The *Why* and *Where* of Data Provenance

Peter Buneman
Sanjeev Khanna
Wang-Chiew Tan

University of Pennsylvania

DL-2 grant IIS 98-17444

http://db.cis.upenn.edu

http://db.cis.upenn.edu/Research/provenance.html

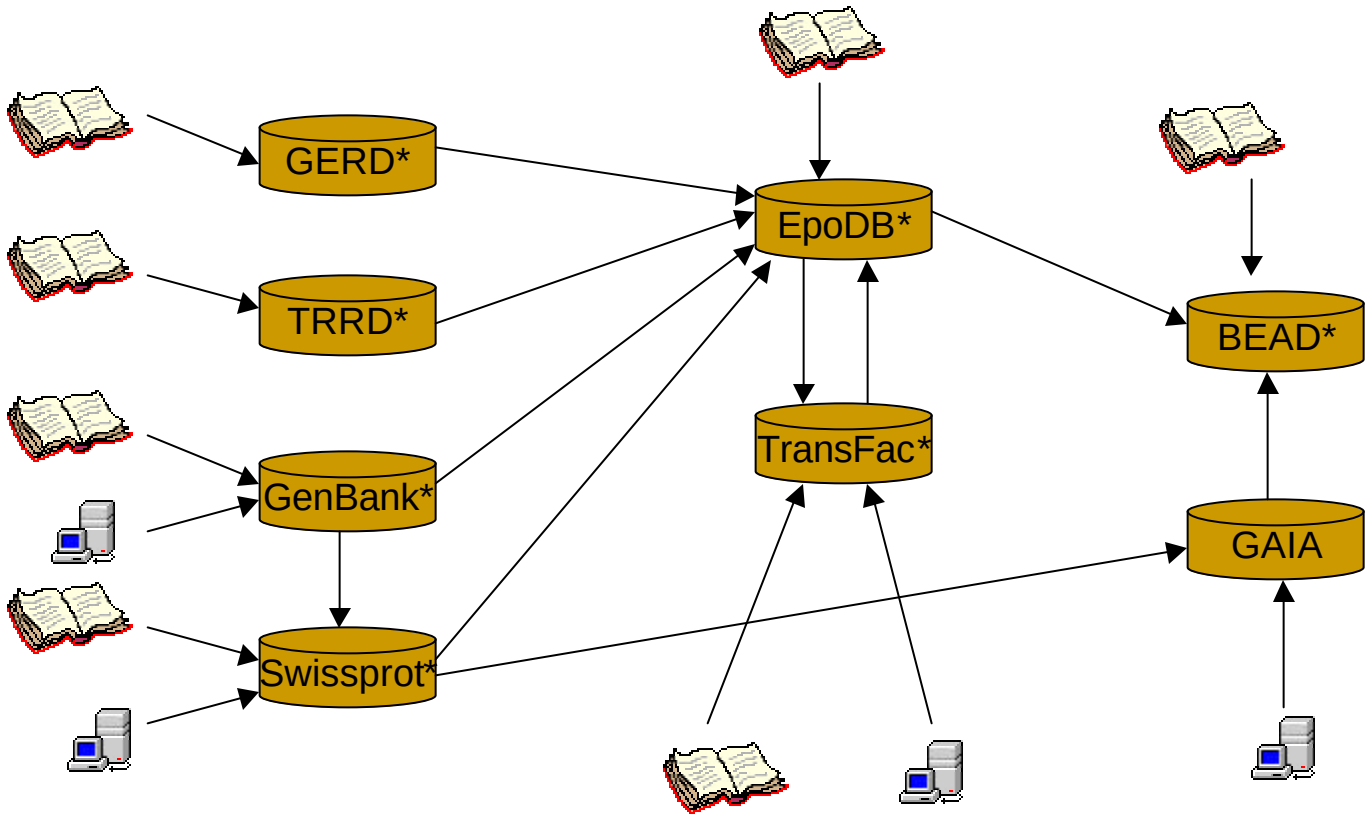
Data Provenance

- When you see some data on the Web, do you know
  - where it came from?
  - why is it there?
- This information (provenance) is typically lost in the process of copying/transforming databases
- Loss of provenance is an acute problem in some scientific databases

Provenance of Data Derived via Queries

- 500 databases in molecular biology
  - only a handful are source data
  - rest are materialized views

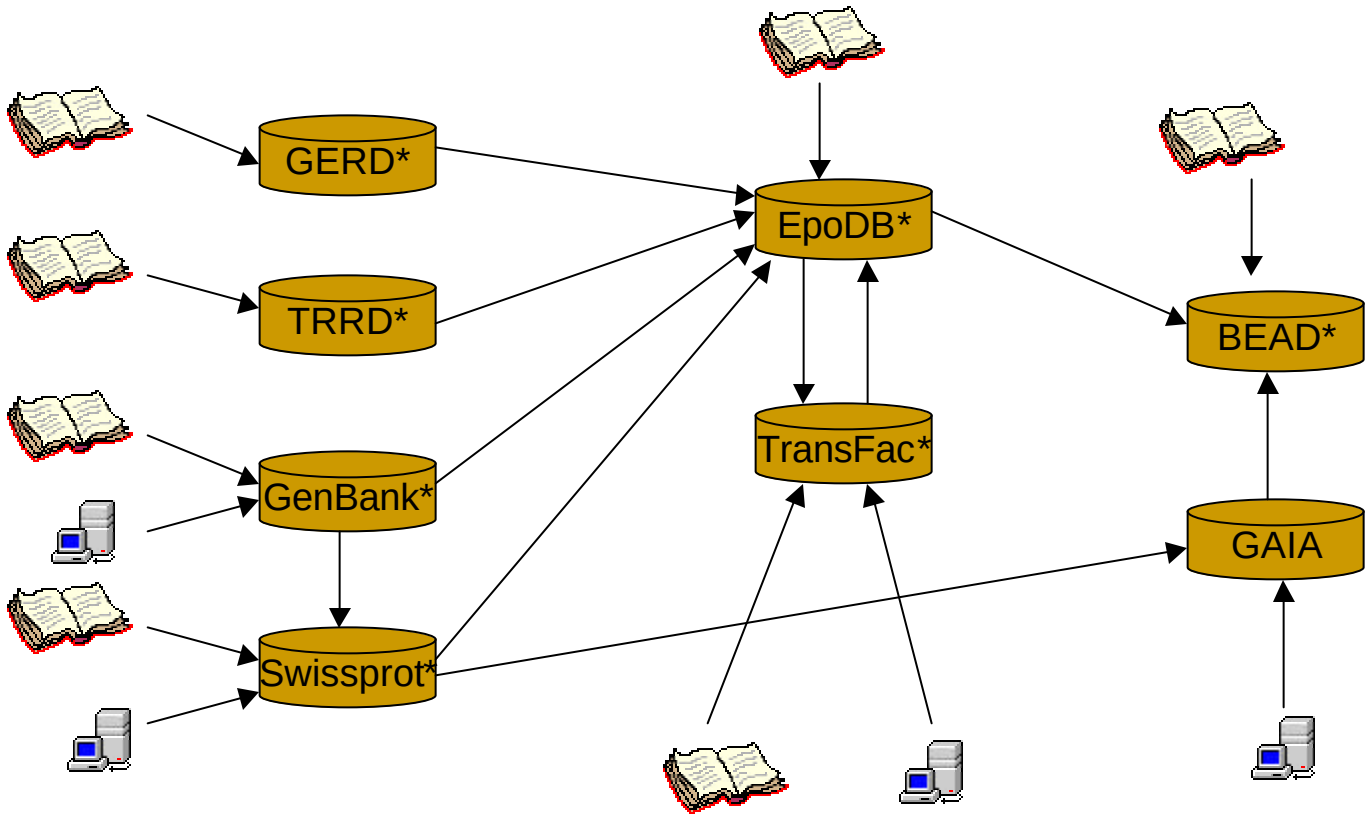
Database Inter-dependence is Complex



Provenance of Data Derived via Queries

- 500 databases in molecular biology
  - only a handful are source data
  - rest are materialized views
- We see a piece of data in some view
  - the origins of this piece of data
  - the process by which it was included
- If the data comes from a curated database, can we carry through relevant annotations ?

Database Inter-dependence is Complex

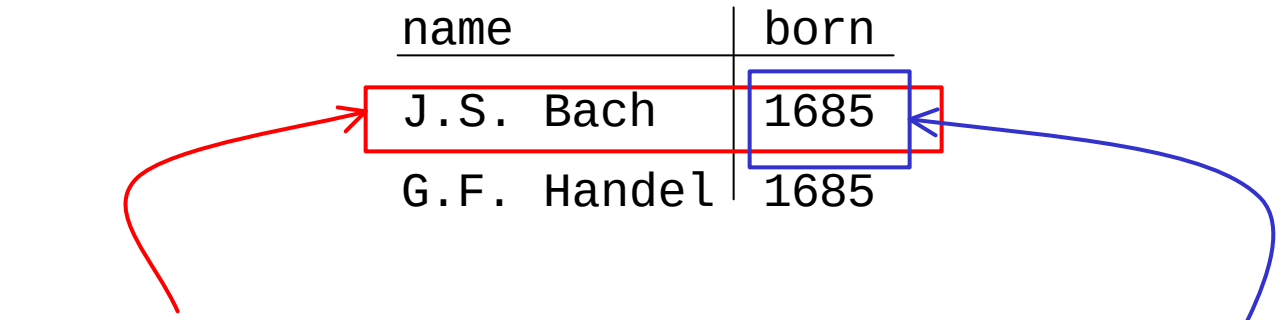


Two kinds of provenance

name	born	period
J.S. Bach	1685	baroque
G.F. Handel	1685	baroque
W.A. Mozart	1756	classical

```
SELECT name, born
FROM   composer
WHERE  born < 1700
```

name	born
J.S. Bach	1685
G.F. Handel	1685



Why is this tuple in the database?

Where does this value come from?

Why vs. Where-Provenance

SELECT name, telephone
FROM employee
WHERE salary > SELECT AVERAGE salary FROM employee

- What specific part of input data contributed to my inclusion in the output ?
  - the entire employee database
- Given that I am in the output, where did my telephone number come from ?
  - just my record in the employee database

The problem at hand

- Given a database $D=Q(D_1,\ldots,D_n)$ and a component c of D, what is the "why" and "where" provenance of c?

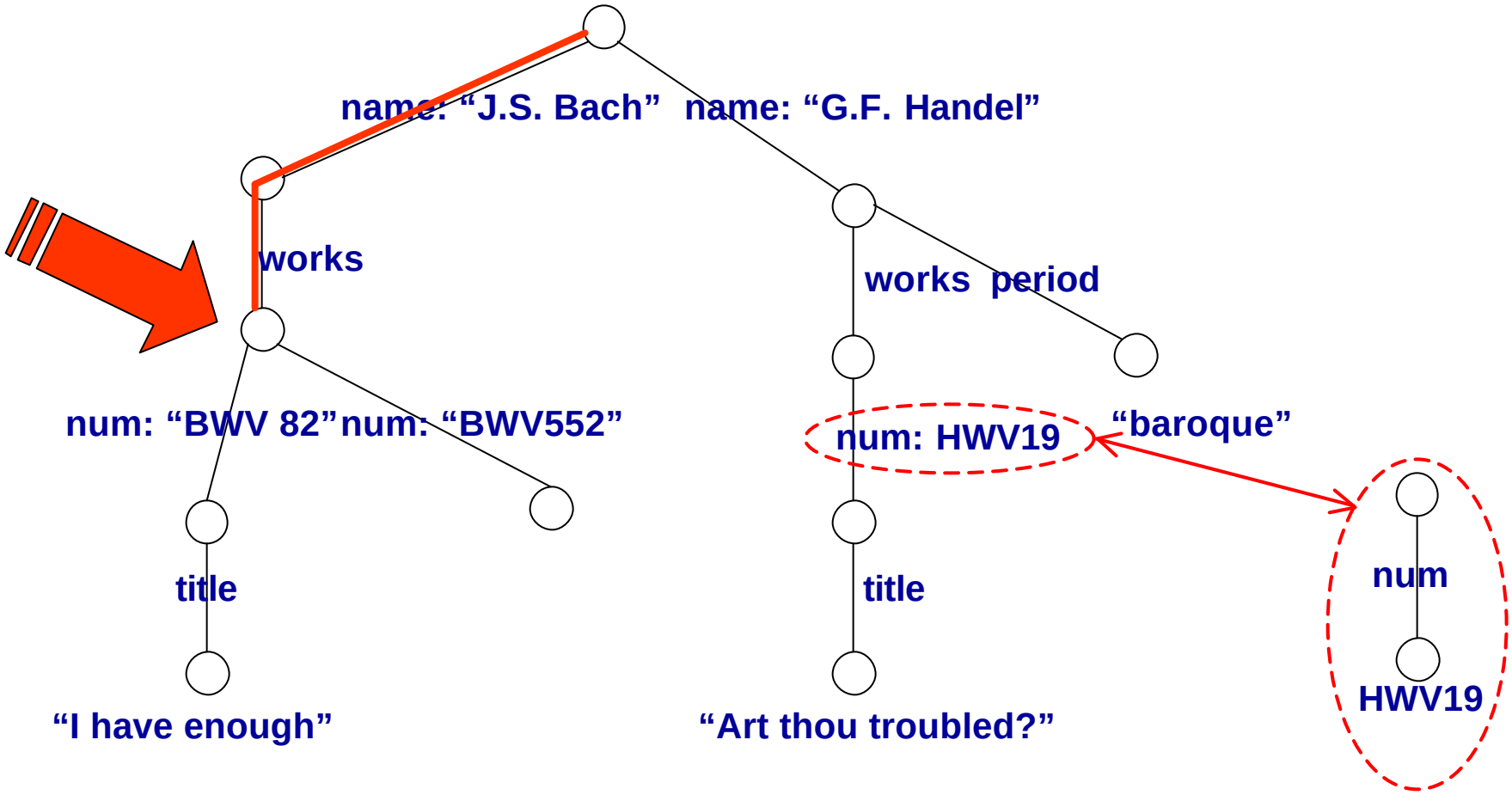
- We expect the answer, in each case, to be a set of components of $D_1,\ldots,D_n$
  - What are the relevant components ?
  - How do we identify them ?

The Goals of this Project

Construct data that knows
why it exists and where it came from

- Describe provenance
- Infer provenance
- Annotate data with provenance

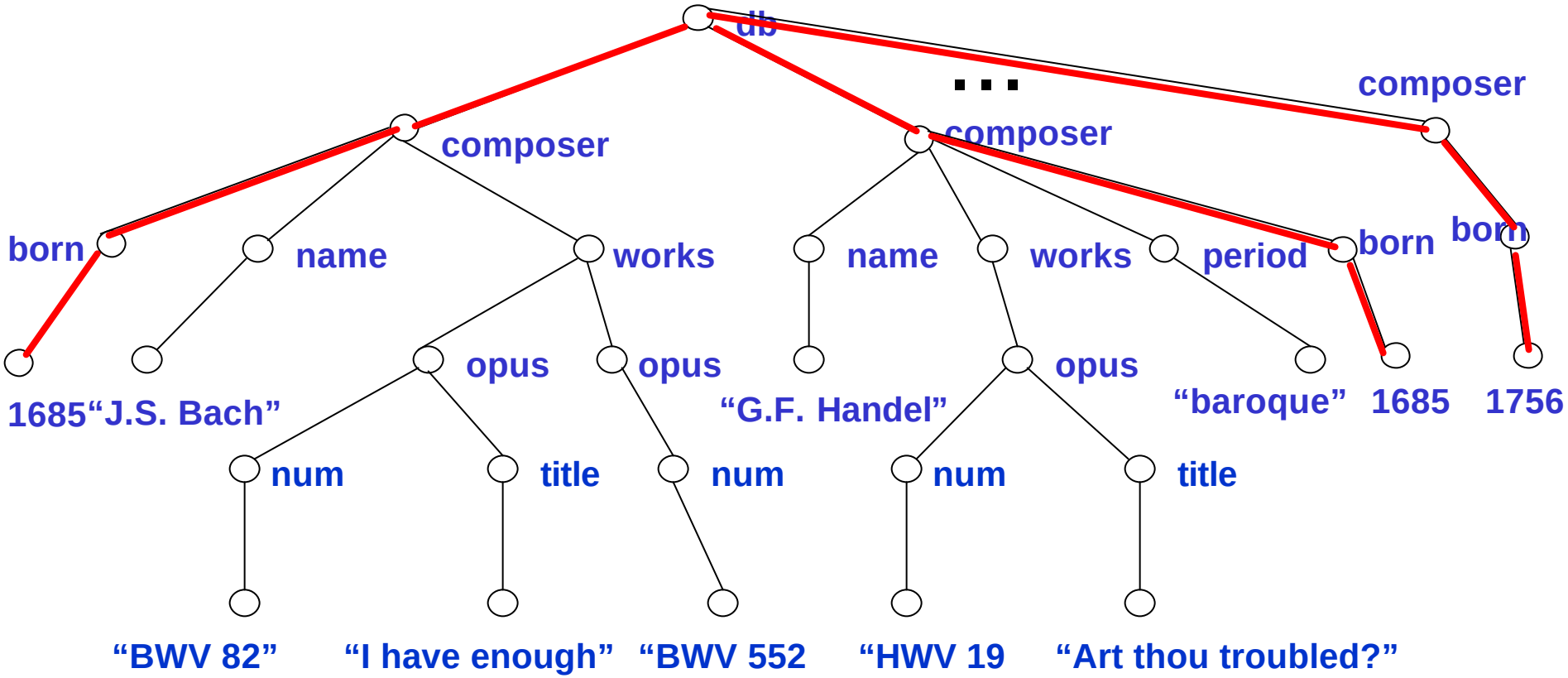
Semistructured data -- deterministic model



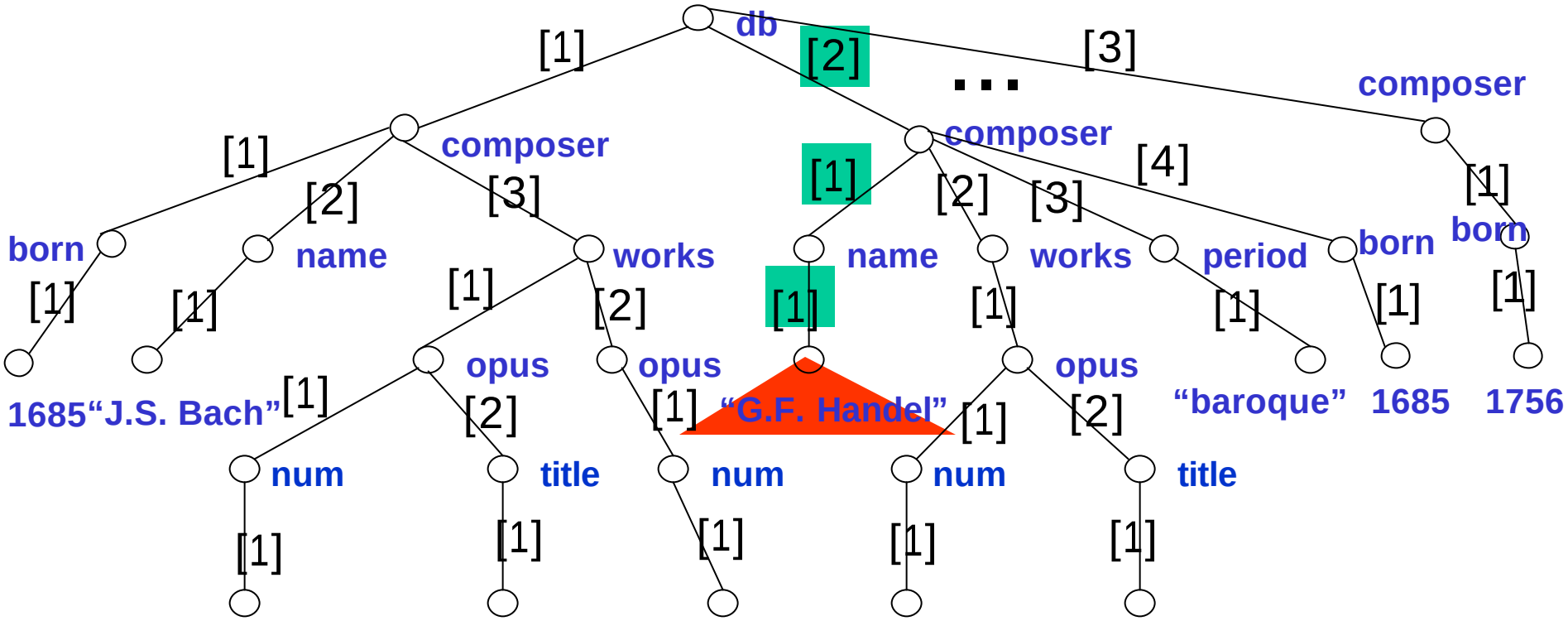
A *deterministic* model for semi-structured (and structured) data

- Based on common model of semistructured data as an edge-labeled graph.
- Less general
  - Deterministic (outgoing edges are distinct)
- More general
  - Labels can have structure

Relationship with XML



Relationship with XML



An Example

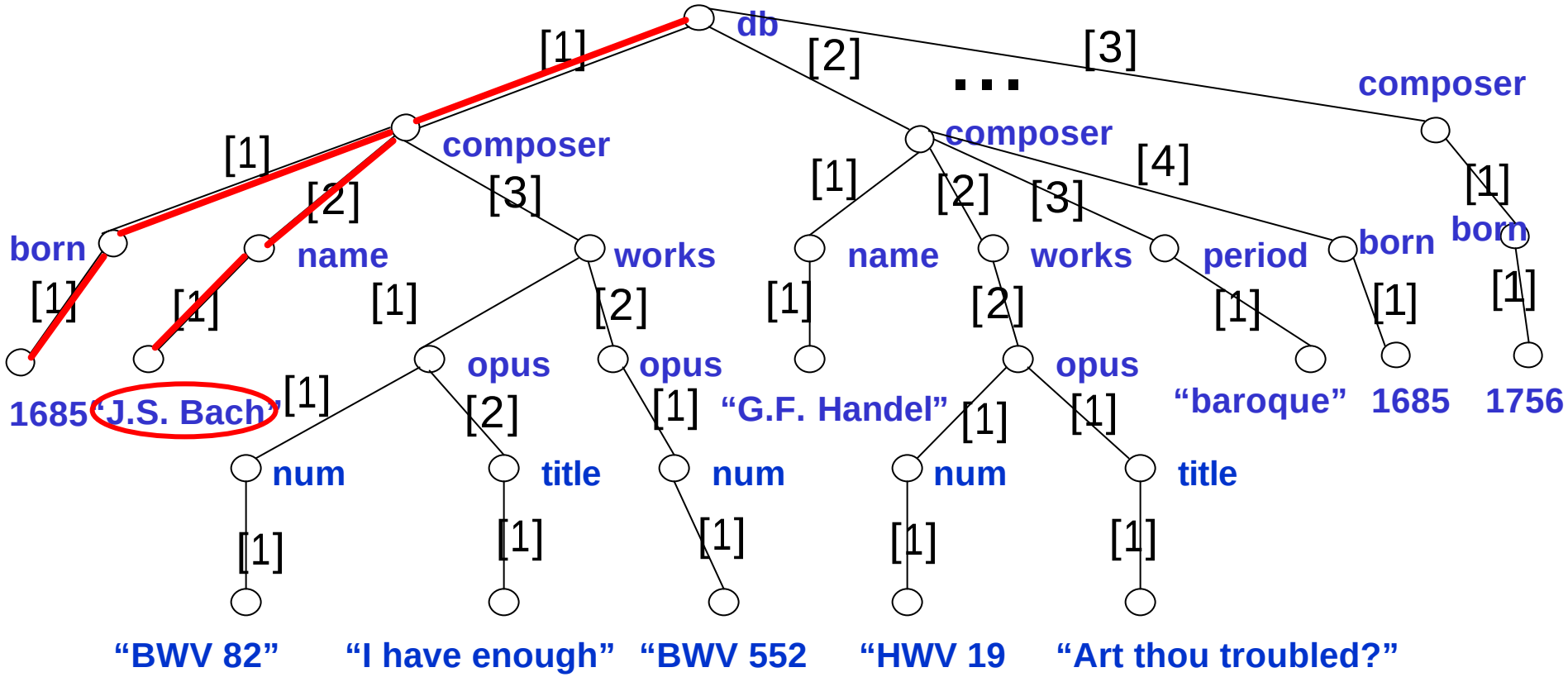
```
where <db.composer>
          <born> $b </born>
          <name> $n </name>
        </db> in "composers.xml"
construct  <year  ID=F($b) born=$b>
              <name> $n </name>
            </year>
```

Find all composers born in the same year

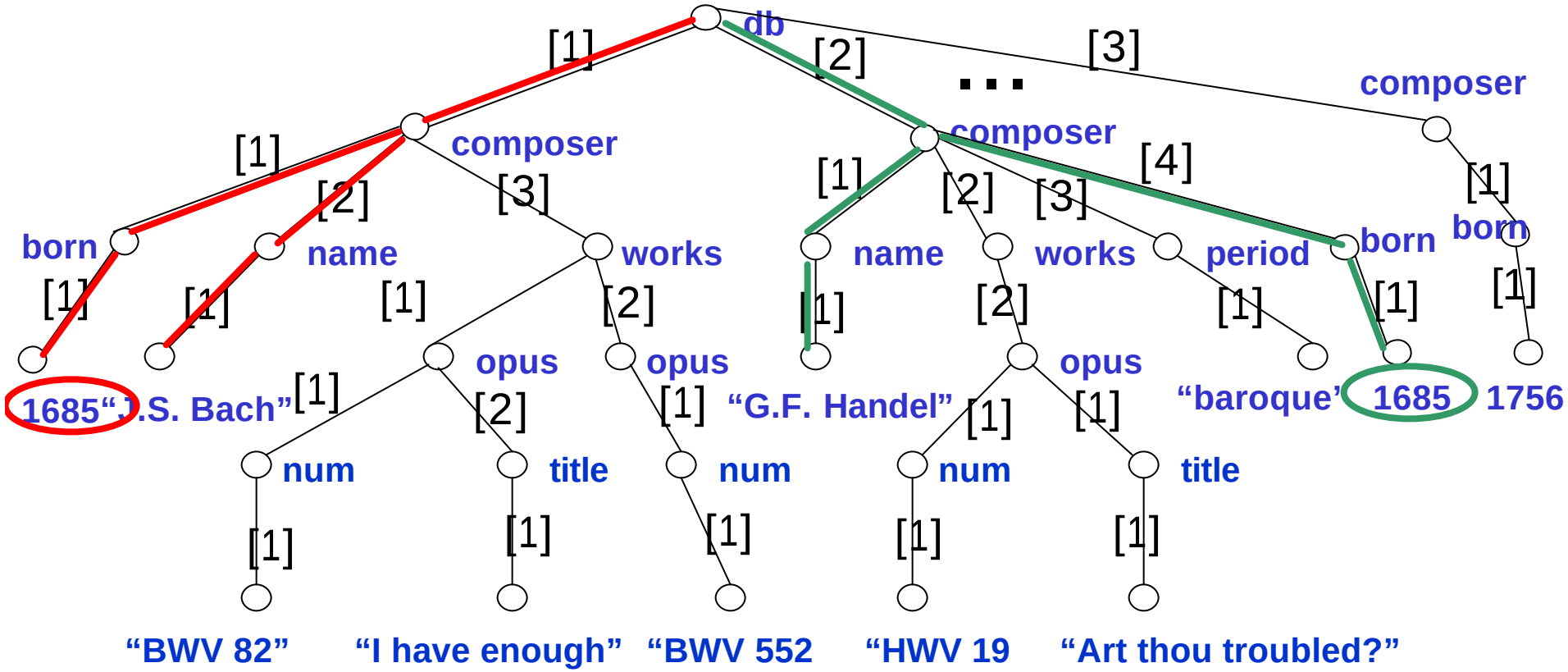
```
<year born=1685>
  <name> G.F. Handel </name>
  <name> J.S. Bach </name>
</year>
```

Why?

The Why-Provenance



The Why-Provenance



An Example

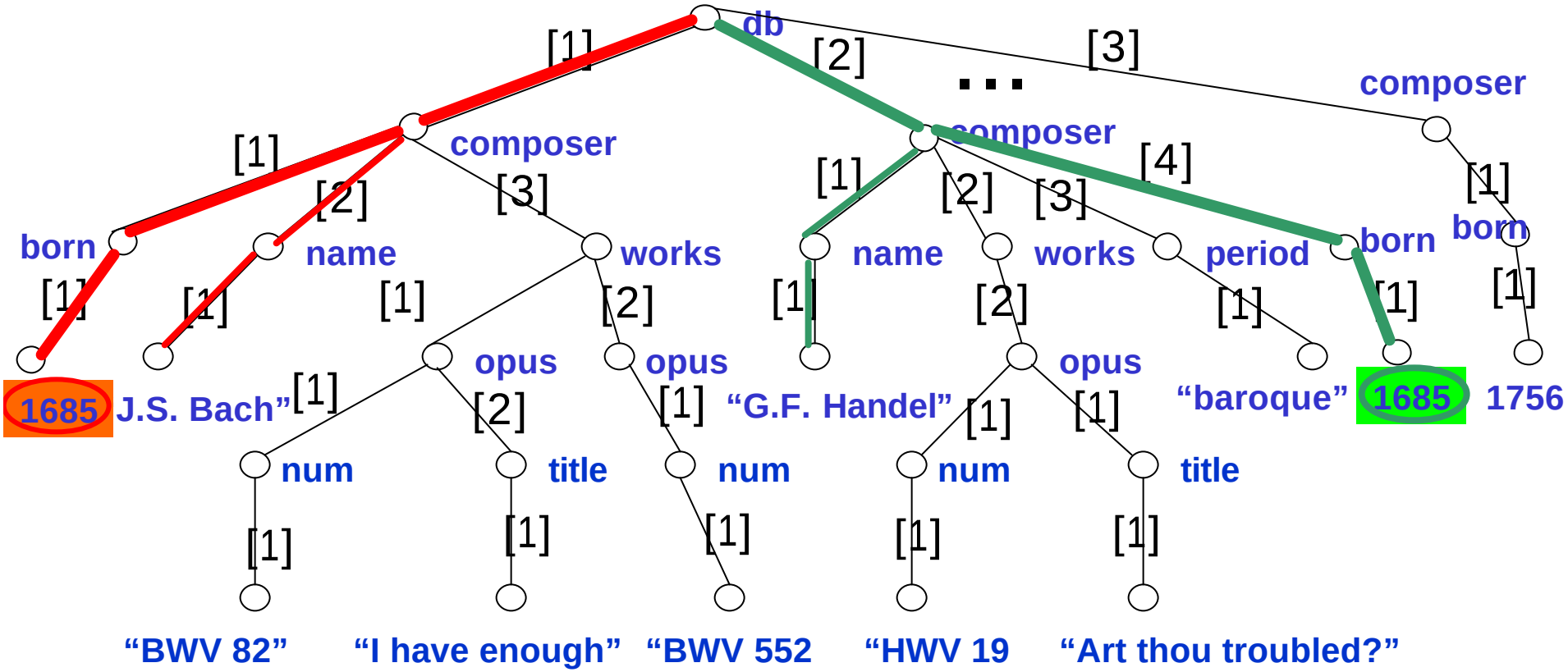
```
where <db.composer>
          <born> $b </born>
          <name> $n </name>
        </db> in "composers.xml"
construct  <year  ID=F($b) born=$b>
              <name> $n </name>
            </year>
```

Find all composers born in the same year

```
<year born=1685>
  <name> G.F. Handel </name>
  <name> J.S. Bach </name>
</year>
```

Where?

The Where-Provenance



A Query Language

- A complex query can be transformed into an equivalent canonical form:

$$Q = Q_1 \cup \ldots \cup Q_n$$

- Each Q_i has the form

where p_1 in D_1,

…

p_n in D_n,

condition

collect e

Each p_i is a "path"-pattern
Each D_i is some database name
Output e is a "path"-pattern as well

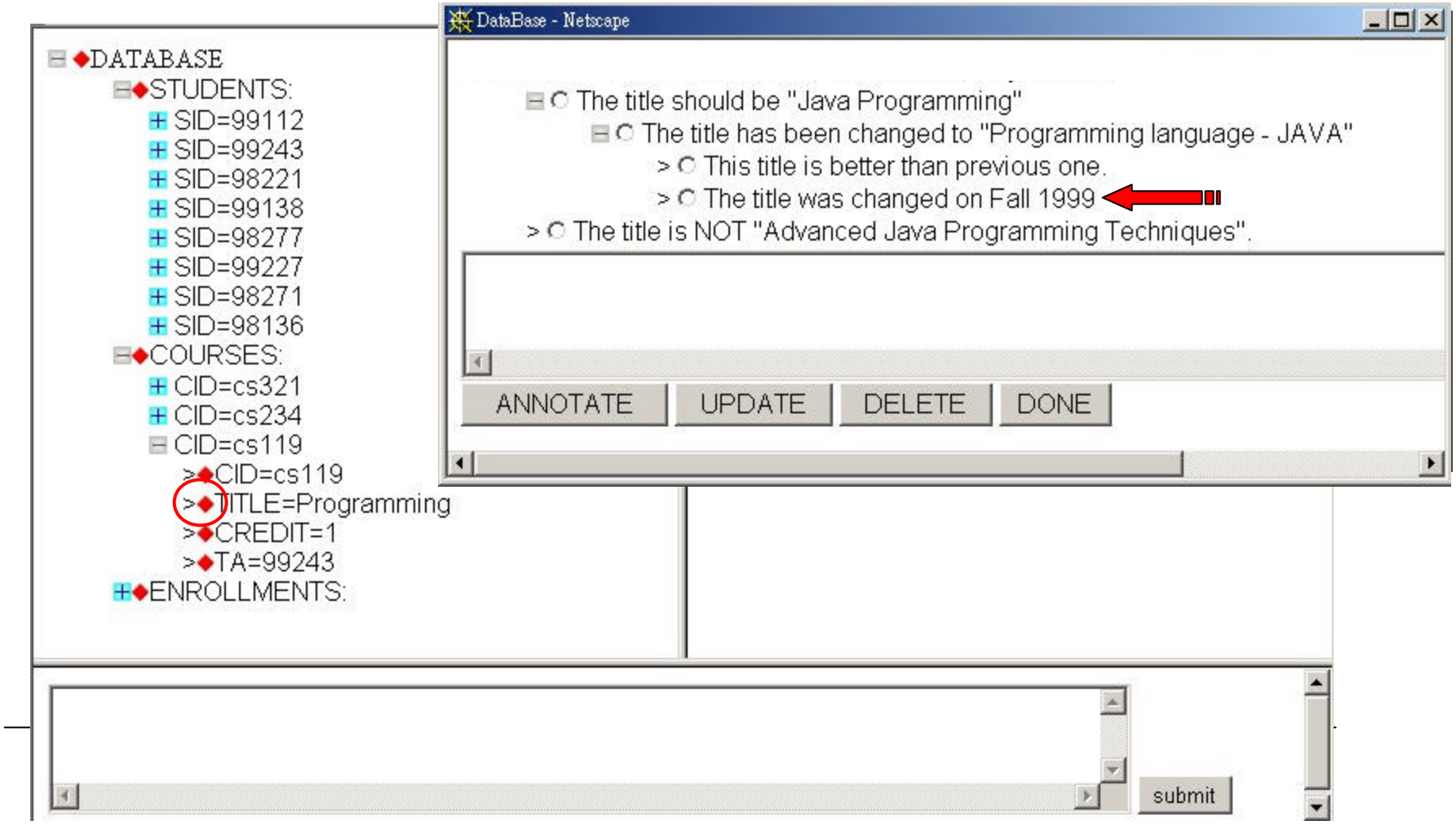
How does this help compute provenance?

- We are interested in the provenance of some component of the database.
- This component is specified by a path *p* (a path expression without variables)
- Map the variables in the output expression e to the path *p*
- Track these variables though the query to identify provenance

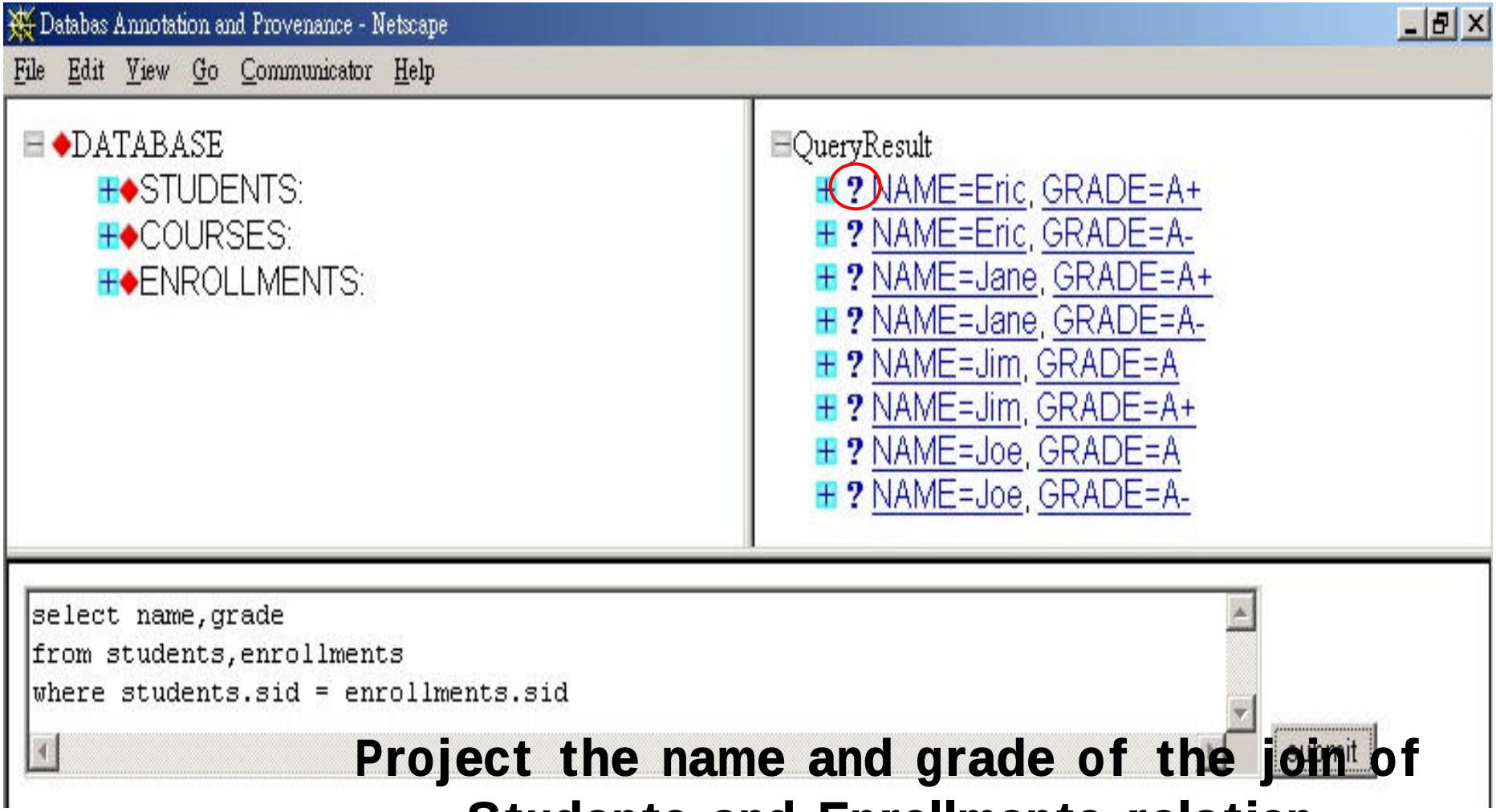
Demo

Computing Data Provenance and Carry Annotations through Queries

http://hobbit.cis.upenn.edu:8080

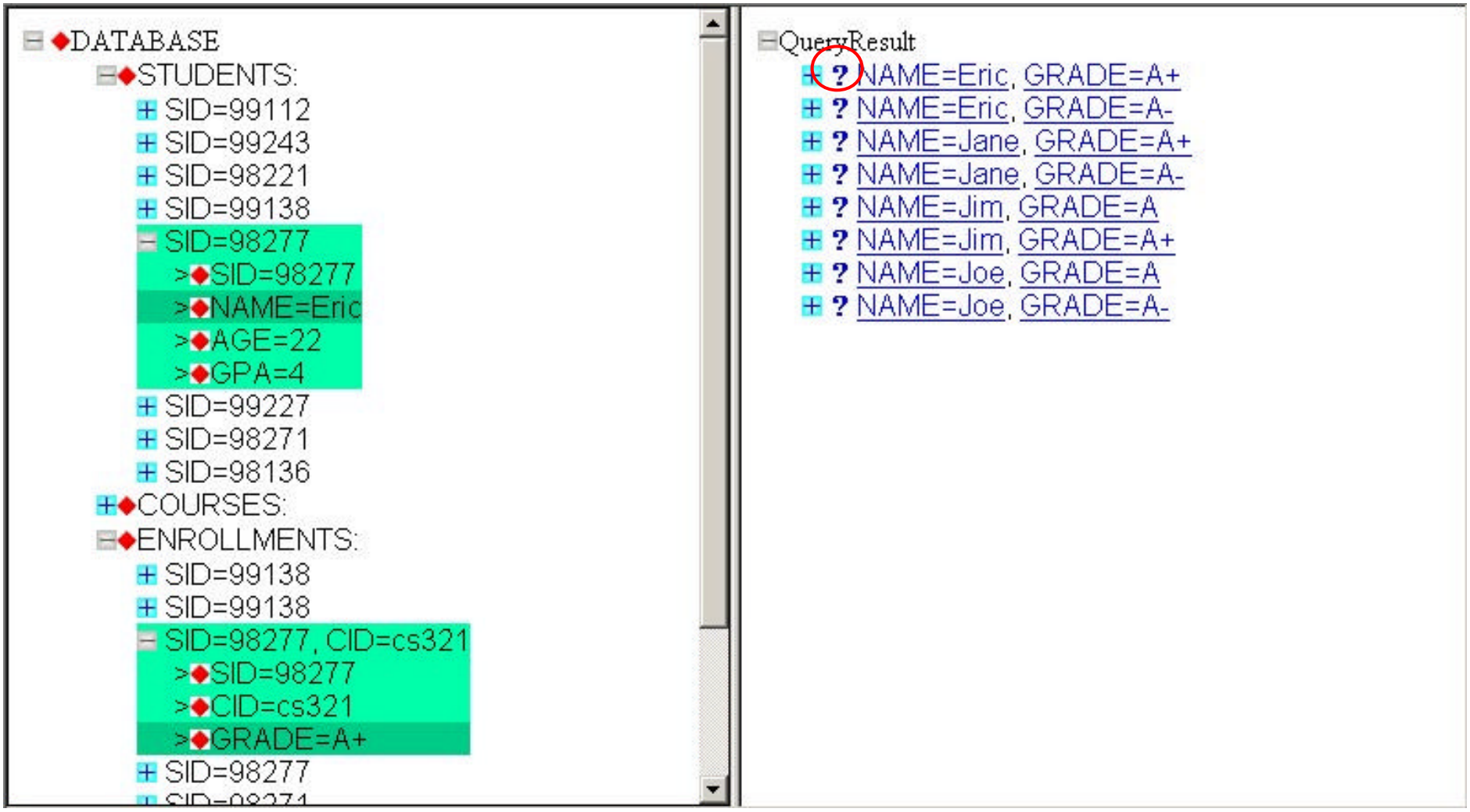


A Join Query



Project the name and grade of the join of Students and Enrollments relation

A Join Query


DATABASE
STUDENTS:
SID=99112
SID=99243
SID=98221
SID=99138
SID=98277
SID=98277
NAME=Eric
AGE=22
GPA=4
SID=99227
SID=98271
SID=98136
COURSES:
ENROLLMENTS:
SID=99138
SID=99138
SID=98277, CID=cs321
SID=98277
CID=cs321
GRADE=A+
SID=98277
QueryResult
? NAME=Eric, GRADE=A+
? NAME=Eric, GRADE=A-
? NAME=Jane, GRADE=A+
? NAME=Jane, GRADE=A-
? NAME=Jim, GRADE=A
? NAME=Jim, GRADE=A+
? NAME=Joe, GRADE=A
? NAME=Joe, GRADE=A-

Results

- Previous work (Cui, Widom, and Wiener) provides a solution for relational databases.
  - deals with “why” provenance

- For why-provenance, we get same answers by very different framework

- Where-provenance -- seems new !

Closing Thoughts

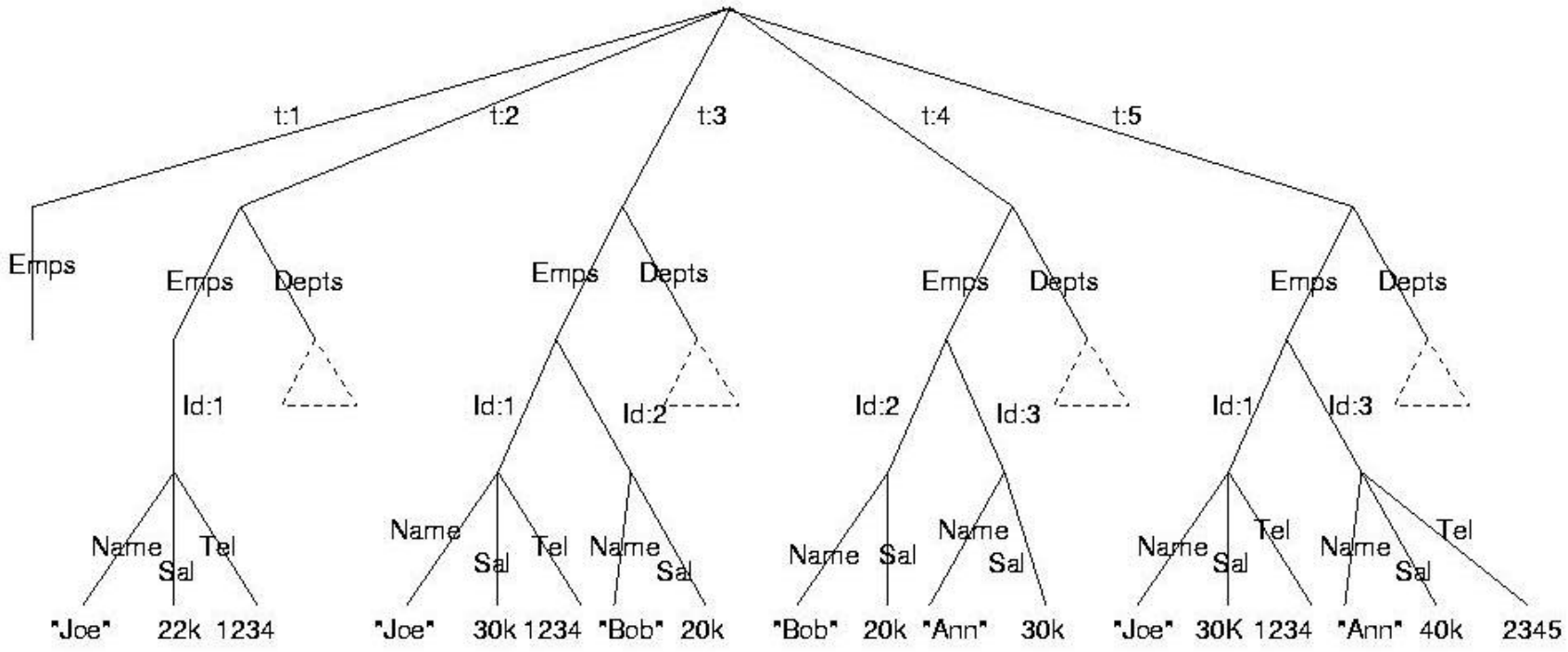
- Data provenance is a subtle issue
- Where provenance allows to carry over meaningful annotations
- Annotations are expensive


http://www.jsbach.org
name: “J.S. Bach”
name: “G.F. Handel”
http://www.intr.net/bleissa/handel/home.html
works
http://www.britannica.com/bcom/eb/article/6/0,5716,118776+10+110129,00.html
works
period
http://www.jsbach.org/bwvs500.html
http://www.jsbach.org/bwv552.html
num: “BWV 82”
num: “BWV552”
num: HWV19
“baroque”
http://www.jsbach.org/bwvs000.html
http://www.jsbach.org/bwv82.html
http://www.intr.net/bleissa/handel/1to42.html#HWV19
http://www.intr.net/bleissa/handel/1to42.html#HWV19/title
title
title
title
http://www.jsbach.org/bwv552.html#title
http://www.jsbach.org/bwv82.html#title
“Ich habe genug”
“St. Anne Fugue”
“Art thou troubled?”

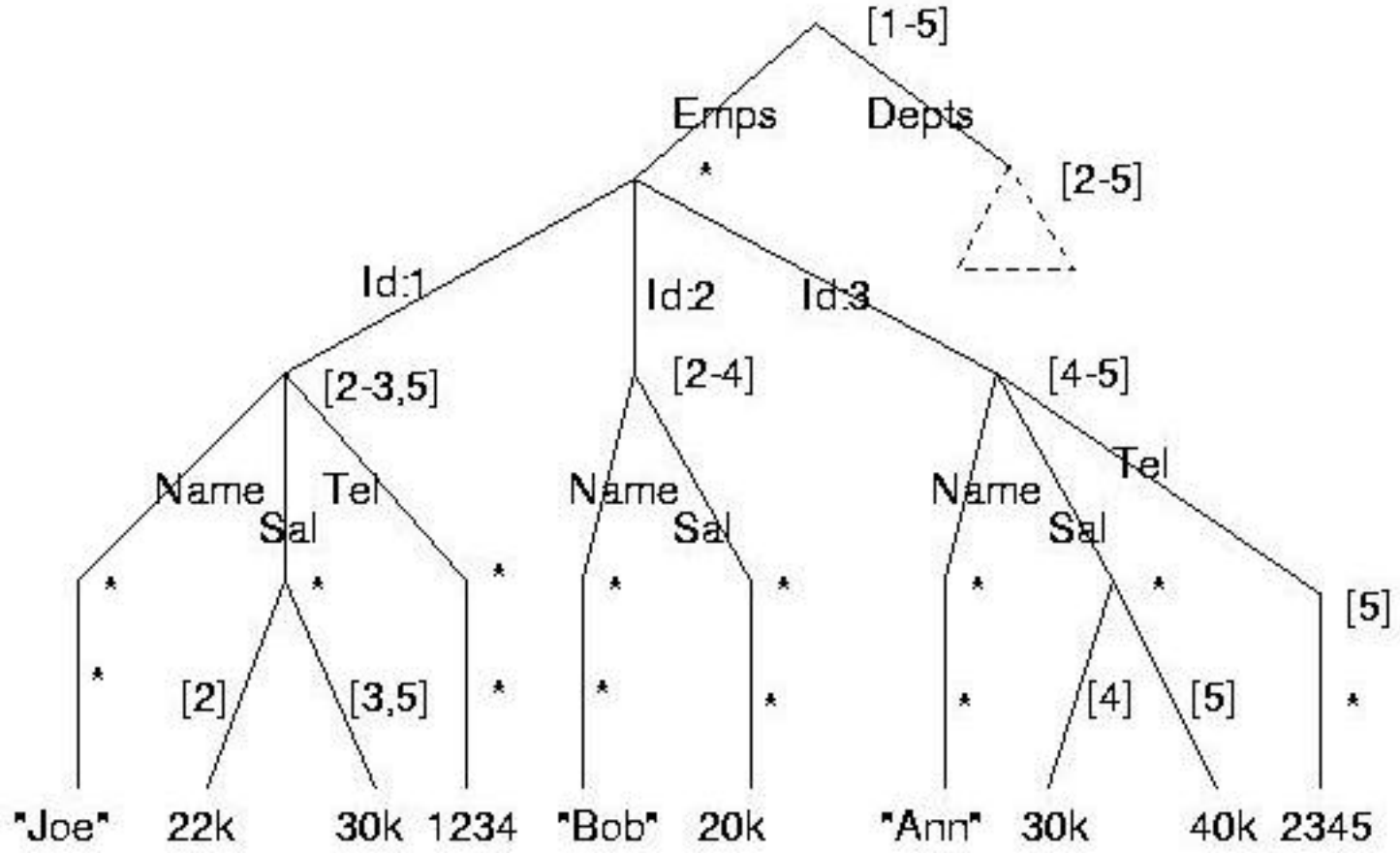
Closing Thoughts

- Data provenance is a subtle issue
- Where provenance allows to carry over meaningful annotations
- Annotations are expensive
  - Imagine recursive annotation trees !
  - What compression techniques work ?
- Meaningful annotations require versioning

Using “persistent” representations



The “compressed” version



END