni .
  ?uni founded ?date . }
  
```

SPARQL QUERIES



```

SELECT ?date WHERE {
  ?seminar type Seminar .
  ?seminar basedIn ?uni .
  ?uni founded ?date .
}

```

?date = "25.01.1755"

RESEARCH MOTIVATION

- ▶ Scalable data integration of numerous heterogeneous sources
 - ▶ Large-scale heterogeneous data integration using RDF molecules:
 - RDF only data : AdPart, DiploCloud
 - Non-RDF data: Hadoop, HBase
 - RDF + Non-RDF: **Semantic Data Lake**
- ▶ A uniform query mechanism does not exist
 - ▶ Semantics of RDF datasets it not considered

SEMANTIC DATA LAKE

- ▶ **Semantic Data Lake** is defined as:

$SDL = \langle O, S, SD \rangle = \langle O, S, \langle \text{Rule}, \text{Access Capability}, \text{Type} \rangle \rangle$

O - Ontology, Global Schema

S - Sources

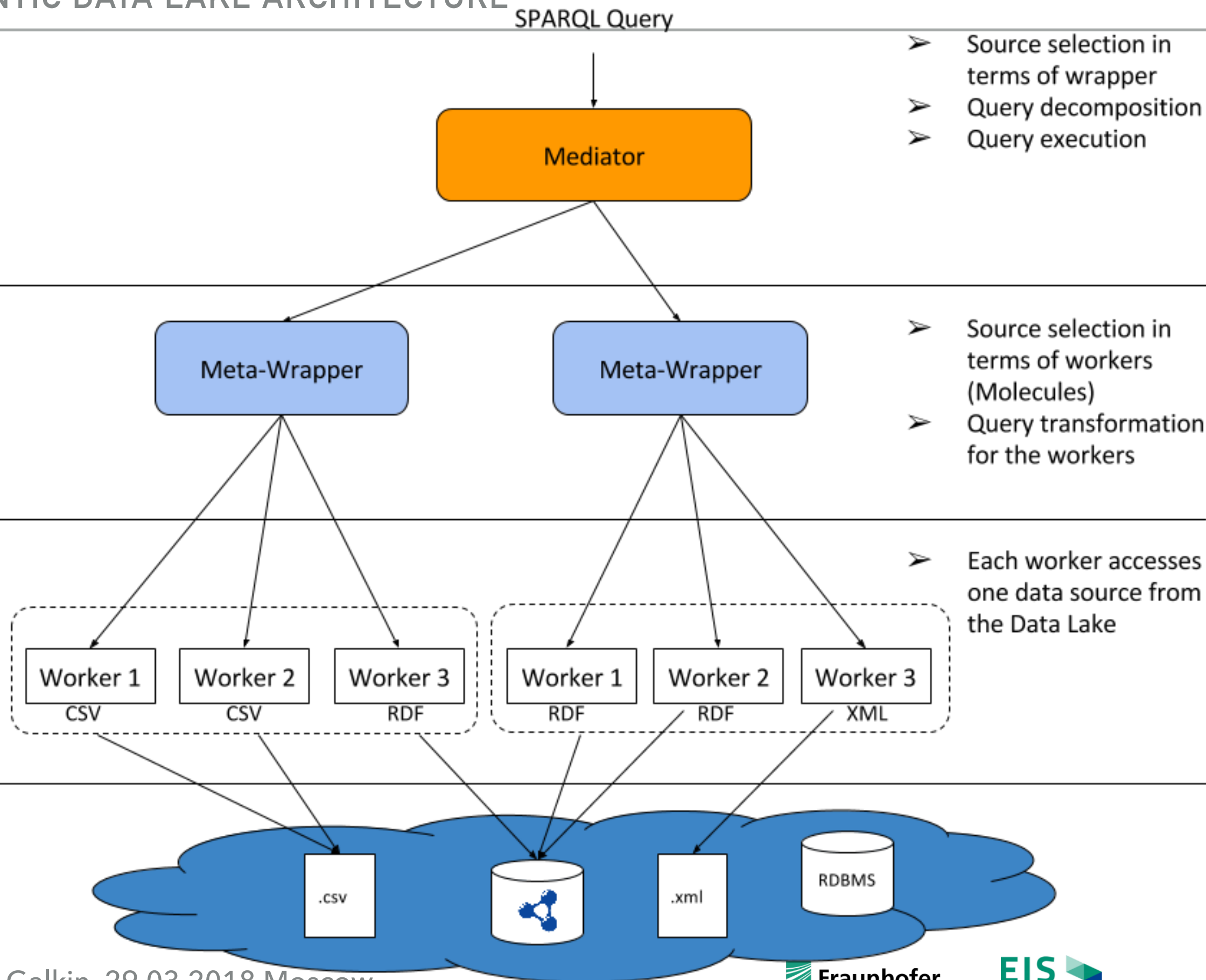
SD - Source Description

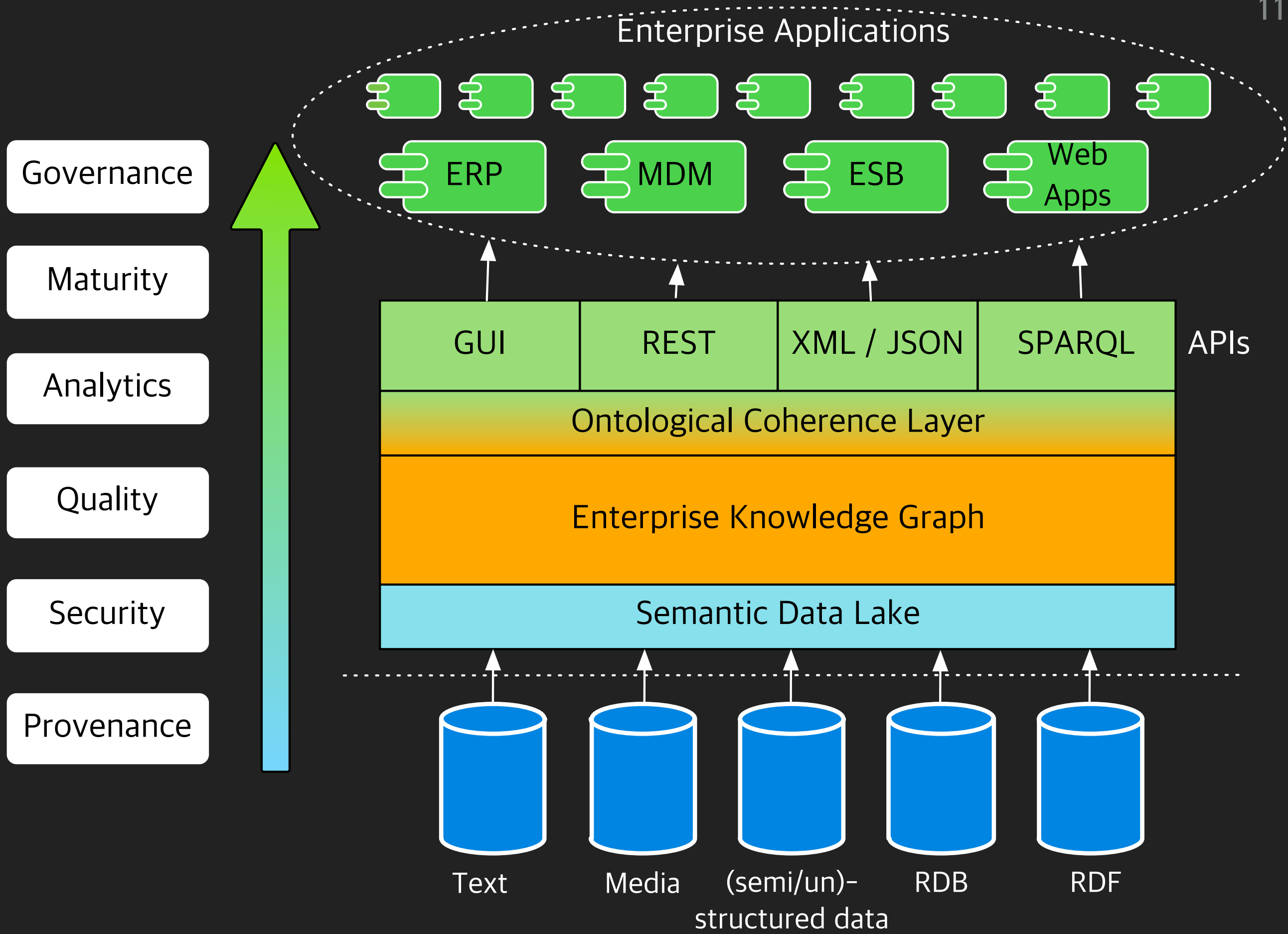
- ▶ **<Source Description>** comprises:

Rule: Mapping Rule among Sources and Ontology

Access Capability: SPARQL, TPF, Wrapper

Type: RDF, RDB, CSV, XML, Plain Text, Multimedia



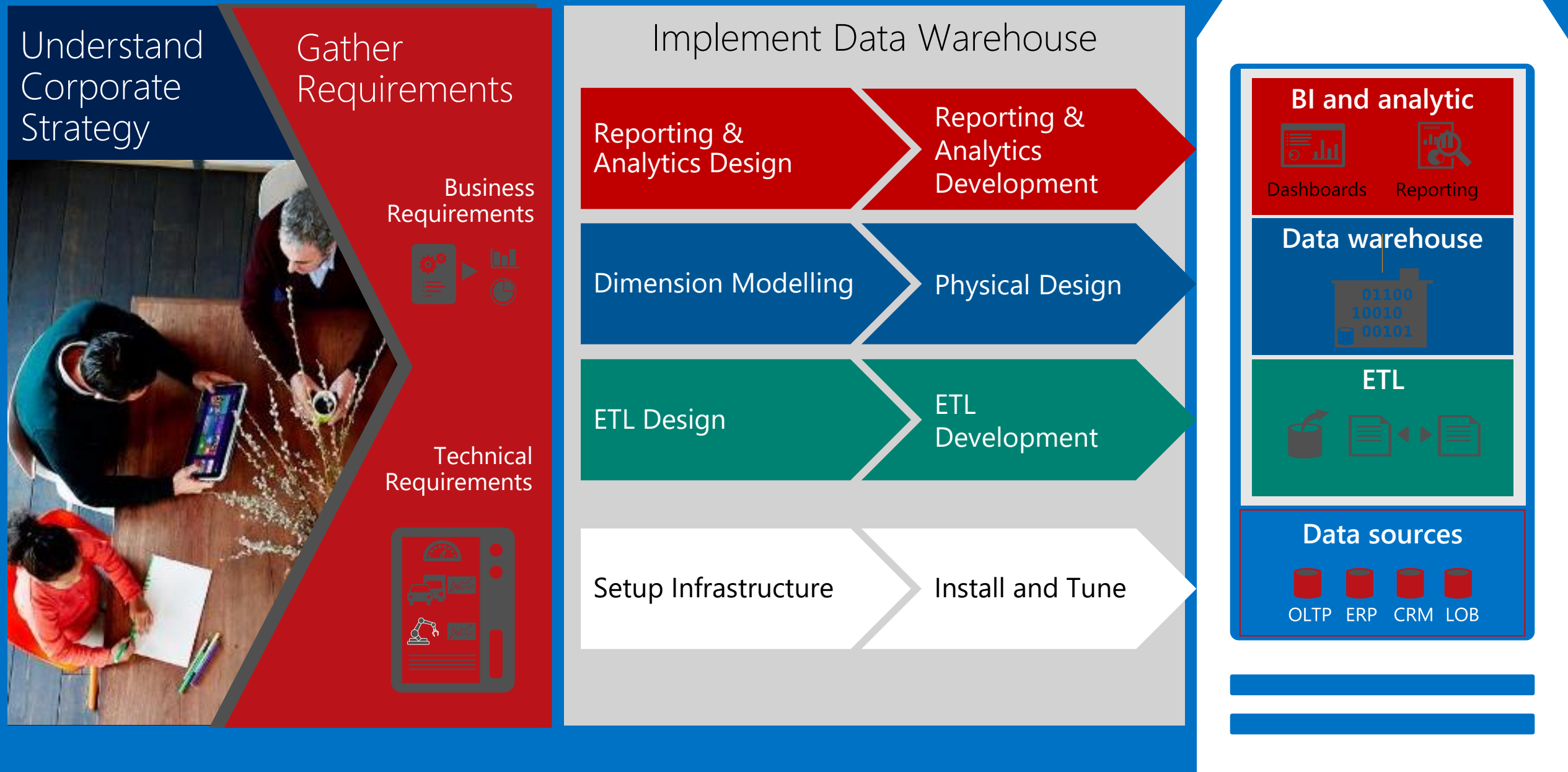


Two Approaches to Information Management for Analytics: Top-Down + Bottom-Up



© Raghu Ramakrishnan @ Microsoft

Data Warehousing Uses A Top-Down Approach



© Raghu Ramakrishnan @ Microsoft

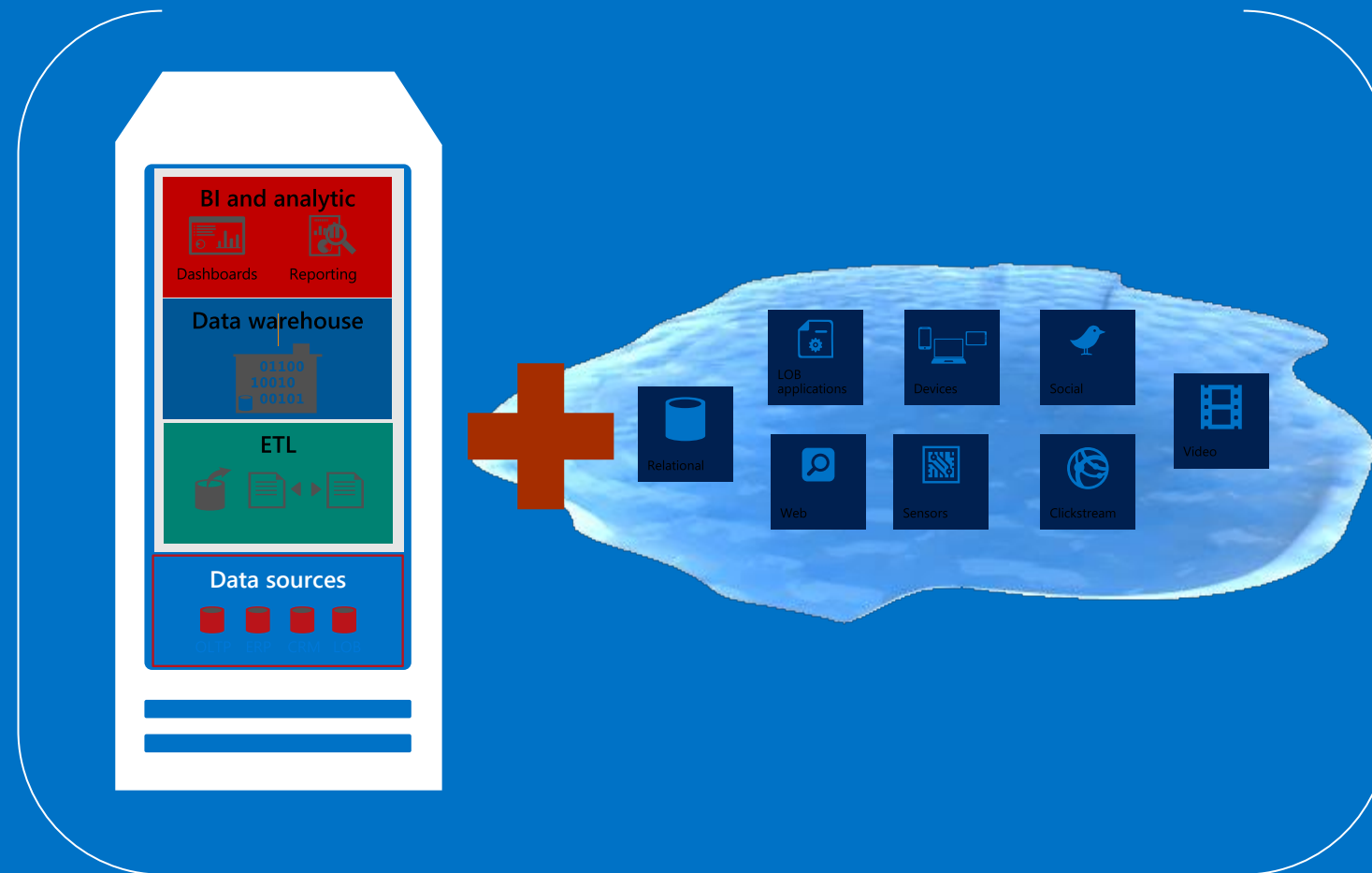
The "Data Lake" Uses A Bottom-Up Approach



© Raghu Ramakrishnan @ Microsoft

Data Lake + Data Warehouse Better Together

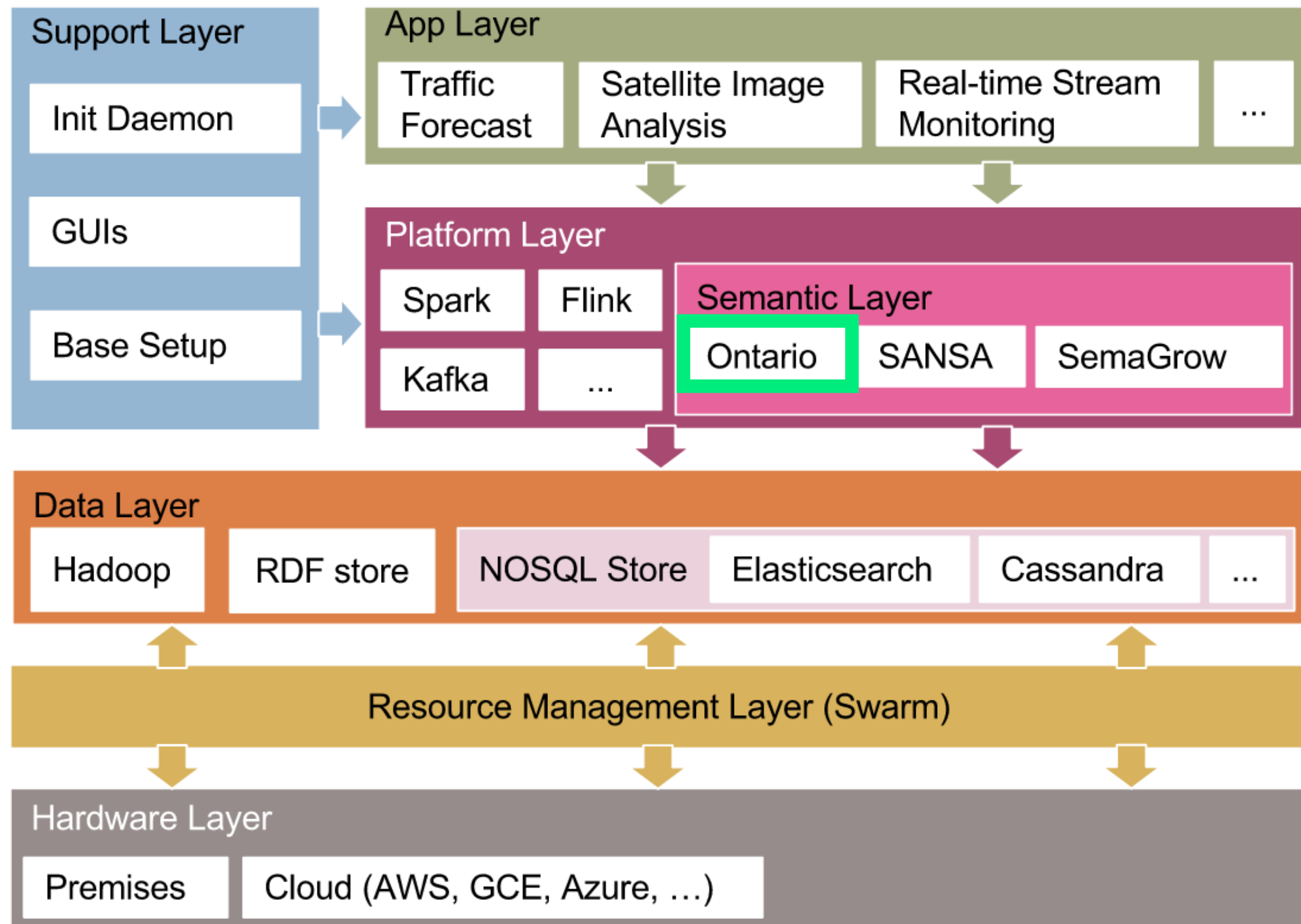
What happened?
What is happening?
Why did it happen?
What are key relationships?



What will happen?
What if?
How risky is it?
What should happen?
What is the best option?
How can I optimize?

© Raghu Ramakrishnan @ Microsoft

ONTARIO - ONTOLOGICAL ARCHITECTURE FOR DATA LAKES



PROBLEMS TO ADDRESS

- **Data Organization:** Build RDF molecules from heterogeneous sources

Source 1	Source 2	Source 3	RDF Molecule
(CSV)	<person>	ex:person123	:MG :name Michael Galkin
Name, LastName	<name> Galkin </name>	a foaf:Person;	:MG :city Bonn
	<city> Bonn </city>	foaf:name 'Michael	:MG :university Uni Bonn
Michael, Galkin	</person>	Galkin'@en;	:MG :sources :Source1
		ex:university	:MG :sources :Source2
		ex:Uni_Bonn .	:MG :sources :Source3

- **Storage:** Clusterize heterogenous data in molecules across distributed environments
- **Querying:** Develop a universal query mechanism

ONTARIO APPLICATIONS - IASIS

EU H2020 project for management and analytics of pharmacogenomics data for planning public health policies.

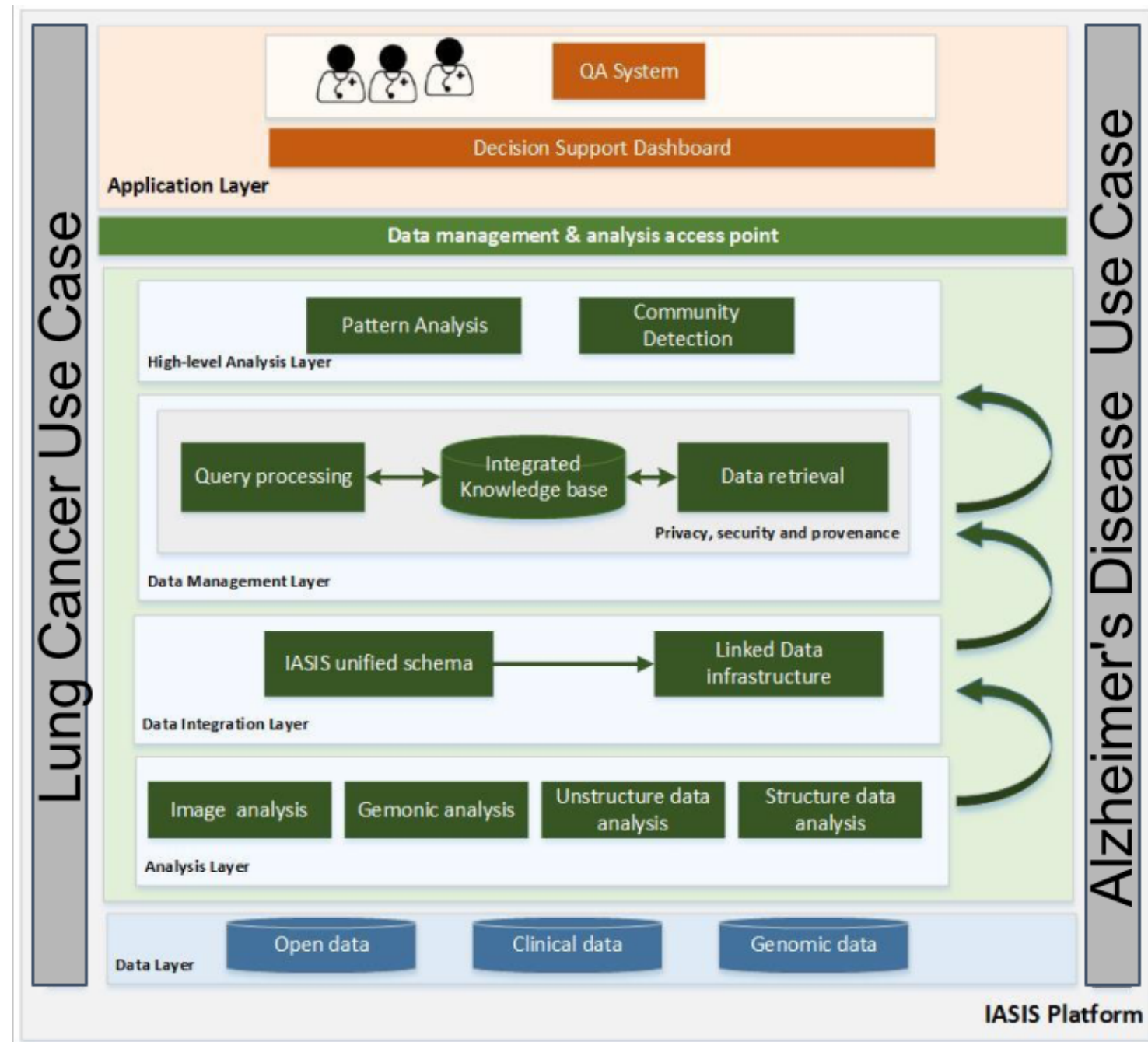
Patient data: molecular, sequencing, electronic records, and images.

Open pharmacogenomics datasets.

Two pilots:

Lung cancer

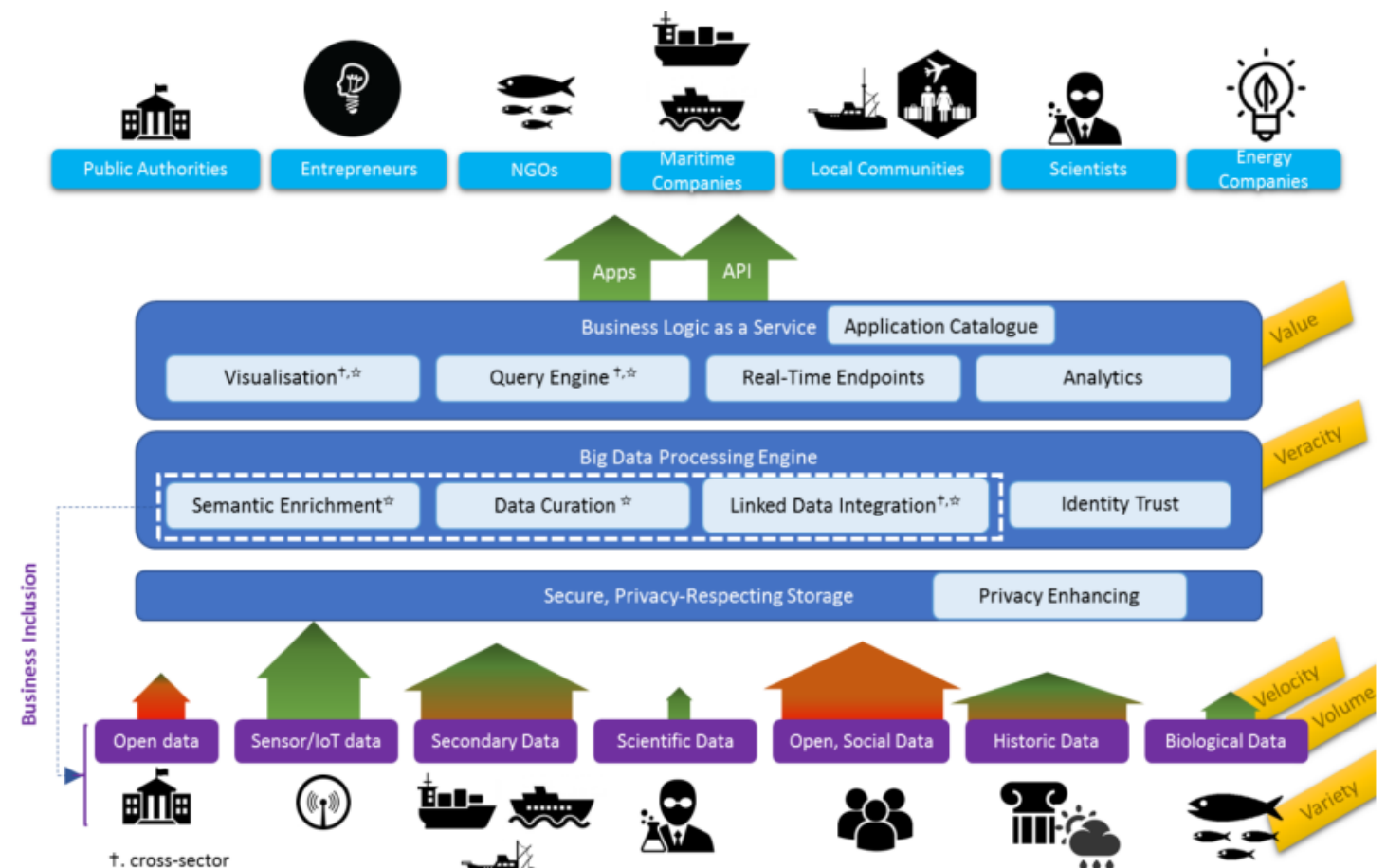
Alzheimer's Disease



ONTARIO APPLICATIONS - BIG DATA OCEAN

EU H2020 project for management and analytics of maritime Big Data.

- ▶ Expected number of sources:
 - ▶ 10,000 datasets
 - ▶ 150TBs
 - ▶ Seven types of sources
- ▶ Four pilots:
 - ▶ Fault prediction and Proactive maintenance
 - ▶ Mare protection
 - ▶ Maritime Security and
 - ▶ Anomaly Detection
 - ▶ Wave Energy Power

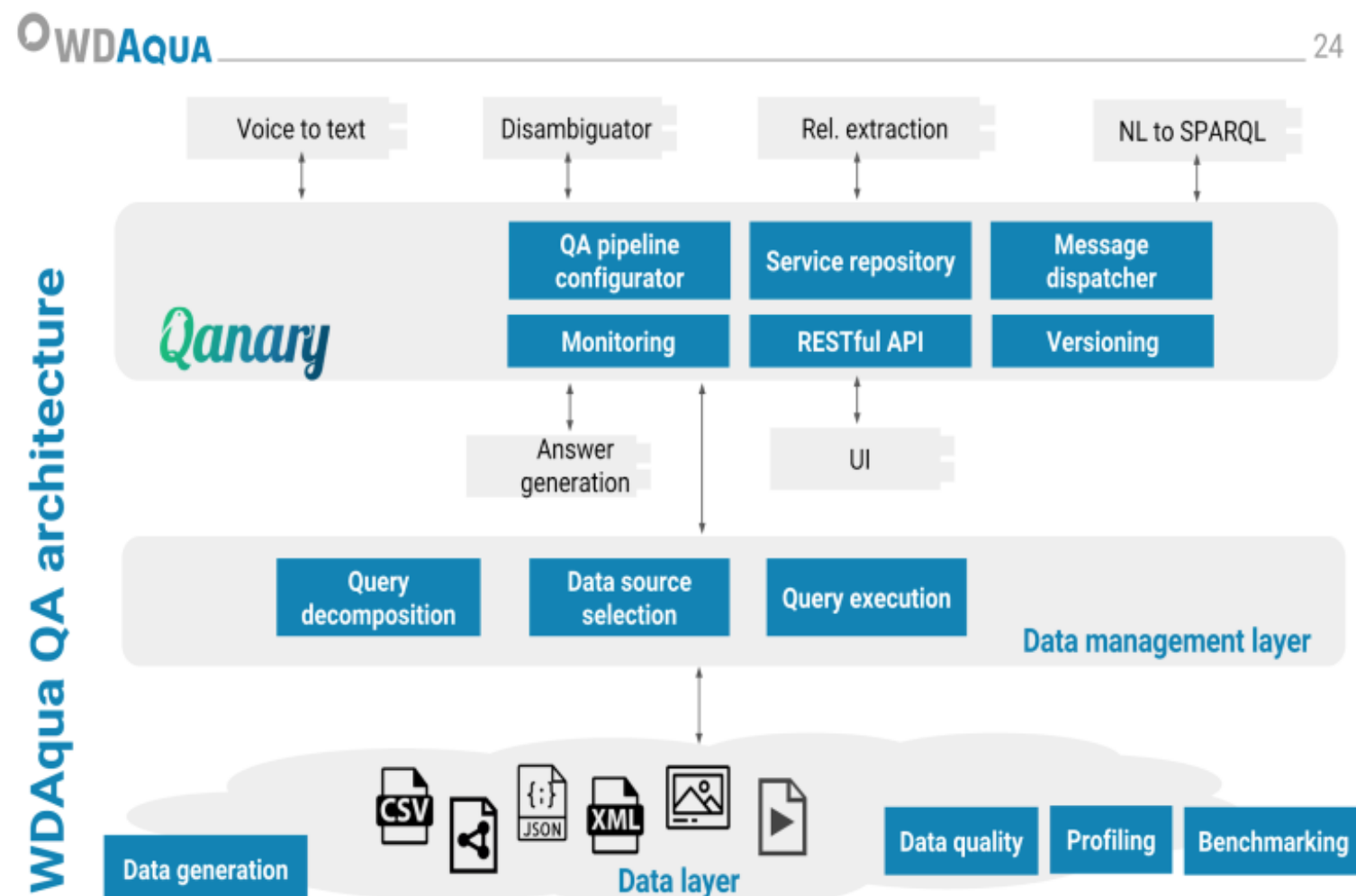


ONTARIO APPLICATIONS - WDAQUA

WDAqua is a Marie Skłodowska Curie Innovative Training Network (ITN)

WDAqua Question Answering (QA) architecture

- ▶ Understands questions expressed in different formats
- ▶ Discovers high-quality datasets suitable for question answering
- ▶ Considers trust, provenance, and data access control
- ▶ Scales QA up to Big Data



MULDER QUERY ENGINE

1. K. Endris, M. Galkin, I. Lytra, M. Mami, M. E. Vidal, S. Auer. *MULDER: Querying the Linked Data Web by Bridging RDF Molecule Templates*. **Best Paper Award @ DEXA 2017**
2. K. Endris, M. Galkin, I. Lytra, M. Mami, M. E. Vidal, S. Auer. *Querying Interlinked Data by Bridging RDF Molecule Templates*. Submitted to the LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems Journal (**Transactions LDKS**)

MOTIVATING EXAMPLE

MOTIVATING EXAMPLE

SELECT DISTINCT ?s WHERE {
t1 ?s foaf:page ?page .
t2 ?s owl:sameAs ?sameas .
t3 ?s geonames:inCountry ?inCountry .
}

SWDF

t1	319
t2	1,112

Geonames

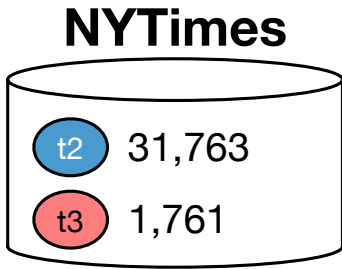
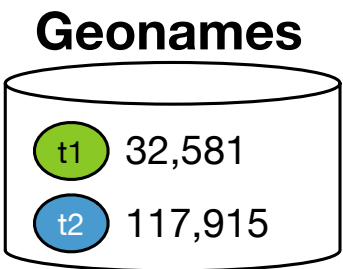
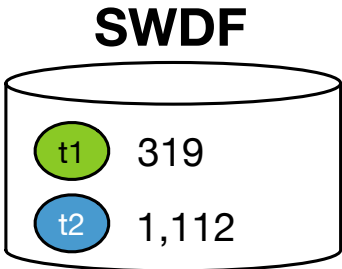
t1	32,581
t2	117,915

NYTimes

t2	31,763
t3	1,761

MOTIVATING EXAMPLE

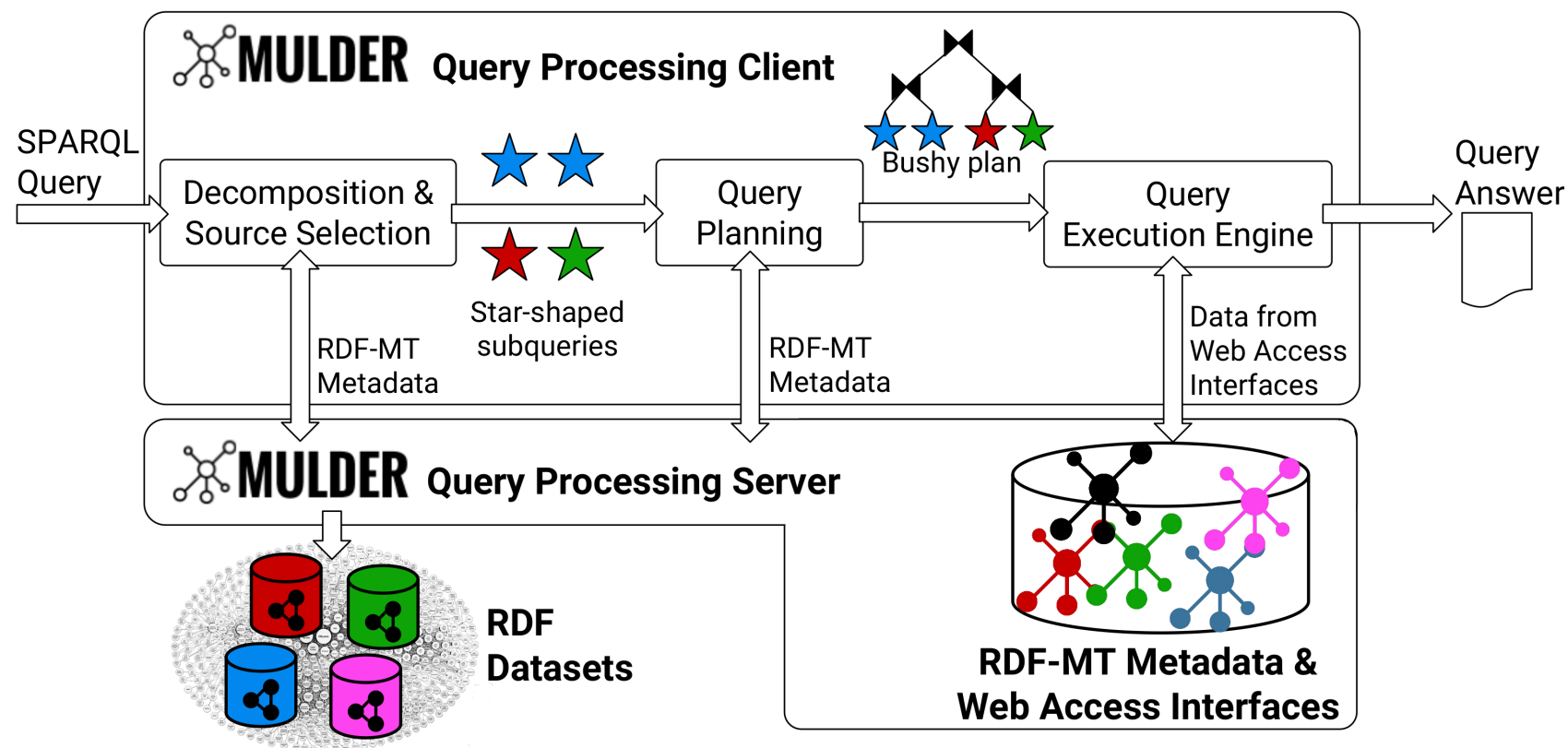
SELECT DISTINCT ?s WHERE {
t1 ?s foaf:page ?page .
t2 ?s owl:sameAs ?sameas .
t3 ?s geonames:inCountry ?inCountry .}



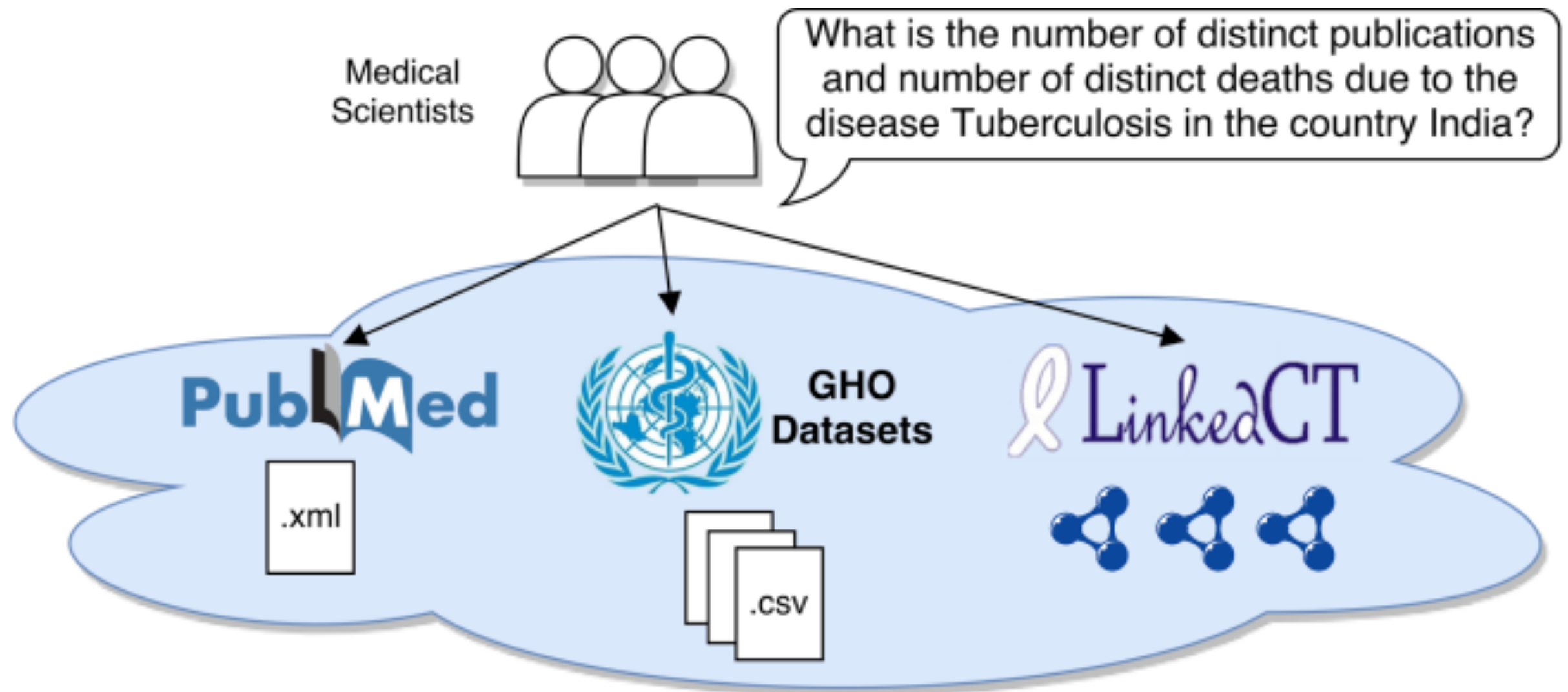
	FedX	ANAPSID SSGS	ANAPSID SSGM	Join	# triples
	t1 @ SWDF, Geo t2 @ SWDF, Geo, NYTimes t3 @ NYTimes	t1 t2 @ SWDF t3 @ NYTimes	t1 @ SWDF, Geo t2 t3 @ NYTimes	t1 swdf t2 swdf t1 geo t2 geo t2 nyt t3 nyt t1 geo t2 geo t3 nyt t1 geo t2 nyt t3 nyt t1 geo t2 geo t2 nyt t3 nyt	371 524 1,249 8 19 20
# sec	239.4	0.338	88.9		
# triples	20	0	19		

MULDER - MOLECULE-BASED DECOMPOSITION FOR RDF

- ▶ **MULDER** federated query engine
- ▶ Leverages RDF molecule templates
- ▶ Supports **SPARQL** and **TPF** clients
- ▶ Plans to add the support for arbitrary data formats apart from RDF
 - CSV
 - JSON
 - XML

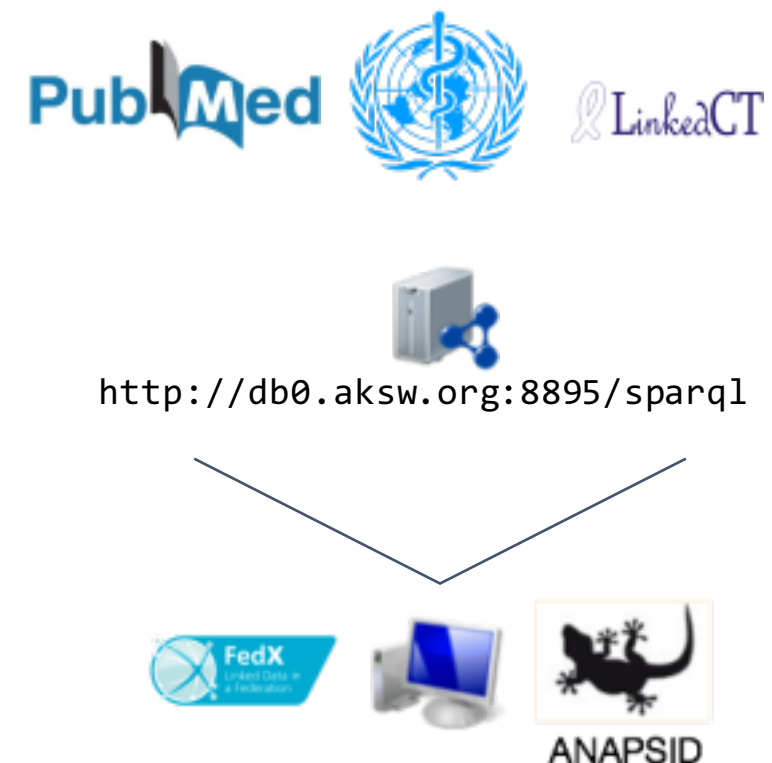


EXAMPLE QUERY



EXAMPLE QUERY

```
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX att: <http://purl.org/linked-data/sdmx/2009/attribute#>
PREFIX eg: <http://example.com/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ct: <http://data.linkedct.org/resource/linkedct/>
PREFIX gho: <http://ghodata/>
PREFIX redd: <http://redd.aksw.org/>
SELECT count(distinct(?publication)) AS ?no_of_publications count(?deaths) AS ?no_of_deaths
WHERE {
    ?item a qb:Observation .
    ?item gho:Country ?country .
    ?item gho:Disease ?disease .
    ?item att:unitMeasure gho:Measure .
    ?item eg:incidence ?deaths .
    ?country rdfs:label "India" .
    ?disease rdfs:label "Tuberculosis".
    ?trial a ct:trials .
    ?trial ct:condition ?condition .
    ?trial ct:location ?location .
    ?trial ct:reference ?publication.
    ?condition owl:sameAs ?disease .
    ?location redd:locatedIn ?country .
    ?publication ct:citation ?citation.
}
```



EXAMPLE QUERY

```

PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX att: <http://purl.org/linked-data/sdmx/2009/attribute#>
PREFIX eg: <http://example.com/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ct: <http://data.linkedct.org/resource/linkedct/>
PREFIX gho: <http://ghodata/>
PREFIX redd: <http://redd.aksw.org/>
SELECT count(distinct(?publication)) AS ?no_of_publications count(?deaths) AS ?no_of_deaths
WHERE {

```

```

?item a qb:Observation .
?item gho:Country ?country .
?item gho:Disease ?disease .
?item att:unitMeasure gho:Measure .
?item eg:incidence ?deaths .

```

```

?country rdfs:label "India" .
?disease rdfs:label "Tuberculosis".

```

```

?trial a ct:trials .
?trial ct:condition ?condition .
?trial ct:location ?location .
?trial ct:reference ?publication.
?condition owl:sameAs ?disease .
?location redd:locatedIn ?country .
?publication ct:citation ?citation.

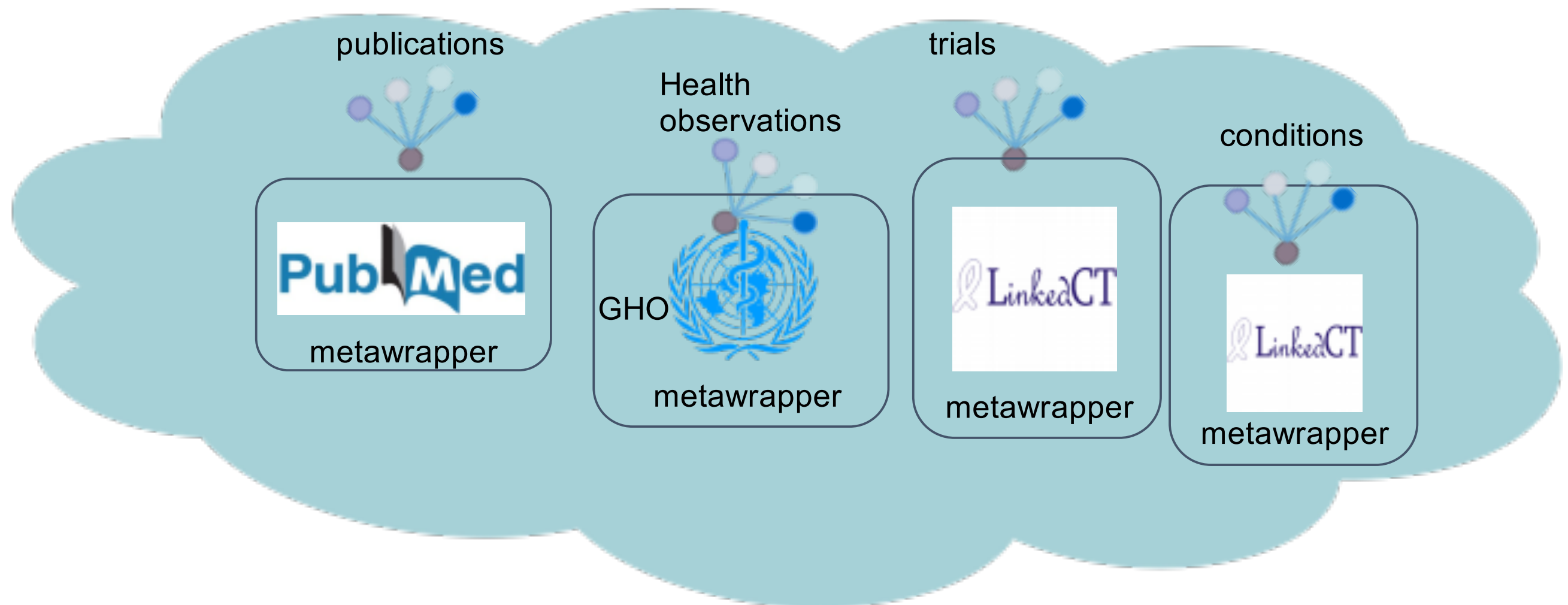
```

```

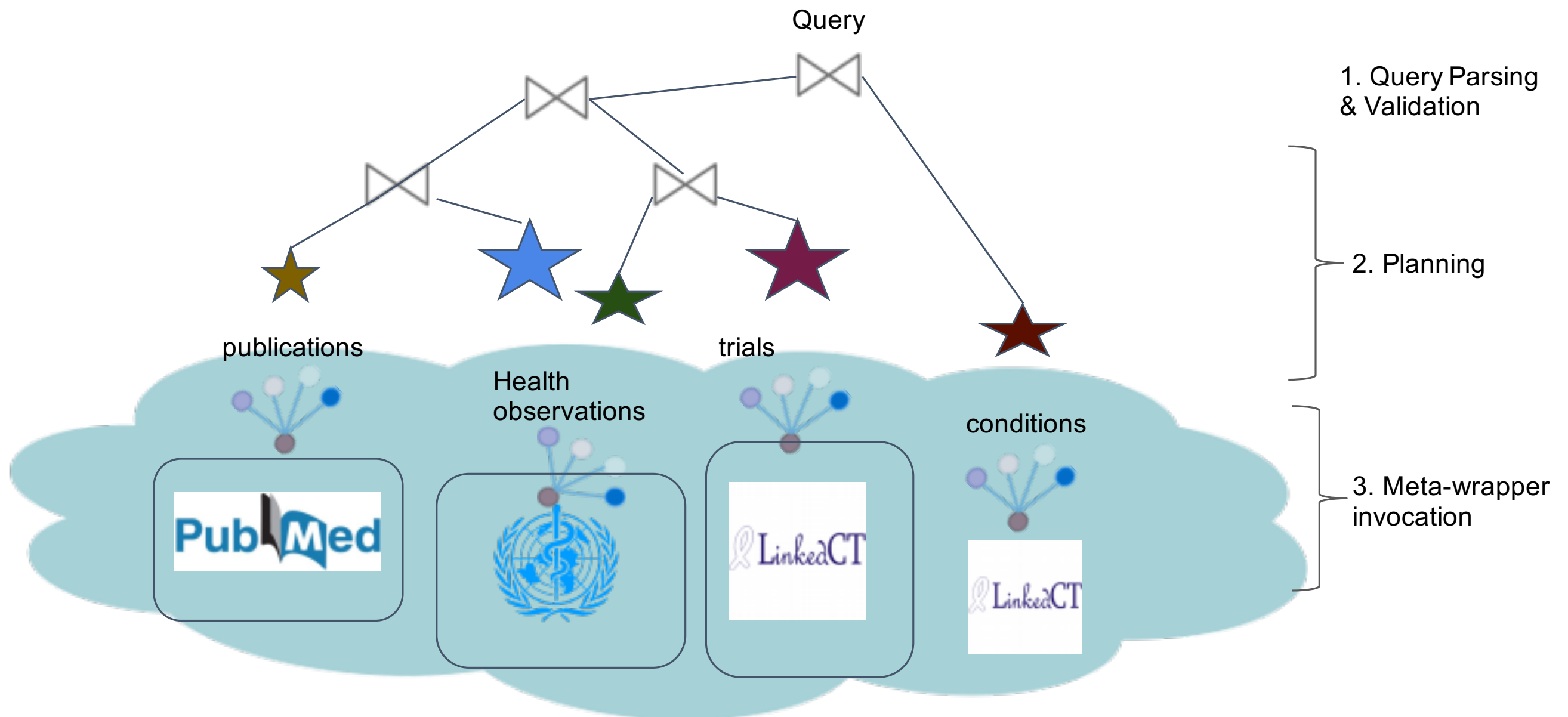
}
```

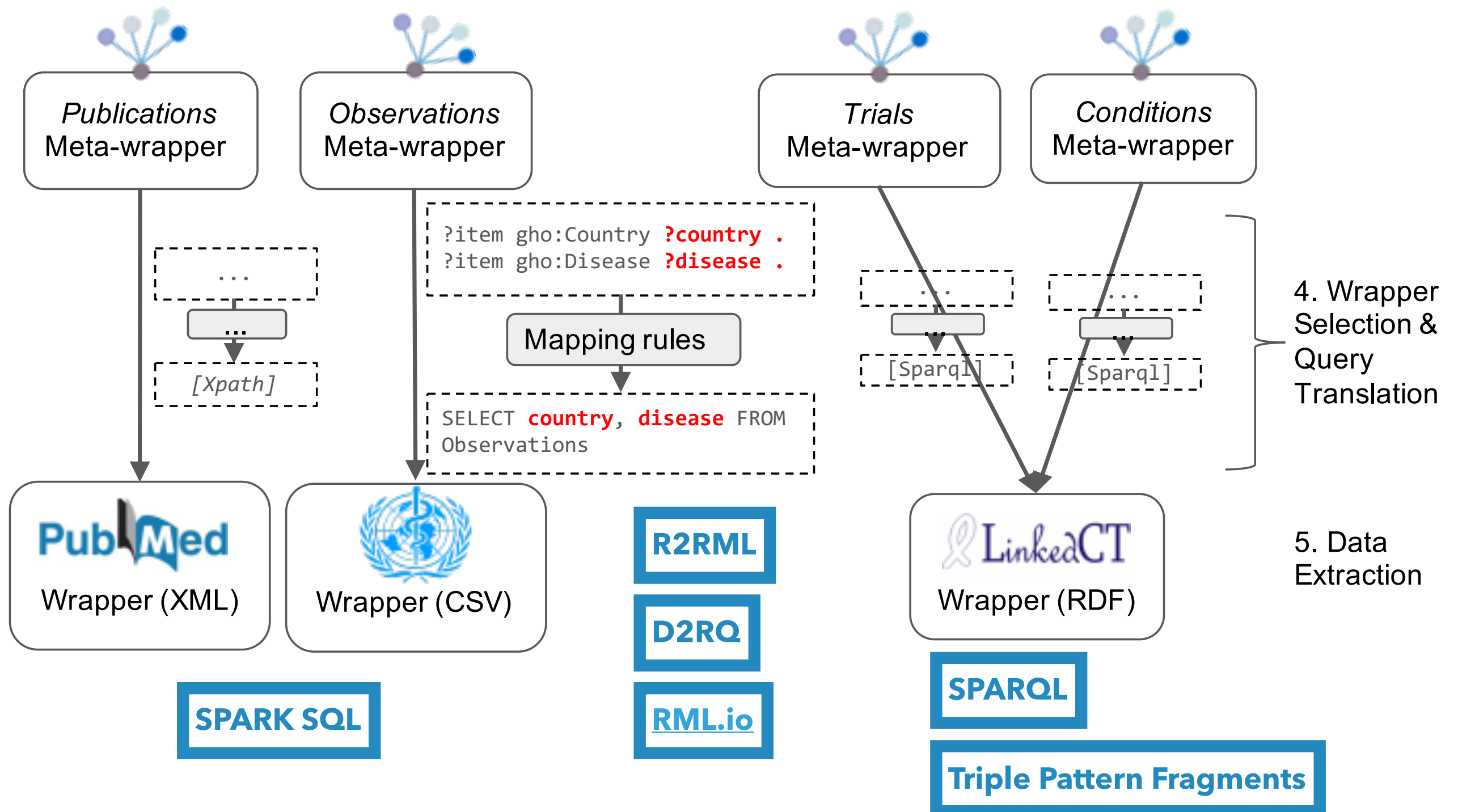


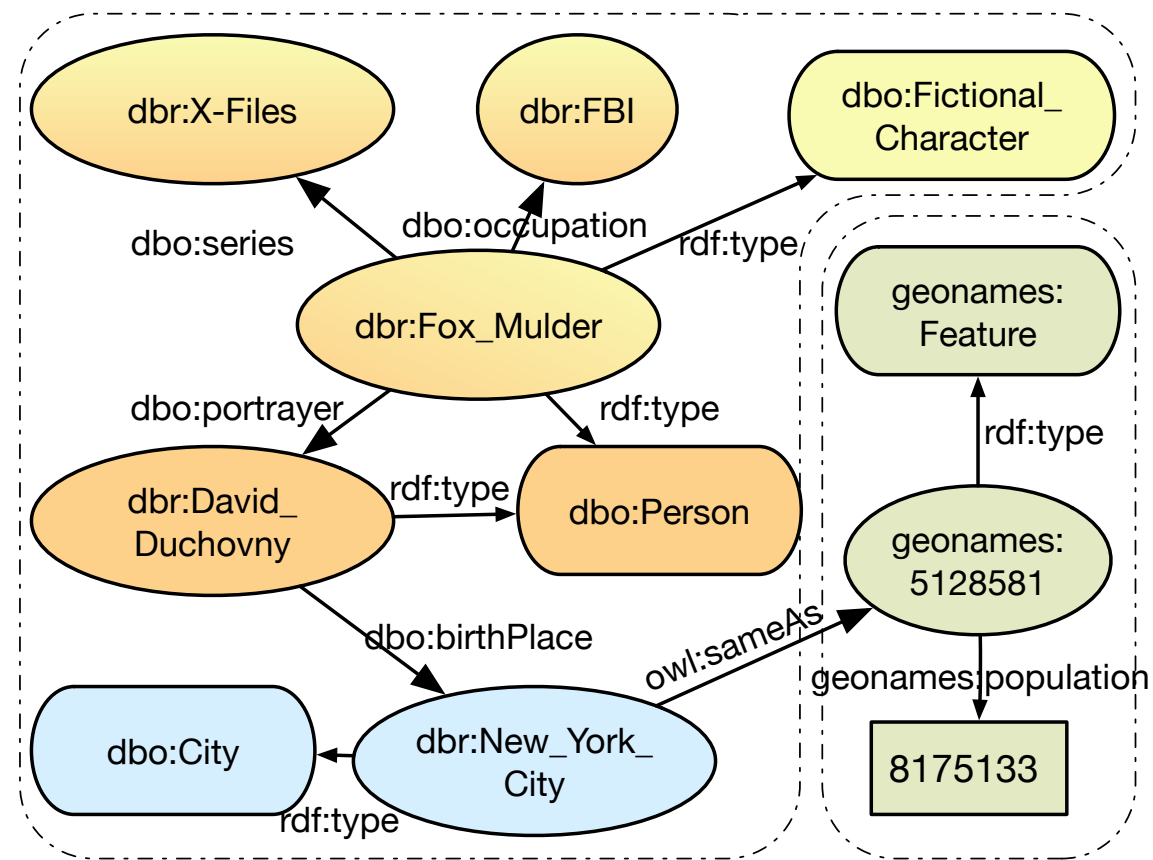
DATA IN THE DATA LAKE

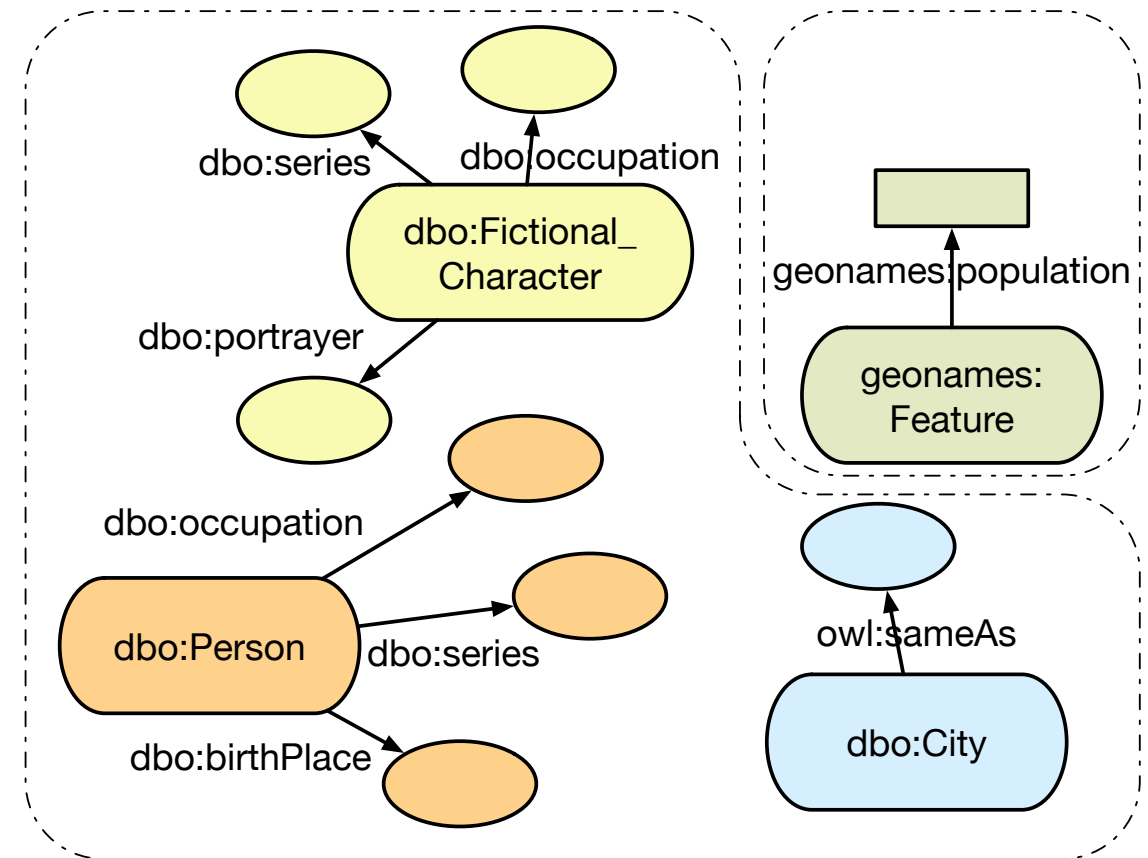
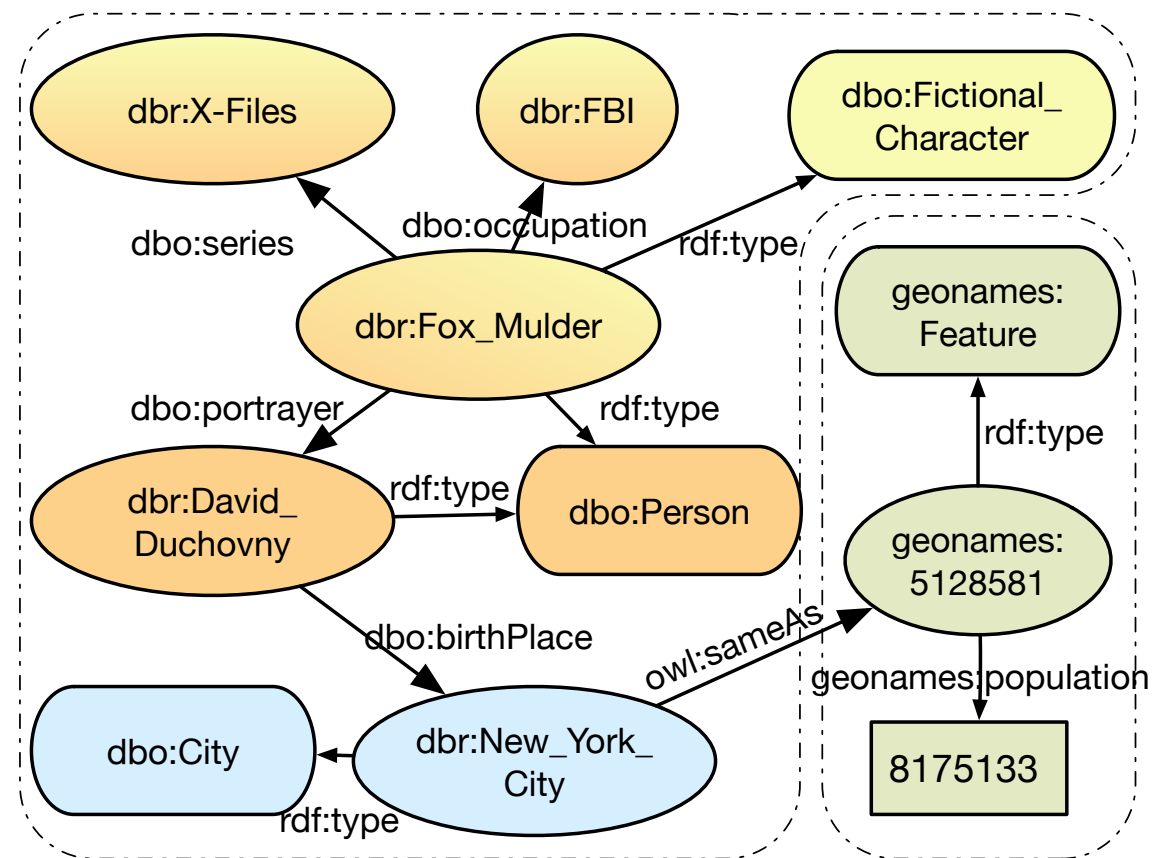


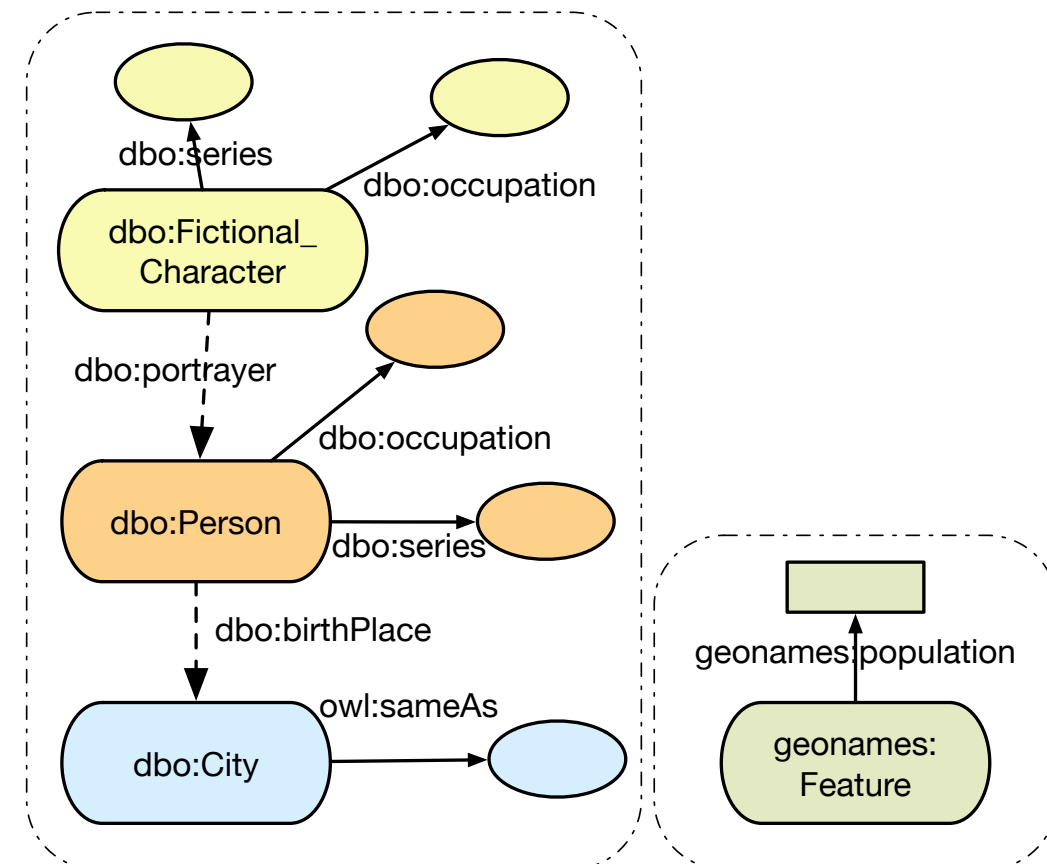
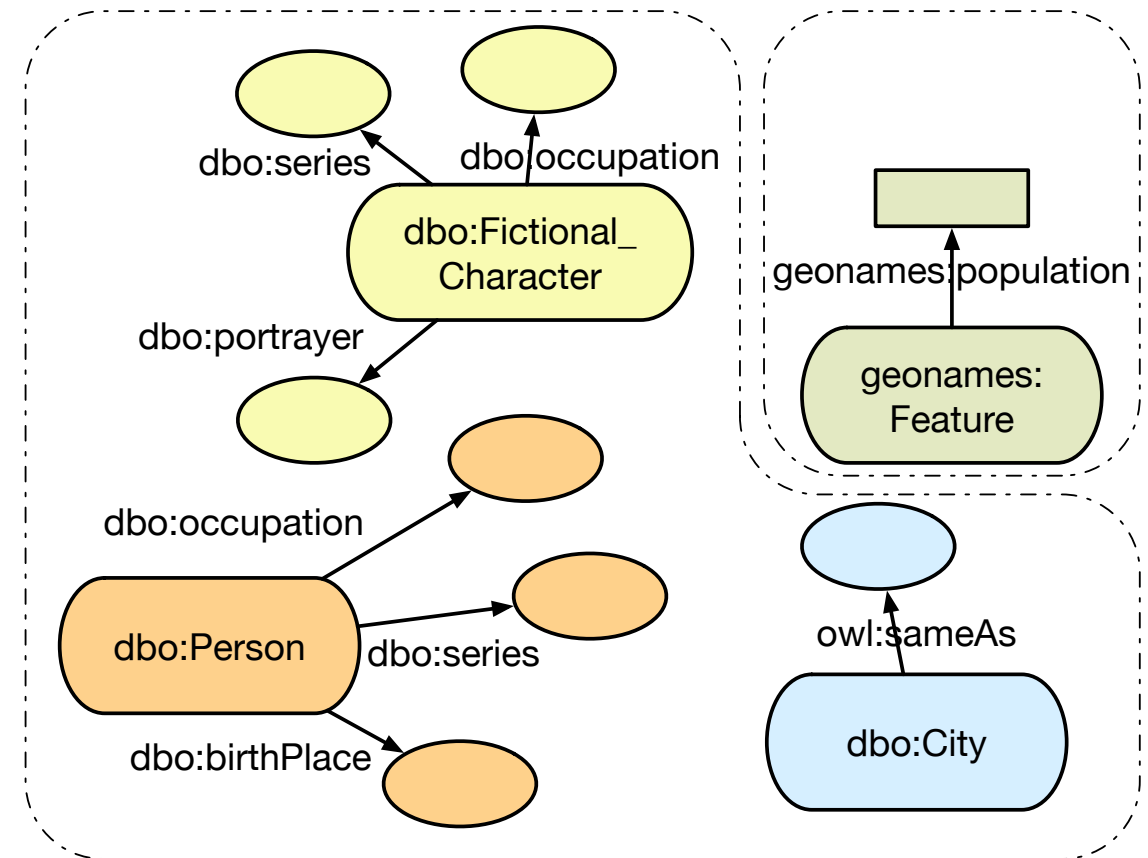
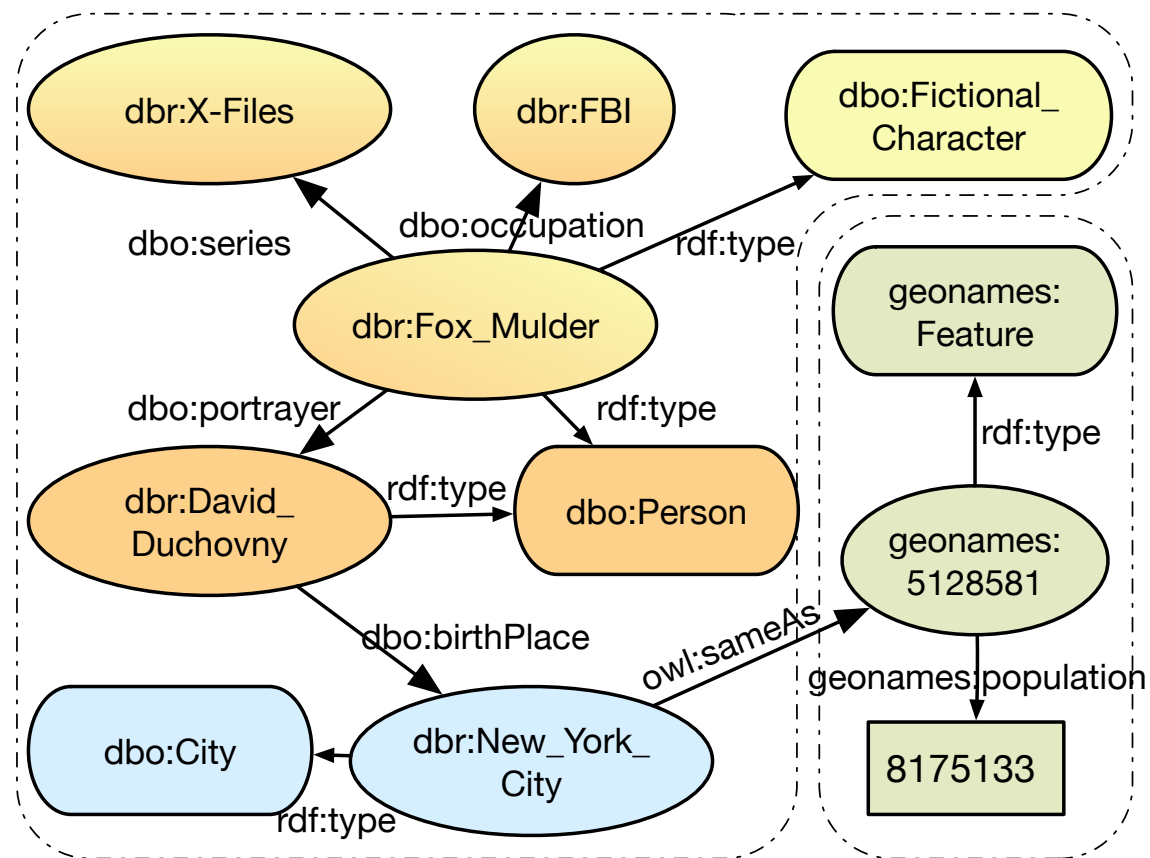
QUERY PLANNING

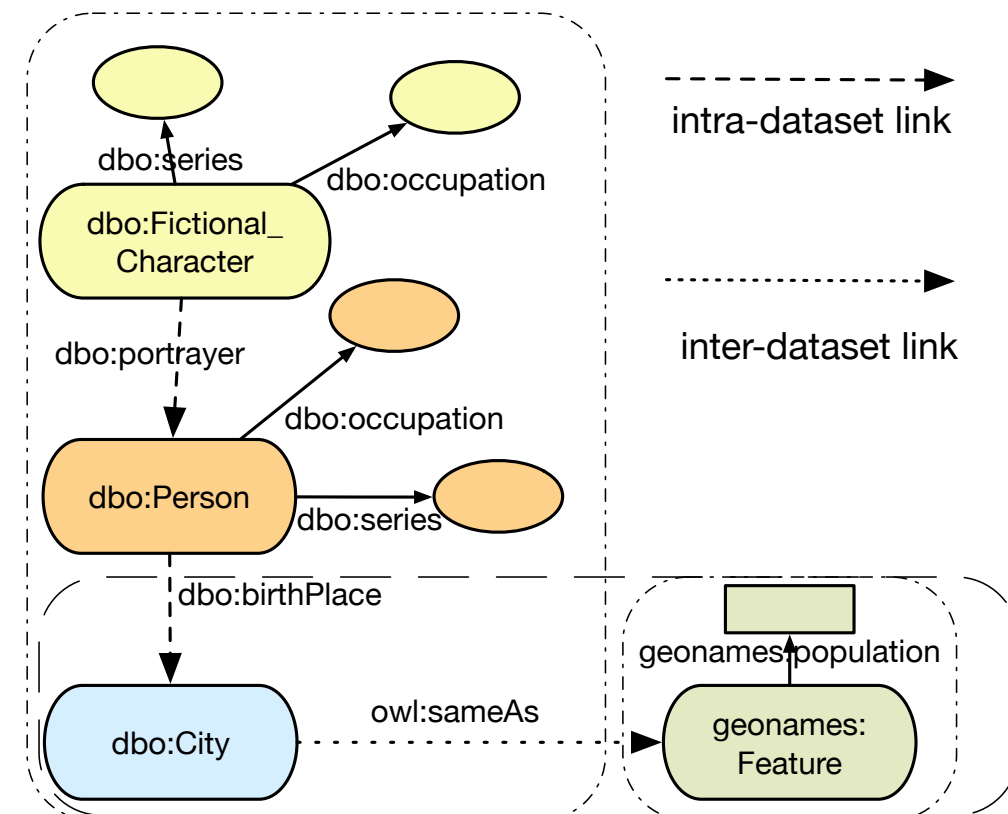
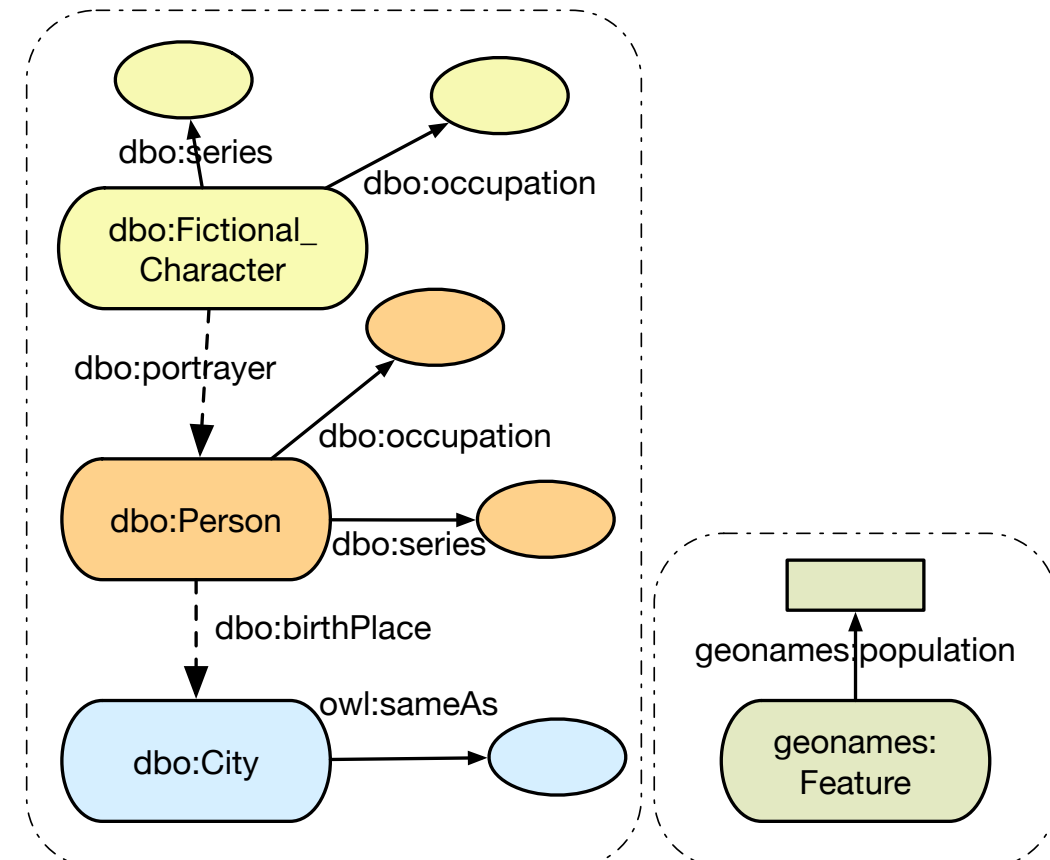
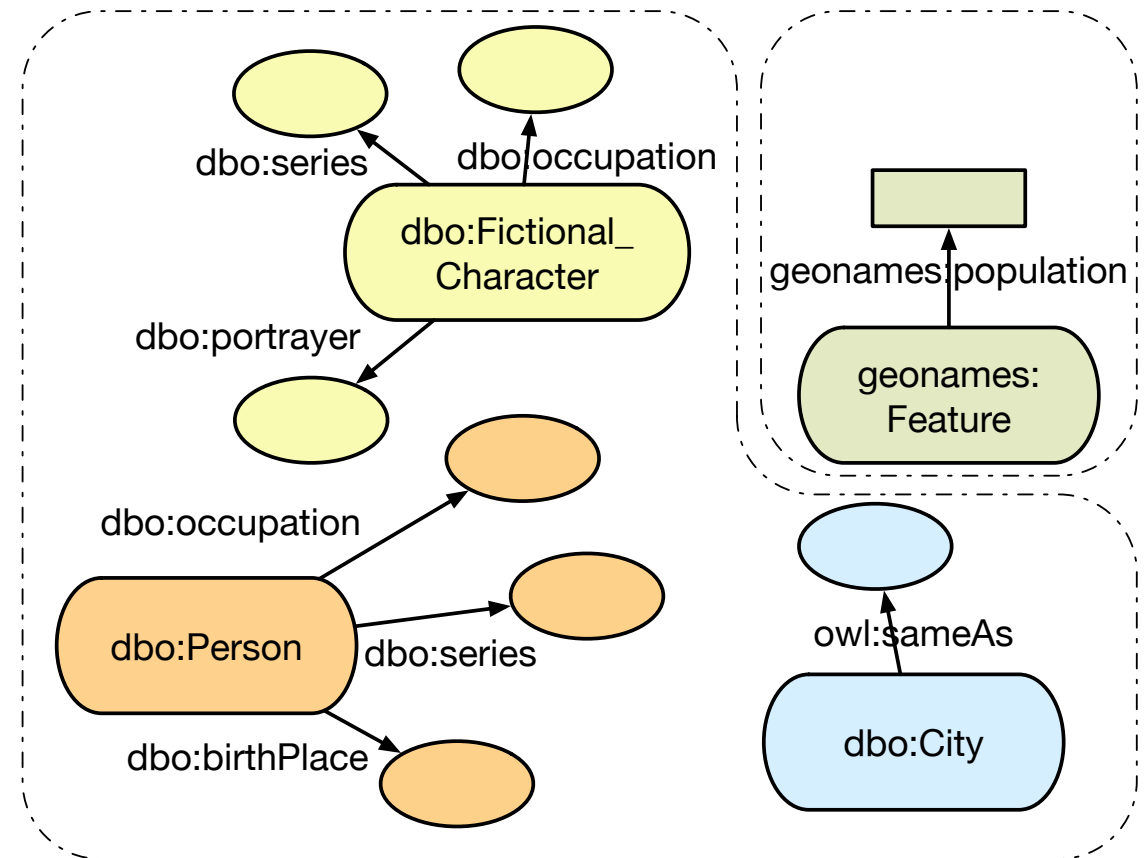
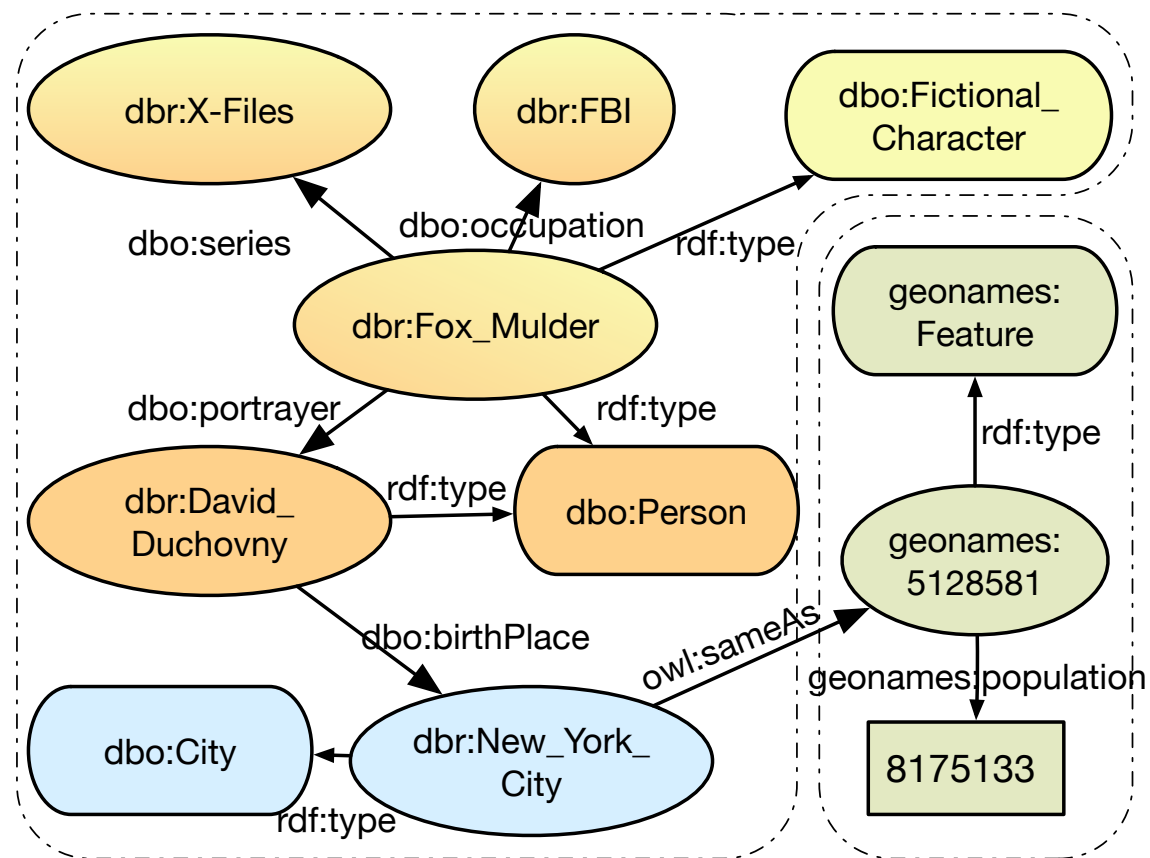












intra-dataset link

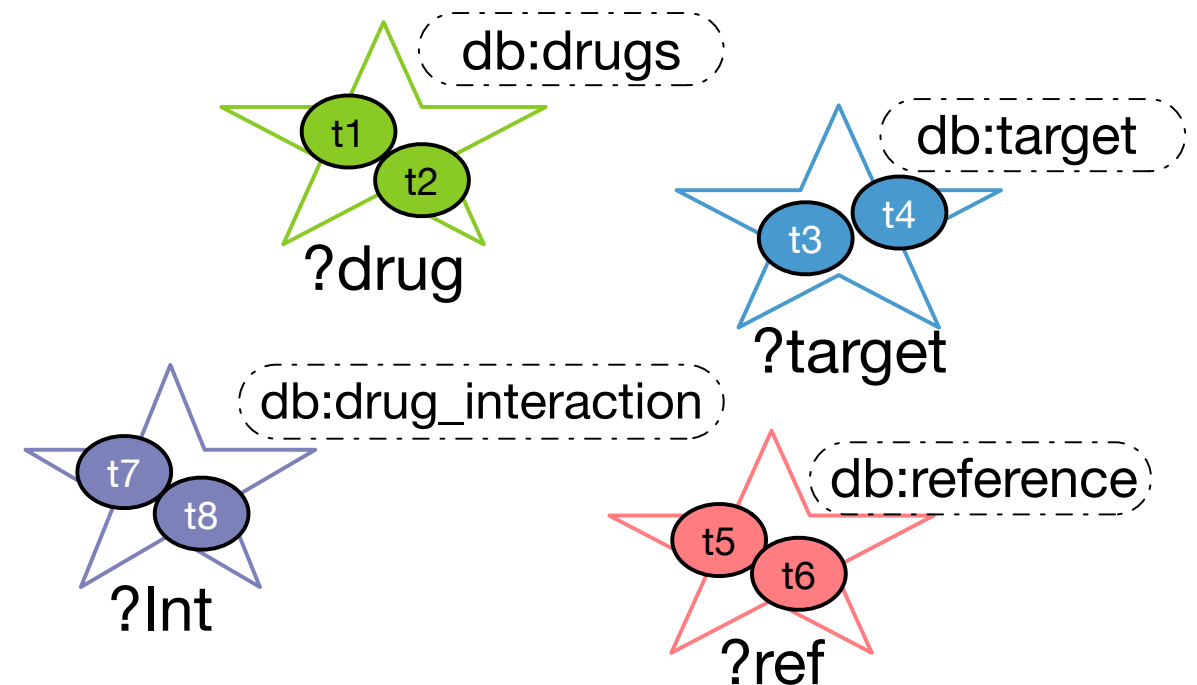
inter-dataset link


```
SELECT ?drug ?target ?ref ?Int WHERE {  
  t1 ?drug db:genericName ?name .  
  t2 ?drug db:target ?target .  
  t3 ?target db:drugReference ?ref .  
  t4 ?target db:proteinSequence ?ps .  
  t5 ?ref rdfs:label ?refLabel .  
  t6 ?ref foaf:page ?page .  
  t7 ?Int db:interactionDrug1 ?drug .  
  t8 ?Int db:interactionDrug2 ?intd. }  
}
```

```

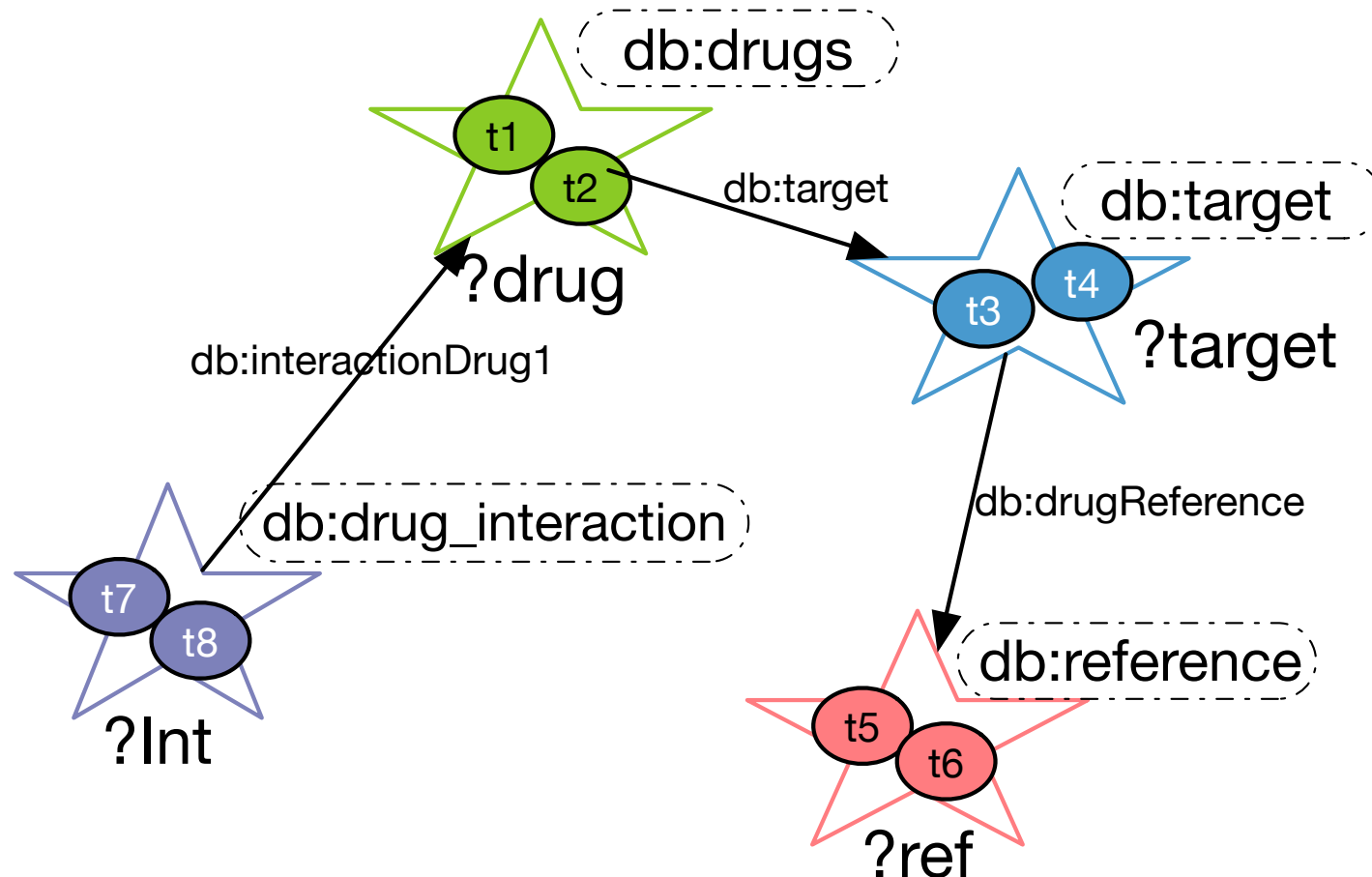
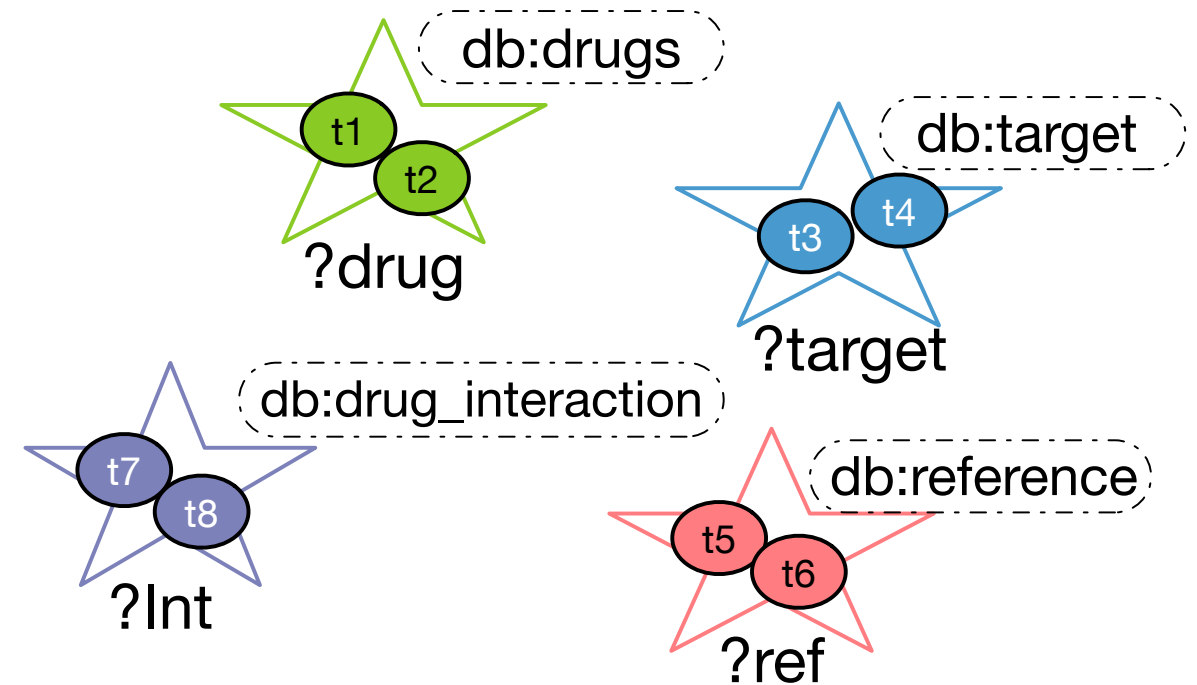
SELECT ?drug ?target ?ref ?Int WHERE {
  t1 ?drug db:genericName ?name .
  t2 ?drug db:target ?target .
  t3 ?target db:drugReference ?ref .
  t4 ?target db:proteinSequence ?ps .
  t5 ?ref rdfs:label ?refLabel .
  t6 ?ref foaf:page ?page .
  t7 ?Int db:interactionDrug1 ?drug .
  t8 ?Int db:interactionDrug2 ?intd. }

```



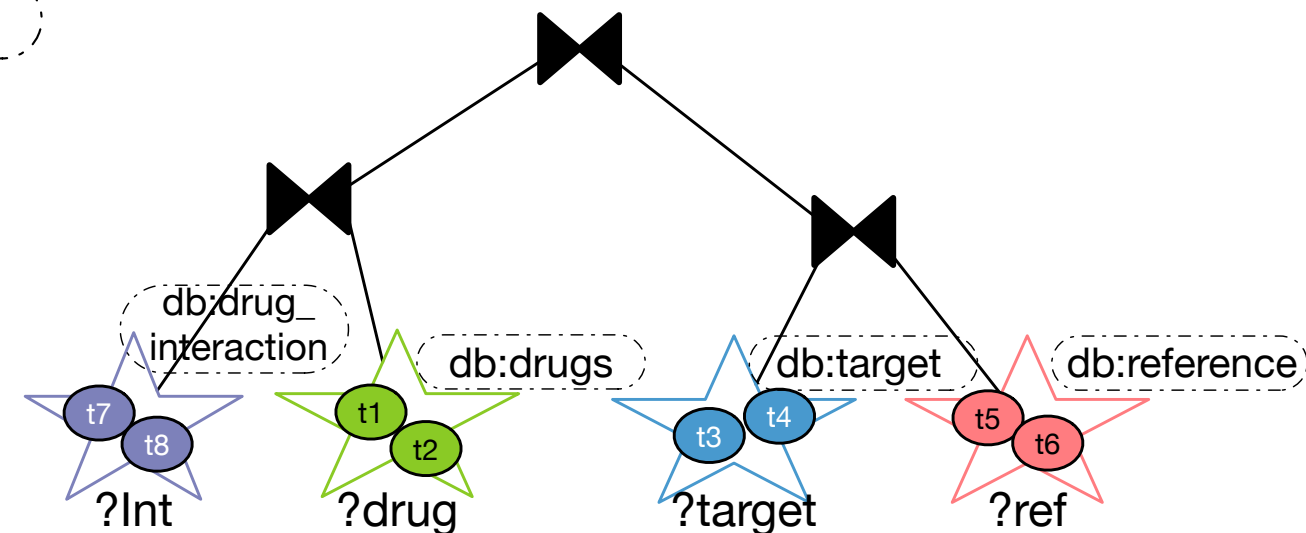
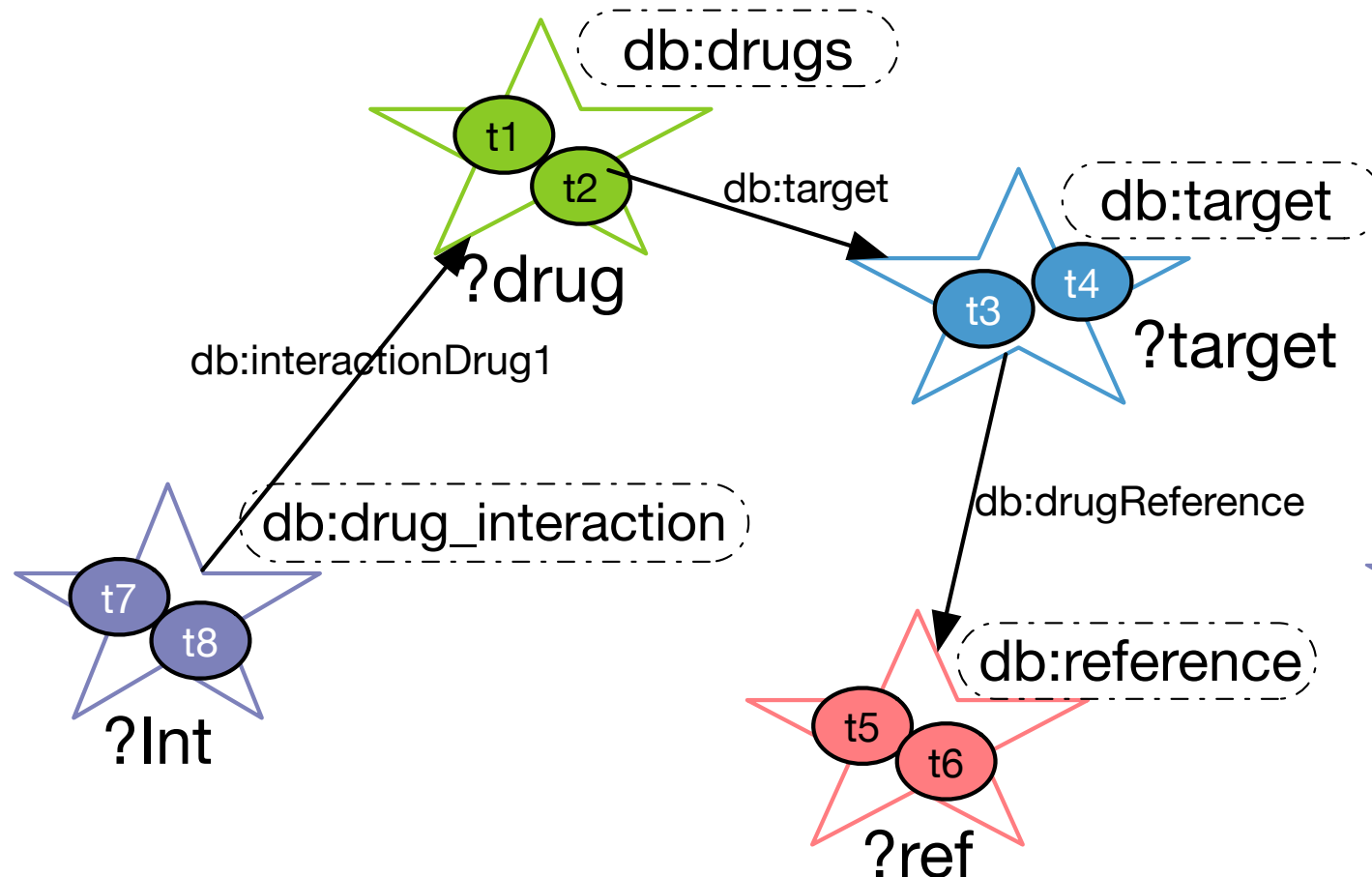
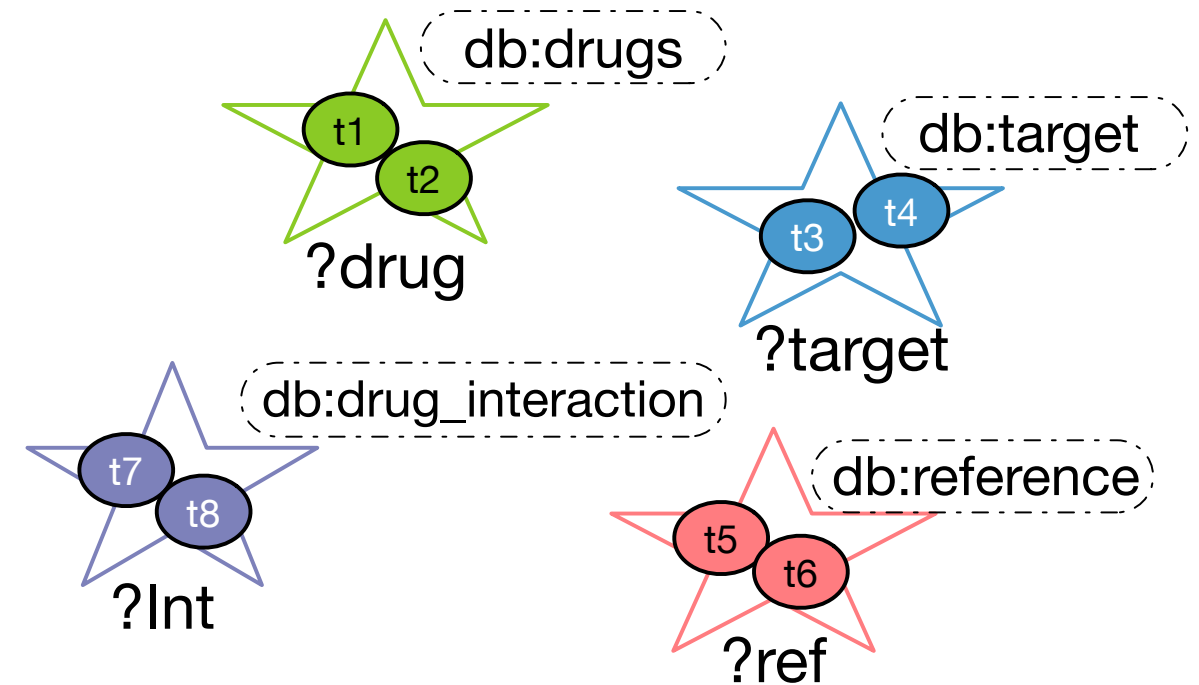
```

SELECT ?drug ?target ?ref ?Int WHERE {
  t1 ?drug db:genericName ?name .
  t2 ?drug db:target ?target .
  t3 ?target db:drugReference ?ref .
  t4 ?target db:proteinSequence ?ps .
  t5 ?ref rdfs:label ?refLabel .
  t6 ?ref foaf:page ?page .
  t7 ?Int db:interactionDrug1 ?drug .
  t8 ?Int db:interactionDrug2 ?intd. }
    
```



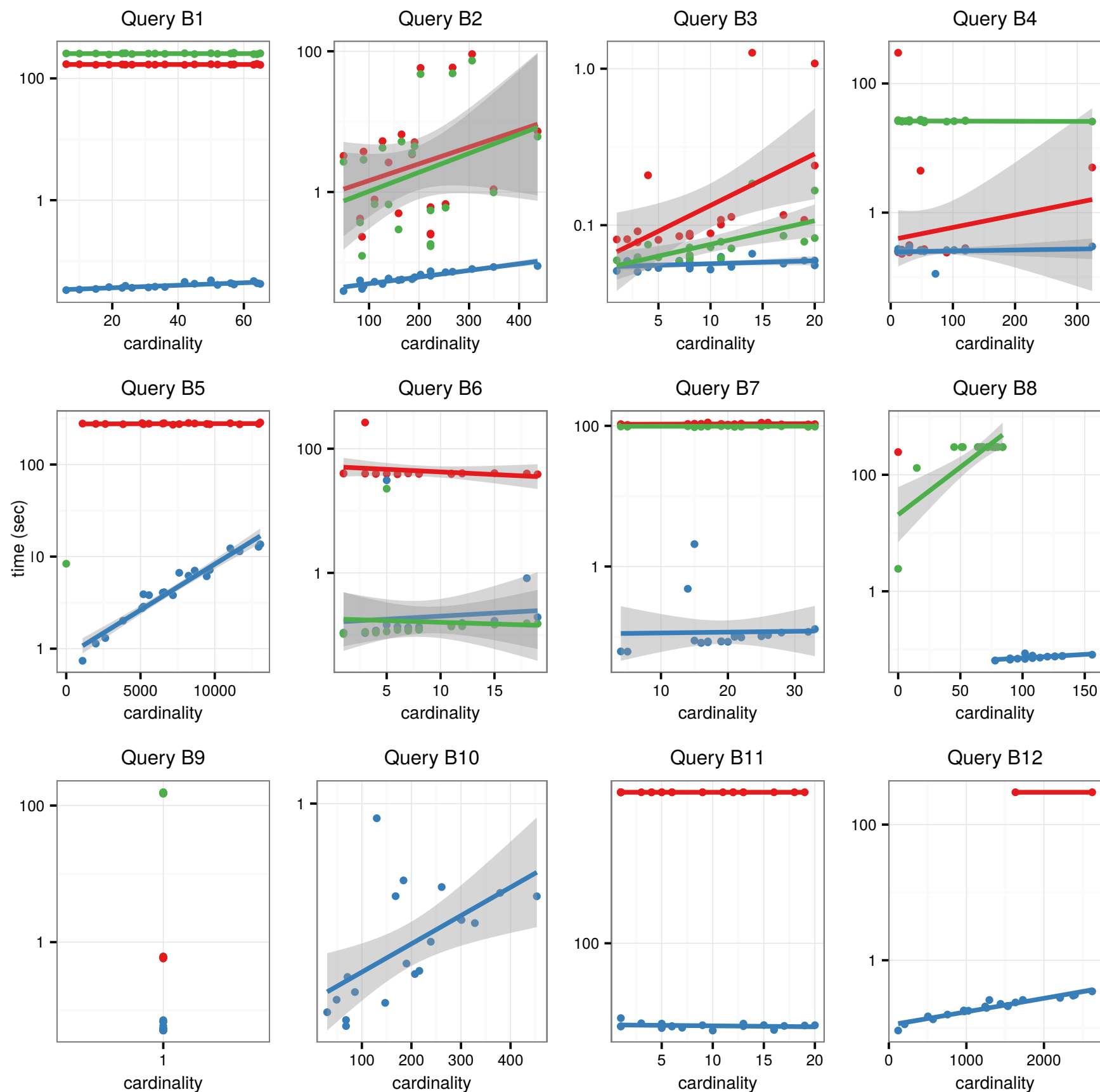
```

SELECT ?drug ?target ?ref ?Int WHERE {
  t1 ?drug db:genericName ?name .
  t2 ?drug db:target ?target .
  t3 ?target db:drugReference ?ref .
  t4 ?target db:proteinSequence ?ps .
  t5 ?ref rdfs:label ?refLabel .
  t6 ?ref foaf:page ?page .
  t7 ?Int db:interactionDrug1 ?drug .
  t8 ?Int db:interactionDrug2 ?intd. }
    
```



HOW TO BUILD MOLECULE TEMPLATES

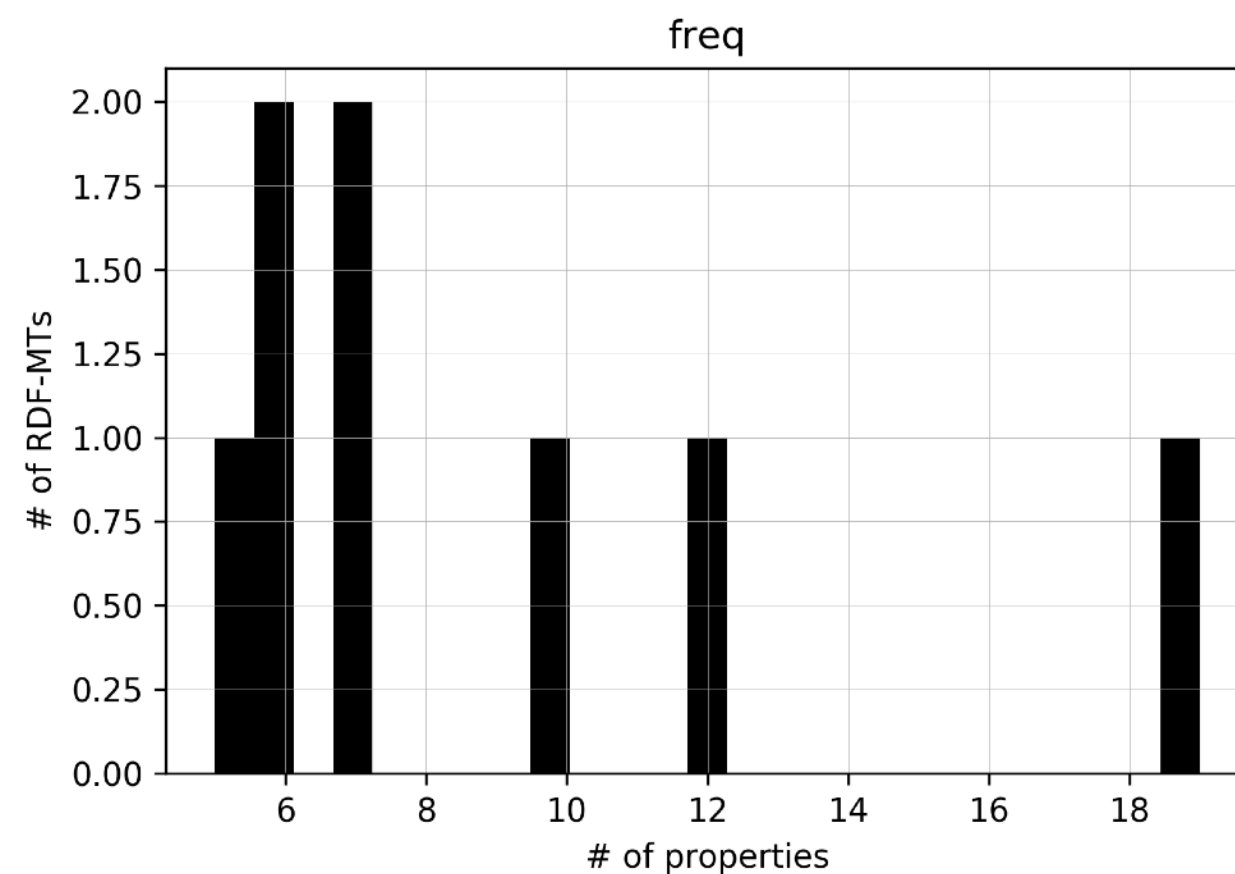
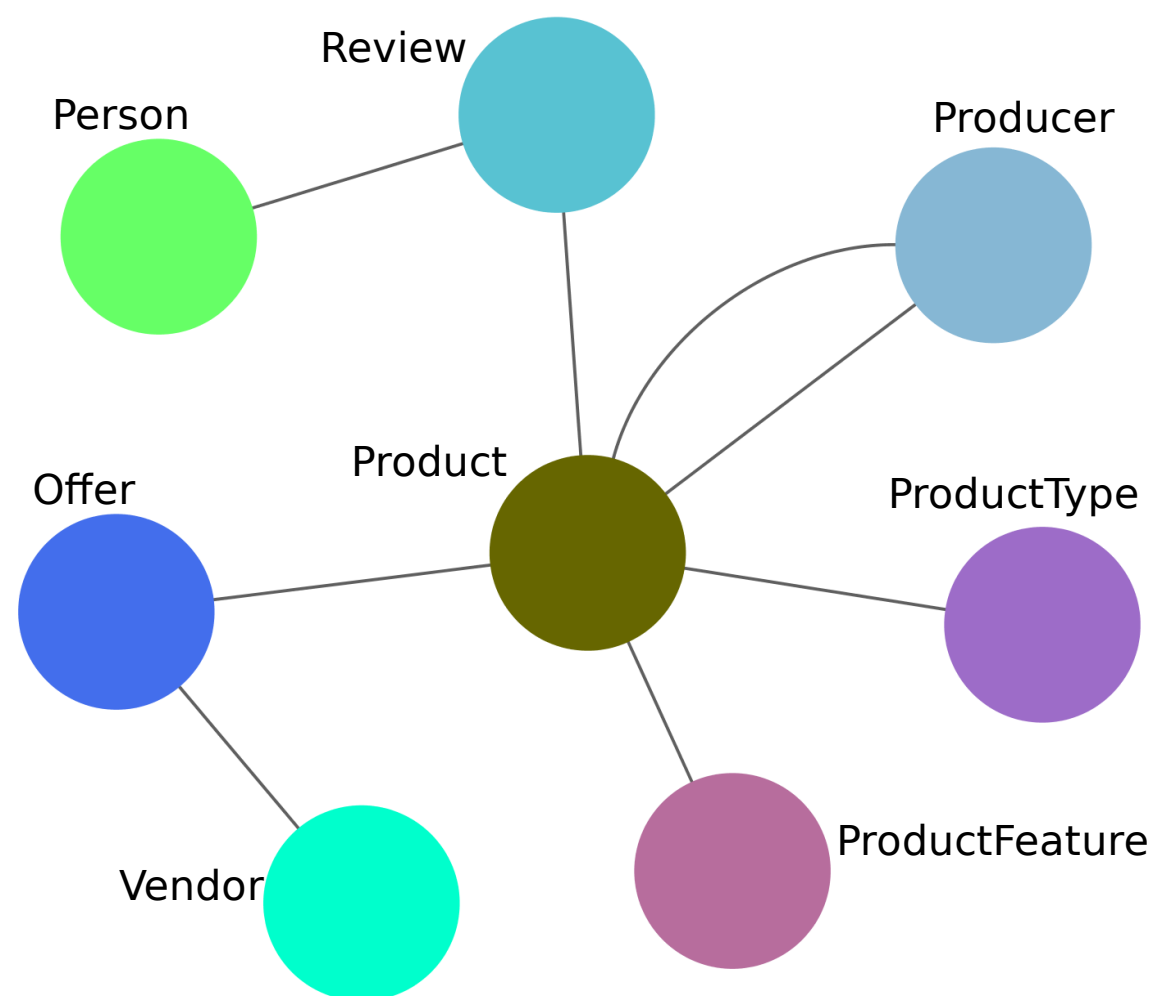
- ▶ MULDER-MT technique
- ▶ Graph Partitioning & Community Detection
 - ▶ METIS - node partitioning
 - ▶ semEP - edge partitioning
- ▶ Benchmarks:
 - ▶ Berlin SPARQL Benchmark (BSBM)
 - ▶ FedBench (9 interlinked datasets)
 - ▶ Life Sciences Linked Open Data (LSLOD) (10 interlinked datasets)
- ▶ Compared Federated Query Engines: ANAPSID, FedX

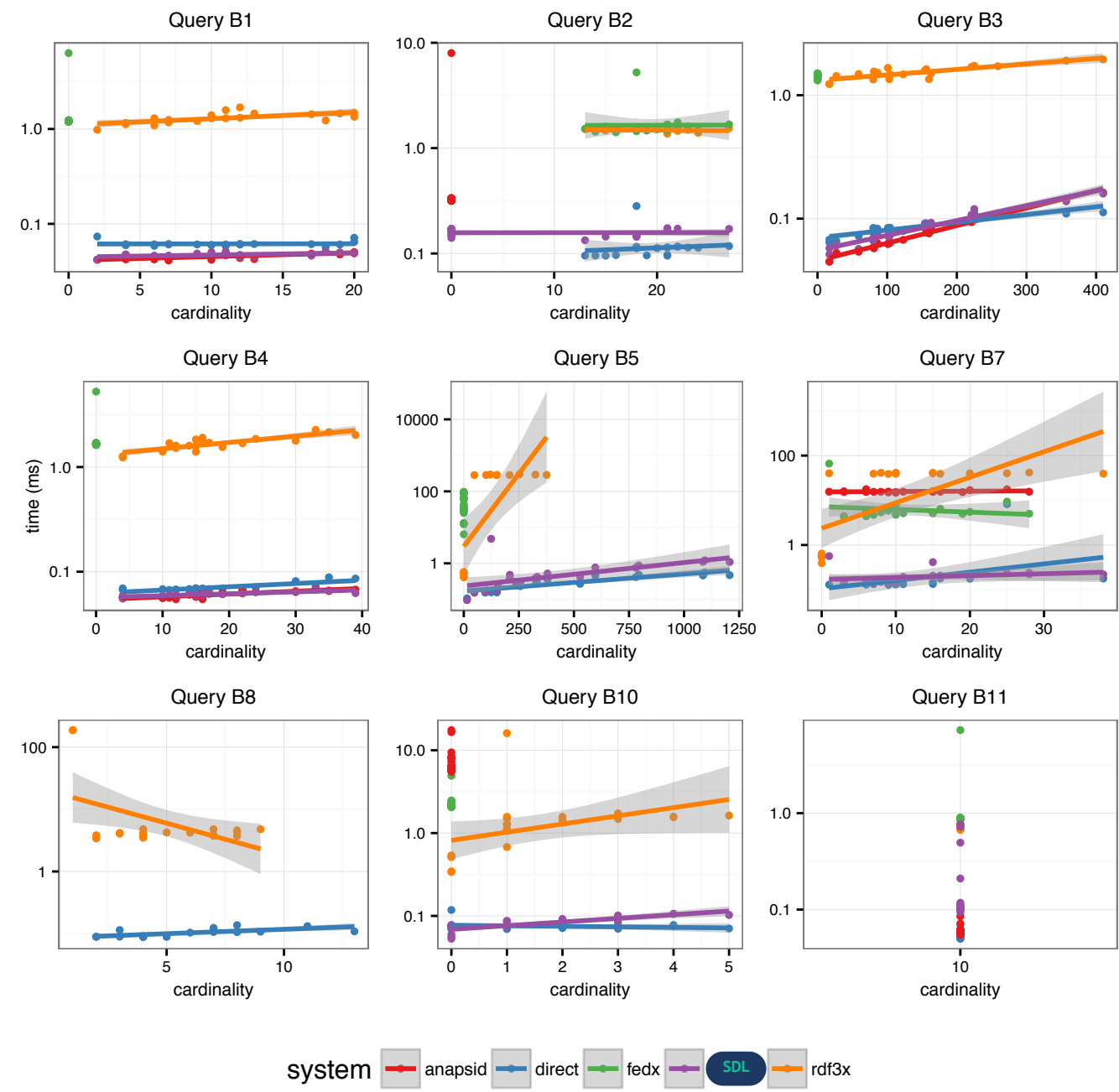
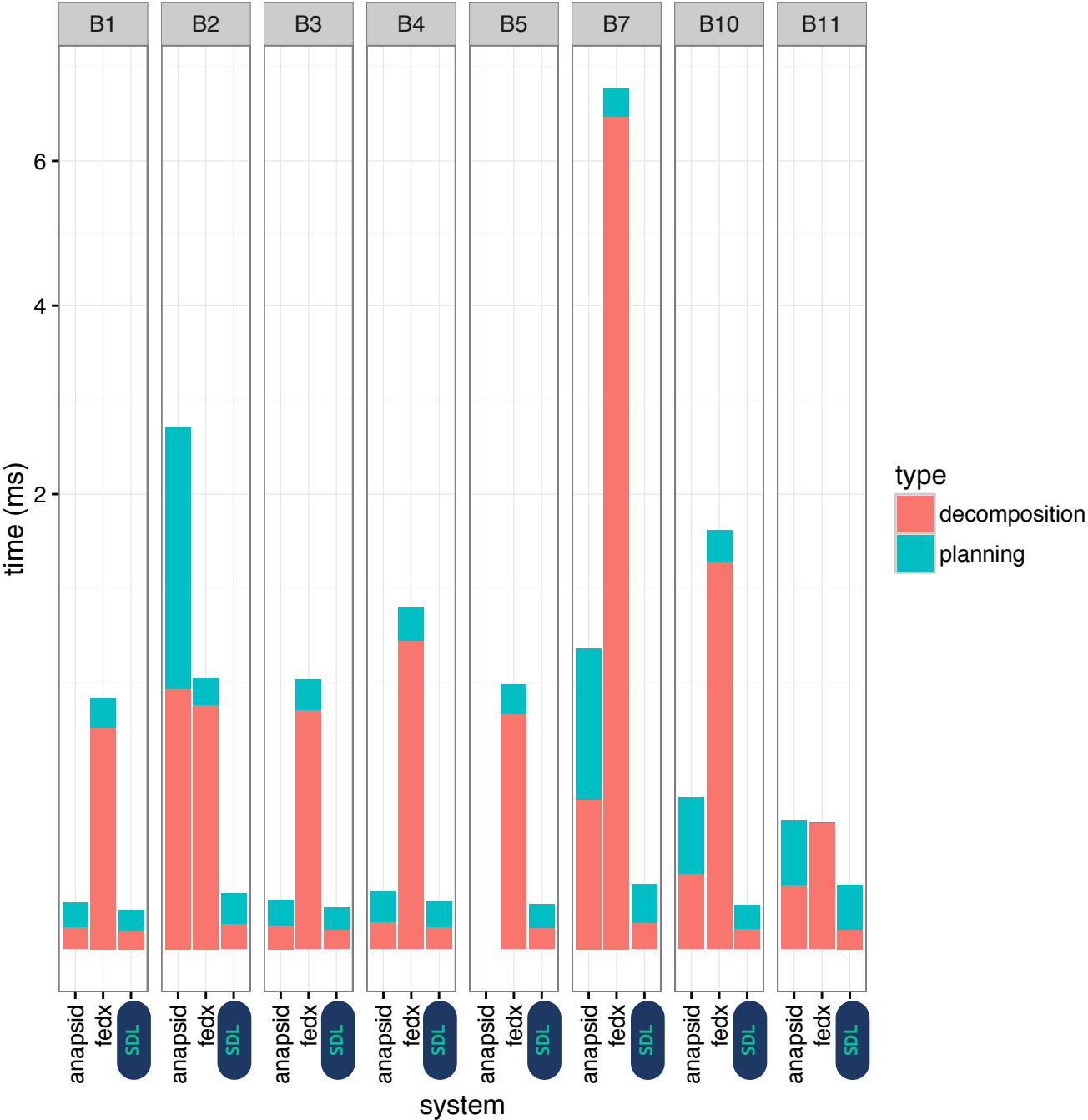


MULDER Molecules
technique (RDF-MT)
outperforms METIS
and semEP

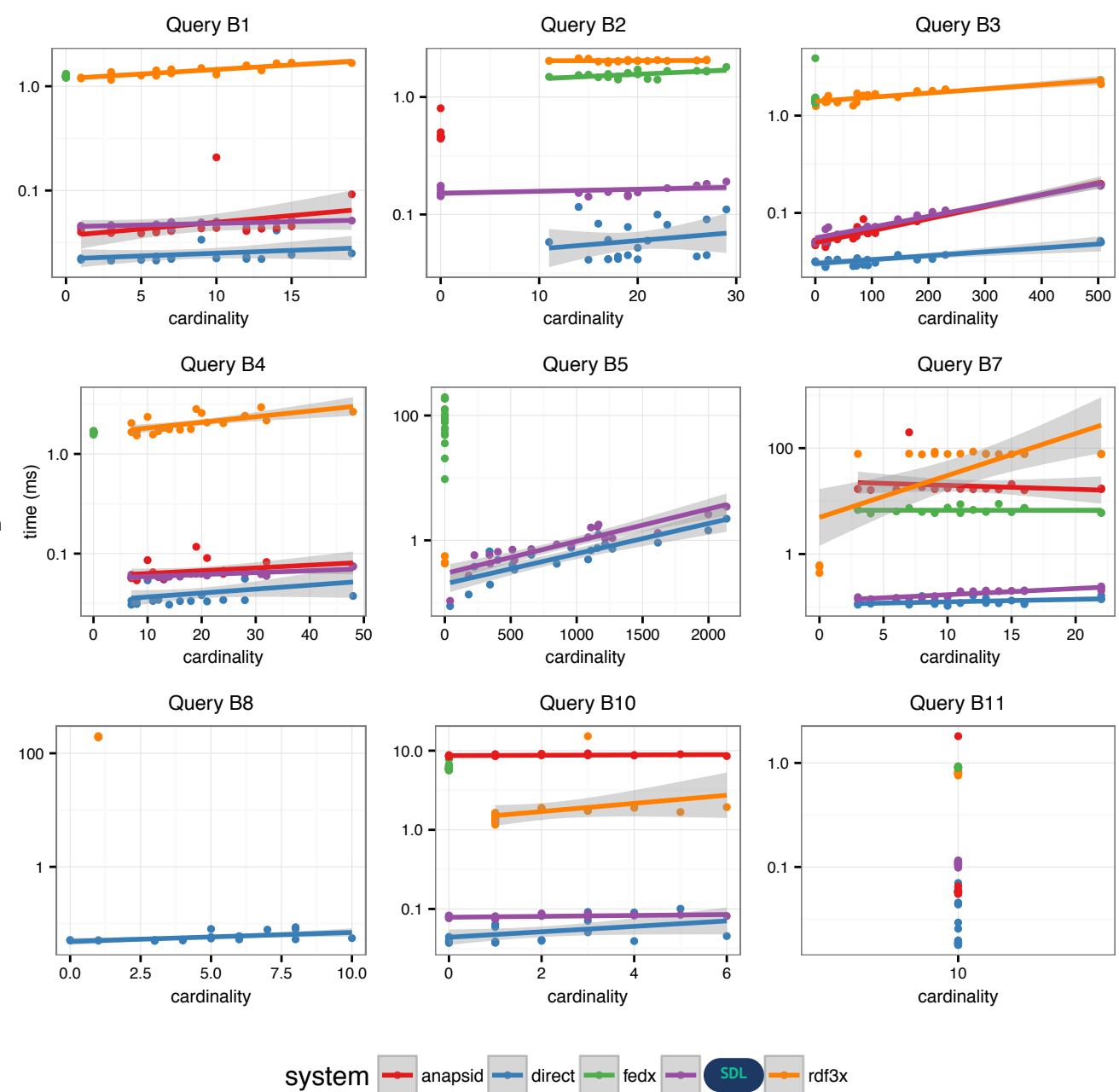
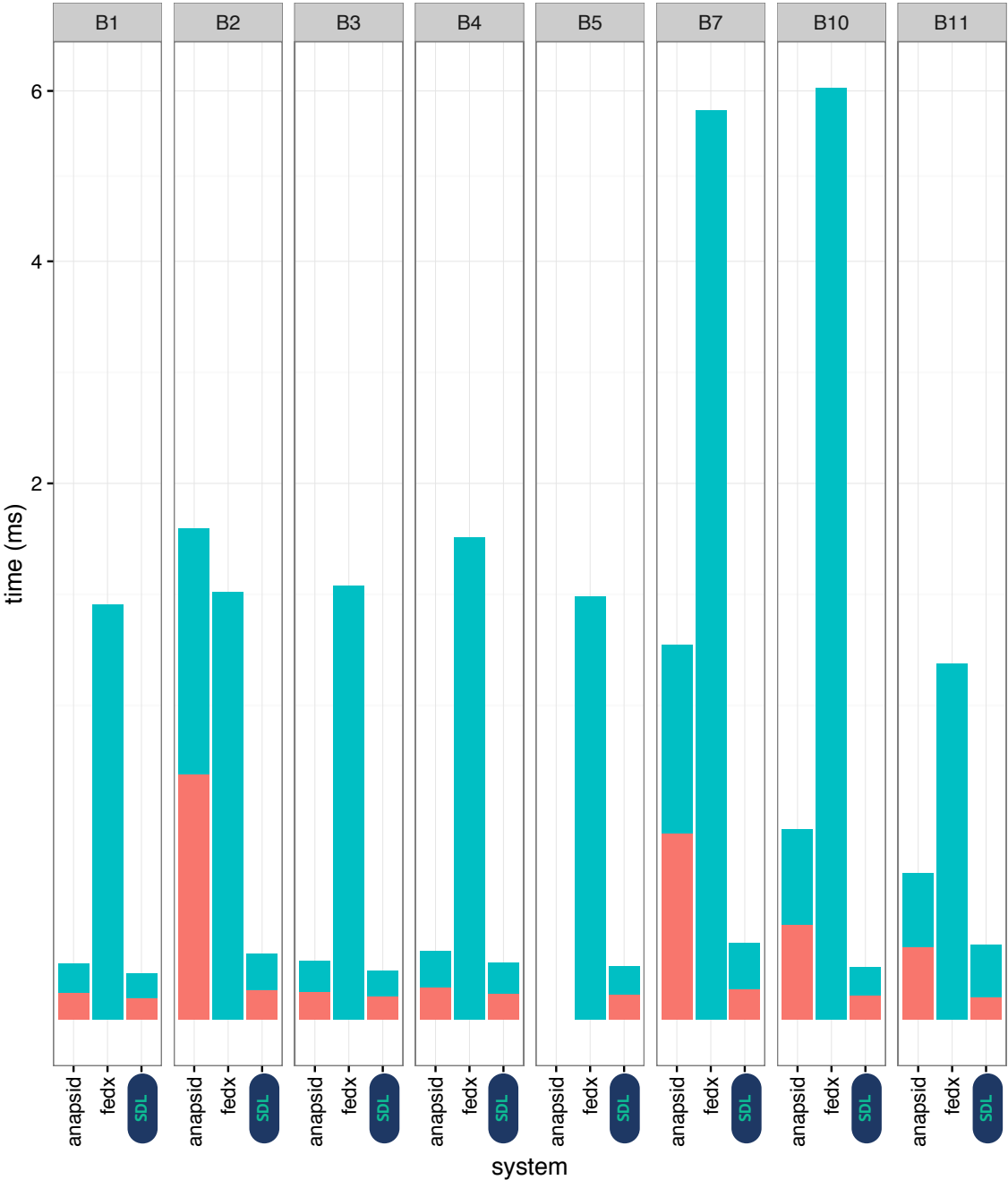
QUERY EXECUTION

BSBM RDF Molecules mined by MULDER

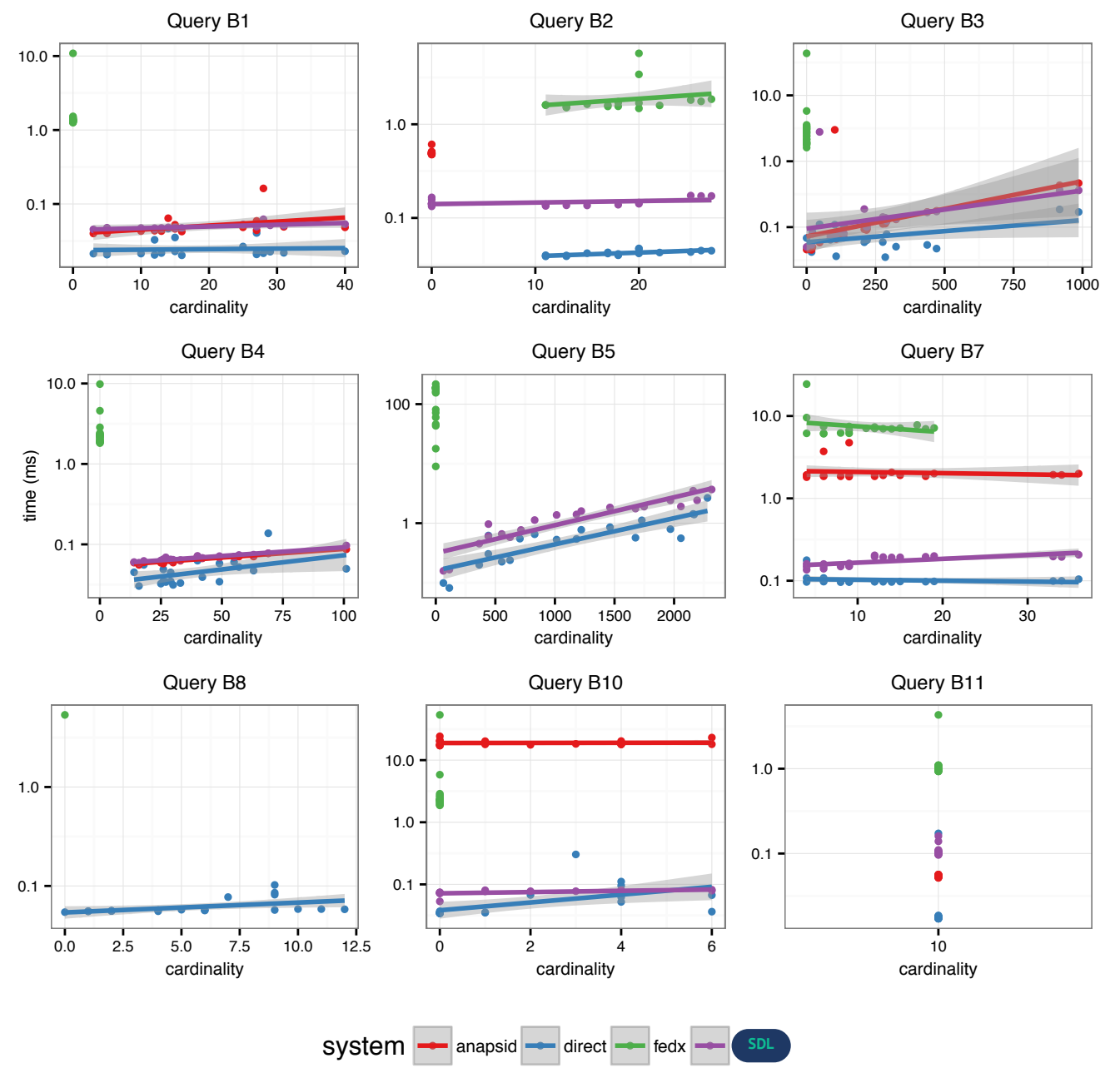
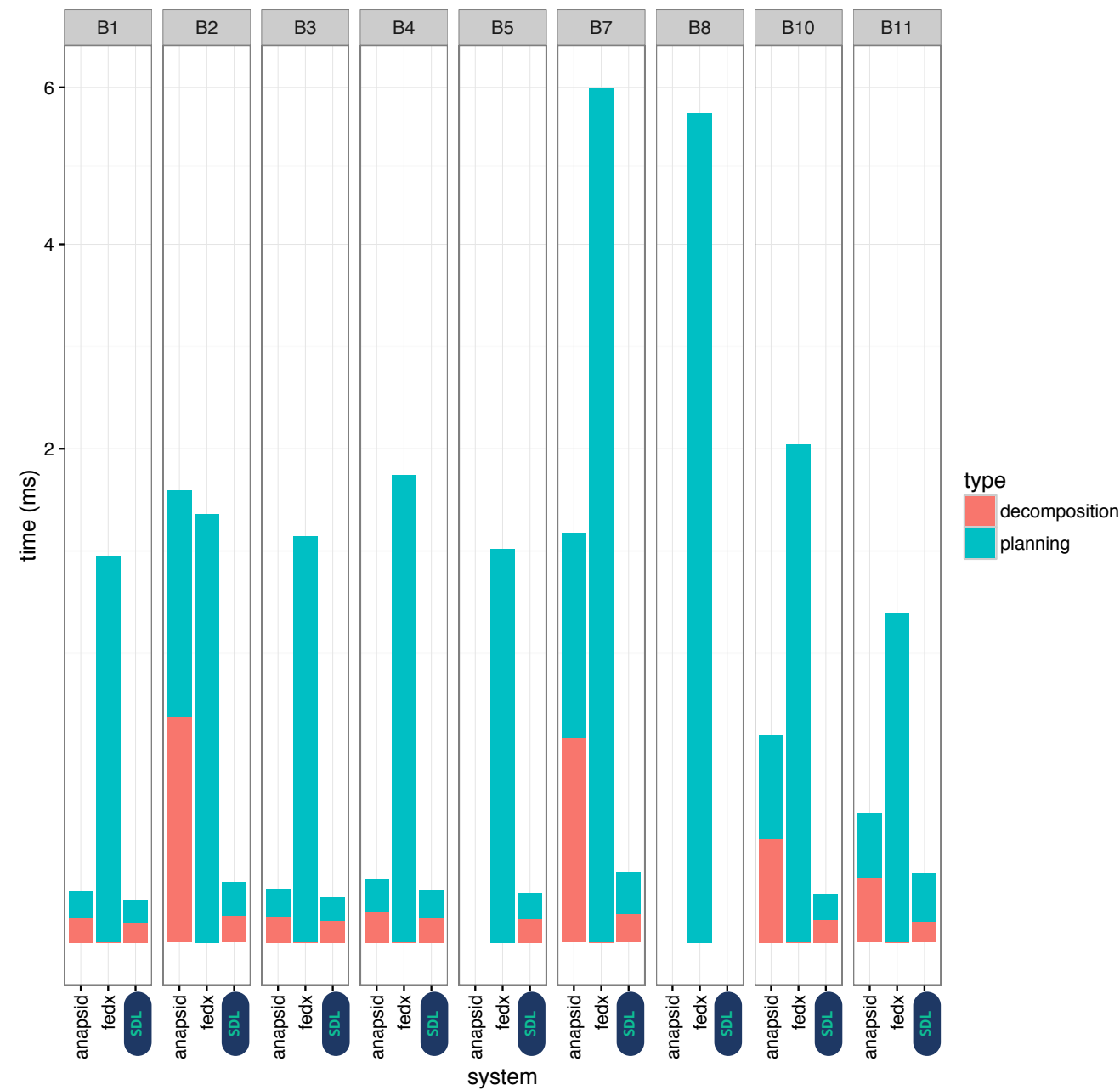




BSBM 100M TRIPLES

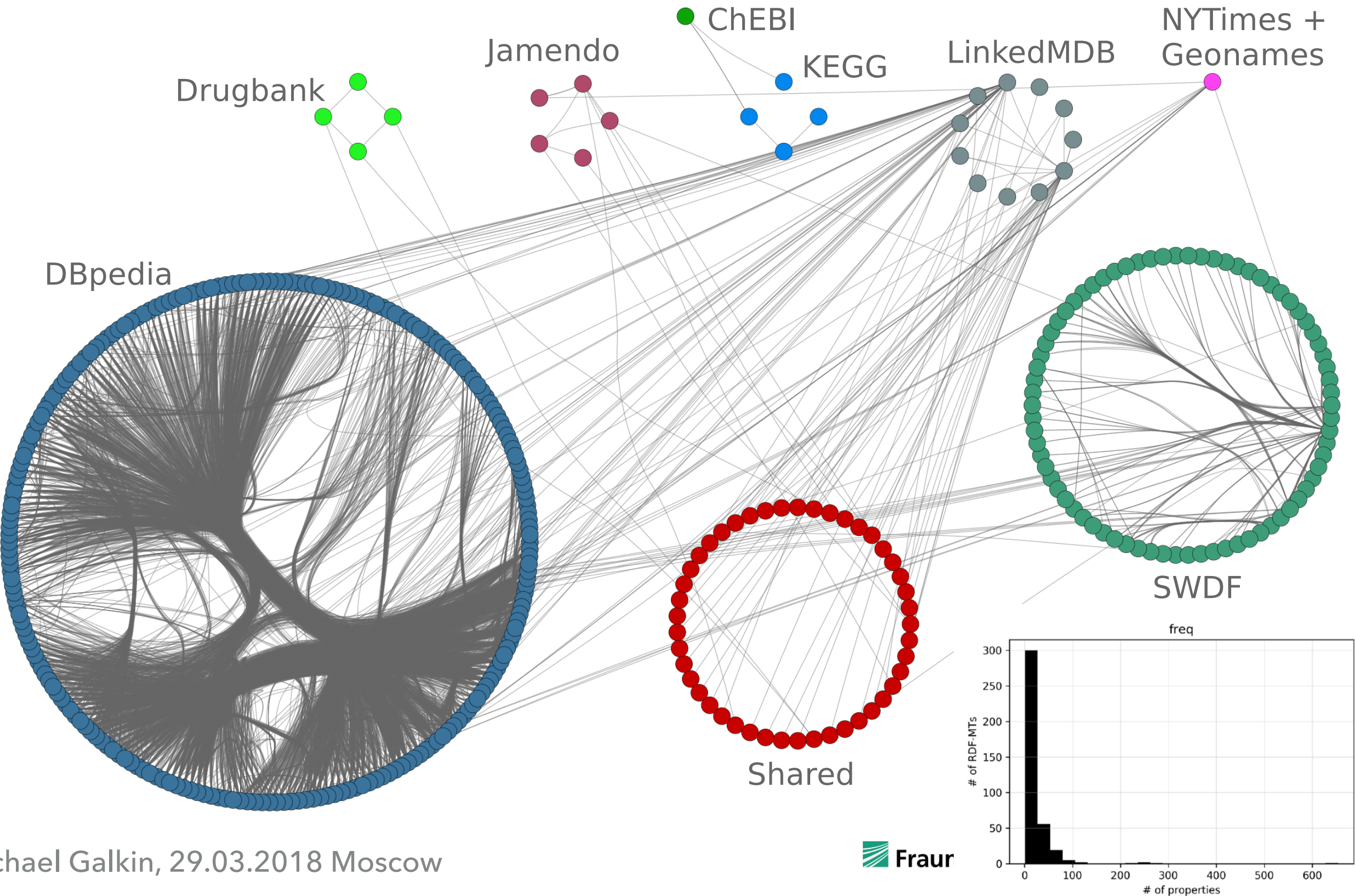


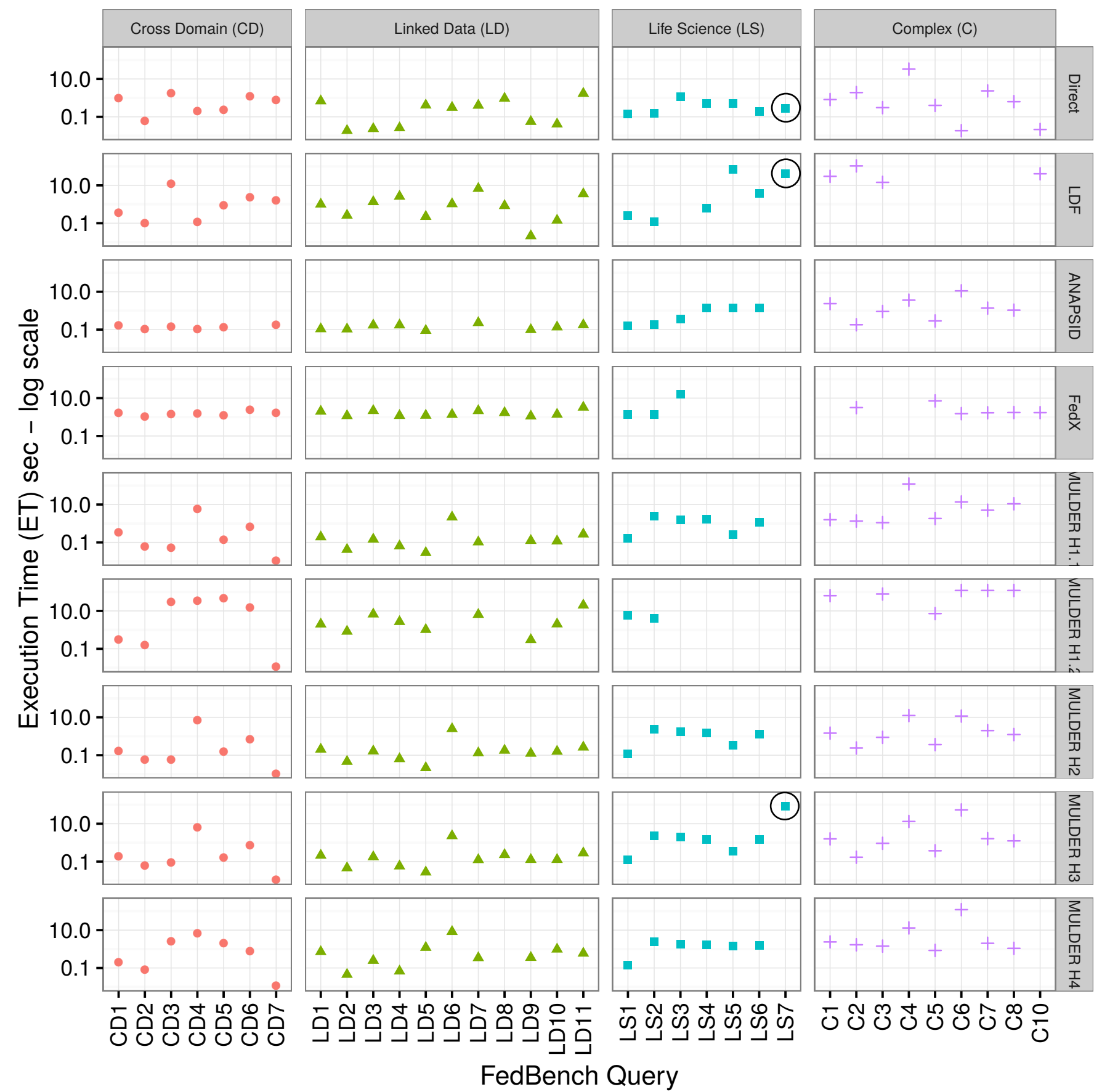
BSBM 200M TRIPLES



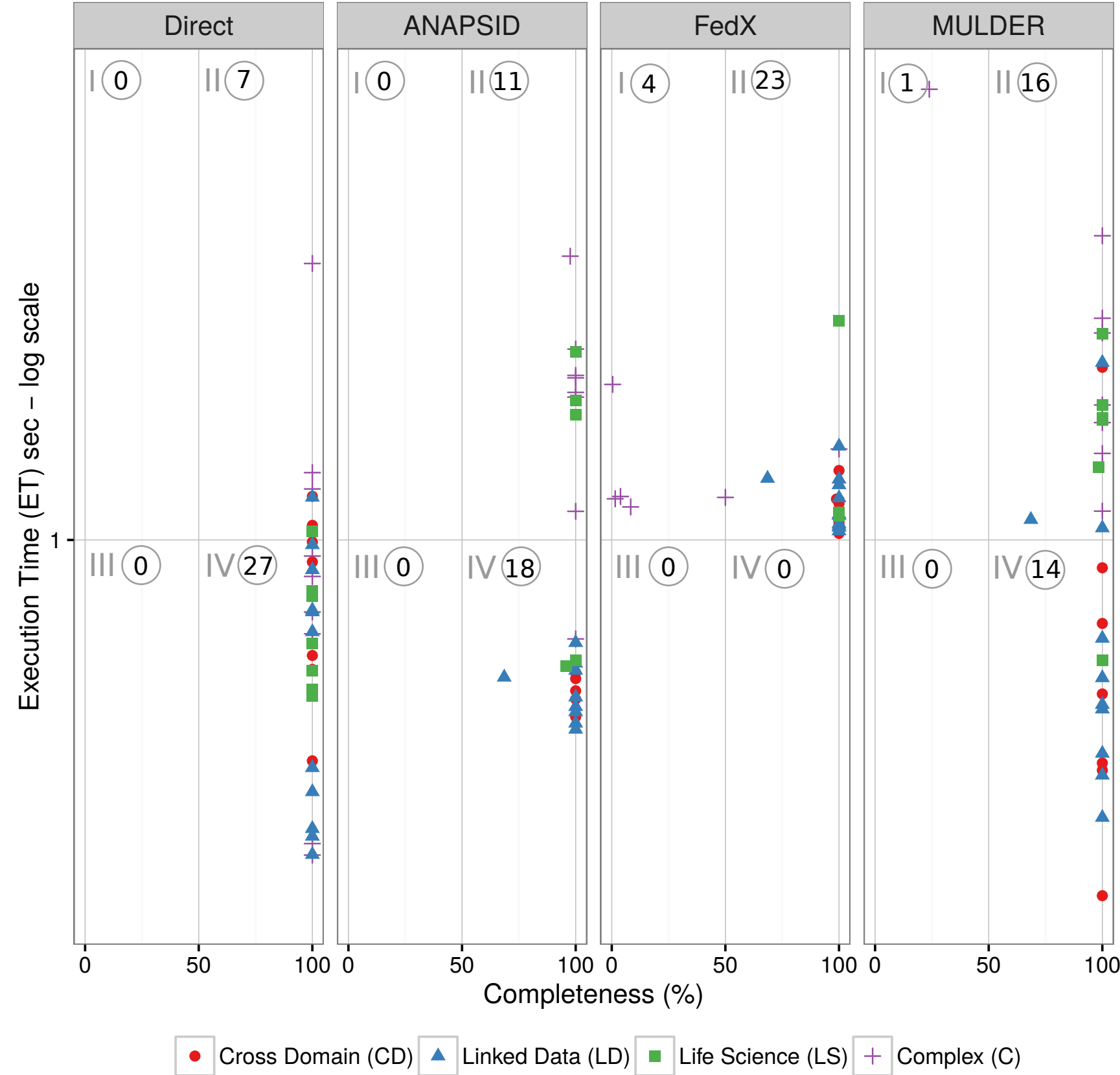
BSBM 400M TRIPLES

FedBench RDF Molecules mined by MULDER



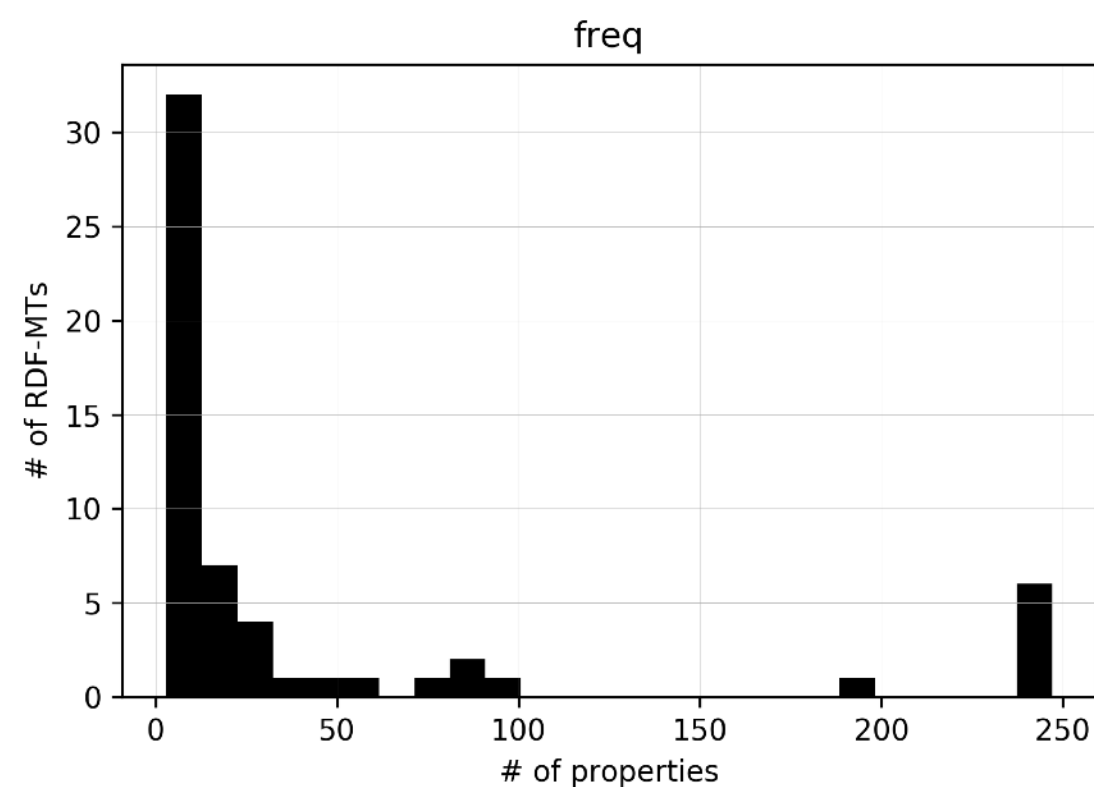
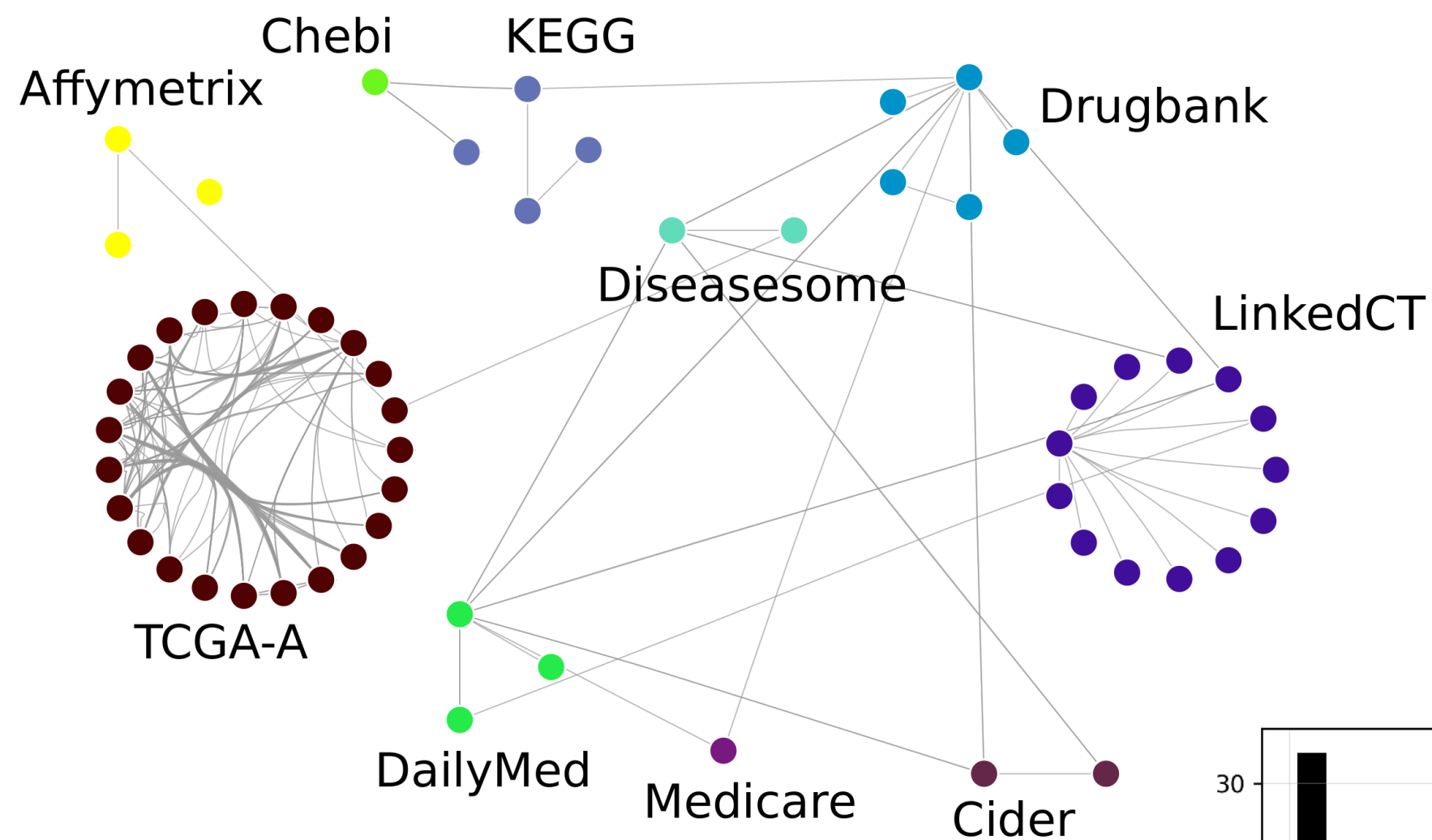


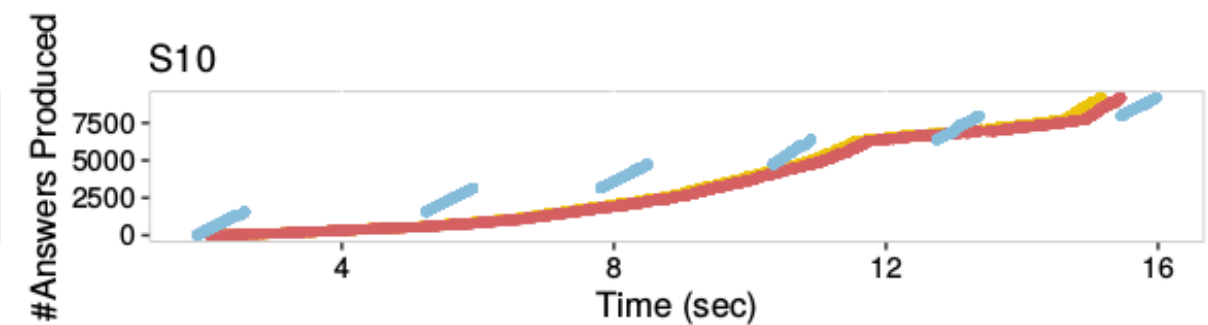
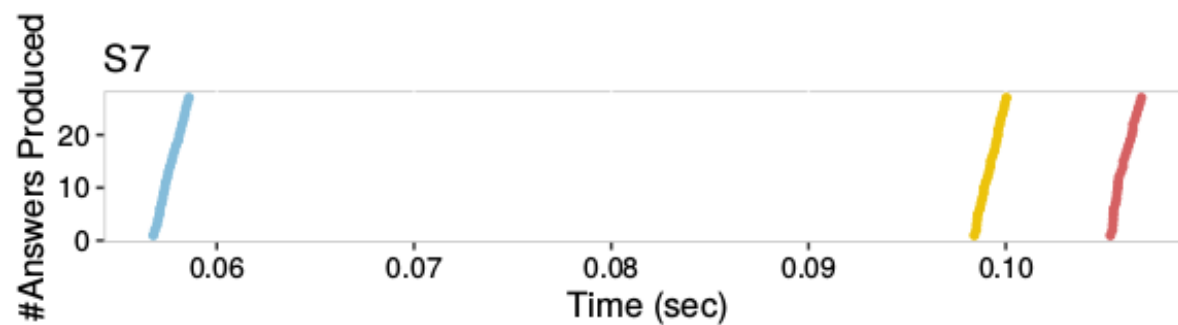
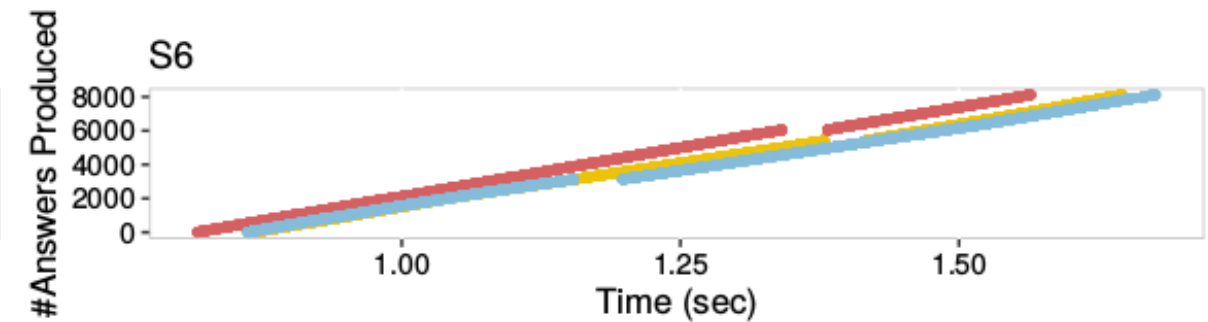
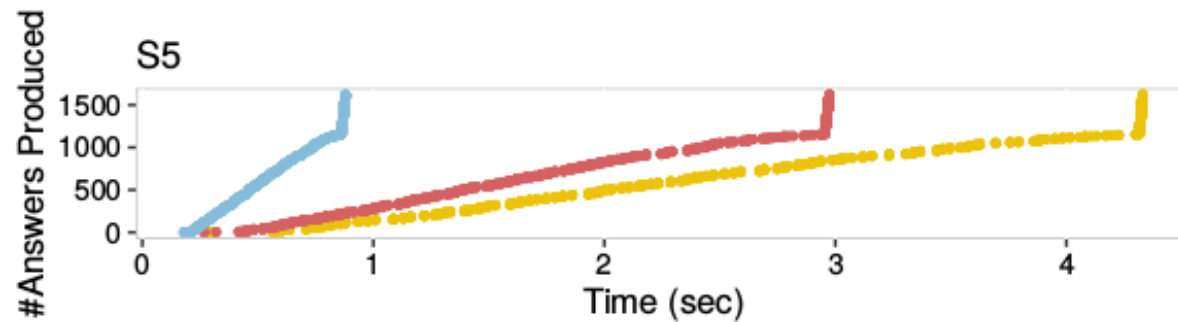
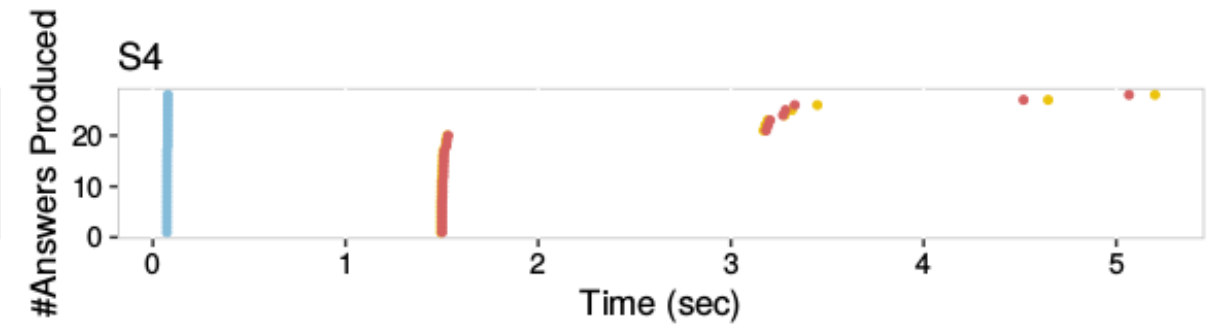
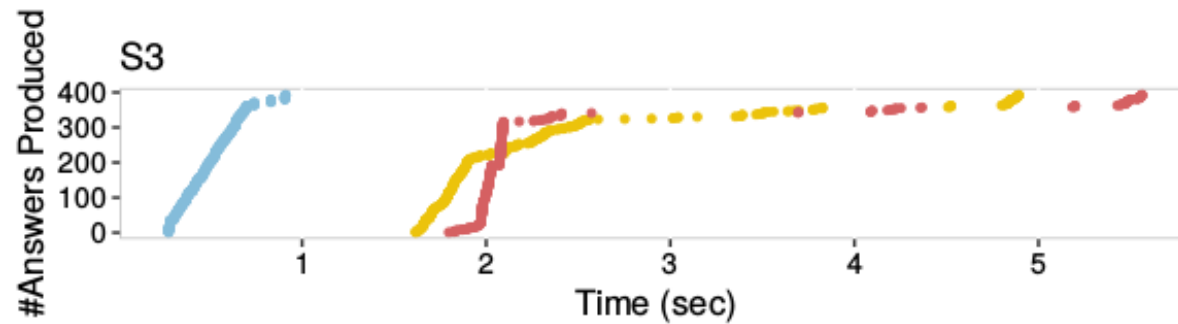
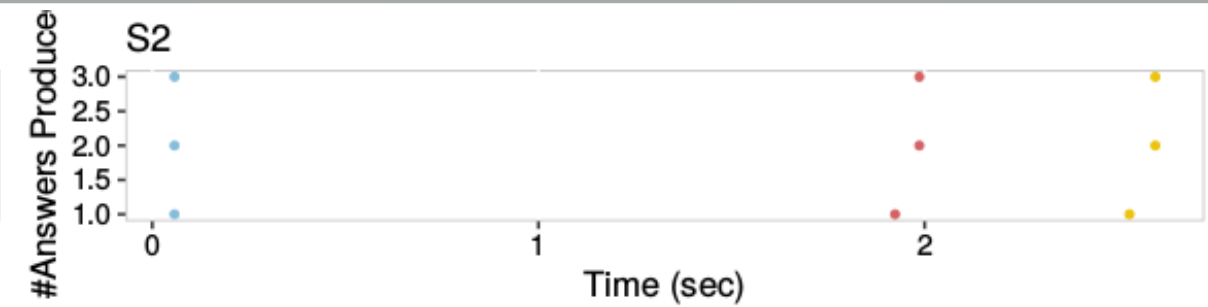
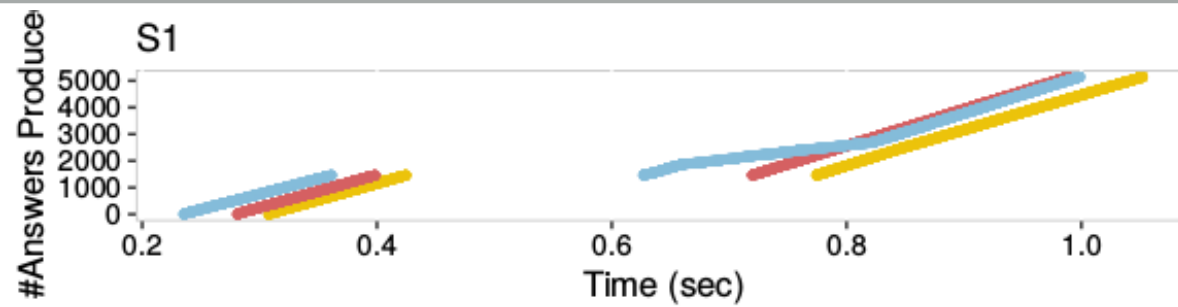
FEDBENCH

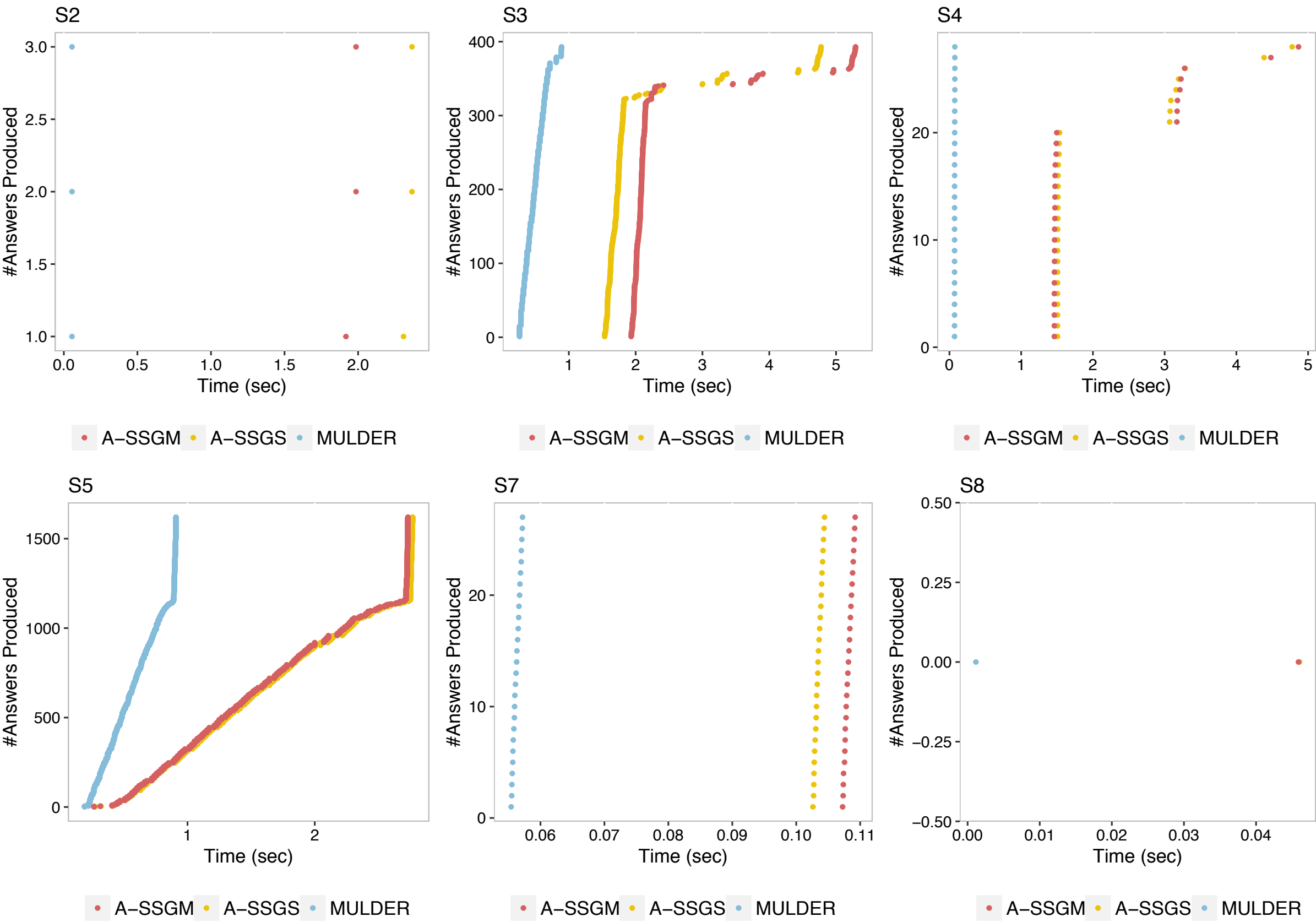


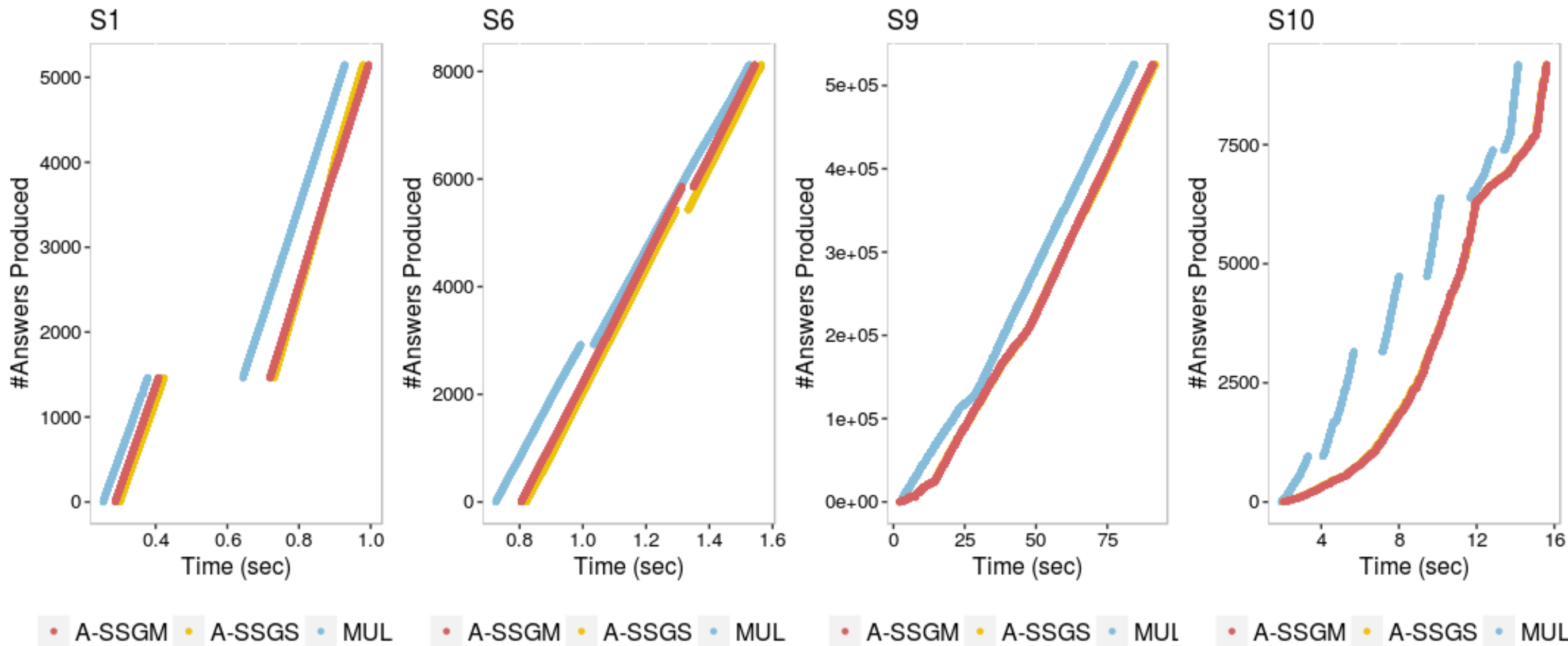
FEDBENCH THROUGHPUT

LSLOD RDF Molecules mined by MULDER









SPARQL MULTI-WAY JOIN

3. M. Galkin, K. Endris, M. E. Vidal, S. Auer. *SMJoin - A Multi-way Join Operator for SPARQL Queries*. **SEMANTICS 2017**
4. M. Galkin, M. E. Vidal, S. Auer. *BatWAn: A Binary and Multi-way Query Plan Analyzer*. **Demo, Poster @ ISWC 2017)**

SPARQL MULTI-WAY JOIN

SPARQL MULTI-WAY JOIN

Stars of two triple patterns

```
SELECT ?drug ?target ?ref ?Int WHERE {  
  t1 ?drug db:genericName ?name .  
  t2 ?drug db:target ?target .  
  t3 ?target db:drugReference ?ref .  
  t4 ?target db:proteinSequence ?ps .  
  t5 ?ref rdfs:label ?refLabel .  
  t6 ?ref foaf:page ?page .  
  t7 ?Int db:interactionDrug1 ?drug .  
  t8 ?Int db:interactionDrug2 ?intd. }
```

SPARQL MULTI-WAY JOIN

Stars of two triple patterns

```
SELECT ?drug ?target ?ref ?Int WHERE {
  t1 ?drug db:genericName ?name .
  t2 ?drug db:target ?target .
  t3 ?target db:drugReference ?ref .
  t4 ?target db:proteinSequence ?ps .
  t5 ?ref rdfs:label ?refLabel .
  t6 ?ref foaf:page ?page .
  t7 ?Int db:interactionDrug1 ?drug .
  t8 ?Int db:interactionDrug2 ?intd. }
```

Stars of three triple patterns

```
SELECT ?person ?city WHERE {
  t1 ?person :name ?name .
  t2 ?person :birthDate ?bday .
  t3 ?person :birthPlace ?city .
  t4 ?city :label ?label .
  t5 ?city :population ?population .
  t6 ?city :country :Germany . }
```

SPARQL MULTI-WAY JOIN

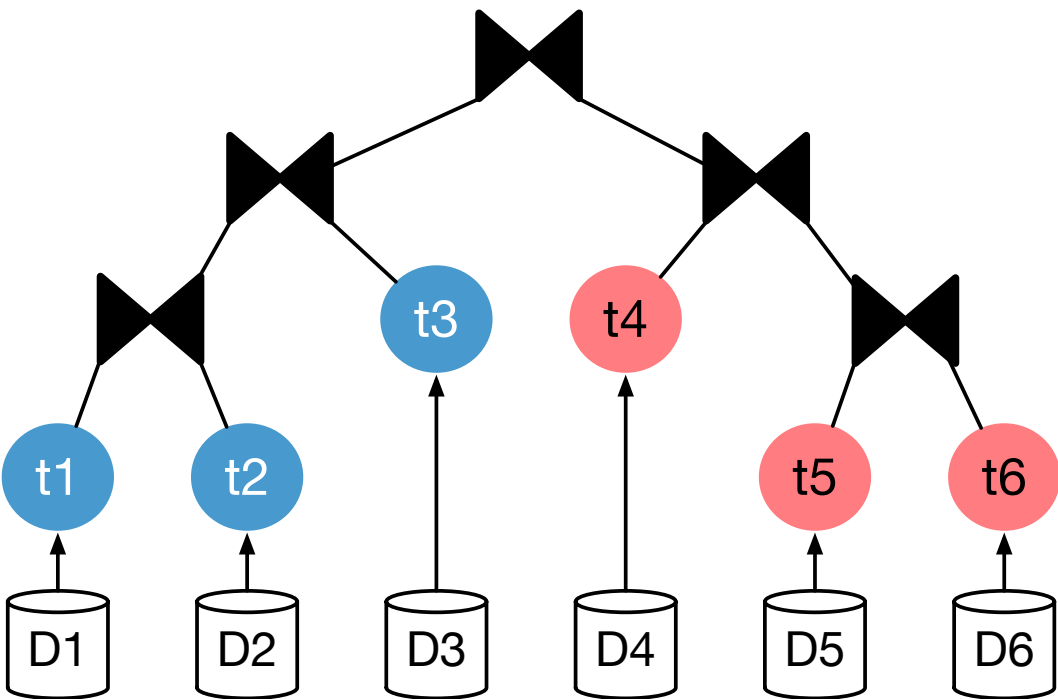
Stars of two triple patterns

SELECT ?drug ?target ?ref ?Int **WHERE** {
t1 ?drug db:genericName ?name .
t2 ?drug db:target ?target .
t3 ?target db:drugReference ?ref .
t4 ?target db:proteinSequence ?ps .
t5 ?ref rdfs:label ?refLabel .
t6 ?ref foaf:page ?page .
t7 ?Int db:interactionDrug1 ?drug .
t8 ?Int db:interactionDrug2 ?intd. }

Stars of three triple patterns

SELECT ?person ?city **WHERE** {
t1 ?person :name ?name .
t2 ?person :birthDate ?bday .
t3 ?person :birthPlace ?city .
t4 ?city :label ?label .
t5 ?city :population ?population .
t6 ?city :country :Germany .

Bushy Tree Binary Join Plan

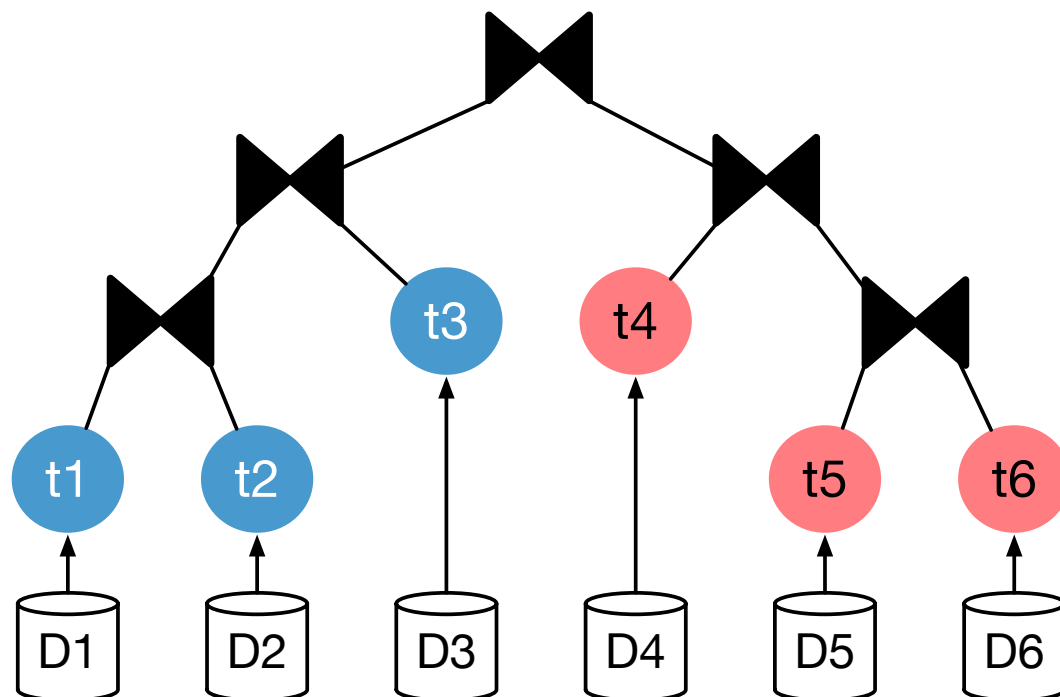


SPARQL MULTI-WAY JOIN

Stars of two triple patterns

```
SELECT ?drug ?target ?ref ?Int WHERE {
  t1 ?drug db:genericName ?name .
  t2 ?drug db:target ?target .
  t3 ?target db:drugReference ?ref .
  t4 ?target db:proteinSequence ?ps .
  t5 ?ref rdfs:label ?refLabel .
  t6 ?ref foaf:page ?page .
  t7 ?Int db:interactionDrug1 ?drug .
  t8 ?Int db:interactionDrug2 ?intd. }
```

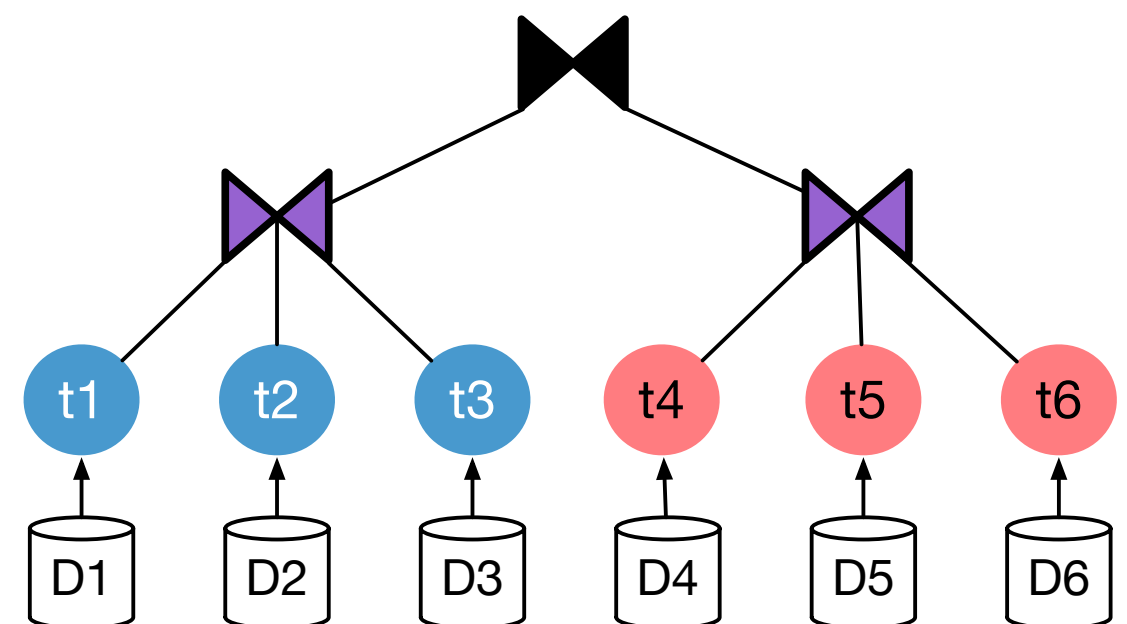
Bushy Tree Binary Join Plan



Stars of three triple patterns

```
SELECT ?person ?city WHERE {
  t1 ?person :name ?name .
  t2 ?person :birthDate ?bday .
  t3 ?person :birthPlace ?city .
  t4 ?city :label ?label .
  t5 ?city :population ?population .
  t6 ?city :country :Germany . }
```

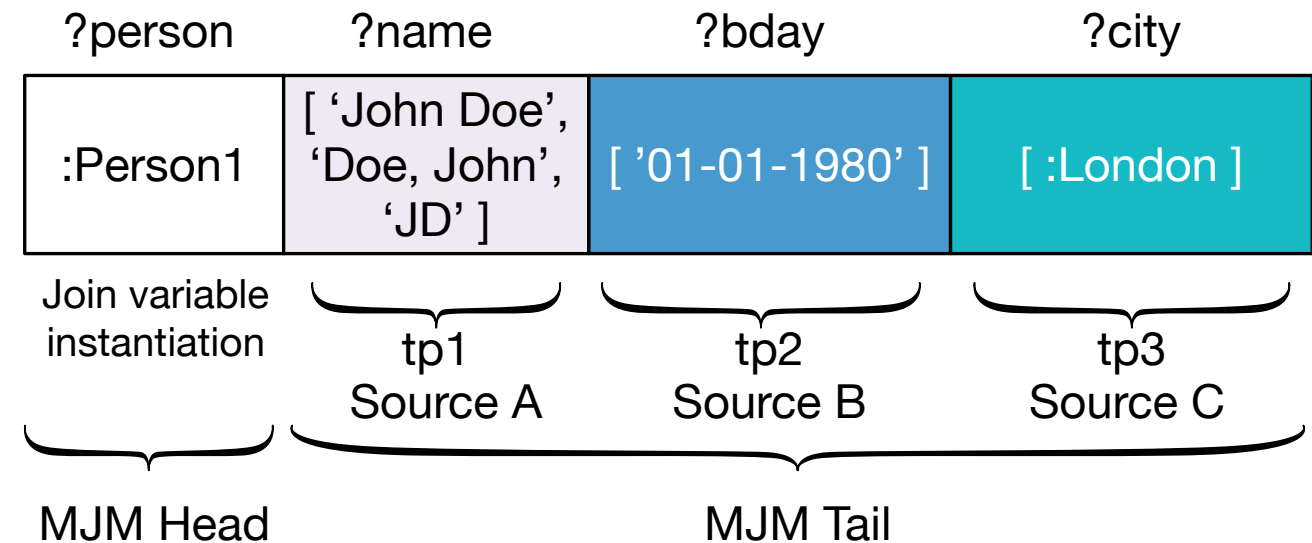
Multi-way Join Plan



SPARQL MULTI-WAY JOIN

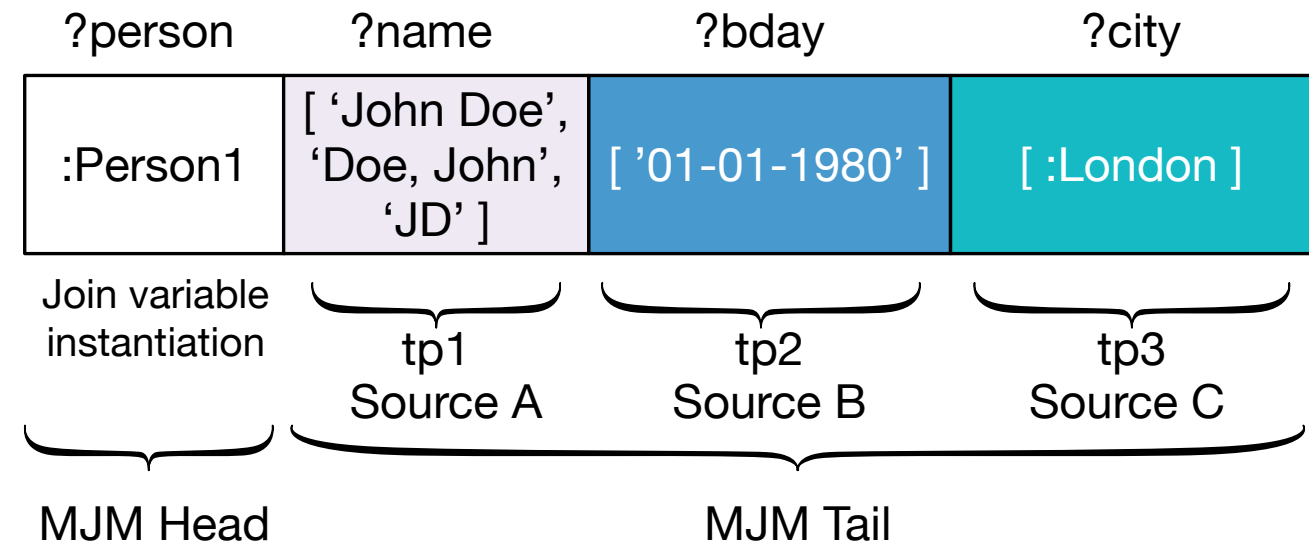
SPARQL MULTI-WAY JOIN

Multi-Join Mapping (MJM)- an internal SMJoin data structure that tracks results produced by the sources after evaluating triple patterns in a query BGP.

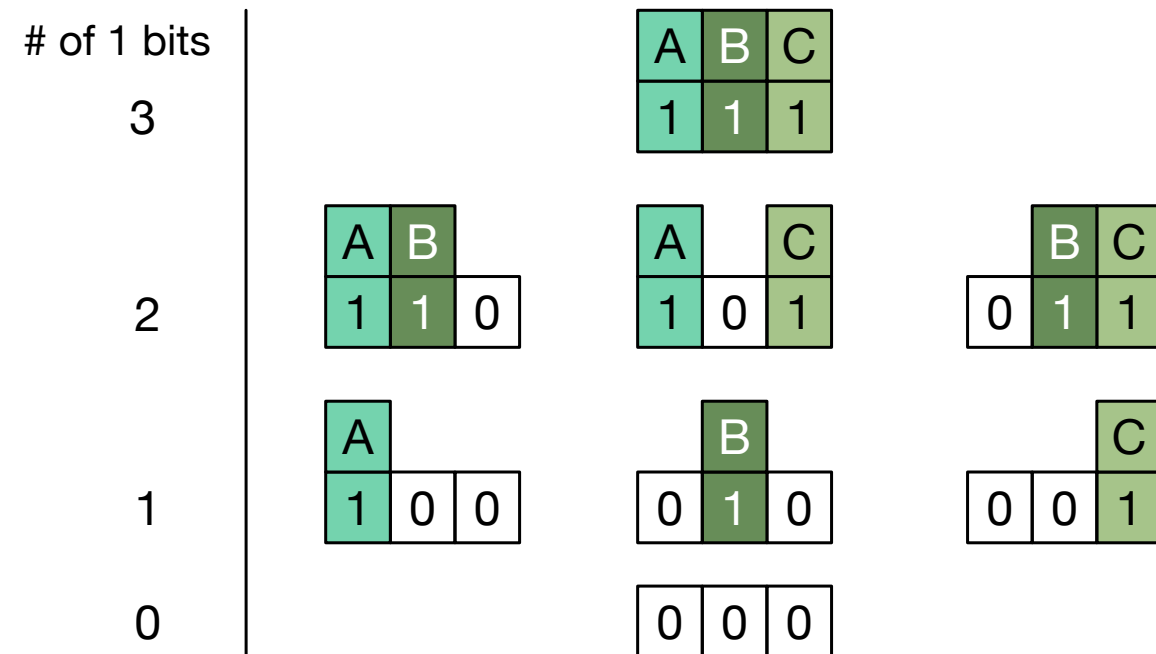


SPARQL MULTI-WAY JOIN

Multi-Join Mapping (MJM)- an internal SMJoin data structure that tracks results produced by the sources after evaluating triple patterns in a query BGP.

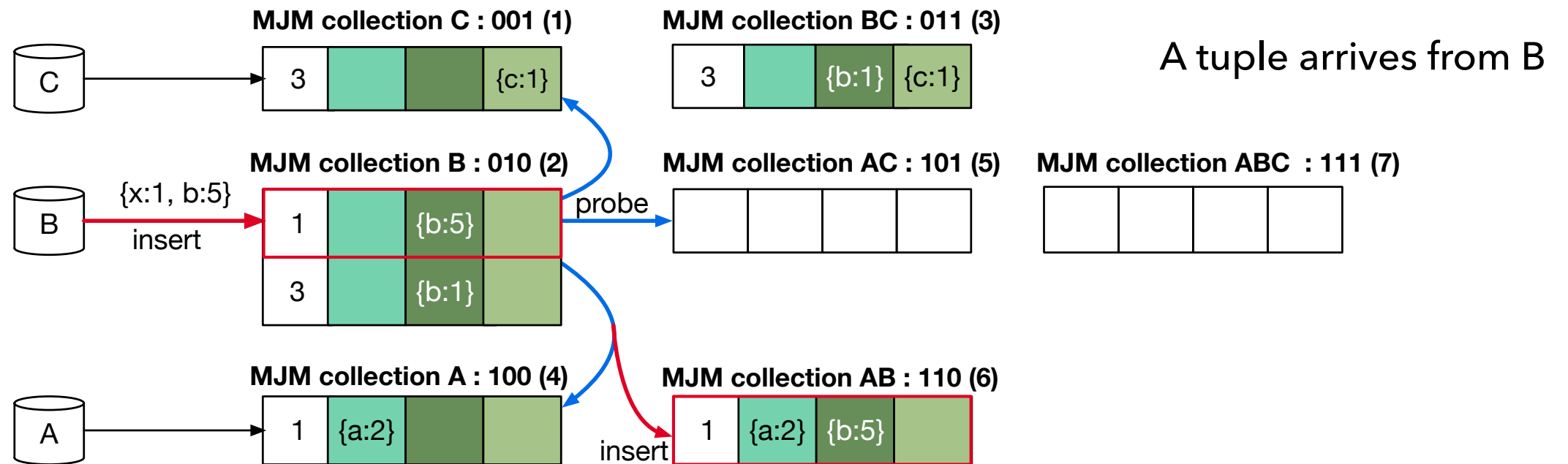


MJMs are organized and indexed as a lattice.

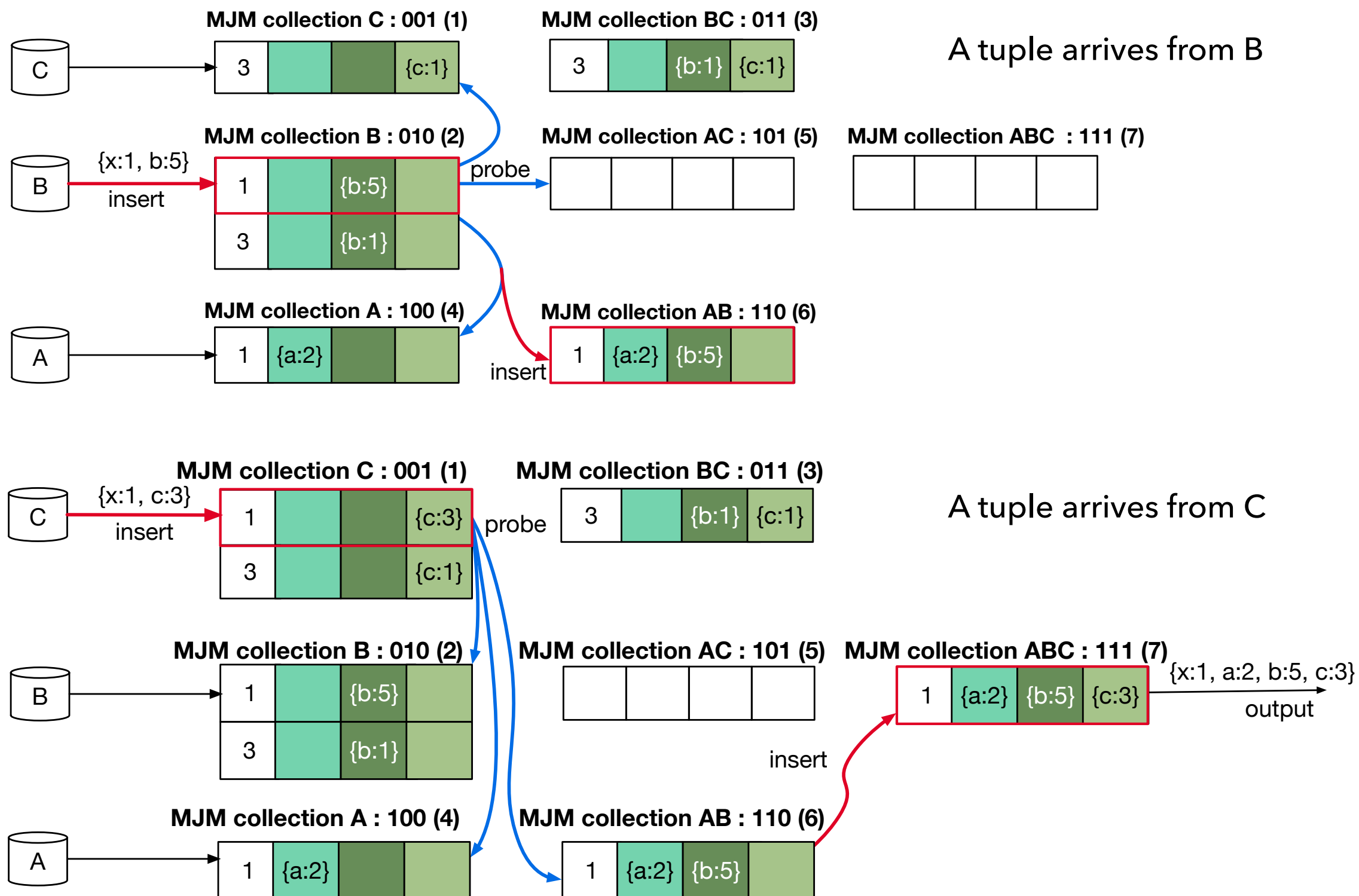


SPARQL MULTI-WAY JOIN

SPARQL MULTI-WAY JOIN

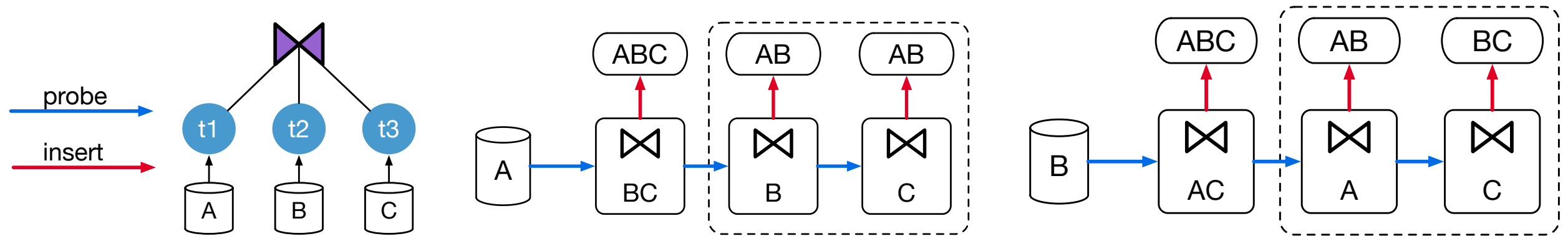


SPARQL MULTI-WAY JOIN



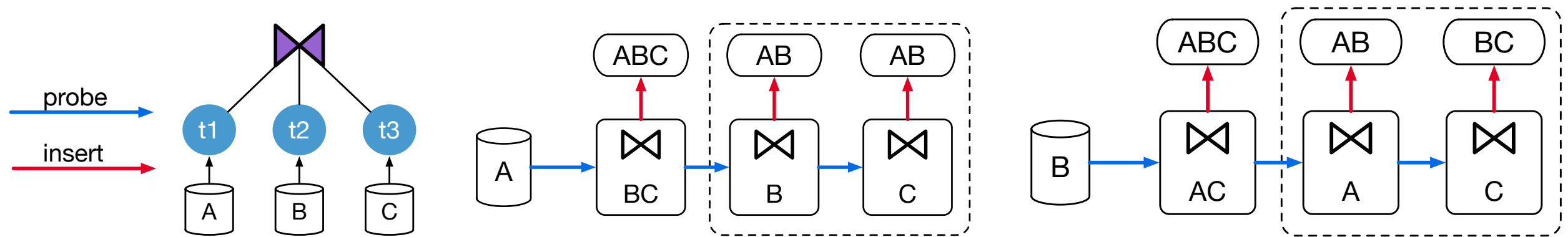
SPARQL MULTI-WAY JOIN

SPARQL MULTI-WAY JOIN



The SMJoin Router manages a probing sequence in a special order that facilitates the fastest output rate

SPARQL MULTI-WAY JOIN

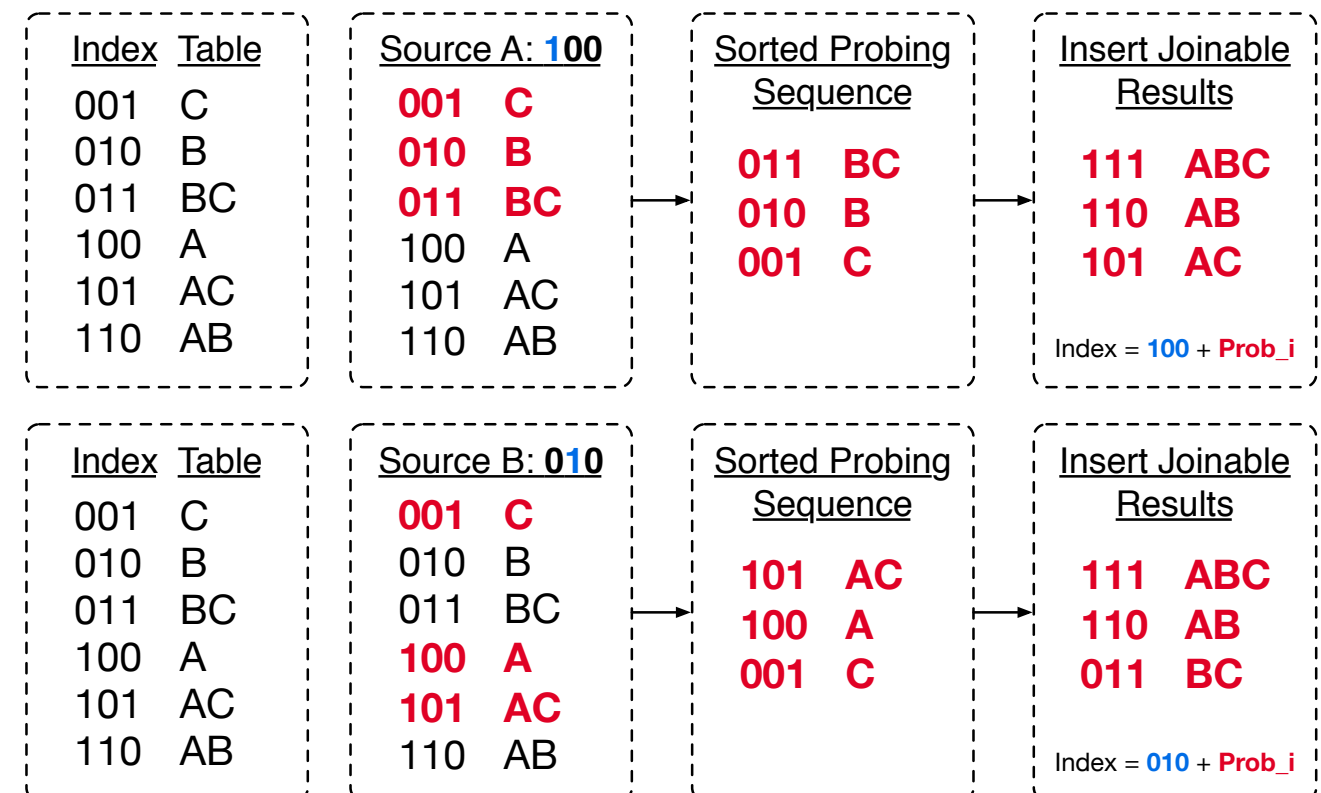


The SMJoin Router manages a probing sequence in a special order that facilitates the fastest output rate

The SMJoin Router:

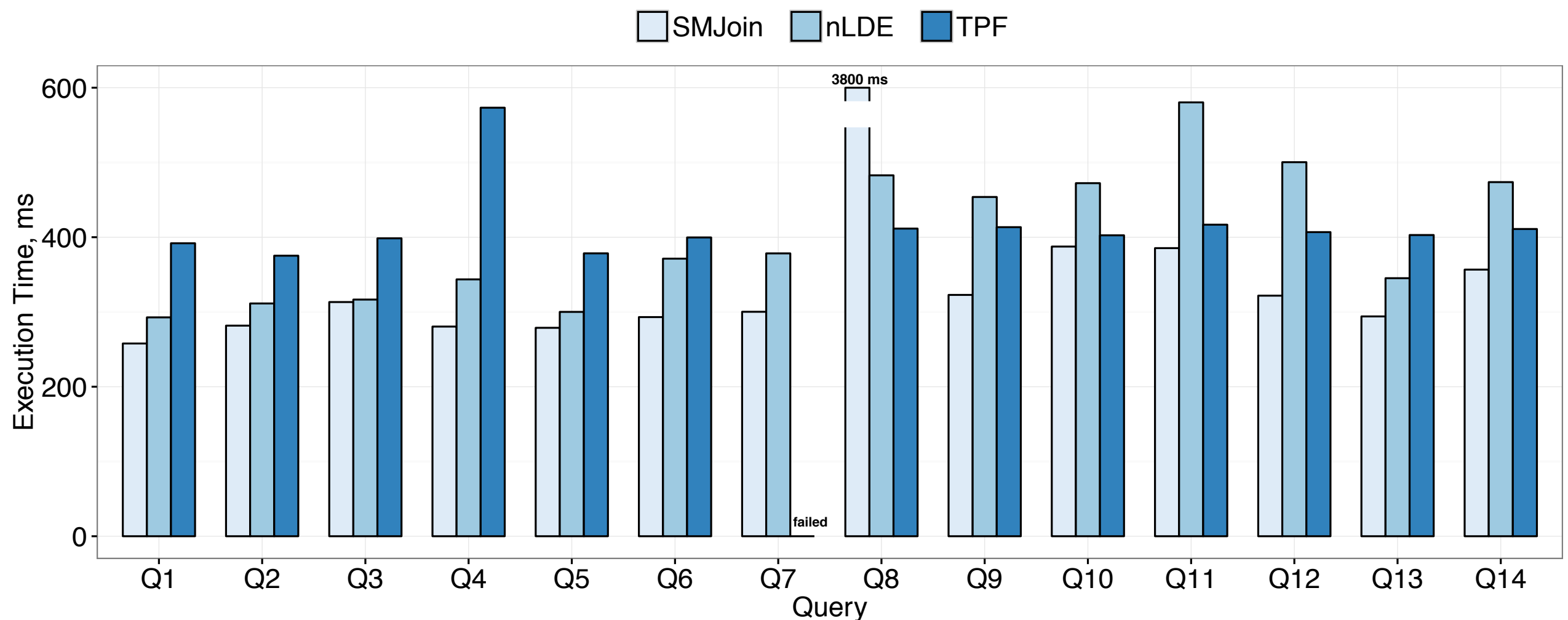
1. Selects only relevant MJM collections for the probing sequence
2. Sorts MJM collections in a desc order by the amount of 1 bits in a binary representation
3. Inserts a joinable result into an MJM collections with the index:

$$\text{Insert}_i = \text{Source OR Probed}$$

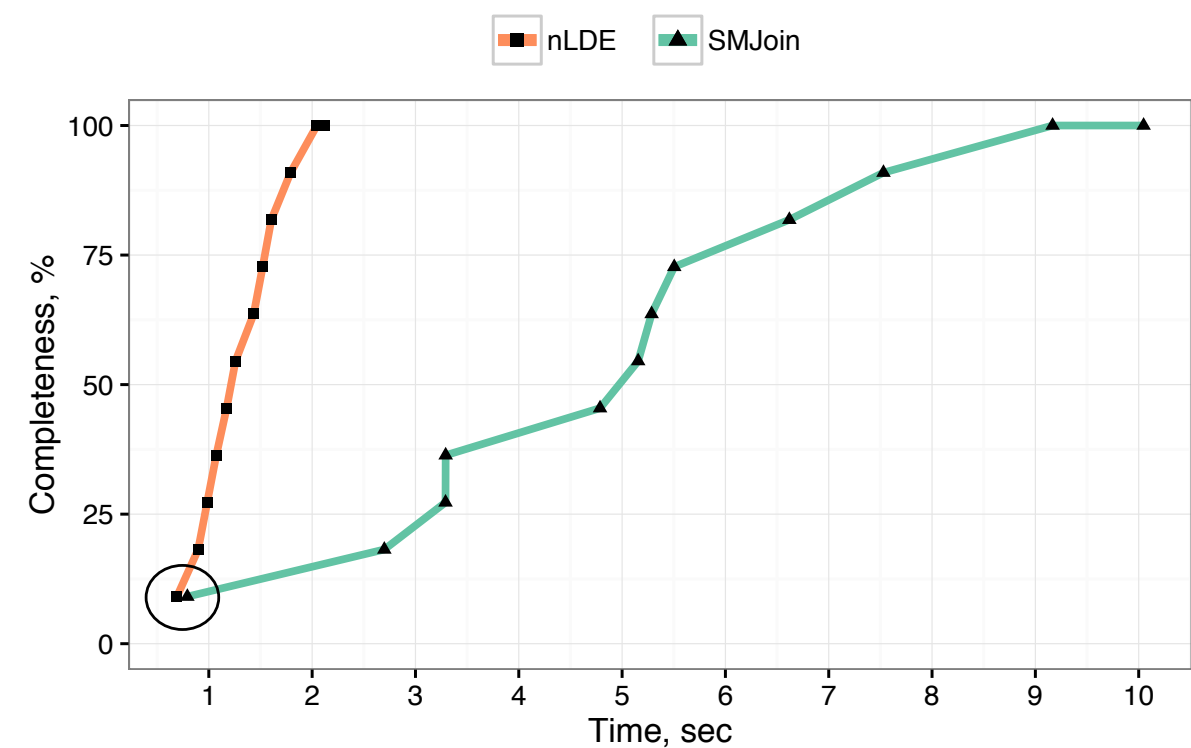
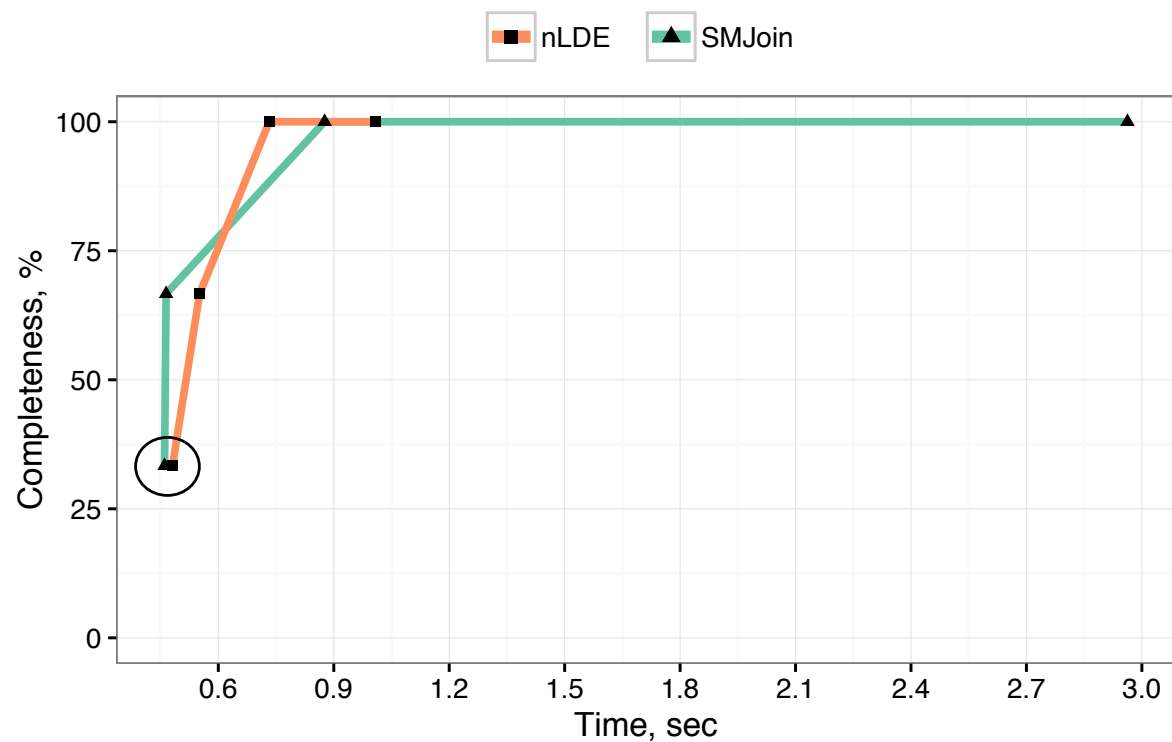
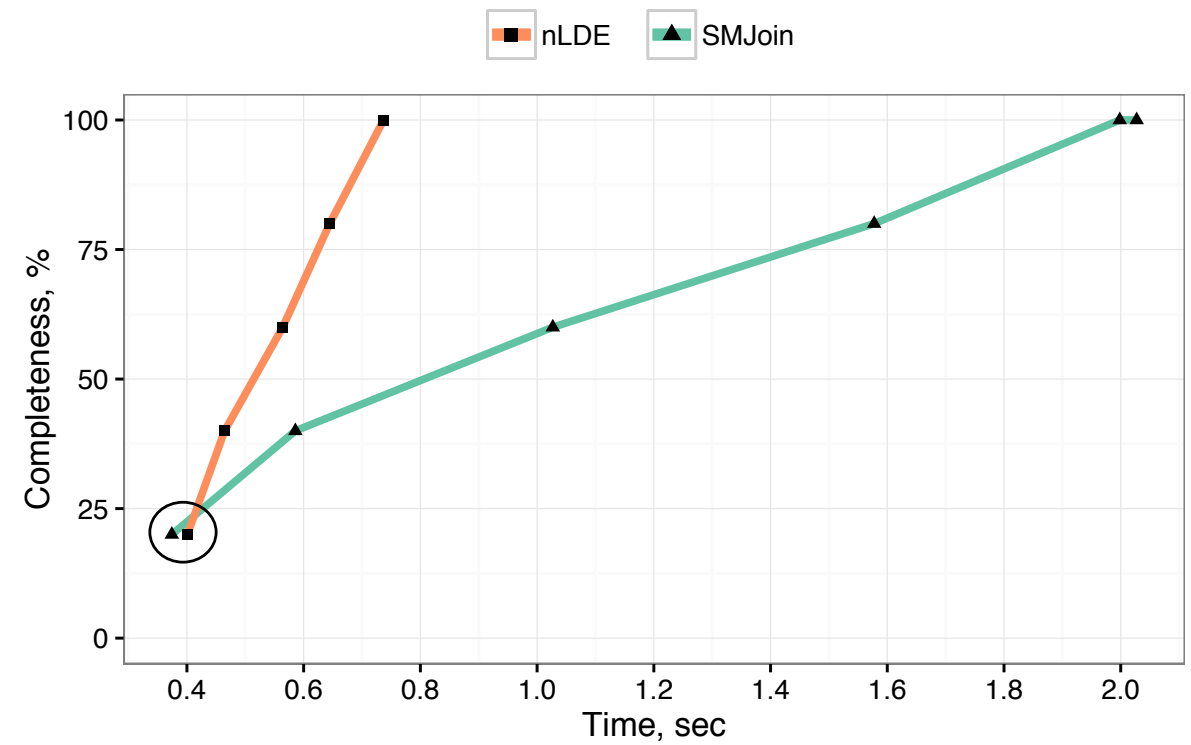
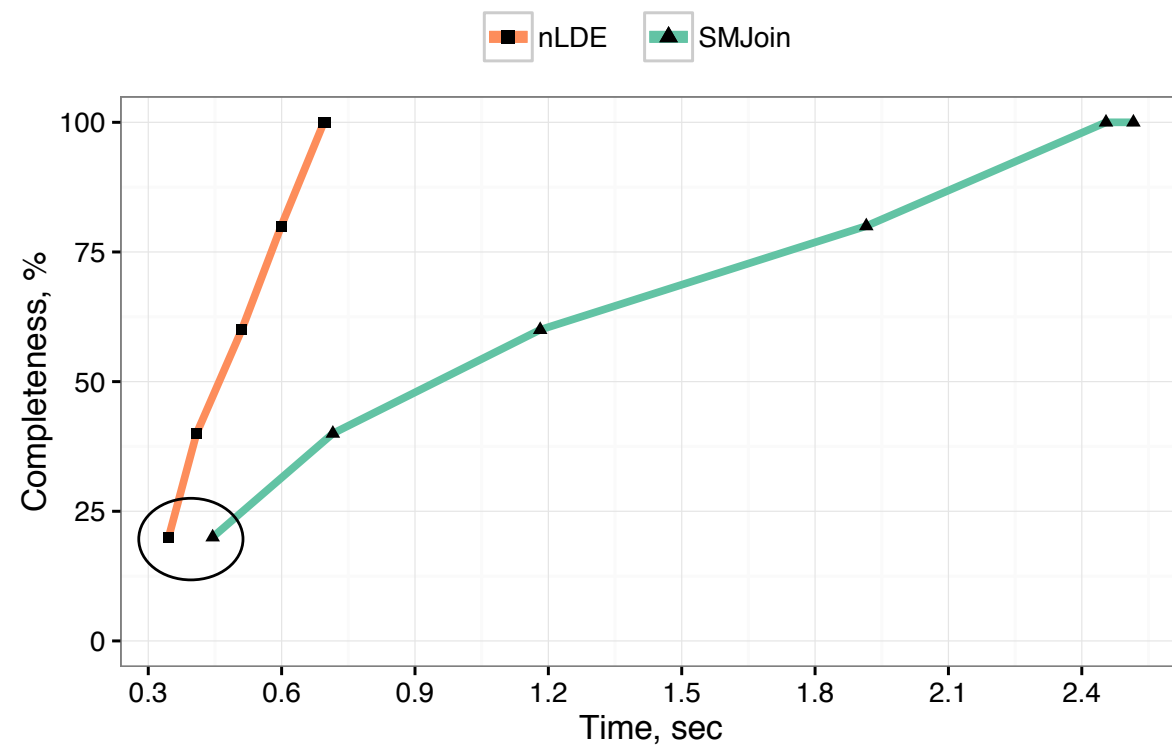


EVALUATION

- ▶ Dataset: DBpedia 2015-10, 800M triples, HDT format
- ▶ Compared systems: MULDER + SMJoin, nLDE, TPF Client



EVALUATION



QUERY PLAN OPTIMIZATION

MOTIVATION

Query execution conditions may drastically affect plan's performance

MOTIVATION

Query execution conditions may drastically affect plan's performance

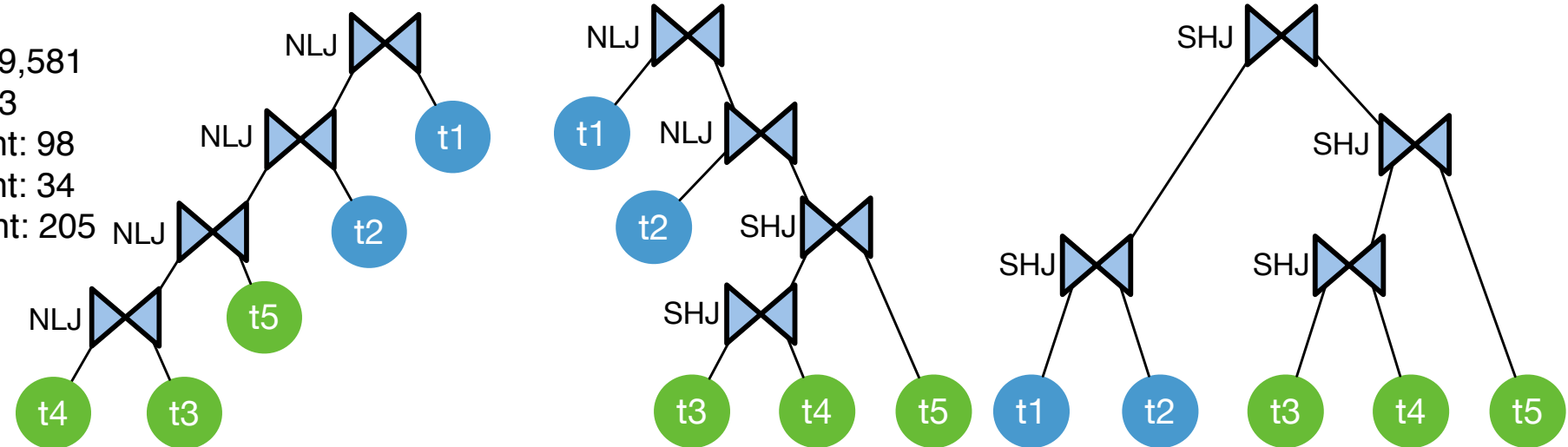
SELECT DISTINCT * WHERE {

t1 ?s dct:subject ?o1. #Count: 115,259,581
t2 ?s dbo:director ?o2. #Count: 385,773
t3 ?s dbo:genre dbr:Concert. #Count: 98
t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
t5 ?s dbo:genre dbr:Ranchera. } #Count: 205

MOTIVATION

Query execution conditions may drastically affect plan's performance

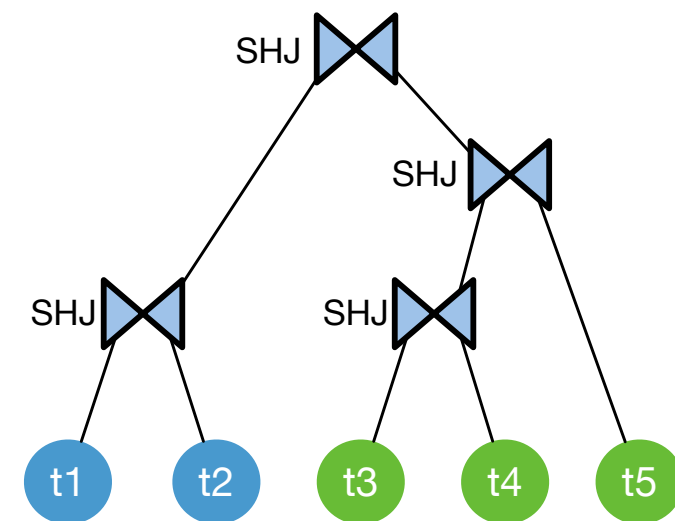
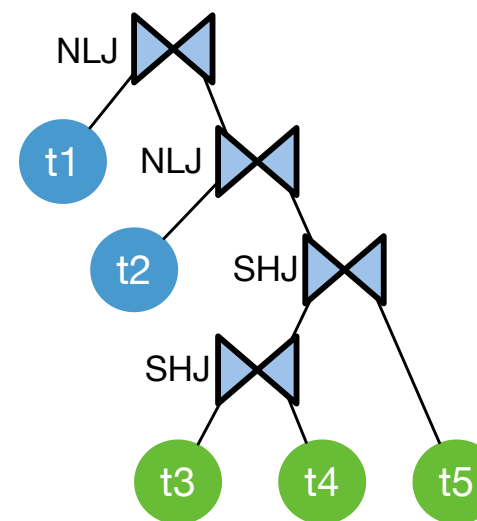
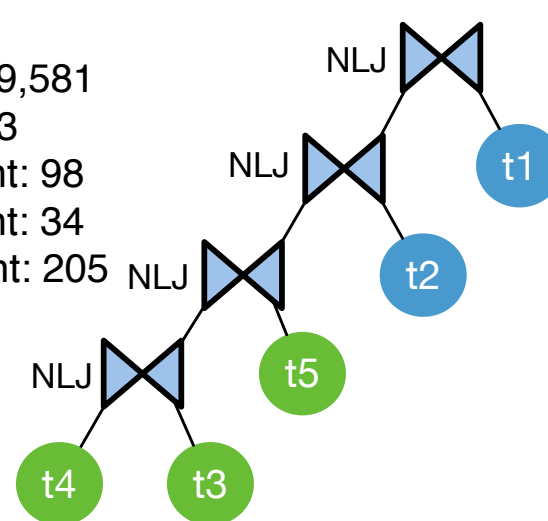
SELECT DISTINCT * WHERE {
 t1 ?s dct:subject ?o1. #Count: 115,259,581
 t2 ?s dbo:director ?o2. #Count: 385,773
 t3 ?s dbo:genre dbr:Concert. #Count: 98
 t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
 t5 ?s dbo:genre dbr:Ranchera. } #Count: 205



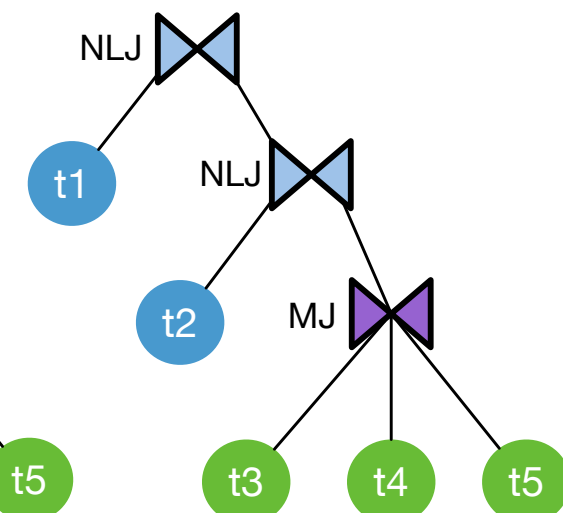
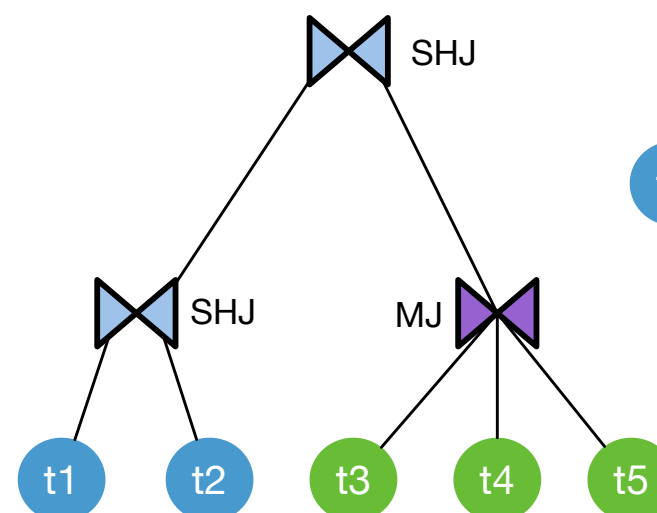
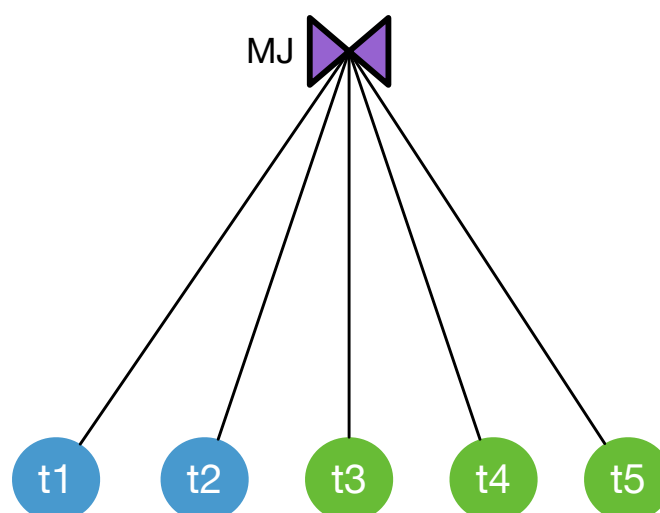
MOTIVATION

Query execution conditions may drastically affect plan's performance

SELECT DISTINCT * WHERE {
 t1 ?s dct:subject ?o1. #Count: 115,259,581
 t2 ?s dbo:director ?o2. #Count: 385,773
 t3 ?s dbo:genre dbr:Concert. #Count: 98
 t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
 t5 ?s dbo:genre dbr:Ranchera. } #Count: 205



Plan	ET, ms	Results
NLJs, (b)	634	12
SHJs, (c)	timeout	0
NLJs + SHJs, (d)	243	12
MJ-only, (f)	timeout	0
MJ + SHJs, (g)	timeout	0
MJ + NLJs, (h)	195	12



APPROACH

Adaptive query plans that are able to combine binary and multi-way joins

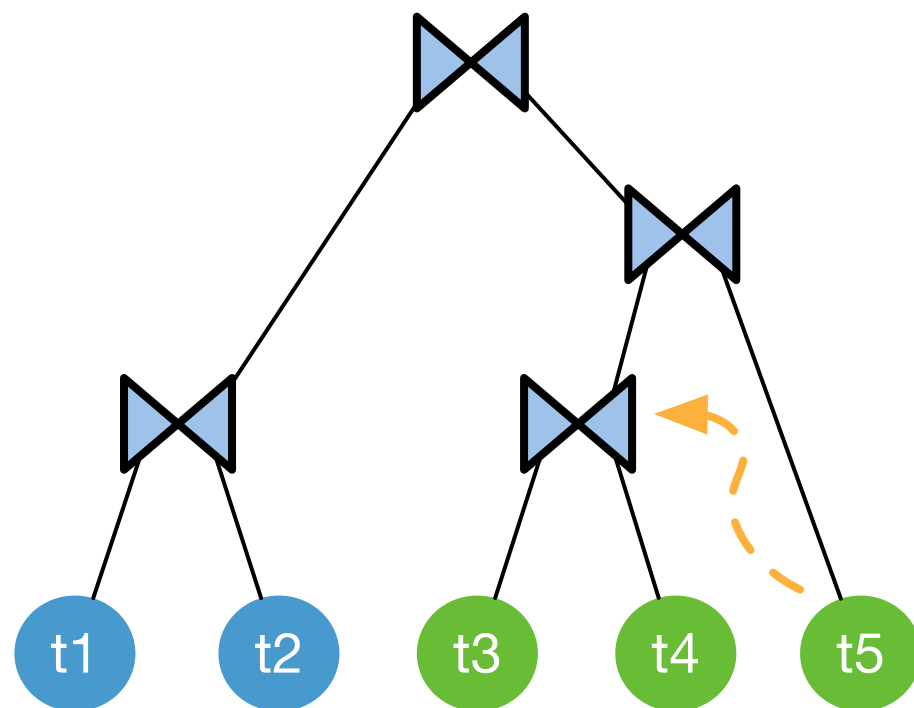
```
SELECT DISTINCT * WHERE {  
  t1 ?s dct:subject ?o1.    #Count: 115,259,581  
  t2 ?s dbo:director ?o2.    #Count: 385,773  
  t3 ?s dbo:genre    dbr:Concert.    #Count: 98  
  t4 ?s dbo:artist    dbr:Ana_Gabriel. #Count: 34  
  t5 ?s dbo:genre    dbr:Ranchera. } #Count: 205
```

APPROACH

Adaptive query plans that are able to combine binary and multi-way joins

SELECT DISTINCT * WHERE {

t1 ?s dct:subject ?o1. #Count: 115,259,581
t2 ?s dbo:director ?o2. #Count: 385,773
t3 ?s dbo:genre dbr:Concert. #Count: 98
t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
t5 ?s dbo:genre dbr:Ranchera. } #Count: 205

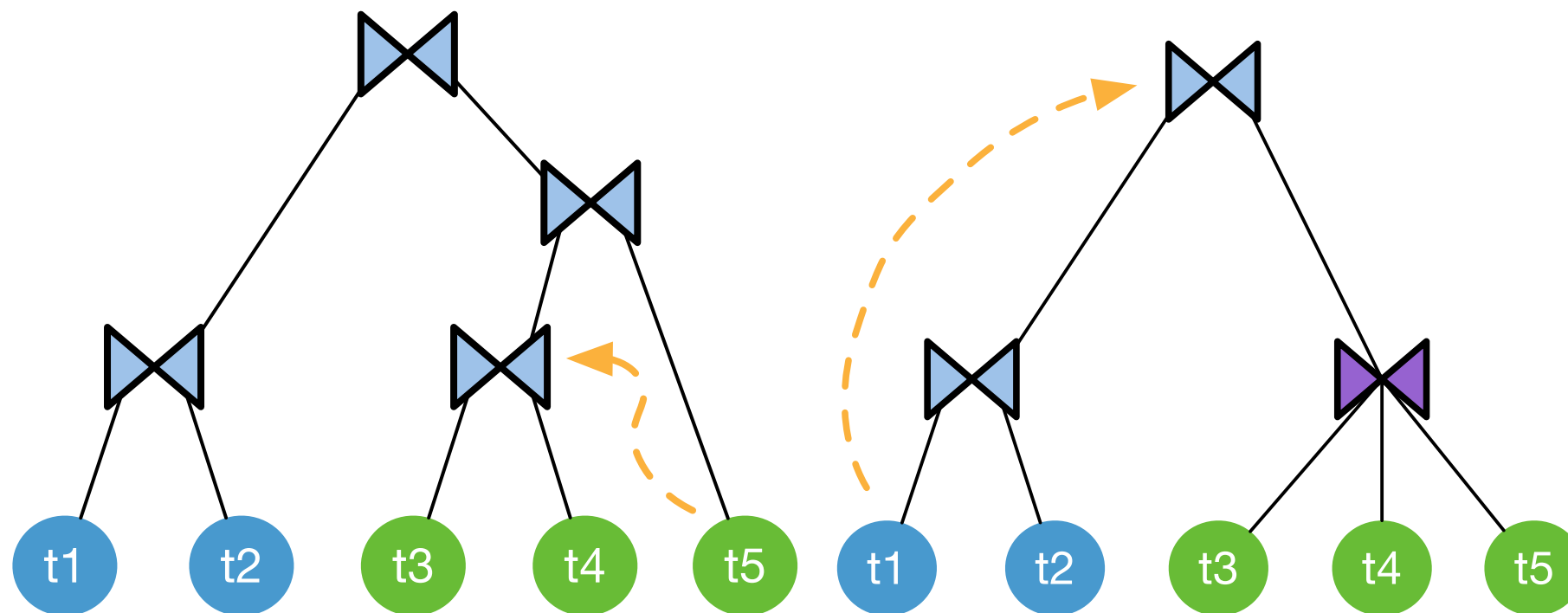


APPROACH

Adaptive query plans that are able to combine binary and multi-way joins

SELECT DISTINCT * WHERE {

t1 ?s dct:subject ?o1. #Count: 115,259,581
 t2 ?s dbo:director ?o2. #Count: 385,773
 t3 ?s dbo:genre dbr:Concert. #Count: 98
 t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
 t5 ?s dbo:genre dbr:Ranchera. } #Count: 205

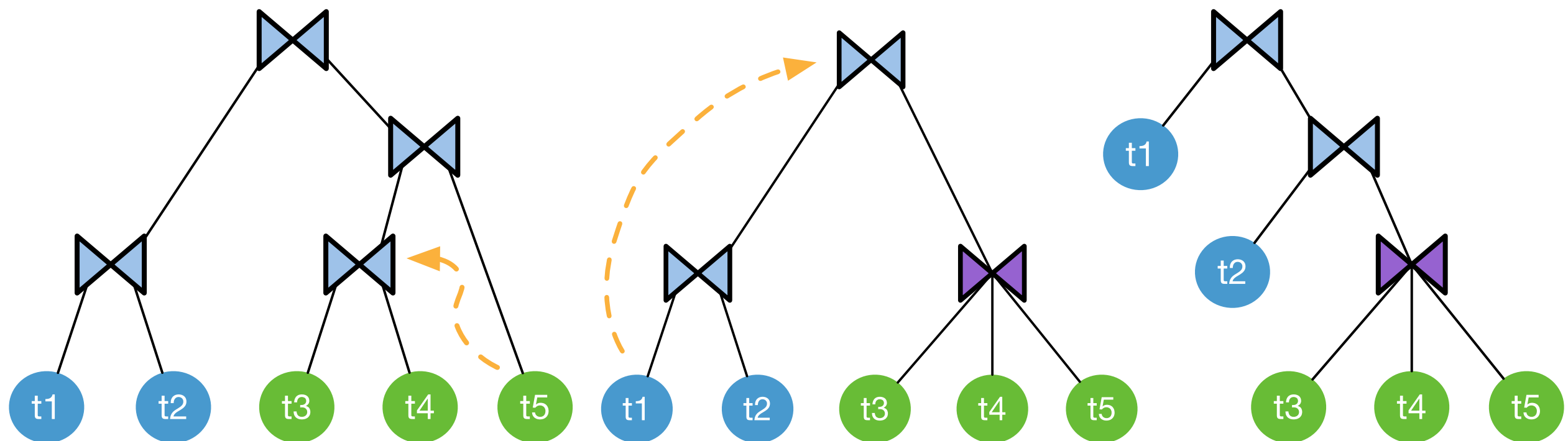


APPROACH

Adaptive query plans that are able to combine binary and multi-way joins

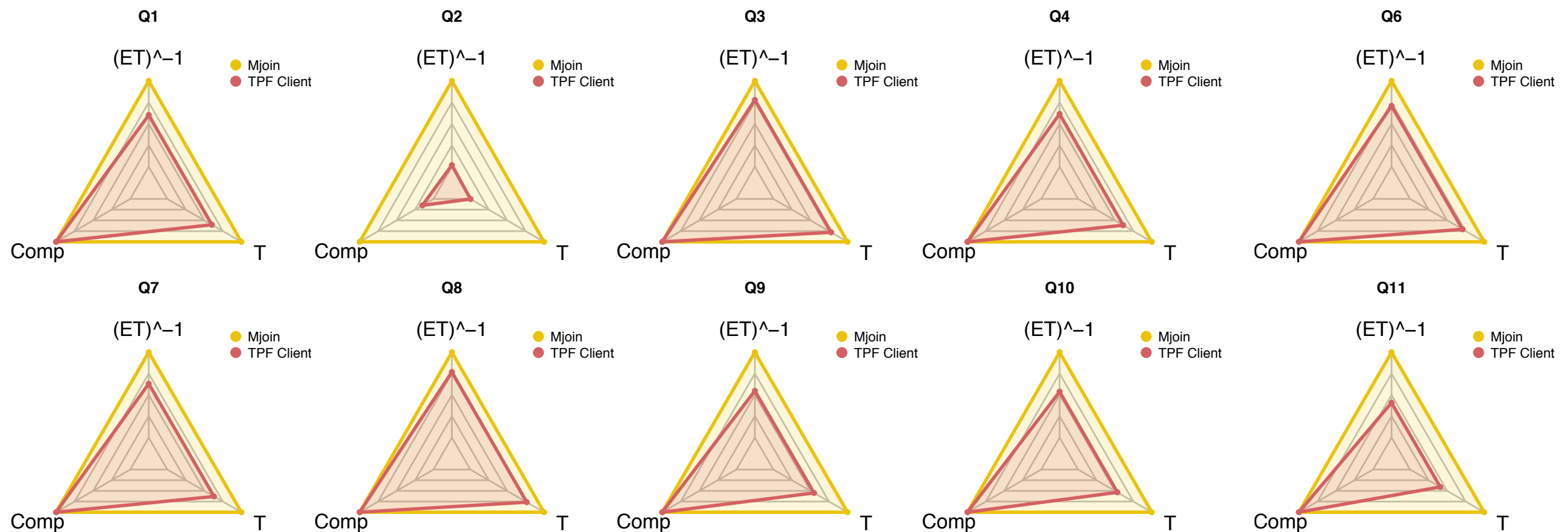
SELECT DISTINCT * WHERE {

t1 ?s dct:subject ?o1. #Count: 115,259,581
 t2 ?s dbo:director ?o2. #Count: 385,773
 t3 ?s dbo:genre dbr:Concert. #Count: 98
 t4 ?s dbo:artist dbr:Ana_Gabriel. #Count: 34
 t5 ?s dbo:genre dbr:Ranchera. } #Count: 205



EXPERIMENTS

- ▶ Dataset: DBpedia 2015-10, 800M triples, HDT format
- ▶ Compared systems: nLDE + SMJoin, TPF Client

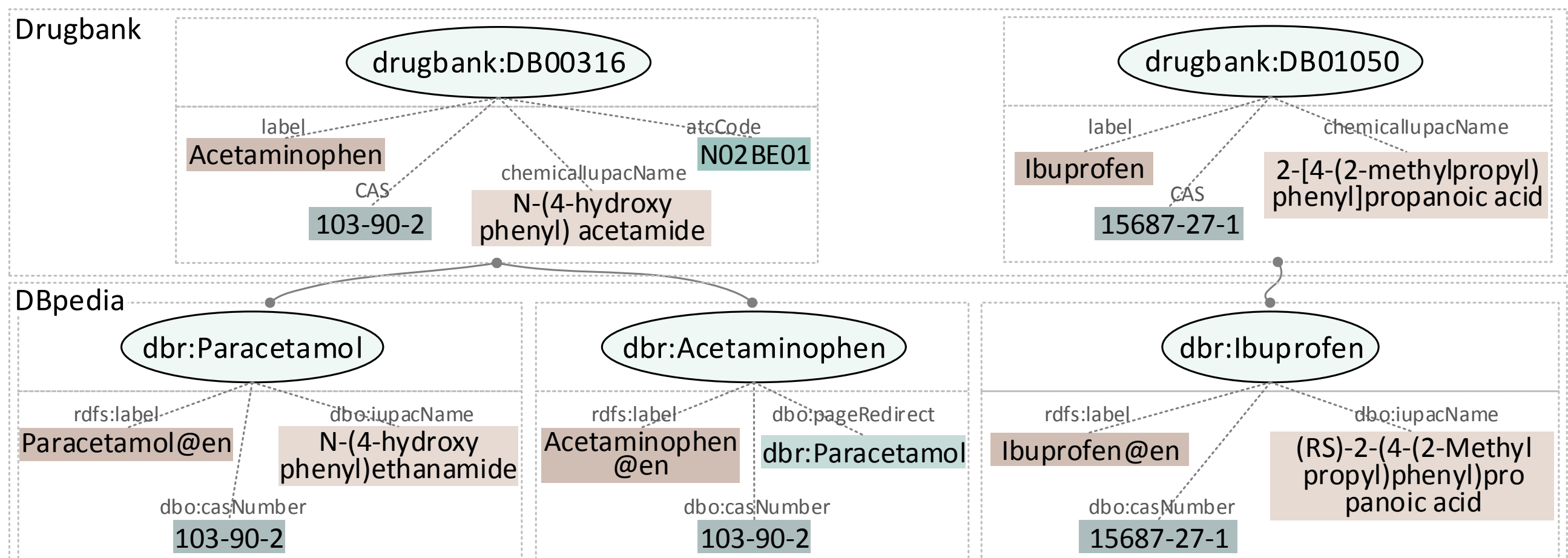


SEMANTIC JOIN OPERATOR

5. M. Galkin, D. Collarana, I. Traverso, M. E. Vidal, C. Lange, S. Auer. *SJoin: A Semantic Join Operator to Integrate Heterogeneous RDF Graphs*. [DEXA 2017](#)
6. M. Galkin, M. E. Vidal. *Towards a Multi-way Similarity Join Operator*. [ADBIS 2017](#)
7. M. Galkin, M. E. Vidal. *Towards a Multi-way Similarity Join Operator*. Submitted to the Journal of Applied Mathematics and Computer Science ([AMCS Journal](#))

MOTIVATING EXAMPLE

Equivalent RDF entities are encoded using heterogeneous vocabularies (schemas) in different sources



MOTIVATING EXAMPLE

Equivalent RDF entities are encoded using heterogeneous vocabularies (schemas) in different sources

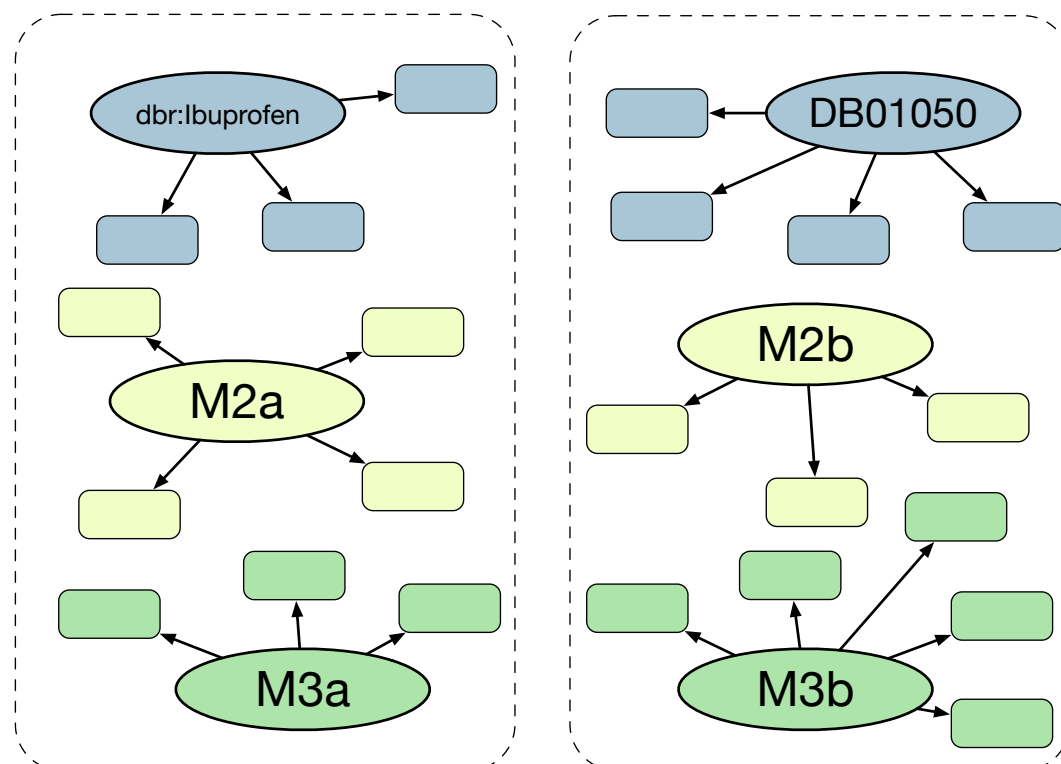
- ▶ Entities from **Drugbank** and **DBpedia** are syntactically different but **semantically equivalent**
- ▶ **DB00316** is 1-to-2 same as **dbr:Paracetamol** and **dbr:Acetaminophen**
- ▶ **DB01050** equals 1-to-1 to **dbr:Ibuprofen**

PROBLEM DEFINITION

Given $\varphi(G)$, $\varphi(D)$ and $\varphi(F)$ as sets of RDF molecules where $M_G \subset \varphi(G)$, $M_D \subset \varphi(D)$, $M_F \subset \varphi(F)$ are molecules, find a homomorphism

PROBLEM DEFINITION

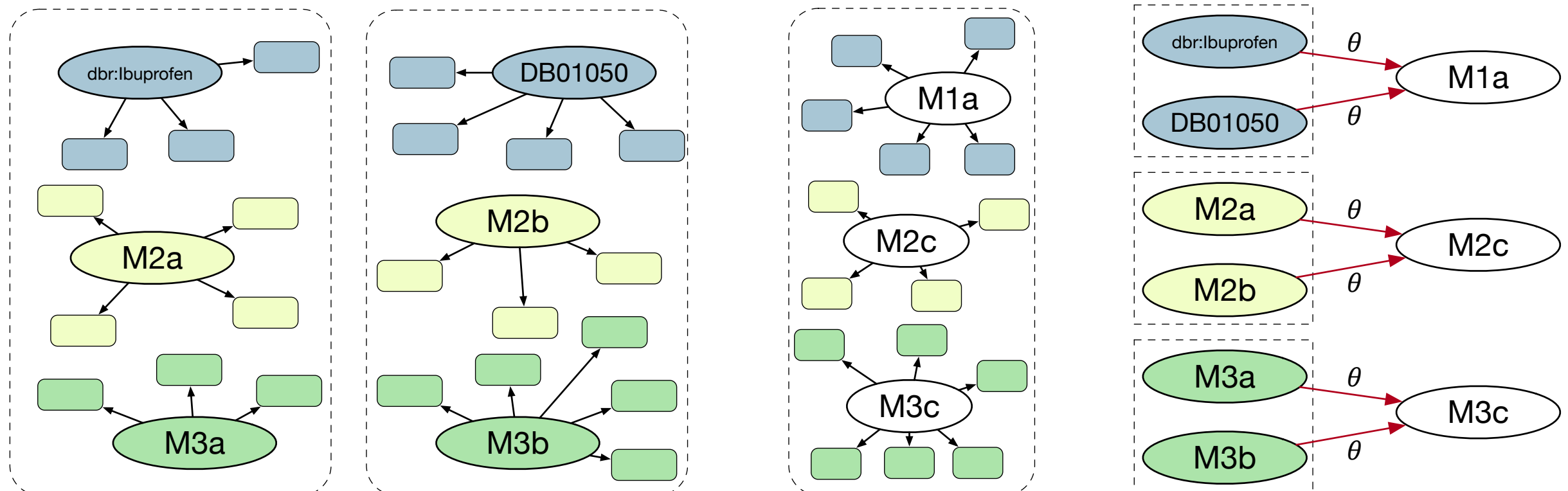
Given $\varphi(G)$, $\varphi(D)$ and $\varphi(F)$ as sets of RDF molecules where $M_G \subset \varphi(G)$, $M_D \subset \varphi(D)$, $M_F \subset \varphi(F)$ are molecules, find a homomorphism



PROBLEM DEFINITION

Given $\phi(G)$, $\phi(D)$ and $\phi(F)$ as sets of RDF molecules where $M_G \subset \phi(G)$, $M_D \subset \phi(D)$, $M_F \subset \phi(F)$ are molecules, find a homomorphism

$$\theta : \phi(G) \cup \phi(D) \rightarrow 2^{\phi(F)}$$



APPROACH

- ▶ SJoin: a semantic join operator that implement the homomorphism
- ▶ Blocking SJoin – for 1-1 weighted perfect matching of RDF molecules
- ▶ Non-Blocking SJoin – for N-M matching of RDF molecules

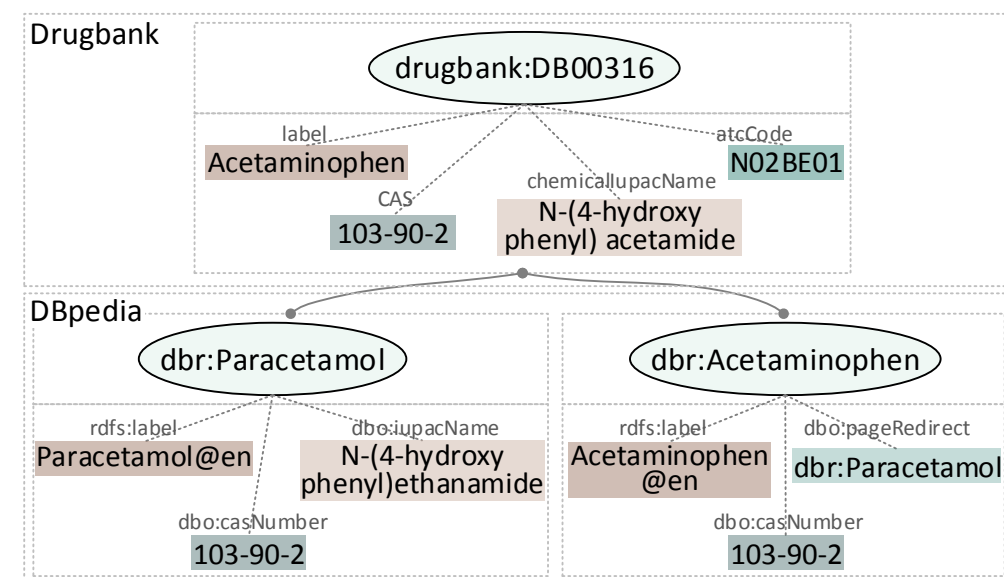
RESOURCE SIMILARITY MOLECULE

RSM is a data structure that maps a molecule from graph A to most similar molecules from graph B

$$\text{RSM} = (\text{URI}_{iA}, \text{Tail}_{iA}) [(\text{URI}_{jB}, \text{Tail}_{jB}), (\text{URI}_{kB}, \text{Tail}_{kB})]$$

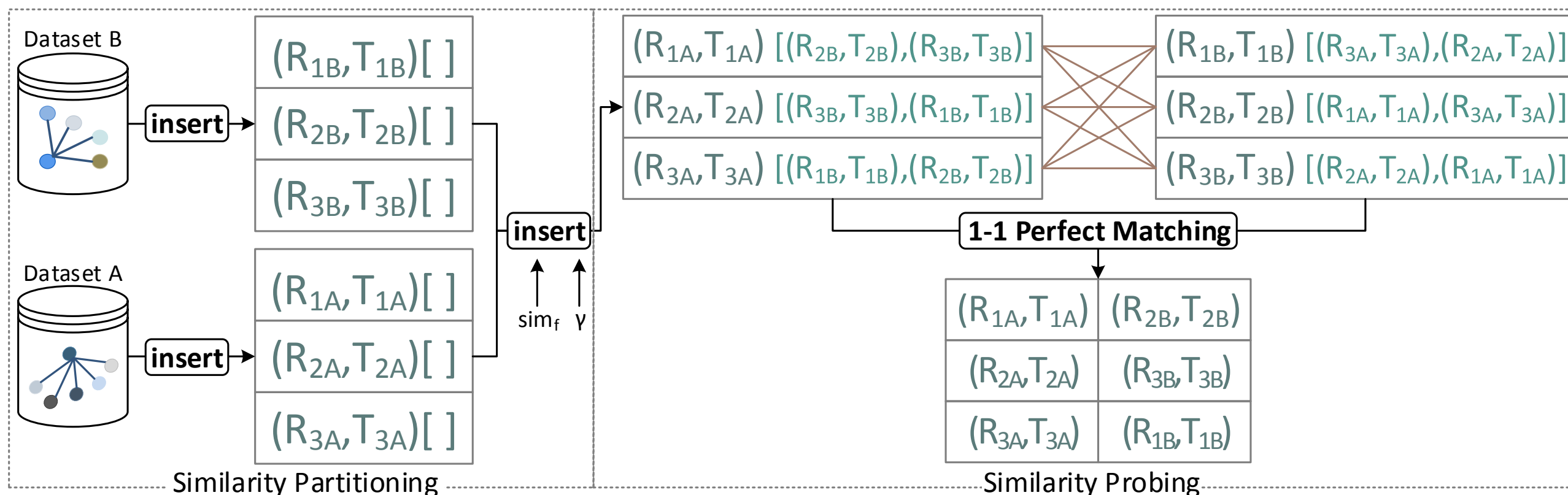
$$\text{Tail} = [(\text{prop}, \text{val}), (\text{prop}, \text{val}), (\text{prop}, \text{val})]$$

$$\text{ExampleRSM} = (\text{DB00316}, T) [(\text{dbr:Paracetamol}, T), (\text{dbr:Acetaminophen}, T)]$$



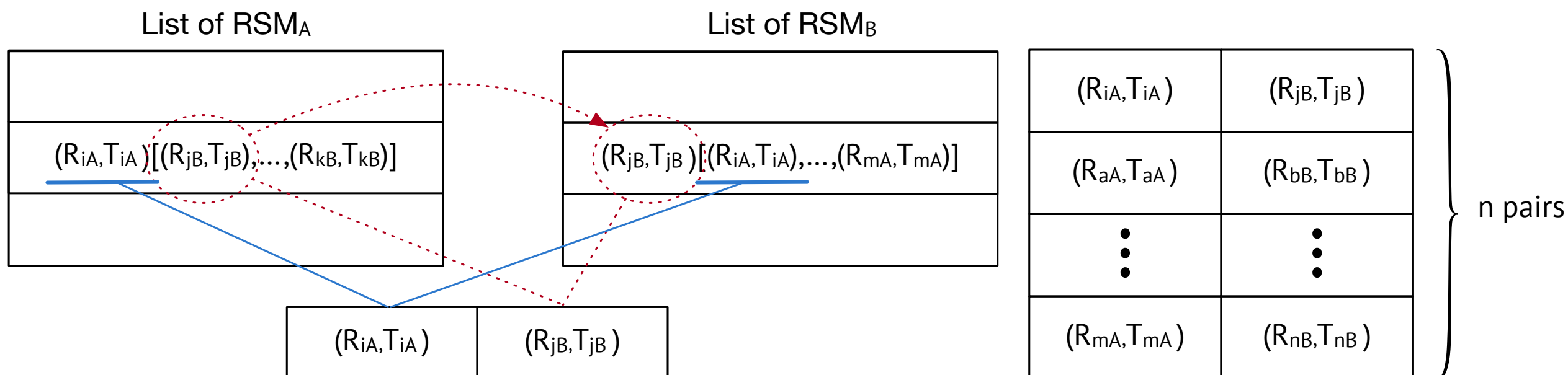
BLOCKING SJOIN

1. Load: Extract all RDF molecules from graphs
2. Insert:
 1. Compute a similarity matrix among molecules with a given threshold
 2. Sort most similar molecules for each RSM
3. Probe: Compute 1-1 perfect matching. Blocking!



1-1 PERFECT MATCHING

If RSM_{iA} has the most similar molecule M_{jB} , and RSM_{jB} has the most similar molecule M_{iA} , then yield a result (M_{iA}, M_{jB})

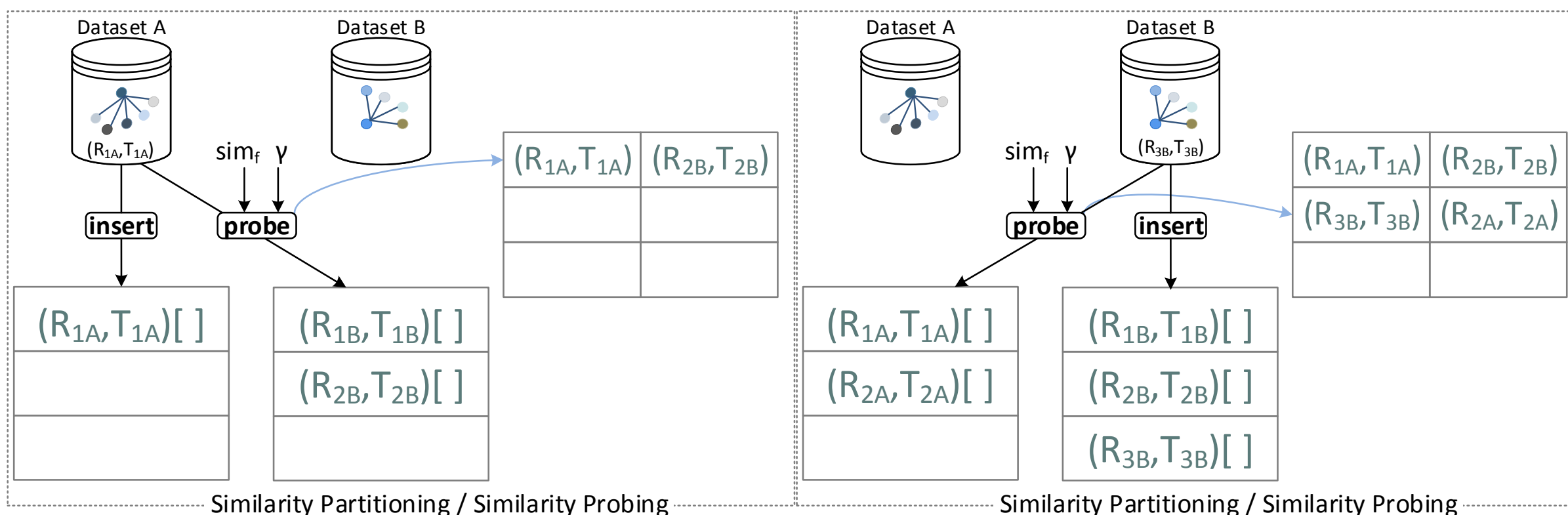


(a) 1-1 matching from the bipartite graph of RMS

(b) Matched pairs

NON-BLOCKING SJOIN

1. Load: RDF molecules arrive one by one
2. Probe:
 1. Compute a similarity score between M_{iA} and arrived molecules M_B
 2. Yield a result if $\text{sim}(M_{iA}, M_{jB}) \geq \gamma$
3. Insert: Put M_{iA} in the list of arrived molecules from the graph A

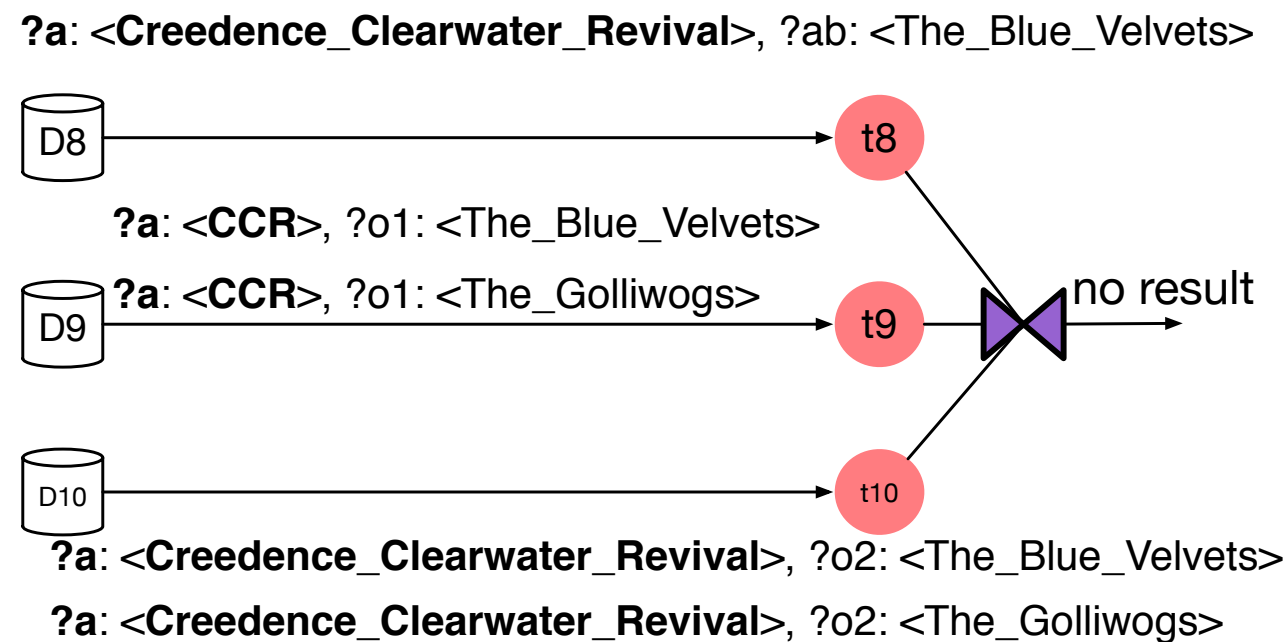


TIME COMPLEXITY ANALYSIS

- ▶ Partitioning: similarity matrix computation
- ▶ Sorting: QuickSort
- ▶ Matching: SJoin or Hungarian algorithms

Stage	Blocking SJoin Complexity	Non-Blocking SJoin Complexity
Partitioning	$O(n^2 \cdot O(Sim_f))$	$O(n^2 \cdot O(Sim_f))$
Sorting	$O(n \log n)$	
Matching	$O(n^3)$	
Overall	$O(n^2 \cdot O(Sim_f)) + O(n^3)$	$O(n^2 \cdot O(Sim_f))$

MULTI-WAY SIMILARITY JOIN OPERATOR



- ▶ Input: RDF molecules
- ▶ CCR and Creedence_Clearwater_Revival refer to the same entity
- ▶ Traditional ops - no join

MULTI-WAY SIMILARITY JOIN OPERATOR

?a: <Creedence_Clearwater_Revival>, ?ab: <The_Blue_Velvets>



?a: <CCR>, ?o1: <The_Blue_Velvets>

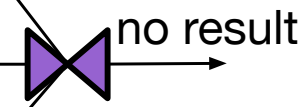


?a: <CCR>, ?o1: <The_Golliwogs>



?a: <Creedence_Clearwater_Revival>, ?o2: <The_Blue_Velvets>

?a: <Creedence_Clearwater_Revival>, ?o2: <The_Golliwogs>



no result

- ▶ Input: RDF molecules
- ▶ CCR and Creedence_Clearwater_Revival refer to the same entity
- ▶ Traditional ops - no join

?a: <Creedence_Clearwater_Revival>, ?ab: <The_Blue_Velvets>



?a: <CCR>, ?o1: <The_Blue_Velvets>



?a: <CCR>, ?o1: <The_Golliwogs>



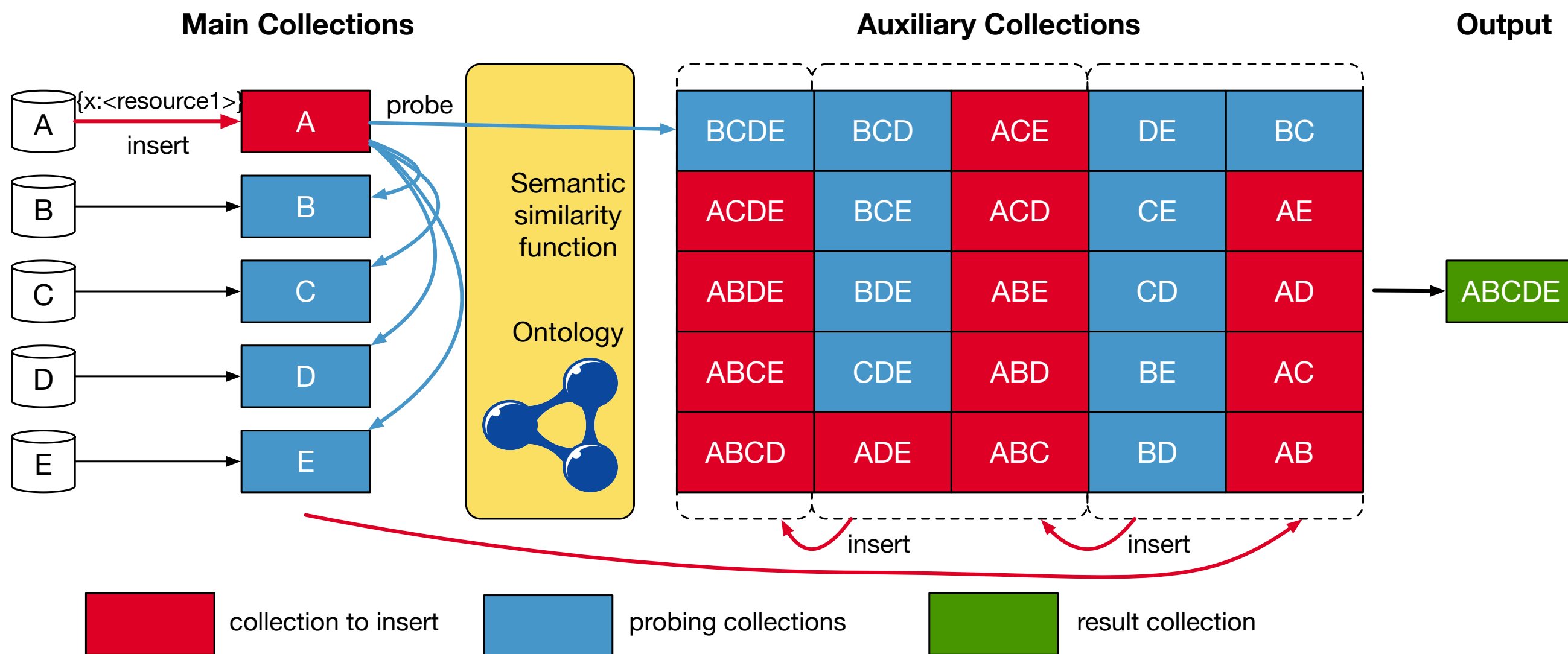
?a: <Creedence_Clearwater_Revival>, ?o2: <The_Blue_Velvets>

?a: <Creedence_Clearwater_Revival>, ?o2: <The_Golliwogs>

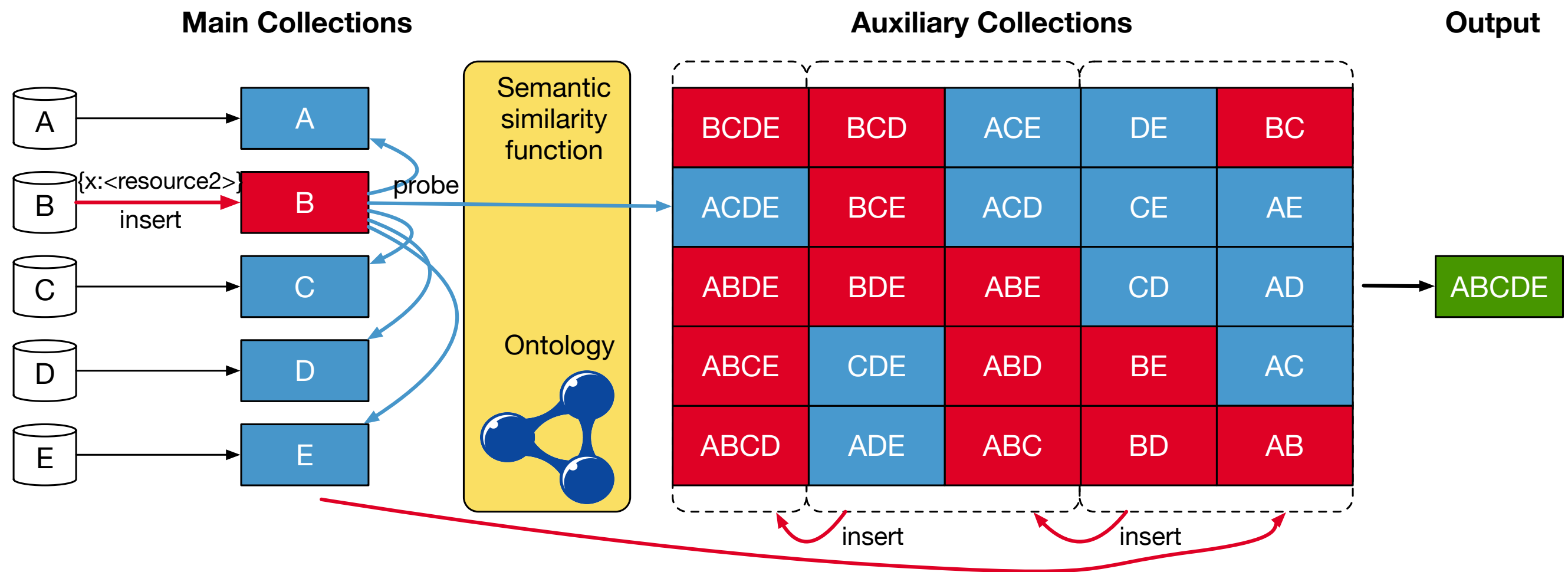


```
{?a:<CCR>, ?ab:<The_Blue_Velvets>,
  ?o1:<The_Blue_Velvets>, ?o2: <The_Blue_Velvets> }
{?a:<CCR>, ?ab:<The_Blue_Velvets>,
  ?o1:<The_Golliwogs>, ?o2: <The_Blue_Velvets>}
{?a:<CCR>, ?ab:<The_Blue_Velvets>,
  ?o1:<The_Blue_Velvets>, ?o2: <The_Golliwogs>}
{?a:<CCR>, ?ab:<The_Blue_Velvets>,
  ?o1:<The_Golliwogs>, ?o2: <The_Golliwogs>}
```

MULTI-WAY SIMILARITY JOIN OPERATOR (MSIMJOIN)



MULTI-WAY SIMILARITY JOIN OPERATOR (MSIMJOIN)



QUERY EXECUTION

EVALUATION

- ▶ Experiment 1: 500 DBpedia molecules of type Person, with randomly deleted/edited triples
- ▶ Experiment 2: 500 and 1000 molecules from DBpedia and Wikidata of type Person

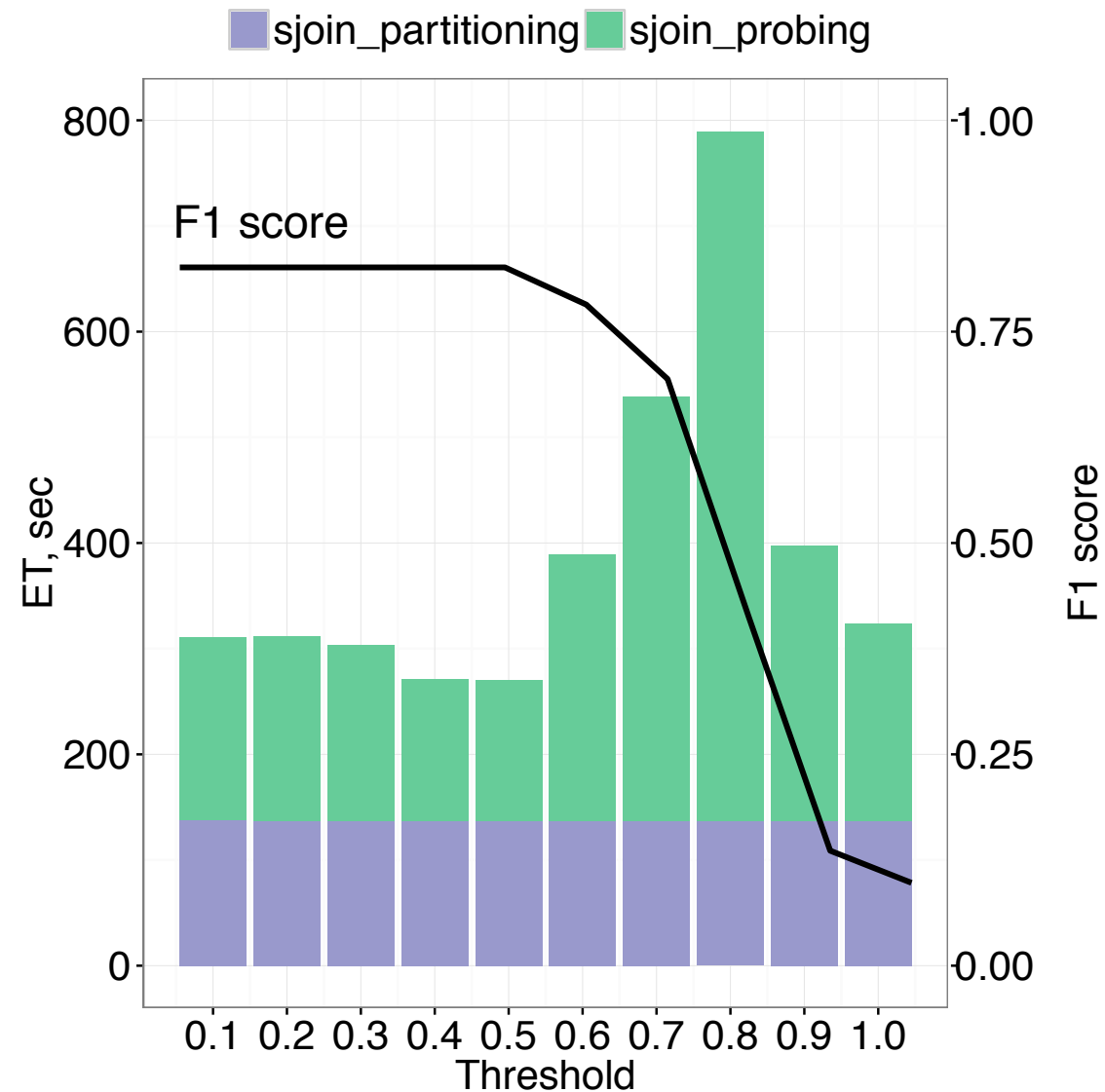
	Experiment 1: People		Experiment 2: People			
	DBpedia D1	DBpedia D2	DBpedia	Wikidata	DBpedia	Wikidata
<i>Molecules</i>	500	500	500	500	1000	1000
<i>Triples</i>	17,951	17,894	29,263	16,307	54,590	29,138

EVALUATION

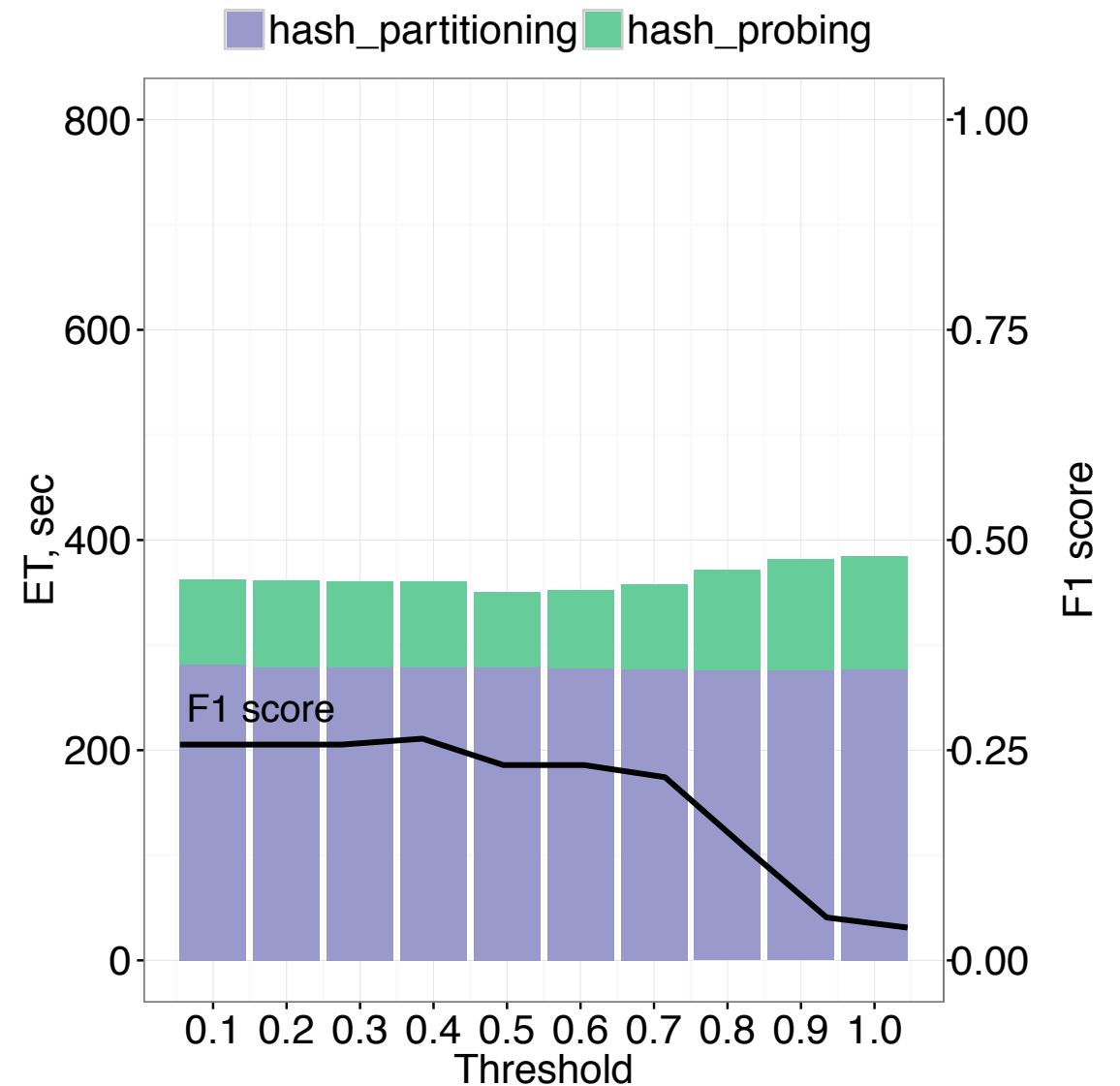
- ▶ Experiment 1: 500 DBpedia molecules of type Person, with randomly deleted/edited triples
- ▶ Experiment 2: 500 and 1000 molecules from DBpedia and Wikidata of type Person
- ▶ Blocking operator metrics:
 - Partitioning time
 - Probing time
 - F1 score
- ▶ Non-blocking operator metric:
 - Completeness over time
- ▶ Variables:
 - Join operator
 - Similarity function (GADES, Jaccard)
 - Threshold value
 - # Molecules

	Experiment 1: People		Experiment 2: People			
	DBpedia D1	DBpedia D2	DBpedia	Wikidata	DBpedia	Wikidata
<i>Molecules</i>	500	500	500	500	1000	1000
<i>Triples</i>	17,951	17,894	29,263	16,307	54,590	29,138

EXPERIMENT 1 - BLOCKING - GADES

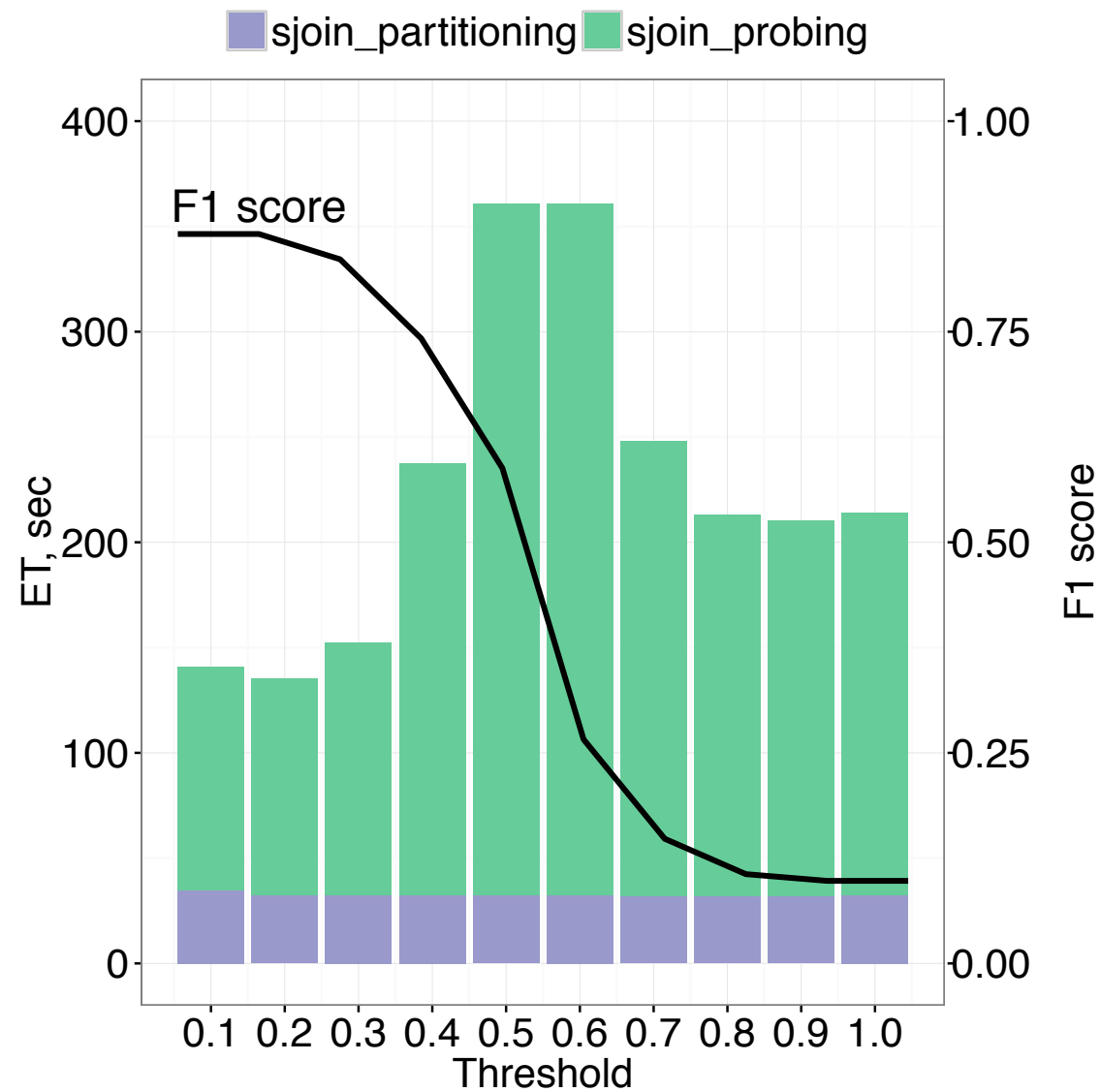


(a) SJoin performance

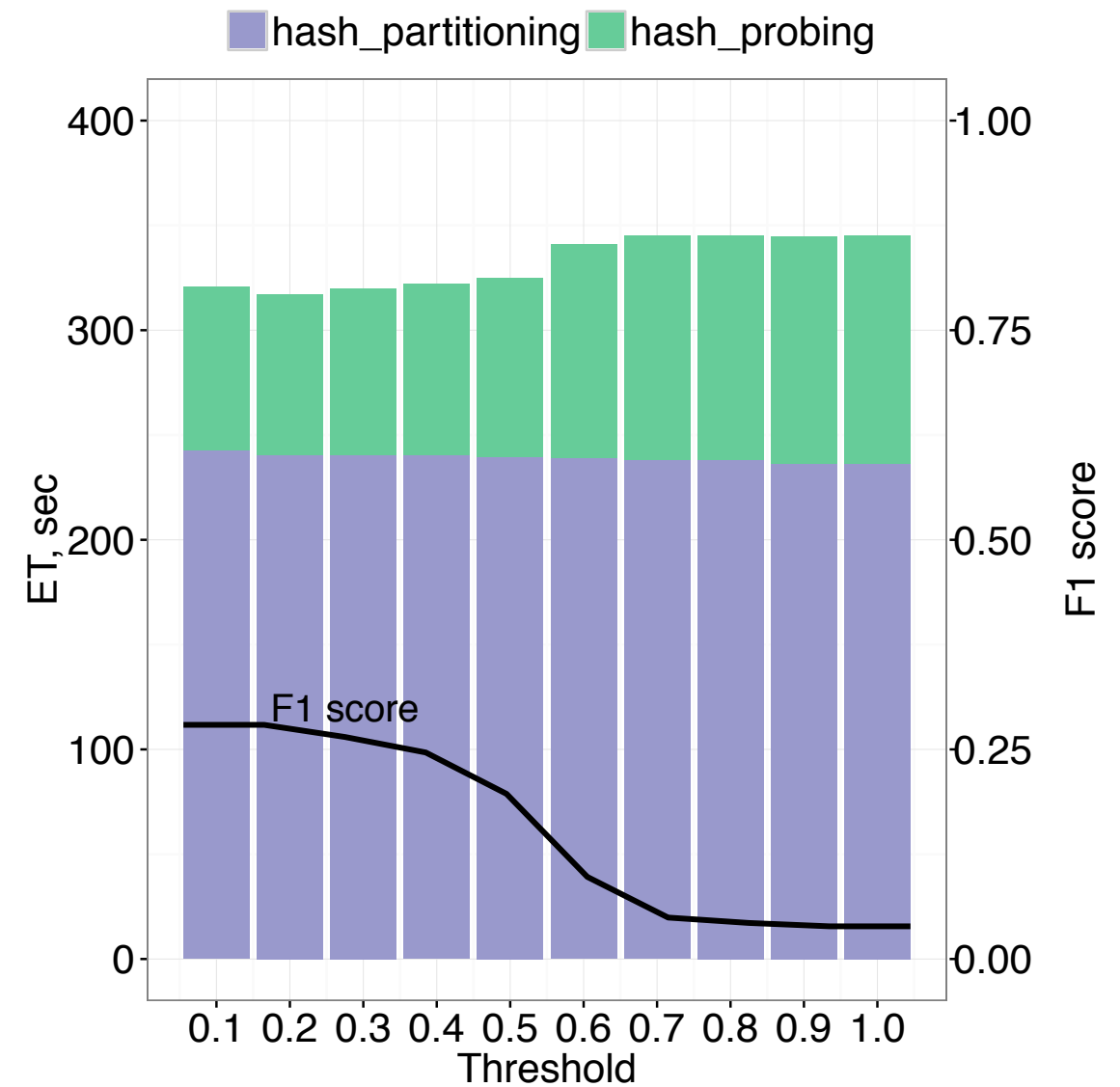


(b) Hash Join performance

EXPERIMENT 1 - BLOCKING - JACCARD



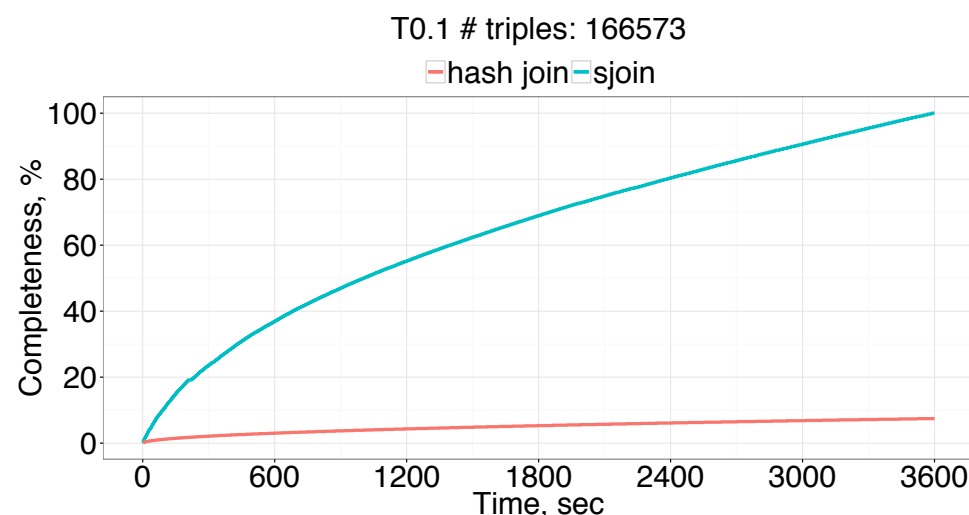
(a) SJoin performance



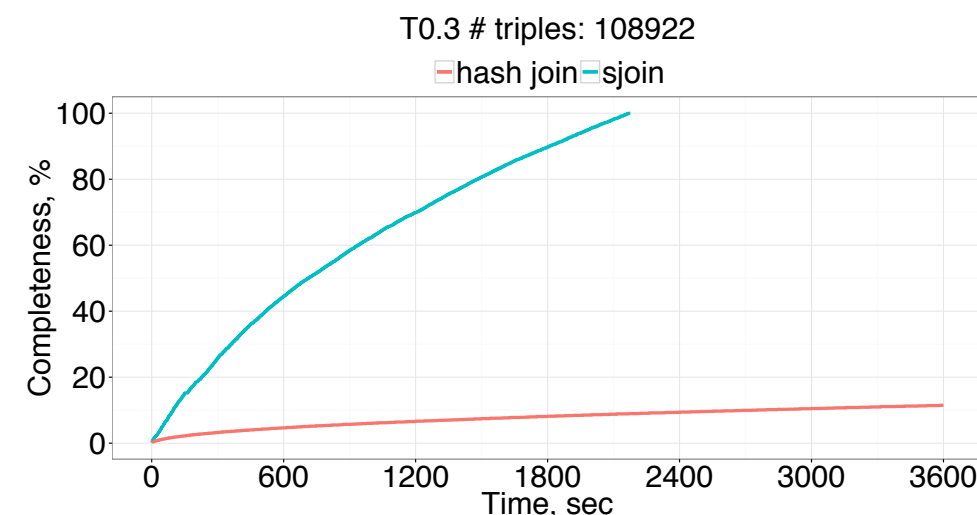
(b) Hash Join performance

EXPERIMENT 1 - NON-BLOCKING - GADES

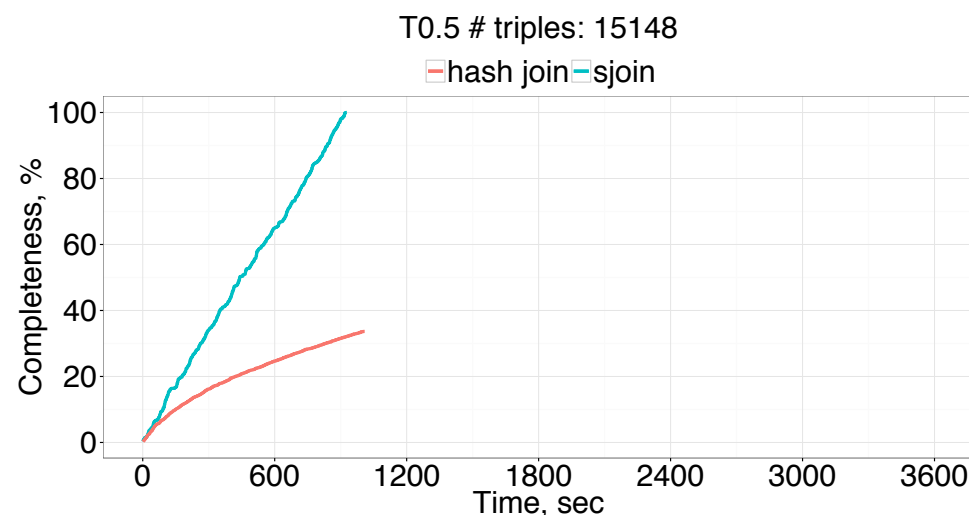
With GADES SJoin is able to complete integration in all threshold values.



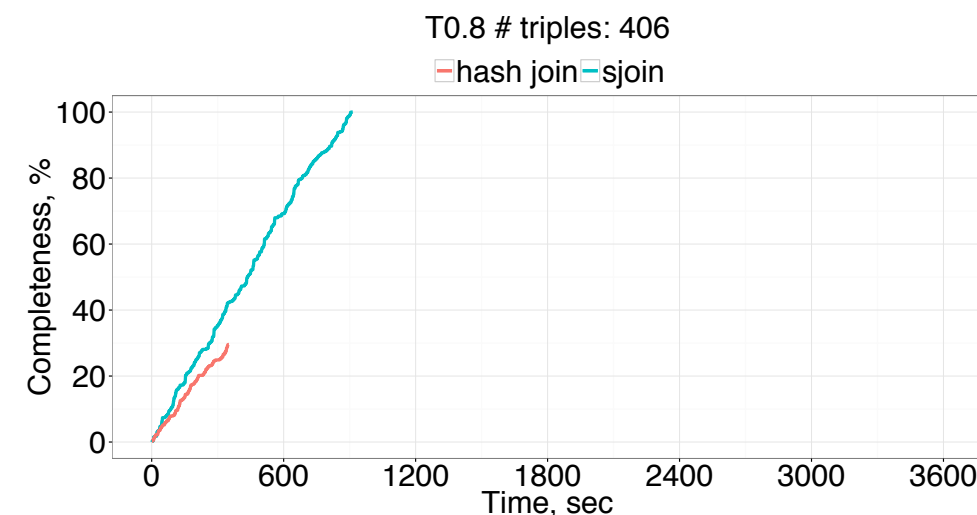
(a) $T = 0.1$



(b) $T = 0.3$



(c) $T=0.5$

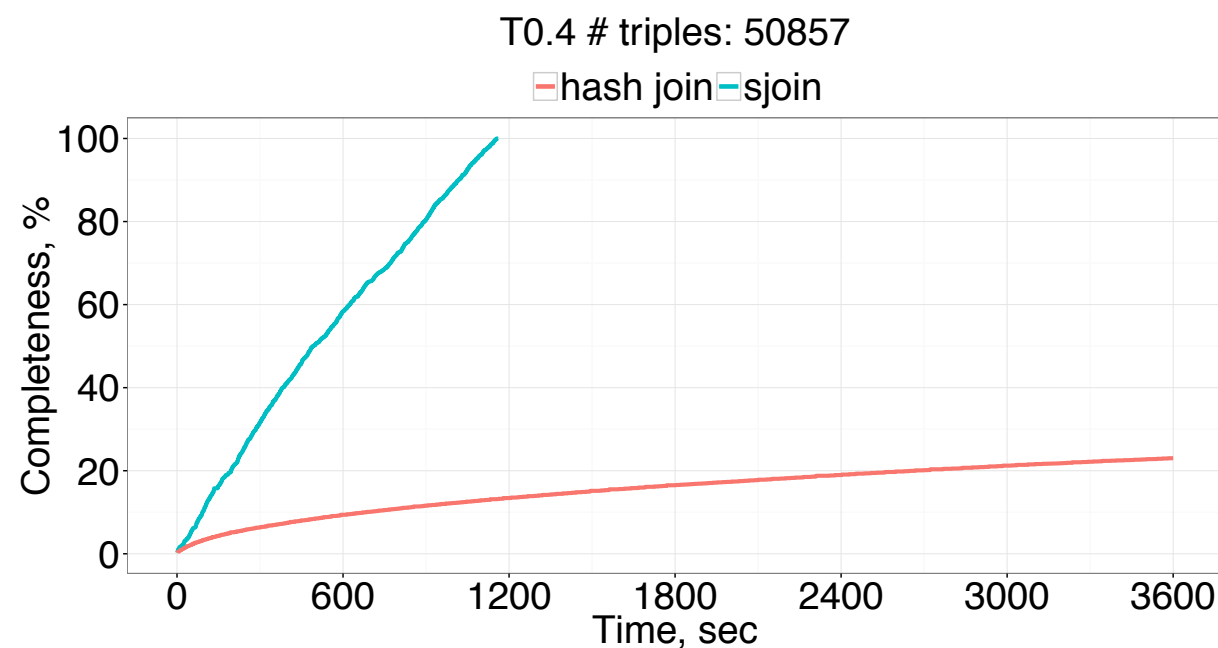


(d) $T=0.8$

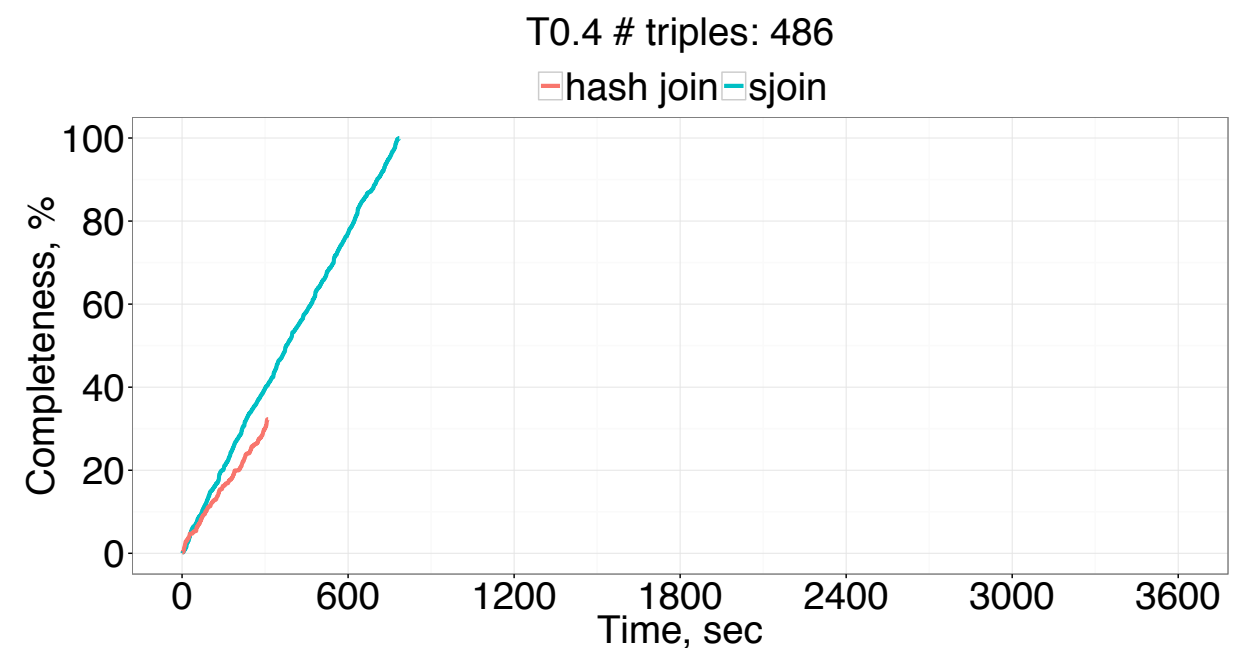
Threshold reduces the amount of compared molecules, hence join converges faster.

EXPERIMENT 1 - NON-BLOCKING - GADES VS JACCARD

- ▶ Fixed threshold: 0.4
- ▶ GADES identifies 100x more results than Jaccard
- ▶ SJoin completes in all cases



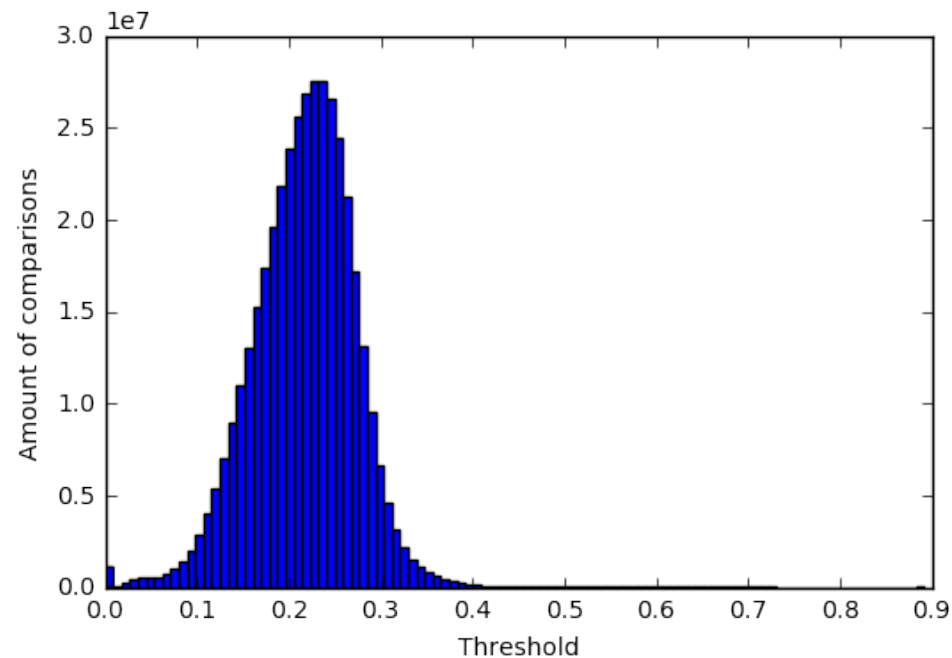
(a) $T = 0.4$, GADES



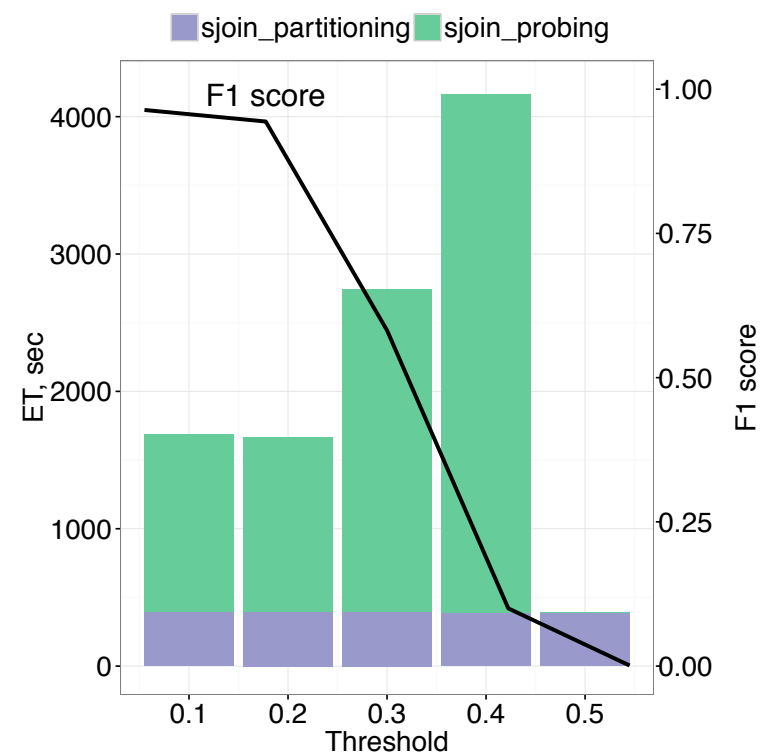
(b) $T = 0.4$, Jaccard

EXPERIMENT 2 - BLOCKING - GADES

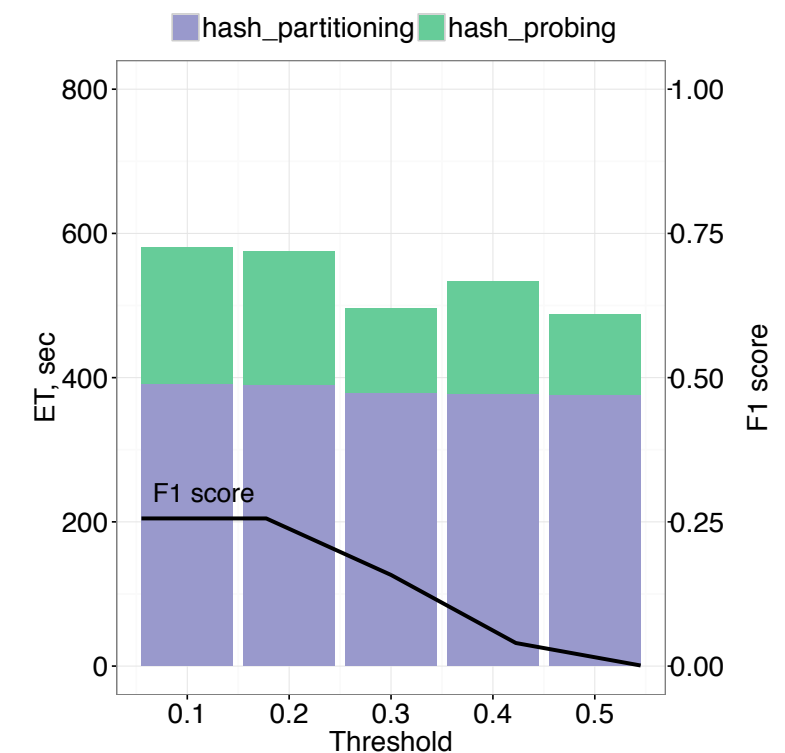
- ▶ GADES identifies similarities between DBpedia and Wikidata < 0.4
- ▶ SJoin + GADES maintains higher F1 score than Hash Join at the cost of execution time



(a) GADES distribution



(b) SJoin



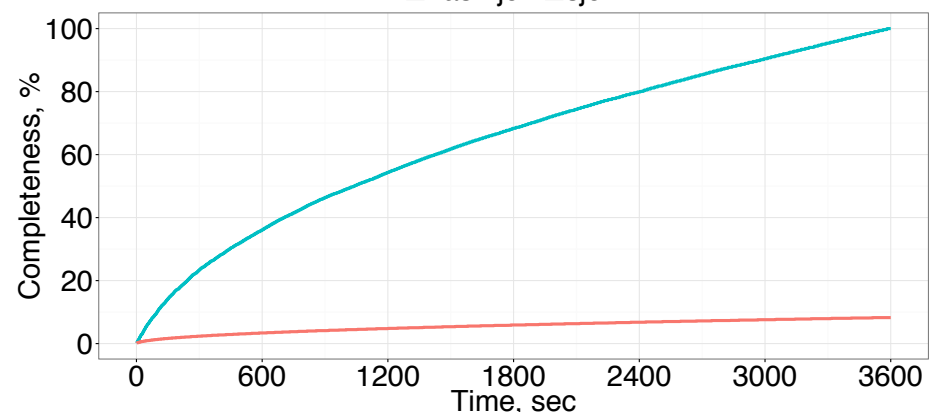
(c) Hash Join

EXPERIMENT 2 - NON-BLOCKING - GADES

500 molecules

T0.2 # triples: 153904

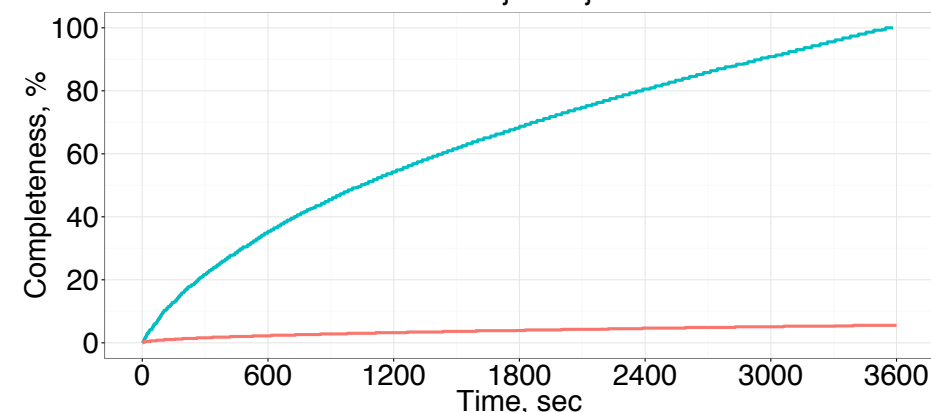
hash join sjoin



1000 molecules

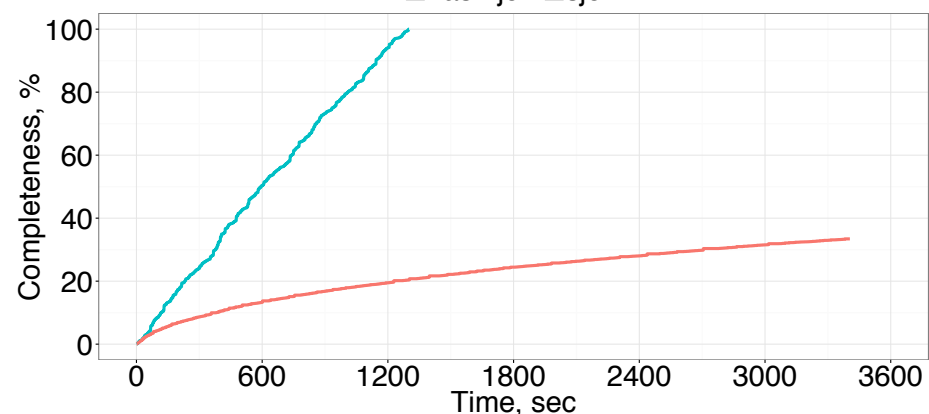
T0.2 # triples: 160062

hash join sjoin



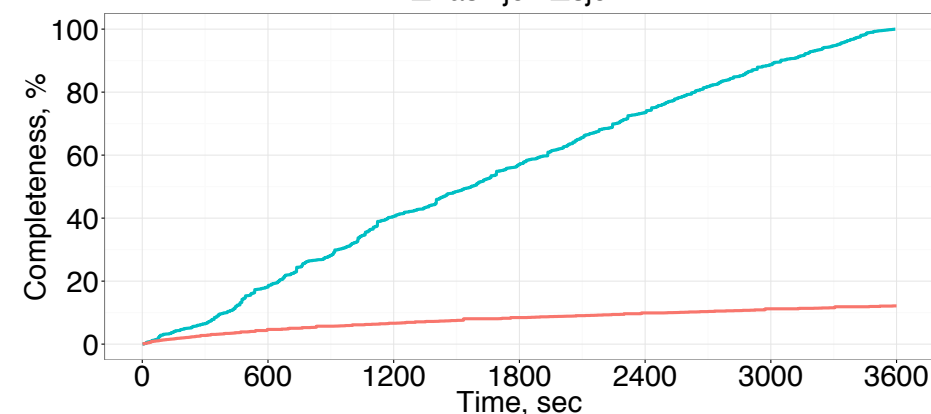
T0.3 # triples: 32705

hash join sjoin



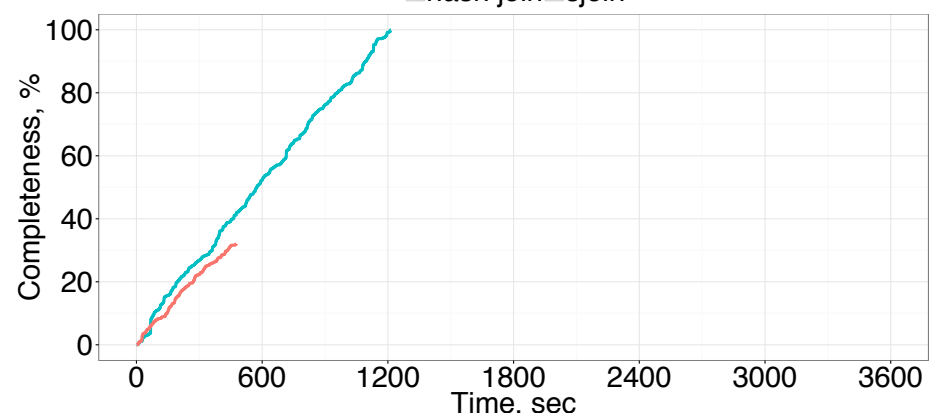
T0.3 # triples: 67169

hash join sjoin



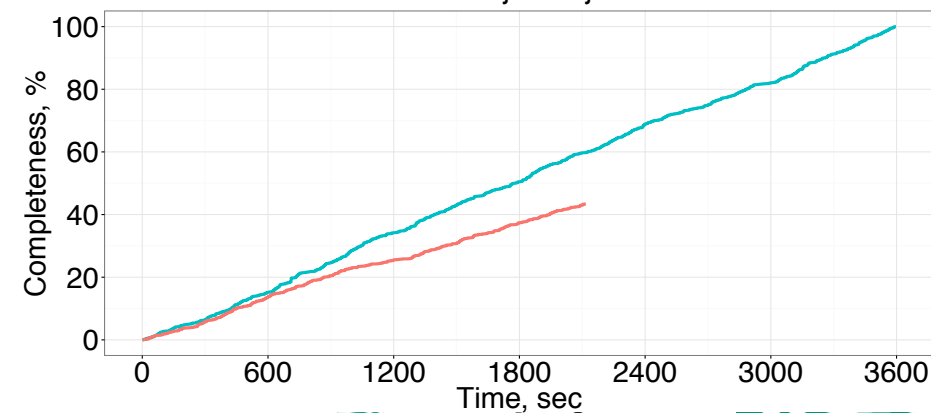
T0.4 # triples: 639

hash join sjoin



T0.4 # triples: 3466

hash join sjoin



OVERVIEW

- ▶ **MULDER** - a federated query engine for heterogeneous data processing
- ▶ **SMJoin** - a multi-way join operator for star-shaped subqueries
- ▶ **SJoin** - a semantic join operator
- ▶ **MSimJoin** - a multi-way semantic join operator

THANK YOU!

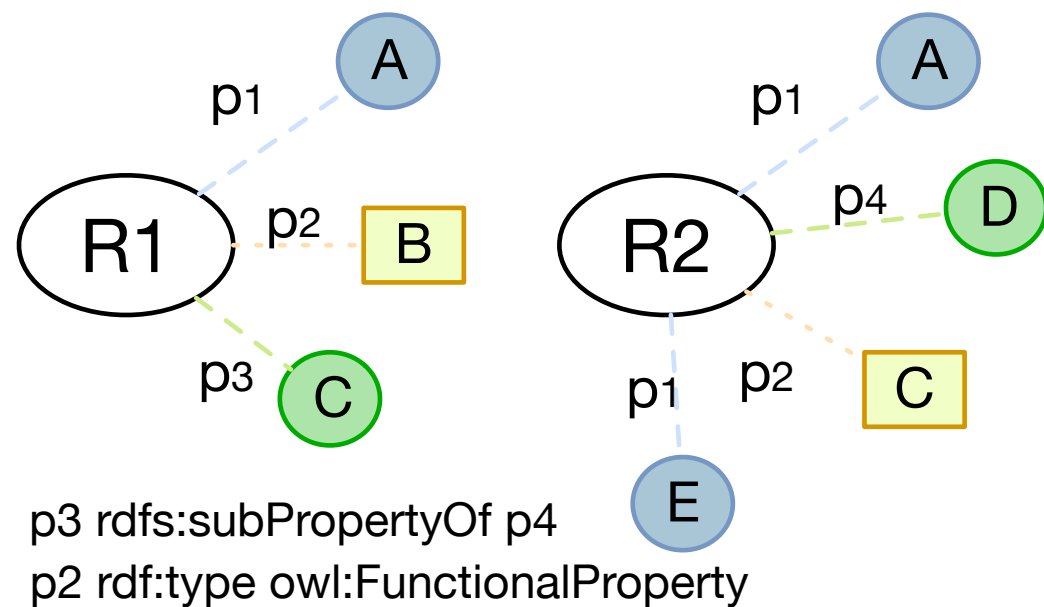
galkin@cs.uni-bonn.de

mikhail.galkin@iais.fraunhofer.de

SJOIN FUSION POLICIES

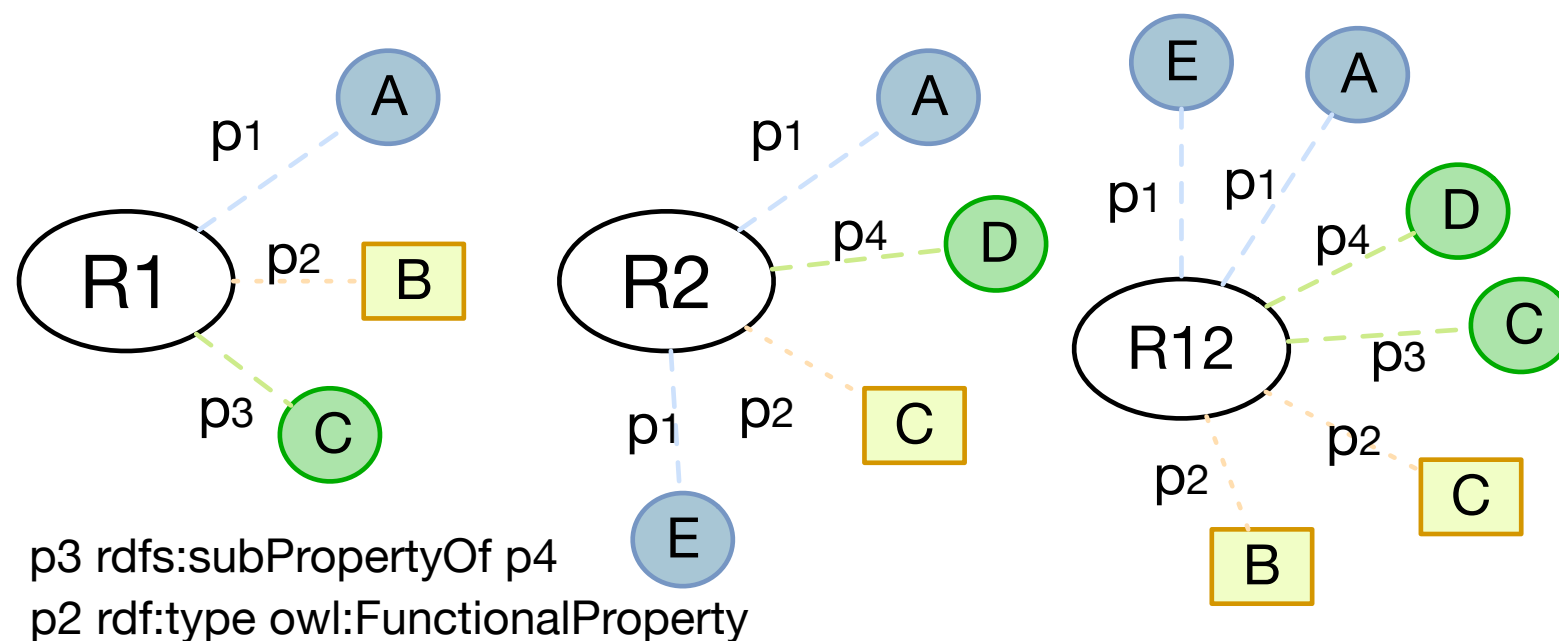
SJOIN FUSION POLICIES

1. Semantically equivalent molecules R1 and R2 with two axioms



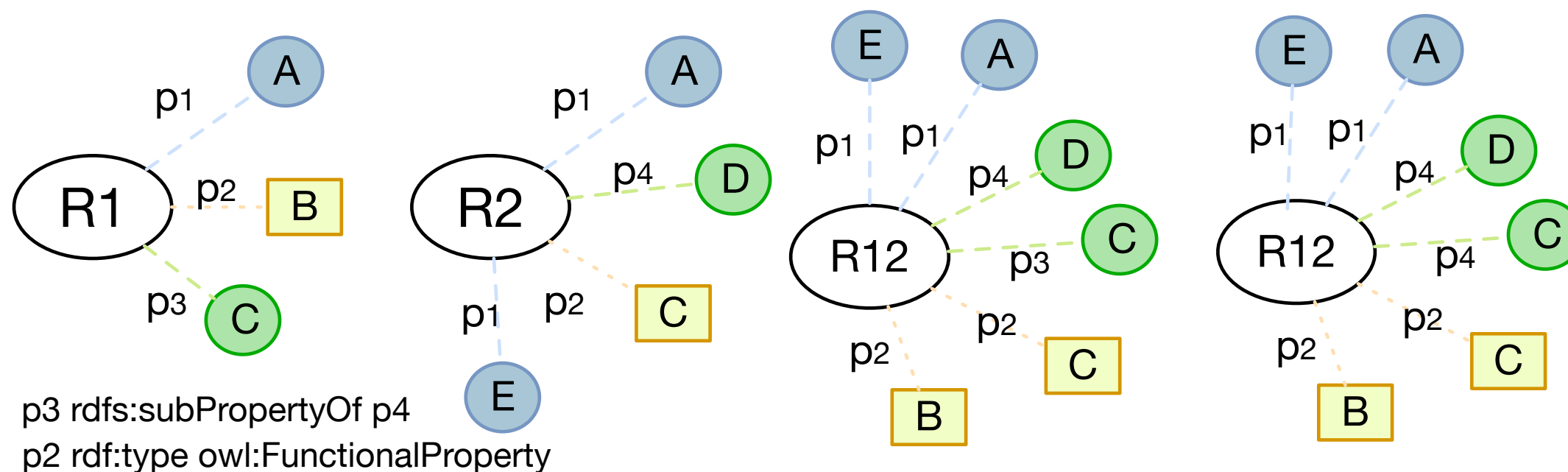
SJOIN FUSION POLICIES

1. Semantically equivalent molecules R1 and R2 with two axioms
2. Union policy: merge all



SJOIN FUSION POLICIES

1. Semantically equivalent molecules R1 and R2 with two axioms
2. Union policy: merge all
3. Subproperty policy: replace p3 with p4



SJOIN FUSION POLICIES

1. Semantically equivalent molecules R1 and R2 with two axioms
2. Union policy: merge all
3. Subproperty policy: replace p3 with p4
4. Authoritative policy: R1 is authoritative, p2 is functional, hence (p2,C) is discarded

