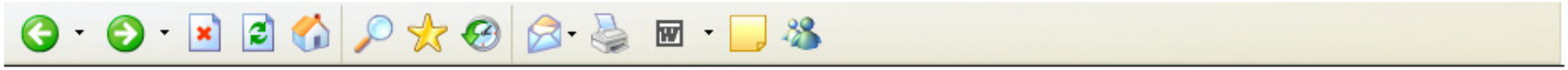
Scalable Semantic Web Data Management Using Vertical Partitioning

Daniel Abadi[2→1], Adam Marcus[2], Samuel Madden[2], and Kate Hollenbach[2]

[1]Yale University [2]MIT

September 27, 2007



DBLife Search Login Help The Cimple Project Wiki

Daniel J. Abadi

Mentions **1** - **10** out of **139**

Saturday
Aug 25, 2007

"Sam Madden's Publications" page, Sam Madden's homepage
New pattern used

...ICEDB: Intermittently Connected Continuous Query Processing. In Proceedings of ICDE, 2007. [PDF] Daniel Abadi, Daniel Myers, David DeWitt, Samuel Madden. Materialization Strategies in a Column-Oriented DBMS ....
http://db.lcs.mit.edu/madden/pubs.php Cached Annotated

"Sam Madden's Publications" page, Sam Madden's homepage
...[PDF]2007 Michael Stonebraker, Samuel Madden, Daniel Abadi, Stavros Harizopoulos, Nabil Hachem, Pat...

Wednesday
Aug 22, 2007

"Sam Madden's Publications" page, Sam Madden's homepage
...| Publications | Talks | DB Group] View Publications 2007 Michael Stonebraker, Samuel Madden, Daniel Abadi, Stavros Harizopoulos, Nabil Hachem, Pat Helland. The End of an Architectural Era (Itâ€™s Time for...
http://db.lcs.mit.edu/madden/pubs.php Cached Annotated

"Sam Madden's Publications" page, Sam Madden's homepage
...Scheduling. In Proceedings of SOSP, 2007. [PDF] Daniel Abadi, Adam Marcus, Samuel Madden, Katherine...

Tuesday
Jul 24, 2007

"VLDB 2007 - 33rd Very Large Data Bases Conference" page, 33rd International Conference on Very Large Data Bases website
Page monitored for first time

...(Yahoo! Research, USA) Scalable Semantic Web Data Management Using Vertical Partitioning Daniel Abadi, Adam Marcus, Samuel Madden, Katherine Hollenbach (MIT) Research Session 11: Time-Series Data...
http://www.vldb2007.org/program/details_wednesday.html Cached Annotated

Tuesday
Jul 24, 2007

"VLDB 2007 - 33rd Very Large Data Bases Conference" page, 33rd International Conference on Very Large Data Bases website
Page monitored for first time

...End of an Architectural Era (It's Time for a Complete Rewrite) Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos (MIT, USA), Nabil Hachem (AvantGarde Consulting, USA), Pat Helland...
http://www.vldb2007.org/program/details_tuesday.html Cached Annotated

http://www.mit.edu/people/dna/ Annotated
PhD Candidate
Electrical Engineering and Computer Science, Massachusettes Institute of Technology, USA

139 total mentions occuring in 40 pages.
1 new mentions found in the last 24 hours.

Related People

- Samuel Madden
- Mitch Cherniack
- Wolfgang Lindner
- Stavros Harizopoulos

more

Related Topics

- sensor networks
- xml
- algorithms
- search

more

Publications

- Materialization Strategies in a Column-Oriented DBMS
- Column Stores for Wide and Sparse Data
- Performance Tradeoffs in Read-Optimized Databases
- Integrating compression and execution in column-oriented database systems

more **(filtered)**

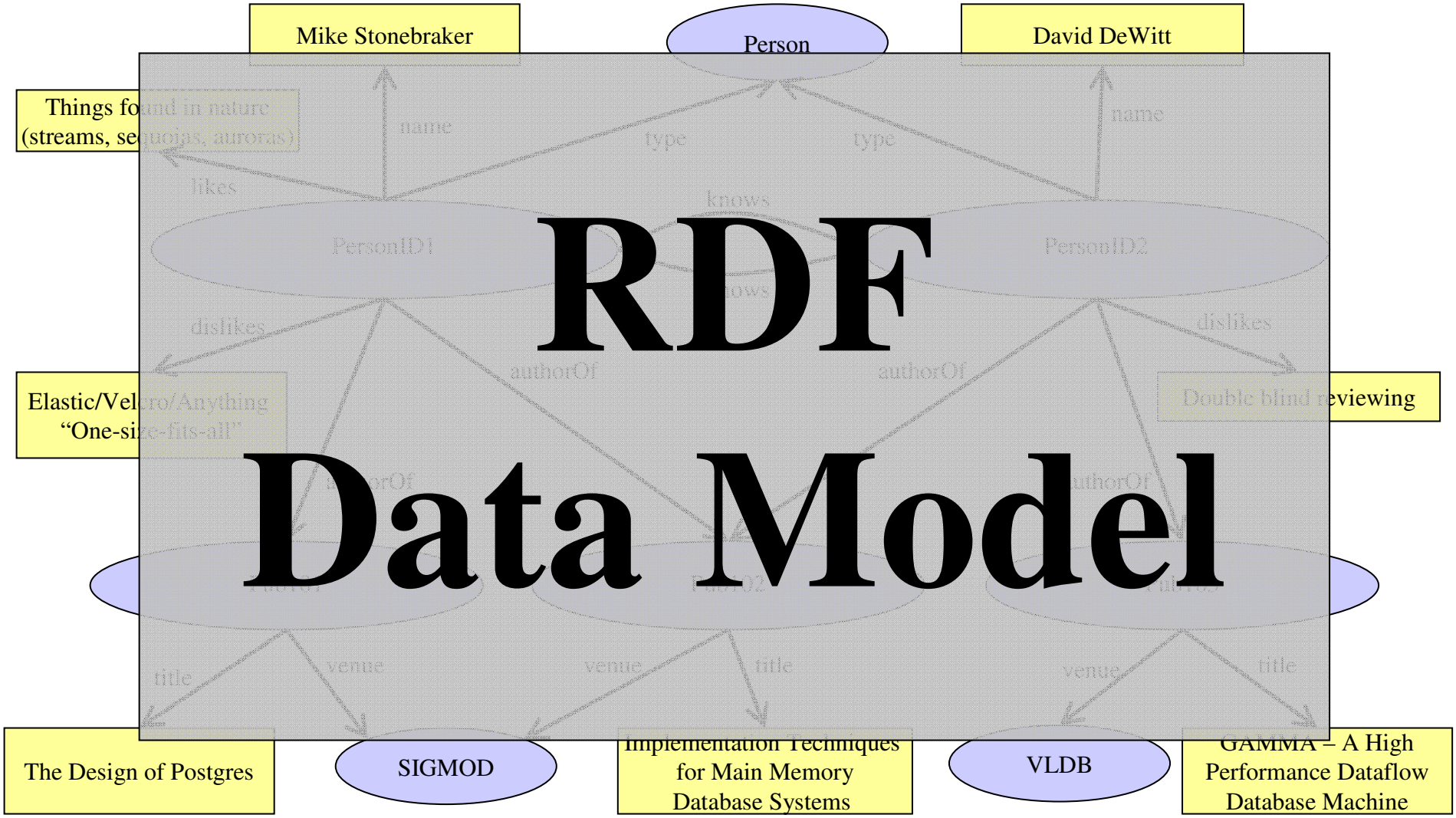
Related Organizations

- Massachusettes Institute of Technology
- Yale University
- Harvard University

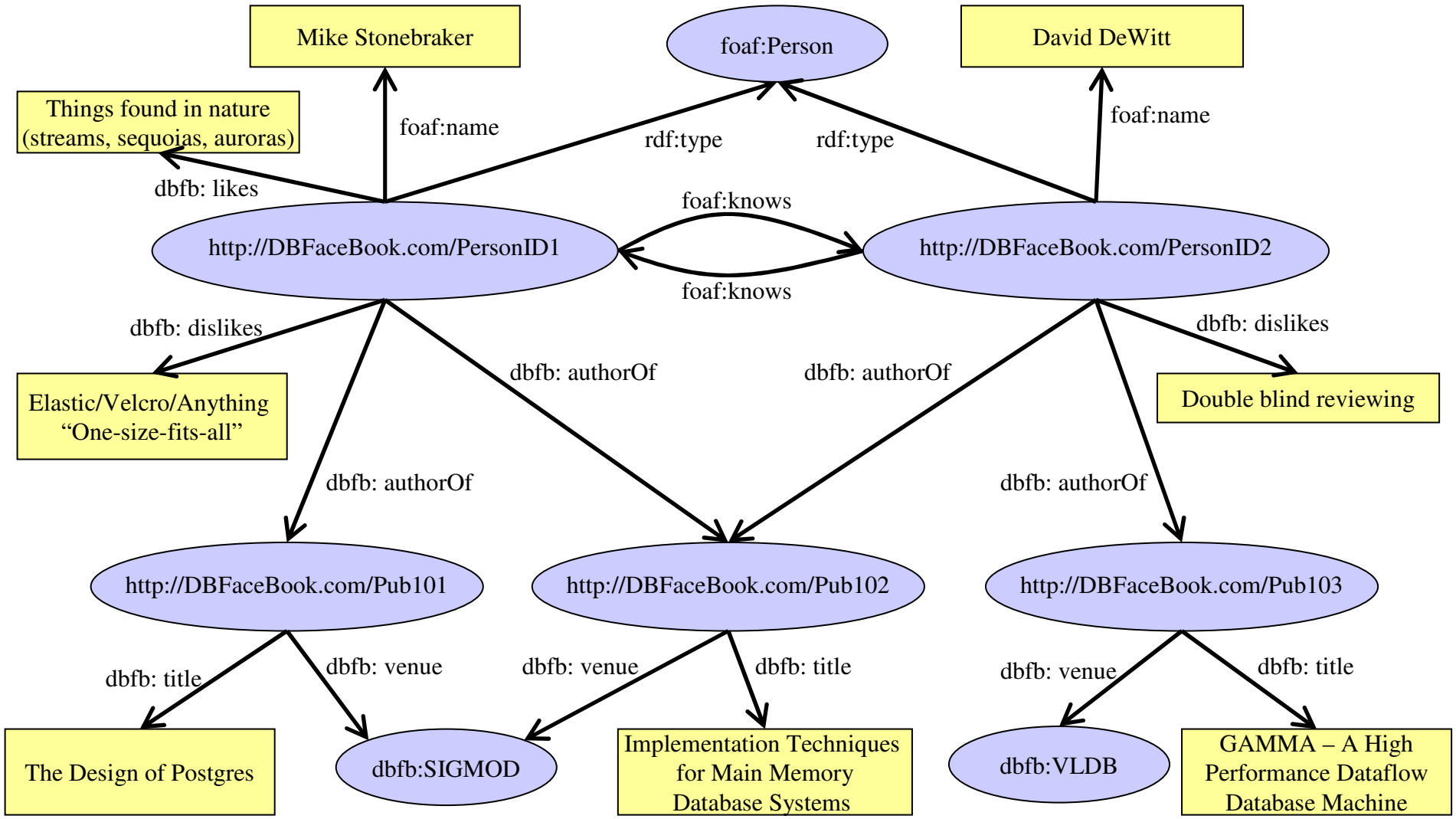
RDF Data Is Proliferating

- Semantic Web vision: make Web machine-readable
- RDF is the data model behind Semantic Web
- Increasing amount of data published using RDF
  - Swoogle indexes 2,271,350 Semantic Web documents
- Biologists seem sold on Semantic Web
  - Integrated data from Swiss-Prot, TrEMBL, and PIR protein databases available in RDF (500 million statements)

DBFacebook: A New Social Networking Application



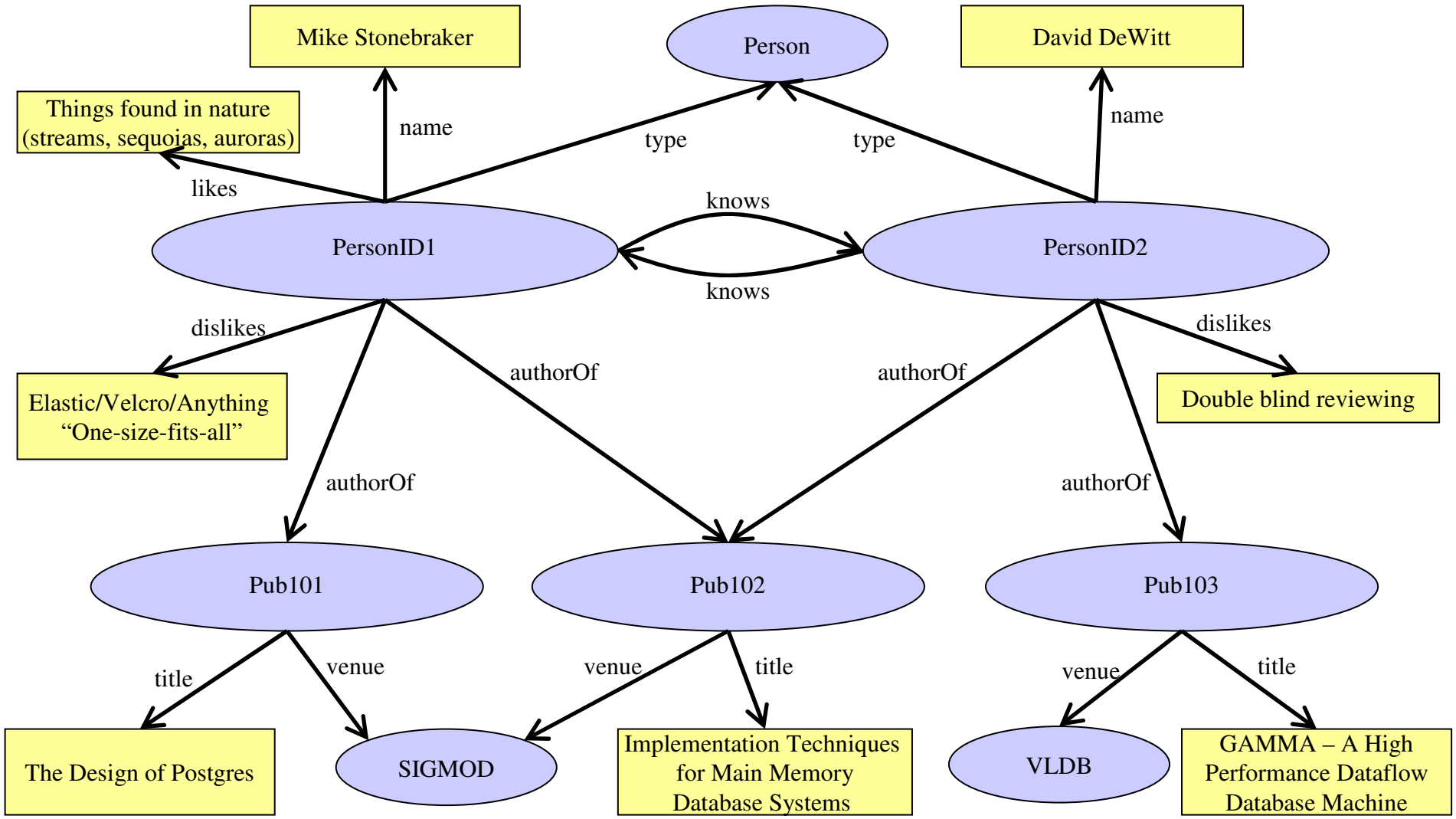
DBFacebook: A New Social Networking Application



RDF Data Management

- Early projects built their own RDF stores
- Trend now towards storing in RDBMSs
- Paper examines 3 approaches for storing RDF data in a RDBMS …

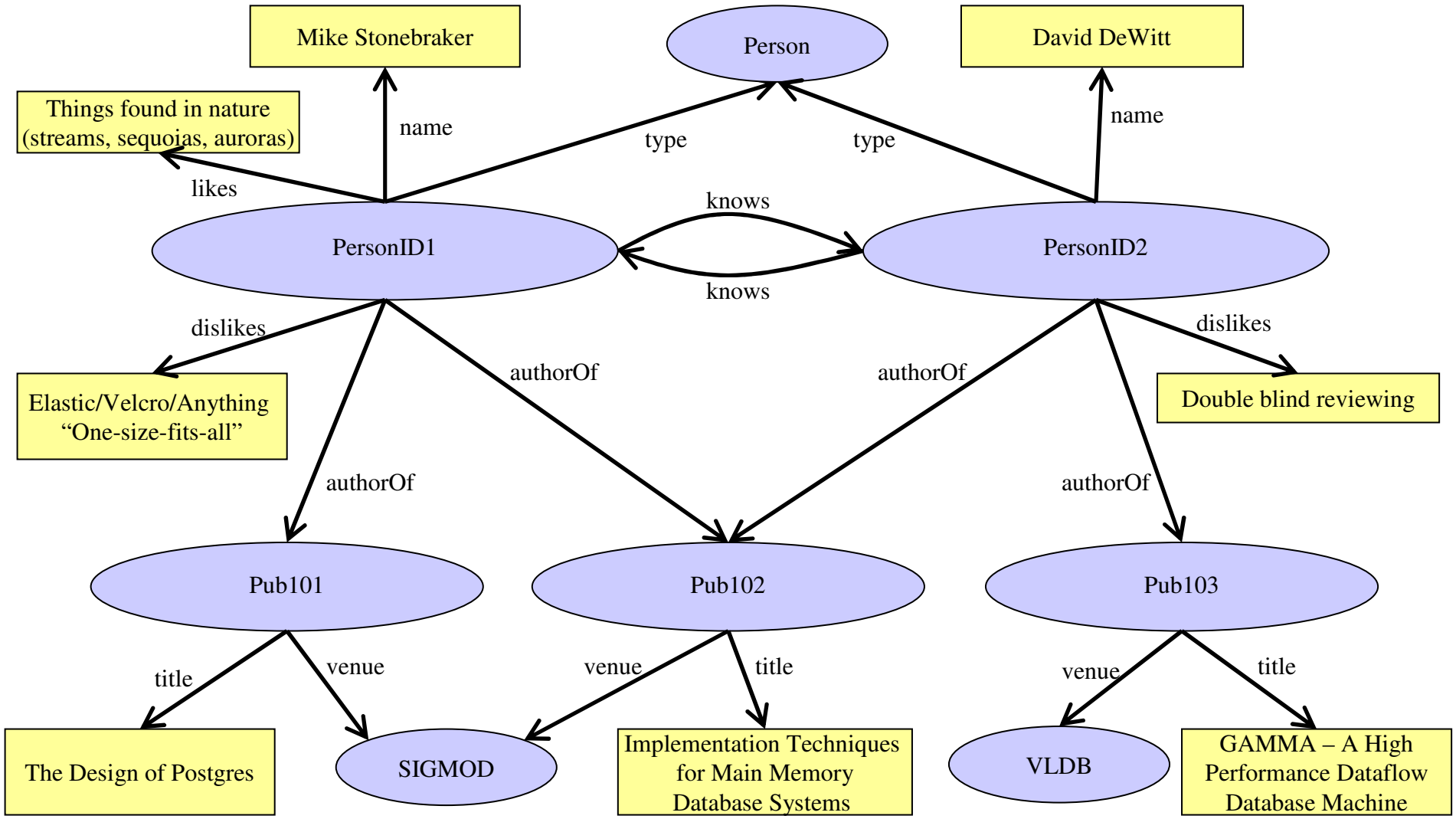
DBFacebook RDF Graph



Approach 1: Triple Stores

Subject	Property	Object
PersonID1	type	Person
PersonID1	name	"Mike Stonebraker"
PersonID1	likes	"Things found in nature (streams, sequoias, auroras)"
PersonID1	dislikes	"Elastic/Velcro/Anything 'One-size-fits-all'"
PersonID1	authorOf	Pub101
PersonID1	authorOf	Pub102
PersonID2	type	Person
PersonID2	name	"David DeWitt"
PersonID2	dislikes	"Double blind reviewing"
PersonID2	authorOf	Pub102
PersonID2	authorOf	Pub103
Pub101	title	"The Design of Postgres"
Pub101	venue	SIGMOD
Pub102	title	"Implementation Techniques for Main Memory Databases"
Pub102	venue	SIGMOD
Pub103	title	"GAMMA – A High Performance Dataflow Database"
Pub103	venue	VLDB

DBFacebook RDF Graph

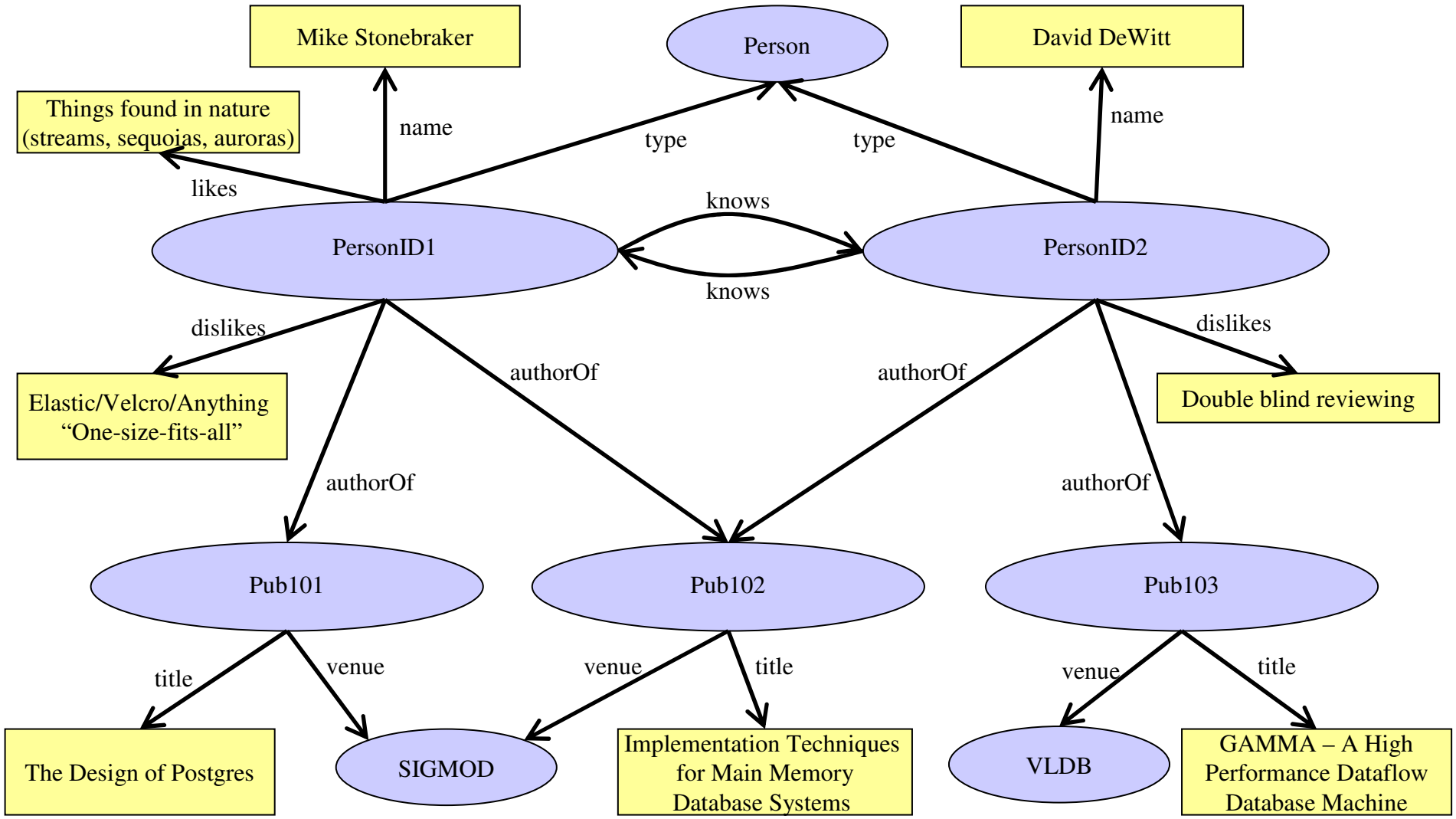


Approach 2: Property Tables

Subject	name	likes	dislikes
PersonID1	Mike Stonebraker	Things found in nature (streams, sequoias, auroras)	Elastic/Velcro/ Anything ‘One-size-fits-all’
PersonID2	David DeWitt	NULL	Double Blind Reviewing

Subject	title	venue
Pub101	“The Design of Postgres”	SIGMOD
Pub102	“Implementation Techniques for Main Memory Databases”	SIGMOD
Pub103	“GAMMA – A High Performance Dataflow Database”	SIGMOD

DBFacebook RDF Graph



Approach 3: One-table-per-property

name

Subject	Object
PersonID1	Mike Stonebraker
PersonID2	David DeWitt

dislikes

Subject	Object
PersonID1	Elastic/Velcro/ Anything ‘One-size-fits-all’
PersonID2	Double Blind Reviewing

likes

Subject	Object
PersonID1	Things found in nature (streams, sequoias, auroras)

authorOf

Subject	Object
PersonID1	Pub101
PersonID1	Pub102
PersonID2	Pub102
PersonID2	Pub103

Paper Contributions

- Explores advantages/disadvantages of these approaches
  - Triples stores are the dominant choice
  - Property Tables implemented by Jena and Oracle
  - We propose the one-table-per-property approach
- Shows how a column-store can be extended to implement the one-table-per-property approach
- Introduces benchmark for evaluating RDF stores

Results Synopsis

- Triple-store really slow on benchmark with 50M triples
- Property-tables and one-table-per-property approaches are factor of 3 faster
- One-table-per-property with column-store yields another factor of 10

Querying RDF Data

- SPARQL is the dominant language
- Examples:
  - SELECT ?name
    WHERE { ?x type Person .
    ?x name ?name }

  - SELECT ?likes ?dislikes
    WHERE { ?x title “Implementation Techniques for
    Main Memory Databases” .
    ?y authorOf ?x .
    ?y likes ?likes .
    ?y dislikes ?dislikes }

Translation to SQL over triples is easy

Subject	Property	Object
PersonID1	type	Person
PersonID1	name	“Mike Stonebraker”
PersonID1	likes	“Things found in nature (streams, sequoias, auroras)”
PersonID1	dislikes	“Elastic/Velcro/Anything ‘One-size-fits-all’”
PersonID1	authorOf	Pub101
PersonID1	authorOf	Pub102
PersonID2	type	Person
PersonID2	name	“David DeWitt”
PersonID2	dislikes	“Double blind reviewing”
PersonID2	authorOf	Pub102
PersonID2	authorOf	Pub103
Pub101	title	“The Design of Postgres”
Pub101	venue	SIGMOD
Pub102	title	“Implementation Techniques for Main Memory Databases”
Pub102	venue	SIGMOD
Pub103	title	“GAMMA – A High Performance Dataflow Database”
Pub103	venue	VLDB

SPARQL → SQL (over triple store)

- **Query 1 SPARQL:**

```
SELECT ?name
WHERE { ?x type Person .
        ?x name ?name }
```

- **Query 1 SQL:**

```
SELECT B.object
FROM triples AS A, triples as B
WHERE A.subject = B.subject
  AND A.property = "type"
  AND A.object = "Person"
  AND B.predicate = "name"
```

SPARQL → SQL (over triple store)

- **Query 2 SPARQL:**

```
SELECT ?likes ?dislikes
WHERE { ?x title “Implementation Techniques for
                  Main Memory Databases” .
        ?y authorOf ?x .
        ?y likes ?likes .
        ?y dislikes ?dislikes }
```

- **Query 2 SQL:**

```
SELECT C.object, D.object
FROM triples AS A, triples AS B, triples AS C, triples AS D
WHERE A.subject = B.object
  AND A.property = “title”
  AND A.object = “Implementation Techniques
                  for Main Memory Databases”
  AND B.property = “authorOf”
  AND B.subject = C.subject
  AND C.property = “likes”
  AND C.subject = D.subject
  AND D.property = “dislikes”
```

Triple Stores

- Accessing multiple properties for a resource require subject-subject joins
- Path expressions require subject-object joins
- Can improve performance by:
  - Indexing each column
  - Dictionary encoding string data
- Ultimately: Do not scale

Property Tables Can Reduce Joins

Subject	name	likes	dislikes
PersonID1	Mike Stonebraker	Things found in nature (streams, sequoias, auroras)	Elastic/Velcro/ Anything ‘One-size-fits-all’
PersonID2	David DeWitt	NULL	Double Blind Reviewing

Left-over triples

Subject	Property	Object
PersonID1	authorOf	Pub101
PersonID1	authorOf	Pub102
PersonID2	authorOf	Pub102
PersonID2	authorOf	Pub103
…	…	…

Property Tables

- Complex to design
  - If narrow: reduces nulls, increases unions/joins
  - If wide: reduces unions/joins, increases nulls
- Implemented in Jena and Oracle
  - But main representation of data is still triples

Table-Per-Property Approach

name

Subject	Object
PersonID1	Mike Stonebraker
PersonID2	David DeWitt

dislikes

Subject	Object
PersonID1	Elastic/Velcro/ Anything ‘One-size-fits-all’
PersonID2	Double Blind Reviewing

likes

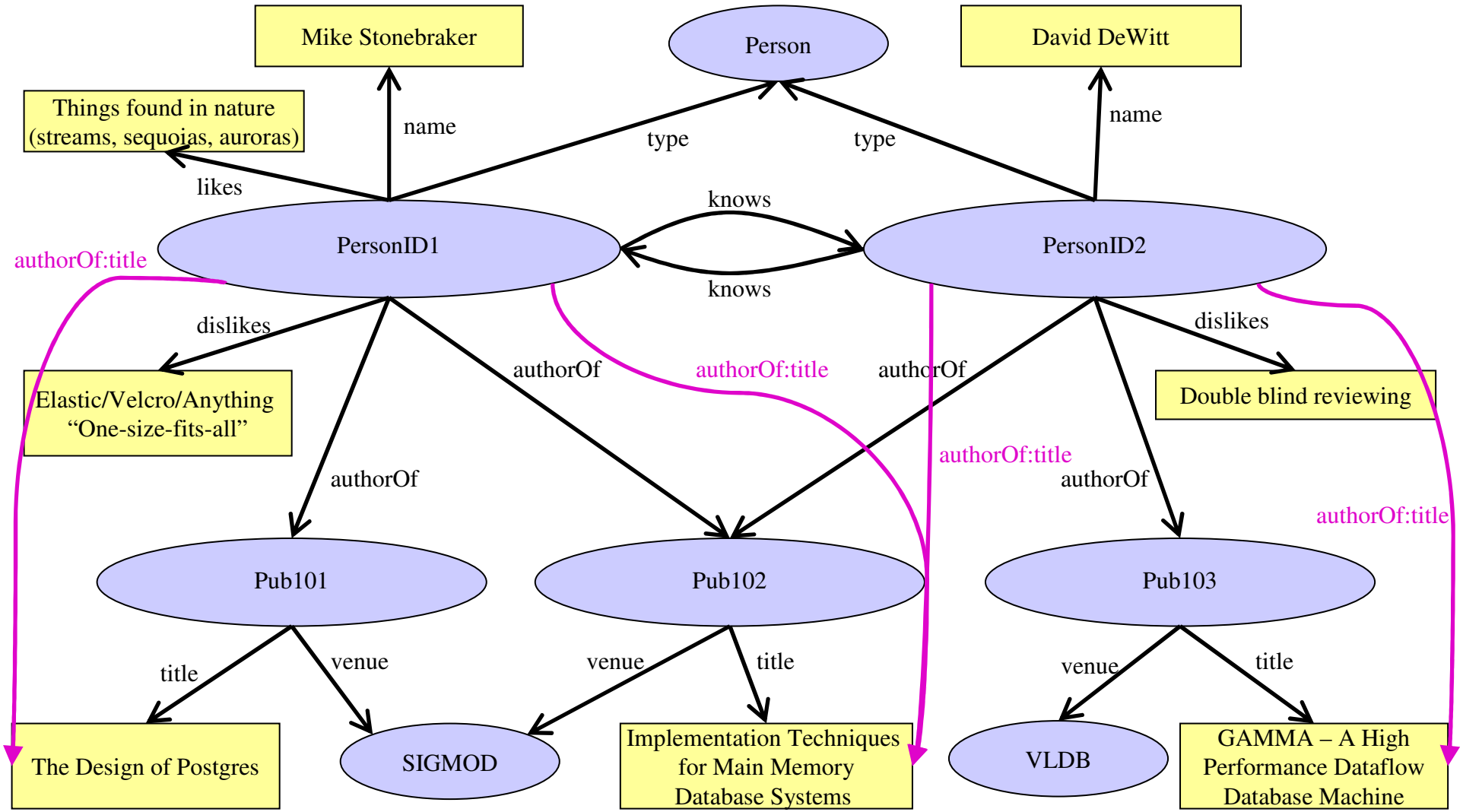
Subject	Object
PersonID1	Things found in nature (streams, sequoias, auroras)

authorOf

Subject	Object
PersonID1	Pub101
PersonID1	Pub102
PersonID2	Pub102
PersonID2	Pub103

- + Nulls not stored
- + Easy to handle multi-valued attributes
- + Only need to read relevant properties
- – Still need joins (but they are linear merge joins)

Materialized Paths



Accelerating Path Expressions

- Materialize Common Paths
  - Improved property table performance by 18-38%
  - Improved one-table-per-property performance by 75-84%
- Use automatic database designer (e.g., C-Store /Vertica) to decide what to materialize

Subject	authorOf:title
PersonID1	The Design of the Postgres
PersonID1	Implementation Techniques for Main Memory Database Systems
PersonID2	Implementation Techniques for Main Memory Database Systems
PersonID2	GAMMA – A High Performance Dataflow Database Machine

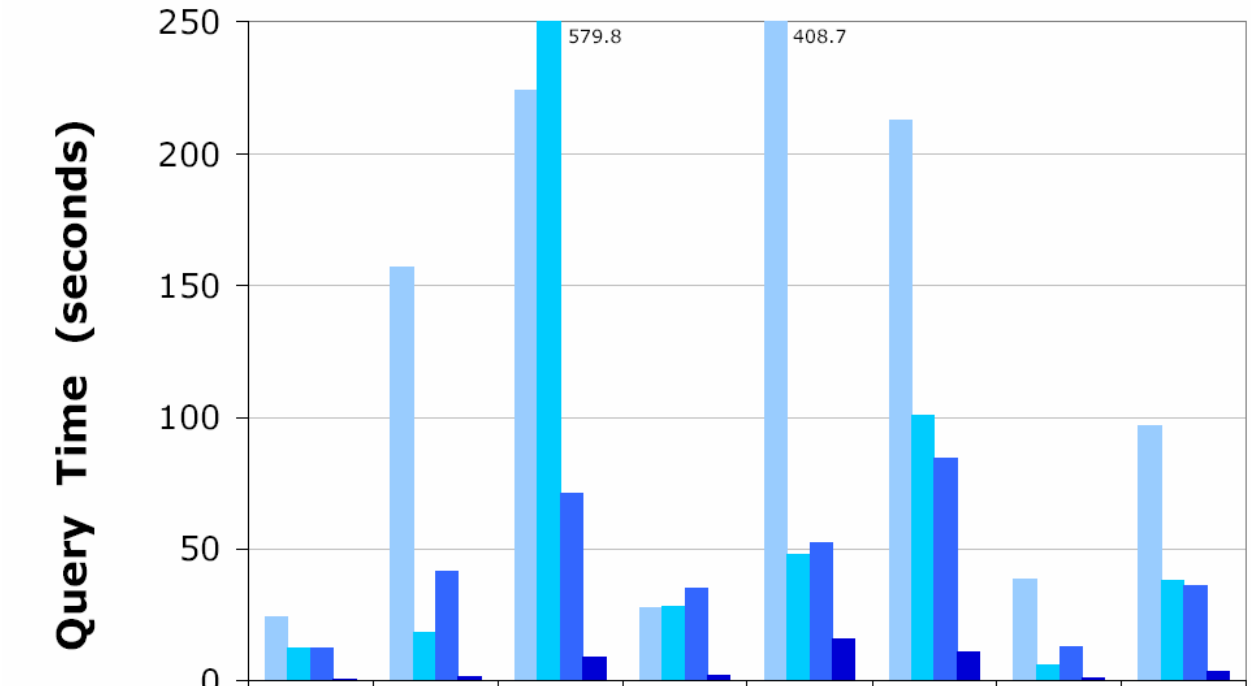
One-table-per-property → Column-Store

- Can think of one-table-per-property as vertical partitioning super-wide property table
- Column-store is a natural storage layer to use for vertical partitioning
- Advantages:
  - Tuple Headers Stored Separately
  - Column-oriented data compression
  - Do not necessarily have to store the subject column
  - Carefully optimized merge-join code

Library Benchmark

- Data
  - Real Library Data (50 million RDF triples)
  - Data acquired from a variety of diverse sources (some quite unstructured)
- Queries
  - Automatically generated from the Longwell RDF browser
- Details in paper …

Results



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Geo. Mean
Triple Store	24.63	157	224.3	27.67	408.7	212.7	38.37	97
Prop. Table	12.66	18.37	579.8	28.54	47.85	101	6.1	38
Vert. Part.	12.66	41.7	71.3	35.49	52.34	84.6	13.25	36
C-Store	0.66	1.64	9.28	2.24	15.88	10.81	1.44	3

Conclusions and Future Work

- Experimented with storing RDF data using different schemas in RDMS (both row and column-oriented)
- Future work: build a fully-functional RDF database
  - Extracts and loads RDF data from structured, semi-structured, and unstructured data sources
  - Translates SPARQL to queries over vertical schema
  - Performs reasoning inside the DB
  - Use with biology research
- Excited about this work? Then …

Come To Yale!