The LDBC Social Network Benchmark: Business Intelligence Workload

Gábor Szárnyas, Jack Waudby, Benjamin A. Steer, Dávid Szakállas, Altan Birler, Mingxi Wu, Yuchen Zhang, Peter Boncz

VLDB | 2023-08-30 | Vancouver

LDBC: Linked Data Benchmark Council

Non-profit company founded in 2012

The TPC for graph data management

Designs graph benchmarks

Governs the use of benchmarks

ldbcouncil.org

github.com/ldbc

LDBC members

3 sponsor companies

ANT GROUP

火山引擎
Beijing Volcano (ByteDance)

ORACLE
Labs

18 member companies including

aws

intel

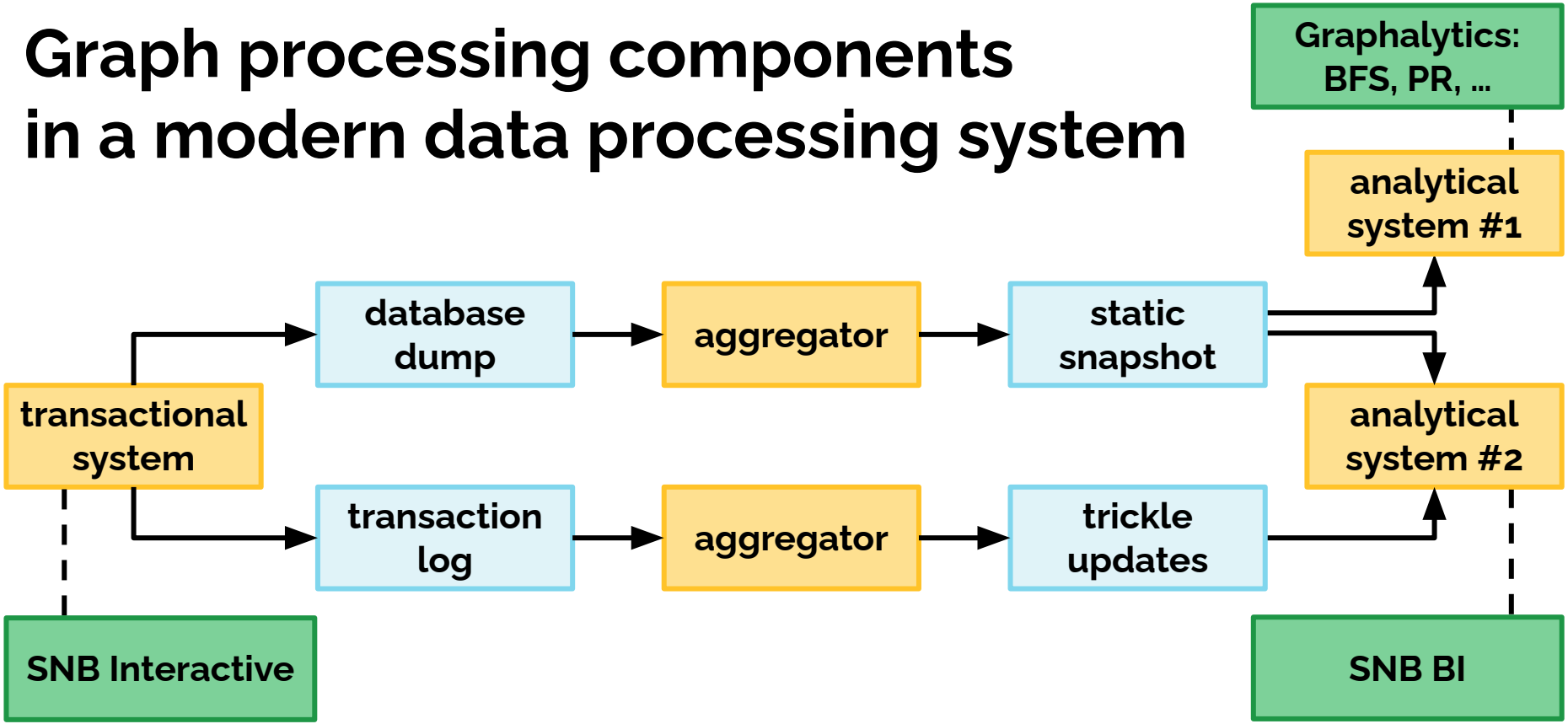
neo4j

TigerGraph

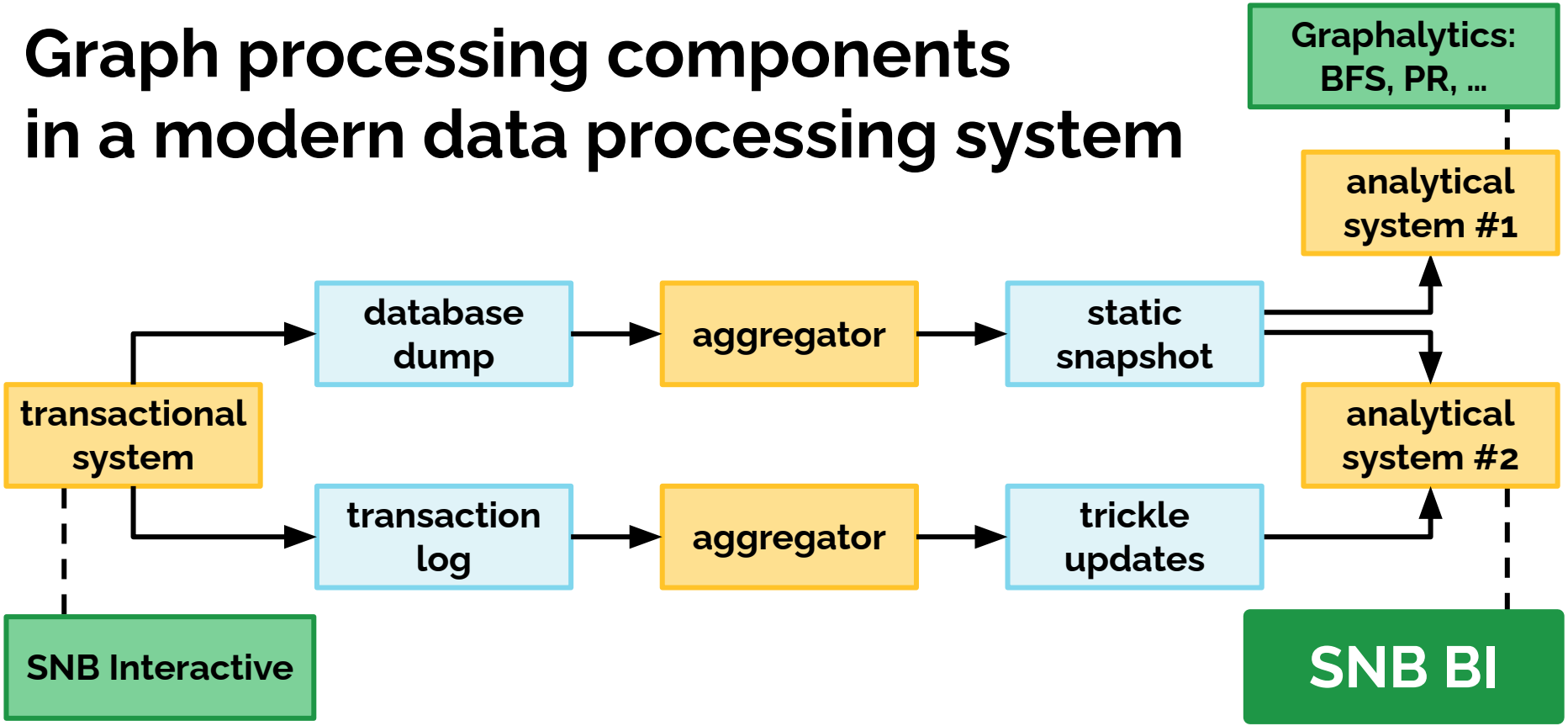
Total membership: 24 organizations and 65+ individuals

Graph processing components

Graph processing components in a modern data processing system



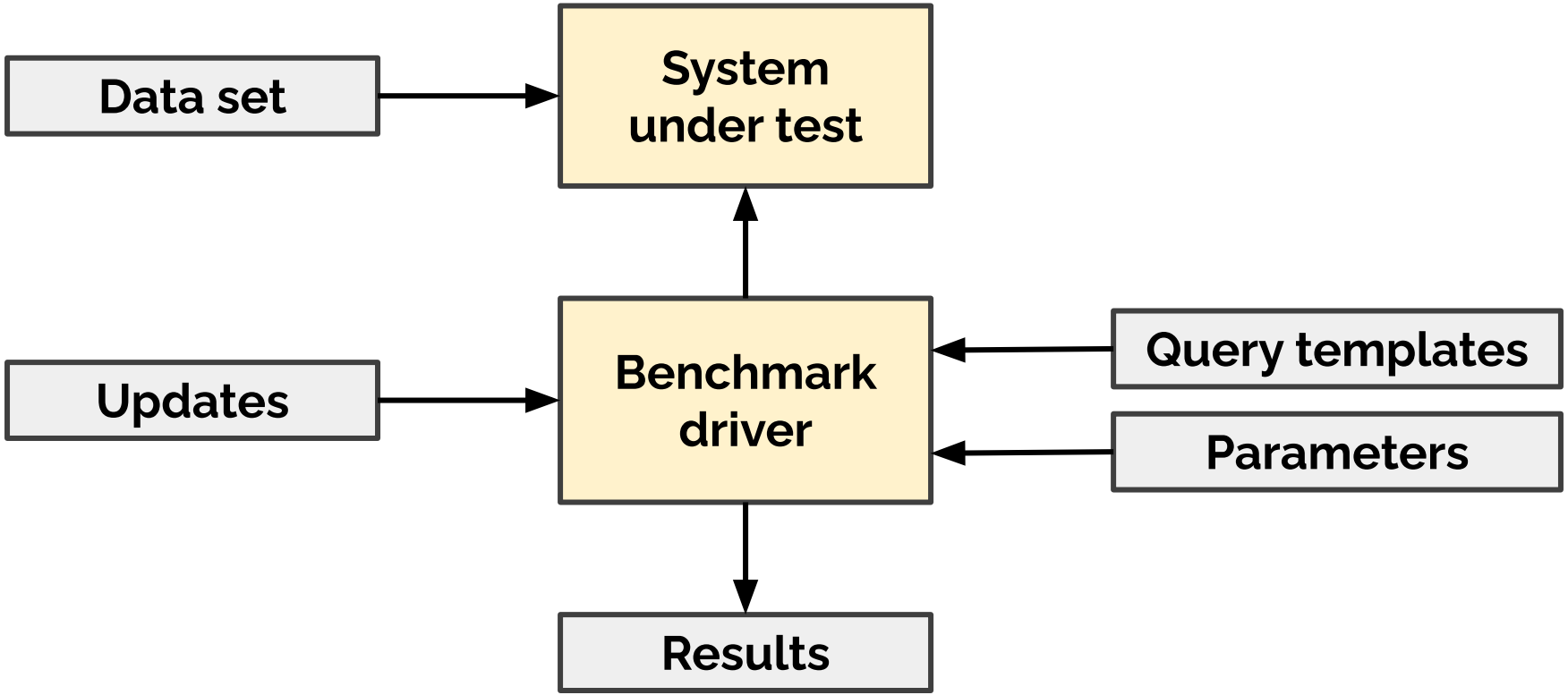
Graph processing components in a modern data processing system

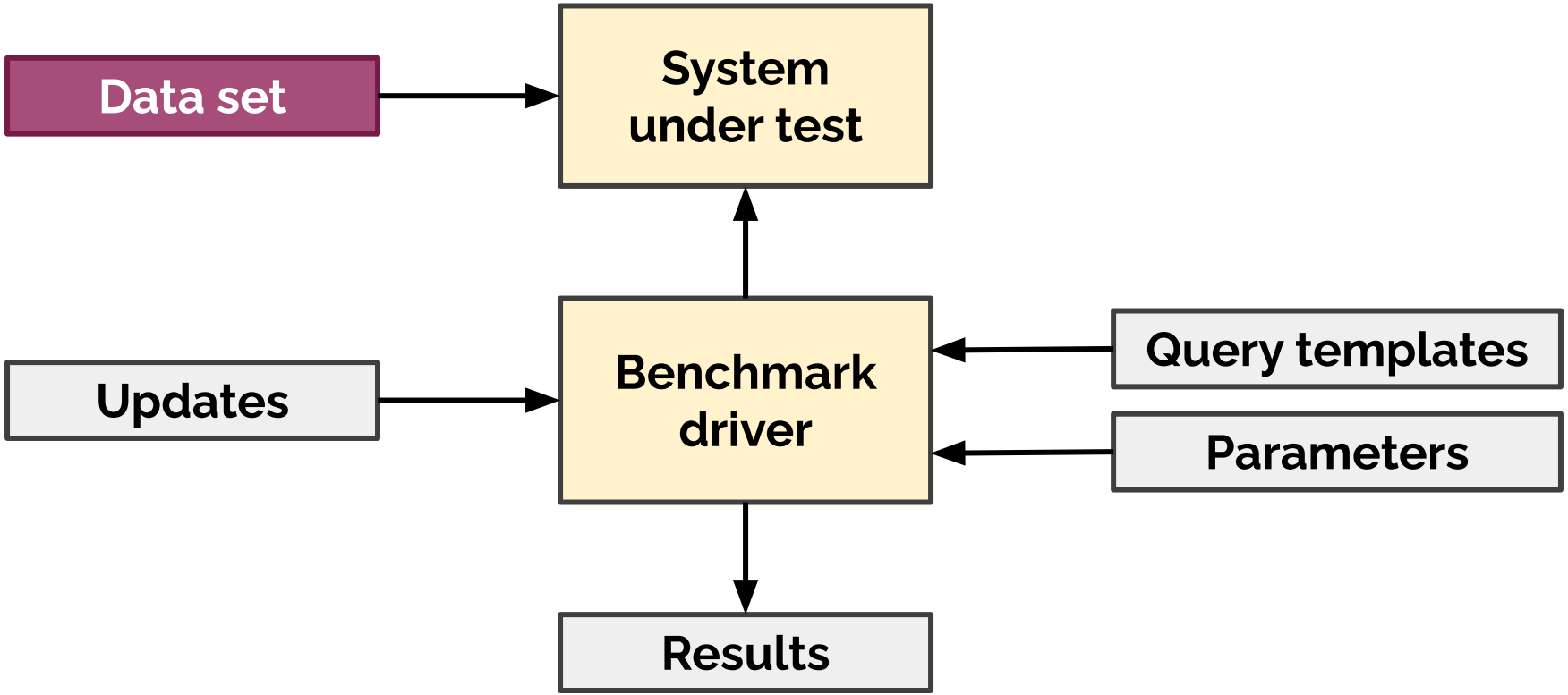


LDBC SNB Business Intelligence workload

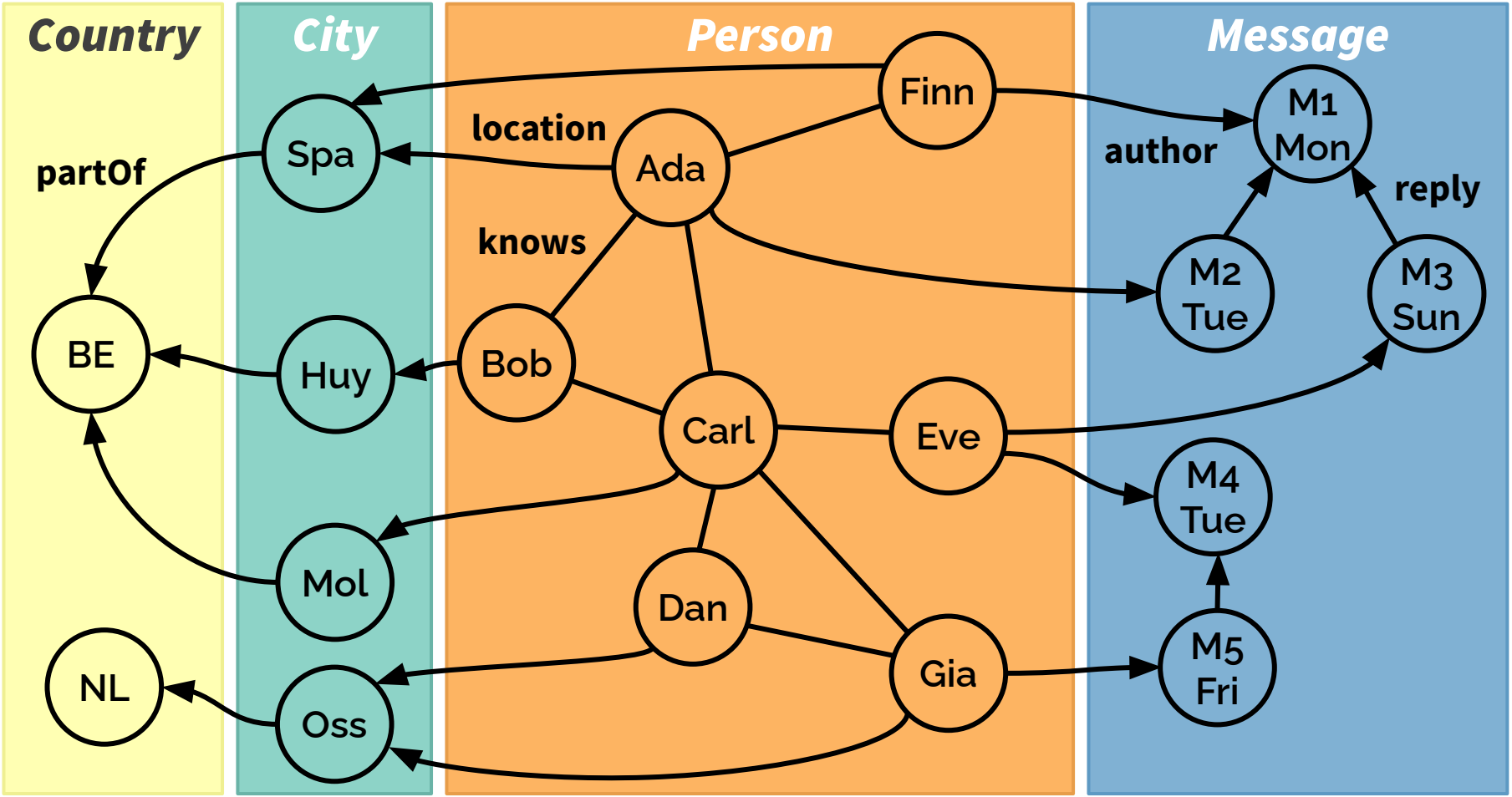
LDBC SNB Business Intelligence workload

An analytical data system benchmark that focuses on “graphy” features


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Example social network graph



Data set features

14 nodes types,
20 edges types

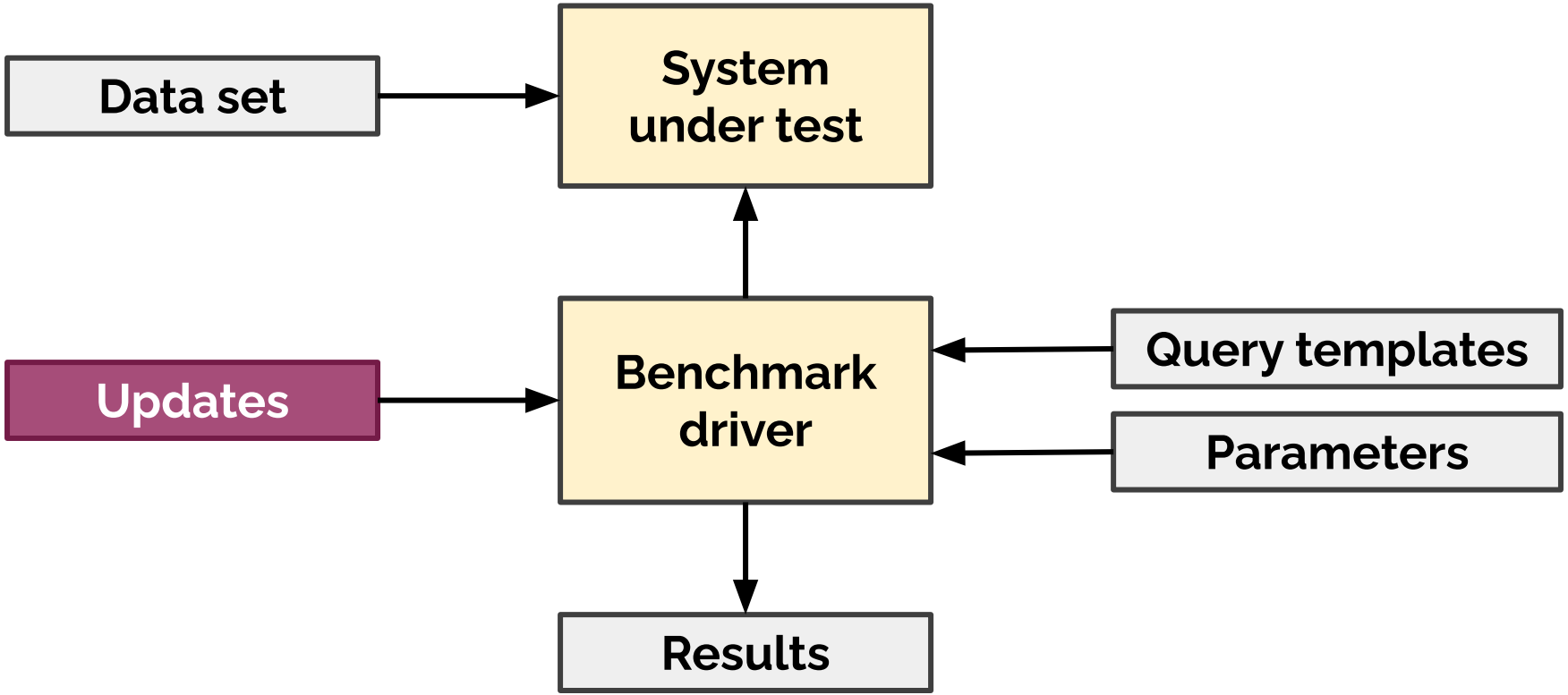
scale factor (SF):
CSV size in GiB

largest data set:
SF30,000

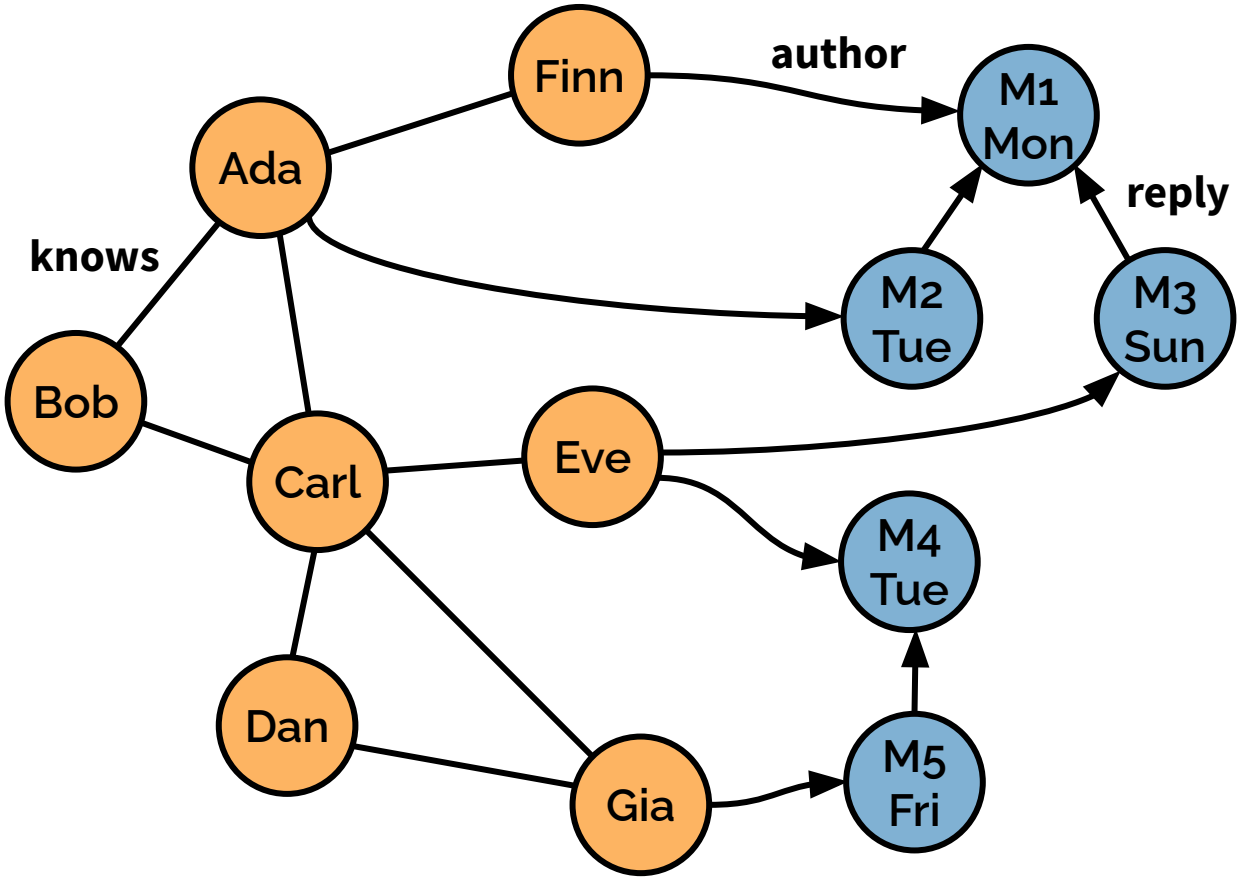
realistic degree
distributions

realistic
attribute names

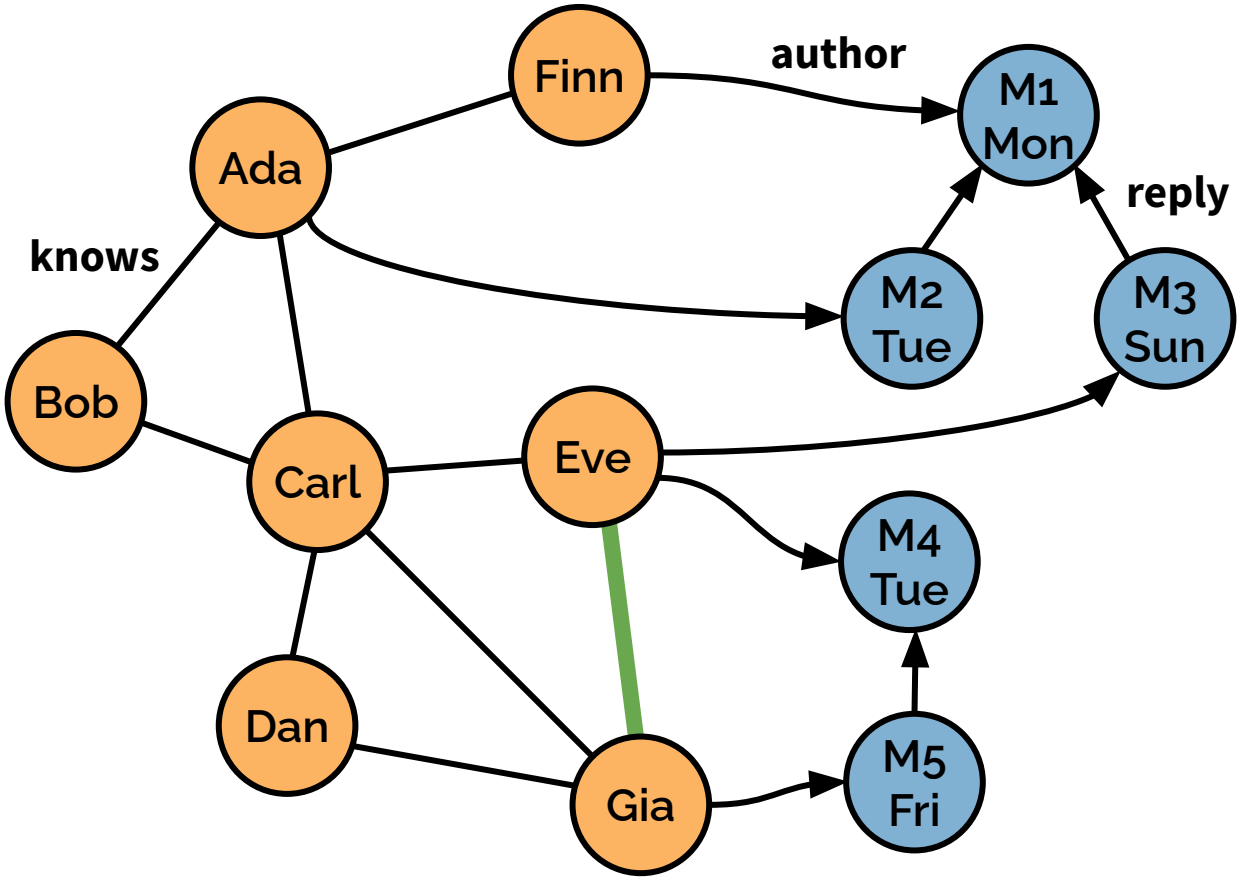
correlations along
interests and studies


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Updates


Finn
author
M1
Mon
Ada
reply
knows
M2
Tue
M3
Sun
Bob
Carl
Eve
M4
Tue
Dan
Gia
M5
Fri

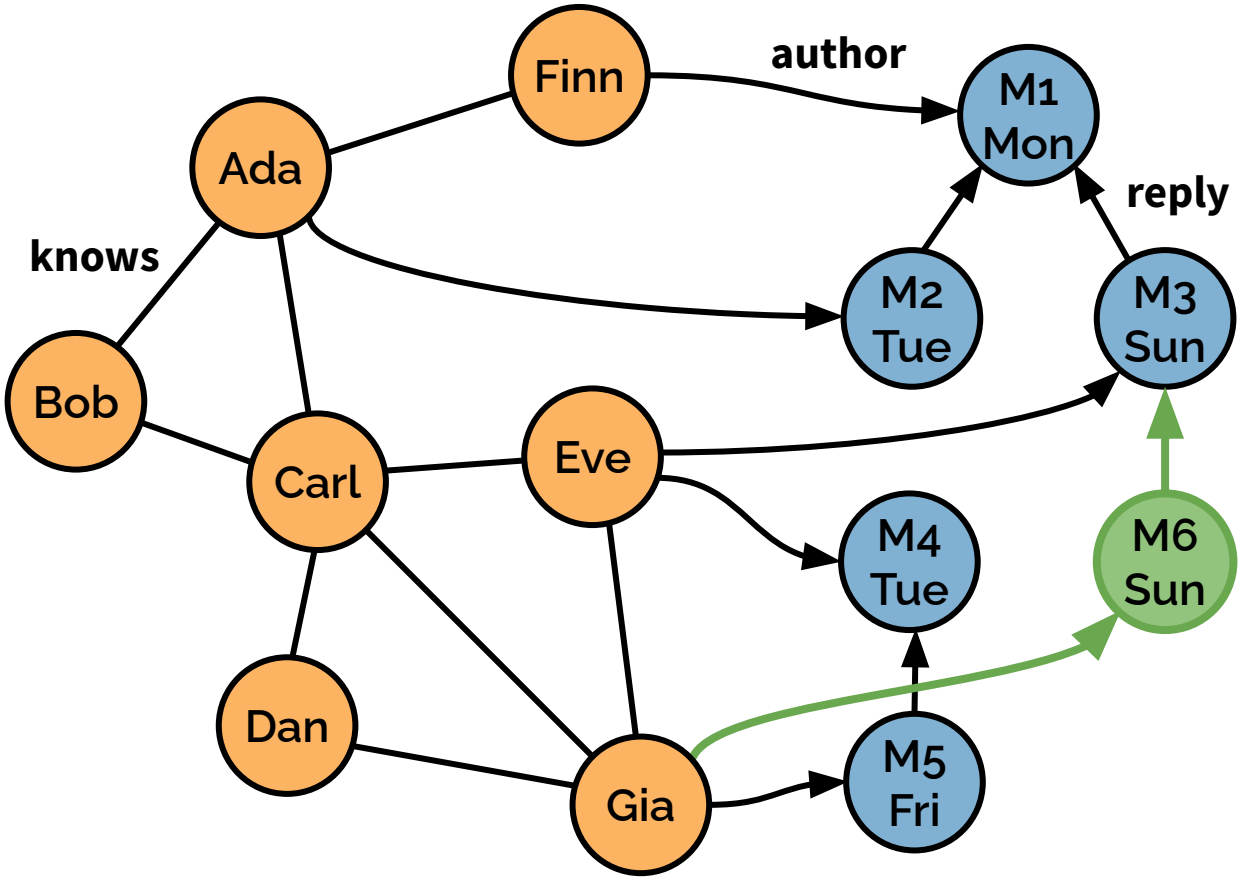
Update 1: Insert knows edge



Updates

+ knows("Eve", "Gia")

Update 2: Insert Message node

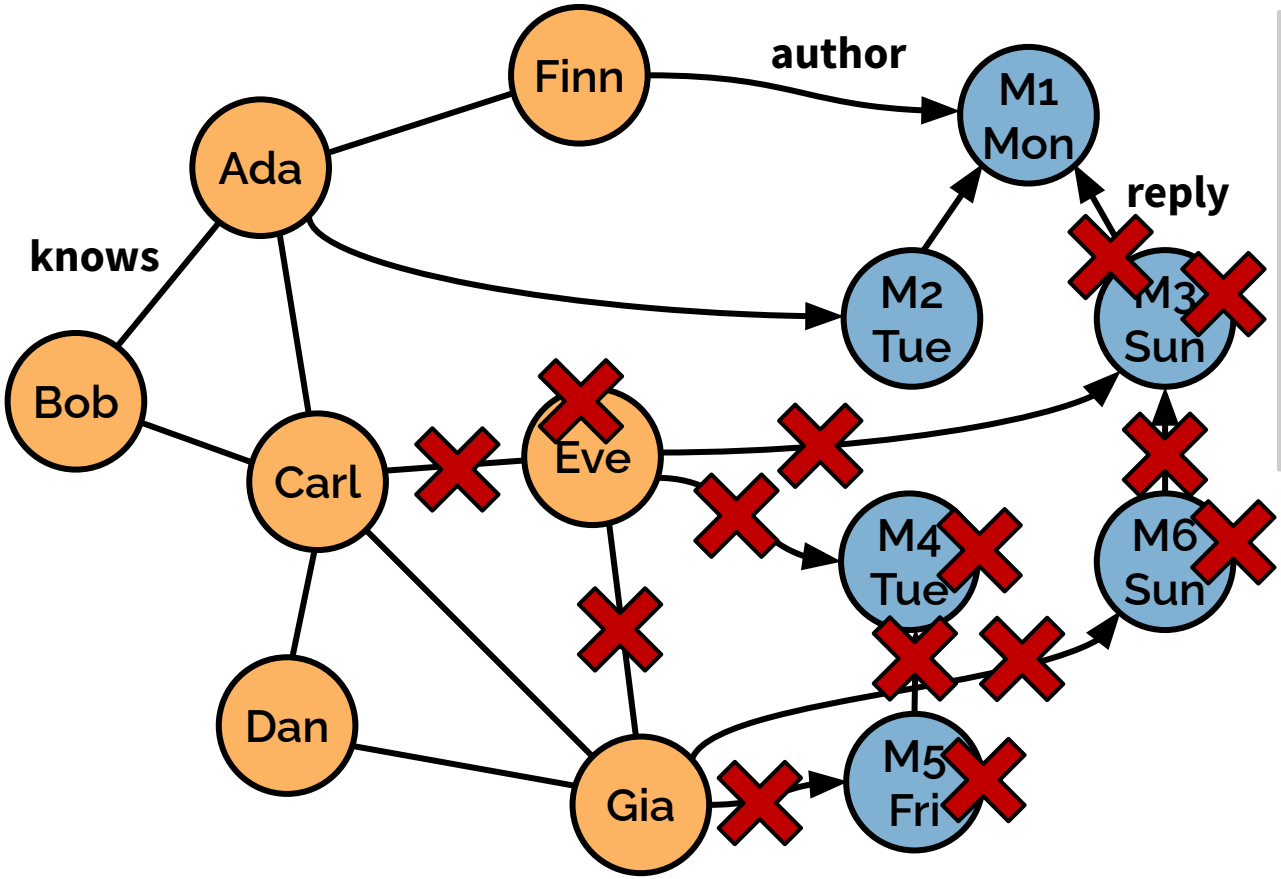


Updates

+ knows("Eve", "Gia")

+ Message("Gia", "M3")

Update 3: Delete Person node



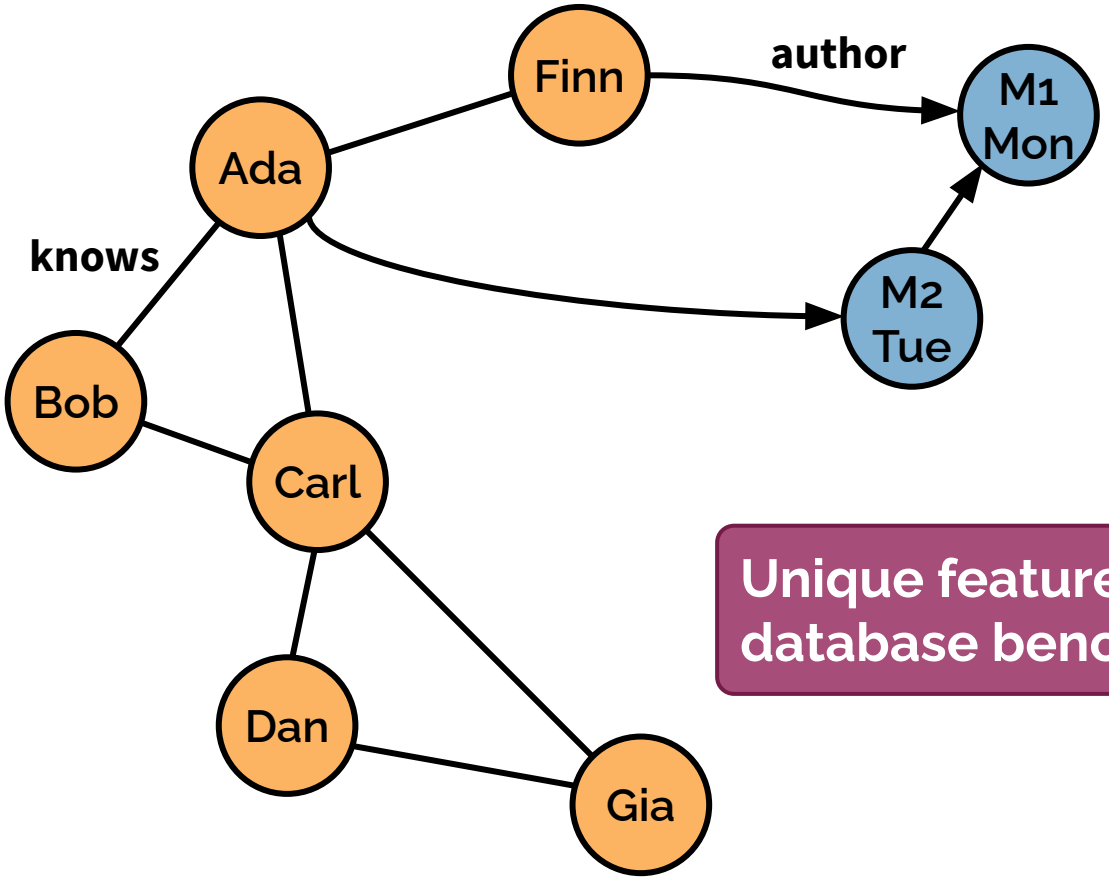
Updates

+ knows("Eve", "Gia")

+ Message("Gia", "M3")

– Person("Eve")

Update 3: Delete Person node



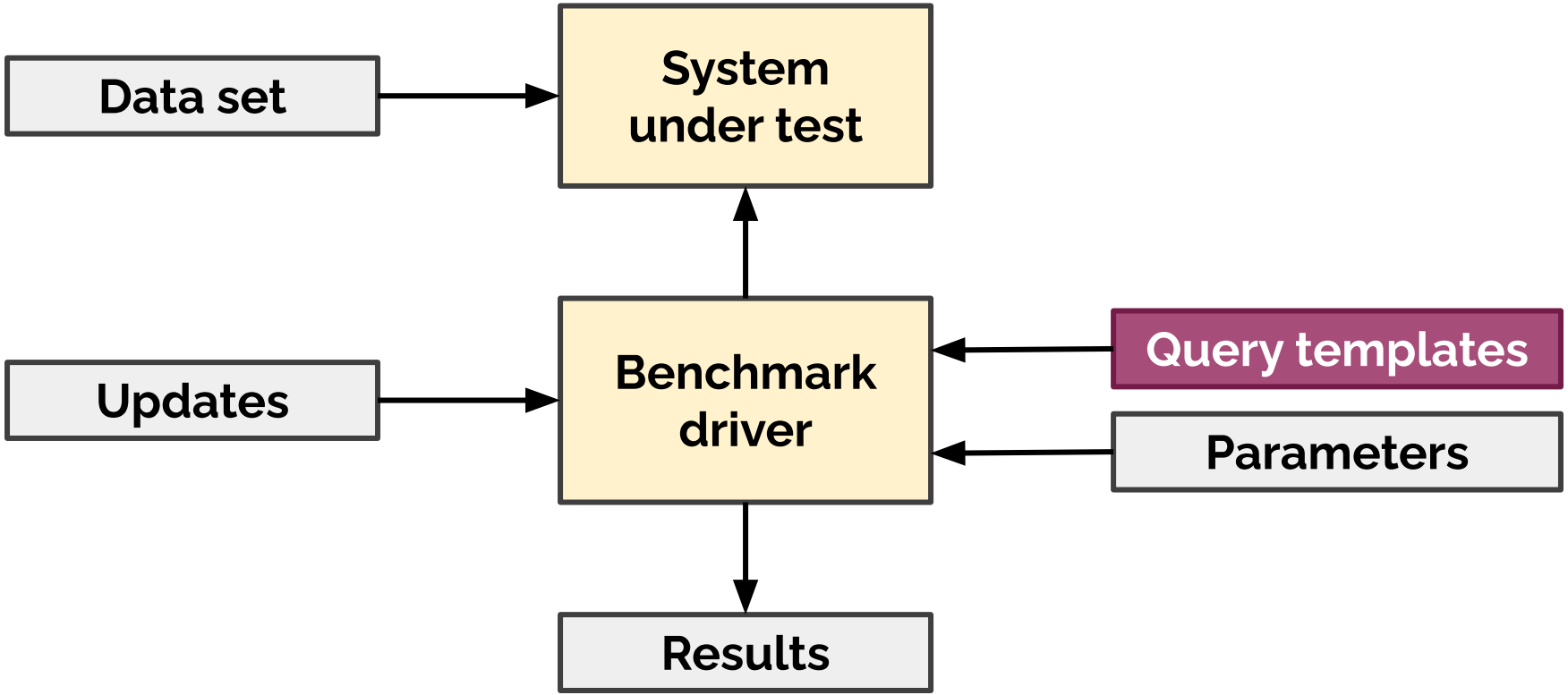
Updates

+ knows("Eve", "Gia")

+ Message("Gia", "M3")

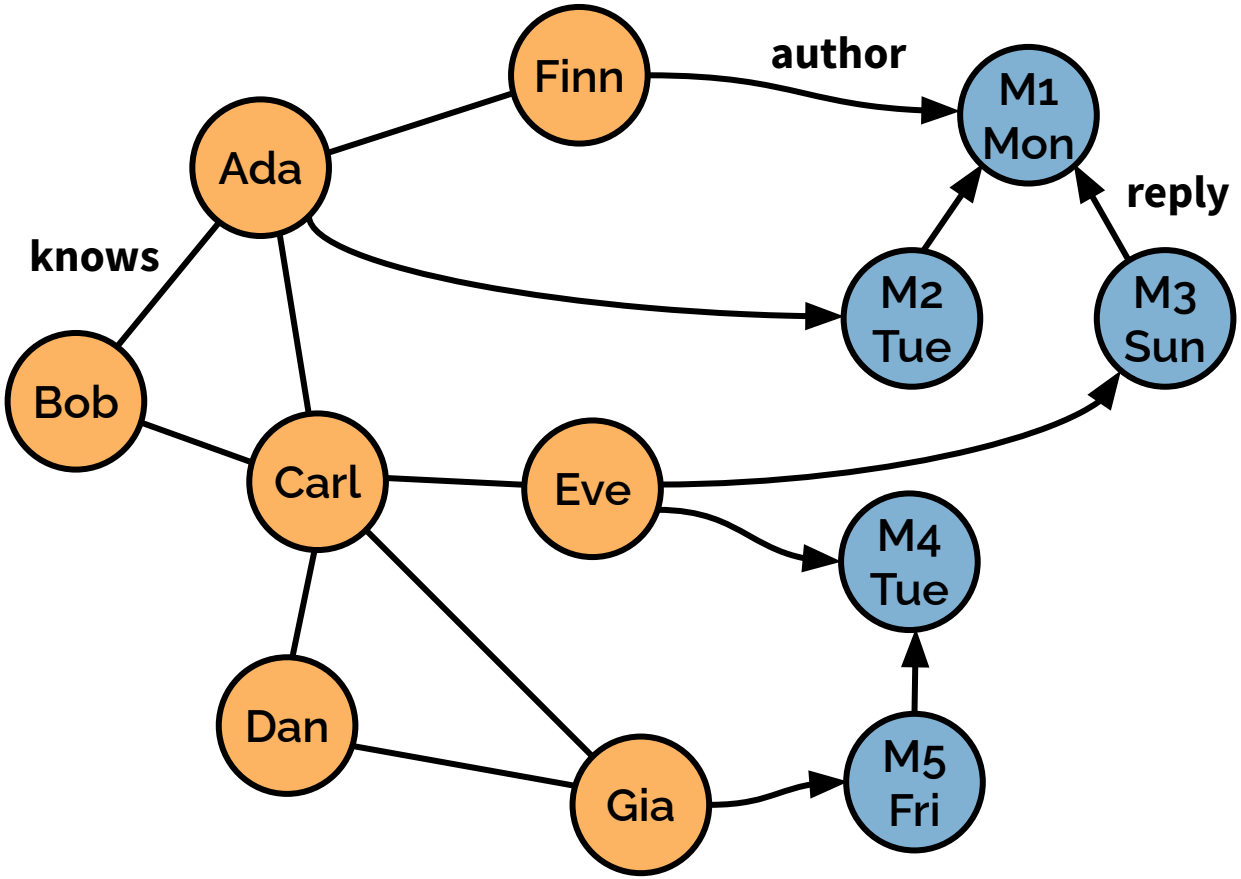
– Person("Eve")

Unique feature! LDBC SNB is the only database benchmark with deep deletes


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Query 1: Message categorization

Query 1: Message categorization



Q1($cutoffDate)

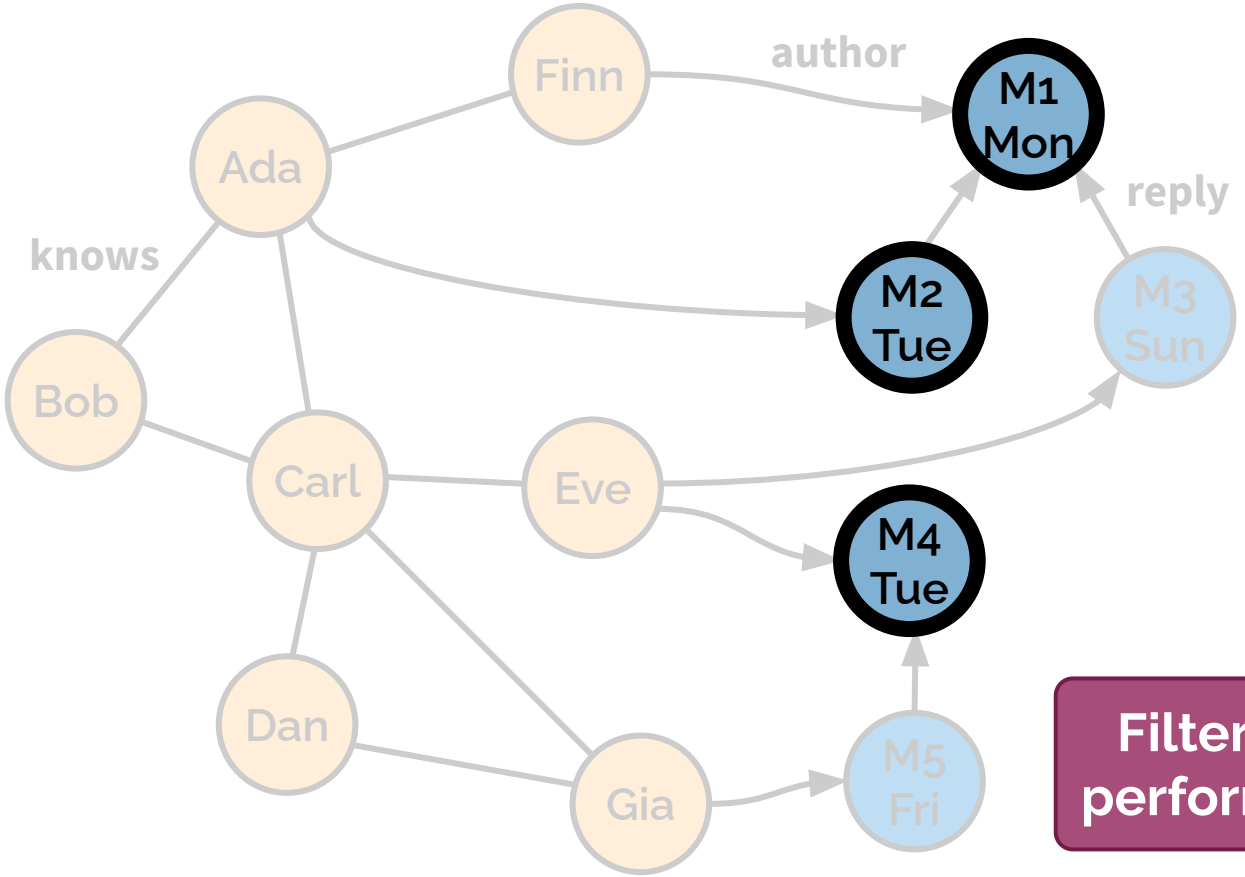
Msg

date < **$cutoffDate**

group by:

- type
- year
- length

Query 1: Message categorization



Q1(*"Wed"*)

Msg

date < ***"Wed"***

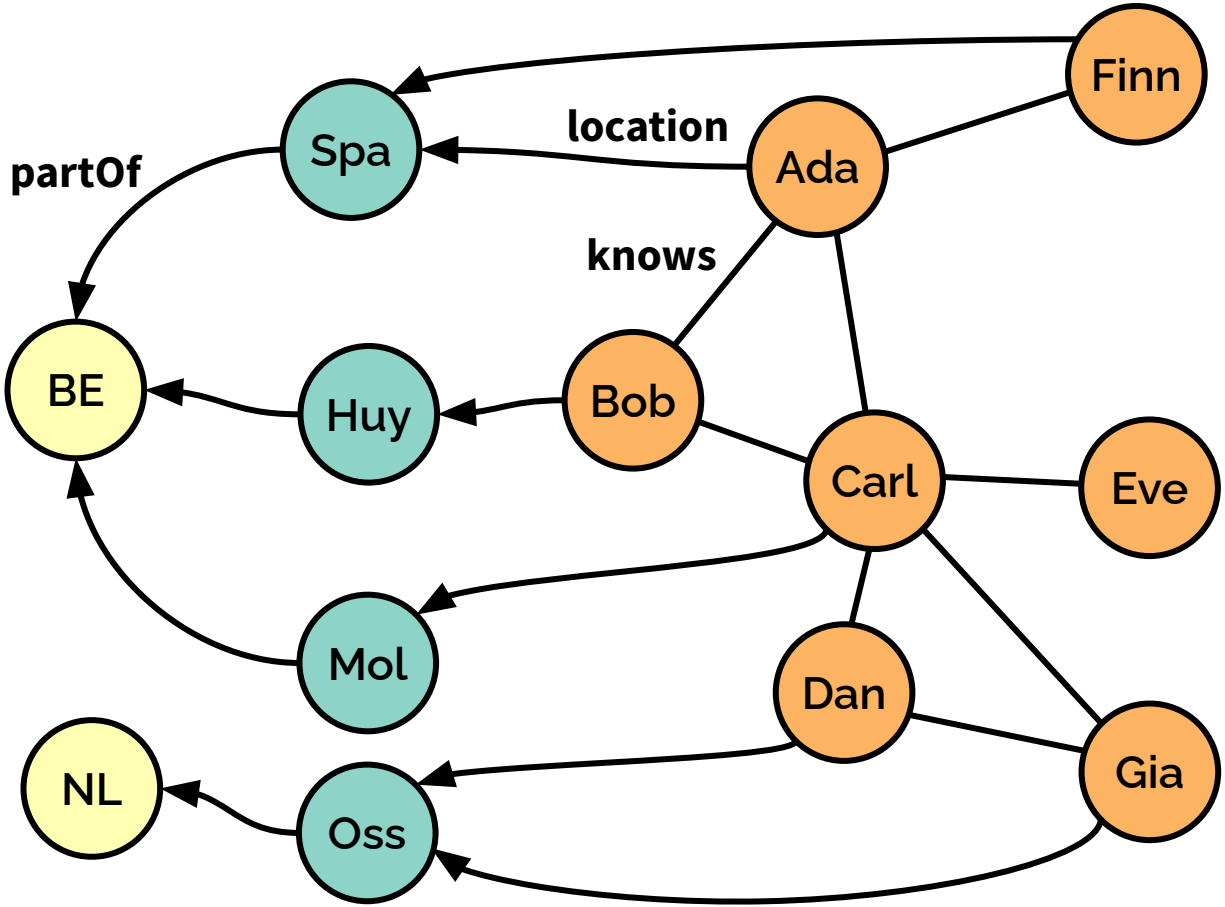
group by:

- type
- year
- length

Filtering and aggregation performances are important

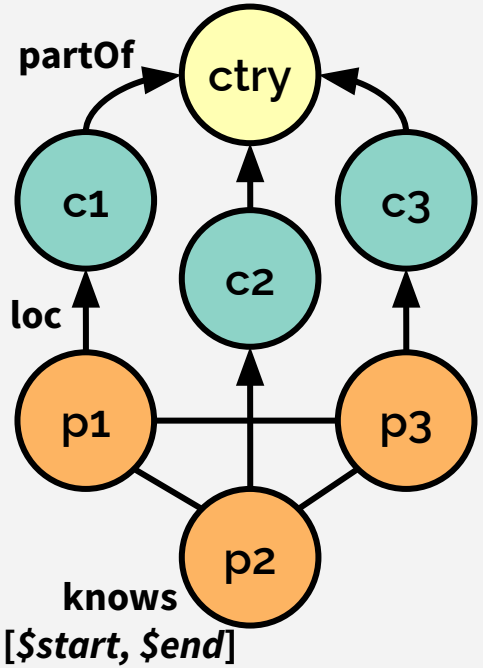
Query 11: Person triangles

Query 11: Person triangles

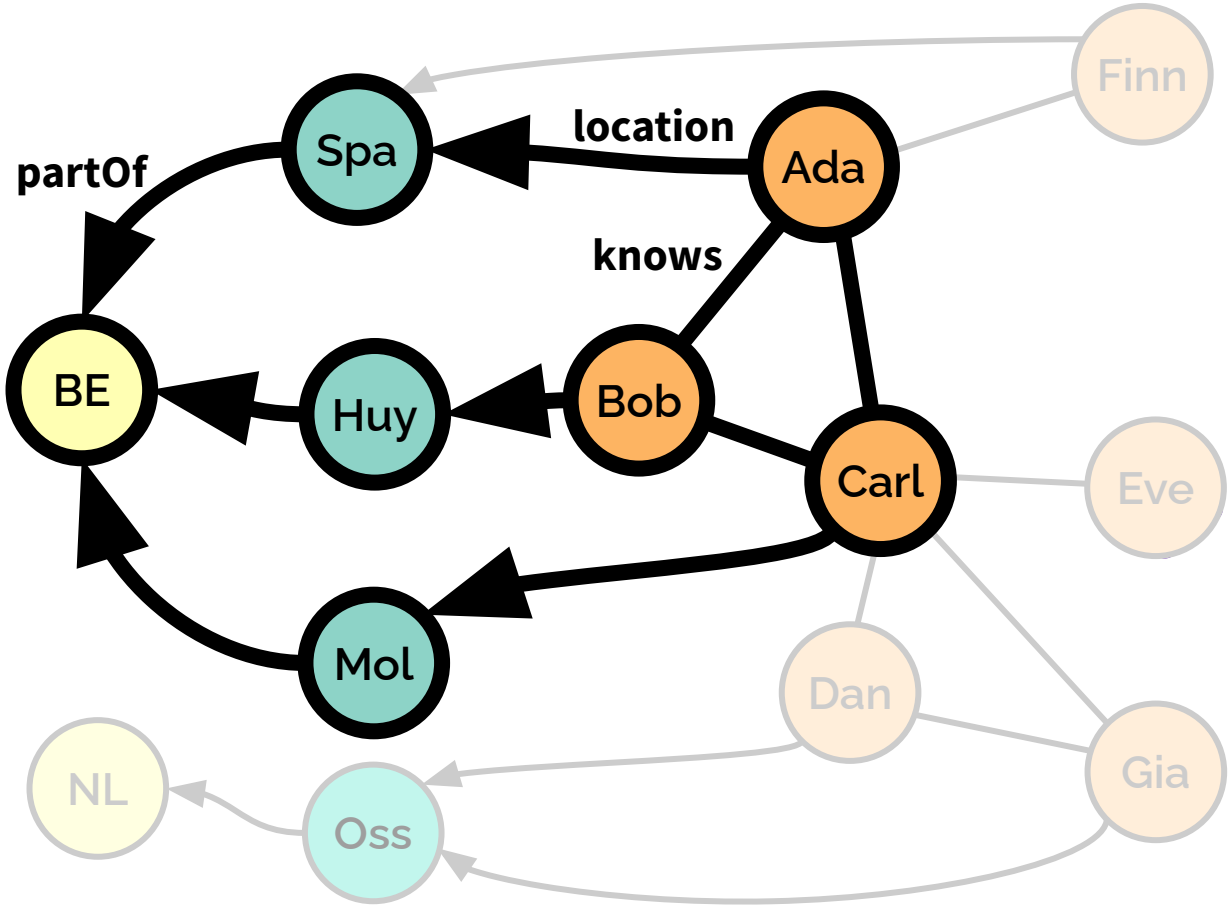


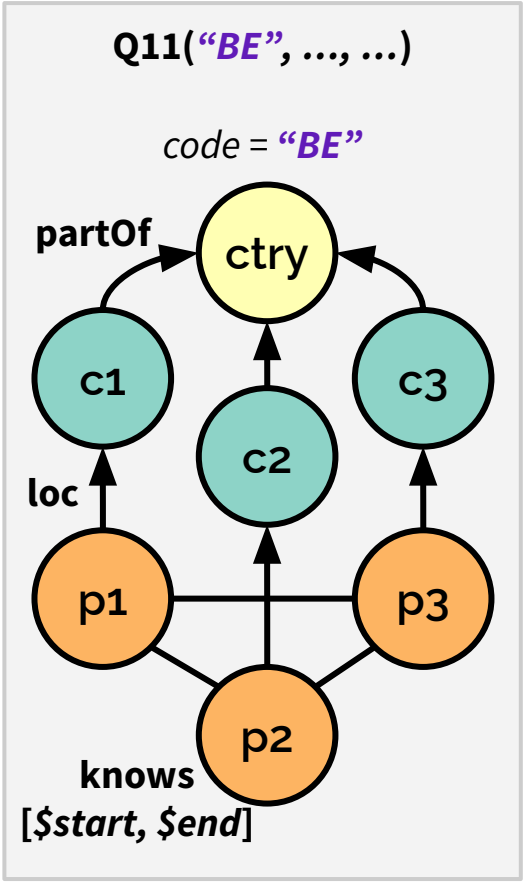
Q11(*$country, $start, $end*)

code = ***$country***



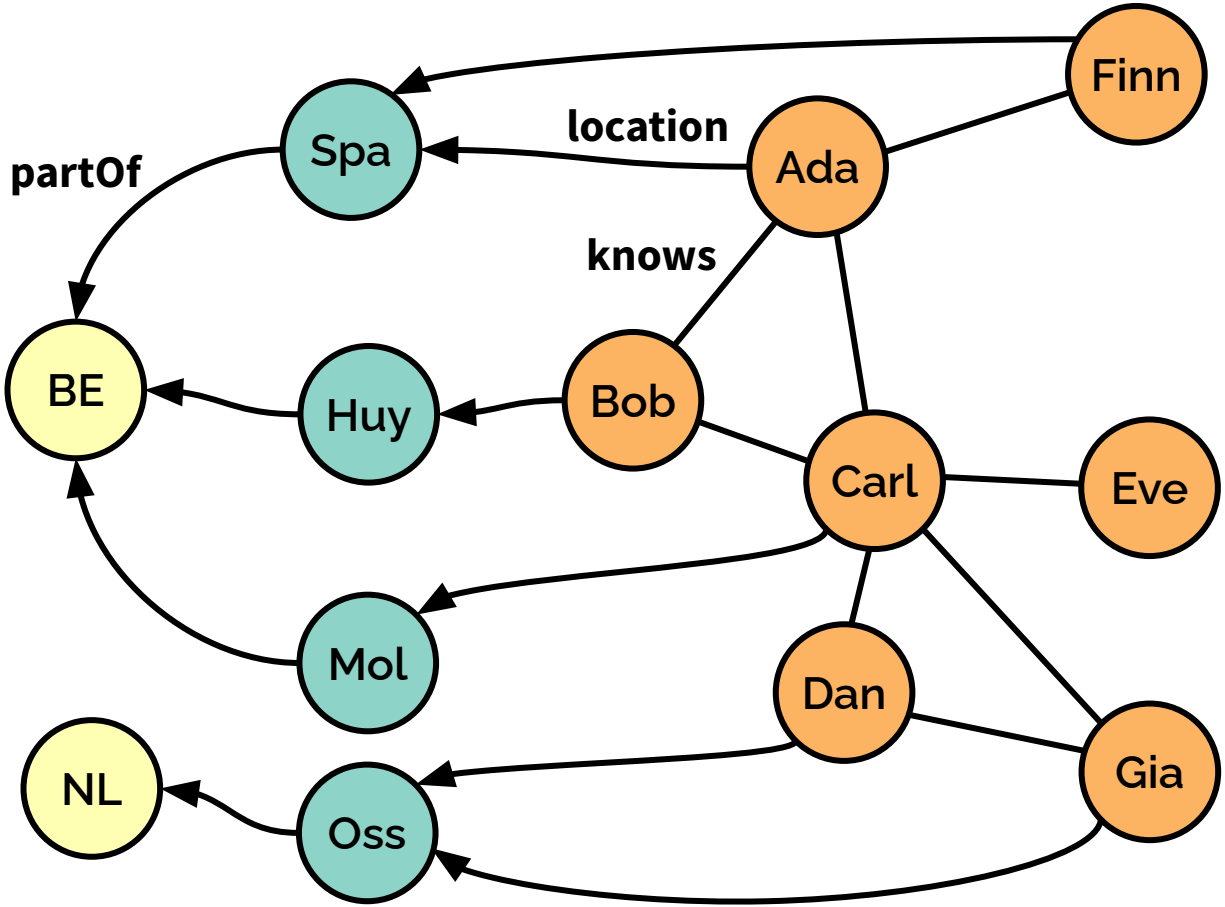
Query 11: Person triangles

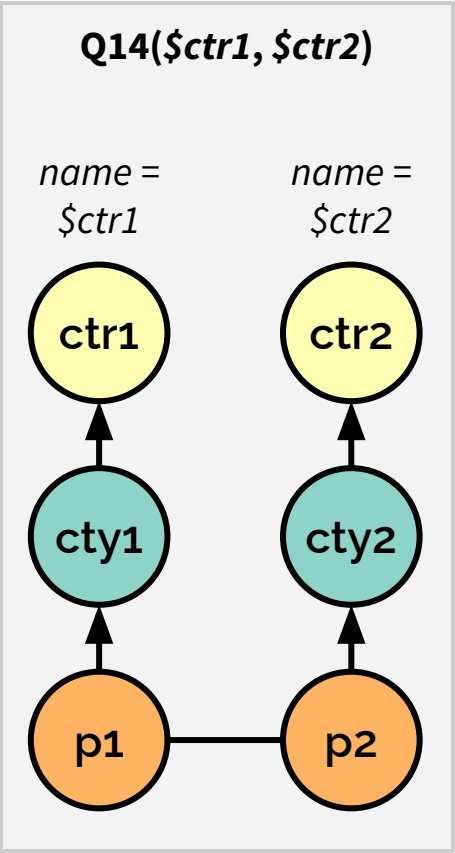




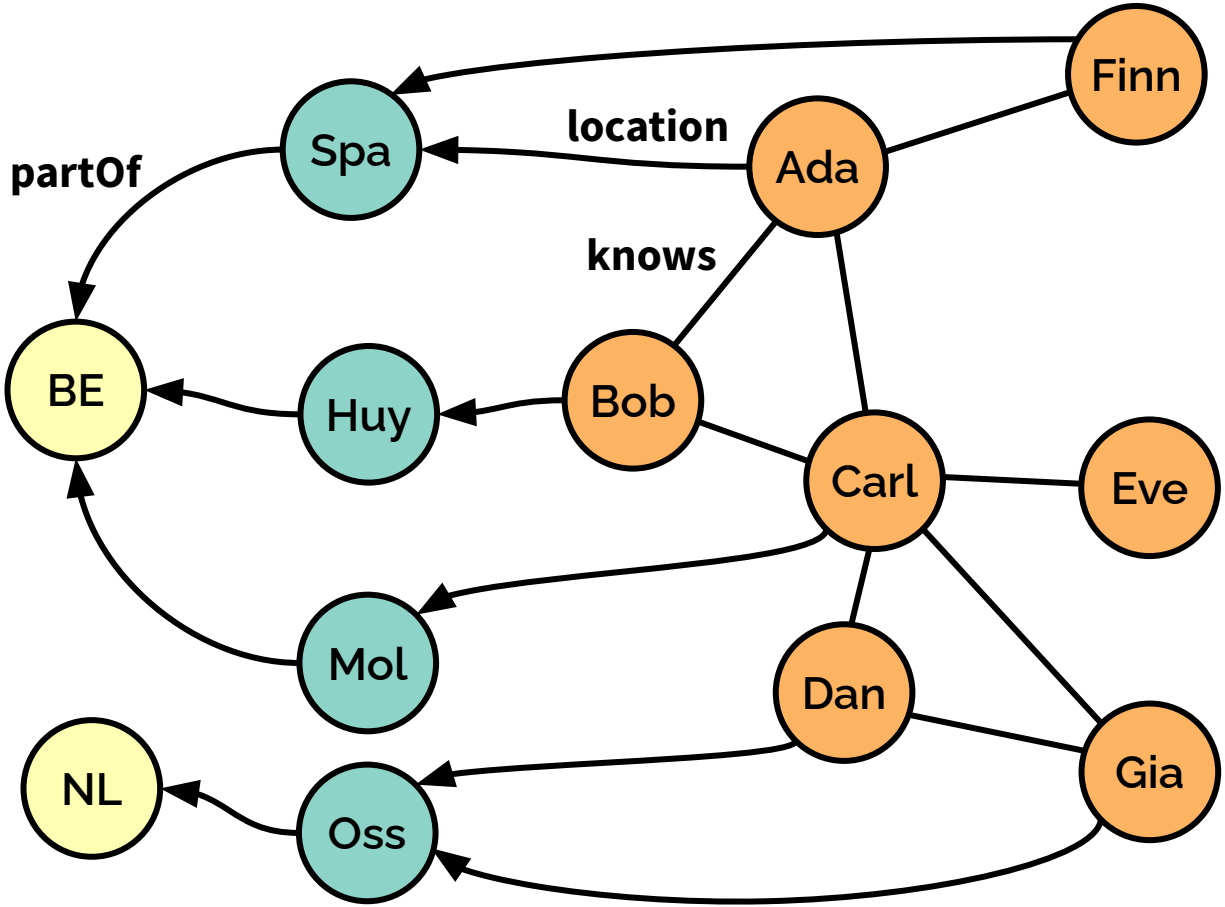
Query 14: International dialog

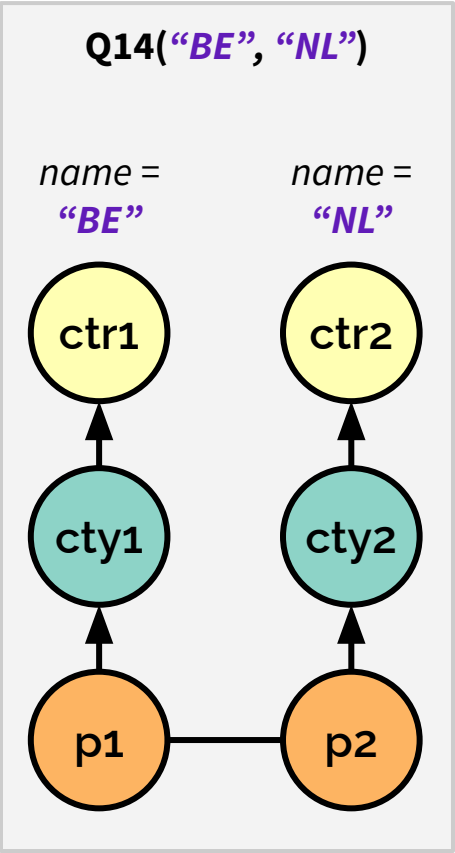
Query 14: International dialog



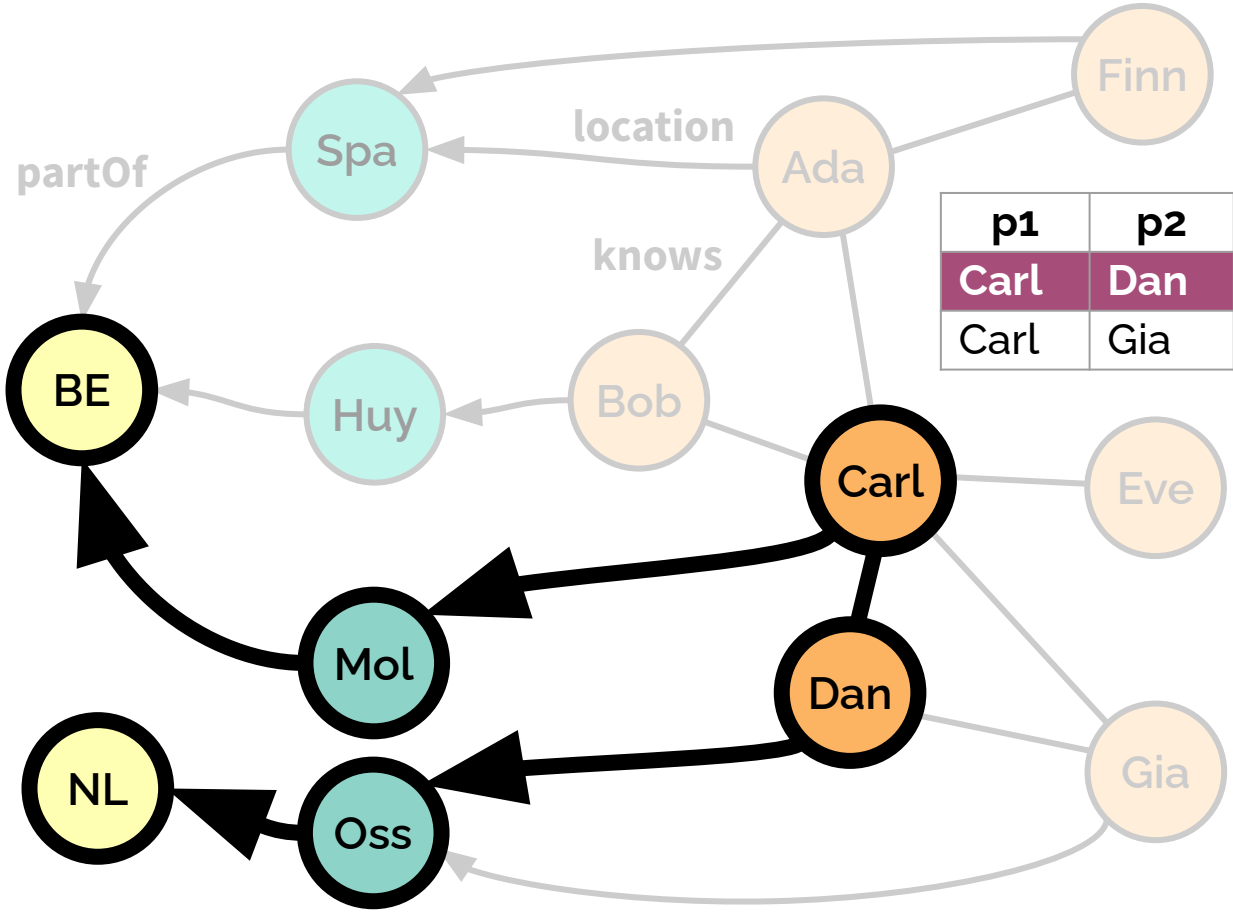


Query 14: International dialog

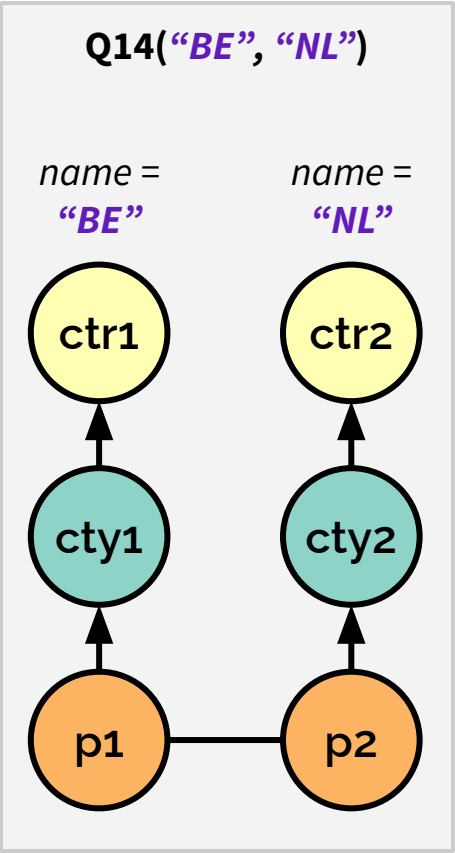




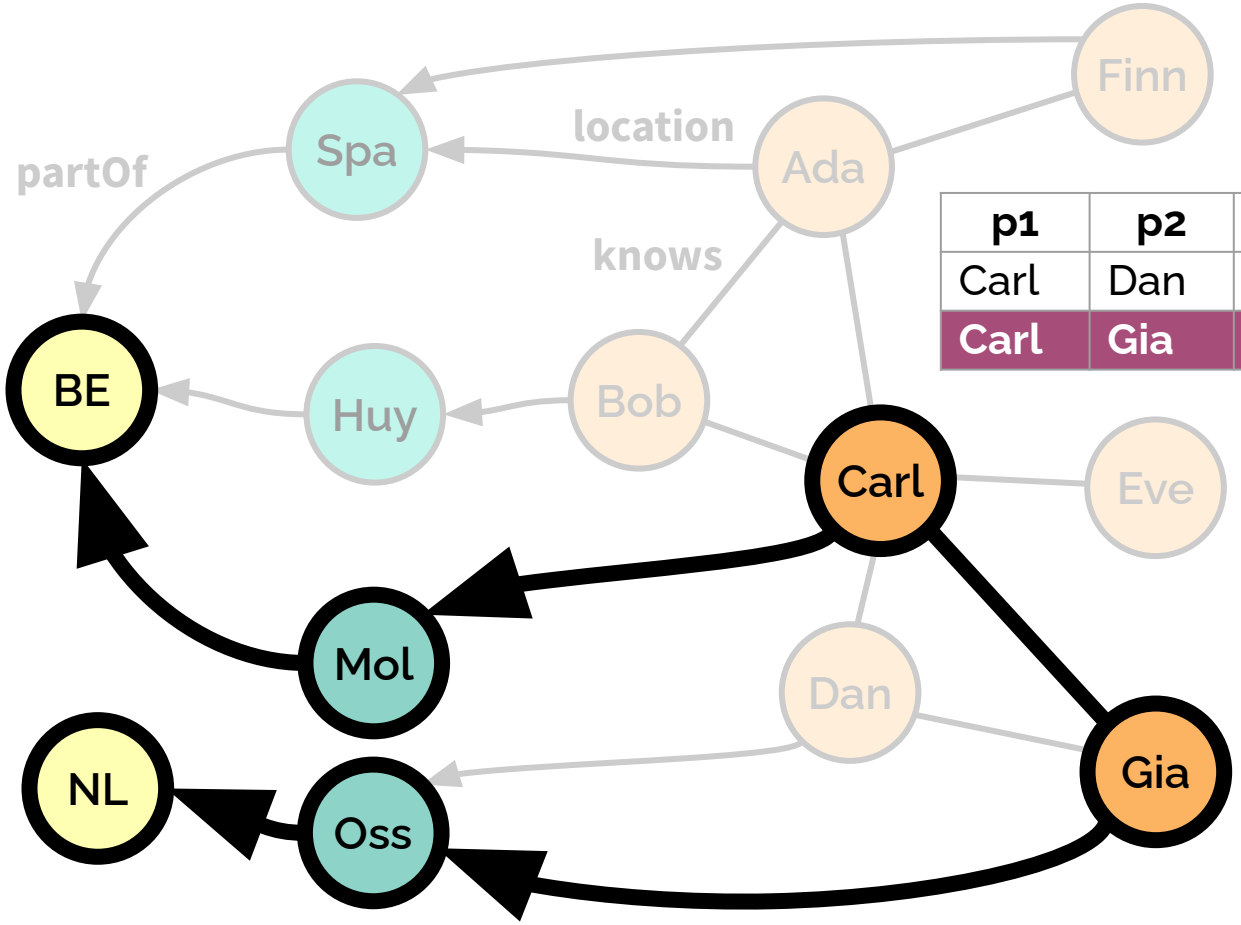
Query 14: International dialog



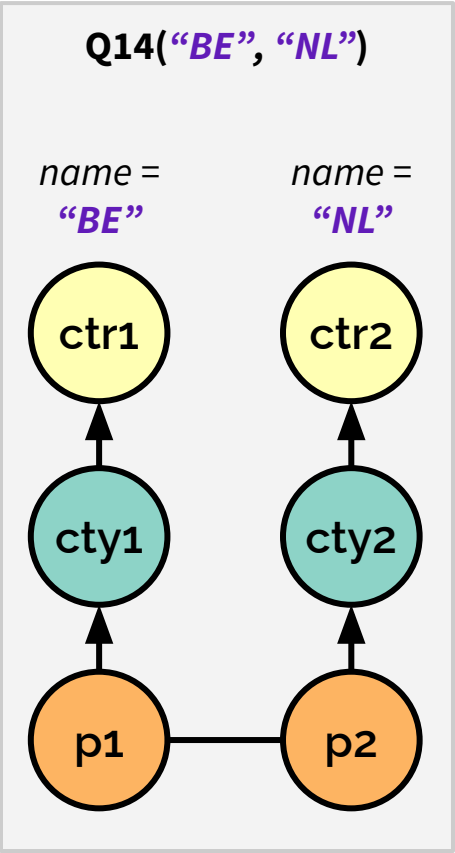
p1	p2	w
Carl	Dan	...
Carl	Gia	...



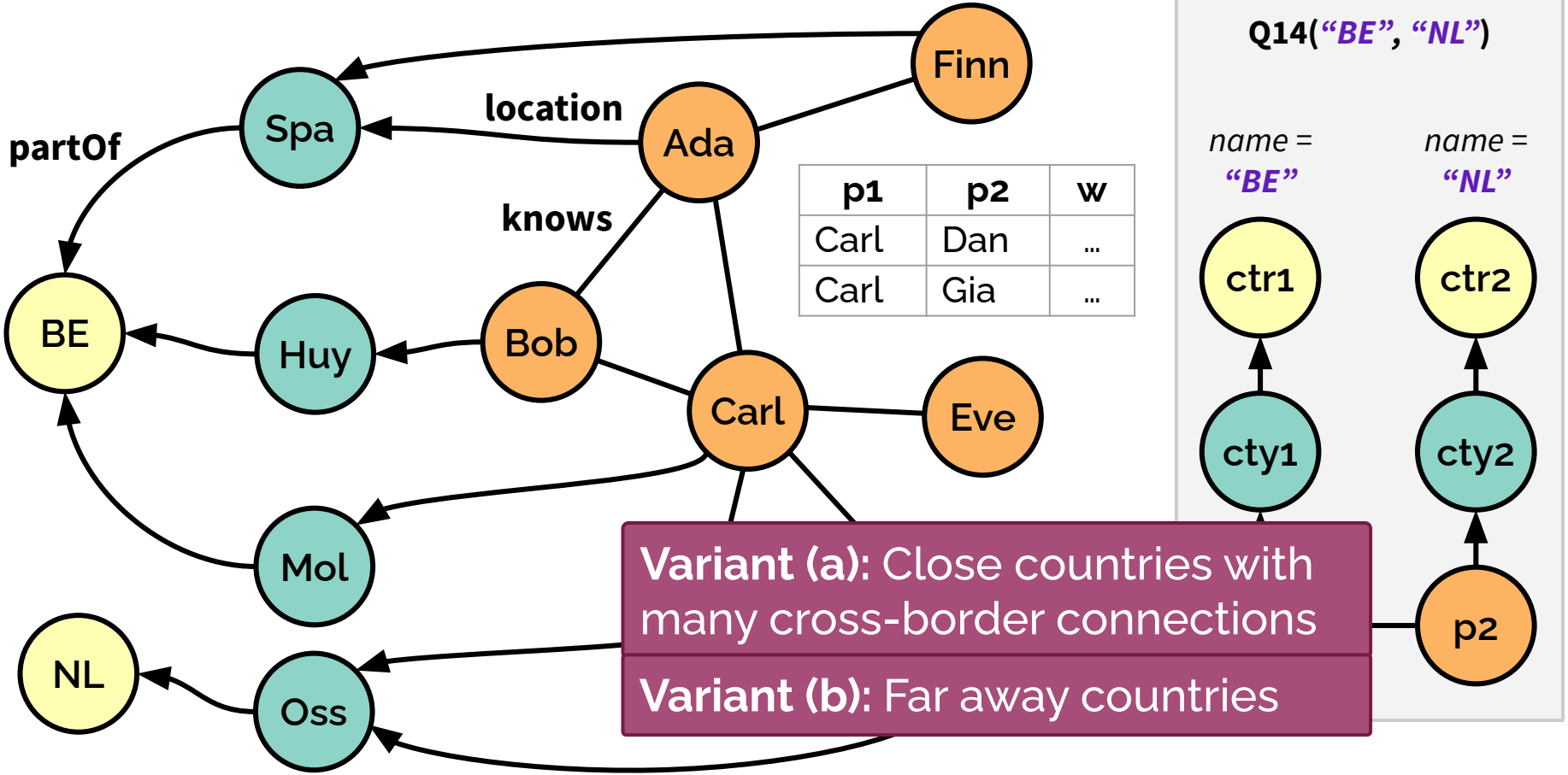
Query 14: International dialog



p1	p2	w
Carl	Dan	...
Carl	Gia	...



Query 14: International dialog



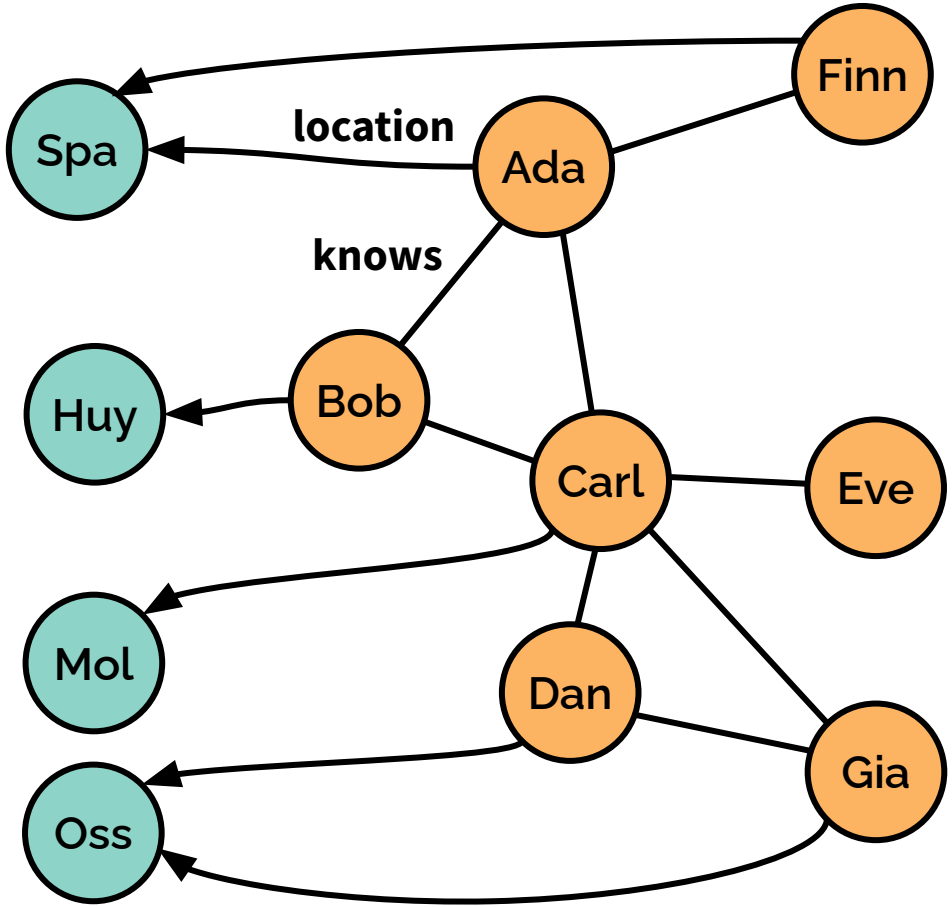
p1	p2	w
Carl	Dan	...
Carl	Gia	...

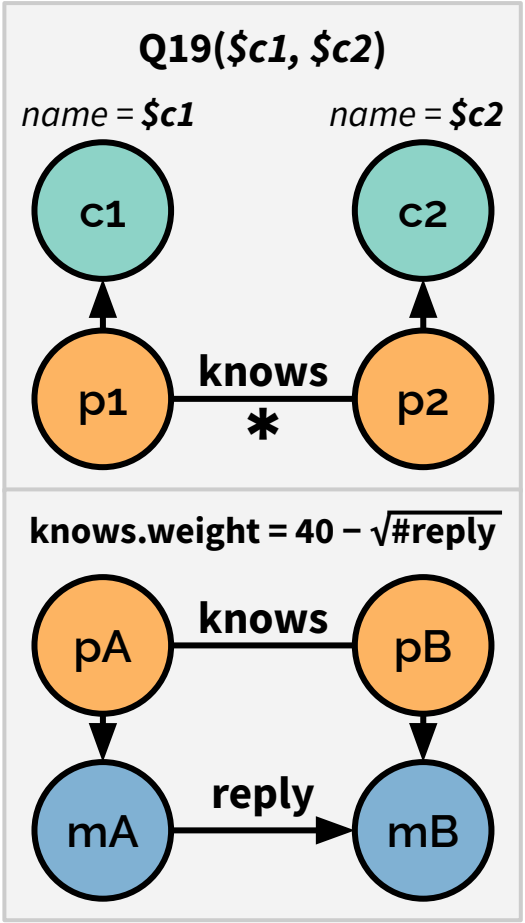
Variant (a): Close countries with many cross-border connections

Variant (b): Far away countries

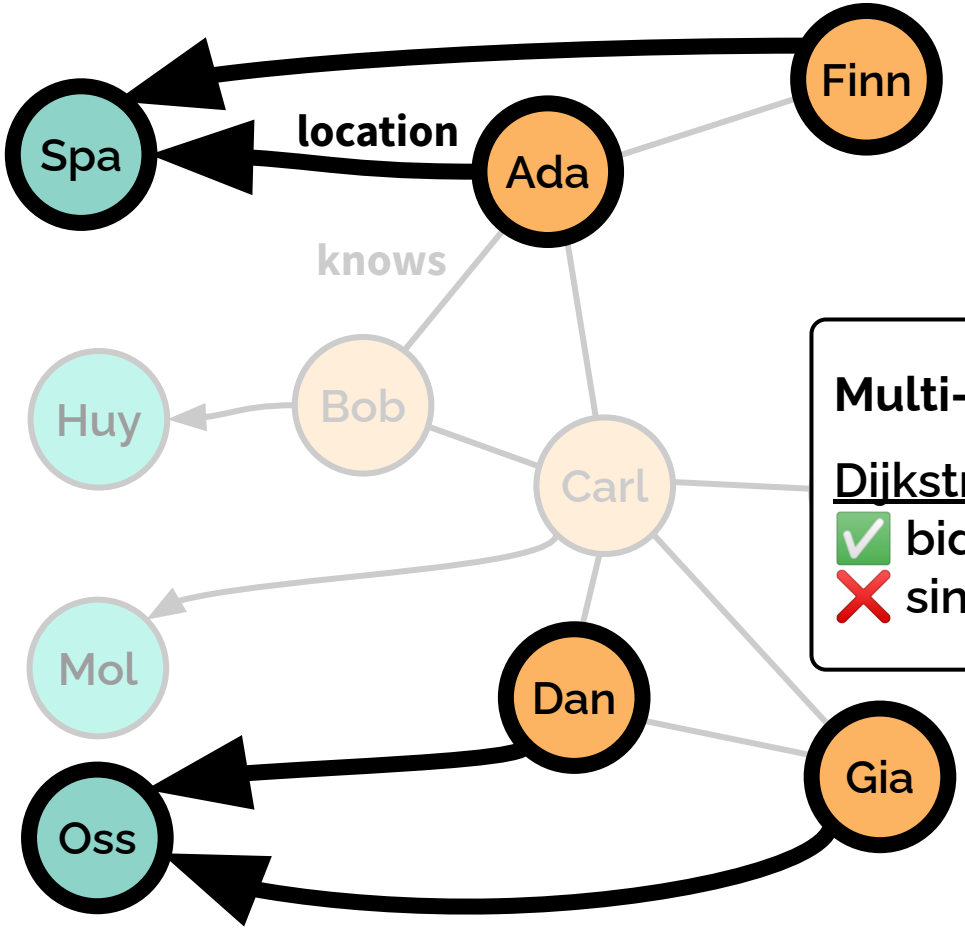
Query 19: Cheapest paths (weighted shortest)

Query 19: Cheapest paths (weighted shortest)

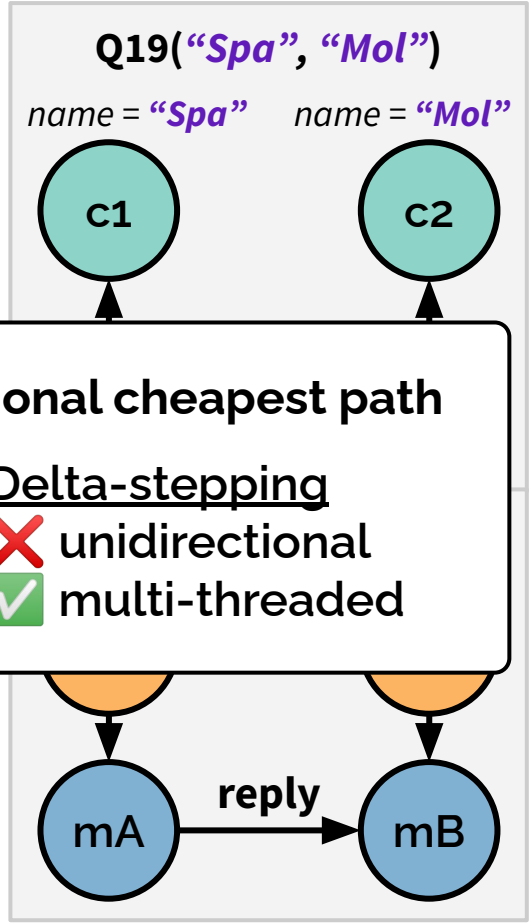




Query 19: Cheapest paths (weighted shortest)

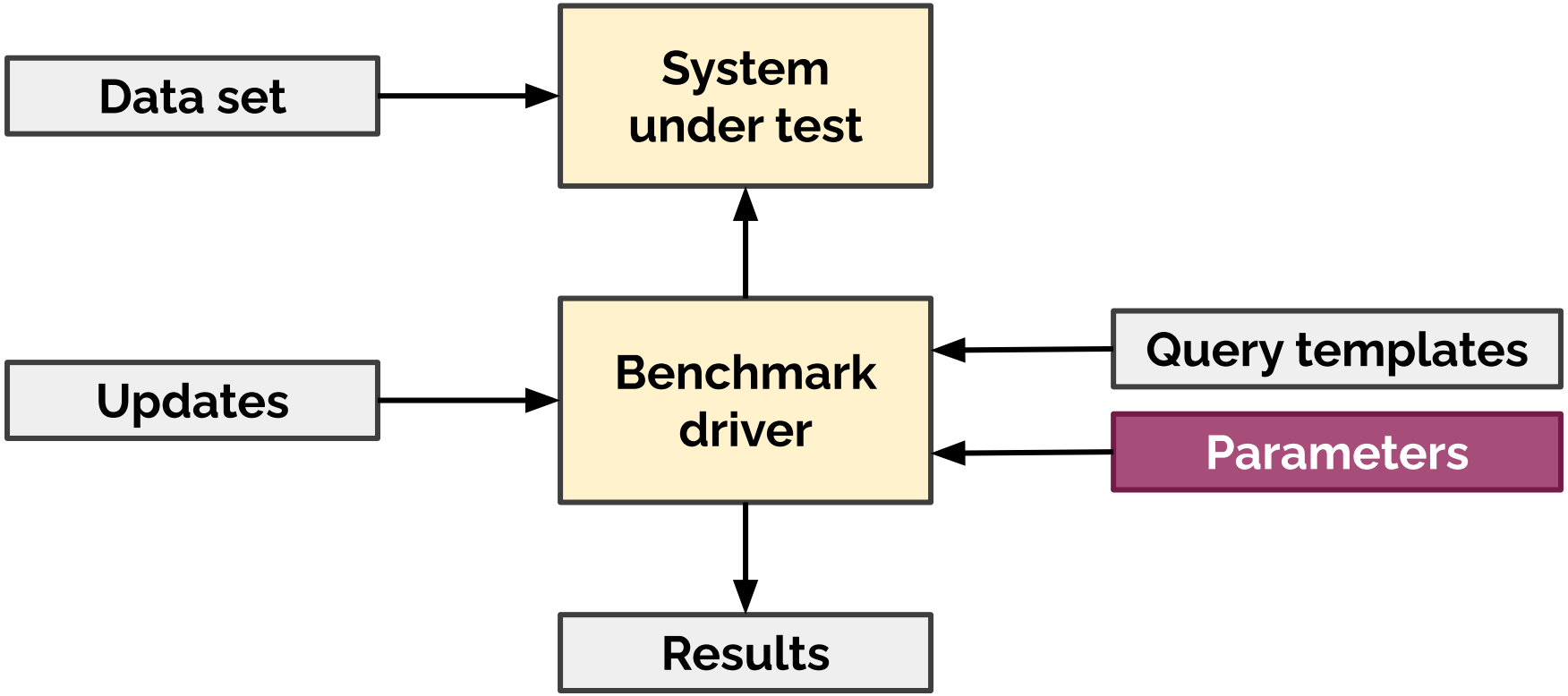


src	trg	w
Ada	Dan	...
Ada	Gia	...
Finn	Dan	...
Finn	Gia	...



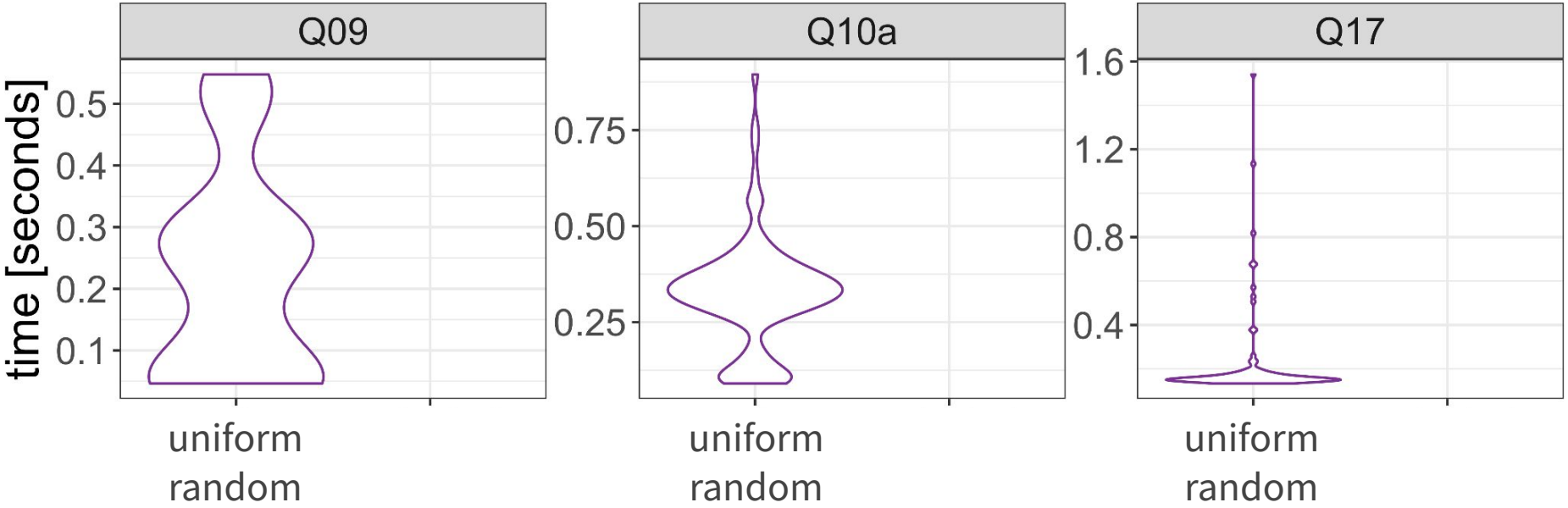
Multi-source bidirectional cheapest path

Dijkstra's algorithm	Delta-stepping
✅ bidirectional	❌ unidirectional
❌ single-threaded	✅ multi-threaded

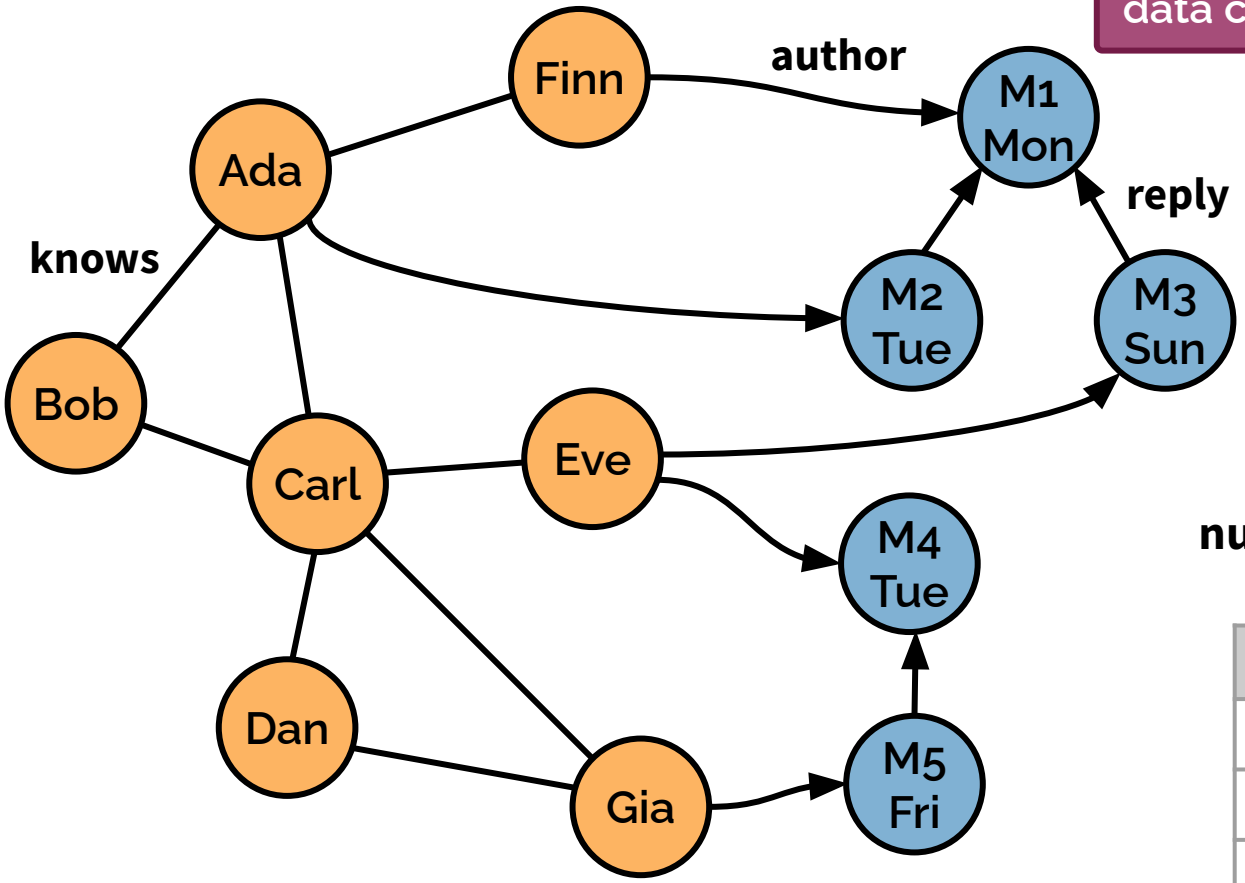

Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Parameter selection

- ***Uniform random parameters*** → unstable distributions



Factor tables capture
data cube-like summary statistics



numFriendsOfFriends

name	#1-hop	#2-hop
Bob	2	4
Ada	3	3
Carl	5	1
…		

numMessages-PerDay

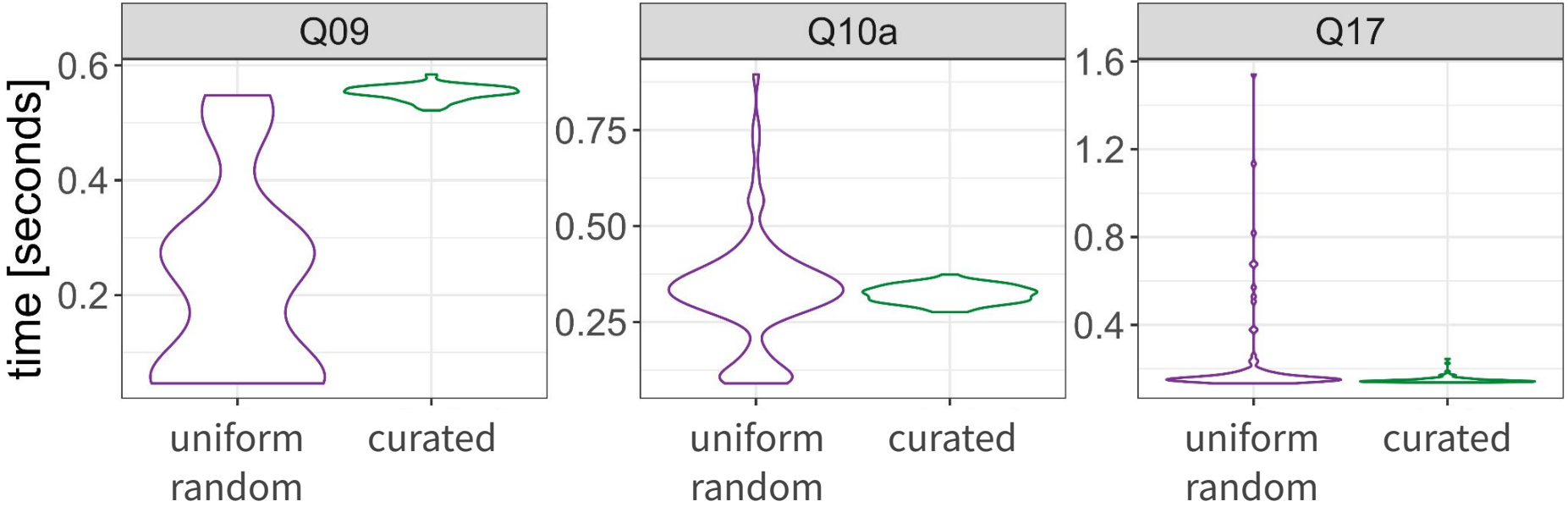
day	#
Mon	1
Tue	2
…	

numPersons-PerCity

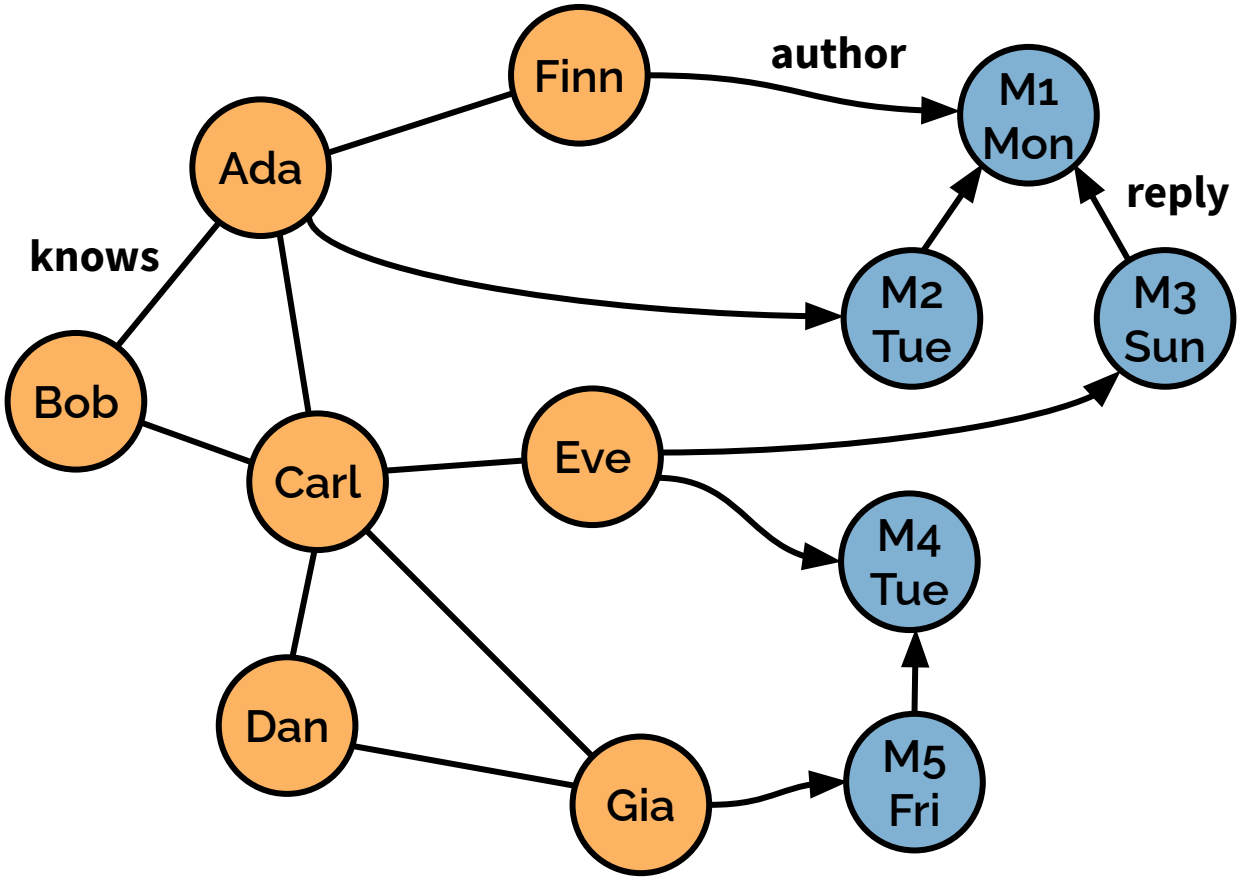
city	#
Spa	2
Mol	2
…	

Parameter selection

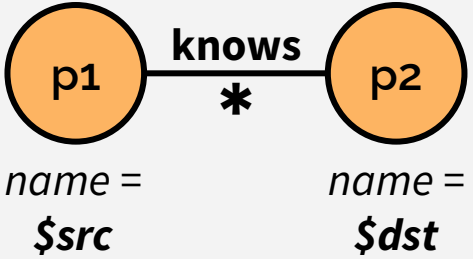
- ***Uniform random parameters*** → unstable distributions
- ***Curated parameters*** → tighter distributions, closer to bell curves



Shortest path

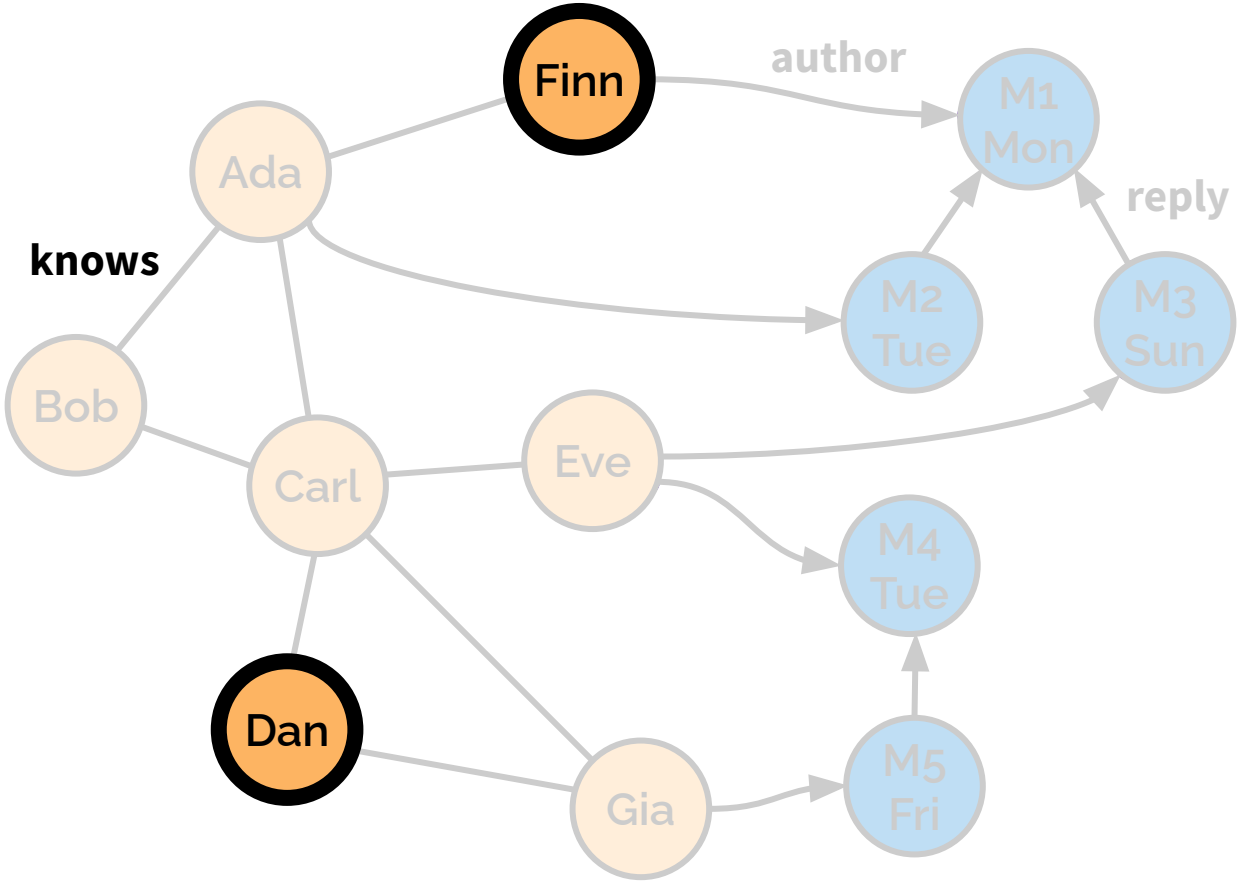


shortest($src, $dst)



Updates

Shortest path



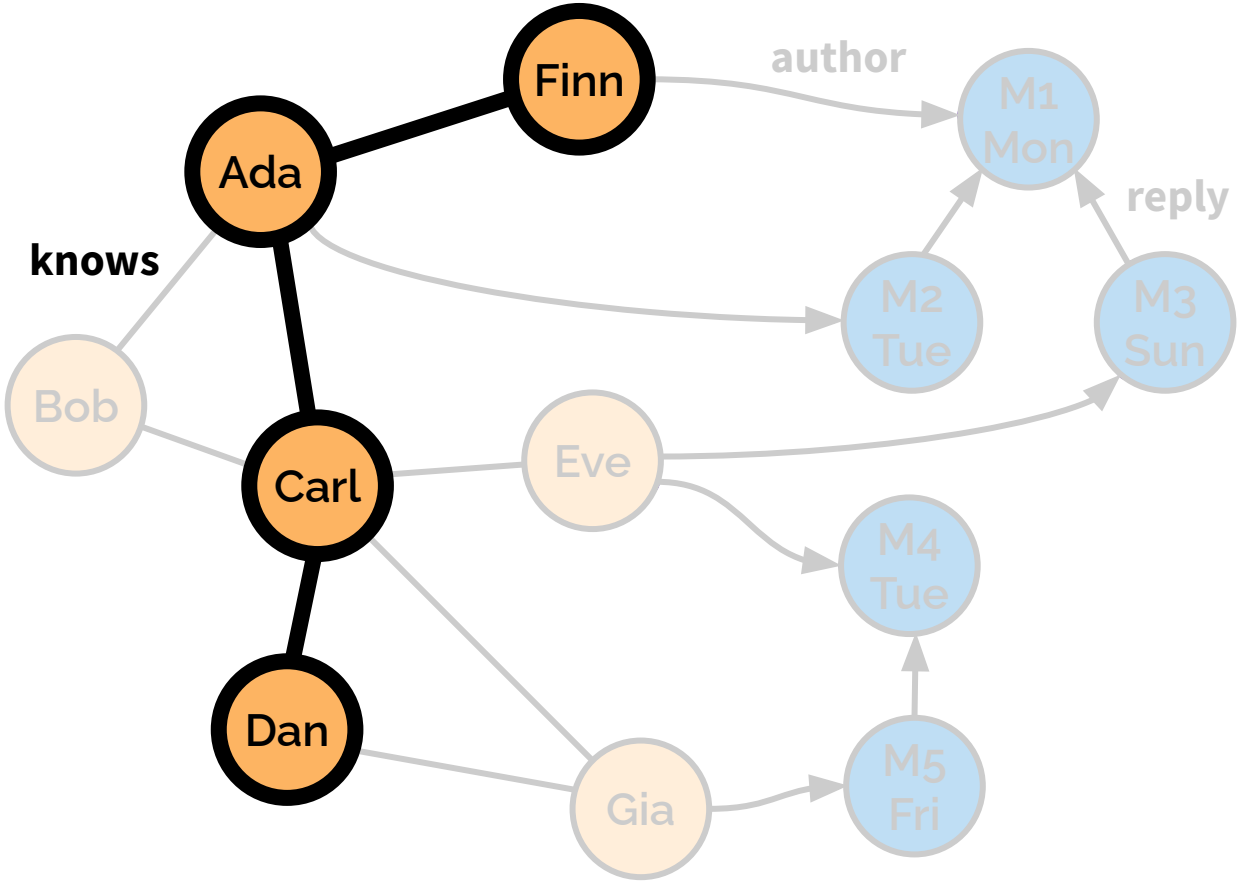
shortest(*"Finn"*, *"Dan"*)

name = ***"Finn"***

name = ***"Dan"***

Updates

Shortest path



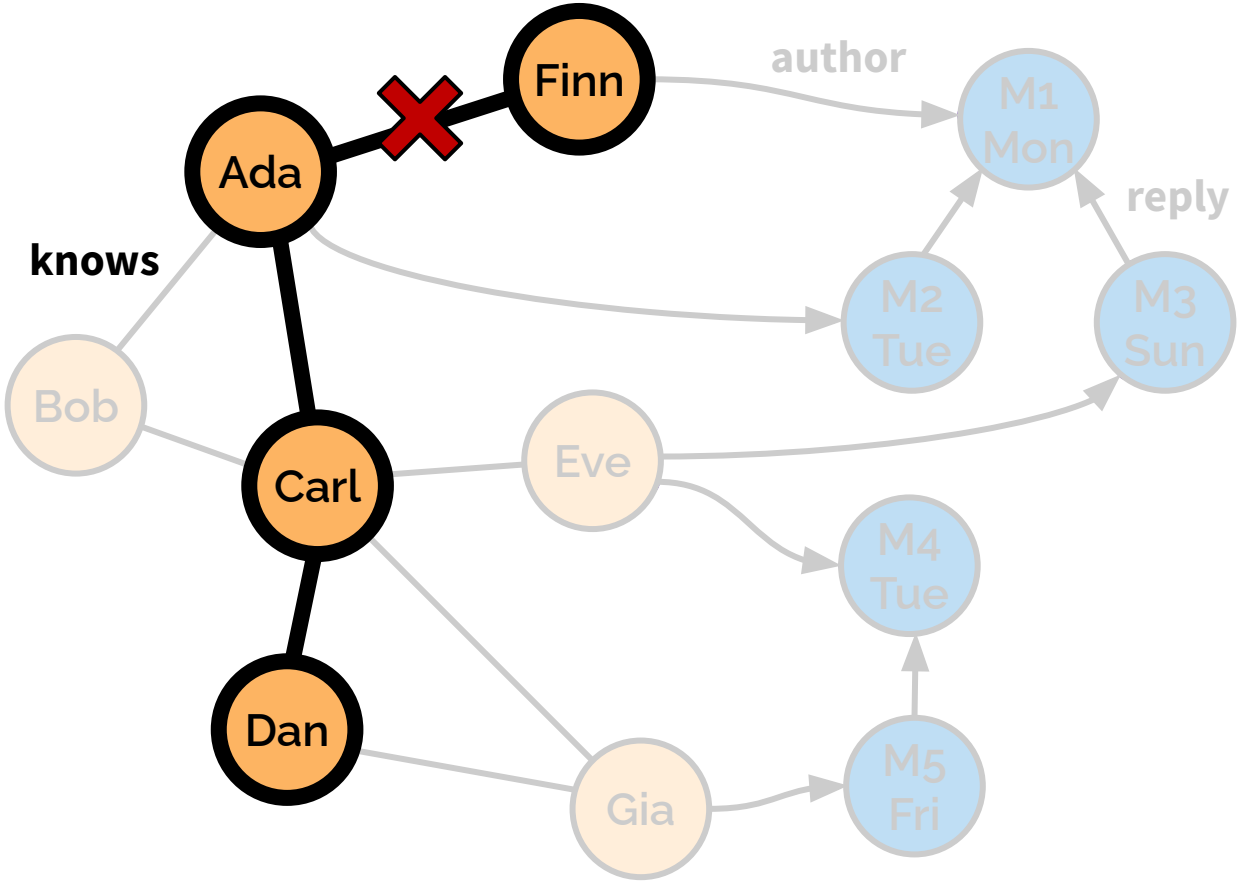
shortest(*"Finn"*, *"Dan"*)

name = ***"Finn"***

name = ***"Dan"***

Updates

Shortest path



shortest(*"Finn"*, *"Dan"*)

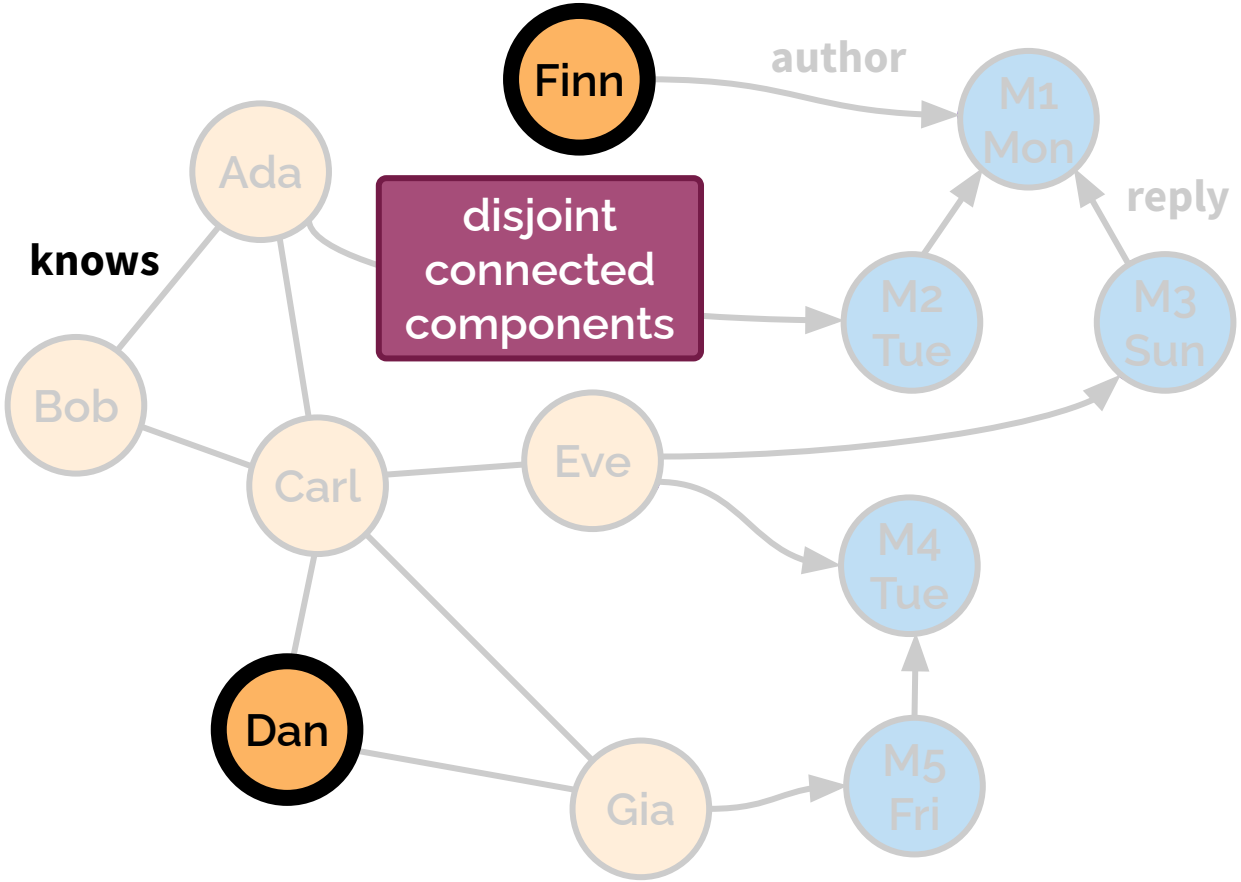
name = ***"Finn"***

name = ***"Dan"***

Updates

– knows("Ada", "Finn")

Shortest path



shortest(*"Finn"*, *"Dan"*)

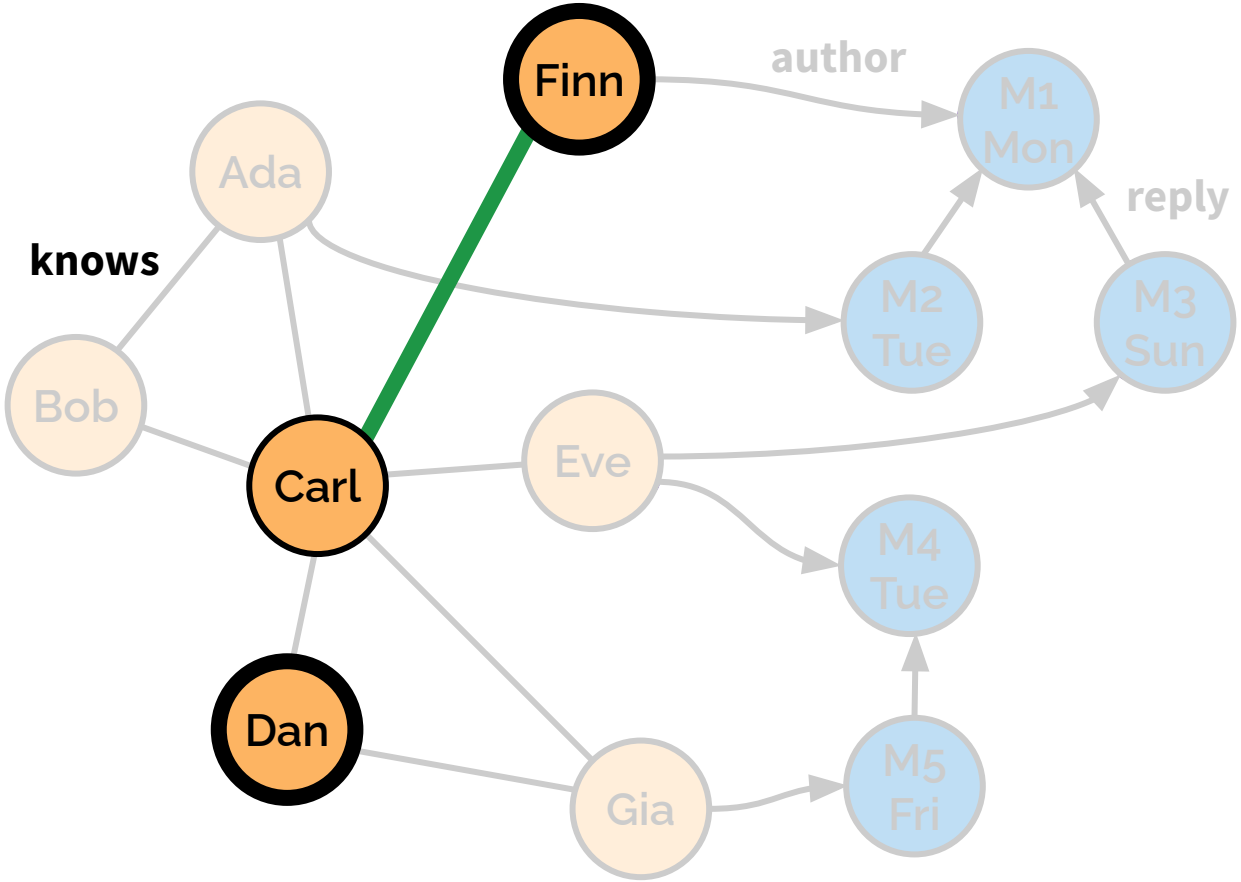
name = ***"Finn"***

name = ***"Dan"***

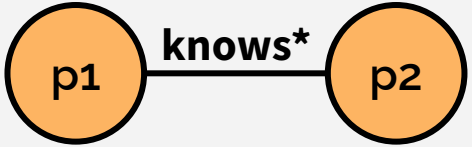
Updates

– knows("Ada", "Finn")

Shortest path



shortest(*"Finn"*, *"Dan"*)



name = ***"Finn"***

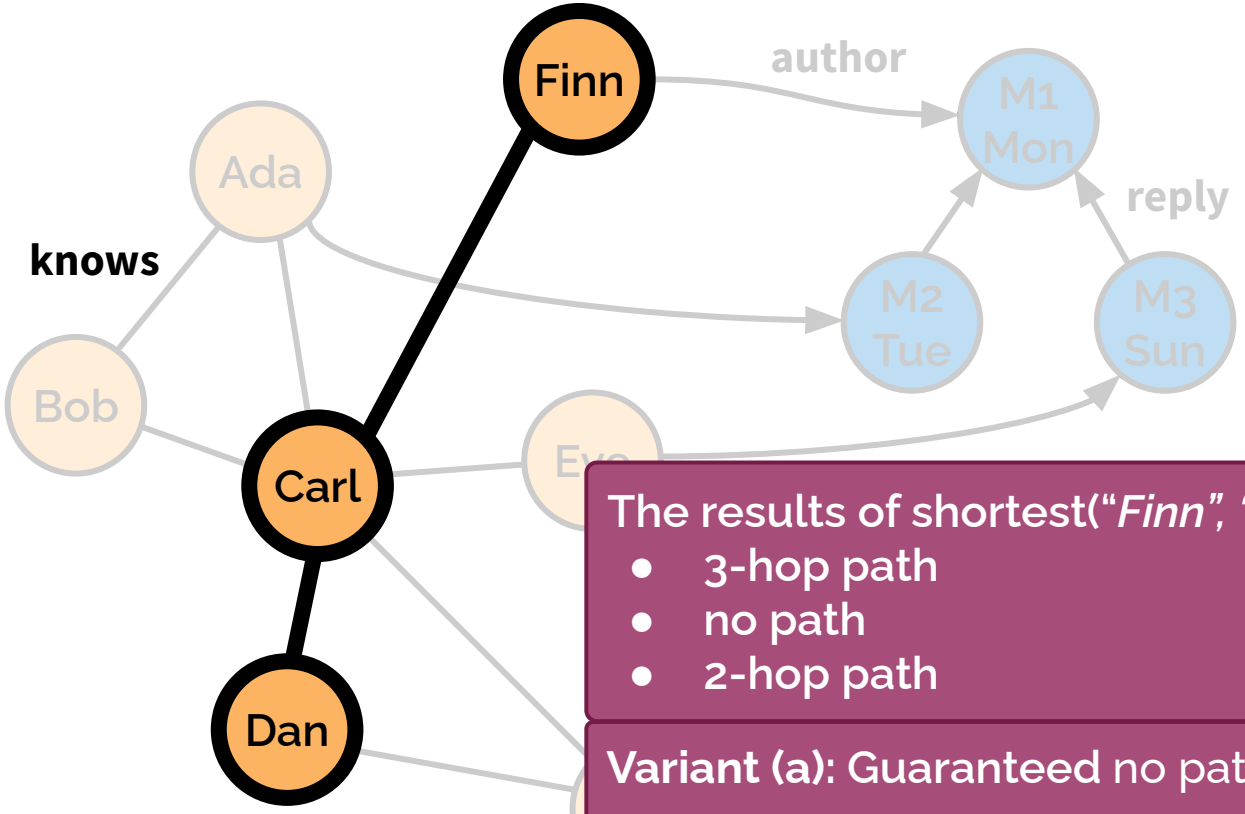
name = ***"Dan"***

Updates

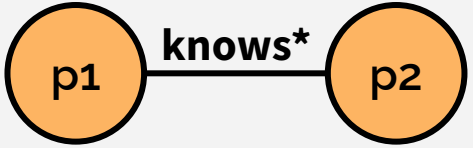
– knows("Ada", "Finn")

+ knows("Finn", "Carl")

Shortest path



shortest(*"Finn"*, *"Dan"*)



name = ***"Finn"***

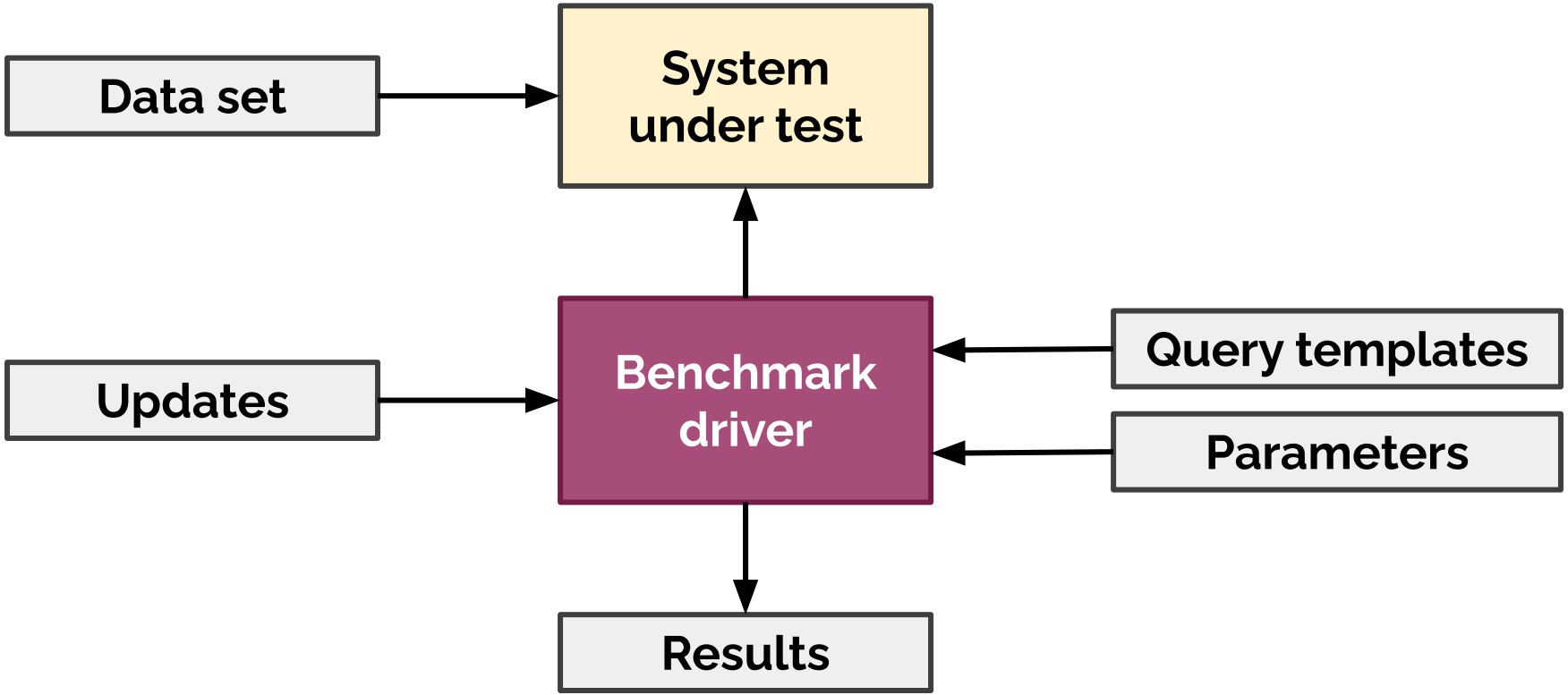
name = ***"Dan"***

The results of shortest("*Finn*", "*Dan*") change over time:

- 3-hop path
- no path
- 2-hop path

Variant (a): Guaranteed no path

Variant (b): Guaranteed path with length <= 4 hops


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Benchmark driver

1) Executes the benchmark

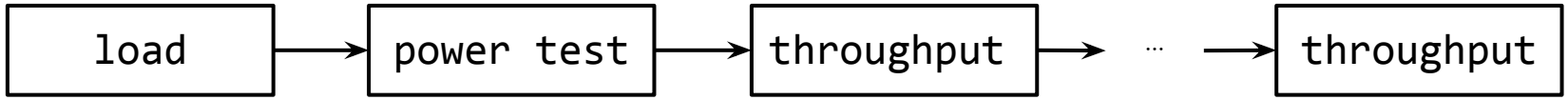
2) Cross-validates systems

3) Calculates final scores

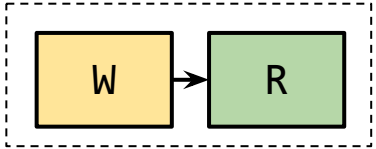
- **Power score:** Geometric mean of individual query runtimes
- **Throughput score:** Extrapolated daily throughput performance

Workload execution

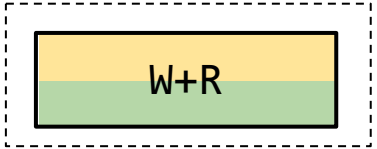
Execution happens in daily batches:



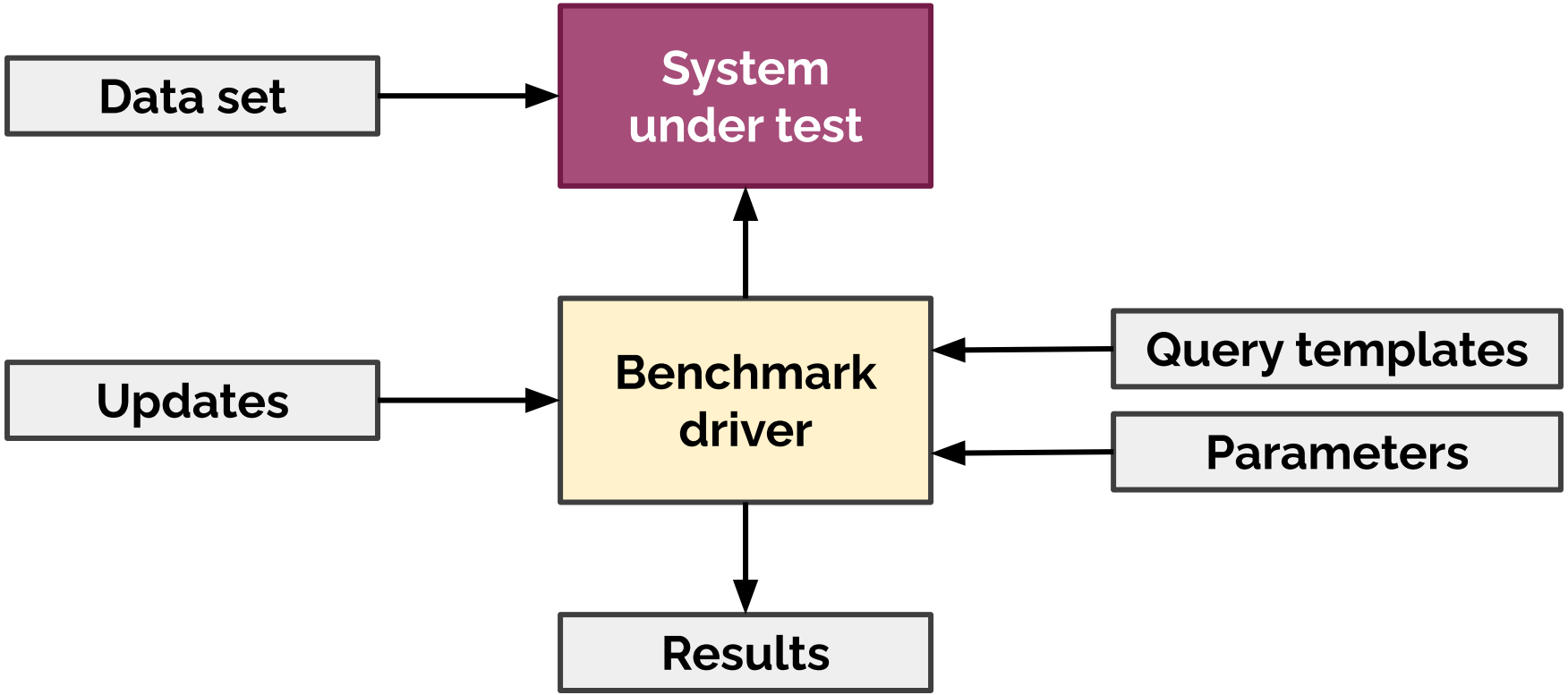
- **Writes:** 1 day of inserts and deletes
- **Reads:** 20 instances per query variant



Sequential model

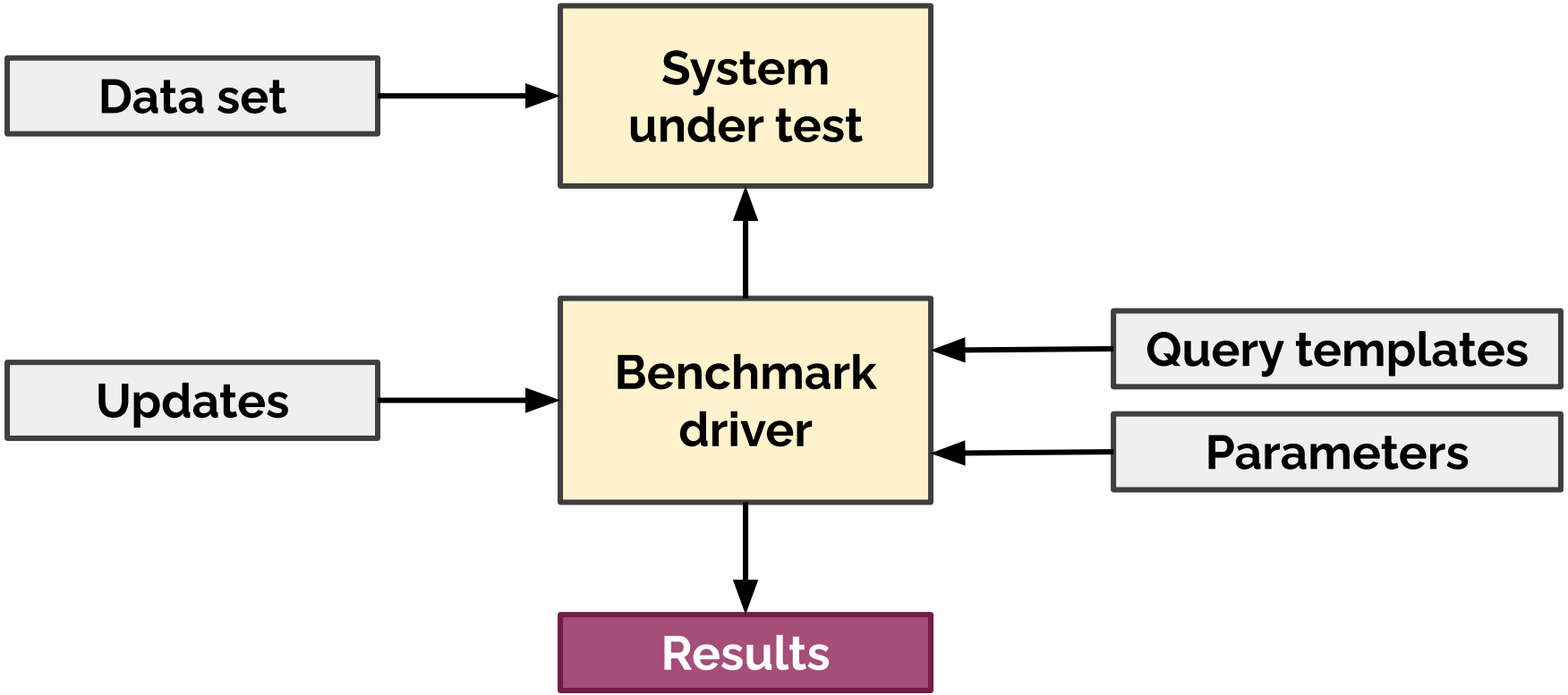


Concurrent model


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Implementations

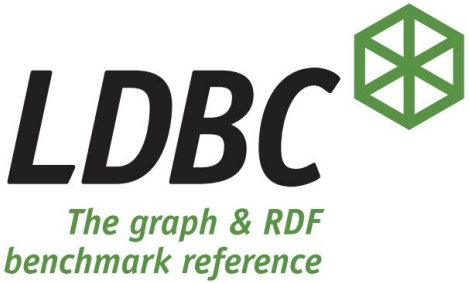
system	data model	language	LOC
neo4j	graph	Cypher	495
UMBRA	relational	SQL	755
TigerGraph	graph	GSQL	832


Data set
System under test
Updates
Benchmark driver
Query templates
Parameters
Results

Benchmark results

	Umbra			TigerGraph	
	SF30	**SF100**	**SF300**	**SF1,000**	**SF10,000**
power@SF	75,761.75	103,308.45	110,473.72	17,821.02	61,319.43
throughput@SF	n/a	28,996.42	26,251.13	7,655.88	23,132.08
load time	68.70	211.92	668.81	4,786.00	6,321.00
		...			
total execution time	3,333.71	4,122.39	4,910.60	20,908.95	63,314.11
experiment cost	$18.79	$21.26	$24.34	$66.75	$1,849.97

Audited results



Results for

- SF100
- SF1,000
- SF10,000

Full Disclosure Report
of the LDBC Social Network Benchmark

Audit of the LDBC Social Network Benchmark's
Business Intelligence Workload over TigerGraph

April 6, 2023

Related work on graph query processing

Algorithms and implementations

path finding

relational operators

pattern matching

direction-optimizing (push/pull) BFS (2012)

factorized joins (2012)

landmark labeling (2013)

worst-case optimal (multi-way) joins (2013)

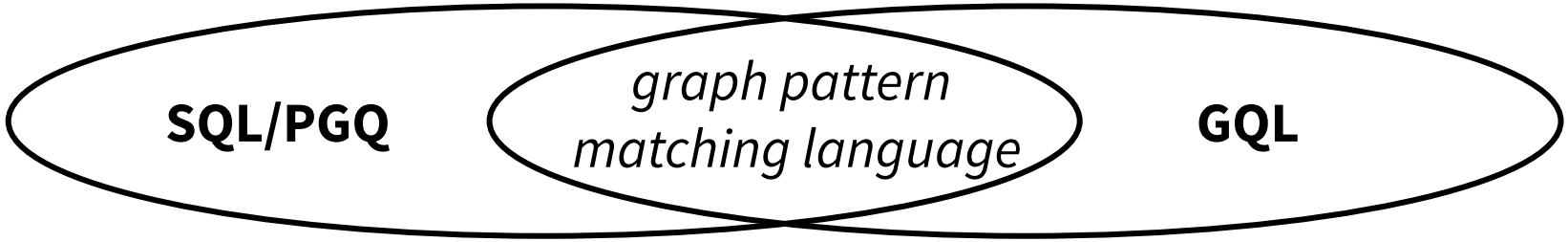
MSBFS (2014)

systems papers (Umbra, Graphflow, DuckPGQ)

Future outlook

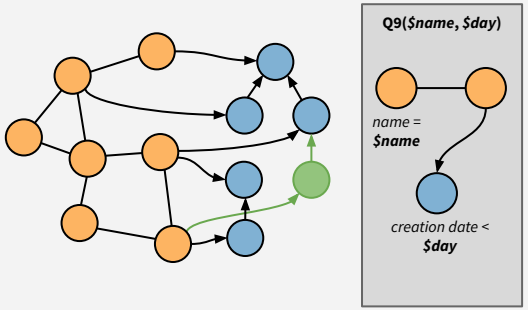
New ISO standard query languages

- **SQL/PGQ** (Property Graph Queries), part of SQL:2023
- **GQL** (Graph Query Language), to be released in 2024

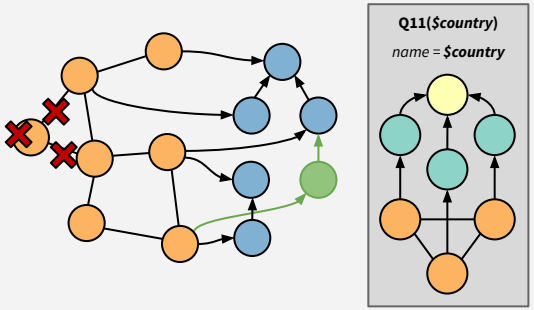


- LDBC has a **liaison with ISO** which allows access to the standard drafts
- Preparing audits of implementations using these languages

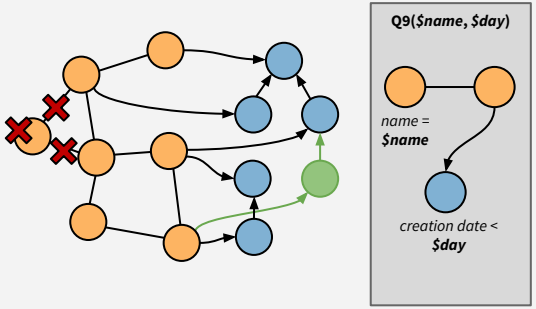
SNB Interactive v1



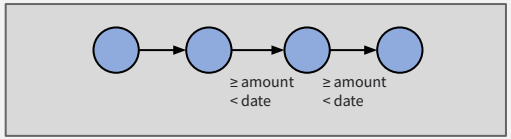
SNB Business Intelligence



SNB Interactive v2



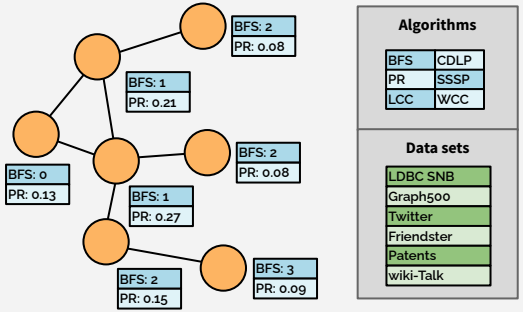
Financial Benchmark



Traversal with truncation

Strict latency bound (P99 < 100 ms)

Graphalytics



Semantic Publishing Benchmark

Target: **RDF/SPARQL**

Domain: **Media/publishing industry**

Inferencing & continuous updates

LDBC

The graph & RDF benchmark reference