

# RDF-Constraints and SPARQL

Thomas Hornung and Michael Meier, 15.04.2008

*joint work with Georg Lausen, Norbert Kuchlin, and  
Michael Schmidt*

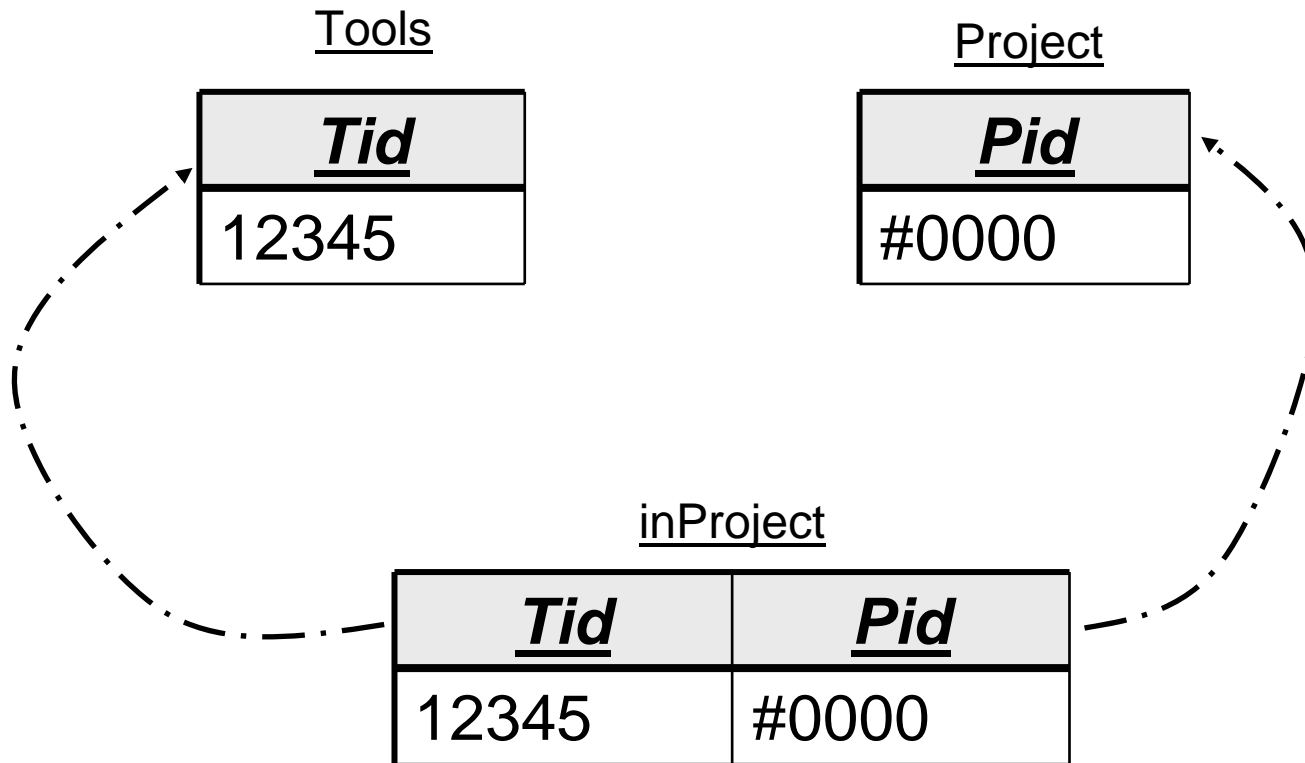
# Table of Contents

- **Part One: Constraints**
  - Encoding Constraints in RDF
  - Checking Constraints
  
- **Part Two: SPARQL Performance Benchmark**
  - RDF Data Generator & 16 Queries
  - Evaluation Results

# Motivation

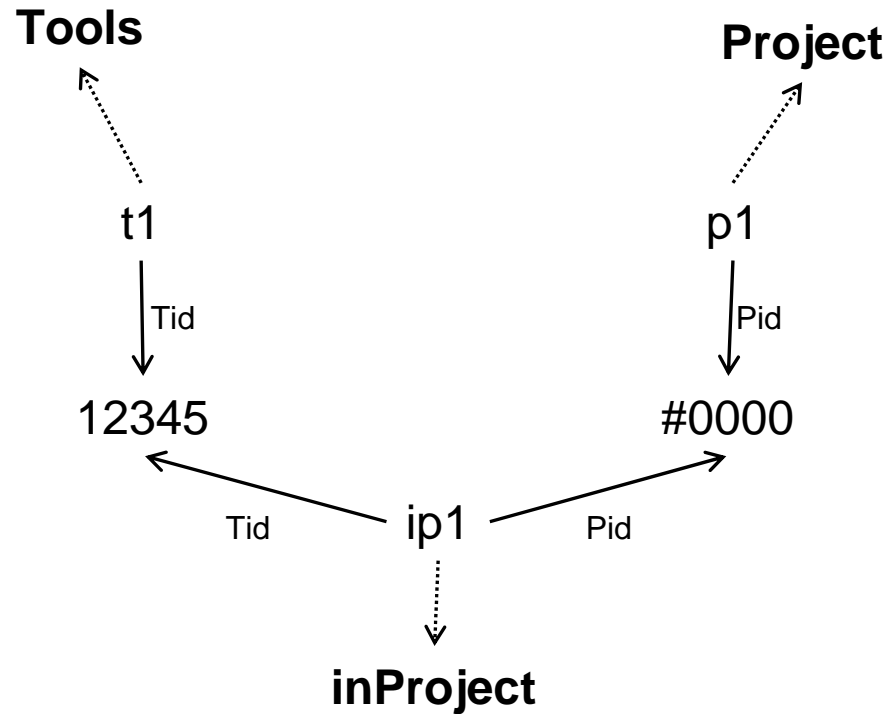
- Export relational databases to RDF
  - Data consistency
  - Updates

# An example



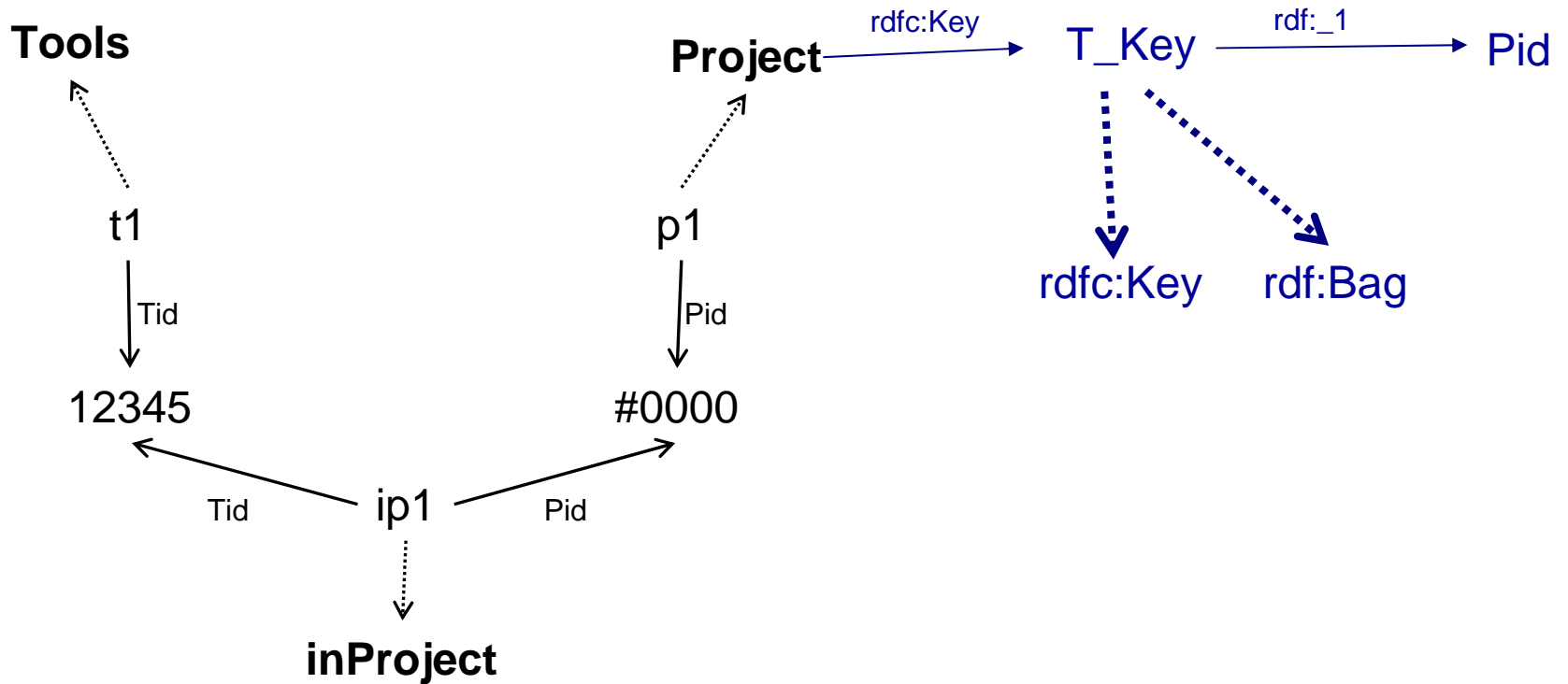
# An example

rdf:type



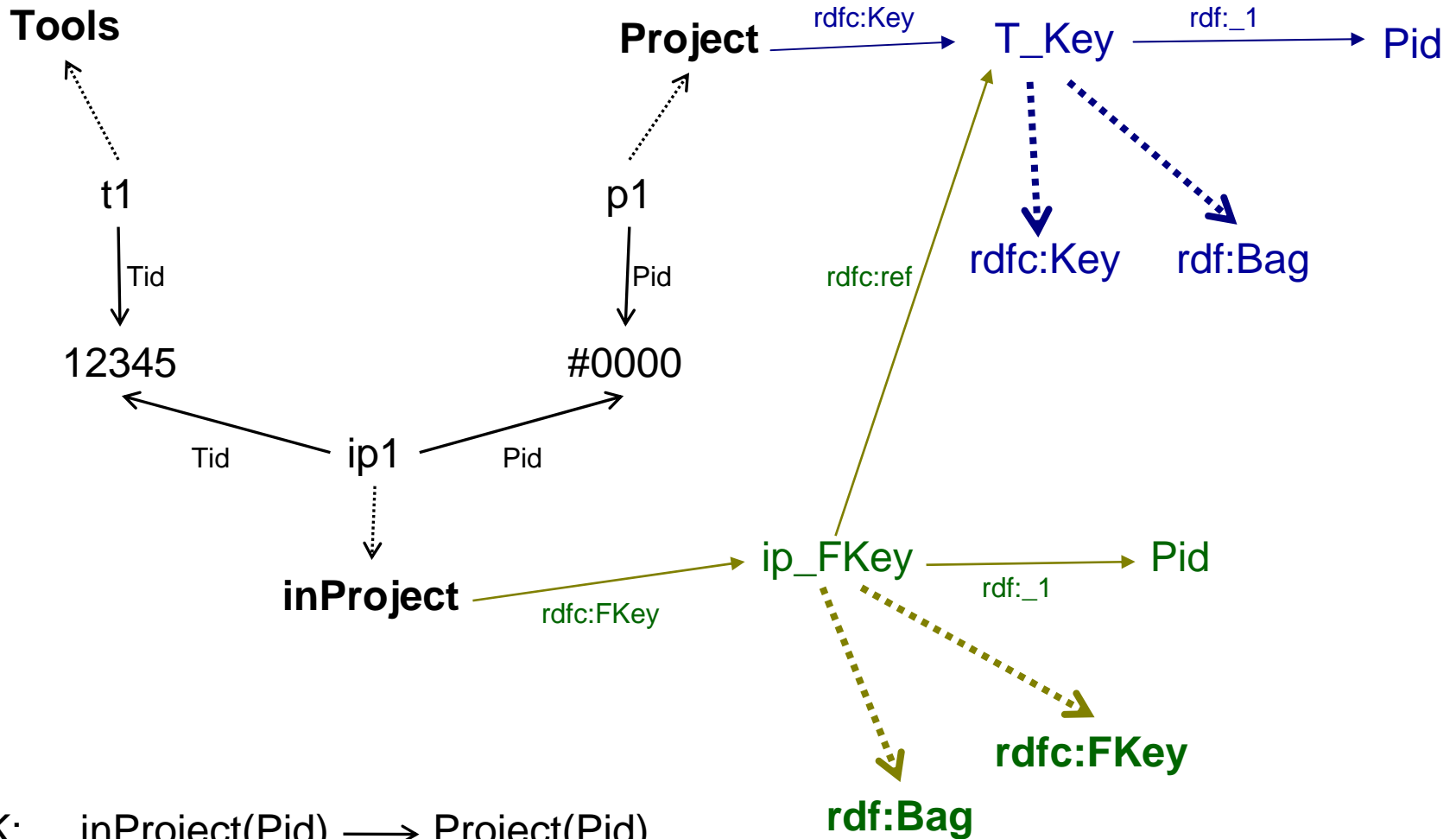
No representation of key and foreign key constraints.

# An example



Encoding in the same graph as the data via a new namespace **rdfc**.

# An example



# Checking a foreign key

*ASK {*

*?x rdf:type inProject; Pid ?Pid.*

*OPTIONAL {*

*?y rdf:type Project; Pid ?Pid.*

*} FILTER (!bound(?y))*

*}*

# Part Two: SPARQL Performance Benchmark

# Motivation

- Up-to-date no benchmark for SPARQL has been proposed
- SP<sup>2</sup>B fills this gap
  - Settled in the DBLP scenario
  - Data generator for creating arbitrarily large datasets + 16 benchmark queries

# The SP<sup>2</sup>Bench Data Generator

- Creates bibliography documents similar to DBLP in a deterministic and incremental way
- Mirrors vital key characteristics found in original DBLP data
  - Structure of entities (Articles, Journals, Books, ...)
  - Relations between authors
  - Quantity of entities (development over time)
  - Citation system

# The SP<sup>2</sup>Bench Queries

- 14 SELECT and 2 ASK Queries
- Different SPARQL solution modifiers
- Closed World Negation
- (Comparably) long predicate chains and bushy patterns
- Amenable to a variety of SPARQL optimizations
- Varying selectivity, output size, etc.

# Benchmark Results

- Considered SPARQL Engines:
  - ARQ (memory/SDB)
  - Sesame (memory/native)
  - Virtuoso
  - Redland
- Findings:
  - Significant differences between engines
  - Room for improvement in current implementation
  - Poor support for several SPARQL specifics

# Current Focus

- Benchmarking of RDF stores on top of relational DBMS:
  - Oracle
  - MySQL
  - MonetDB
- Different storage schemes:
  - Triple tables
  - Property tables
  - Vertical partitioning

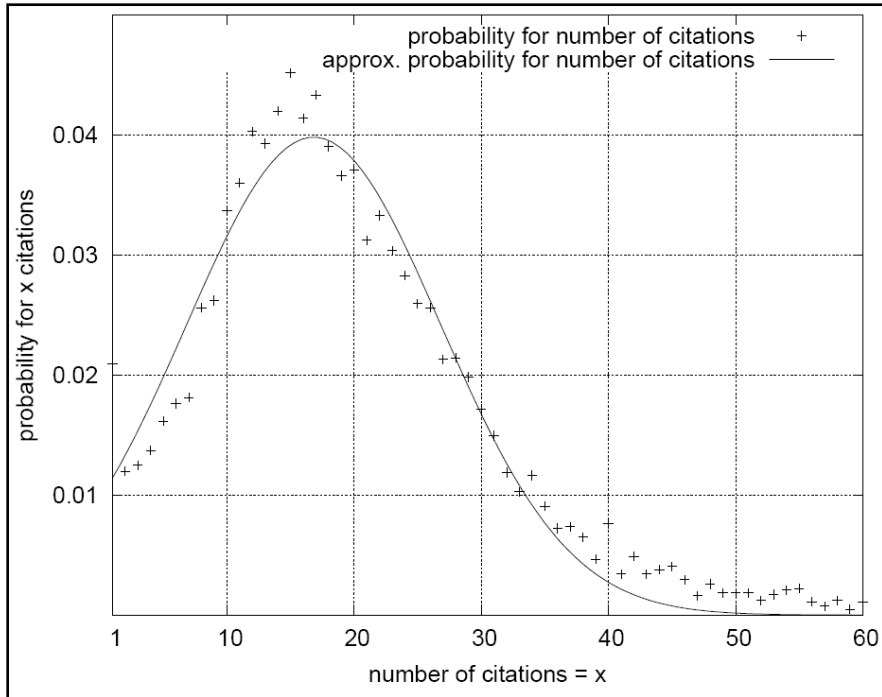
Abadi et. al. (VLDB 2007)

# Initial Results

- Column-oriented DBMS are clearly preferable to row-store DBMS
- Vertical partitioning does not solve general problems
- General drawbacks of relational approach:
  - Selectivity estimation is often suboptimal
  - Translating more complex patterns, such as CWN, results in queries with left joins and filters

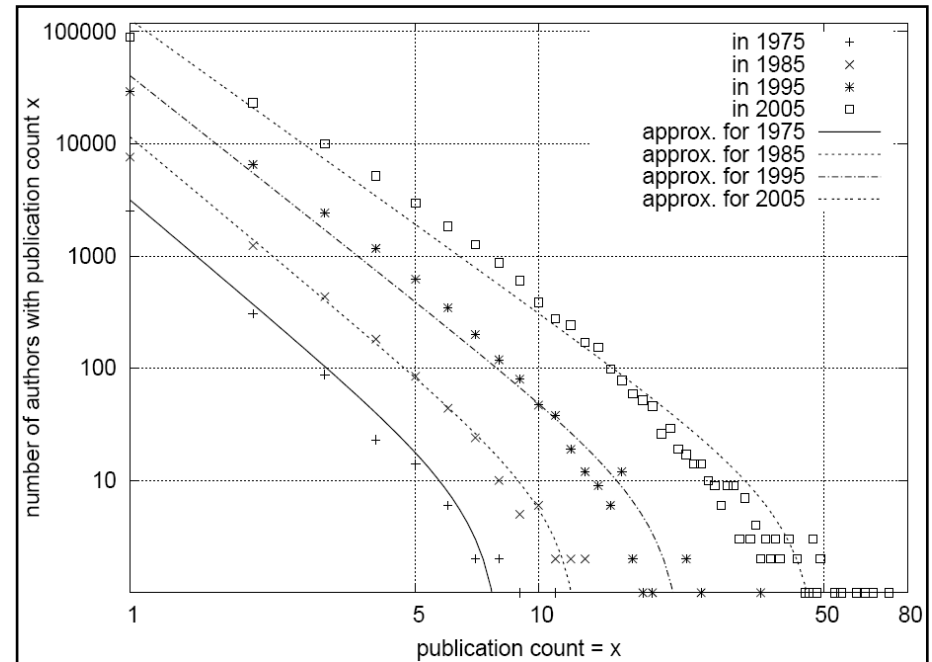
# Part Three: Additional Resources

# The SP<sup>2</sup>Bench Data Generator

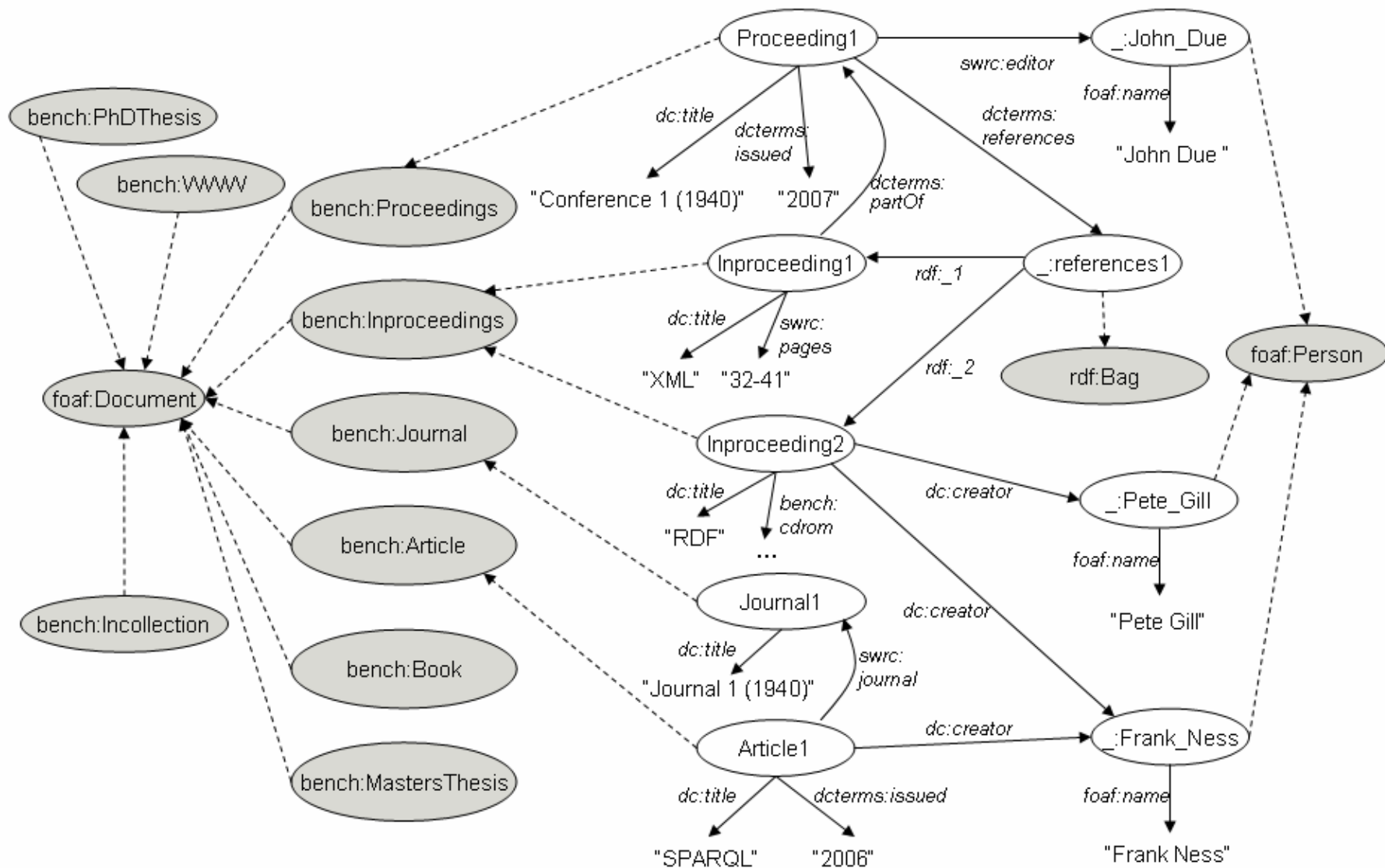


Distribution of Citations

## Distribution of Publications



# The DBLP RDF Schema



# Benchmark Queries

```
SELECT ?yr
WHERE {
  ?proc rdf:type bench:Journal.
  ?proc dc:title "Journal 1 (1940)"^^xsd:string.
  ?proc dcterms:issued ?yr.
}
```

Q1

- Simple
- Constant result size (exactly 1 result)
- Might be answered very fast with index

# Benchmark Queries

```
SELECT DISTINCT ?person ?name Q5
WHERE {
  ?article rdf:type bench:Article.
  ?article dc:creator ?person.
  ?inproc rdf:type bench:Inproceedings.
  ?inproc dc:creator ?person2.
  ?person foaf:name ?name.
  ?person2 foaf:name ?name2.
  FILTER(?name=?name2). }
```

Q5a

```
SELECT DISTINCT ?person ?name
WHERE {
  ?article rdf:type bench:Article.
  ?article dc:creator ?person.
  ?inproc rdf:type bench:Inproceedings.
  ?inproc dc:creator ?person.
  ?person foaf:name ?name. }
```

Q5b

- Equivalent in our scenario
- Tests implicit vs. explicit joins
- We found that Q5a is much more challenging for current engines

# Benchmark Queries

```
SELECT ?yr ?name ?doc
WHERE {
  ?class rdfs:subClassOf foaf:Document.
  ?doc rdf:type ?class.
  ?doc dcterms:issued ?yr.
  ?doc dc:creator ?author.
  ?author foaf:name ?name.
OPTIONAL {
  ?class2 rdfs:subClassOf foaf:Document.
  ?doc2 rdf:type ?class2.
  ?doc2 dcterms:issued ?yr2.
  ?doc2 dc:creator ?author2.
  FILTER (?author=?author2 && ?yr2<?yr). }
FILTER(!bound(?author2)). }
```

Q6

- Closed-World-Negation
- Returns, for each year, the set of all publications authored by persons that have not published in years before

# Benchmark Query (Triples)

(a) `SELECT DISTINCT T2.obj AS person,  
                  T3.obj AS name  
FROM Triples T1  
      JOIN Triples T2 ON T1.subj=T2.subj  
      JOIN Triples T3 ON T2.obj=T3.subj,  
      Triples T4  
      JOIN Triples T5 ON T4.subj=T5.subj  
      JOIN Triples T6 ON T5.obj=T6.subj  
WHERE T1.pred='rdf:type'  
      AND T2.pred='dc:creator' AND T3.pred='foaf:name'  
      AND T4.pred='rdf:type'   AND T5.pred='dc:creator'  
      AND T6.pred='foaf:name'  AND T1.obj='bench:Article'  
      AND T4.obj='bench:Inproceedings' AND T3.obj=T6.obj`

(b) `SELECT DISTINCT T2.obj AS person,  
                  T5.obj AS name  
FROM Triples T1  
      JOIN Triples T2 ON T1.subj=T2.subj  
      JOIN Triples T3 ON T2.obj=T3.obj  
      JOIN Triples T4 ON T3.subj=T4.subj  
      JOIN Triples T5 ON T3.obj=T5.subj  
WHERE T1.pred='rdf:type'   AND T2.pred='dc:creator'  
      AND T3.pred='dc:creator' AND T4.pred='rdf:type'  
      AND T5.pred='foaf:name'  AND T1.obj='bench:Article'  
      AND T4.obj='bench:Inproceedings'`

Q5

# Triples Table

<b>Subject</b>	<b>Predicate</b>	<b>Object</b>
s1	foaf:name	Alice
s1	foaf:mbox	alice@gmx.net
s2	foaf:name	Bob
s2	foaf:mbox	bob@web.de

# Partition Table

<b>Subject</b>	<b>foaf_name</b>	<b>foaf_mbox</b>
s1	Alice	alice@gmx.net
s2	Bob	bob@web.de

# Vertical Partitioning

**Table foaf\_name**

<b>Subject</b>	<b>Object</b>
s1	Alice
s2	Bob

**Table foaf\_mbox**

<b>Subject</b>	<b>Object</b>
s1	alice@gmx.net
s2	bob@web.de

# More Constraints

- Let  $C, C_1, C_2$  be classes and  $Q_i, R_i$  properties
  - *Primary Keys*: **Key**( $C, [Q_1, \dots, Q_n]$ )
  - *Foreign Keys*: **FKey**( $C_1, [Q_1, \dots, Q_n], C_2, [R_1, \dots, R_n]$ )
  - *Cardinality Constraints*: **Min**( $C, n, R$ ), **Max**( $C, n, R$ ) for  $n \in \mathbb{N}$
  - *Singleton Constraints*: **Single**( $C$ )
  - *Subclass Constraint*: **SubC**( $C_1, C_2$ )
  - *Subproperty Constraint*: **SubP**( $Q_1, Q_2$ )
  - *Property Domain/Range*: **PropD**( $Q, C$ ), **PropR**( $Q, C$ )

# Satisfiability

Given a set of constraints is there a non-empty RDF graph that satisfies the constraints?

- **Primary keys + Foreign Keys**
- **Singleton**
- **Max-Cardinality**
- **Subclass + Subproperty**
- **Property Domain + Property Range**



satisfiable

# Satisfiability

- **Primary keys + Foreign Keys**
- **Singleton**
- **Max-Cardinality**
- **Subclass + Subproperty**
- **Property Domain + Property Range**
- **Min-Cardinality**



undecidable

# Satisfiability

- **Unary primary keys**
- **Unary foreign keys**
- **Min-Cardinality + Max-Cardinality**
- **Subclass + Subproperty**
- **Property Domain + Property Range**



EXPTIME