The Linked Data Benchmark Council (LDBC):
12 years of fostering competition in the graph processing space

Gábor Szárnyas

GraphSys @ ICPE | 2024-05-11 | London

About me: Gábor Szárnyas

Interests: data management with a focus on modelling, performance, and education

PhD

2014–2019

postdoc

2020–2023

developer relations

2023–

Agenda

- Graph processing
- The need for benchmarks
- LDBC overview
- Technical aspects
- Non-technical aspects

Pain point: more difficult than expected

Graph processing

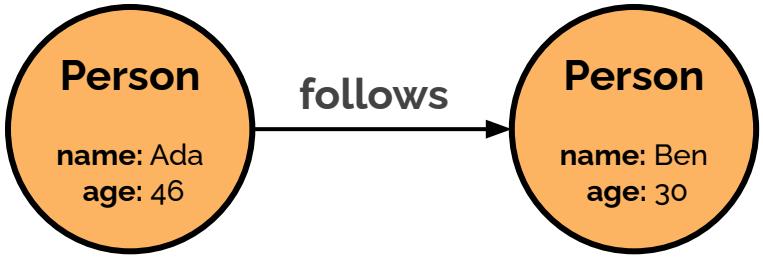
Data modelling: Tabular vs. graph

Person

id	name	age
1	Ada	46
2	Ben	30

follows

person1	person2
1	2



Waves of the “attributed graph” data model

year	data model	declarative language
1969	network model (CODASYL)	no
1988	object-oriented model	no
1999	RDF	SPARQL
2010	property graph	Cypher, Gremlin, …

problem #1:
usually no standard query language

problem #2:
performance limitations

50 years ago, RDBMSs had similar problems

The need for benchmarks

Competition drives performance!

Initially: *benchmark wars* in the 1980s

Objective system-to-system comparison is very difficult

Vendors are motivated to boast good results

Need an independent authority and a standard specification:

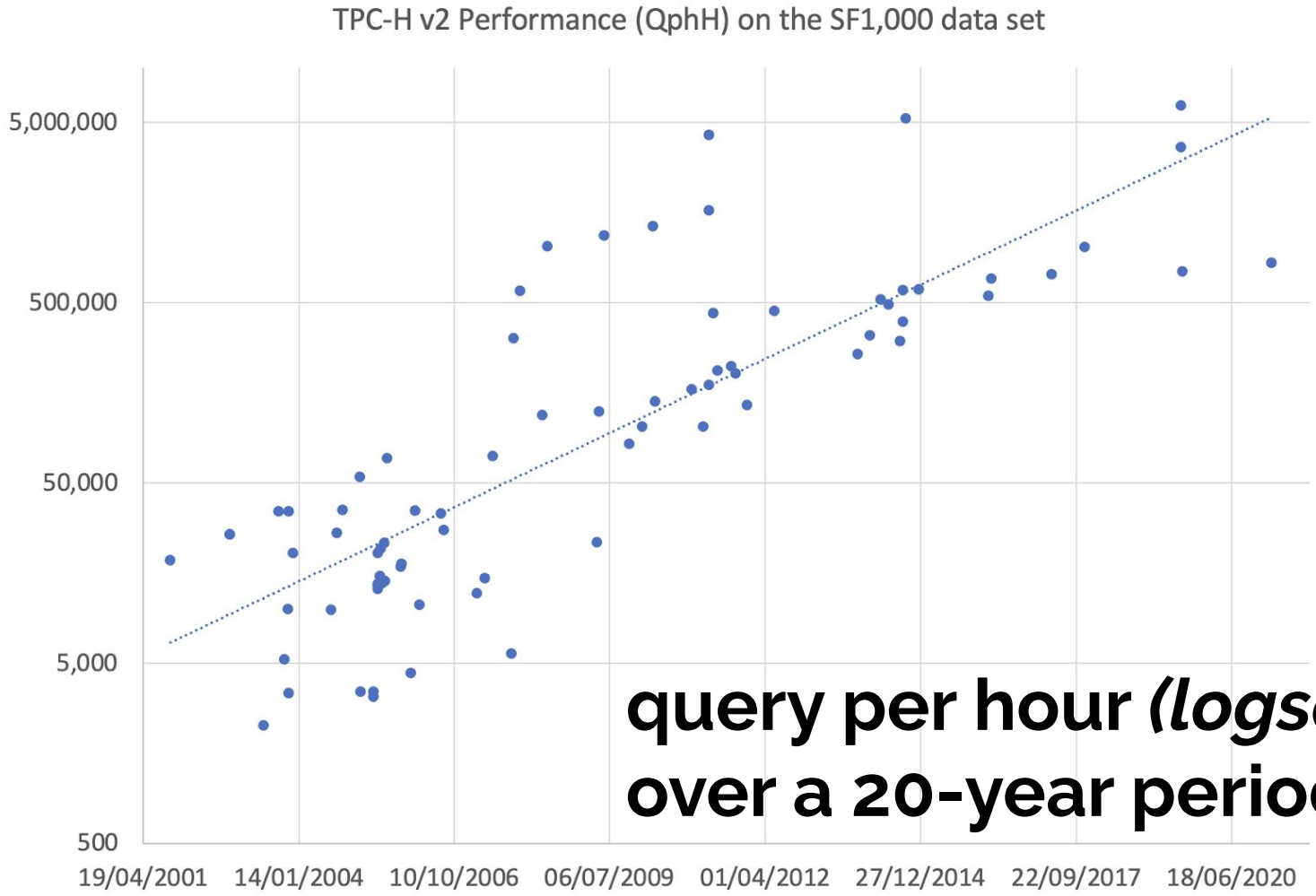
- standard data sets
- a process for verifying results

TPC: Transaction Processing Performance Council

Non-profit founded in 1988

Benchmark specifications with a stringent auditing process

Influential benchmarks: TPC-C, TPC-H, TPC-DS



query per hour *(logscale)*
over a 20-year period

LDBC

LDBC: Linked Data Benchmark Council

Non-profit company

Mission: Accelerate progress in graph data management

Method: Design graph benchmarks and govern their use

Foster collaboration between researchers & practitioners

ldbcouncil.org

github.com/ldbc

Sponsor Companies

ANT GROUP

火山引擎 Beijing Volcano (ByteDance)

ORACLE Labs

Companies and Research Institutes

Alibaba Cloud

aws

intel

neo4j

TigerGraph

JCC Consulting, Inc.

MEM GRAPH

ontotext

fabarta

*Sparsity

创邻科技 CREATE LINK

pometry

ULTIPA

ArangoDB

海致星图 StarGraph

NebulaGraph

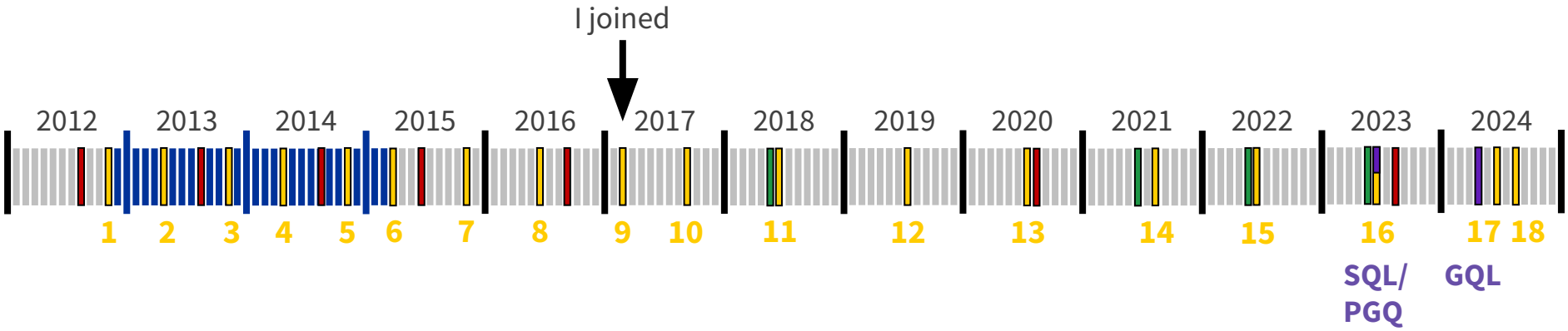
RelationalAI

thatDot

Birkbeck UNIVERSITY OF LONDON

FORTH

LDBC timeline



EU FP7 project

Benchmark papers

Schema/language papers

Technical User Community meeting

ISO standards

LDBC benchmarks

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

Similarities to TPC benchmarks

macro/application-level benchmarks	scale factors: SF30 = 30GiB CSV	flexible hardware and software setup
third-party auditors	FDRs with metrics, e.g. throughput@SF	benchmark approval and renewal

TPC-inspired choke points

A choke point is a **difficult aspect of query processing** with a significant performance impact

The TPCTC'12 paper analyzed TPC-H based on the lessons learnt when implementing the benchmark on Vectorwise, Virtuoso, and HyPer

Examples:

- Join ordering
- Efficient antijoins and outer joins
- Handling paths

P. Boncz, T. Neumann, O. Erling:
TPC-H analyzed: Hidden messages and lessons learned from an influential benchmark. TPCTC'12

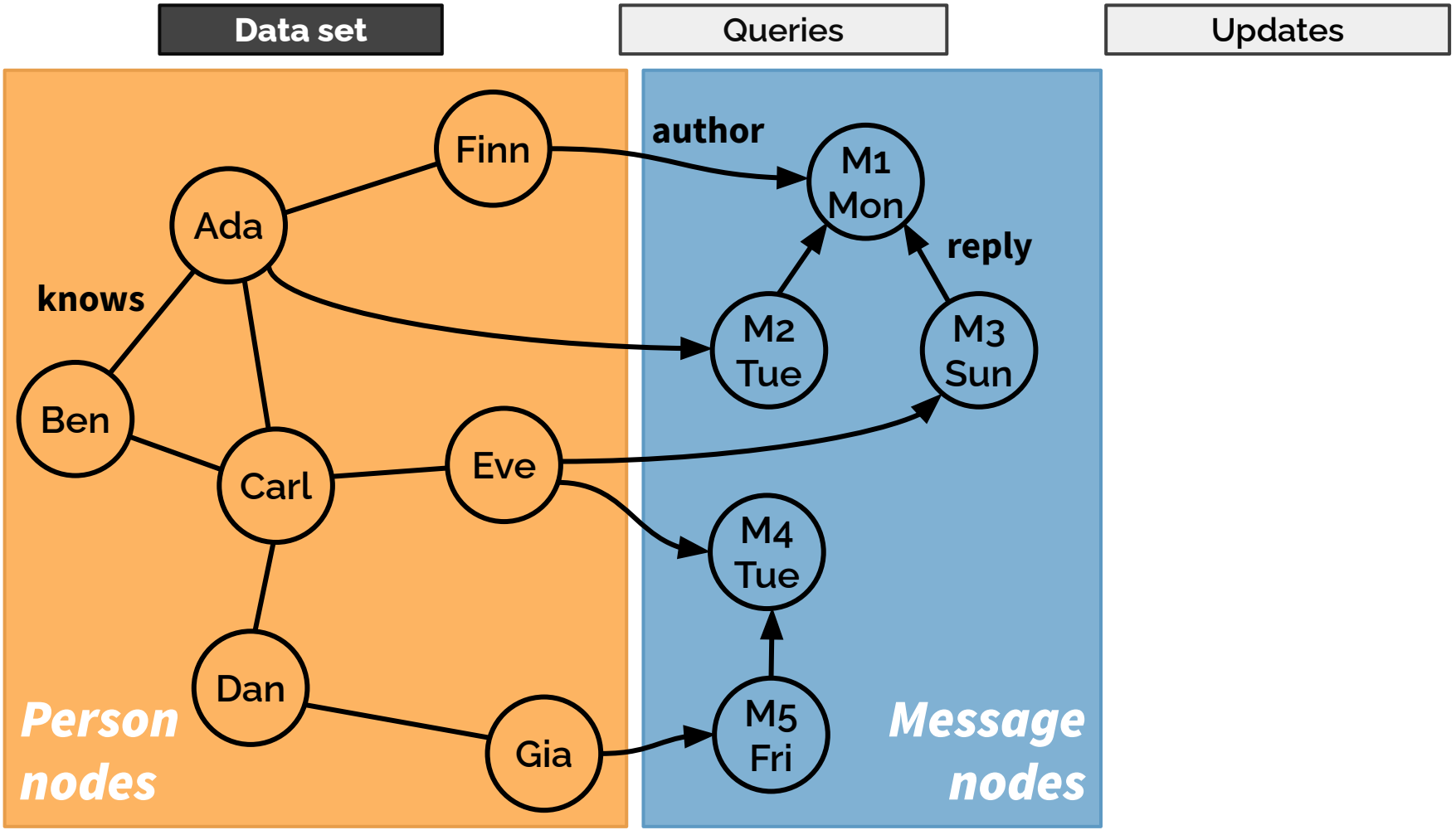
The Social Network Benchmark (SNB) suite

Data set and queries

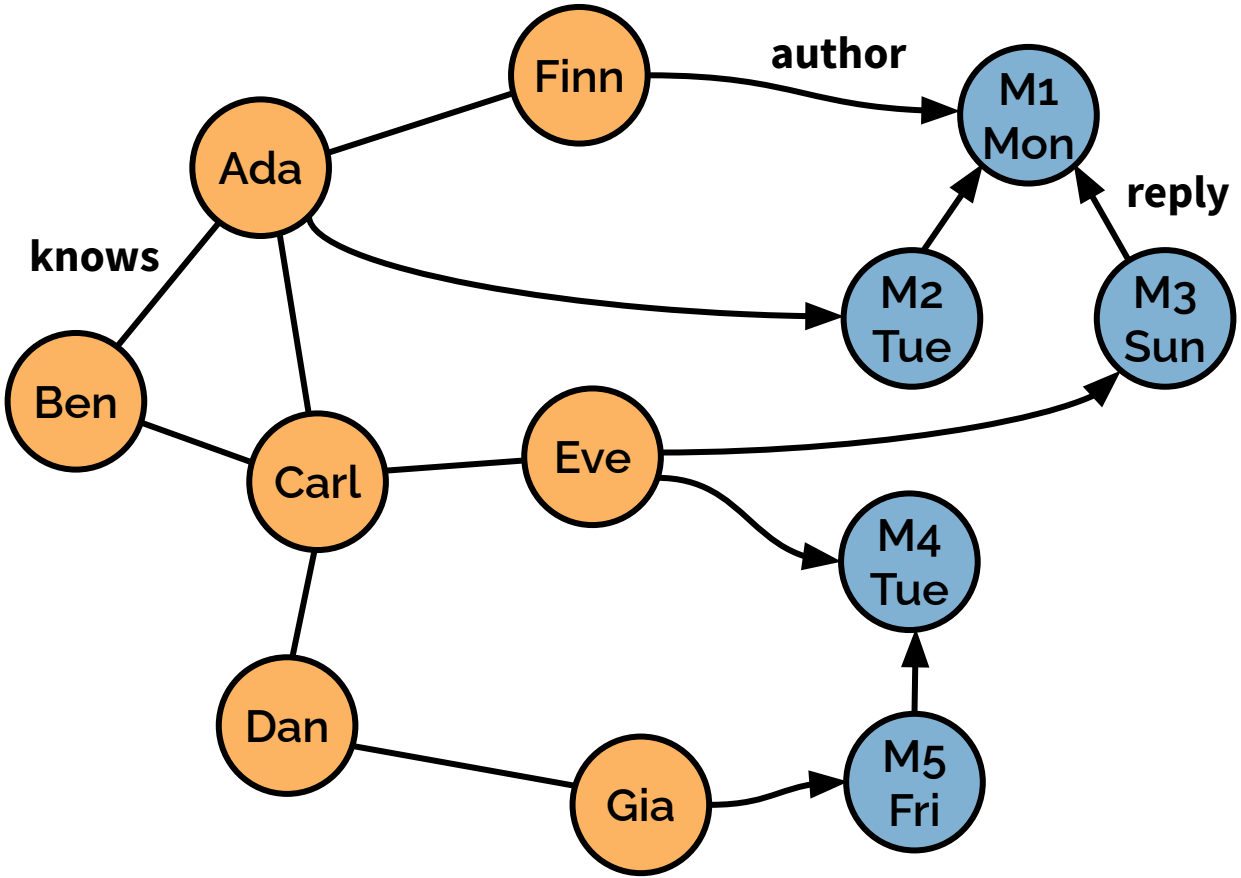
Data set

Queries

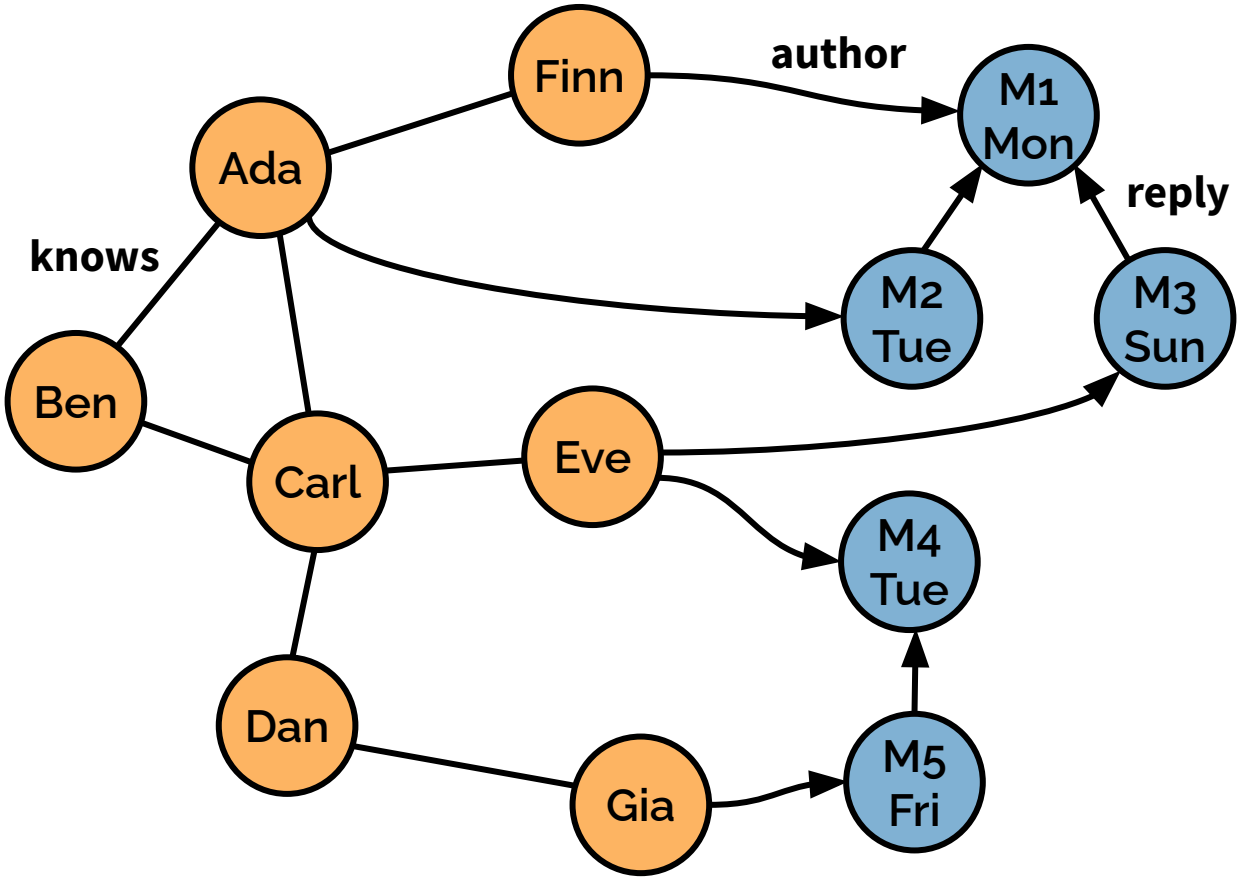
Updates


Data set
Queries
Updates
Finn
author
M1
Mon
Ada
reply
knows
M2
Tue
M3
Sun
Ben
Carl
Eve
M4
Tue
Dan
Gia
M5
Fri
Person
nodes
Message
nodes

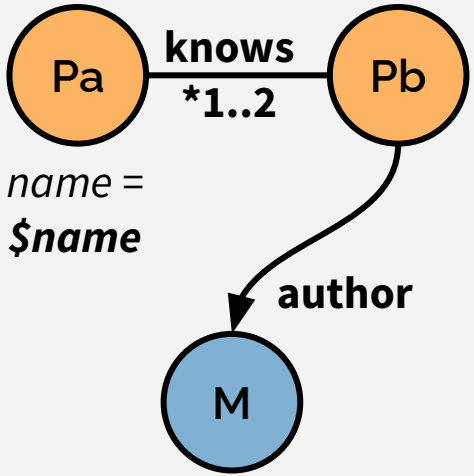
Data set | **Queries** | Updates



Data set | **Queries** | Updates



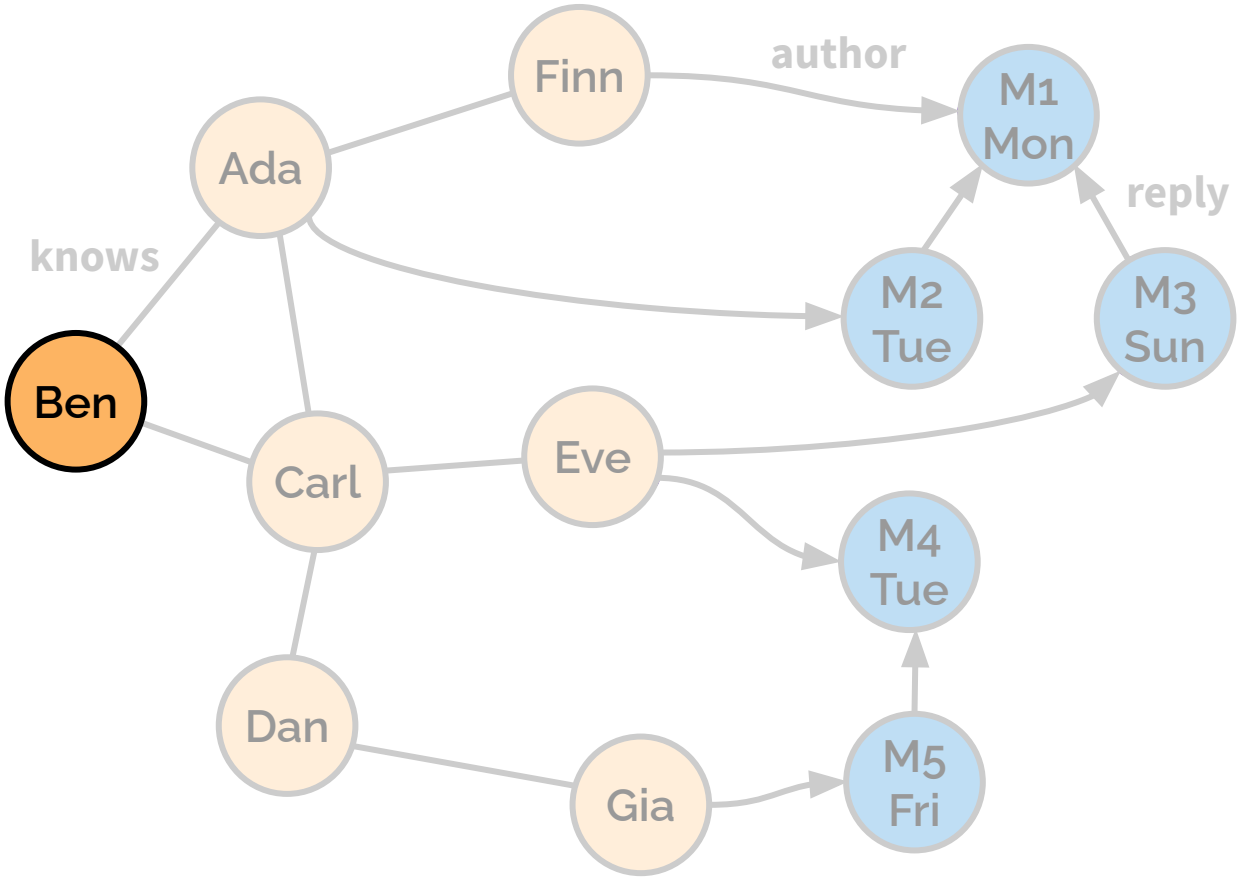
Q9(*$name*, *$day*)



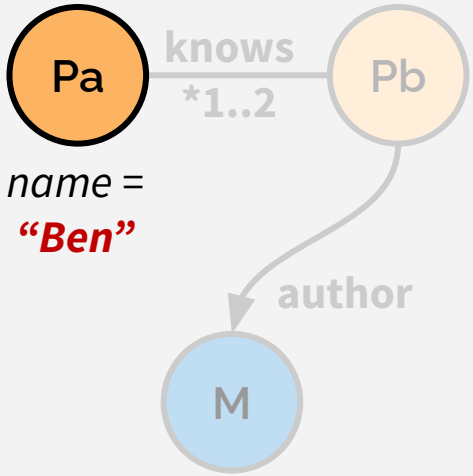
Data set

Queries

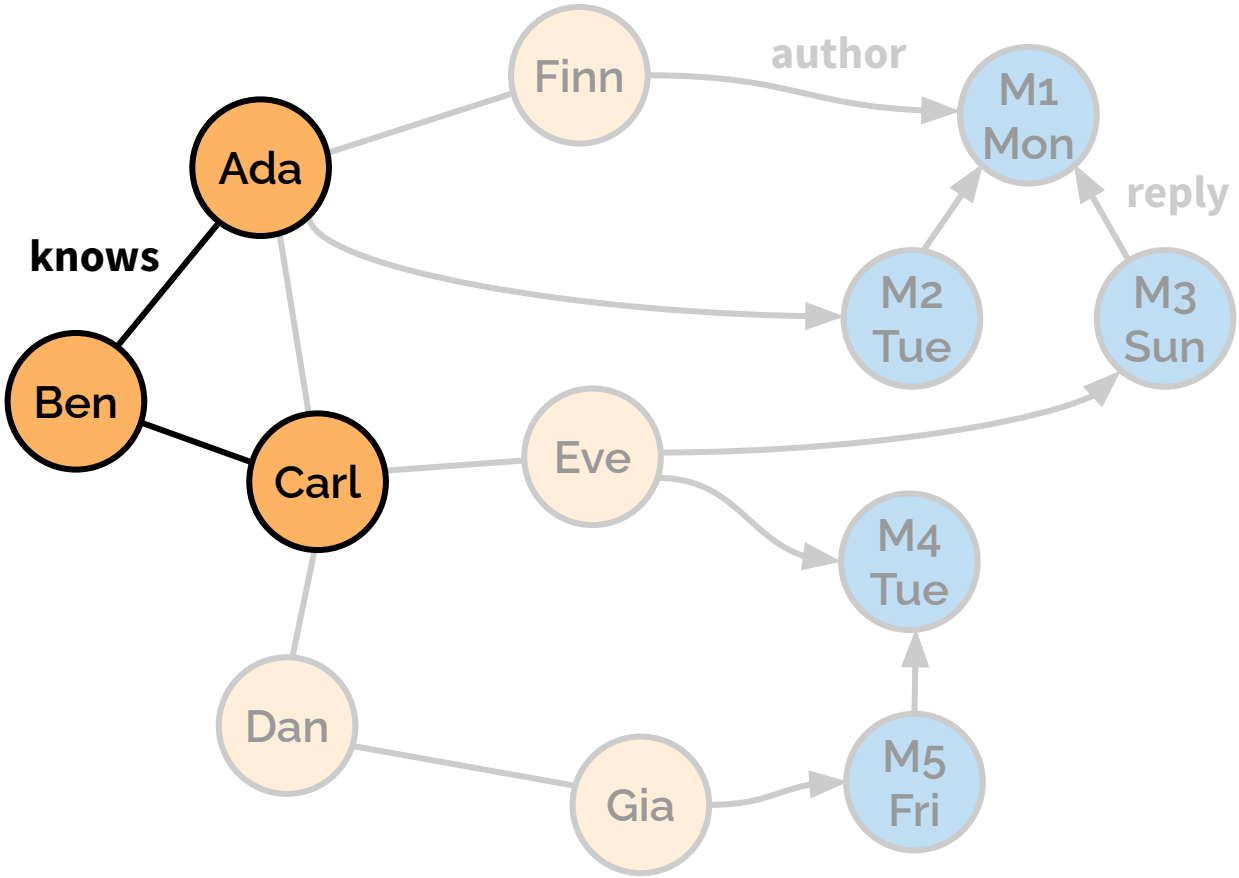
Updates



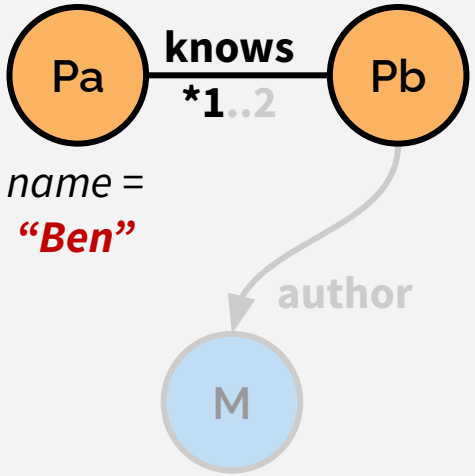
Q9(*"Ben"*, *"Sat"*)



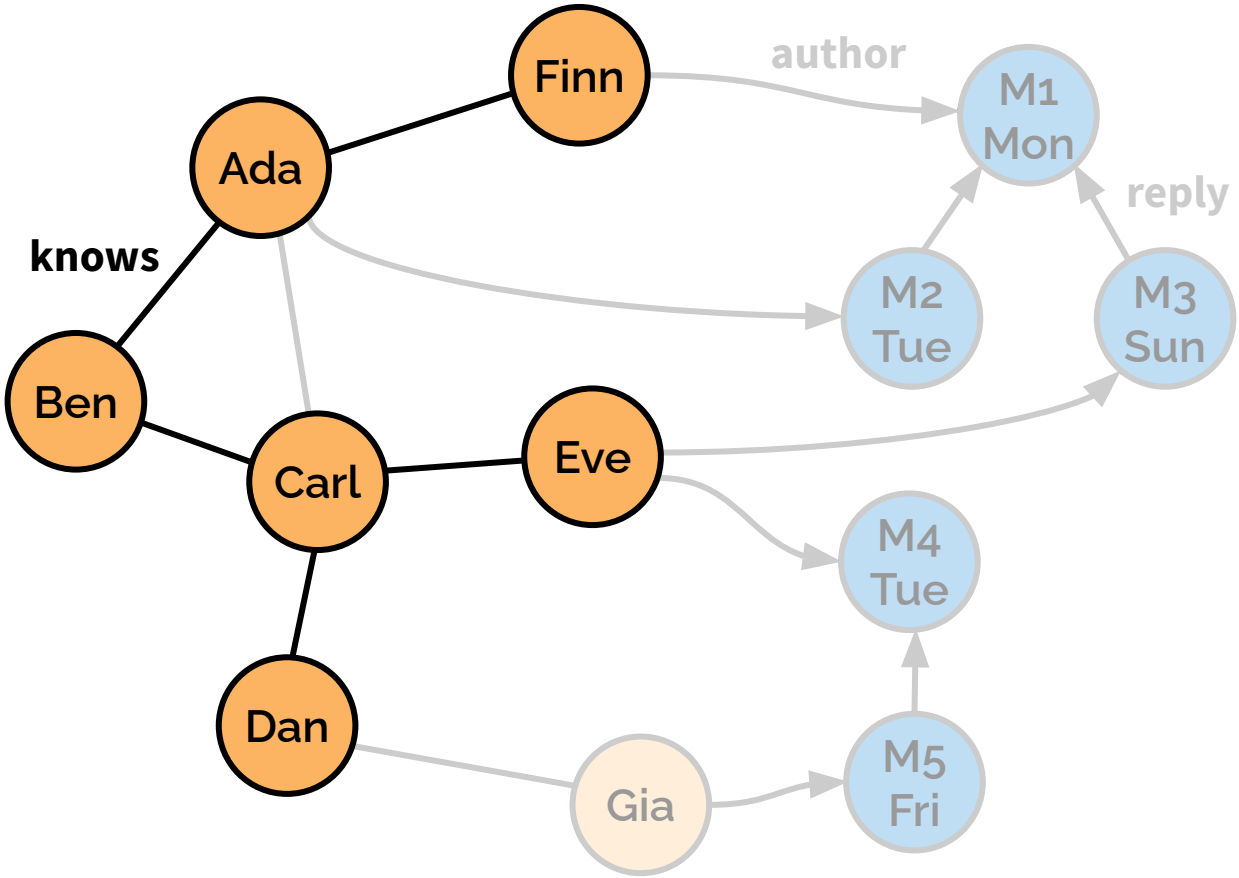
Data set | **Queries** | Updates



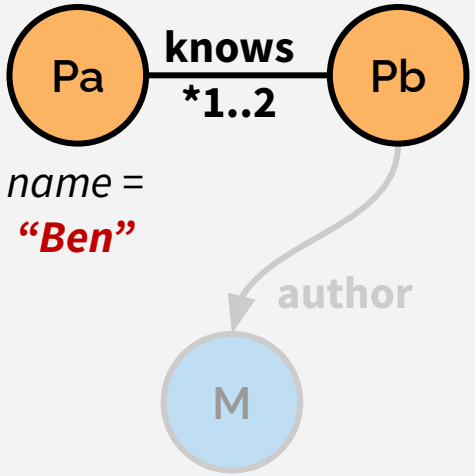
Q9(*"Ben"*, *"Sat"*)



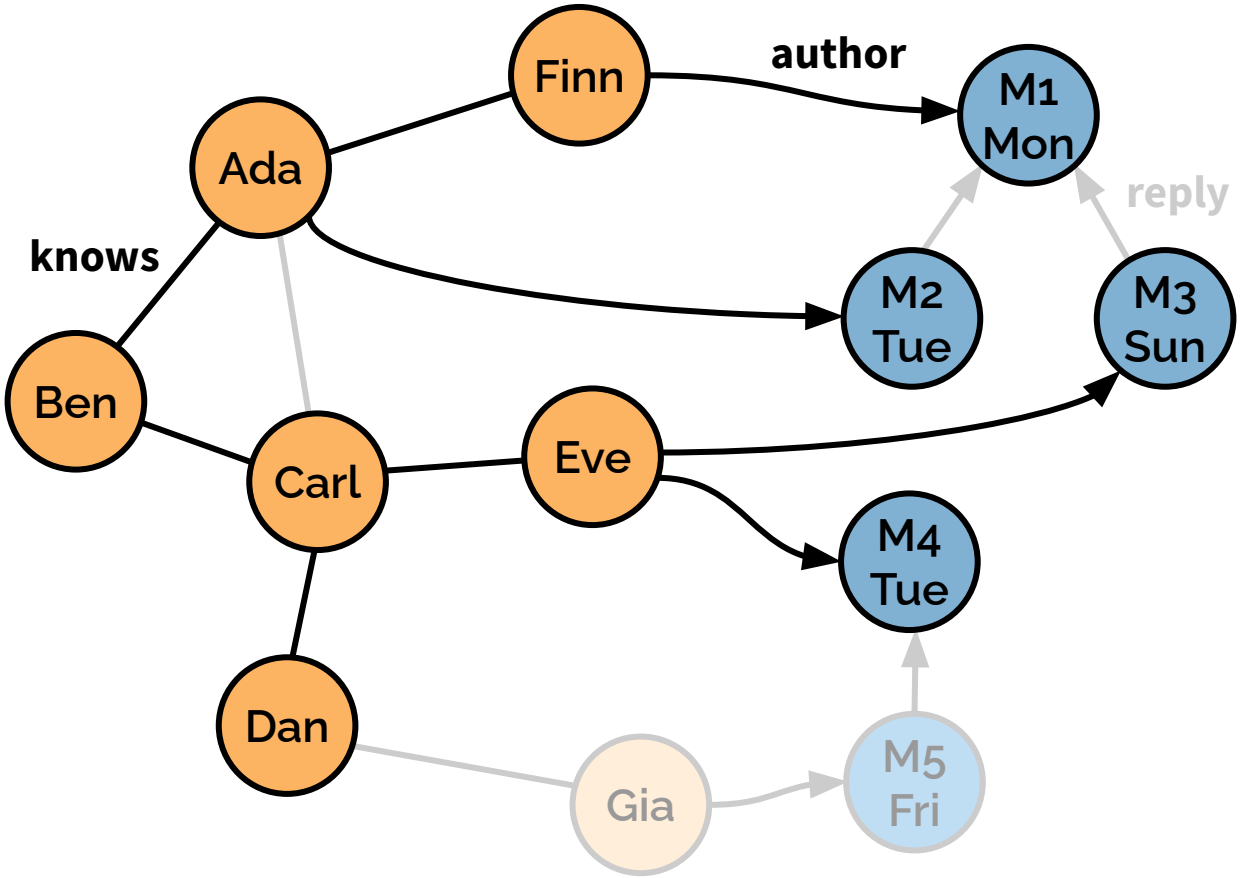
Data set | **Queries** | Updates



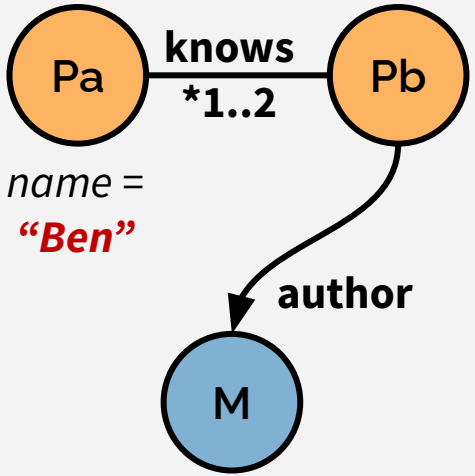
Q9(*"Ben"*, *"Sat"*)



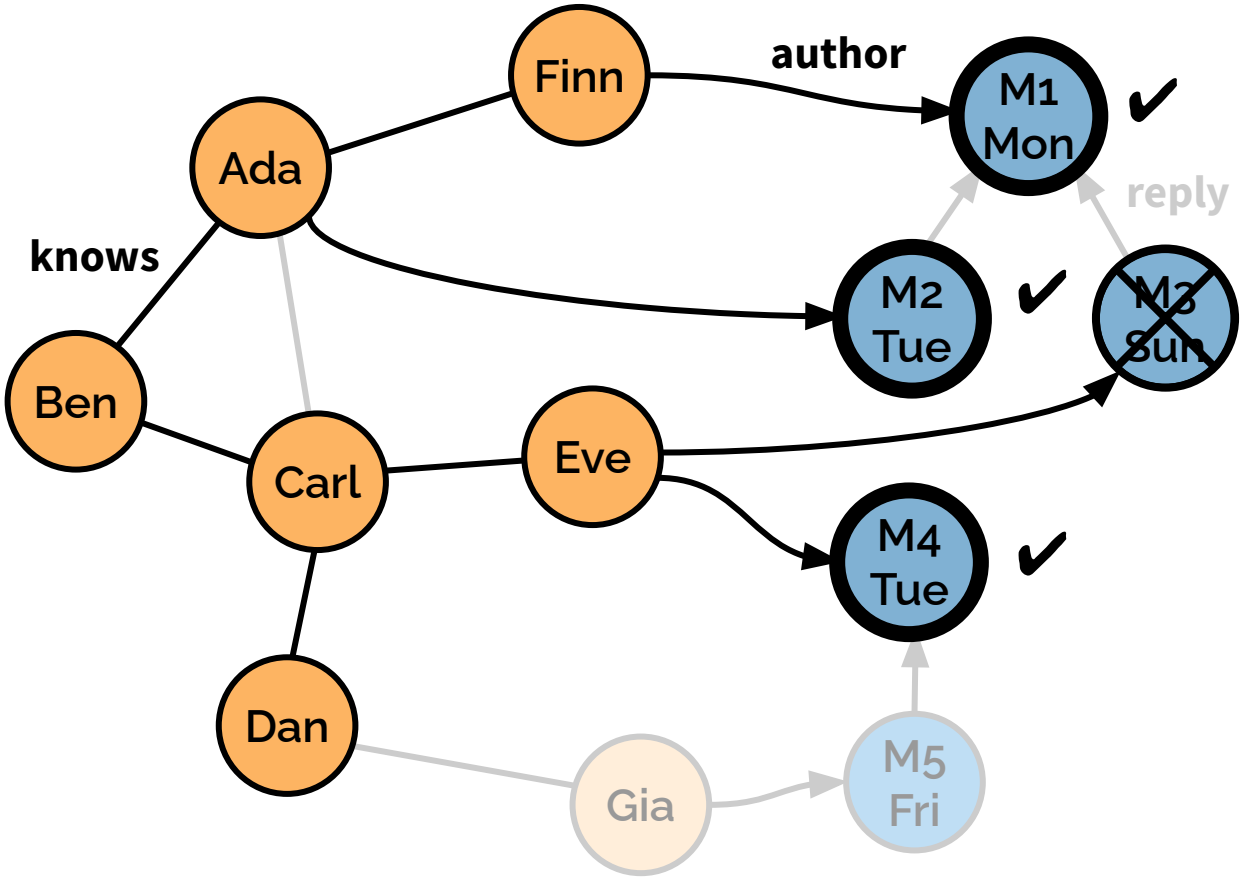
Data set | **Queries** | Updates



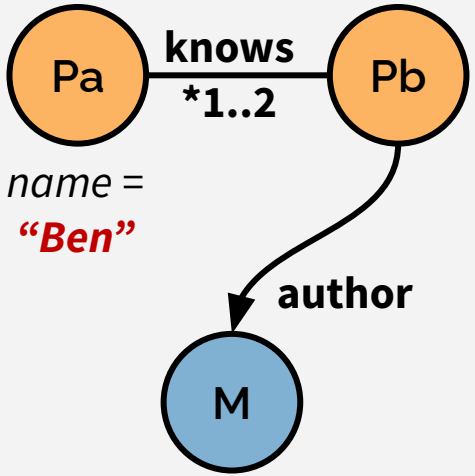
Q9(*"Ben"*, *"Sat"*)



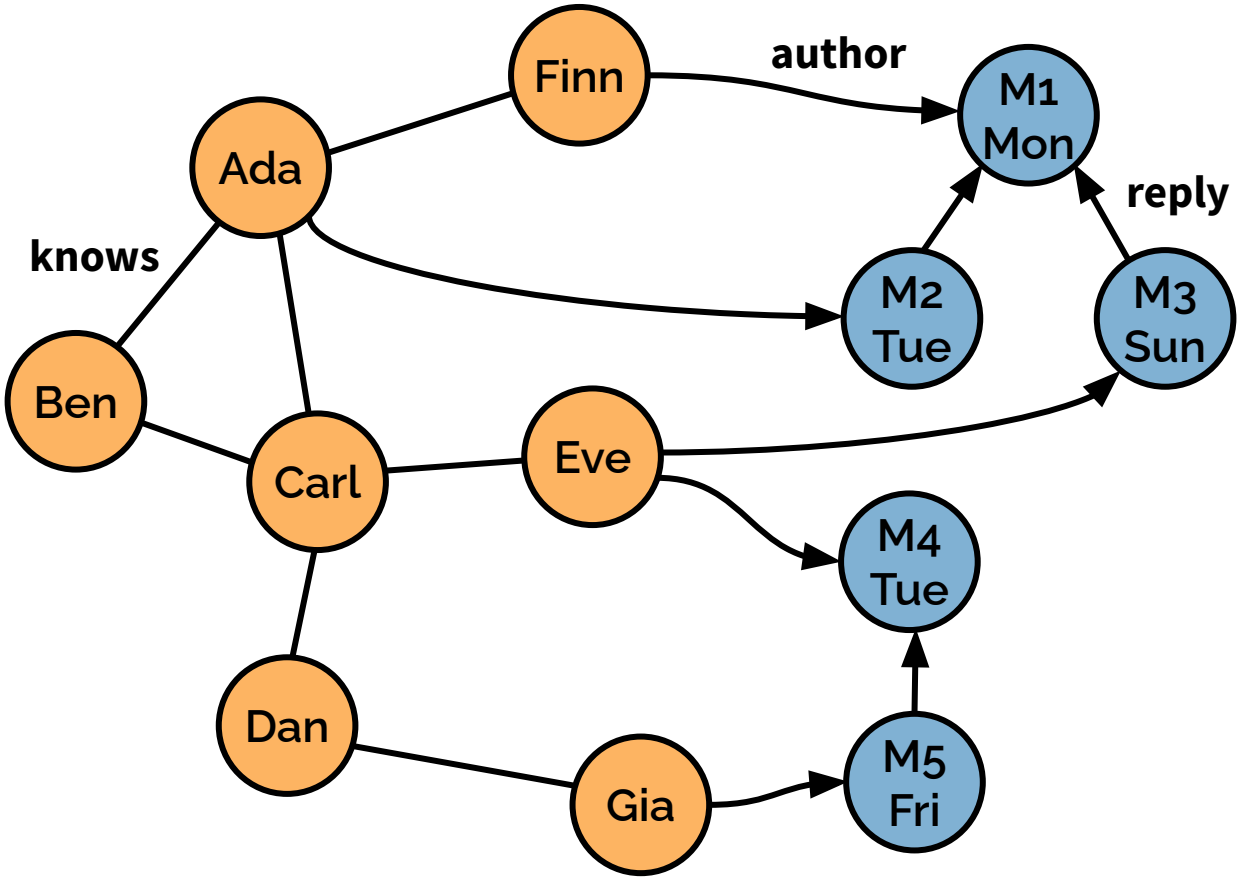
Data set | **Queries** | Updates



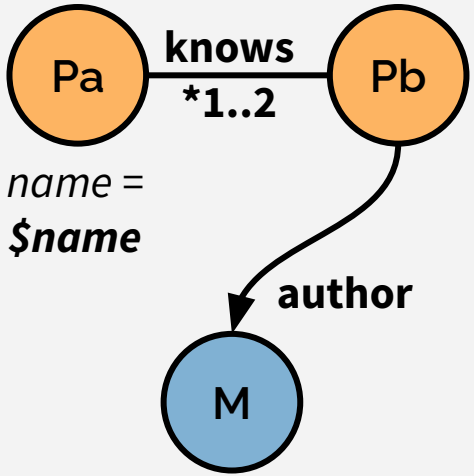
Q9(*"Ben"*, *"Sat"*)



Data set | **Queries** | Updates



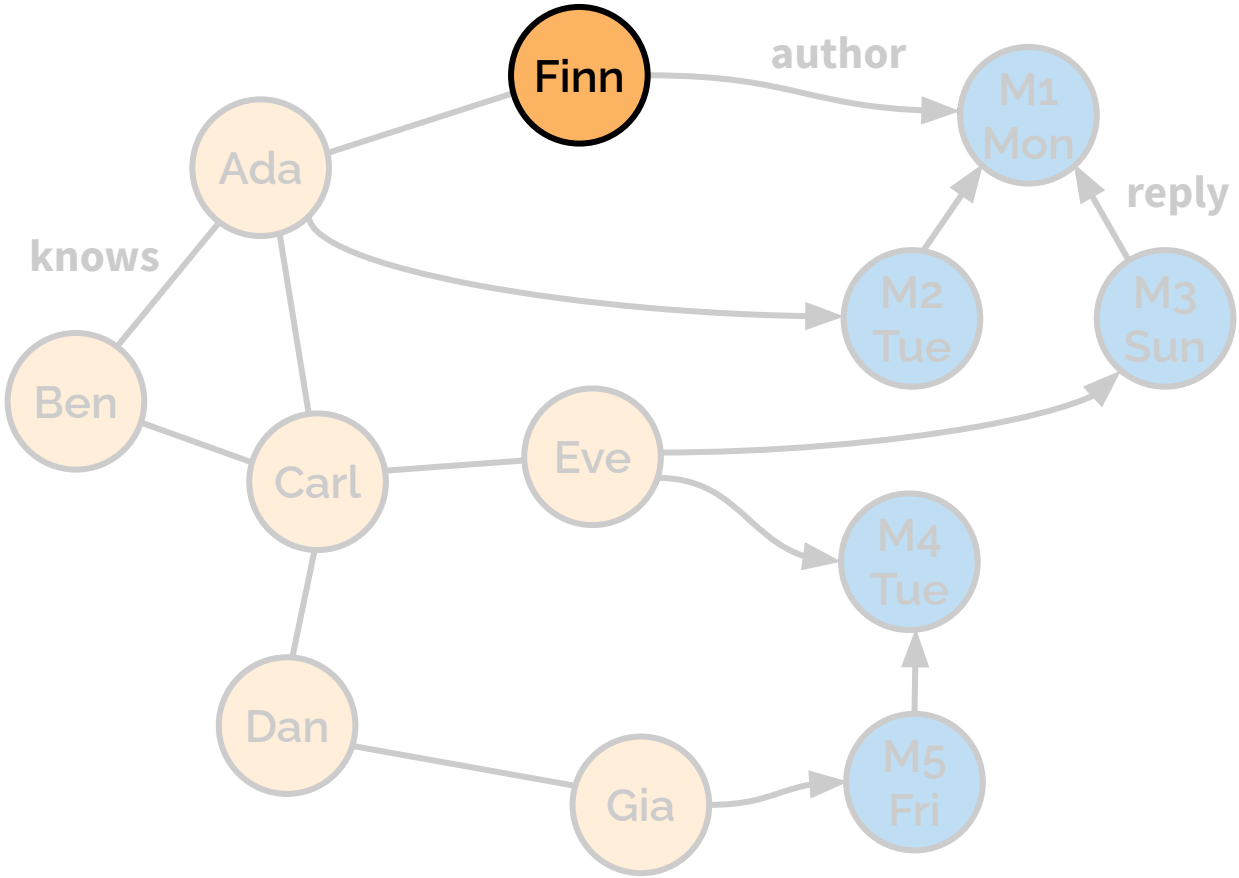
Q9(*$name*, *$day*)



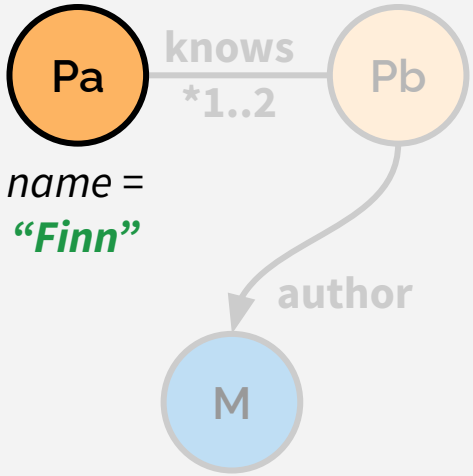
Data set

Queries

Updates



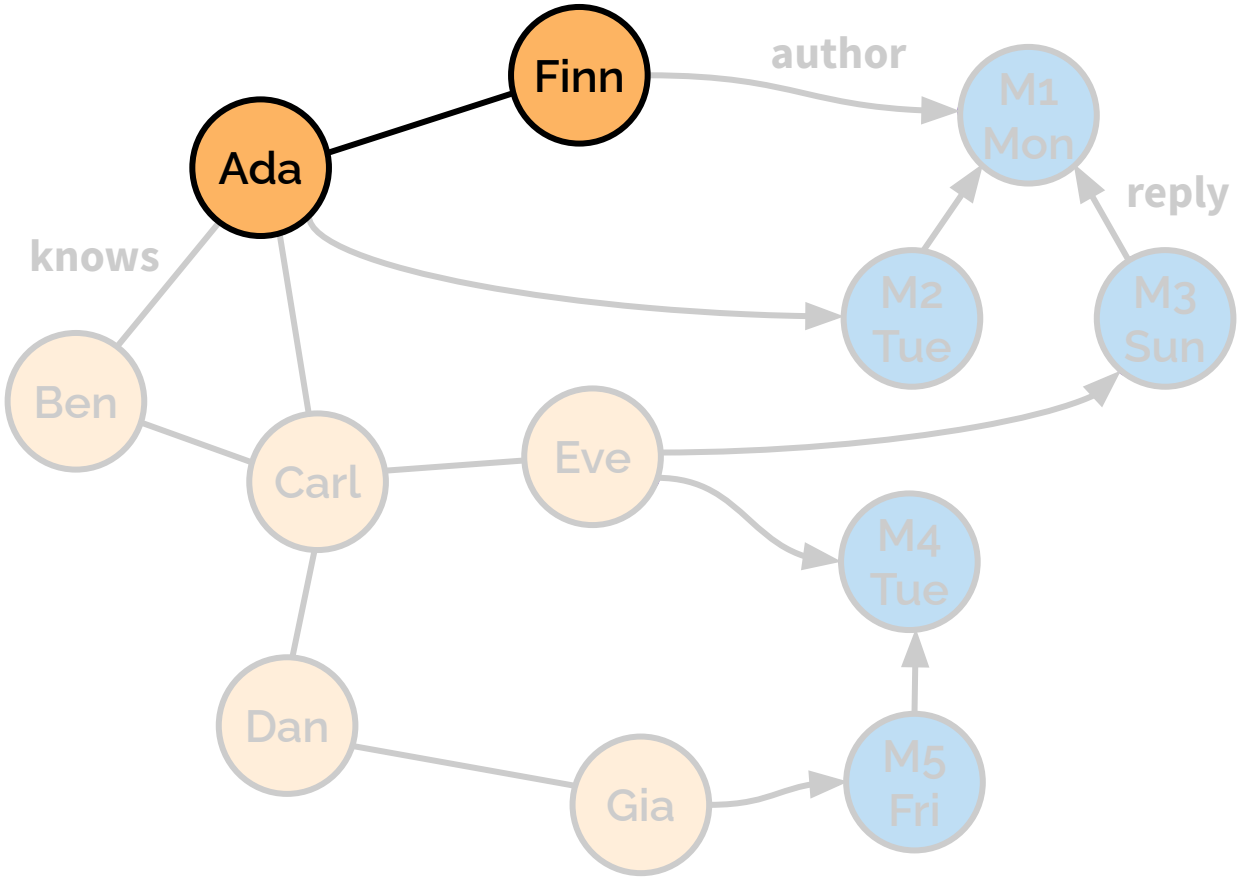
Q9(*"Finn"*, *"Wed"*)



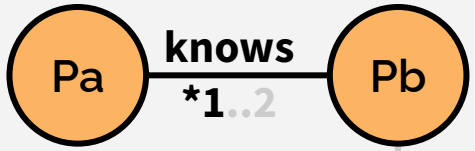
Data set

Queries

Updates



Q9(*"Finn"*, *"Wed"*)



name =
"Finn"

author

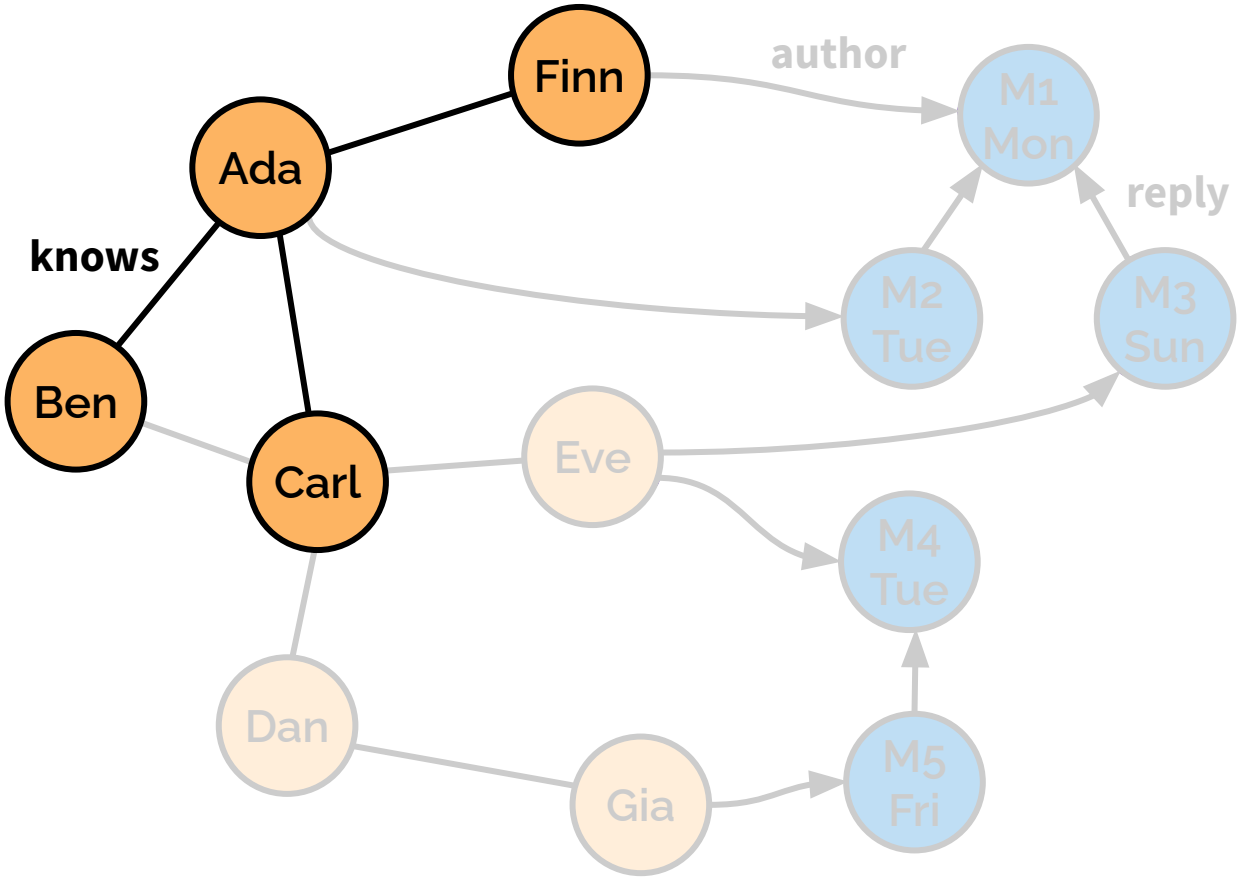
M

*creation date < **"Wed"***

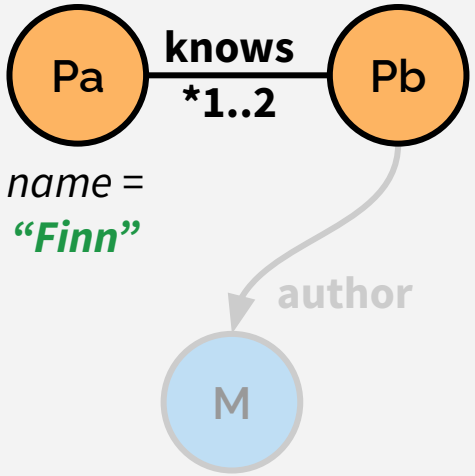
Data set

Queries

Updates



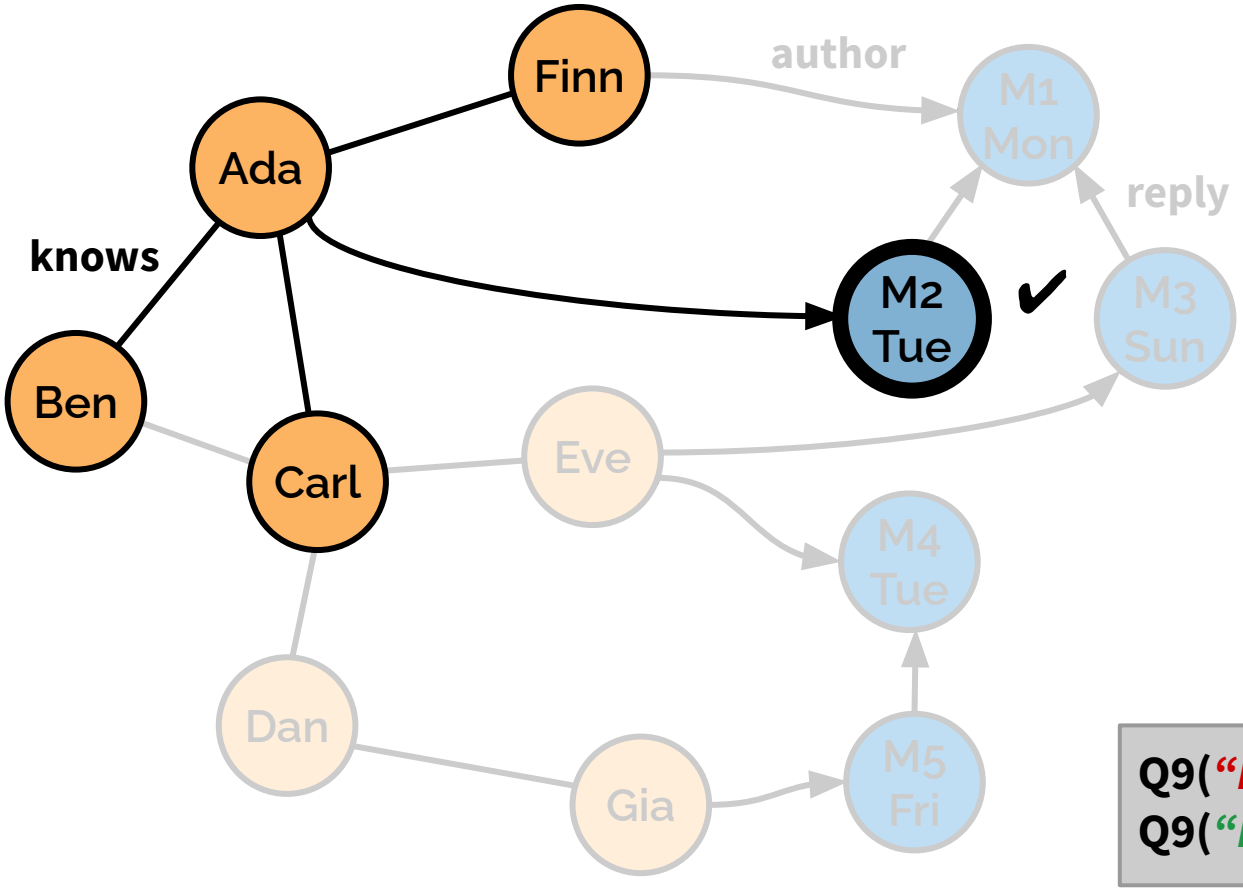
Q9(*"Finn"*, *"Wed"*)



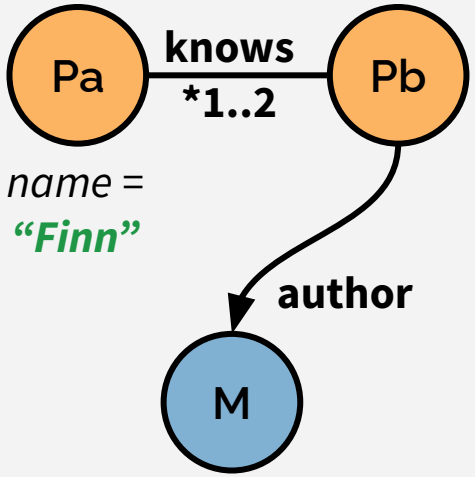
Data set

Queries

Updates



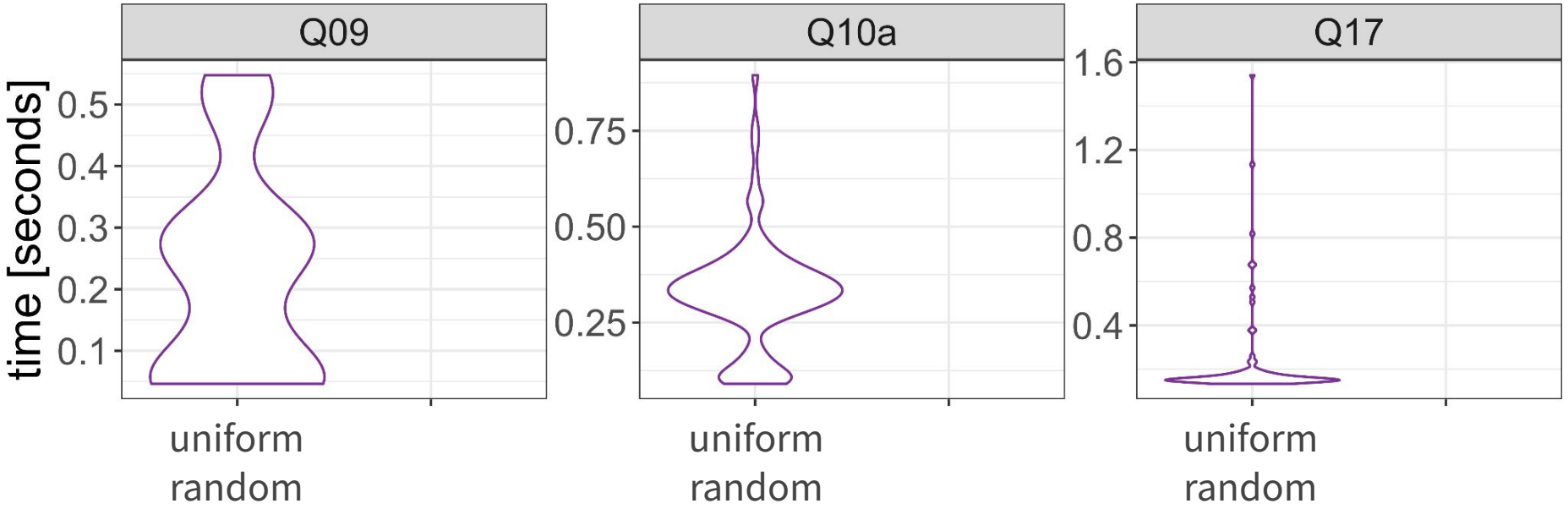
Q9(*"Finn"*, *"Wed"*)



Q9(*"Ben"*, *"Sat"*): 10 nodes
Q9(*"Finn"*, *"Wed"*): 5 nodes

Parameter selection

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

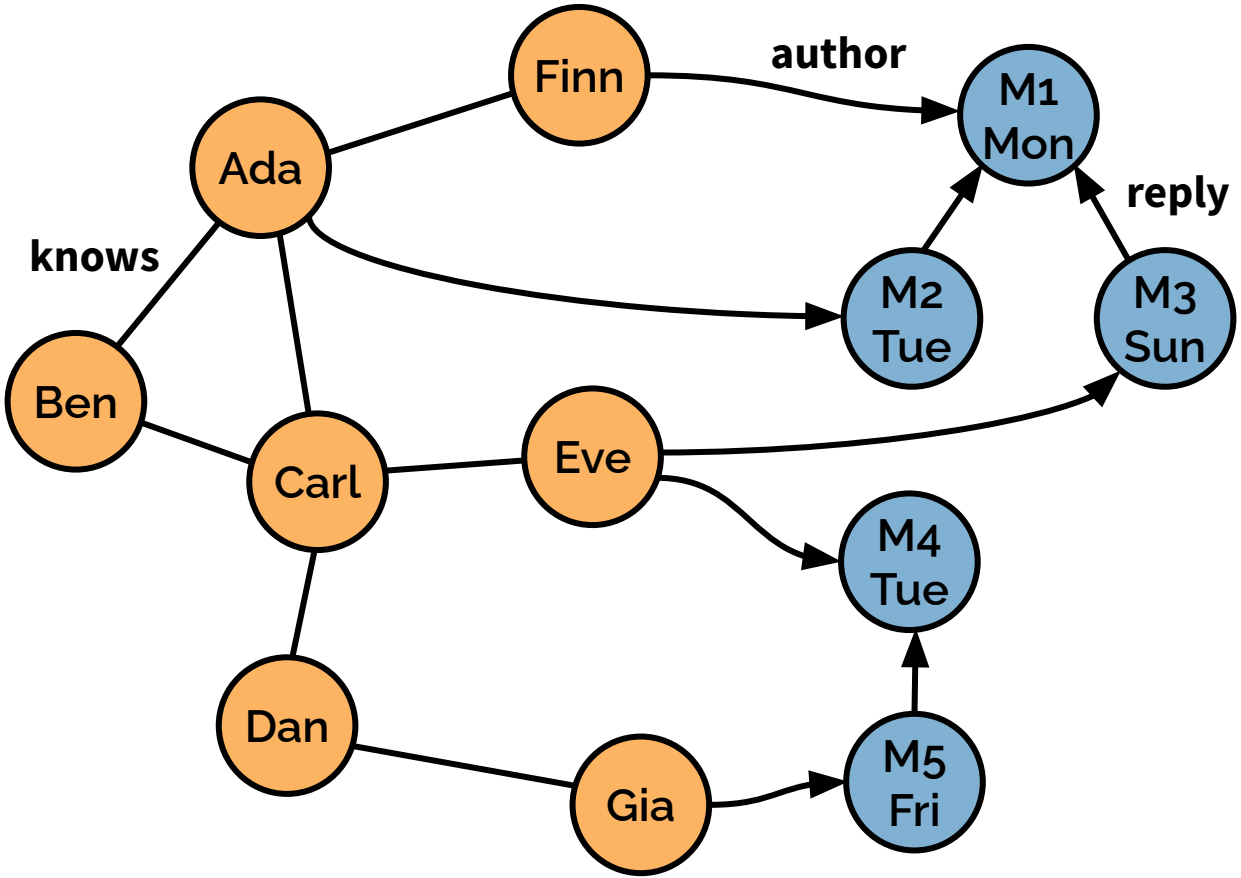


Parameter curation

A. Gubichev, P. Boncz

TPCTC 2014

Data set | **Queries** | Updates



Statistics ("factors")

numFriendsOfFriends

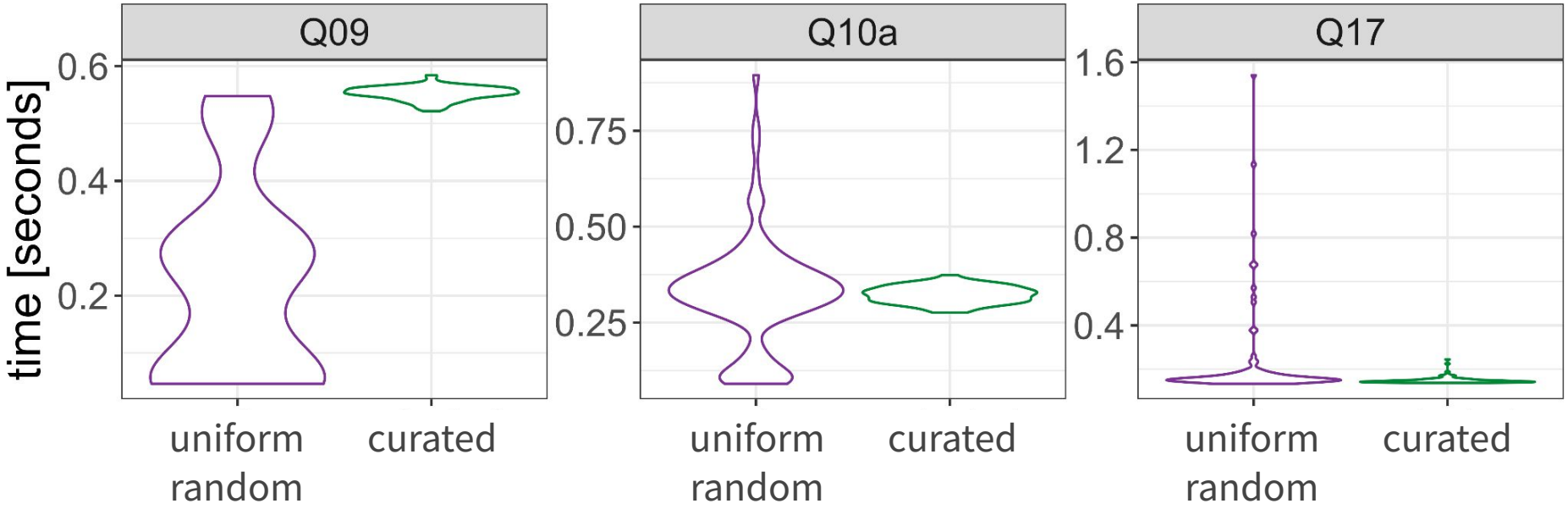
name	#1-hop	#2-hop
Ben	2	3
Carl	4	2
Ada	3	2
…		

numMessagesPerDay

day	#
Mon	1
Tue	2
…	

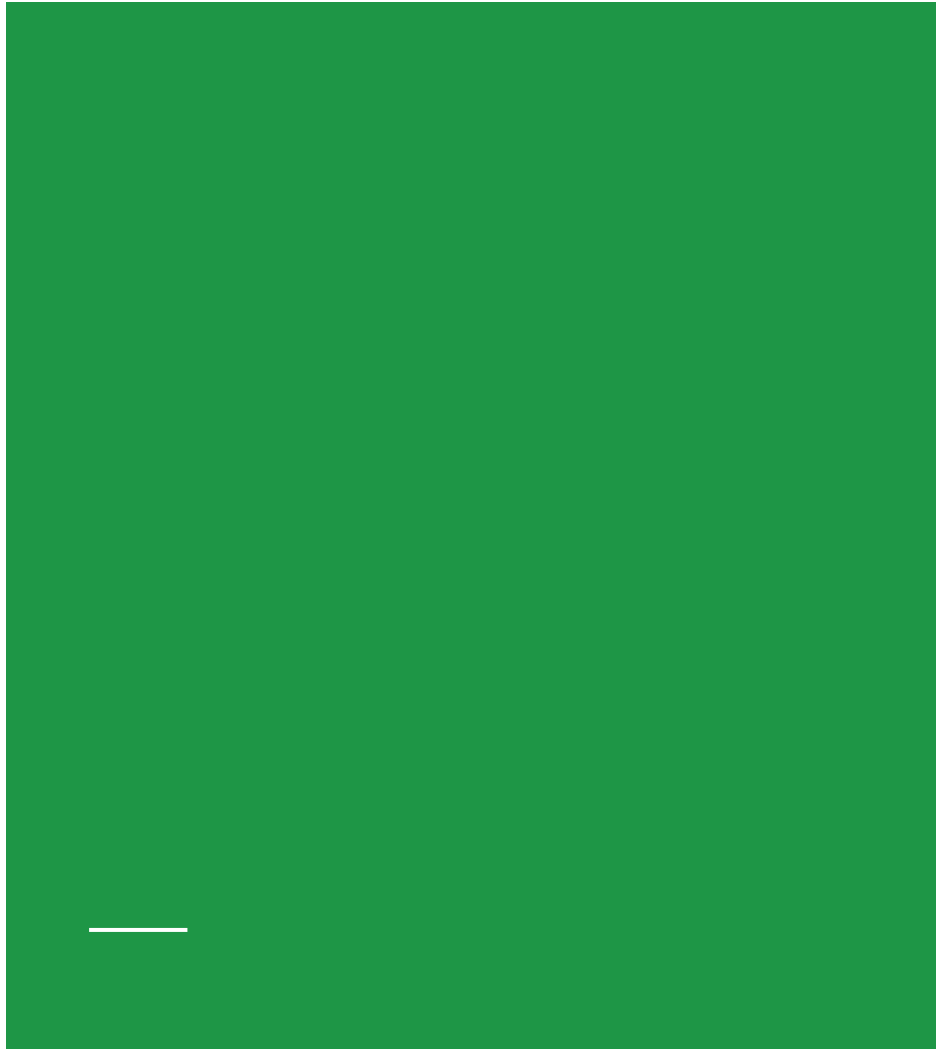
Parameter selection

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

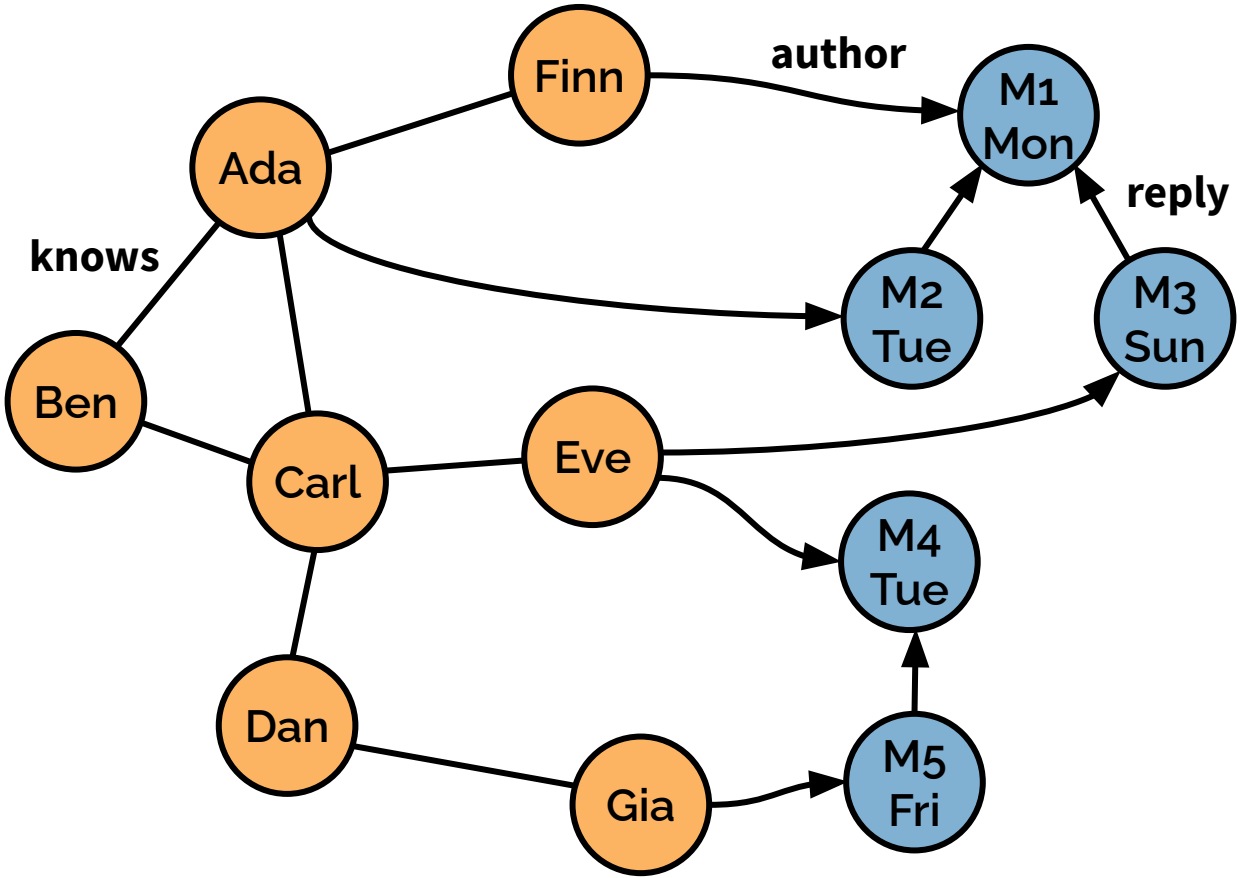


Updates

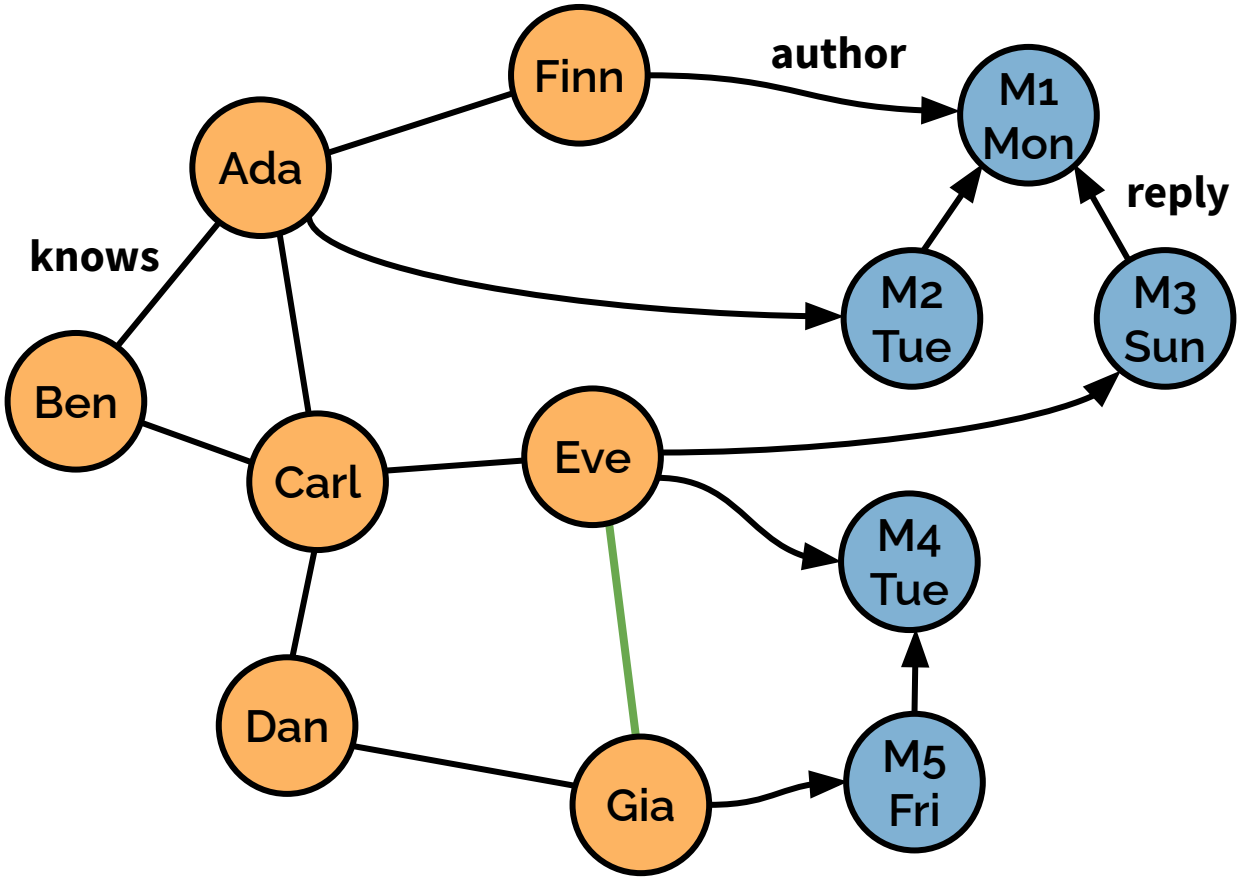
Inserts and deletes



Data set | Queries | **Updates**



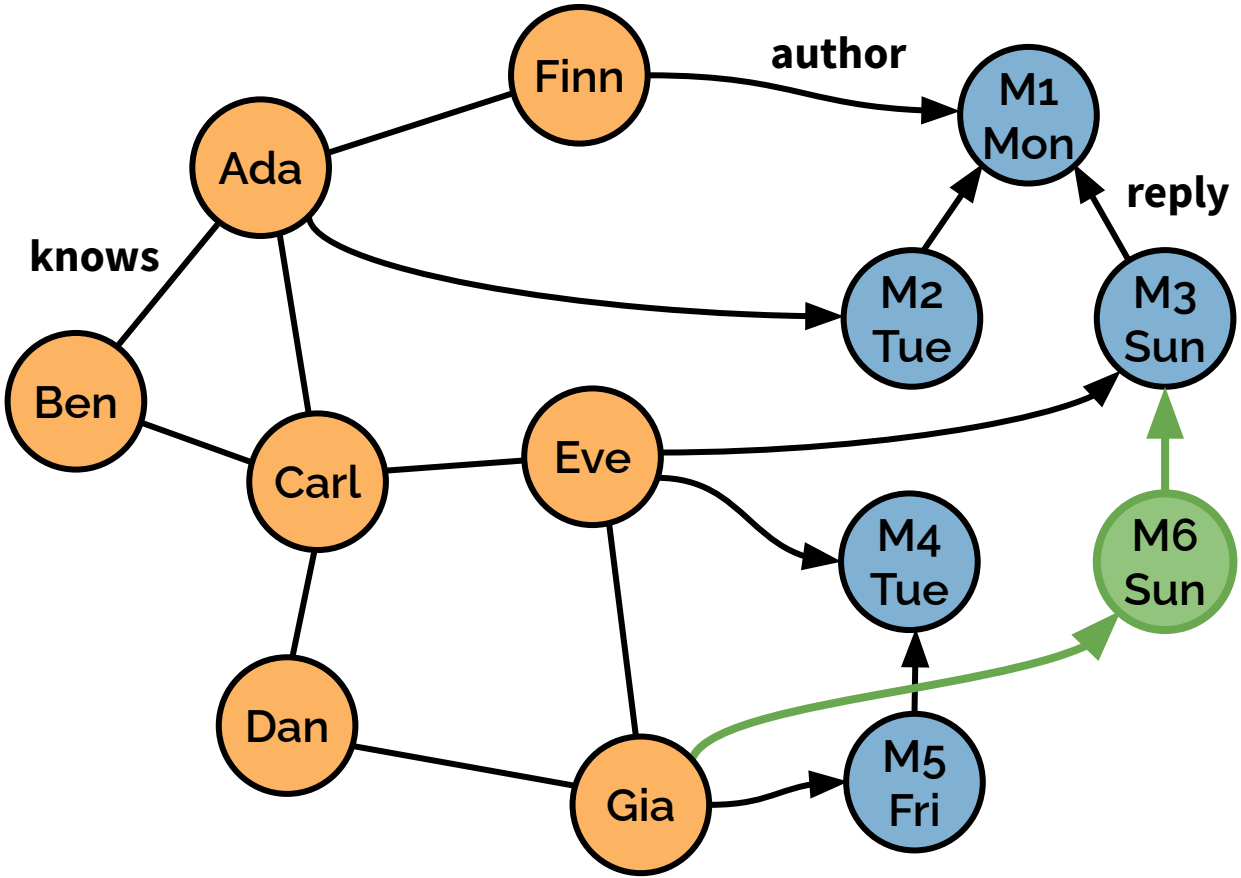
Data set | Queries | **Updates**



Updates

+ knows(“Eve”, “Gia”)

Data set | Queries | **Updates**

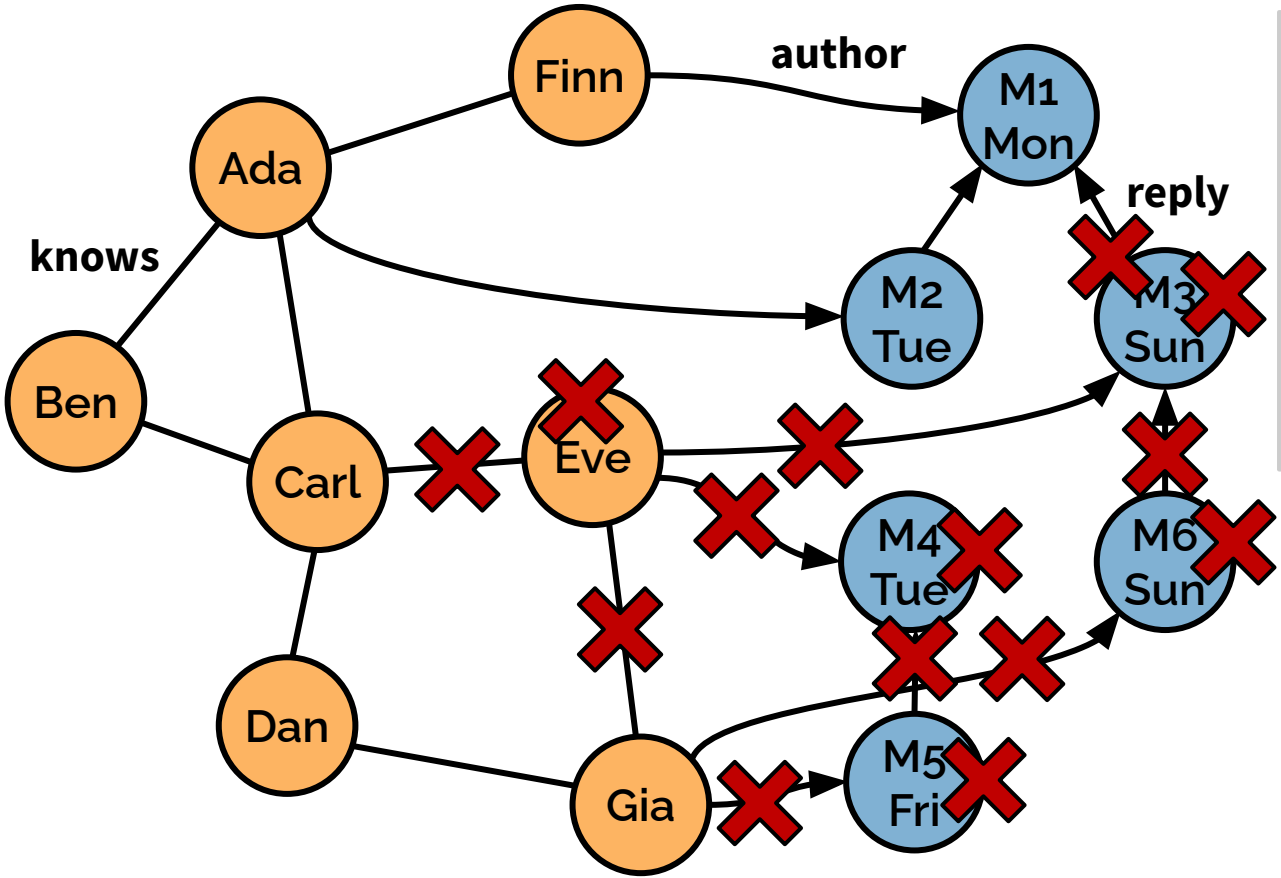


Updates

+ knows("Eve", "Gia")

+ Comment("Gia", "M3")

Data set | Queries | **Updates**



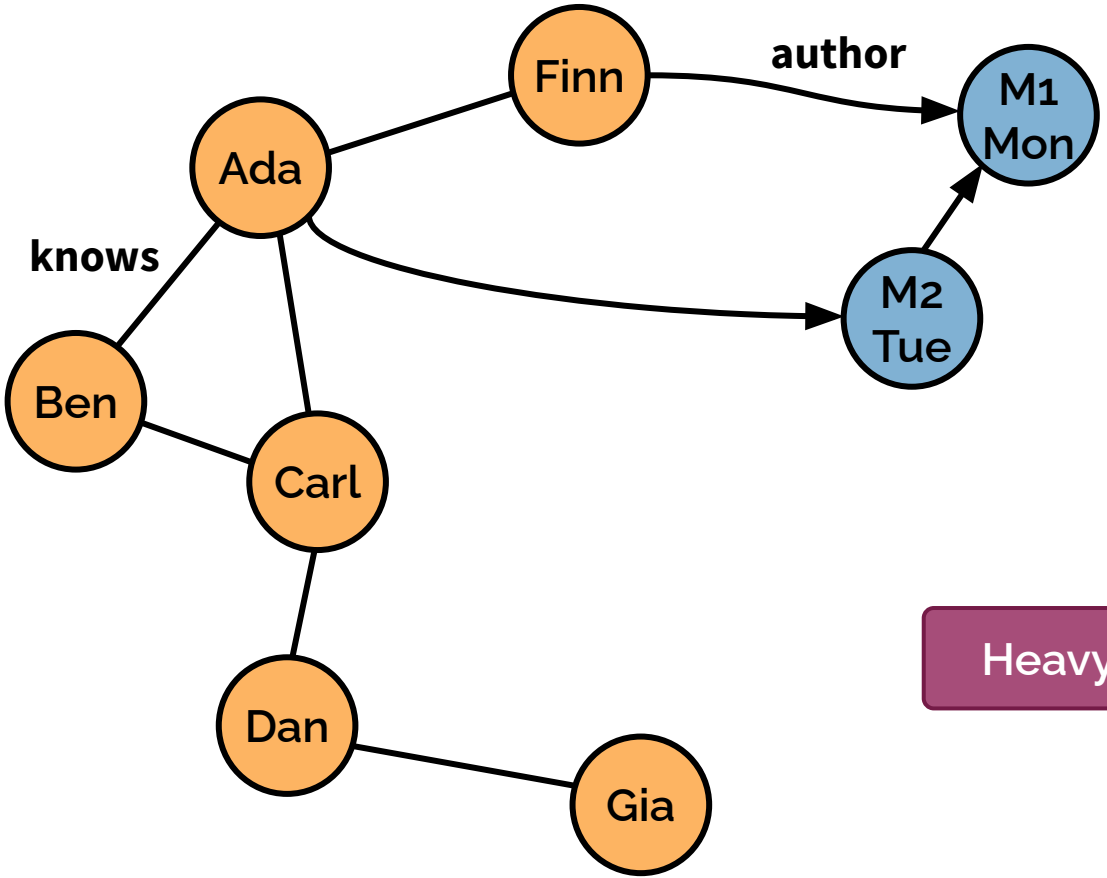
Updates

+ knows("Eve", "Gia")

+ Comment("Gia", "M3")

– Person("Eve")

Data set | Queries | **Updates**



Updates

+ knows(“Eve”, “Gia”)

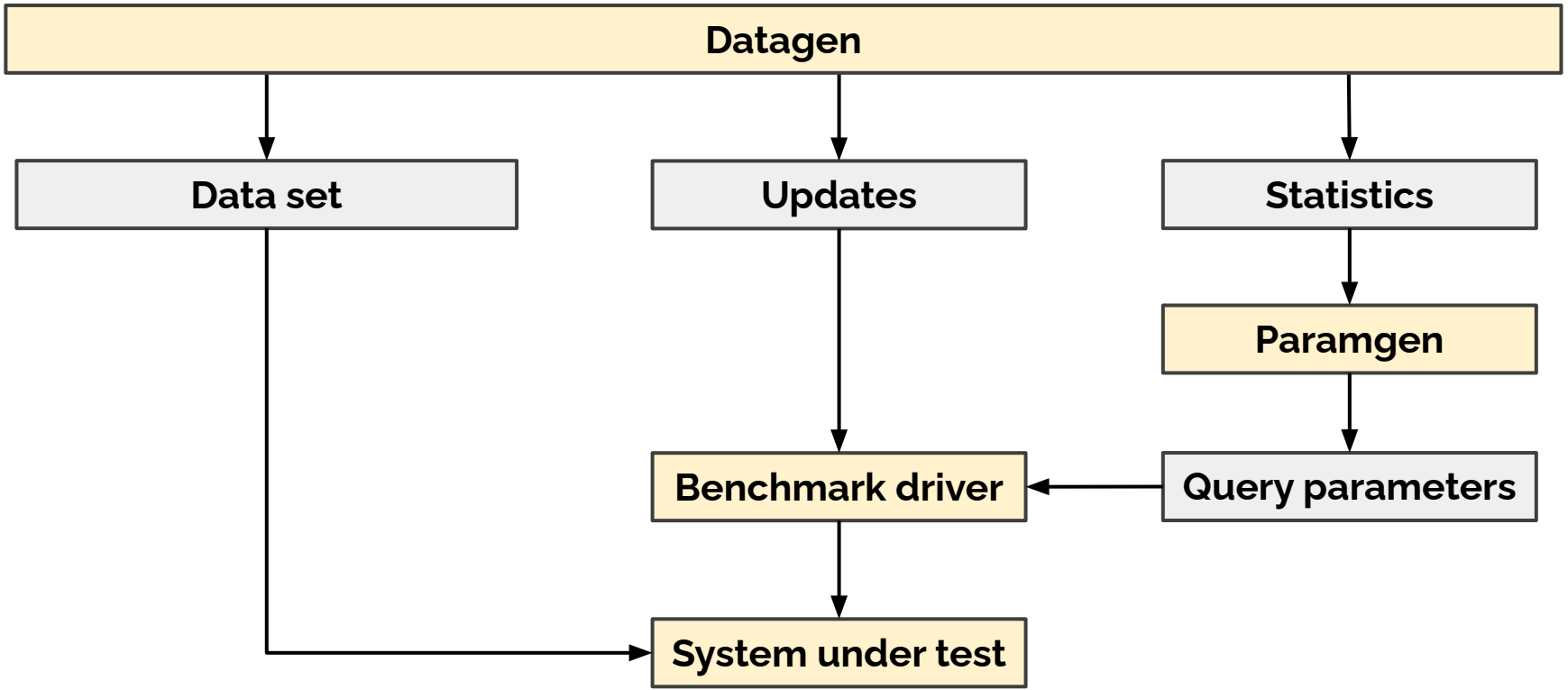
+ Comment(“Gia”, “M3”)

– Person(“Eve”)

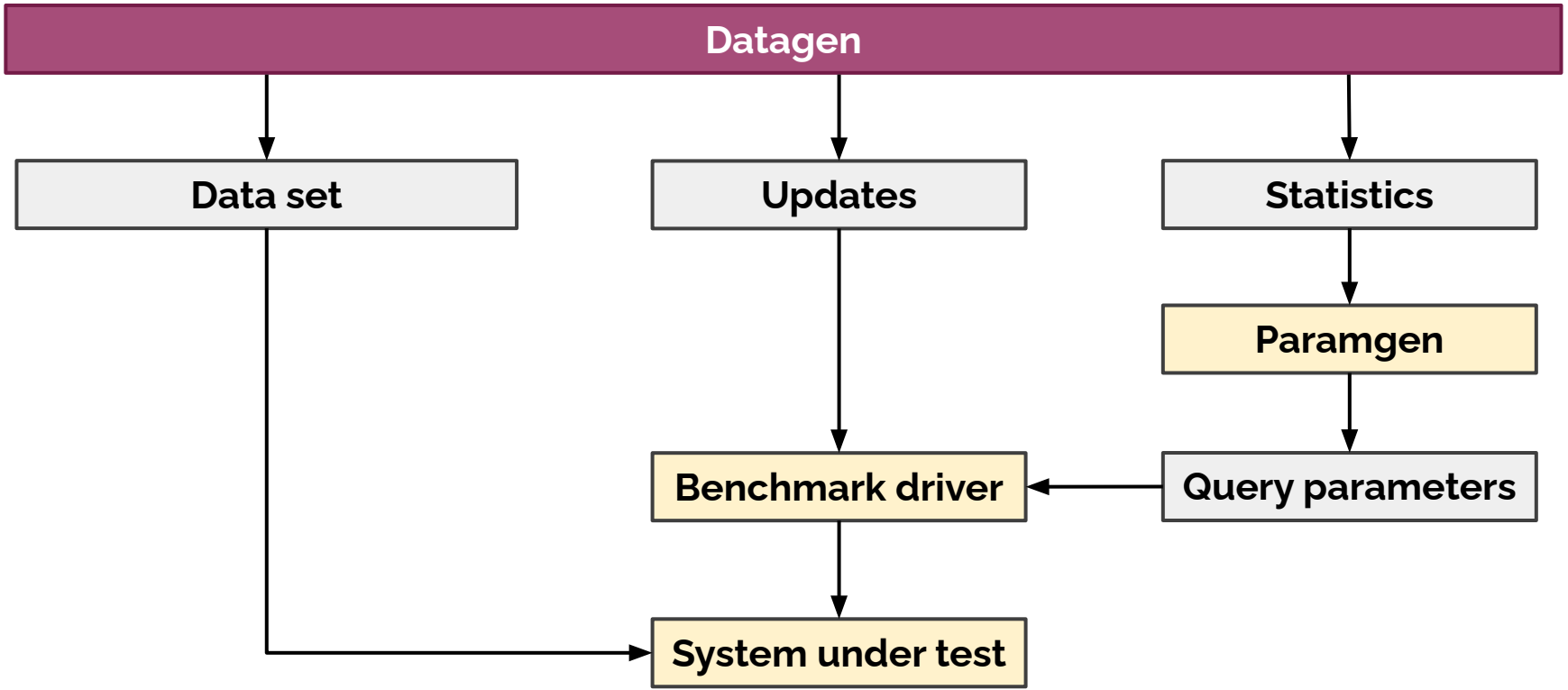
Heavy-hitting operation!

Benchmark framework

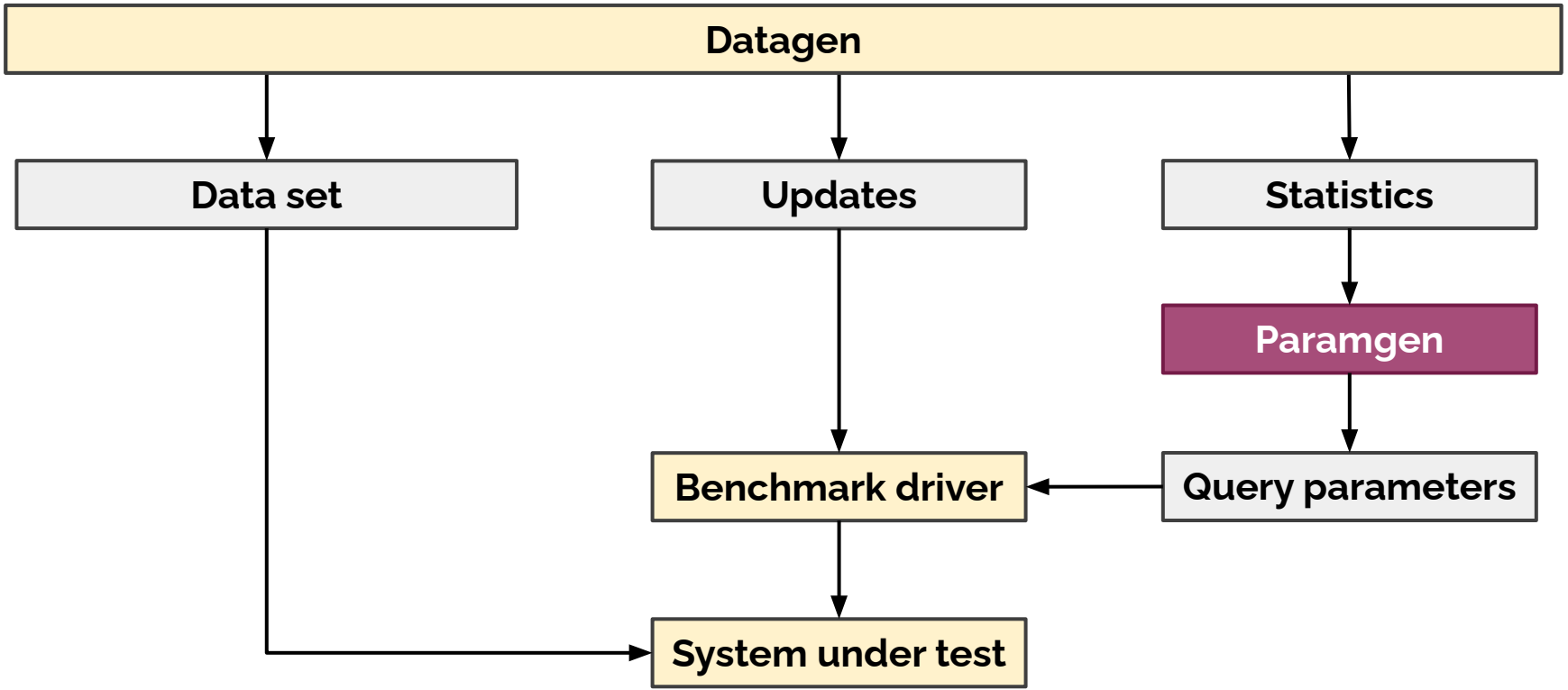
Benchmark workflow



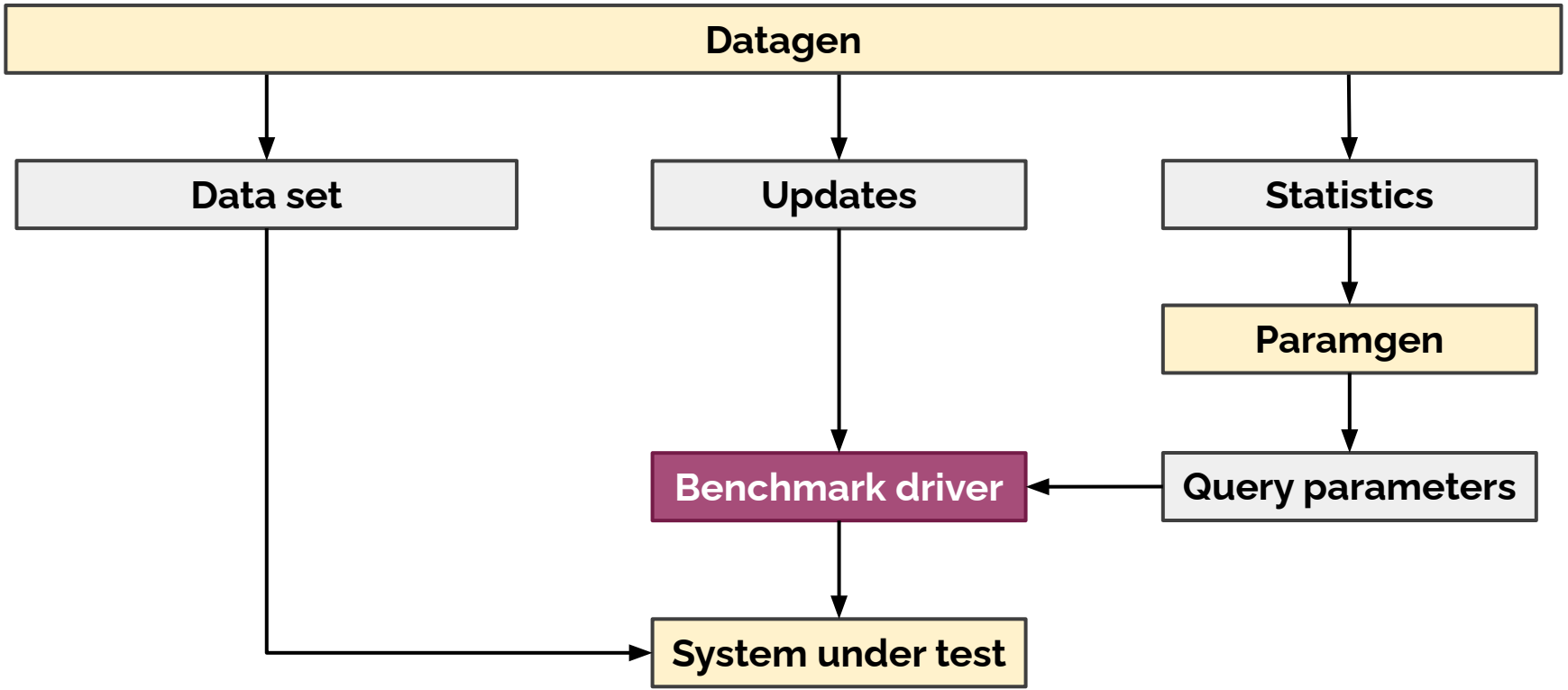
Benchmark workflow



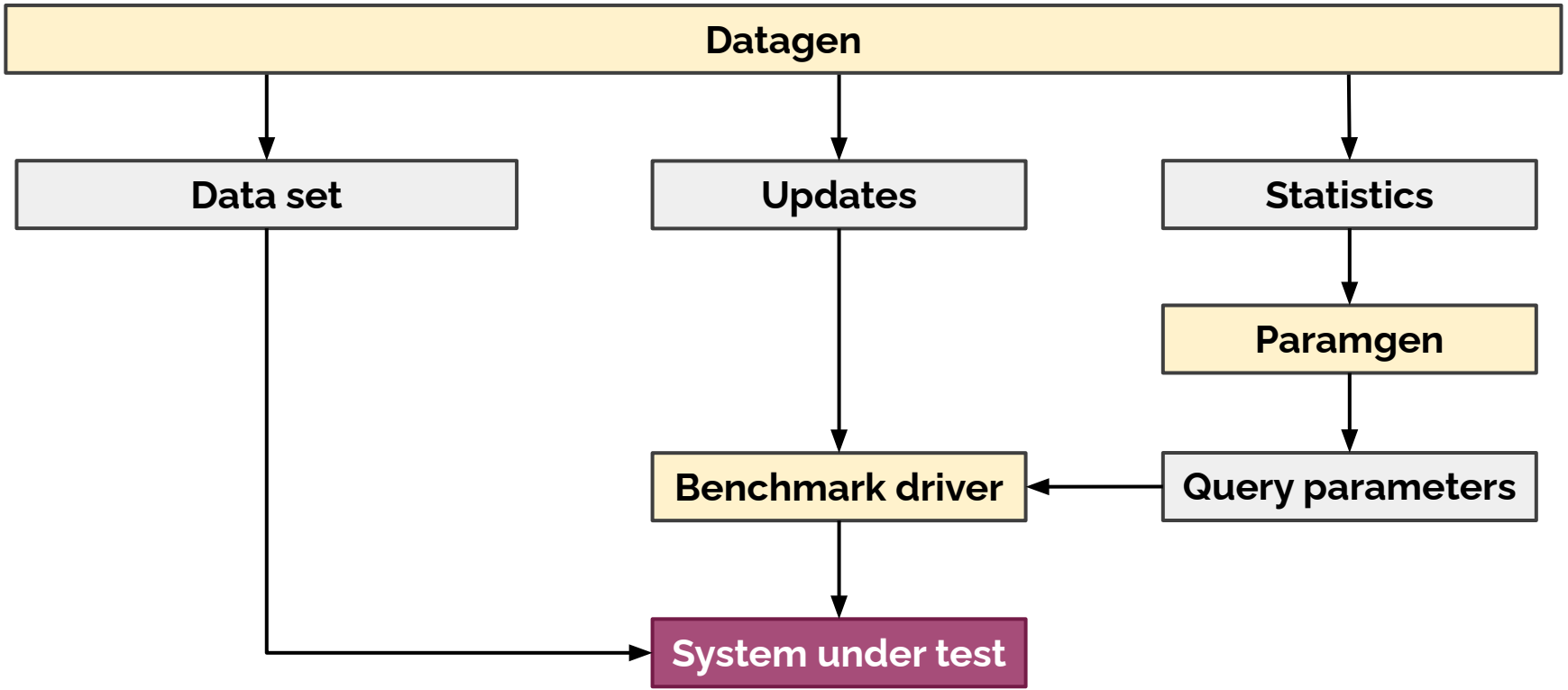
Benchmark workflow



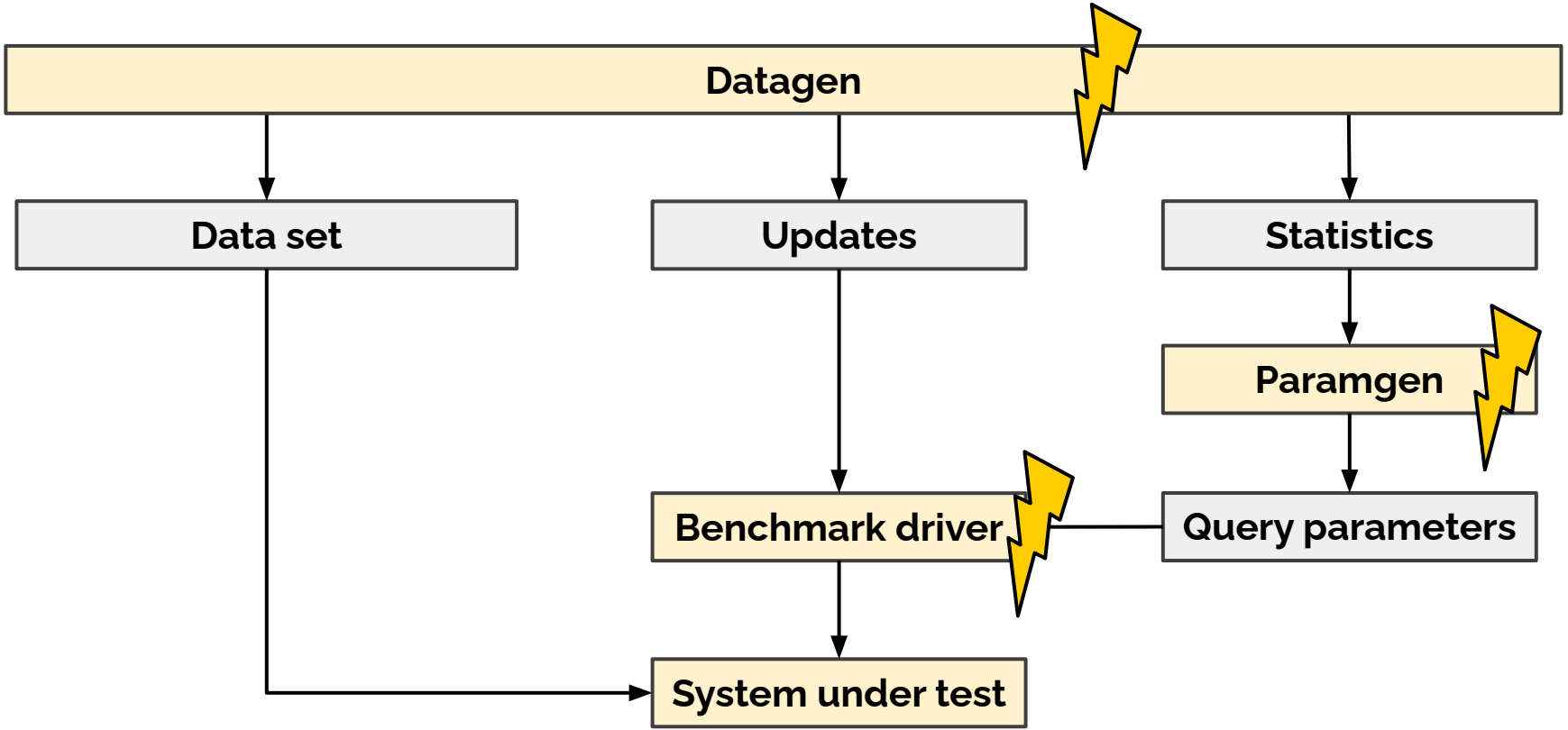
Benchmark workflow



Benchmark workflow



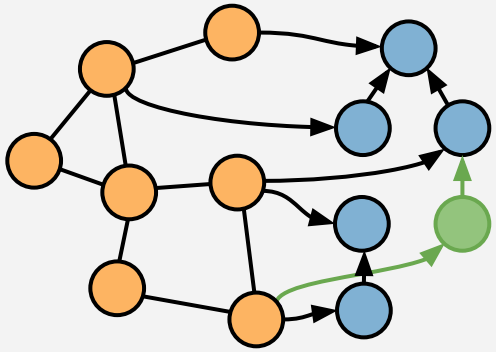
Benchmark workflow

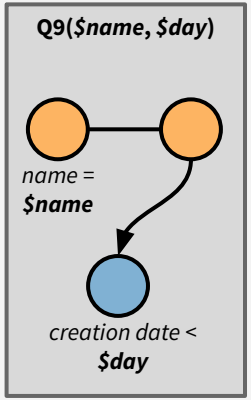


SNB workloads

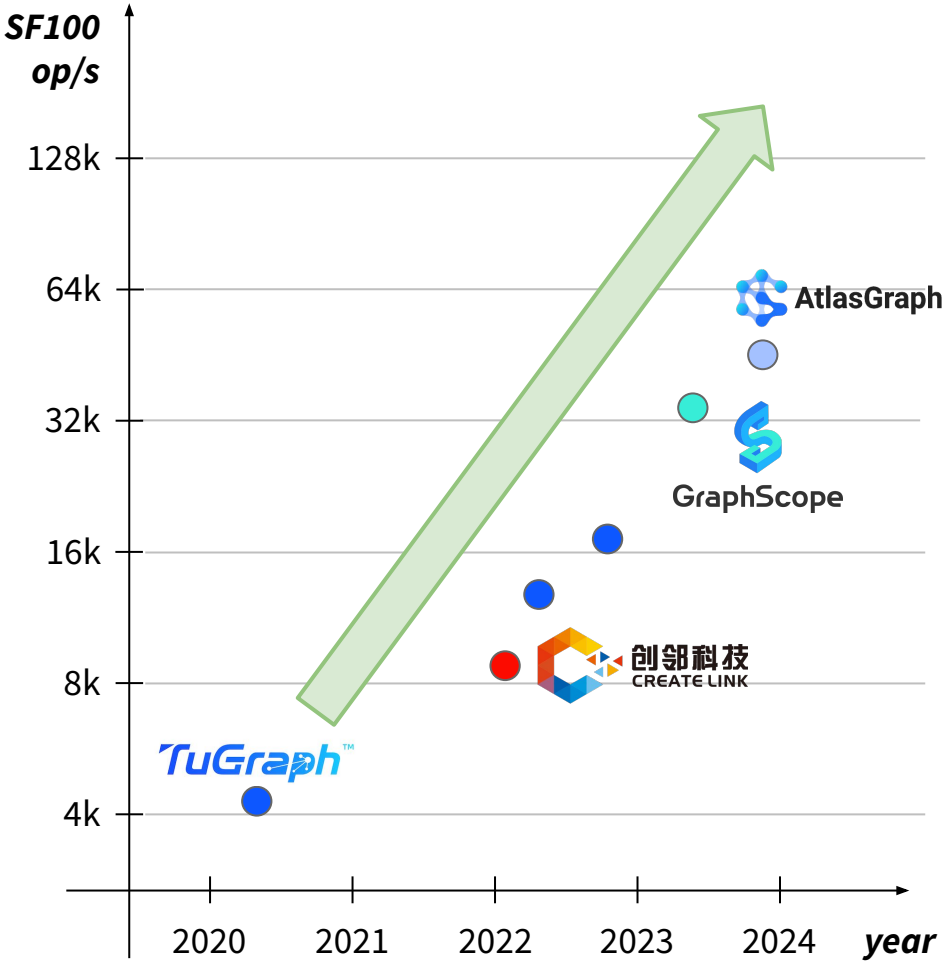
- OLTP: Interactive
- OLAP: Business Intelligence

SNB Interactive v1 (2015)

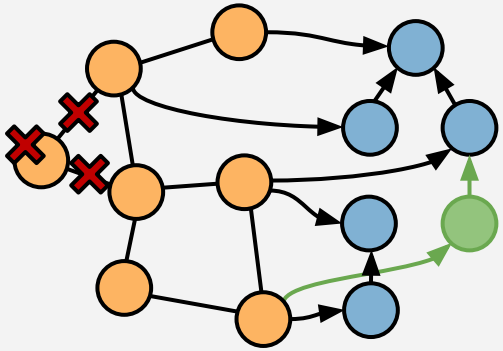


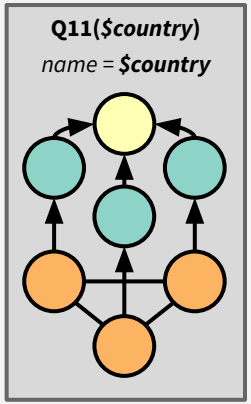


- Queries start in 1–2 person nodes
- 14 complex reads, 7 short reads
- 8 insert operations run concurrently
- Goal: High throughput (ops/s)



SNB Business Intelligence (2022)





- Queries touch on large portions of the data
- 20 complex read queries, insert & delete ops
- Both bulk and concurrent updates allowed
- Goal: High throughput & low query runtimes

Audited results

TigerGraph

SF100
SF1,000
SF10,000

TuGraph™

SF30,000

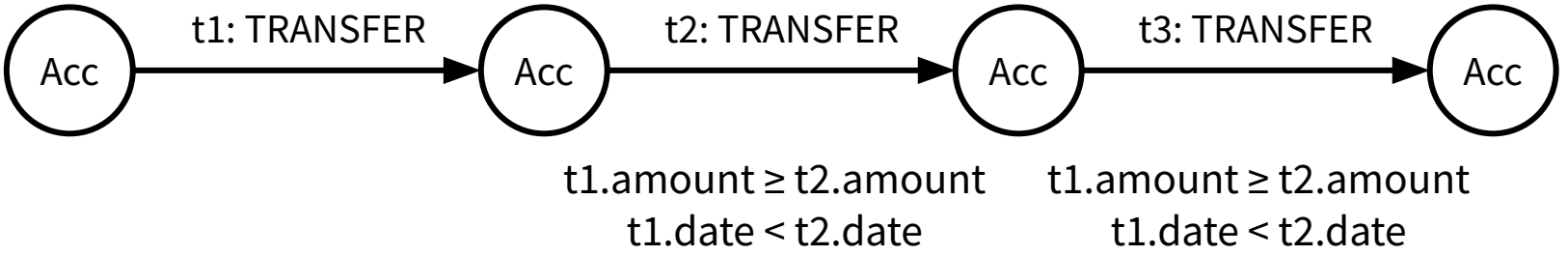
Financial Benchmark

Target: Distributed transactional systems

Financial Benchmark (FinBench)

Developed by the Ant Group, Create Link, Ultipa, etc.

- Strict latency requirements (P99 < 100 ms)
- Relaxed consistency guarantees
- Truncation (sampling) on most recent edges
- Interesting queries, e.g. REM path queries (Regular Expression with Memory)



Financial Benchmark (FinBench) – Timeline

Dec 2020: idea raised

Mar 2021: task force established

June 2023: v0.1 approved (max. scale factor: 10)

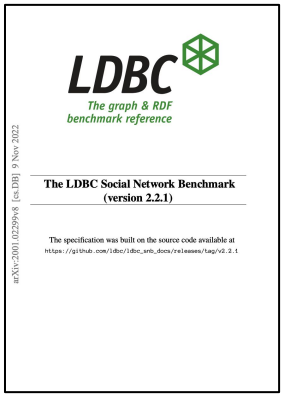
currently: v0.2 is under development

Using benchmarks

Making benchmarks easy to use

For each workload:

- Specification
- Academic paper
- Data generator
- Pre-generated data sets
- Benchmark driver
- 2+ reference implementations



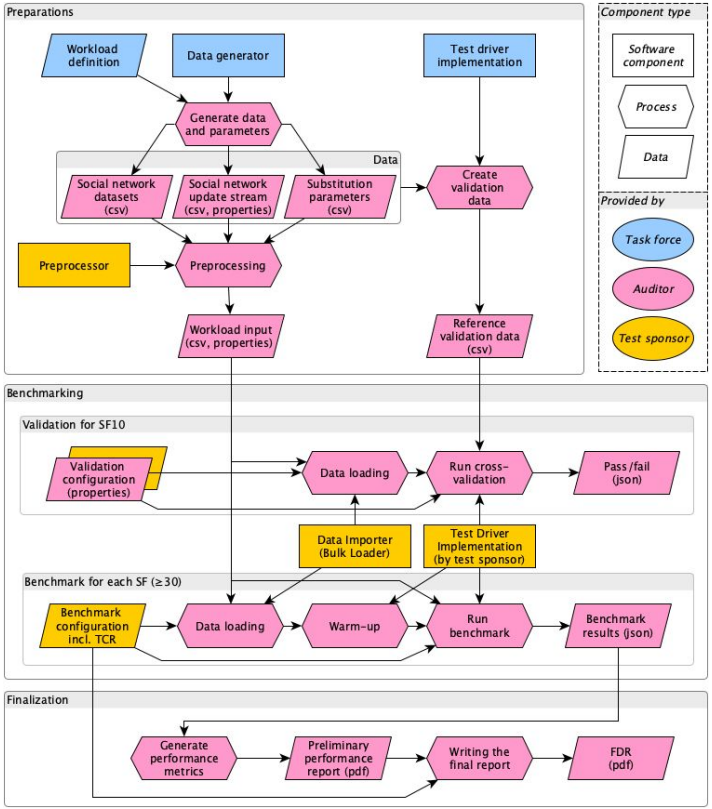
Auditing and trademark

Auditing process:

- Auditors are trained by the LDBC task forces and they take an *auditor exam* to get certified.
- An audit typically costs $20k+ (plus infra) and takes multiple weeks.

Trademark:

- LDBC is **trademarked** worldwide. Only a **result produced by a certified auditor is an “LDBC benchmark result”**
- Unofficial benchmark results must come with a disclaimer: “This is NOT an official LDBC benchmark result”



Pricing

TPC Pricing Specification (v2.9.0), 60+ pages

Clause 4.1: **Minimum Maintenance Requirements**

> *Licensed Compute Services, Physically Acquired hardware, and software maintenance must be figured at a standard Pricing which provides* ***7 days/week, 24 hours/day coverage****. Software maintenance updates must also be included in the pricing. [...]*
>
> *The Response Time for Problem Recognition* ***must not exceed 4 hours****.*

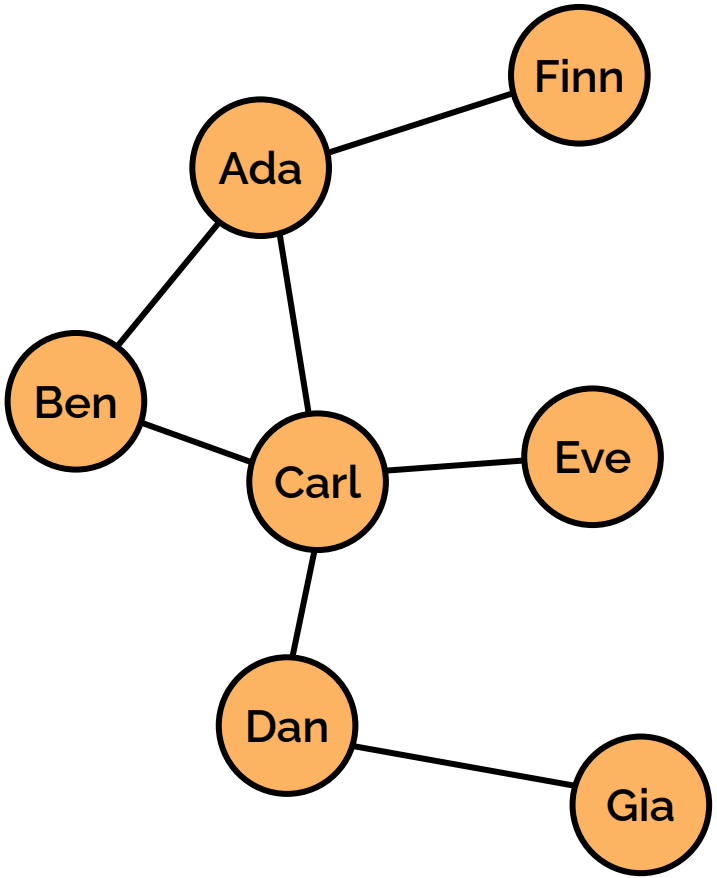
This is “enterprise-grade” support 

The LDBC Graphalytics Benchmark

Graph processing frameworks
(Apache Giraph, NetworKit, GraphBLAS, etc.)

LDBC Graphalytics

- Graphalytics = graph + analytics
- An LDBC benchmark for graph algorithm implementations
- A macrobenchmark
- No audits – competitions with leaderboard ranking
  (similar to HPC benchmarks such as Top500 and Graph500)



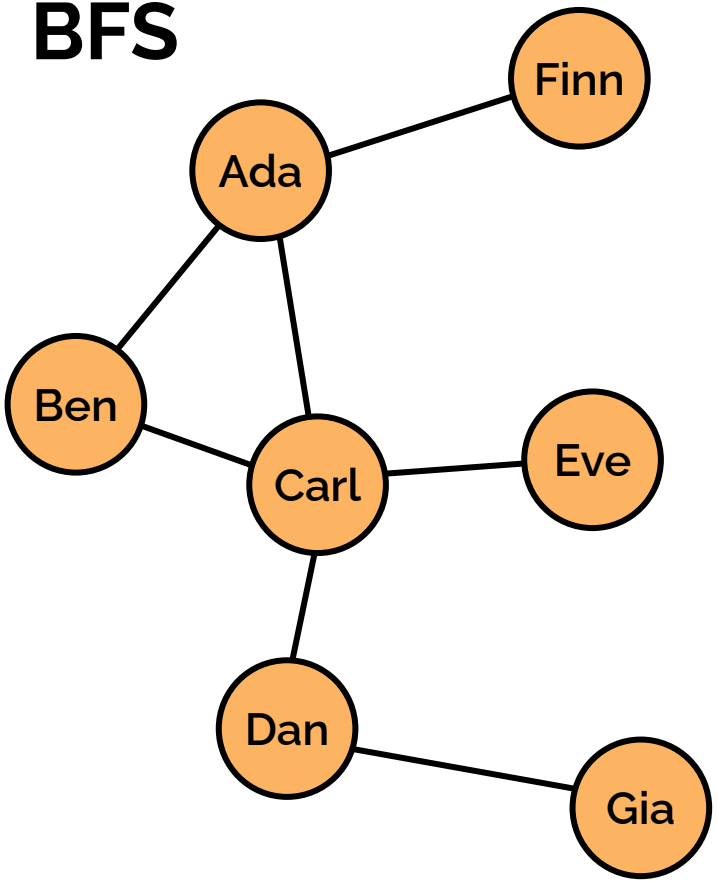
The data sets contain untyped, unattributed graphs with (optional) edge weights

LDBC SNB Datagen
Graph500
Twitter
Friendster
Patents
wiki-Talk
...

Largest graph:

- 450M vertices
- 34B edges

BFS

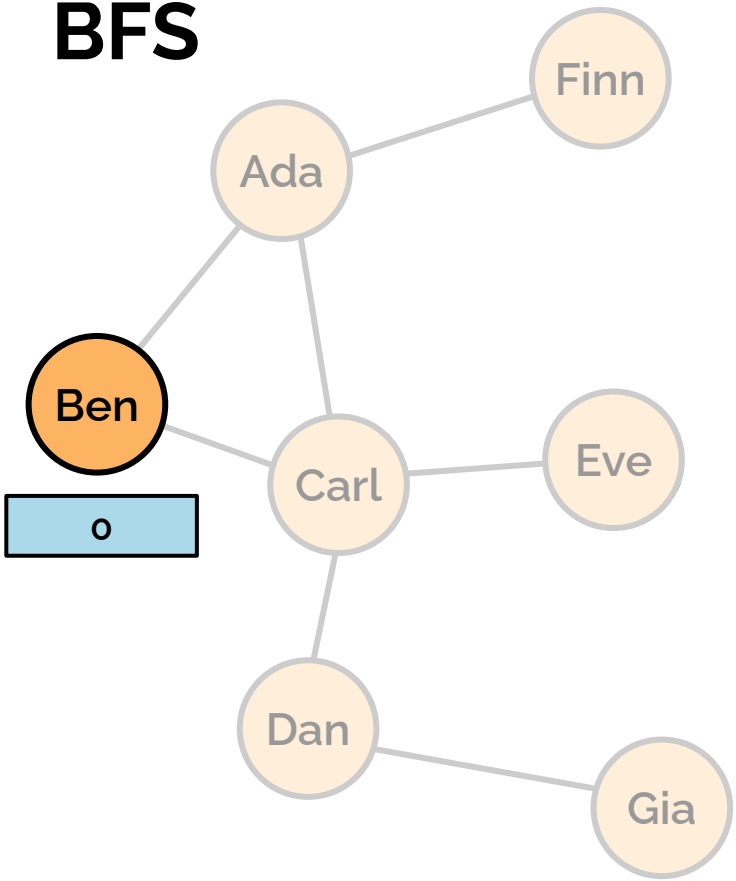


Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

Label: level of traversal

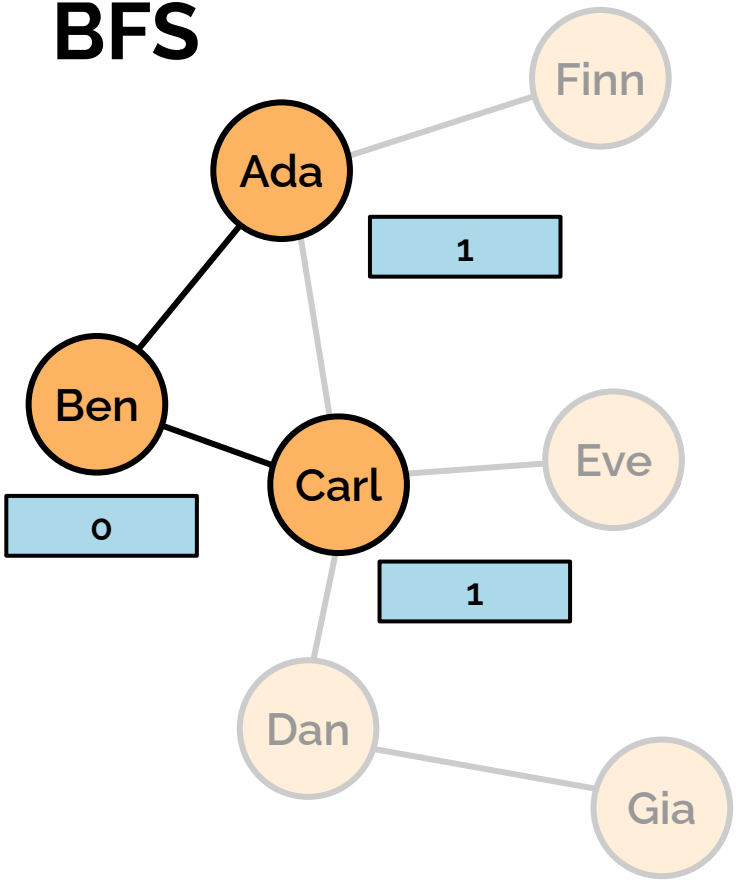
BFS



Graphalytics algorithms

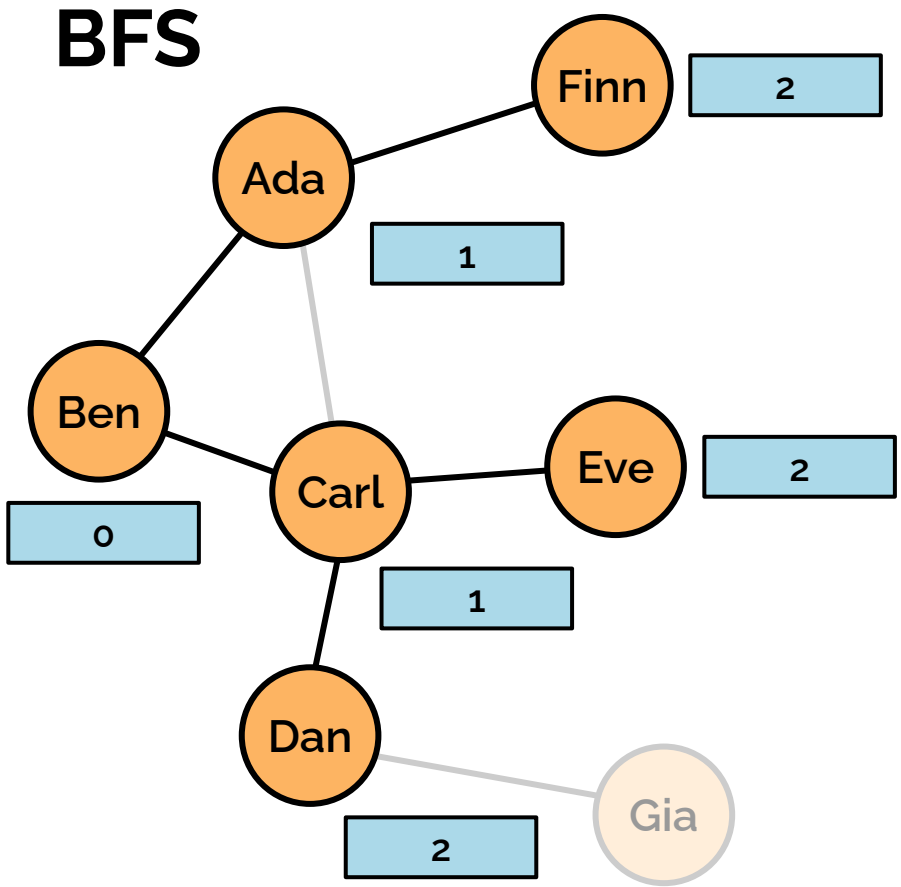
Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

BFS



Graphalytics algorithms

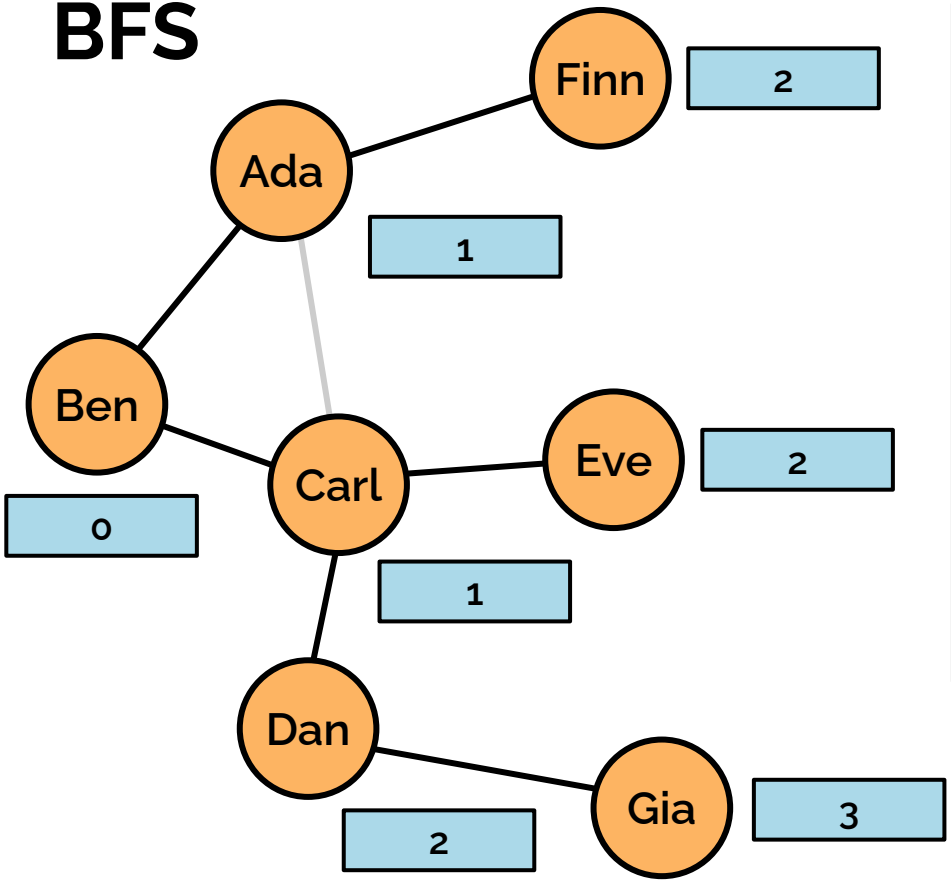
Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)



Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

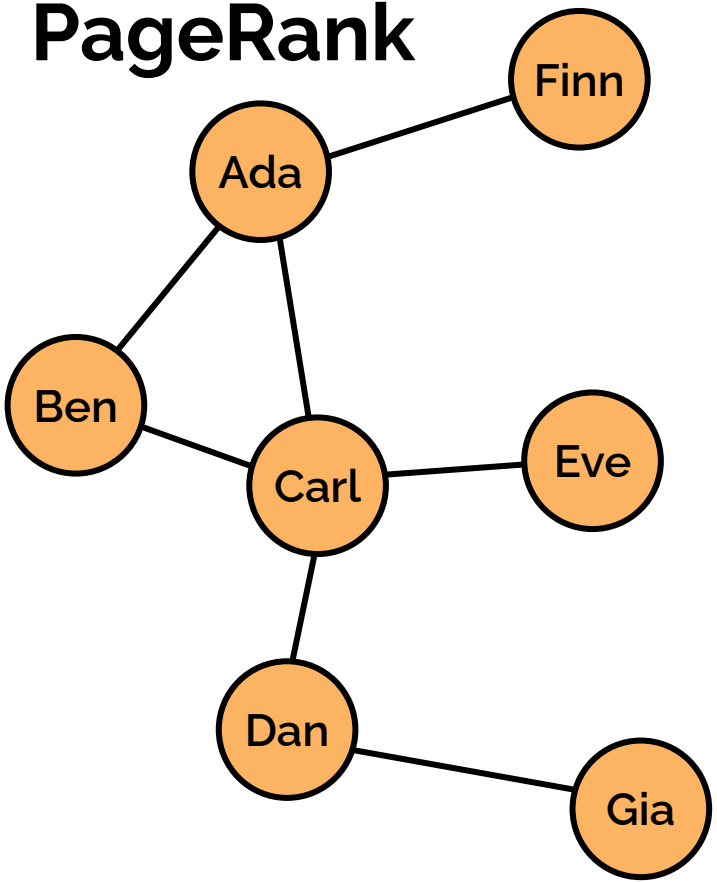
BFS



Graphalytics algorithms

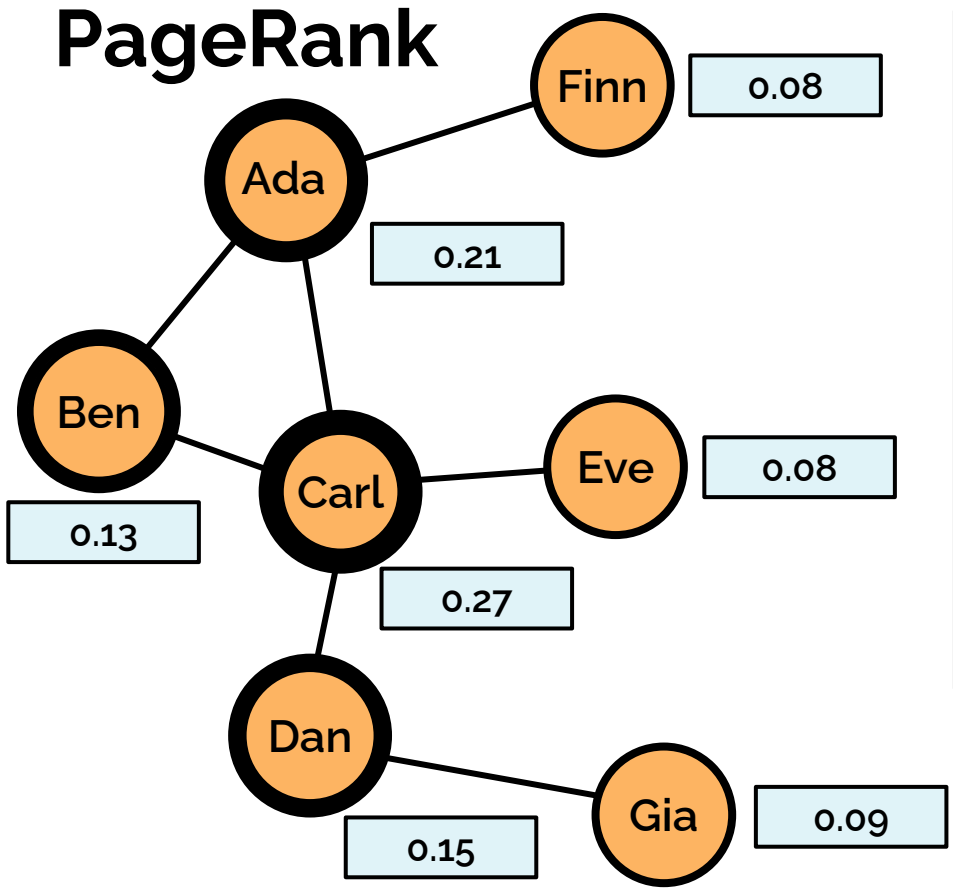
Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

PageRank



Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

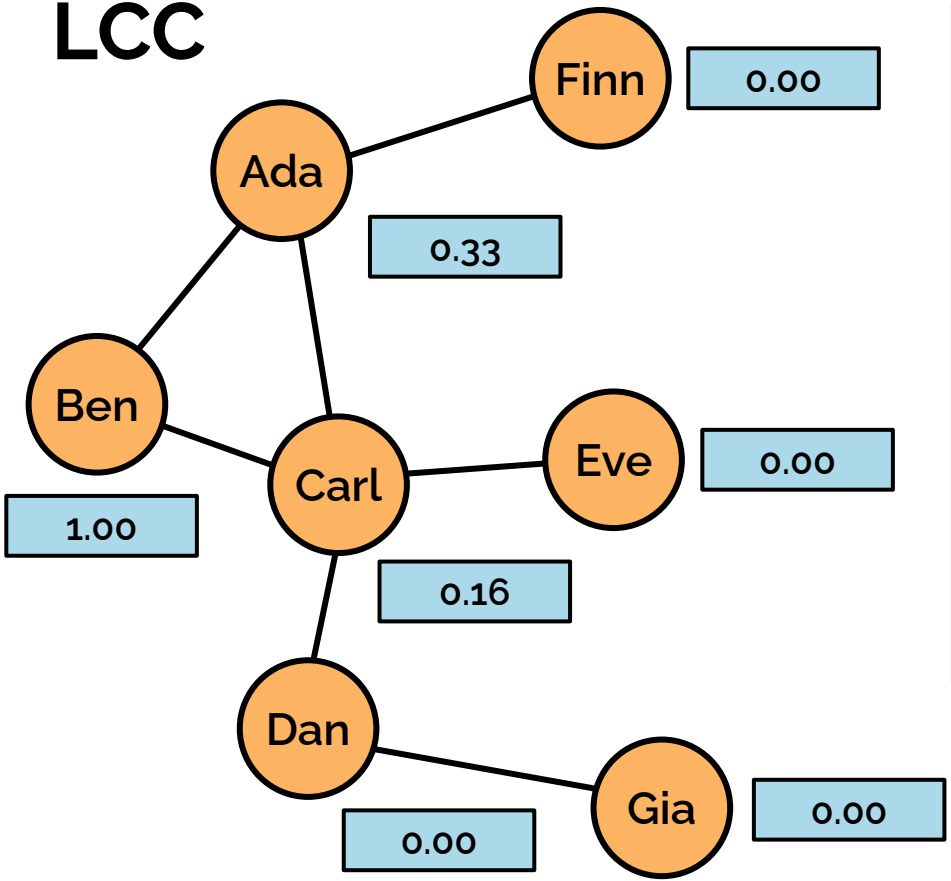


Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

The PageRank variant in Graphalytics redistributes the PageRank values from sinks among all vertices.
(Important for directed graphs.)

LCC



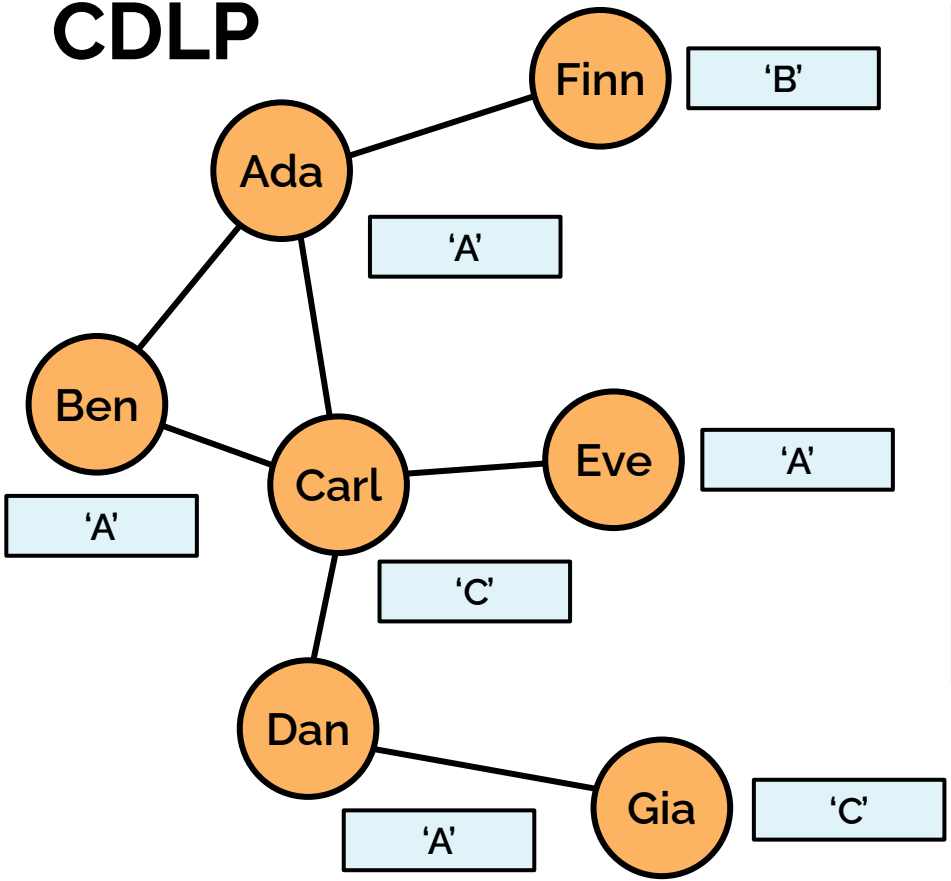
Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: “Ben”)

For each vertex, LCC is #triangles / #wedges.

Similar to triangle count.

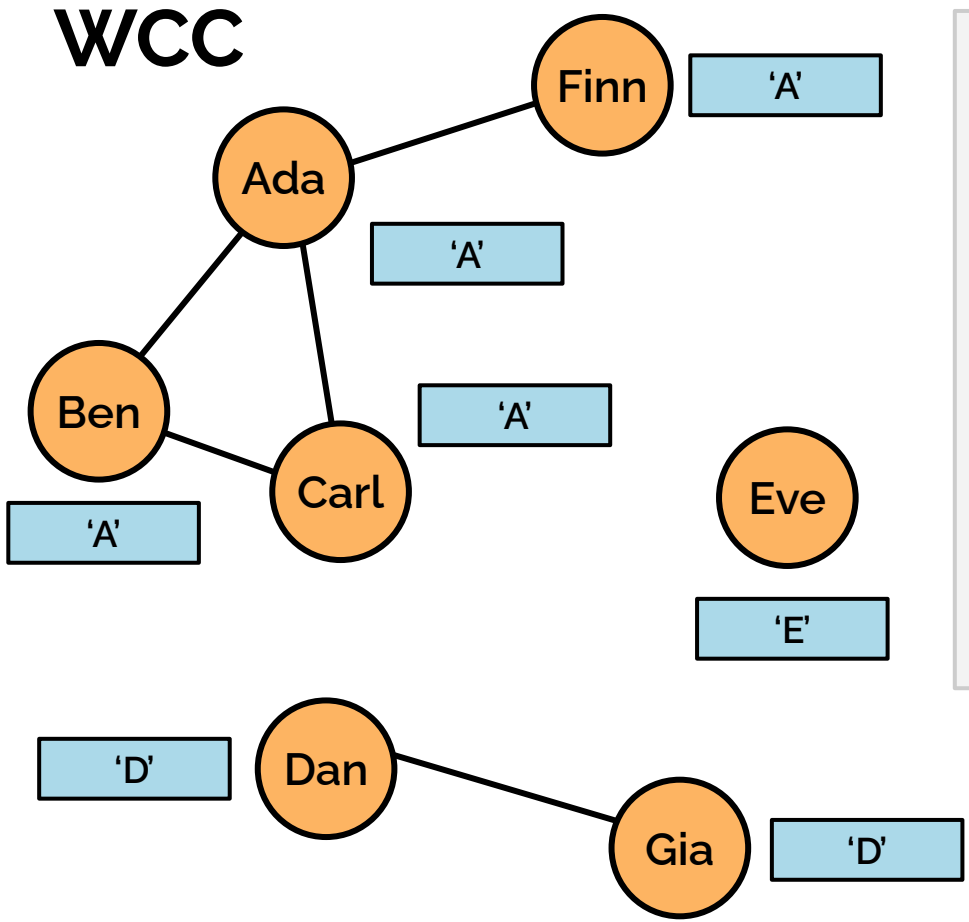
CDLP



Graphalytics algorithms

Breadth-first search(*source*: "Ben")
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: "Ben")

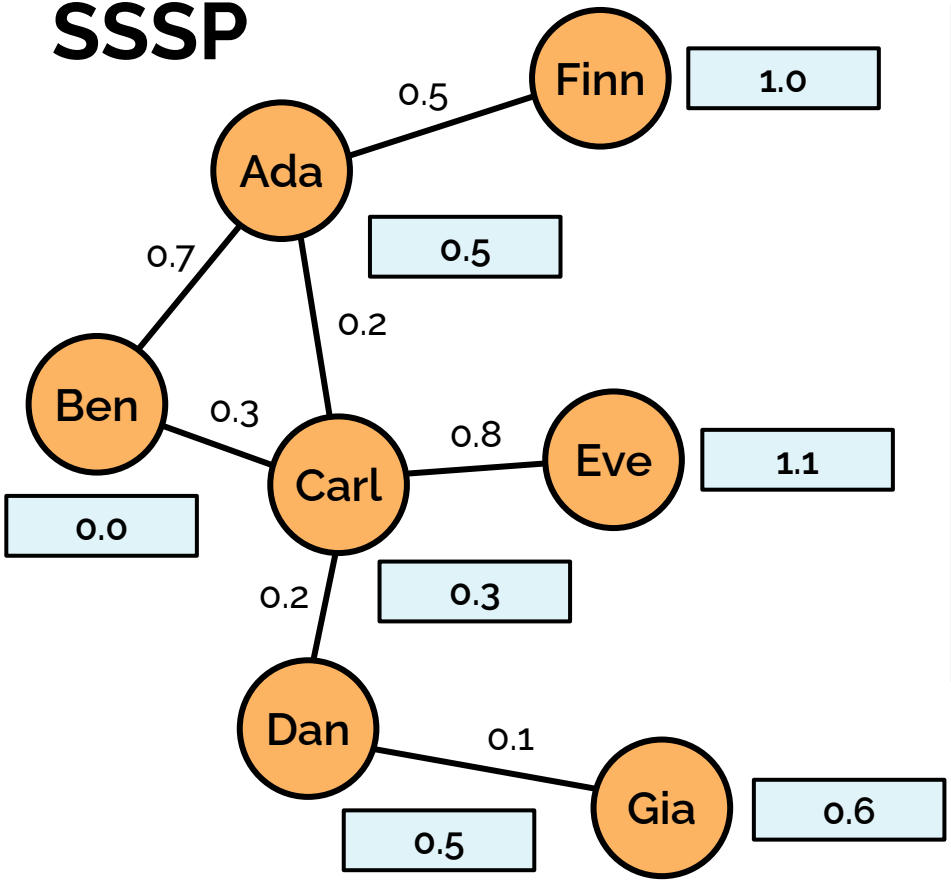
In each iteration, the next label of a vertex is selected as *the minimum mode value among the labels of the neighbours.*



Graphalytics algorithms

Breadth-first search(*source*: "Ben")
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(source: "Ben")

SSSP



Graphalytics algorithms

Breadth-first search(*source*: “Ben”)
PageRank(*damping factor*: 0.85, *iterations*: 5)
Local clustering coefficient
Community detection using LP(*iterations:* 2)
Weakly connected components
Single-source shortest paths(*source*: “Ben”)

Most implementations are expected to use the delta-stepping SSSP algorithm.

Provisional Graphalytics leaderboard (2024)

#	Size	Platform	Environment name	Makespan throughput per dollar	Processing throughput per dollar
1	S	libgrape-gpu	ecs.ebmgn7e.32xlarge	0.08	58.35
2	S	libgrape-lite	ecs.r7.16xlarge	1.45	37.40
3	S	GraphBLAS Intel Xeon Gold 6342	bare metal, dedicated server	6.26	17.99
4	S	GraphBLAS Intel Xeon Platinum 8369	bare metal, dedicated server	7.84	15.73
5	S	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	3.05	11.45
1	M	libgrape-gpu	ecs.ebmgn7e.32xlarge	0.03	37.13
2	M	libgrape-lite	ecs.r7.16xlarge	0.49	25.44
3	M	GraphBLAS Intel Xeon Gold 6342	bare metal, dedicated server	1.94	5.88
4	M	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	1.12	5.74
5	M	GraphBLAS Intel Xeon Platinum 8369	bare metal, dedicated server	2.66	5.32

#	Size	Platform	Environment name	Makespan throughput per dollar	Processing throughput per dollar
1	L	libgrape-gpu	ecs.ebmgn7e.32xlarge	0.03	16.58
2	L	libgrape-lite	ecs.r7.16xlarge	0.14	8.99
3	L	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	0.43	2.27
4	L	GraphBLAS Intel Xeon Gold 6342	bare metal, dedicated server	0.62	2.13
5	L	GraphBLAS Intel Xeon Platinum 8369	bare metal, dedicated server	0.92	1.94
1	XL	libgrape-gpu	ecs.ebmgn7e.32xlarge	0.01	3.98
2	XL	libgrape-lite	ecs.r7.16xlarge	0.06	2.07
3	XL	GraphBLAS Intel Xeon Platinum 8369	bare metal, dedicated server	0.24	0.39
4	XL	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	0.08	0.28
1	2XL	libgrape-gpu	ecs.ebmgn7e.32xlarge	0.01	0.59
2	2XL	libgrape-lite	ecs.r7.16xlarge	0.05	0.33
3	2XL	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	0.02	0.05
4	2XL	GraphBLAS Intel Xeon Platinum 8369	bare metal, dedicated server	0.03	0.04
1	3XL	libgrape-lite	ecs.r7.16xlarge	0.01	0.07
2	3XL	Geacompute	ecs.c8i.24xlarge / ecs.c8a.48xlarge	0.01	0.01

Graph query languages

Graph query languages: Tower of Babel

Cypher

GSQL

SPARQL

DQL

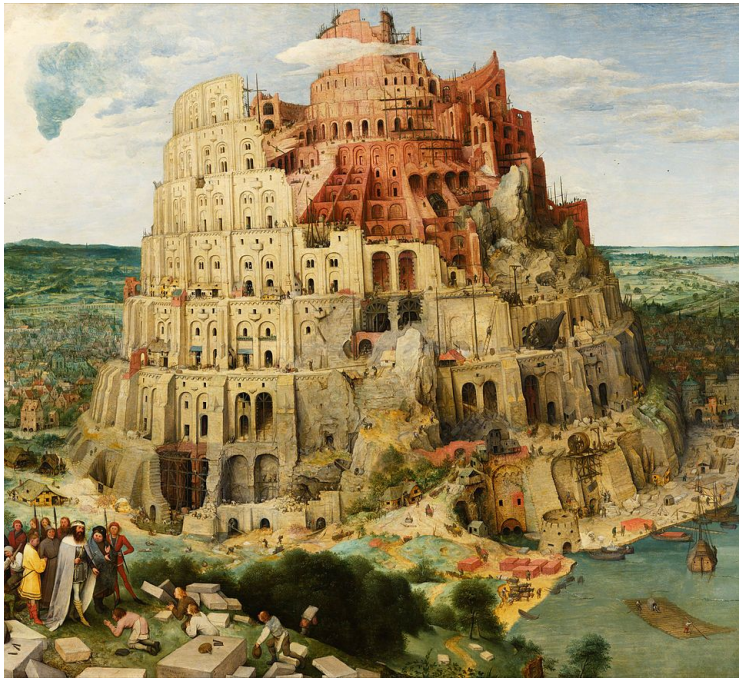
AQL

TypeQL

Gremlin

nGQL

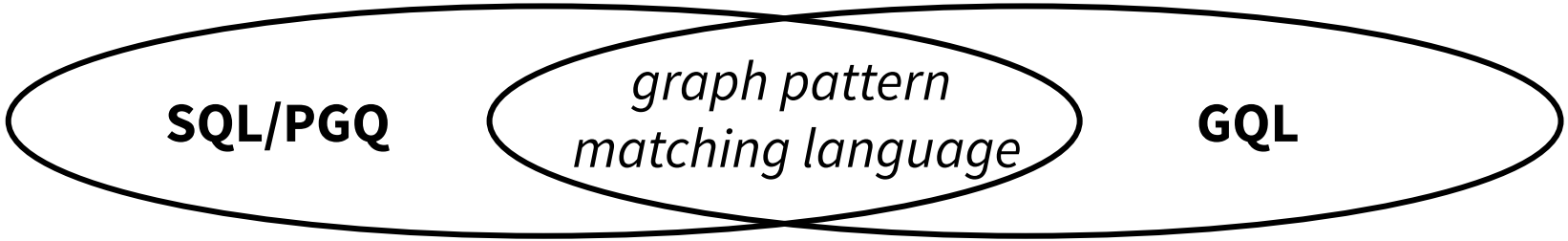
Datalog



LDBC benchmarks define queries in plain text

New standard query languages

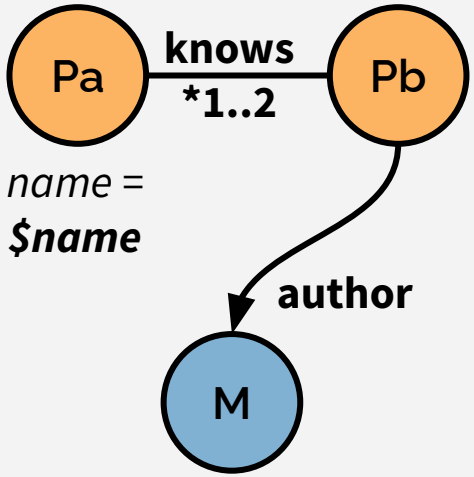
- **LDBC G-CORE** design language (SIGMOD’18)
- **ISO SQL/PGQ** (Property Graph Queries), part of SQL:2023
- **ISO GQL** (Graph Query Language), published in April 2024



SQL:1992

```
SELECT DISTINCT m.id
FROM (
 SELECT k.p2id AS id
 FROM person Pa,
      knows k
 WHERE Pa.name = $name
   AND Pa.id = k.p1id
 UNION
 SELECT k2.p2id AS id
 FROM person Pa,
      knows k1,
      knows k2
 WHERE Pa.name = $name
   AND Pa.id = k1.p1id
   AND k1.p2id = k2.p1id
   AND k1.p1id <> k2.p2id
 ) Pb,
 Message m
WHERE Pb.id = m.authorId
  AND m.creationDate < $day
```

Q9(*$name*, *$day*)



SQL/PGQ (SQL:2023)

```
SELECT id
FROM GRAPH_TABLE (socialNetwork
 MATCH ANY ACYCLIC
   (Pa:Person WHERE Pa.name = $name)
   -[:knows]-{1,2} (Pb:Person)
   -[:author]-> (m:Message)
   HERE m.creationDate < $day
   LUMNS (m.id))
```

Graph pattern matching language with visual graph syntax inspired by Cypher

GQL

```
MATCH ANY ACYCLIC
  (Pa:Person WHERE Pa.name = $name)
  -[:knows]-{1,2} (Pb:Person)
  -[:author]-> (m:Message)
WHERE m.creationDate < $day
RETURN DISTINCT m.id
```

Q13($src, $dst)

SQL/PGQ (SQL:2023)

```
SELECT length FROM GRAPH_TABLE (sn
 MATCH p = ANY SHORTEST
  (Pa:Person WHERE Pa.name = $src)-[:knows]-*
  (Pb:Person WHERE Pb.name = $dst)
COLUMNS (path_length(p) AS length))
```

SQL:1999

```
WITH RECURSIVE ps(sp, ep, path, eR) AS (
  SELECT p1id AS sp, p2id AS ep, [p1id, p2id] AS path, (p2id = $dst) AS eR
    FROM knows WHERE sp = $src UNION ALL SELECT ps.sp AS sp, p2id AS ep,
    array_append(path, p2id) AS path, max(CASE WHEN p2id = $dst THEN 1 ELSE 0 END)
    OVER (ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS eR
    FROM ps JOIN knows ON ps.ep = p1id WHERE NOT EXISTS
    (SELECT 1 FROM ps pps WHERE list_contains(pps.path, p2id)) AND ps.eR = 0)
SELECT min(length(path)) AS length FROM ps WHERE ep = $dst
```

SQL/PGQ and GQL

SQL/PGQ: a large SQL extension

- If widely adopted, it can be a threat to graph systems
- Add support to DuckDB: DuckPGQ project (CWI)

GQL: a standalone language

- LDBC released an open-source toolkit this week
- A few companies already signed up to implement it

LDBC has a **liaison with ISO** which allows its members to access to the standard drafts

Graph schema

Graph schema

- No schema = performance disaster
- Many decisions:
  - undirected edges
  - multiple labels
  - mandatory or optional (weak) schema
  - inheritance
  - composition
  - nested data structures
- Interesting research questions on complexity

LDBC working groups

Graph schema: Balancing expressive power, usability and tractability

- PG-Keys: Keys for Property Graphs (SIGMOD’21)
- PG-Schema: Schemas for Property Graphs (SIGMOD’23)

Graph query languages: Formalizing semantics, ensuring tractability

- G-CORE (SIGMOD’18)
- Graph Pattern Matching in GQL and SQL/PGQ (SIGMOD’23)
- GPC: A Pattern Calculus for Property Graphs (PODS’23)

Running a benchmark organization

Non-technical aspects

LDBC organization

LDBC is registered in the UK as a non-profit company

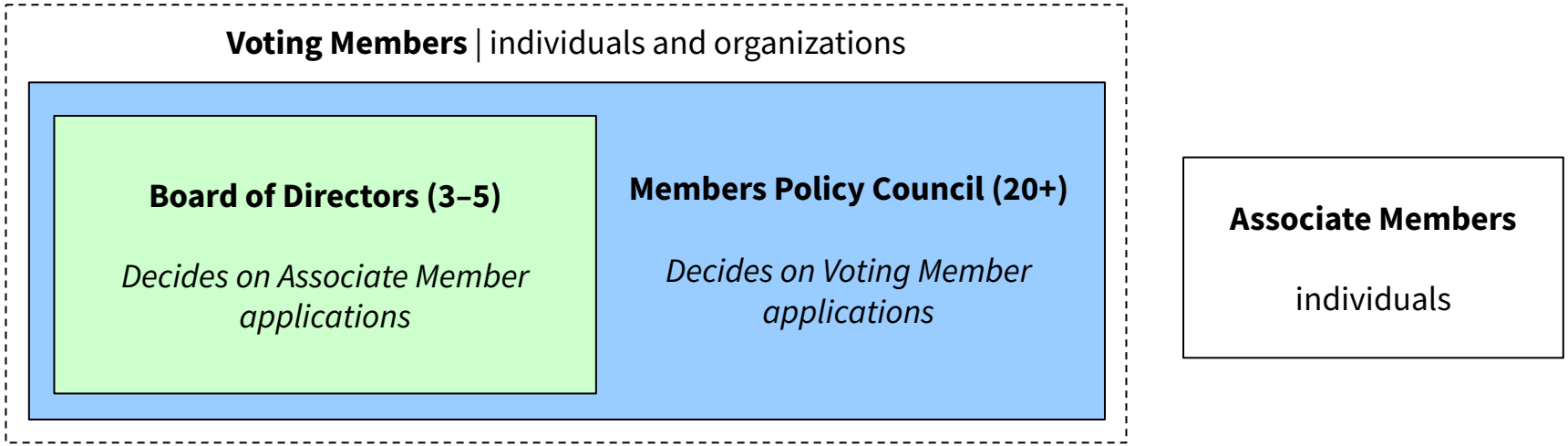
Annual membership fees:

- **sponsors:** **8,800 GBP** ~ 11,000 USD
- **companies:** **2,200 GBP** ~ 2,800 USD
- **institutions:** **1,100 GBP** ~ 1,400 USD

Yearly revenue is approximately 80,000 USD

A small budget for an organization of 20+ companies

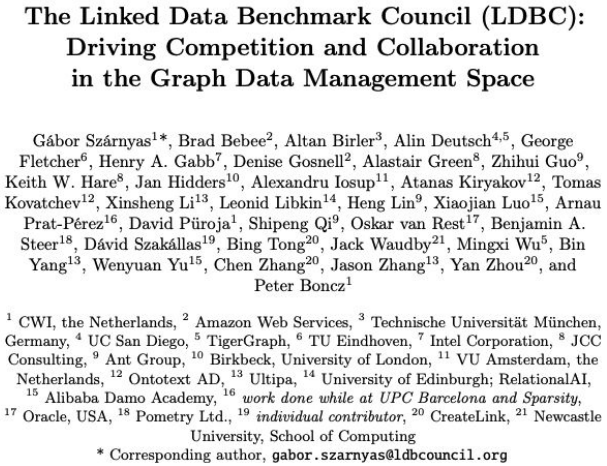
Organizational structure



The main body is the Members Policy Council (mostly company representatives)

The membership form is 32 pages (patent declaration, CLA, etc.)

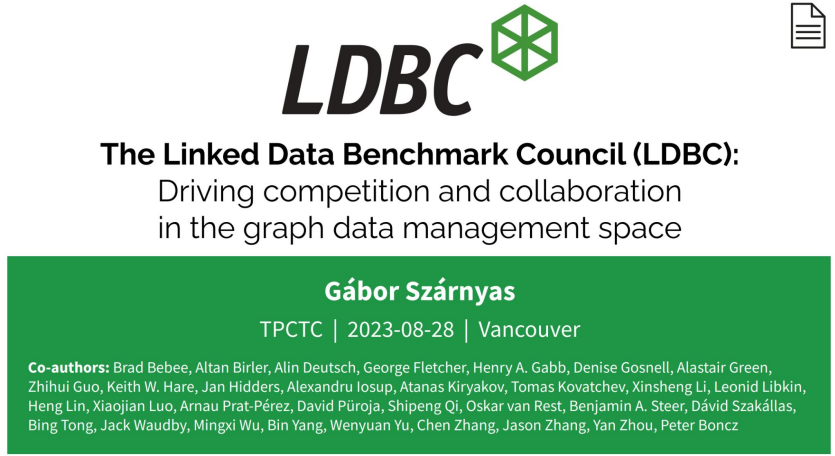
More information: TPCTC 2023 paper



The Linked Data Benchmark Council (LDBC): Driving Competition and Collaboration in the Graph Data Management Space

Gábor Szárnyas[1]*, Brad Bebee[2], Altan Birler[3], Alin Deutsch[4,5], George Fletcher[6], Henry A. Gabb[7], Denise Gosnell[2], Alastair Green[8], Zhihui Guo[9], Keith W. Hare[8], Jan Hidders[10], Alexandru Iosup[11], Atanas Kiryakov[12], Tomas Kovatchev[12], Xinsheng Li[13], Leonid Libkin[14], Heng Lin[9], Xiaojian Luo[15], Arnau Prat-Pérez[16], David Püroja[1], Shipeng Qi[9], Oskar van Rest[17], Benjamin A. Steer[18], Dávid Szakállas[19], Bing Tong[20], Jack Waudby[21], Mingxi Wu[5], Bin Yang[13], Wenyuan Yu[15], Chen Zhang[20], Jason Zhang[13], Yan Zhou[20], and Peter Boncz[1]

[1] CWI, the Netherlands, [2] Amazon Web Services, [3] Technische Universität München, Germany, [4] UC San Diego, [5] TigerGraph, [6] TU Eindhoven, [7] Intel Corporation, [8] JCC Consulting, [9] Ant Group, [10] Birkbeck, University of London, [11] VU Amsterdam, the Netherlands, [12] Ontotext AD, [13] Ultipa, [14] University of Edinburgh; RelationalAI, [15] Alibaba Damo Academy, [16] *work done while at UPC Barcelona and Sparsity*, [17] Oracle, USA, [18] Pometry Ltd., [19] *individual contributor*, [20] CreateLink, [21] Newcastle University, School of Computing
* Corresponding author, gabor.szarnyas@ldbcouncil.org



Running a benchmark organization

- is a multi-decade overtaking
- involves running a fully remote organization
- …where everyone is a part-time employee
- …with limited funding

You can build your own organization or join us

Difficult to bet on technology

A benchmark organization is a multi-decade project

- LDBC's software is mostly written Java – it went out of fashion and came back (!)
- Apple changed their main architecture
- Go and Rust became really mainstream

Recurring idea: a complete rewrite of the SNB driver and datagen in C++/Go/Rust

Maybe textual specification for everything is not a bad idea after all!

Future outlook

Field of graph databases

Identity crisis: *main use cases = social networks, recommendation, fraud detection*

75% of systems have a lower score on the DB Engines ranking compared to May 2023

Rank			DBMS	Database Model	Score		
May 2024	Apr 2024	May 2023			May 2024	Apr 2024	May 2023
1.	1.	1.	Neo4j	Graph	44.46	-0.01	-6.65
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	29.04	-0.81	-6.95
3.	3.	3.	Aerospike	Multi-model	5.78	-0.32	-0.63
4.	4.	4.	Virtuoso	Multi-model	4.26	+0.06	-1.31
5.	5.	5.	ArangoDB	Multi-model	3.32	-0.44	-1.55
6.	↑ 7.	↑ 11.	GraphDB	Multi-model	3.32	+0.22	+0.84
7.	↓ 6.	↓ 6.	OrientDB	Multi-model	3.19	-0.08	-1.30
8.	8.	↑ 9.	Memgraph	Graph	3.02	+0.02	+0.38
9.	9.	↓ 7.	Amazon Neptune	Multi-model	2.20	-0.38	-0.70
10.	10.	10.	NebulaGraph	Graph	2.14	+0.02	-0.47
11.	11.	↑ 13.	Stardog	Multi-model	2.02	-0.03	+0.12
12.	12.	↓ 8.	JanusGraph	Graph	1.94	+0.03	-0.74
13.	13.	↓ 12.	TigerGraph	Graph	1.83	0.00	-0.20
14.	14.	14.	Fauna	Multi-model	1.52	-0.03	-0.37
15.	15.	15.	Dgraph	Graph	1.45	-0.03	-0.42

Graph databases: Relative growth

No longer “hyped”

Graph analytical systems

Very high attrition rate, especially among academic systems

Promising new offerings, e.g.:

- software: GraphBLAS
- hardware: Intel PIUMA, Cerebras, SambaNova, Groq, Graphcore, Untether, …

These only got limited traction

Main challenges

“Missed the boat” (arguably)

LDBC benchmarks don’t sufficiently cover some important recent technologies:

- binary file formats (e.g. Parquet)
- cloud infrastructure and cloud-native systems
  - serverless services (Lambda)
  - serverless database systems (Aurora)
- ML workloads
  - graph neural networks
  - knowledge graphs
  - vector databases

LDBC's main challenges

Fully developing a new LDBC benchmark takes 5+ person-years:

- Without a standard language, implementations take a long time
- Hard to obtain a good baseline system (chicken-or-egg problem)

Audits:

- Most audited results use imperative languages
- Audits are long and expensive: 3 weeks – 3 months (!)

Generating and storing large-scale data sets is expensive

Your main challenges

Figure out how to stay relevant in the presence of the current AI hype

Find a strong value proposition for graph processing systems

- GDBMS shouldn’t just be a nicer way to express and execute BFS
- They should offer many OOMs of performance improvements!

Interesting system: Kùzu (UWaterloo)

Summary

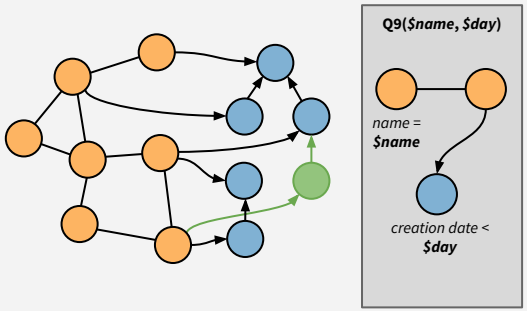
The Linked Data Benchmark Council (LDBC): 12 years of fostering competition in the graph processing space

- LDBC is driving competition
- A slow start but it is working

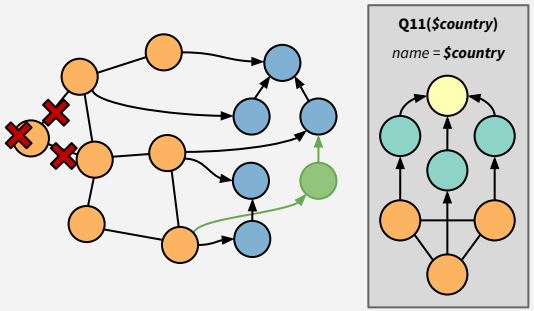
Running LDBC:

- it’s a lot of work and it’s a long-term project
- …but a lot of fun, dipping into many communities (DB, HPC, netsci, semweb, SW eng)

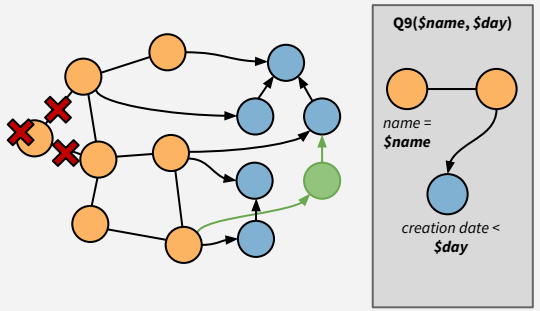
SNB Interactive v1



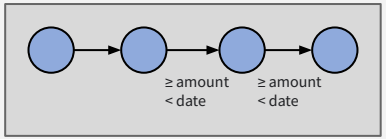
SNB Business Intelligence



SNB Interactive v2



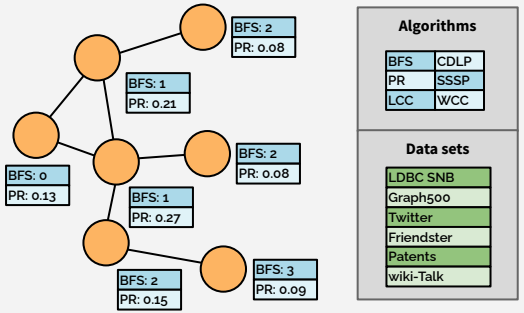
Financial Benchmark



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