Research Principles Revealed

Jennifer Widom

Stanford University

But First, Some Thanks

- Four Extra-Special People
- Superb Students
- Terrific Collaborators

Extra-Special #1

Laura Haas

- Hired a PL/logic person with minimal DB experience
- The Perfect Manager
  - Mentored instead of managed
  - Ensured I could devote nearly all of my time to research
  - Sported a great button

Extra-Special #2

Stefano Ceri

- Incredible run of summer collaborations (IBM and Stanford)
- Jennifer ∧ Stefano ⇨ Success

Details **Intuition**

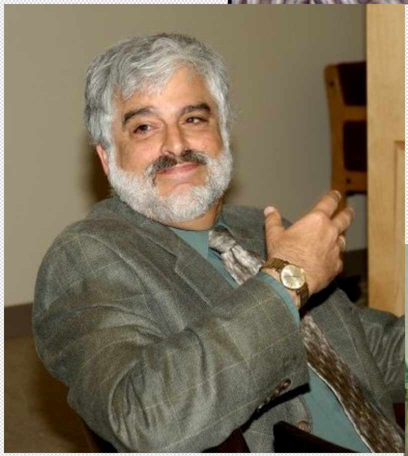
Extra-Special #3 and #4

Hector Garcia-Molina and Jeff Ullman

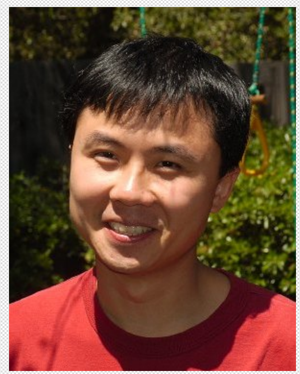
- Colleagues, mentors, book co-authors
- Neighbors, baby-sitters, sailing crew, kids sports photographers, …

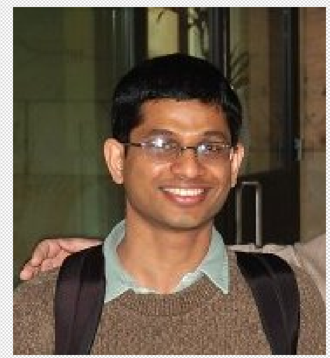
{ Hector, Jeff, Jennifer }

- Research collaborations in all 2^3 subsets



Superb Ph.D. Students





Terrific Collaborators*

Serge Abiteboul
Brian Babcock
Elena Baralis
Omar Benjelloun
Sudarshan Chawathe
Bobbie Cochrane
Shel Finkelstein
Alon Halevy
Rajeev Motwani
Anand Rajaraman
Shuky Sagiv
Janet Wiener

* Significant # co-authored papers in DBLP

Now to the "Technical" Part …

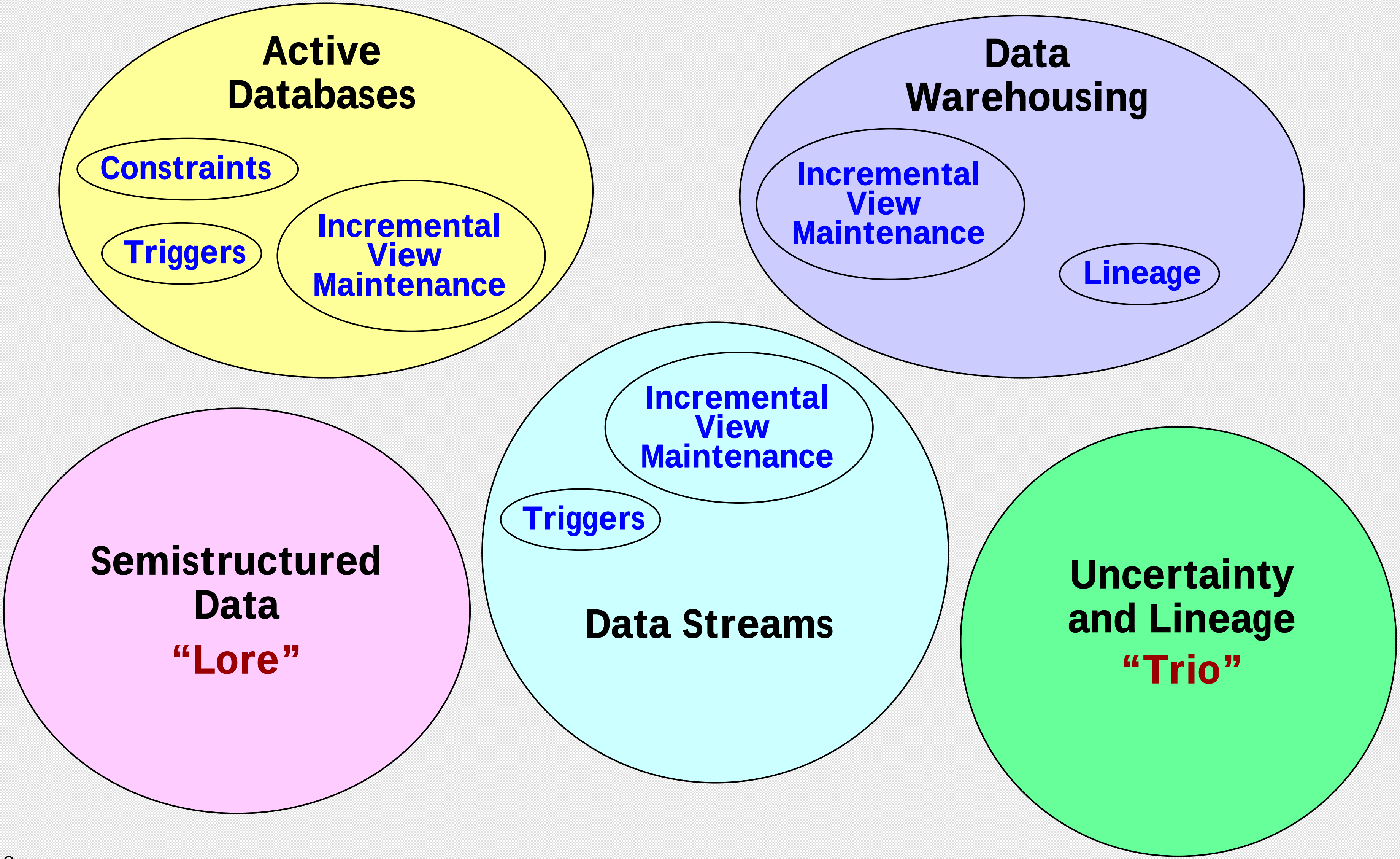
Research Principles Revealed

1. Topic Selection
2. The Research
3. Dissemination

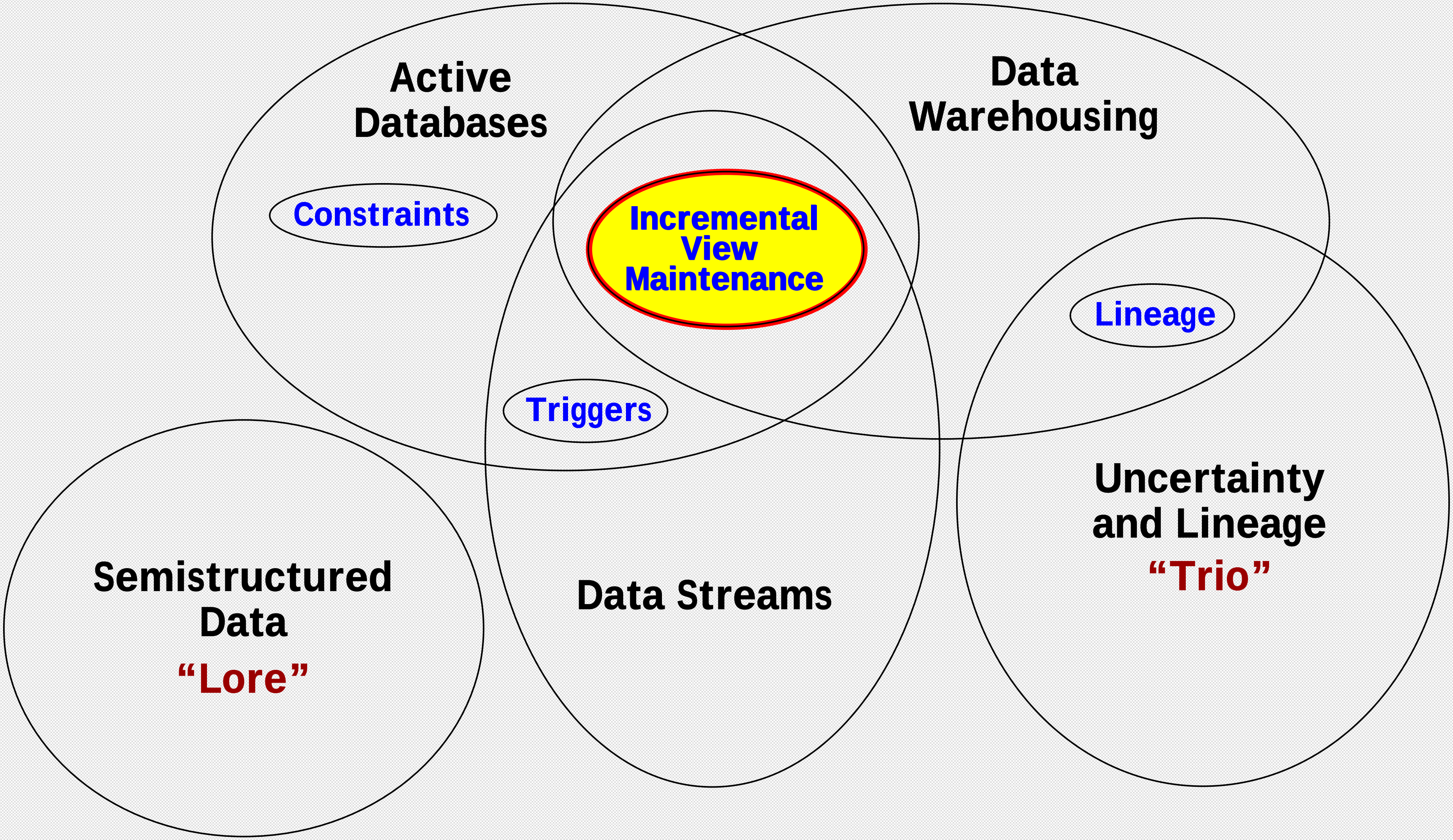
Disclaimer

These principles work for me.
Your mileage may vary!

Major Research Areas



Major Research Areas



Finding Research Areas

I’m not a visionary
(In fact, I’m “anti-visionary”)

- Never know what my next area will be
- Some combination of “gut feeling” and luck

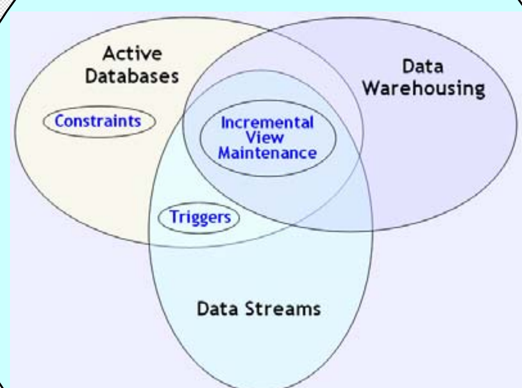
Finding Research Areas

Active Databases

Data Warehousing

Semistructured Data

Data Streams



Uncertainty and Lineage

Finding Research Topics

One recipe for a successful database research project

- Pick a simple but fundamental assumption underlying traditional database systems

  Drop it

- Must reconsider all aspects of data management and query processing
  - Many Ph.D. theses
  - Prototype from scratch

Finding Research Topics

Example “simple but fundamental assumptions”

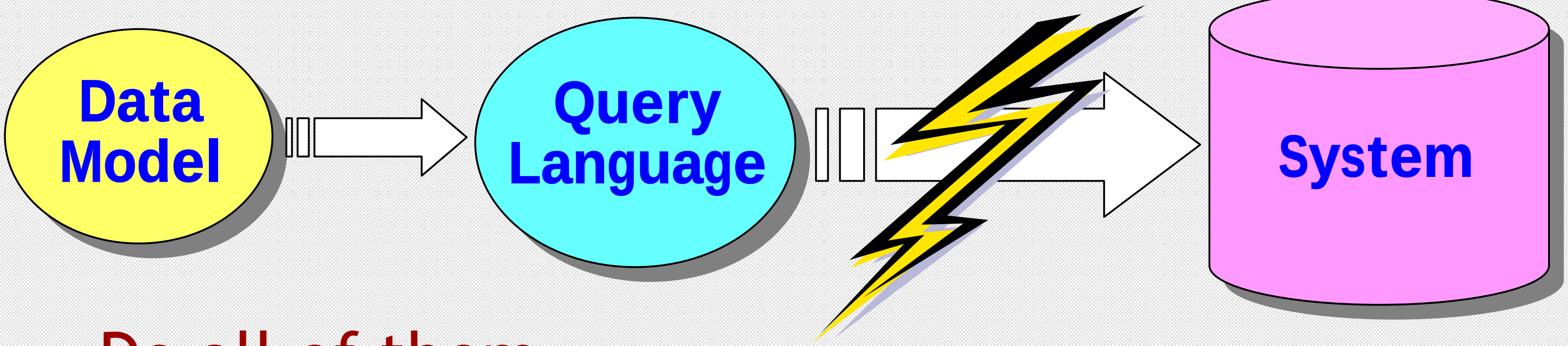
- Schema declared in advance **Semistructured data**
- Persistent data sets **Data streams**
- Tuples contain values **Uncertain data**

Reconsidering “all aspects”

- Data model
- Query language
- Storage and indexing structures
- Query processing and optimization
- Concurrency control, recovery
- Application and user interfaces

The Research Itself

Critical triple for any new kind of database system



- Do all of them
- In this order
- Cleanly and carefully (a research luxury)

➔ Solid foundations, then implementation

Nailing Down a New Data Model

Cleanly and carefully

Nailing Down a New Data Model

Example: "A data stream is an unbounded sequence of [tuple timestamp] pairs"

Temperature Sensor 1:

[(72) 2:05] [(75) 2:20] [(74) 2:21] [(74) 2:24] [(81) 2:45] …

Temperature Sensor 2:

[(73) 2:03] [(76) 2:20] [(73) 2:22] [(75) 2:22] [(79) 2:40] …

Nailing Down a New Data Model

Example: "A data stream is an unbounded sequence of [tuple timestamp] pairs"

Temperature Sensor 1:

[(72) 2:05] [(75) 2:20] [(74) 2:21] [(74) 2:24] [(81) 2:45] …

Temperature Sensor 2:

[(73) 2:03] [(76) 2:20] **[(73) 2:22] [(75) 2:22]** [(79) 2:40] …

- Duplicate timestamps in streams?
- If yes, is order relevant?

Nailing Down a New Data Model

Example: "A data stream is an unbounded sequence of [tuple timestamp] pairs"

Temperature Sensor 1:

[(72) 2:05] **[(75) 2:20]** [(74) 2:21] **[(74) 2:24]** [(81) 2:45] …

Temperature Sensor 2:

[(73) 2:03] **[(76) 2:20]** [(73) 2:22] **[(75) 2:22]** [(79) 2:40] …

- Are timestamps coordinated across streams?

  Duplicates? Order relevant?

Nailing Down a New Data Model

Example: "A data stream is an unbounded sequence of [tuple timestamp] pairs"

Temperature Sensor 1:

[(72) 2:05] [(75) 2:20] [(74) 2:21] [(74) 2:24] [(81) 2:45] …

Temperature Sensor 2:

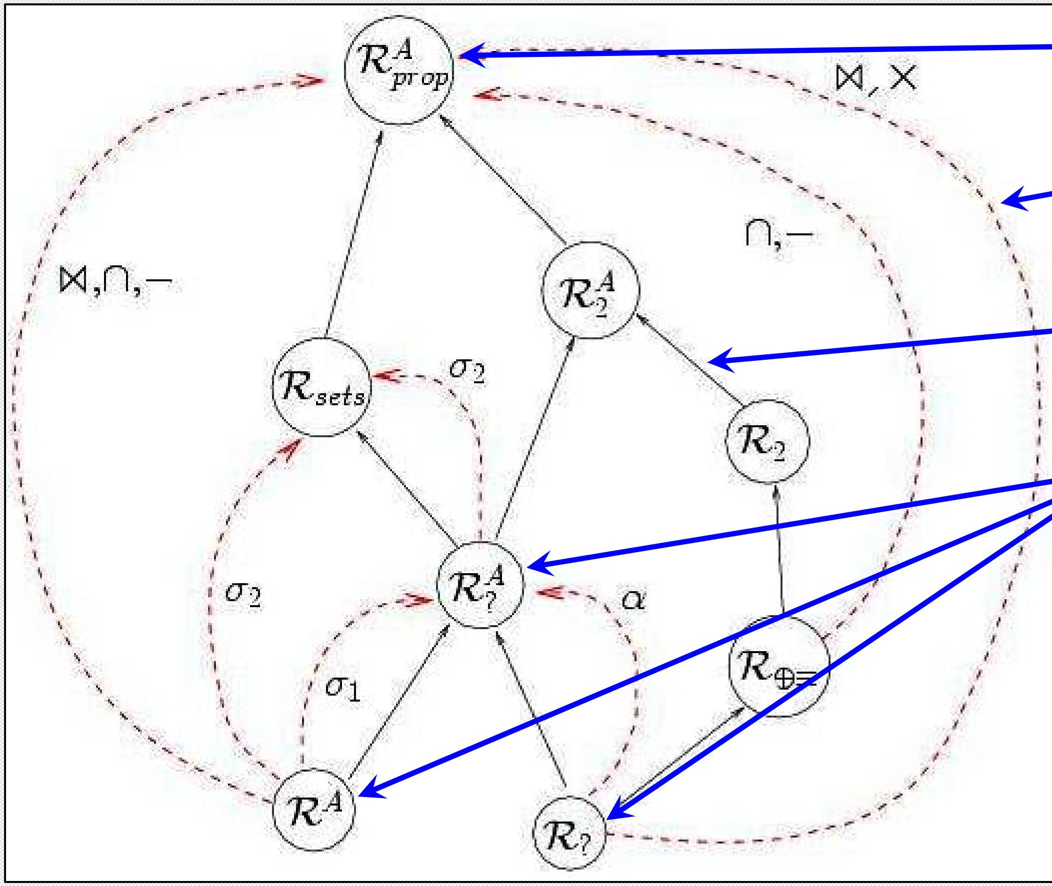
[(73) 2:03] [(76) 2:20] [(73) 2:22] [(75) 2:22] [(79) 2:40] …

Sample Query (continuous)

"Average discrepancy between sensors"

Result depends heavily on model

Data Model for Trio Project



Only "complete" model

Closure properties

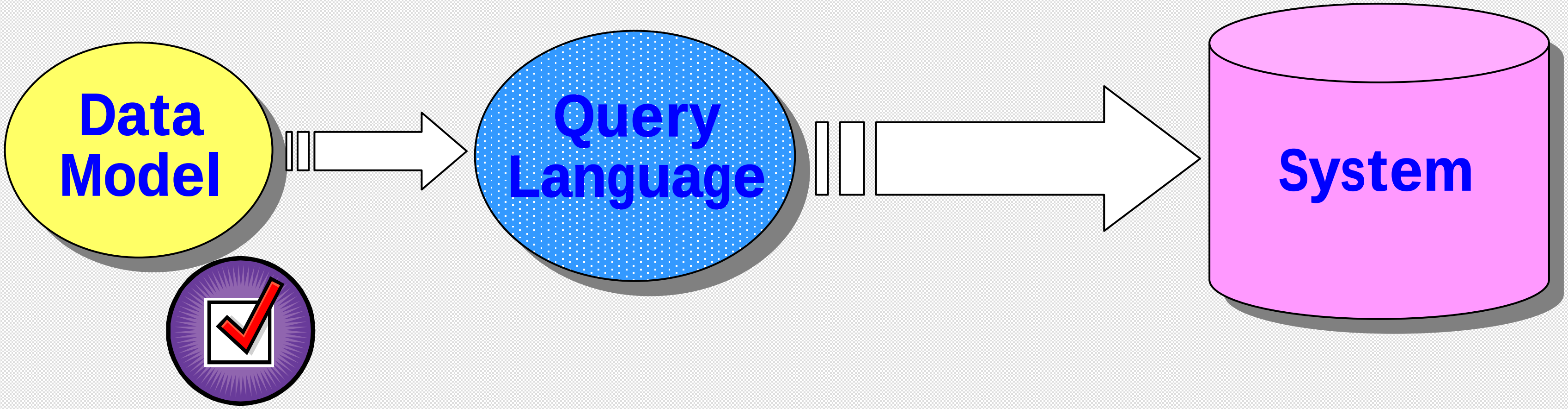
Relative expressiveness

Only understandable models

In the end, lineage saved the day

R Possible models

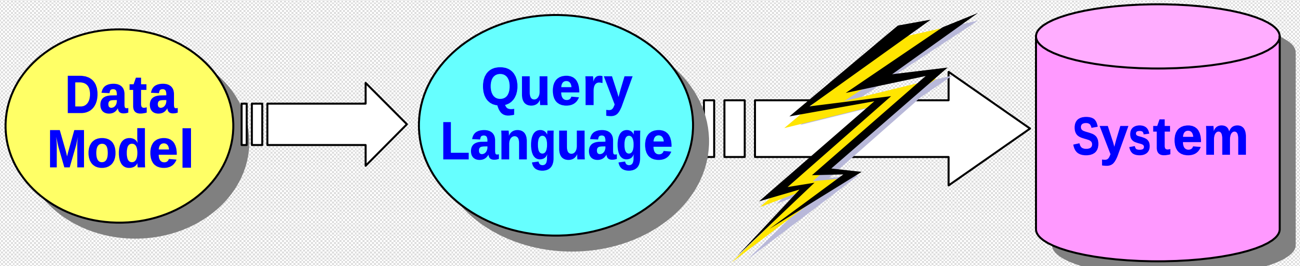
The Research Triple



Query Language Design

- Notoriously difficult to publish
- But potential for huge long-term impact

- Semantics can be surprisingly tricky



- Cleanly and carefully

➔ Solid foundations, then implementation

The IBM-Almaden Years

Developing an active rule (trigger) system

Transition tables,
Conflicts,
Confluence, …

"Write
Code!"

"We finished
our rule system
ages ago"

The IBM-Almaden Years

Developing an active rule (trigger) system

"Yeah, but what does it do?"

"We finished our rule system *ages* ago"

The IBM-Almaden Years

Developing an active rule (trigger) system

"Yeah, but what does it do?"

"Umm … I'll need to run it to find out"

The IBM-Almaden Years

Developing an active rule (trigger) system

Disclaimer

These principles work for me.
Your mileage may vary.

Tricky Semantics Example #1

Semistructured data (warm-up)

Query: SELECT Student WHERE Advisor='Widom'

```
<Student>
   <ID> 123 </ID>
   <Name> Susan </Name>
   <Major> CS </Major>
</Student>
<Student>
      ...
</Student>
```

- Error?
- Empty result?
- Warning?

Tricky Semantics Example #1

Semistructured data (warm-up)

Query: SELECT Student WHERE Advisor='Widom'

```
<Student>
   <ID> 123 </ID>
   <Name> Susan </Name>
   <Major> CS </Major>
</Student>
<Student>
     ...
</Student>
```

Lore

- Empty result
- Warning

Tricky Semantics Example #1

Semistructured data (warm-up)

Query: SELECT Student WHERE Advisor='Widom'

```
<Student>
  <ID> 123 </ID>
  <Advisor> Garcia </Advisor>
  <Advisor> Widom </Advisor>
</Student>
<Student>
    ...
</Student>
```

Lore

Implicit ∃

Tricky Semantics Example #2

Trigger 1:	WHEN X makes sale > 500
	THEN increase X's salary by 1000

Trigger 2:	WHEN average salary increases > 10%
	THEN increase everyone's salary by 500

Inserts: Sale(Mary,600) Sale(Mary,800) Sale(Mary,550)

- How many increases for Mary?
- If each causes average > 10%, how many global raises?
- What if global raise causes average > 10%?

Tricky Semantics Example #3

Temperature Sensor:

[(72) 2:00] [(74) 2:00] [(76) 2:00] [(60) 8:00] [(58) 8:00] [(56) 8:00]

Query (continuous):

Average of most recent three readings

Tricky Semantics Example #3

Temperature Sensor:
[(72) 2:00] [(74) 2:00] [(76) 2:00] [(60) 8:00] [(58) 8:00] [(56) 8:00]

Query (continuous):
Average of most recent three readings

System A: 74, 58

Tricky Semantics Example #3

Temperature Sensor:

[(72) 2:00] [(74) 2:00] [(76) 2:00] [(60) 8:00] [(58) 8:00] [(56) 8:00]

Query (continuous):

Average of most recent three readings

System A: 74, 58

System B: 74, 70, 64.7, 58

The "It's Just SQL" Trap

Tables: **Sigmod**(year,loc,…) **Climate**(loc,temp,…)

Query: Temperature at SIGMOD 2010

```
SELECT S.temp
FROM   Sigmod S, Climate C
WHERE S.loc = C.loc AND S.year = 2010
```

Sigmod (year, *loc*)	
2010	***London* ‖ *New York***

Climate (loc, *temp*)	
London	**[*55 – 68*]**
New York	**[*64 – 79*]**

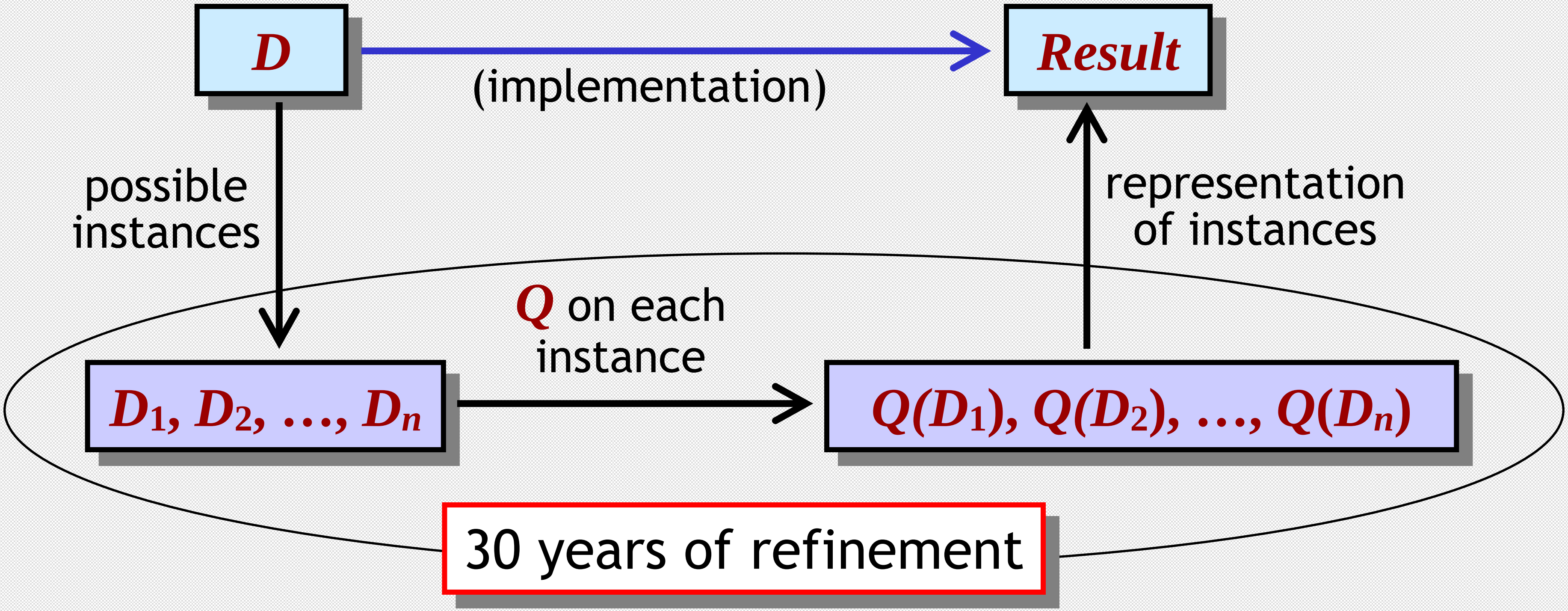
The “It’s Just SQL” Trap

- Syntax is one thing (actually it’s nothing)
- Semantics is another, as we’ve seen
  - Semistructured
  - Continuous
  - Uncertain
  - <Insert future new model here>

Taming the Semantic Trickiness

- Reuse existing (relational) semantics whenever possible

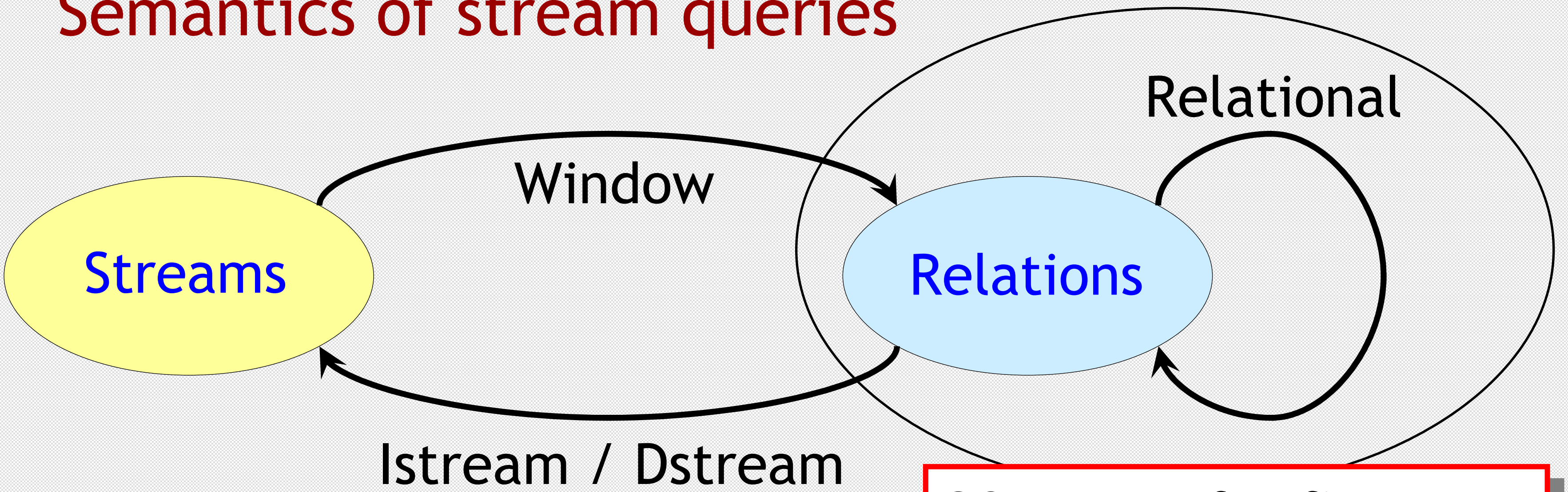
Uncertain data — semantics of query ***Q***



Taming the Semantic Trickiness

- Reuse existing (relational) semantics whenever possible

Semantics of stream queries



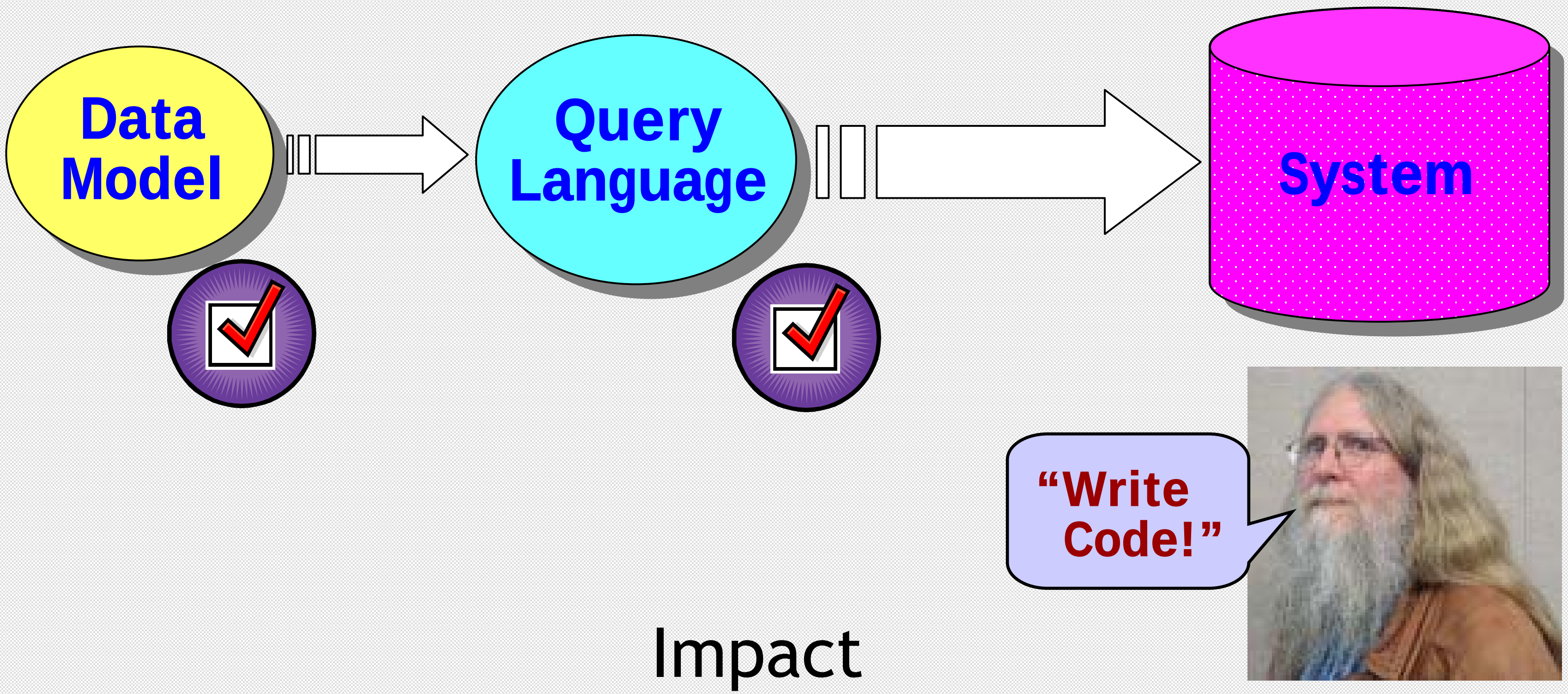
30 years of refinement

Taming the Semantic Trickiness

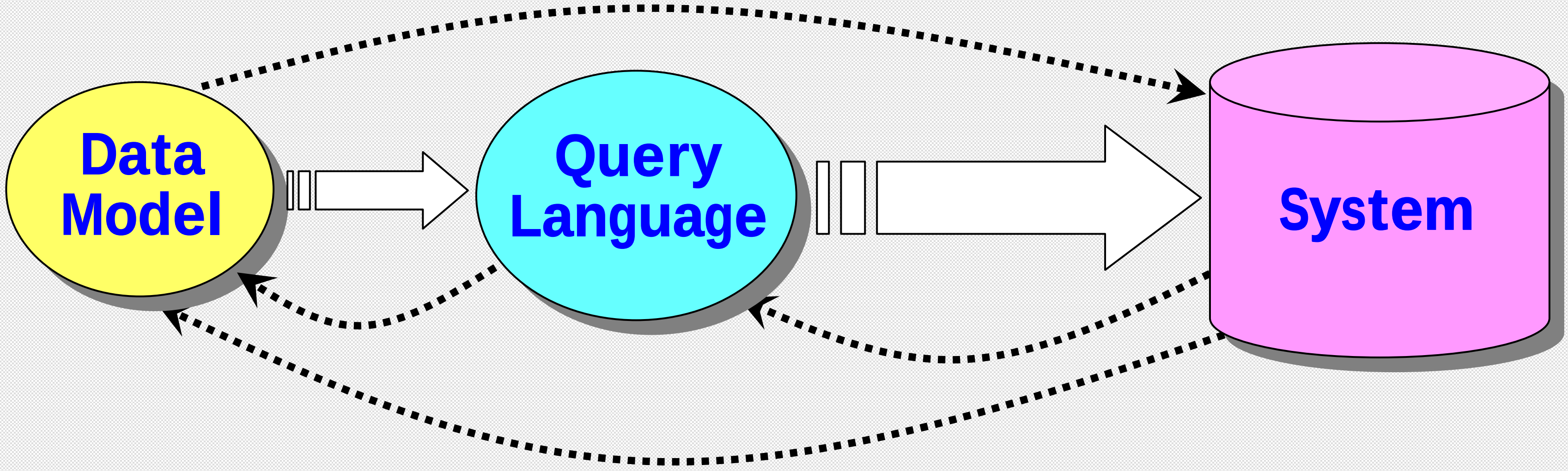
- Reuse existing (relational) semantics whenever possible
  - Active databases: “transition tables”
  - Lore: semantics based on OQL

3 years of refinement

The Research Triple



Truth in Advertising



- As research evolves, always revisit all three
- Cleanly and carefully!

Disseminating Research Results

- ✶ If it's important, don't wait
  - No place for secrecy (or laziness) in research
  - Every place for being first with new idea or result
- Post on Web, inflict on friends
- SIGMOD/VLDB conferences are not the only place for important work

  Send to workshops, SIGMOD Record, …
- Make software available and easy to use

  Decent interfaces, run-able over web

Summary: Five Points

1 Don’t dismiss the **Intuition** types (intuition ≠ visionary)

And don’t forget the **Details**

2 Data Model + Query Language + System

Solid foundations, then implementation

3 QL semantics: surprisingly tricky

Reuse existing (relational) semantics whenever possible

Summary: Five Points

4 Don't be secretive or lazy

Disseminate ideas, papers, and software

5 If all else fails, try stirring in the key ingredient:

Incremental View Maintenance

Thank You

Serge Abiteboul
Brian Babcock
Elena Baralis
Omar Benjelloun
Sudarshan Chawathe
Bobbie Cochrane

Shel Finkelstein
Alon Halevy
Rajeev Motwani
Anand Rajaraman
Shuky Sagiv
Janet Wiener