Data Fusion and Peer Data Management

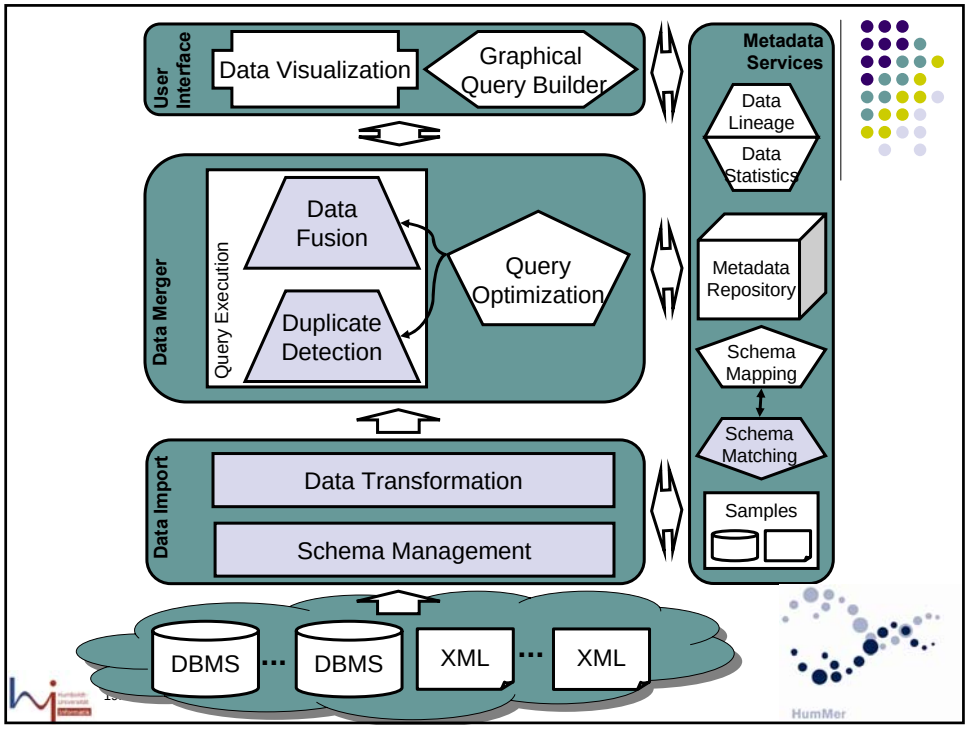
Almaden, 13. January 2006
Felix Naumann
naumann@informatik.hu-berlin.de
Humboldt-Universität zu Berlin

Humboldt-Universität zu Berlin

Campus Adlershof

Overview

- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P



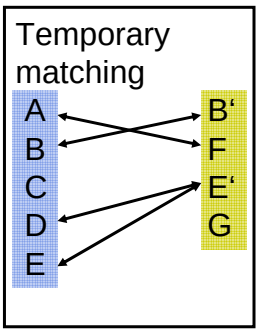
Then and now

Schema Matching - Duplicate-based - Single Table	Alexander Bilke	Schema Matching - Complex schemas - n:m matches
Duplicate Detection - XML - Top-down	Melanie Weis	Duplicate Detection - Bottom-up - Though the middle
Data Fusion - Conflict resolution - FUSE BY	Jens Bleiholder	Data Fusion - SQL interface & GUI - Visualization

Duplicate-driven Schema Matching with DUMAS

A	B	C	D	E
Max	Michel	m	601- 4839204	601- 4839204
...	...	...	...	...

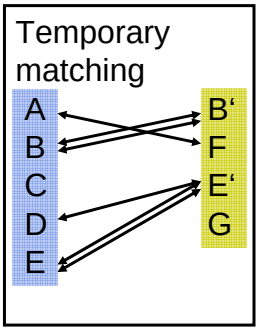
B‘	F	E‘	G
Michel	maxm	601- 4839204	UNIX
...	...	...	...



Duplicate-driven Schema Matching with DUMAS

A	B	C	D	E
Max	Michel	m	601- 4839204	601- 4839204
Sam	Adams	m	541- 8127100	541- 8121164

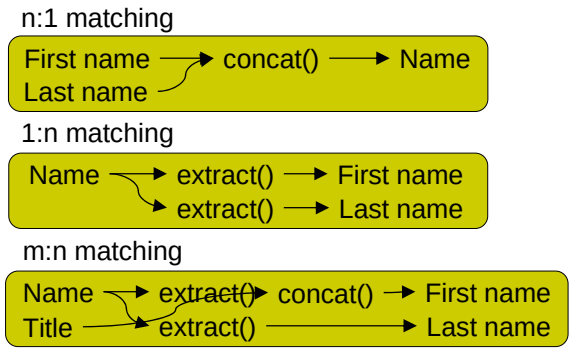
B‘	F	E‘	G
Michel	maxm	601- 4839204	UNIX
Adams	beer	541- 8127164	WinXP



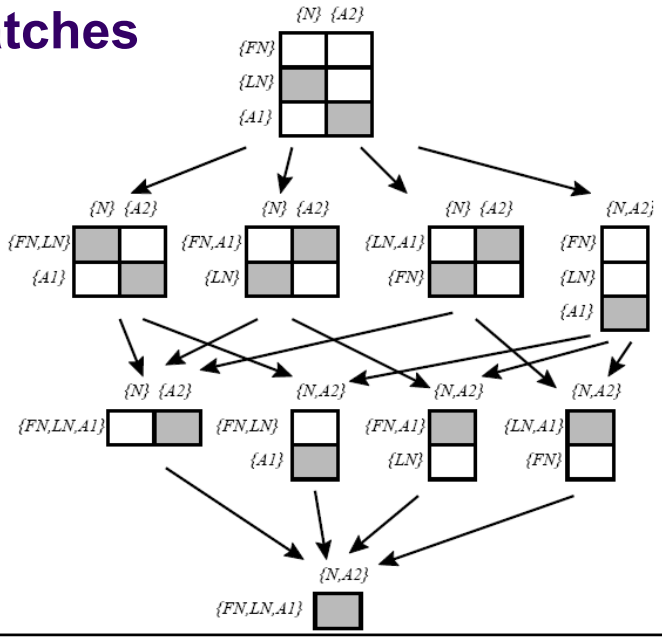
Schema Matching – Complex Schemata

- Start with well-matched tables.
- Iteratively expand these tables by joining on key/foreign keys
  - In source schema
  - In target schema
- Re-perform duplicate-based matching
  - Reuse duplicates from earlier run
  - Stop if previous matches are lost.
  - Stop if no additional matches are found
  - Better: Stop if no additional matches are found in two consecutive iterations

Schema Matching – n:m Matches

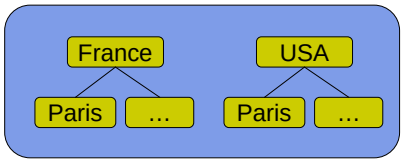


Schema Matching – n:m Matches

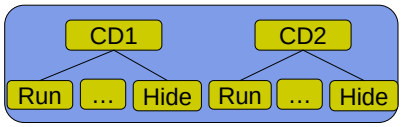


Top-down, bottom-up, and through the middle

Top-down [SIGMOD'05]
- Compare elements only if parents are equal or duplicates
- Improved efficiency



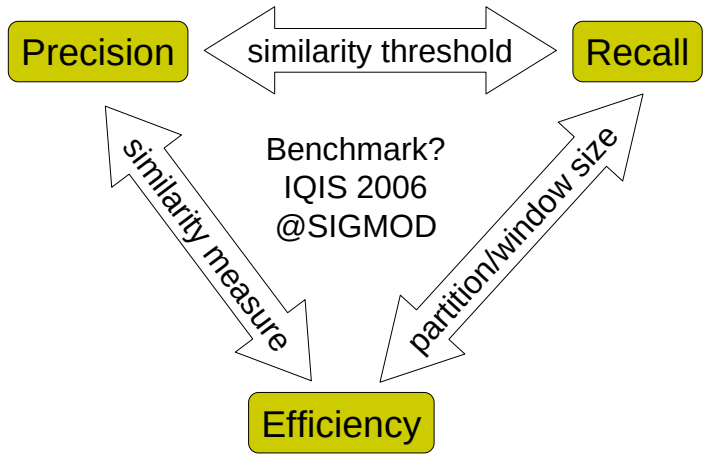
Bottom-up [EDBT'06]
- SXNM
- Similar Children ⇒ Duplicate
- Improved effectiveness



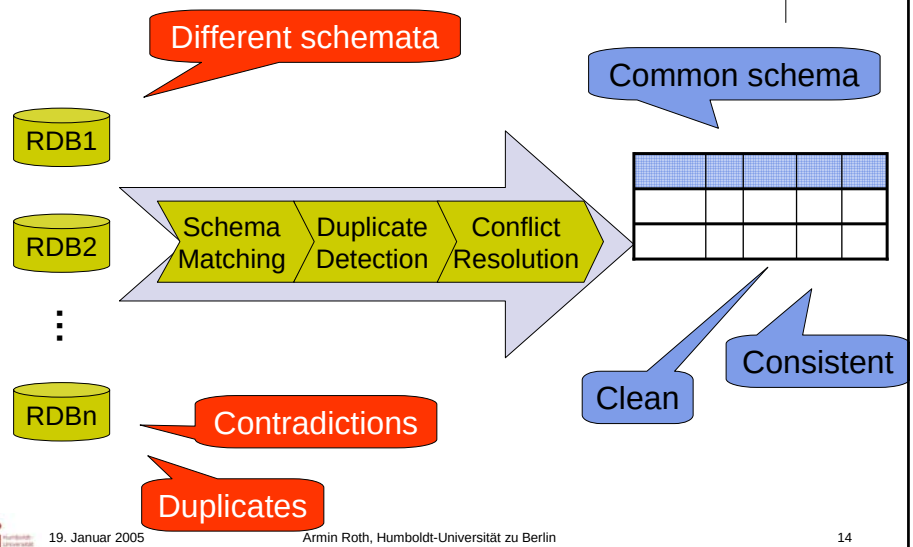
Through the middle [ICDE'06]
- Begin with most promising pairs
- AdamA and ReconA
- Best of both worlds

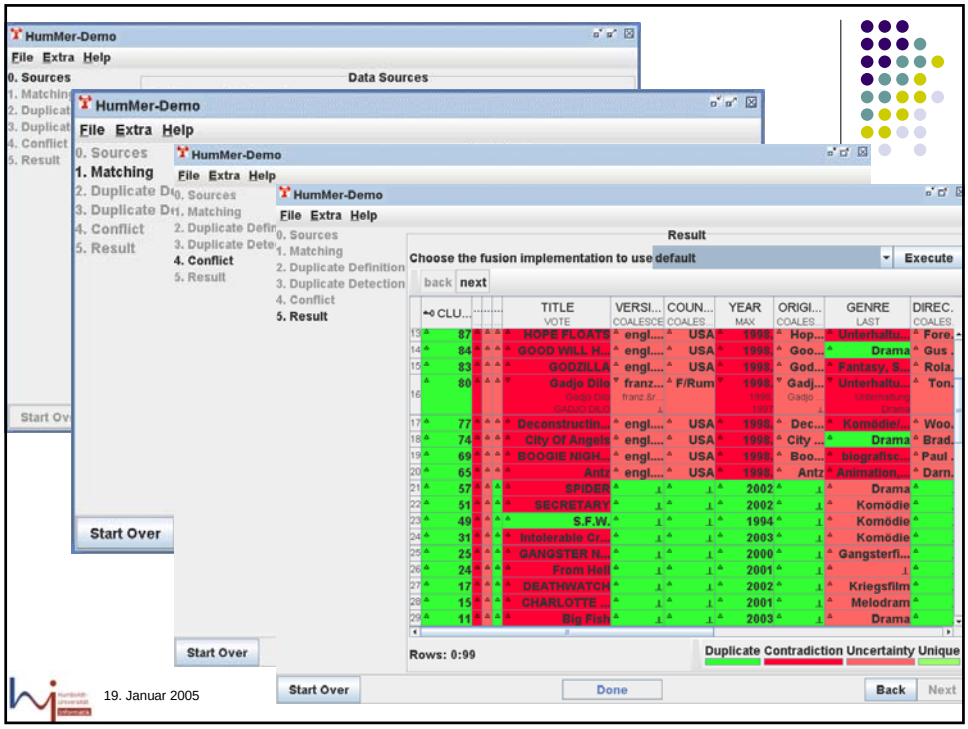
Further enhancements
- Object filter
- Edit distance filter
- Transitivity

Evaluating Duplicate Detection



Putting it all together: HumMer





Fuse'em – HumMer sans SM and DD

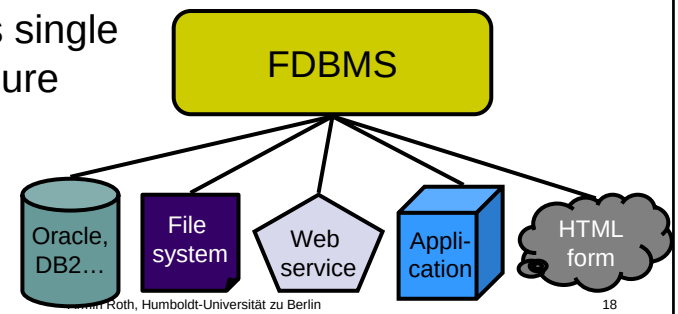
- SQL interface
  - Most of standard SQL
  - Plus FUSE BY
  - Plus library of conflict resolution functions
  - Based on XXL
- Future work
  - Efficiency!
- Demo
  - later?

Overview

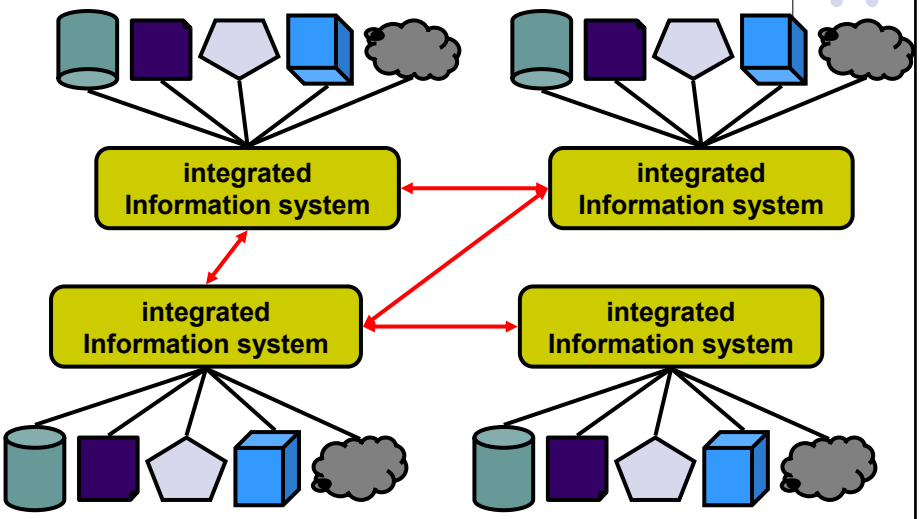
- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P

Federated Databases

- Global schema & direct access to data sources
  - Complex to build and maintain (evolution)
  - Scalability and flexibility “poor”
- Mediator is single point of failure



PDMS generalize integrated information systems

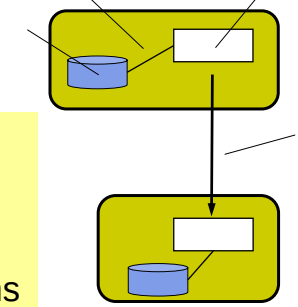


Peers

Local mappings

Peer schema

Local data sources
(local schema)



Peer mappings
LaV, GaV, GLaV

Functionality

- Pose queries
- Plan queries
- Execute query plans

Applications for PDMS

- Peers must be able to
  - Reformulate and pass along queries
  - Receive, transform and pass back query results
- Applications
  - Scientific, life sciences data
  - Disaster data management
  - Health information systems
  - Large scale loosely coupled data integration
  - Semantic Web?

PDMS Projects

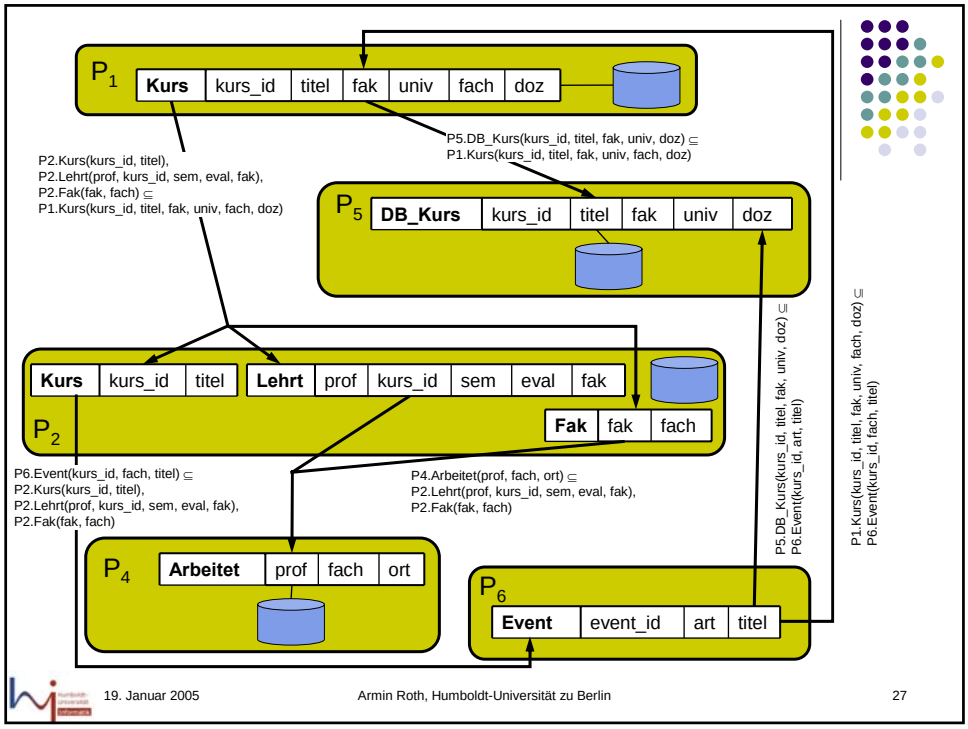
- Mariposa
  - Stonebraker et al.
- Piazza
  - Halevy, Ives, Tatarinov et al.
- Hyperion
  - Miller et al.
- Edutella
  - Nejdl et al.
- PeerDB
  - Ng, Ooi, Tan
- PIER
  - Hellerstein et al.
- System P
  - Us
  - Key differences
    - Fuzzy Pruning
    - Local Planning

PDMS ≠ P2P

- P2P
  1. Files
  2. Simple queries
     - Filename, metadata
  3. Incomplete results
  4. Simple schema
  5. Highly dynamic
  6. Millions of peers
  7. Results shipped directly
- PDMS
  1. Objects
  2. Complex queries
     - Query language (SQL, etc.)
  3. Complete results (expected)
  4. Complex schema
  5. Controlled dynamics
  6. Dozens of peers
  7. Results shipped along mapping-path

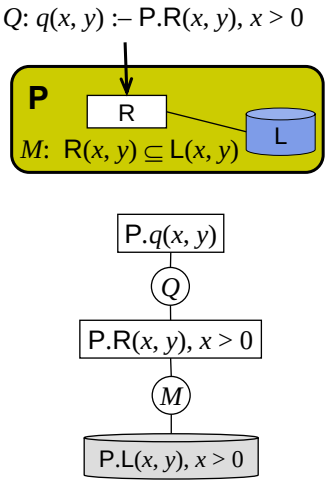
Overview

- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P

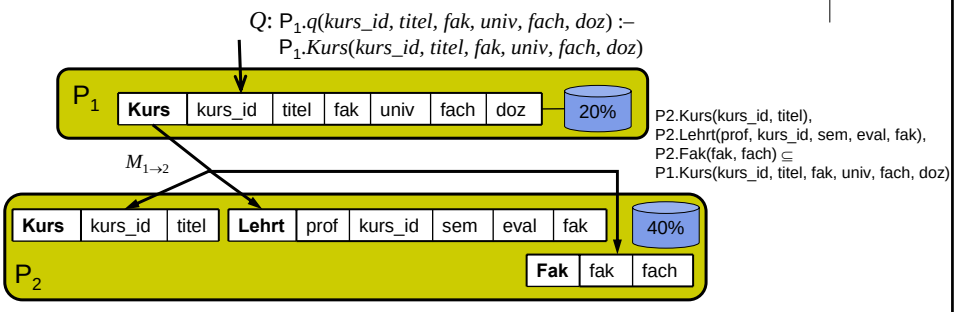


Query Planning: Rule-Goal Tree [HIST03]

- Goal-node ▭: goals of the (reformulated) queries + comparison predicates
- Rule-nodes ◯ : represent peer mappings
- Rule-goal tree created by expanding GaV mappings and LaV mappings

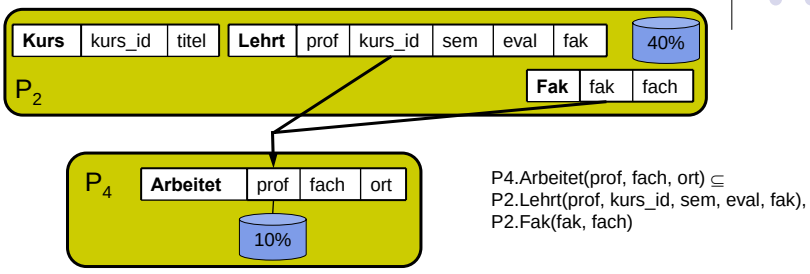


GaV-query reformulation




```
[] Peer001.q(kurs_id, titel, fak, univ, fach, doz)
    () Q
        [] Peer001.Kurs(kurs_id, titel, fak, univ, fach, doz)
            () M1→2
                [] Peer002.Kurs(kurs_id, titel)
                [] Peer002.Lehrt(prof__1, kurs_id, sem__2, eval__3, fak)
                [] Peer002.Fak(fak, fach)
```


LaV-query reformulation



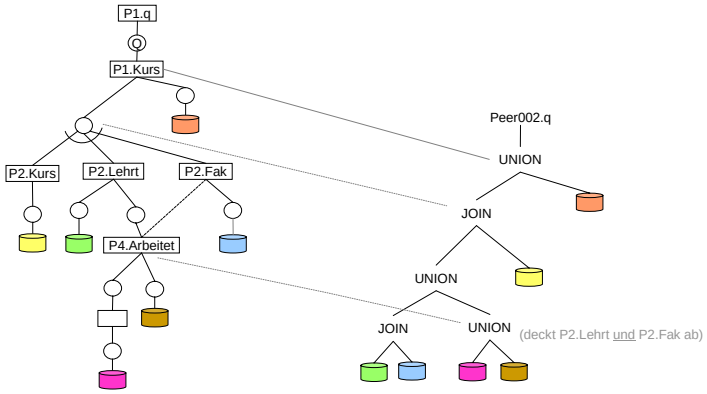

```
[] Peer001.q(kurs_id, titel, fak, univ, fach, doz)
    () Q
        [] Peer001.Kurs(kurs_id, titel, fak, univ, fach, doz)
            () M1→2
                [] Peer002.Kurs(kurs_id, titel)
                [] Peer002.Lehrt(prof__1, kurs_id, sem__2, eval__3, fak)
                    () M2→4
                        [] Peer004.Arbeitet(prof__1, fach, ort__6)
                            [unc] Peer002.Fak(fak, fach)
                [] Peer002.Fak(fak, fach)
```

Query planning and comparison predicates

- Comparison predicates are in
  - User query
  - mappings
- Comparison predicates accumulate along mapping paths, reducing result size.

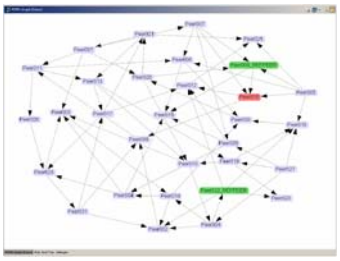
From rule-goal-tree to query plan

- Branching goal-node: UNION
- Branching rule-node: JOIN



Poor efficiency due to redundancies

- Redundant mapping-paths lead to high fan-out of rule-goal trees.
- Example [Schw06]:
  - 31 Peers
  - Avg. connectivity: 5
  - 34378 Union- und 17035 Join-Operationen

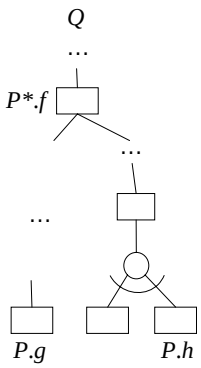


Overview

- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P

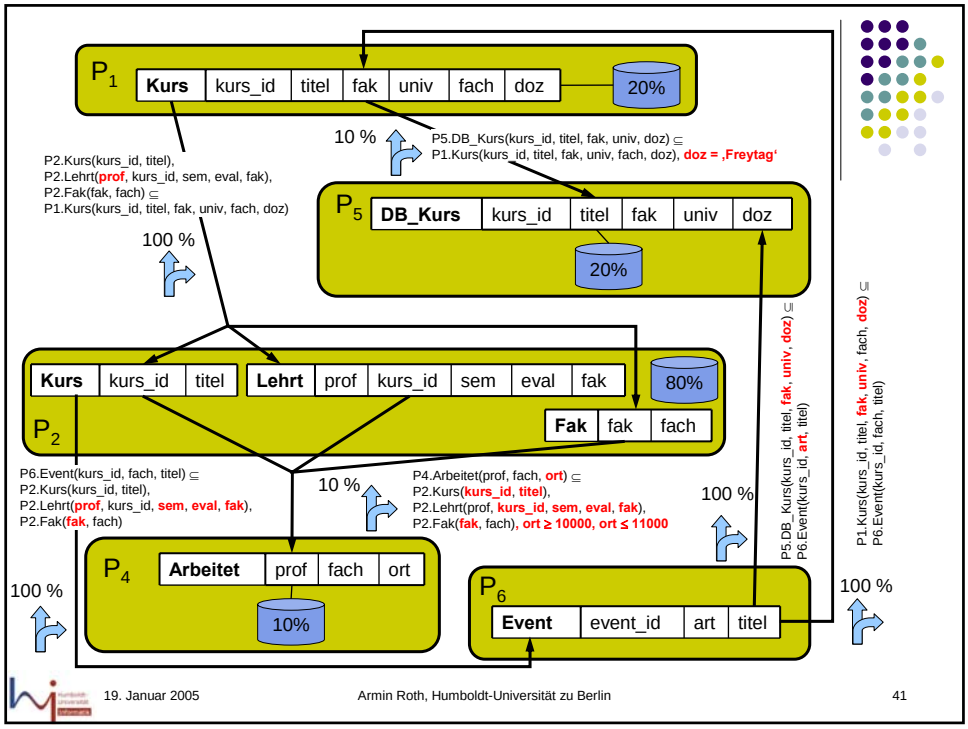
Containment-based Pruning in Piazza [TH04]

- Prune h if
  (i) g contains h
  (g and h at same peer)
  (ii) no joins between g and f
- Advantage:
  - Efficiency gain by one magnitude
  - No results are lost
- Disadvantage:
  - Non-local knowledge necessary
  - Autonomy of peers compromised



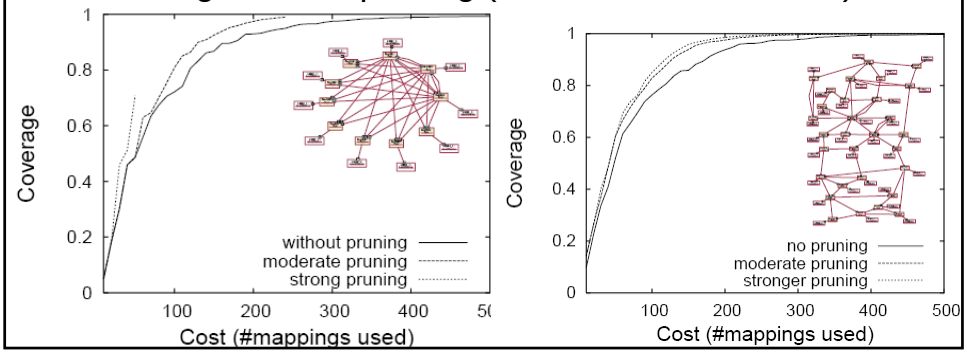
Completeness in PDMS

- Optimization goal: Completeness
- Extensional completenes: Ratio of tuples
- Intensional completeness: Density of non-null values
- Projections and selections in peer-mappings lead to loss of information



Completeness-driven pruning

- Favor mappings with low information loss
- Strategies must be folly localized
  - Threshold-based pruning
  - Budget-based pruning (bounds execution cost)

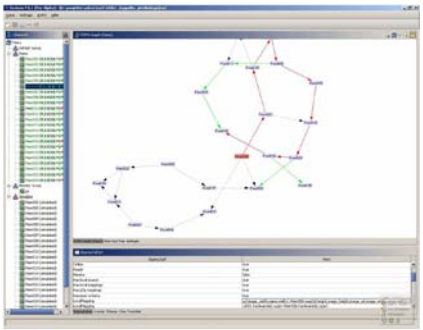


Overview

- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P

System P [RN05]

- An actual PDMS, not a simulation
- Relational data model
  - Point and range queries
- GaV- and LaV-Mappings
- Completeness-driven query planning
- Visualization of query execution
- Current work:
  - Estimating selectivities with self-adaptive histograms
  - Cost model
  - Parallel query execution



Questions?

- Data Fusion with HumMer
  - Architecture
  - Brief update
- Peer Data Management with System P
  - PDMS architecture
  - Query planning in PDMS
  - Pruning Query Plans
  - System P

References

[CGLR04] Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Logical foundations of peer-to-peer data integration. In: Proc. of the Symposium on Principles of Database Systems (PODS), 2004.

[HIST03] Halevy, A.Y., Ives, Z., Suciu, D., Tatarinov, I.: Schema mediation in peer data management systems. In: Proc. of the Int. Conf. on Data Engineering (ICDE), 2003.

[Hübn06] Hübner, T.: Entwicklung einer Testumgebung für ein Peer Data Management System. Humboldt-Universität zu Berlin, Diplomarbeit, 2006.

[RN05] Roth, A., Naumann, F.: Benefit and cost of query answering in PDMS. In: Proc. of the Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P), 2005.

[Schw06] Schweigert, M.: Entwurf eines Peer Data Management Systems mit Steuerungs- und Simulationskomponente. Humboldt-Universität zu Berlin, Diplomarbeit, 2006.

[TH05] Tatarinov, I., Halevy, A.: Effcient query reformulation in peer data management systems. In: Proc. of the ACM Int. Conf. on Management of Data (SIGMOD), 2004.