Data Consolidation in Three Steps

Milano, May 9, 2008
Felix Naumann
The HPI – Hasso Plattner Institut

- Founded in 1998 as a Public Private Partnership
- Adjoined with the University of Potsdam
  - Capital of Brandenburg, bordering Berlin
- 400 students – Bachelor, Master, and PhD
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
- Summary

Data Fusion in Three Steps:
Resolving Inconsistencies at Schema-, Tuple-, and Value-level

Felix Naumann\textsuperscript{1} \quad Alexander Bilke\textsuperscript{2} \quad Jens Bleiholder\textsuperscript{1} \quad Melanie Weis\textsuperscript{1}

\textsuperscript{1} Humboldt-Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
{naumann|bleiho|weis}@informatik.hu-berlin.de

\textsuperscript{2} Technische Universität Berlin
Strasse des 17. Juni 135, 10623 Berlin, Germany
bilke@cs.tu-berlin.de

Data Fusion

JENS BLEIHOLDER and FELIX NAUMANN
Hasso-Plattner-Institut, Potsdam, Germany

Felix Naumann | Information Systems | Hasso Plattner Institut
Information Integration

Web Service A

Web Service B

Integration Identification Fusion Optimization Visualization

<title>Federated Database Systems</title>
<author>Amit Sheth</author> <author>James Larson</author>
</publication>

<title>Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases</title>
<author>Scheth & Larson</author> <year>1990</year>
</publication>
Information Integration

Web Service A

Web Service B

Integration
Identification
Fusion
Optimization
Visualization

Fusion
Optimization
Visualization

Schema Integration

Schema Mapping

Information Systems
Hasso Plattner Institut
Information Integration

Schema Integration

Web Service A

Web Service B

Integration Identification Fusion Optimization Visualization

Fusion Optimization Visualization

Felix Naumann | Information Systems | Hasso Plattner Institut
Information Integration

Federated Database Systems

Amit Sheth
James Larson

Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases

Scheth & Larson

1990

Integration Identification Fusion Optimization Visualization

Felix Naumann | Information Systems | Hasso Plattner Institut
Information Integration

Federated Database Systems
Amit Sheth
James Larson

Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases
Seth & Larson

Fusion
Optimization
Visualization

Integration
Identification

Felix Naumann | Information Systems | Hasso Plattner Institut
Information Integration

Web Service A

1sec.

Web Service B

5sec.

Integration Identification Fusion Optimization Visualization

Federated Database Systems

Amit Sheth

James Larson

Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases

Scheth & Larson

1990

Fusion Optimization Visualization

Felix Naumann | Information Systems | Hasso Plattner Institut
Information Integration

Web Service A

Web Service B

1 sec.

5 sec.

Federated Database Systems

Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases

Amit Sheth

James Larson

Scheth & Larson

1990

Integration Identification Fusion Optimization Visualization

Felix Naumann | Information Systems | Hasso Plattner Institut
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
- Summary
Schema Mapping Example

- **ARTICLE**
  - artPK
  - title
  - pages
- **AUTHOR**
  - artFK
  - name

- **PUBLICATION**
  - pubID
  - title
  - date
  - author

```sql
SELECT artPK AS pubID, title AS title, null AS date, null AS author
FROM ARTICLE
UNION
SELECT null AS pubID, null AS title, null AS date, name AS author
FROM AUTHOR
```
Schematic heterogeneity – solutions

Further interpretations?

```
SELECT artPK AS pubID, title AS title, null AS date, name AS author
FROM ARTICLE, AUTHOR
WHERE ARTICLE.artPK = AUTHOR.artFK
```
Schema Matching – Motivation

Schemata are
- large
- complex
- foreign
- confusing
- different language
- cryptic

> 100 tables, many attributes

Deep Nesting
Foreign keys
XML Schema

Unknown synonyms

Unknown homonyms

|attribute name| ≤ 8
|table name| ≤ 8
Schema Matching Approaches

Individual Approaches
- Schema-based
  - Linguistic
  - Constraint-based

Instance-based
- Linguistic

Hybrid
- Duplicate-based
- Constraint-based

Combined Approaches
- Manual
- Automatic
Instance-based Schema Matching:
- Correspondences based on similar data values or their properties

Conventional solution: Vertical
- Comparison of columns
- = Attribute classification

Our solution: Horizontal
- Comparison of rows
- = Duplicate detection (despite missing attribute correspondences)
### DUMAS Matcher

#### Temporary Matching

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Michel</td>
<td>m</td>
<td>601- 4839204</td>
<td>601- 4839204</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
|   | ... | ... | ... | ... | ...

<table>
<thead>
<tr>
<th></th>
<th>B'</th>
<th>F</th>
<th>E'</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Michel</td>
<td>maxim</td>
<td>601- 4839204</td>
<td>UNIX</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
|   | ... | ... | ... | ... | ...
## DUMAS Matcher

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>Michel</td>
<td>m</td>
<td>601-4839204</td>
<td>601-4839204</td>
</tr>
<tr>
<td>Sam</td>
<td>Adams</td>
<td>m</td>
<td>541-8127100</td>
<td>541-8121164</td>
</tr>
</tbody>
</table>

### Temporary matching

<table>
<thead>
<tr>
<th>B'</th>
<th>F</th>
<th>E'</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michel</td>
<td>maxm</td>
<td>601-4839204</td>
<td>UNIX</td>
</tr>
<tr>
<td>Adams</td>
<td>beer</td>
<td>541-8127164</td>
<td>WinXP</td>
</tr>
</tbody>
</table>
Schema Matching – High-level Matching

```
actor
  firstname
  lastname

actress
  firstname
  lastname

Schauspieler
  Vorname
  Nachname
  Geschlecht

actor
  firstname
  lastname

actress
  firstname
  lastname
```

Felix Naumann | Information Systems | Hasso Plattner Institut
Schema Matching – Extensions

n:1 und 1:n matches
- Many combinations
- Many functions
- Parsing
- iMap

Matching in complex schemata
- Find mapping, not only correspondences

Global Matching
- Match Table and Schema, not just Attributes

n:1 Matching
- First name $\rightarrow$ concat() $\rightarrow$ Name
- Surname $\rightarrow$ Name

1:n Matching
- Name $\rightarrow$ extract() $\rightarrow$ First name
- Surname $\rightarrow$ First name

m:n matching
- Name $\rightarrow$ extract() $\rightarrow$ concat() $\rightarrow$ First name
- Title $\rightarrow$ extract() $\rightarrow$ Last name

Global Matching
- Match Table and Schema, not just Attributes

Felix Naumann | Information Systems | Hasso Plattner Institut
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
- Summary
Duplicate detection is the discovery of multiple representations of the same real-world object.

- **Problem 1:** Representations are not identical.
  - *Fuzzy duplicates*
- **Solution:** Similarity measures
  - Value- and record-comparisons
  - Domain-dependent or domain-independent

- **Problem 2:** Data sets are large.
  - Quadratic complexity: Comparison of every pair of records.
- **Solution:** Algorithms
  - E.g., avoid comparisons by partitioning.
Duplicate Detection

Duplicate Detection

R₁ × R₂

Similarity measure

Algorithm

sim > θ

Duplicate

sim < θ

Non-duplicate

Felix Naumann | Information Systems | Hasso Plattner Institut
Motivation

- Possible effects
  - Example: Portfolio Management Offers
  - Credit maximum not detected
  - Too low inventory levels
  - No quantity discount for multiple orders
  - Total revenue of preferred customers unknown
  - Multiple mailings of same catalog to same household

- General problems
  - Additional, unnecessary IT expenses
  - Low customer satisfaction
  - Potentials and dangers not detected
  - Poor quality financial data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW</td>
<td>20.000</td>
</tr>
<tr>
<td>BaMoWe</td>
<td>5.000.000</td>
</tr>
<tr>
<td>Bayerische Motorenwerke</td>
<td>300.000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Felix Naumann | Information Systems | Hasso Plattner Institut
Ironically, “Duplicate Detection” has many Duplicates

- Household matching
- Mixed and split citation problem
- Match
- Deduplication
- Entity resolution
- Identity uncertainty
- Hardening soft databases
- Duplicate detection
- Record linkage
- Object identification
- Object consolidation
- Entity clustering
- Approximate match
- Merge/purge
- Householding
- Reference reconciliation
- Reference matching
Duplicate Detection – Research

Duplicate Detection

Identity
- Relational
- XML
- DWH
- Domain-independent
- Domain-dependent

Similarity measure
- Edit-based
- Token-based
- Rules
- Data types
- Relationship-aware
- Partitioning

Algorithm
- Filters
- Relation-ships
- Clustering / Learning
- Incremental / Search

Evaluation
- Precision / Recall
- Efficiency
Token-based Similarity Measures

- Tokens
  - Words / Terms
  - n-grams
- Jaccard
  - $|\{\text{common tokens}\}| / |\{\text{all tokens}\}|$
- TFIDF [Cohen et al. 2003]
  - Term frequency: $tf$
  - Inverse document frequency: $idf$
  - $\text{TFIDF: } \log (tf+1) \times \log (idf)$
  - Common words have low weight
  - Similarity measure: Cosine similarity of term vectors weighted by TFID
- And many more
  [Koudas Srivastava 2005]
Edit-based Similarity Measures

  - Common letters within ½ string length
  - Transposed letters
- **Edit-distance / Levenshtein-distance** [Levenshtein 1965]
  - Minimum number of edits from one word to the other
  - Domain-specific costing
  - Dynamic Programming
- **Soundex**
  - 4-letter code for each word
  - $\text{SOUNDEX('Farwick ') } = \text{F620}$
- Frass, Fricke, Fahruschi, Feuerhake

...
Domain-dependent Similarity Measures

- Data Types
  - Special similarity for dates
  - Special similarity for numerical attributes
  - ...

- Rules
  - [Hernandez Stolfo 1998], [Lee et al. 2000]
  - Given two records, r1 and r2.
    - IF last name of r1 = last name of r2, AND first names differ slightly, AND address of r1 = address of r2
    - THEN r1 is equivalent to r2.
Relationship-aware Similarity Measures

- Idea: Not only values of the records, but values of related records are relevant for similarity.
  - Persons: spouse, children, employer
  - Movies: actors
  - CDs: songs
  - Customers: orders, addresses
  - Dimensions in a DWH

  [Ananthakrishna et al. 2002]

<table>
<thead>
<tr>
<th>ID</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New York</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Los Angeles</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>New York</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Los Angeles</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>New York</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Los Angeles</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First Ave</td>
</tr>
<tr>
<td>2</td>
<td>High St.</td>
</tr>
<tr>
<td>3</td>
<td>Broadway</td>
</tr>
<tr>
<td>4</td>
<td>Embarcado</td>
</tr>
<tr>
<td>5</td>
<td>Broadway</td>
</tr>
<tr>
<td>6</td>
<td>Second St</td>
</tr>
<tr>
<td>7</td>
<td>P St.</td>
</tr>
<tr>
<td>8</td>
<td>Pennsylvania</td>
</tr>
<tr>
<td>9</td>
<td>Sunset Bv</td>
</tr>
<tr>
<td>10</td>
<td>Santa Mtn</td>
</tr>
<tr>
<td>11</td>
<td>Ocean Ave</td>
</tr>
</tbody>
</table>
Relationship-aware Similarity Measures – Evaluation

![Graph showing precision vs recall for with and without actors](image)

- **without actors**
- **with actors**

Felix Naumann | Information Systems | Hasso Plattner Institut
Partitioning / Blocking

- Partition the records (horizontally) and compare pairs of records only within a partition.
  - Partitioning by first two zip-digits
    - Ca. 100 partitions in Germany
    - Ca. 100 customers per partition
    - => 495,000 comparisons
  - Partition by first letter of surname
  - ...

- Idea: Partition multiple times by different criteria.
  - Then apply transitive closure on discovered duplicates

Source: wikipedia.de
Sorted Neighborhood
[Hernandez Stolfo 1998]

- **Idea**
  - Sort tuples so that similar tuples are close to each other.
  - Only compare tuples within a small neighborhood (window).

1. **Generate key**
   - E.g.: SSN+“first 3 letters of name” + ...

2. **Sort by key**
   - Similar tuples end up close to each other.

3. **Slide window over sorted tuples**
   - Compare all pairs of tuples within window.

- **Problems**
  - Choice of key
  - Choice of window size

- **Complexity:** At least 3 passes over data
  - Sorting!
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
  - Data Conflicts
    - Relational Operators
    - Conflict Resolution
    - Tools
- Summary
Data conflicts

Two duplicates have different values for semantically same attribute.
Data Fusion

amazon.com

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Author</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>0766607194</td>
<td>ID</td>
<td>H. Melville</td>
<td>$3.98</td>
</tr>
<tr>
<td>0766607194</td>
<td>max_length</td>
<td>Herman Melville</td>
<td>MIN</td>
</tr>
<tr>
<td>0766607194</td>
<td>CONCAT</td>
<td>Moby Dick</td>
<td>$5.99</td>
</tr>
</tbody>
</table>

bn.com
Data conflicts – origins

- No integrity or consistency checks
- Redundant schemata
- Typos, transmission errors, incorrect calculations
- Variants
  - Kantstr. / Kantstrasse / Kant Str. / Kant Strasse
  - Kolmogorov / Kolmogoroff / Kolmogorow
- Typical confusion (OCR)
  - U<->V, 0<->o, 1<->l, etc.
- Obsolete values
  - Different update frequencies, forgotten update
Data conflicts – origins

- Locally consistent but globally inconsistent
- Duplicates
- Different data types
- Local spelling variations and conventions

Across information systems
Examples for errors

- **Addresses**
  - Str. → Straße, Ch. → Chaussee, etc.
  - R.-Breitscheid-Str. 72 a → Rudolf-Breitscheid-Str. 72A
  - 128 spellings for Frankfurt am Main
    - Frankfurt a.M., Frankfurt/M, Frankfurt, Frankfurt a. Main, ...

- **Names**
  - Dr. Ing. h.c. F. Porsche AG
  - Hewlett-Packard Development Company, L.P.

- **Numerical data**
  - 10.000 € = 10T EURO = 10k EUR = 10.000,00€ = 10,000.- €

- **Phone numbers**

- **Birth dates**
**Property mistakenly valued at $189 million**

By rcraw  
Created 12/03/2007 - 4:46pm

Property mistakenly valued at $189 million results in tax adjustments in county

An $18,900 Waconia property that was mistakenly valued at $189 million is “throwing a wrench” into property tax statements and the Carver County budget. County officials issued a press release Monday detailing the problem that came to light last week.

An error was identified in the estimated market valuations used to calculate Pay 2008 Proposed Property Taxes, according to the release. The County Assessor’s Office placed an incorrect estimated market value on a parcel located in the city of Waconia, apparently resulting in extra zeroes being added to the value.

The mistake results in an imbalance in the amount of property taxes the county was expecting to collect. The mistake added about $900,000 in expected revenue, according to County Administrator David Hemze.

The county is planning to consider recommendations to cut the 2008 budget by $900,000 so that proposed property taxes will match tax notices sent to residents in November.

“It kind of threw a wrench into everything,” said Hemze. “It’s unfortunate. It’s a mistake and we’re concentrating on responding to the mistake and trying to ensure that it doesn’t happen again.” If the county does not cut the budget by $900,000, the county portion of property taxes would go up for all properties in the county. The effect would be greatest in Waconia, but Hemze said the average-valued home outside of Waconia would also experience a $29 increase on top of the number indicated on the November tax notices.
FRANKREICH

Telefonkundin erhält Rechnung über 63 Millionen Euro


"Uns bleibt nur, uns bei der Kundin zu entschuldigen und dafür zu sorgen, dass so etwas nie wieder vorkommt", so der lapidare Kommentar des Vizechefs von Télé2.
Data conflict – elimination

- **Error correction**
  - Reference tables
    - Cities, countries, products ...
  - Similarity measures
    - For typos
    - For language-specific variants (Meier, Mayer, ...)
- **Standardization and Transformation**
- **Domain-knowledge (meta data)**
  - Konventions (country/region-specific spelling)
  - Ontologies
  - Thesauri, dictionaries for homonyms, synonyms, ...

- And data fusion...
Completeness and Conciseness

Intensional completeness

Extensional completeness

Conflicts!

Duplicate Detection

Common attributes

Source 2

Source 1

Data Fusion

Common objects

Schema Matching

Felix Naumann | Information Systems | Hasso Plattner Institut
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
  - Data Conflicts
  - Relational Operators
  - Conflict Resolution
  - Tools
- Summary
“Proper” Data Fusion

Source 1(A,B,C) → [a, b, c] → [a, b, c, -] → [a, b, c, d] → Complement. tuples
Source 2(A,B,D) → [a, b, d] → [a, b, -, d] → [a, b, -, -] → [a, b, -, -]

Identical tuples

Subsumed tuples

Conflicting tuples
Relational object integration

**Union**
- Elimination of exact duplicates

**Minimum Union**, [GL94]
- Elimination of subsumed tuples

But
- No duplicate integration
- Conflict resolution

Later: Join, Merge, Group, ...
Minimum Union – Outer Union

- Outer Union pads relations with NULL-values, to unify schemata.
- Then normal UNION.
- Usually not implemented in SQL
Minimum Union – Subsumption

- A tuple $t_1$ subsumes a tuple $t_2$, if
  - Has same schema,
  - $t_2$ has more NULL-values than $t_1$,
  - $t_1 = t_2$ for all non-NULL-values of $t_2$.

- Notation:
  - $R\downarrow$ returns those tuples of $R$, that are not subsumed by any other tuple in $R$.

### How many tuples does $R\downarrow$ have?

#### Table $$R\downarrow$$

<table>
<thead>
<tr>
<th>p_id</th>
<th>fname</th>
<th>lname</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peter</td>
<td>Müller</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>Peter</td>
<td>Müller</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>Wiebke</td>
<td>Meyer</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Semantics of NULL? [GUW02]

- "unknown"
  - There is a value, but I do not know it.
  - E.g.: Unknown birthday
- "inapplicable"
  - There is no meaningful value
  - E.g.: partner for singles
- "withheld"
  - There is a value, but we are not authorized to see it.
  - E.g.: Private phone
Minimum Union – NULL-values

"Value not supplied"

"Value does not exist"

"Value undefined"

„Distinguished" NULL

"Total ignorance" NULL

C.J. Date:
- "Into the Unknown"
- "Much Ado About Nothing"
- "NOT Is Not Not!"
- "Oh No Not Nulls Again"
- ...

From now on: "Unknown"
Merge and Prioritized Merge

**Merge (‡), [GPZ01]**

- Mixes Join and Union to a new operator
  - Fuses complementary tuples (only from different sources)
- COALESCE removes NULL values
- Priorization possible (▷)
- Can be expressed with standard SQL

```sql
( SELECT K.p_id, K.fname, Coalesce(K.lname, C.lname), Coalesce(K.age, C.age) 
FROM K 
LEFT OUTER JOIN 
C ON K.p_id = C.p_id 
) 
UNION 
( SELECT C.p_id, K.fname, Coalesce(C.lname, K.lname), Coalesce(C.age, K.age) 
FROM K 
RIGHT OUTER JOIN 
C ON K.p_id = C.p_id 
) 
```
### Merge – example

#### Kunde K

<table>
<thead>
<tr>
<th>p_id</th>
<th>fname</th>
<th>lname</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peter</td>
<td>Müller</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Franz</td>
<td>Schmidt</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Wiebke</td>
<td>Meyer</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Klaus</td>
<td>Lehmann</td>
<td>28</td>
</tr>
</tbody>
</table>

#### Customer C

<table>
<thead>
<tr>
<th>p_id</th>
<th>lname</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>⊥</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Schmidt</td>
<td>⊥</td>
</tr>
<tr>
<td>3</td>
<td>Meier</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>Weger</td>
<td>47</td>
</tr>
</tbody>
</table>

#### C \( \Box \) K

<table>
<thead>
<tr>
<th>p_id</th>
<th>fname</th>
<th>lname</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peter</td>
<td>Müller</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Franz</td>
<td>Schmidt</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Wiebke</td>
<td>Meier</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>Wiebke</td>
<td>Meyer</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>Klaus</td>
<td>Lehmann</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>⊥</td>
<td>Weger</td>
<td>47</td>
</tr>
</tbody>
</table>
What else is there?

**Match Join** [YaÖz99]
- Complex operator
- *HighConfidence, RandomEvidence, and PossibleAtAll*

**ConQuer** [FuFM05]
- „Consistent Query Answering“
- Rewriting of SQL queries

**Burdick et. al.** [BDJR05]
- Uncertainty in Data Warehouses
- „Possible Worlds“

**Probabilistic Models** [Mich89]
- Extending schema by probabilities
Grouping for Integration

\[ K \cup C \]

<table>
<thead>
<tr>
<th>p_id</th>
<th>fname</th>
<th>lname</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peter</td>
<td>Müller</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Franz</td>
<td>Schmidt</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Wiebke</td>
<td>Meyer</td>
<td>55</td>
</tr>
<tr>
<td>1</td>
<td>(\perp)</td>
<td>Müller</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>(\perp)</td>
<td>Schmidt</td>
<td>(\perp)</td>
</tr>
<tr>
<td>3</td>
<td>(\perp)</td>
<td>Meier</td>
<td>56</td>
</tr>
</tbody>
</table>

\[
\text{SELECT p_id, MAXLEN(fname), CHOOSE(lname,C), MAX(age)} \\
\text{FROM} \\
K \cup C \\
\text{GROUP BY p_id}
\]

- Longest String
- C is favored source
- Highest value

Felix Naumann | Information Systems | Hasso Plattner Institut
SELECT books.isbn, 
    MAXLEN(books.title), 
    MIN(books.price) 
FROM ( 
    SELECT * FROM books1 
    UNION 
    SELECT * FROM books2 
) AS books 
GROUP BY 
    books.isbn
Grouping – Pros and Cons

Pros

- Efficient
  - Implemented by sorting
- Catches duplicates within source and across sources
- Simple / short

Cons

- Restricted to built-in standard aggregate-functions:
  - MAX, MIN, AVG, VAR, STDDEV, SUM, COUNT
- Grouping only by equality
  - ID attribute is necessary
- Outer Union usually not implemented
FUSE BY Syntax Diagram

```
SELECT colref
    | RESOLVE (colref)
    | RESOLVE (colref, function)
    | *

FUSE FROM tablerref

where-clause

FUSE BY (colref) ON ORDER colref

having-clause

order-by-clause
```
Fuse By – Queries

| SELECT * FROM Q1 FUSE BY (Name) | SELECT * FROM Q1 FUSE BY () | SELECT * FROM Q1, Q2 FUSE BY () |

Grouping with coalesce aggregation

Subsumption

Minimum Union

SELECT Name, \text{RESOLVE}(Age, \text{max}), \text{RESOLVE}(\text{Student, vote}), \text{RESOLVE}(\text{Place}), \text{RESOLVE}(\text{Phone})

FUSE FROM Q1, Q2

FUSE BY (Name) ON ORDER Q2.Age DESC
FUSE BY – Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Student</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>Felix</td>
<td>↓</td>
<td>No</td>
<td>Hamburg</td>
</tr>
<tr>
<td>Melanie</td>
<td>22</td>
<td>Yes</td>
<td>↓</td>
</tr>
<tr>
<td>Jens</td>
<td>↓</td>
<td>Yes</td>
<td>Karlsruhe</td>
</tr>
<tr>
<td>Christoph</td>
<td>25</td>
<td>Yes</td>
<td>Berlin</td>
</tr>
<tr>
<td>Sven</td>
<td>26</td>
<td>Yes</td>
<td>Berlin</td>
</tr>
<tr>
<td>Sven</td>
<td>↓</td>
<td>Yes</td>
<td>Berlin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Student</th>
<th>Place</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melanie</td>
<td>22</td>
<td>Yes</td>
<td></td>
<td>030/12345</td>
</tr>
<tr>
<td>Jens</td>
<td>27</td>
<td>↓</td>
<td></td>
<td>030/54321</td>
</tr>
<tr>
<td>Christoph</td>
<td>24</td>
<td>Yes</td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Melanie</td>
<td>21</td>
<td>No</td>
<td></td>
<td>030/98765</td>
</tr>
<tr>
<td>Karsten</td>
<td>24</td>
<td>Yes</td>
<td></td>
<td>↓</td>
</tr>
</tbody>
</table>

SELECT Name,
RESOLVE (Age, max),
RESOLVE (Student, vote),
RESOLVE (Place),
RESOLVE (Phone)
FUSE FROM Q1, Q2
FUSE BY (Name)
ON ORDER Q2.Alter DESC
Query Optimization with Fusion

Correctness?
Completeness?
Efficiency?
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
  - Data Conflicts
  - Relational Operators
  - Conflict Resolution
  - Tools
- Summary
Classification of strategies

conflict resolution strategies

- conflict ignorance
  - Pass it on

- conflict avoidance
  - instance based
    - Take the information
    - No Gossiping
  - metadata based
    - Trust your friends

- conflict resolution
  - instance based
    - deciding
      - Cry with the wolves
      - Roll the dice
    - mediating
      - Meet in the middle
  - metadata based
    - deciding
      - Nothing is older than the news from yesterday
    - mediating

Each with several implementations

Felix Naumann | Information Systems | Hasso Plattner Institut
## Conflict Resolution Functions

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min, Max, Sum,</td>
<td>Standard aggregation</td>
</tr>
<tr>
<td>Count, Avg, StdDev</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>Random choice</td>
</tr>
<tr>
<td>First, Last</td>
<td>Choose first/last value; depends on order</td>
</tr>
<tr>
<td>Longest, Shortest</td>
<td>Choose longest/shortest value</td>
</tr>
<tr>
<td>Choose(source)</td>
<td>Choose value from a particular source</td>
</tr>
<tr>
<td>ChooseDepending(col, val)</td>
<td>Choose depending on val in other column col</td>
</tr>
<tr>
<td>Vote</td>
<td>Majority decision</td>
</tr>
<tr>
<td>Coalesce</td>
<td>Choose first non-null value</td>
</tr>
<tr>
<td>Group, Concat</td>
<td>Group or concatenate all values</td>
</tr>
<tr>
<td>MostRecent</td>
<td>Choose most recent (up-to-date) value</td>
</tr>
<tr>
<td>MostAbstract, MostSpecific</td>
<td>Use a taxonomy / ontology</td>
</tr>
<tr>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
  - Data Conflicts
  - Relational Operators
  - Conflict Resolution
- Tools
- Summary
Visualization of Integrated Data

<table>
<thead>
<tr>
<th>CLU</th>
<th>TITLE</th>
<th>VERSI</th>
<th>COUN</th>
<th>YEAR</th>
<th>ORIGI</th>
<th>GENRE</th>
<th>DIREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>HOPE FLOATS</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>Hop</td>
<td>Unterhaltung</td>
<td>Fore</td>
</tr>
<tr>
<td>14</td>
<td>GOOD WILL H...</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>Goo</td>
<td>Drama</td>
<td>Gus</td>
</tr>
<tr>
<td>15</td>
<td>GODZILLA</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>God</td>
<td>Fantasy</td>
<td>Rola</td>
</tr>
<tr>
<td>16</td>
<td>Gadjo Dilo</td>
<td>franz</td>
<td>Ft/Rum</td>
<td>1998</td>
<td>Gadjo</td>
<td>Unterhaltung</td>
<td>Ten</td>
</tr>
<tr>
<td>17</td>
<td>Deconstruction</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>Dec</td>
<td>Komödie</td>
<td>Woo</td>
</tr>
<tr>
<td>18</td>
<td>City Of Angels</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>City</td>
<td>Drama</td>
<td>Brad</td>
</tr>
<tr>
<td>19</td>
<td>BOOGIE NIGHT...</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>Boo</td>
<td>biografic</td>
<td>Paul</td>
</tr>
<tr>
<td>20</td>
<td>Antz</td>
<td>engl</td>
<td>USA</td>
<td>1998</td>
<td>Antz</td>
<td>Animation</td>
<td>Darn</td>
</tr>
<tr>
<td>21</td>
<td>SPIDER</td>
<td>engl</td>
<td>USA</td>
<td>2002</td>
<td>Spa</td>
<td>Drama</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>SECRETARY</td>
<td>engl</td>
<td>USA</td>
<td>2002</td>
<td>Sect</td>
<td>Komödie</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>S.F.W</td>
<td>engl</td>
<td>USA</td>
<td>1994</td>
<td>S.F.W</td>
<td>Komödie</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Intolerable C...</td>
<td>engl</td>
<td>USA</td>
<td>2003</td>
<td>Intol</td>
<td>Komödie</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>GANGSTER N...</td>
<td>engl</td>
<td>USA</td>
<td>2000</td>
<td>Gangst</td>
<td>Drama</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>From Hell</td>
<td>engl</td>
<td>USA</td>
<td>2001</td>
<td>From</td>
<td>Drama</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>DEATHWATCH</td>
<td>engl</td>
<td>USA</td>
<td>2002</td>
<td>Death</td>
<td>Kriegsfilm</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CHARLOTTE...</td>
<td>engl</td>
<td>USA</td>
<td>2001</td>
<td>Charl</td>
<td>Melodram</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Big Fish</td>
<td>engl</td>
<td>USA</td>
<td>2003</td>
<td>Big</td>
<td>Drama</td>
<td></td>
</tr>
</tbody>
</table>
Bachelorprojekt „Fuz!on“
Bachelorprojekt „Fuz!on“
Bachelorprojekt „Fuz!on“

Fuzzy Fusion

31750025-01 Werner Trimpert Thomas-Man… 89 24943 Kiel 19470524 0461
31758055-01 Artur Heiser Kalligrun I 24939 Kiel 1936016 0461
31765505-01 Siegfried Aswegen Mührer Str. 6 4943 Flensburg 1923000 0461
31772625-01 M. Blankenburg Harmsbr. 48 24116 Kiel 19610727 0461
31780665-01 K. Nagen Peterschr-H 24114 Flensburg 19630013 0461
31789325-01 Manh Tho Knaut Wiedeberger… 37 24943 Flensburg 19280312 0461
31798345-01 horst Bootsmann 6 24937 Flensburg 19281225 0461

Felix Naumann | Information Systems | Hasso Plattner Institut
Tool: fusem
Visualizing semantics and conflicts

(a) Object overlap  
(b) Contradiction Count

(c) Contradicting tuples

<table>
<thead>
<tr>
<th>Object</th>
<th>Semantics</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1</td>
<td>Merge</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>FuseBy</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MatchJoin</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>o2</td>
<td>Merge</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>FuseBy</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MatchJoin</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>o3</td>
<td>Merge</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>FuseBy</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>MatchJoin</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>o4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview

- Introductory example
- Step 1: Schema Matching
- Step 2: Duplicate detection
- Step 3: Data fusion
  - Data Conflicts
  - Relational Operators
  - Conflict Resolution
  - Tools
- Summary
Summary

- **Step 1: Schema Matching**
  - Similarity Measure
  - Combination of methods

- **Step 2: Duplicate Detection**
  - Similarity Measure
  - Algorithm
  - Data Model

- **Step 3: Data Fusion**
  - Relational Operators
  - Conflict Resolution
  - Visualization of Semantics and Overlap
### VIQTOR: Quality Annotations

**Table of Data**

<table>
<thead>
<tr>
<th>did</th>
<th>artist</th>
<th>title</th>
<th>category</th>
<th>genre</th>
<th>year</th>
<th>c澳大</th>
</tr>
</thead>
<tbody>
<tr>
<td>a810b60d</td>
<td>Various Xes</td>
<td>Peppered With Love</td>
<td>reggae</td>
<td>Electronic</td>
<td>2003</td>
<td></td>
</tr>
<tr>
<td>a96925ac</td>
<td>The Pogues</td>
<td>The Pogues</td>
<td>reggae</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a90a270c</td>
<td>Jackie Brown</td>
<td>Look Pon You</td>
<td>reggae</td>
<td>Reggae</td>
<td>1999</td>
<td></td>
</tr>
<tr>
<td>a90c26f9</td>
<td>The Buggles</td>
<td>Younger Than You</td>
<td>reggae</td>
<td>Reggae</td>
<td>2001</td>
<td></td>
</tr>
<tr>
<td>a90b260c</td>
<td>Lee Perry</td>
<td>Dub Fire</td>
<td>reggae</td>
<td>Reggae</td>
<td>2001</td>
<td></td>
</tr>
<tr>
<td>a900249c</td>
<td>CHIM</td>
<td>PNG (XMAS)</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a905590c</td>
<td>FEEL THE R</td>
<td>FEEL THE R</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a90c0b0c</td>
<td>Natural Zion</td>
<td>Natural Zion</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a90a5c0d</td>
<td>Pressure Co</td>
<td>I Want To Tell</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a90f0a50</td>
<td>Best Of World</td>
<td>Jamaica</td>
<td>reggae</td>
<td>Reggae</td>
<td>1994</td>
<td></td>
</tr>
<tr>
<td>a90f0b5d</td>
<td>Lionel Richie</td>
<td>Live &amp; Alive</td>
<td>reggae</td>
<td>Soul</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a90c2500</td>
<td>Latin Prince</td>
<td>Multiple Vibes</td>
<td>reggae</td>
<td>Reggae</td>
<td>1993</td>
<td></td>
</tr>
<tr>
<td>a90d200b</td>
<td>Various Artists</td>
<td>RTI Sound R</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a900250d</td>
<td>Charlie Chaplin</td>
<td>Kings of Rega</td>
<td>reggae</td>
<td>Reggae</td>
<td>2002</td>
<td></td>
</tr>
<tr>
<td>a90d0e00</td>
<td>The Dead Lil Thing</td>
<td>Heartfelt Sea</td>
<td>reggae</td>
<td>Rock</td>
<td>1999</td>
<td></td>
</tr>
<tr>
<td>a90d10c0</td>
<td>Bob Marley</td>
<td>1975-06-24 :...</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a900e0e0</td>
<td>Baha Men</td>
<td>Who Let the Cat Out of the Bag</td>
<td>reggae</td>
<td>Reggae</td>
<td>2000</td>
<td>YEAR: 2000</td>
</tr>
<tr>
<td>a90550d</td>
<td>SES</td>
<td>Surprise</td>
<td>reggae</td>
<td>reggae</td>
<td>2001</td>
<td></td>
</tr>
<tr>
<td>a11c000d</td>
<td>Ras</td>
<td>Rhythmic All A</td>
<td>reggae</td>
<td>Slow Jam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ab00300e</td>
<td>Siemens</td>
<td>Caribbean Sun</td>
<td>reggae</td>
<td>Reggae</td>
<td>2001</td>
<td>YEAR: 2001</td>
</tr>
<tr>
<td>ab03ca0d</td>
<td>La Factora</td>
<td>Total France</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ab004d0e</td>
<td>King Tubby &amp; The Revolutionaries</td>
<td>King Tubby</td>
<td>reggae</td>
<td>Reggae</td>
<td>1996</td>
<td>YEAR: 1996</td>
</tr>
<tr>
<td>ab005e39</td>
<td>Tappa Bush</td>
<td>From the Air</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ab00490c</td>
<td>Jimmy Cliff</td>
<td>Higher &amp; Higher</td>
<td>reggae</td>
<td>Reggae</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ab03470c</td>
<td>Snowpea</td>
<td>Snowpea</td>
<td>reggae</td>
<td>Reggae</td>
<td>1998</td>
<td></td>
</tr>
</tbody>
</table>

**Filtering Criteria**

- **Category:** reggae
PDMS: Incomplete and Selective Mappings

Problem:
Cumulated selections
- implicit in schemata
- explicit in mappings
- Point selections and range selections

Problem:
Cumulated projections
- in schemata
- in mappings