Qualitative Data Cleaning

Xu Chu    Ihab Ilyas
Many Definitions and One Goal

“Extract Value from Data”

For that we ..

- Remove errors
- Fill missing info
- Transform units and formats
- Map and align columns
- Remove duplicate records
- Fix integrity constraints violations
For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights

NYtimes August, 2014

Yes big data is a big business opportunity, but the business value won’t be realized if the information isn’t governed

Forbes Business
Many Technical Challenges

- **Record Linkage and Deduplication**
  - Similarity measures
  - Machine learning for classifying pairs as duplicates or not (unsupervised, supervised, and active)
  - Clustering and handling of transitivity
  - Merging and consolidation of records

A major firm spends 6 months on a single deduplication project of a subset of their data sources
Example: Data Deduplication

Unclean Relation

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ZIP</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Green</td>
<td>51519</td>
<td>30k</td>
</tr>
<tr>
<td>P2</td>
<td>Green</td>
<td>51518</td>
<td>32k</td>
</tr>
<tr>
<td>P3</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P4</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P5</td>
<td>Gree</td>
<td>51519</td>
<td>55k</td>
</tr>
<tr>
<td>P6</td>
<td>Chuck</td>
<td>51519</td>
<td>30k</td>
</tr>
</tbody>
</table>

Clean Relation

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ZIP</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Green</td>
<td>51519</td>
<td>39k</td>
</tr>
<tr>
<td>C2</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>C3</td>
<td>Chuck</td>
<td>51519</td>
<td>30k</td>
</tr>
</tbody>
</table>

Compute Pair-wise Similarity

Cluster Similar Records

Merge Clusters
Many Technical Challenges

- Missing Values
  - Interpreting different types of Nulls
  - Certain answer semantics on possible worlds (many.. many papers)
  - Closed world vs. open-world assumptions and multiple interesting hardness results

*Most real data collected from sensors, surveys, agents, have a high percentage of N/A or nulls, special values (99999) etc.*
Many Technical Challenges

- More Complex Integrity Constraints
  - A declarative language to express data quality rules
  - Ad-hoc repair algorithm to repair violations for each data quality formalism under certain minimality requirements
  - Limited expressiveness (e.g., FD) to get tangible results

*Unfortunately rarely expressed in practice. Most curation tools are rule-based implemented in imperative language*
Example ICs

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>FN</td>
<td>LN</td>
<td>ROLE</td>
<td>CITY</td>
<td>ST</td>
<td>SAL</td>
</tr>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>E</td>
<td>SJ</td>
<td>CA</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>NYC</td>
<td>AZ</td>
</tr>
<tr>
<td>t₄</td>
<td>235</td>
<td>John</td>
<td>Smith</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
</tr>
</tbody>
</table>

Employee Table

Functional dependency:

*City → ST*
## Example ICs

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>CITY</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>E</td>
<td>SJ</td>
<td>CA</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>E</td>
<td>NYC</td>
<td>AZ</td>
<td>75</td>
</tr>
<tr>
<td>t₄</td>
<td>235</td>
<td>John</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
<td>1200</td>
</tr>
</tbody>
</table>

### Employee Table

**Business Rule:**
Two employees of the same role, the one who lives in NYC cannot earn less than the one who does not live in NYC.
### Example ICs

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>CITY</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>E</td>
<td>SJ</td>
<td>CA</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>E</td>
<td>NYC</td>
<td>AZ</td>
<td>75</td>
</tr>
<tr>
<td>t₄</td>
<td>235</td>
<td>John</td>
<td>M</td>
<td>NYC</td>
<td>NY</td>
<td>1200</td>
</tr>
</tbody>
</table>

#### Employee Table

**Business Rule:**
Two employees of the same role in the same city, their salary difference cannot be greater than 100.
## Common Data Quality Issues

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>ZIP</th>
<th>City</th>
<th>State</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Green</td>
<td>60610</td>
<td>Chicago</td>
<td>IL</td>
<td>30k</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>60611</td>
<td>Chicago</td>
<td>IL</td>
<td>32k</td>
</tr>
<tr>
<td>3</td>
<td>Peter</td>
<td>11507</td>
<td>New York</td>
<td>NY</td>
<td>40k</td>
</tr>
<tr>
<td>4</td>
<td>John</td>
<td>11507</td>
<td>New York</td>
<td>NY</td>
<td>40k</td>
</tr>
<tr>
<td>5</td>
<td>Gree</td>
<td>90057</td>
<td>Los Angeles</td>
<td>CA</td>
<td>55k</td>
</tr>
<tr>
<td>6</td>
<td>Chuck</td>
<td>90057</td>
<td>Los Angeles</td>
<td>CA</td>
<td>30k</td>
</tr>
</tbody>
</table>

**Common Data Quality Issues:**
- Duplicates
- Syntactic Error
- Integrity Constraint Violation
- Missing Value
Data Cleaning Process

- Error Detection
  - Qualitative
  - Quantitative (outlier detection)

- Error Repairing
  - Transformation scripts
  - Human involvement
We Will Not Cover

- Details of Deduplication
  - Multiple surveys and tutorials

- Data Profiling: discovering FDs, INDs, etc.
  - Wenfei Fan and Floris Geerts synthesis lecture book
  - Ziawasch Abedjan et al. tutorial

- Consistent Query Answering
  - Leo Betrossi synthesis lecture book
Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

Error Type (What to detect?)
- IC
- Data deduplication
- FD
- CFD
- DC
- Others

Automation (How to detect?)
- Automatic
- Human guided

Analytics Layer (Where to detect?)
- Source
- Target

Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

Error Type (What to detect?)
- IC
- CFD
- DC
- Others

Automation (How to detect?)
- Automatic
- Human guided

Analytics Layer (Where to detect?)
- Source
- Target
FDs and CFDs  [Bohannon et al, ICDE 2007]

- **Functional Dependency (FD):**
  \[ X \rightarrow Y \]
  - Example: City \rightarrow ST or Name,Phone \rightarrow ID

- **Conditional Functional Dependency (CFD):**
  \[(X \rightarrow Y, \mathcal{T}_p)\]
  - An FD defined on a subset of the data
  - Example:
    - ZIP \rightarrow Street is valid on subset of the data where Country = “England”
    - AC = 020 \rightarrow City = London
Matching Dependencies (MDs) [Fan et al, VLDB 2009]

<table>
<thead>
<tr>
<th>FN</th>
<th>LN</th>
<th>St</th>
<th>City</th>
<th>AC</th>
<th>Post</th>
<th>Phn</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>5 Wren St</td>
<td>Ldn</td>
<td>020</td>
<td>WC1H 9SE</td>
<td>3887644</td>
<td>watch</td>
</tr>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>Null</td>
<td>Ldn</td>
<td>020</td>
<td>WC1E 7HX</td>
<td>3887644</td>
<td>necklace</td>
</tr>
</tbody>
</table>

Master: Card

<table>
<thead>
<tr>
<th>FN</th>
<th>LN</th>
<th>St</th>
<th>City</th>
<th>AC</th>
<th>Zip</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>5 Wren St</td>
<td>Ldn</td>
<td>020</td>
<td>WC1H 9SE</td>
<td>3887644</td>
</tr>
</tbody>
</table>

MD: \[\text{Tran}[\text{LN, City, St, Post}] = \text{card}[\text{LN, City, St, Zip}] \wedge \\
\text{Tran}[\text{FN}] \approx \text{Card}[\text{FN}] \Rightarrow \text{Tran}[\text{FN, Phn}] \Leftrightarrow \text{Card}[\text{FN, Tel}]\]
Denial Constraints (DCs) [Chu et al, VLDB 2013]

Formal Definition:

\[ \varphi: \forall t_\alpha, t_\beta, t_\gamma, \ldots \in R, \neg (P_1 \land \ldots \land P_m) \]
\[ P_i: t_x.A \theta t_y.B \text{ or } t_x.A \theta c \]
\[ x, y \in \{\alpha, \beta, \ldots\}, \text{ and } A, B \in R, c \text{ is a constant} \]

- A universal constraint dictates a set of predicate cannot be true together
- Each predicate express a relationship between two cells, or a cell and a constant
Denial Constraints (DCs)

Functional dependency:
\[ CITY \Rightarrow ST \]
\[ \forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.CITY = t_\beta.CITY \land t_\alpha.ST \neq t_\beta.ST) \]

Business Rule:
Two employees of the same Role, the one who lives in NYC cannot earn less than the one who does not live in NYC
\[ \forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ROLE = t_\beta.ROLE \land t_\alpha.CITY = \text{“NYC”} \land t_\beta.CITY \neq \text{“NYC”} \land t_\alpha.SAL < t_\beta.SAL) \]
Other ICs

- **CINDs** [Ma et al, TCS 2014]

- **Metric Functional Dependencies** [Koudas et al, ICDE 2009]

- **Dependable Fixes**
  - Editing Rules [Fan et al, VLDB 2010]
  - Fixing Rules [Wang and Tang, SIGMOD 2014]
  - Sherlock Rules [Interlandi and Tang, ICDE 2015]
Constraint Languages

Language expressiveness

FDs    CFDs    …    DCs    Programmatic Interface

Reasoning and discovery complexity
Integrity Constraints Discovery

- Schema Driven
  - Usually sensitive to the size of the schema
  - Good for long thin tables!

- Instance Driven
  - Usually sensitive to the size of the data
  - Good for fat short tables!

- Hybrid
  - Try to get the best of both worlds
Integrity Constraints Discovery

- **FD Discovery:**
  - TANE: Schema-driven
    - [Huhtala et al, Computer Journal 1999]
  - FASTFD: Instance-driven
    - [Wyss et al, DaWaK, 2001]
  - Hybrid
    - [Papenbrock et al, SIGMOD 2016]

- **DC Discovery:**
  - FASTDC: Instance-driven [Chu et al, VLDB 2013]
Integrity Constraints Discovery

- **FD Discovery:**
  - TANE: Schema-driven
    - [Huhtala et al, Computer Journal 1999]
  - FASTFD: Instance-driven
    - [Wyss et al, DaWaK, 2001]
  - Hybrid
    - [Papenbrock et al, SIGMOD 2016]

- **DC Discovery:**
  - FASTDC: Instance-driven [Chu et al, VLDB 2013]
Given a relational instance $I$ of schema $R$, where $|R| = m$, find (all) minimal, non-trivial FDs that are valid on $I$. An FD is

- **Valid** on $I$ if there does not exist two tuples that violate the FD
- **Minimal** if removing an attribute from its LHS makes it invalid
- **Trivial** if the RHS is a subset of the LHS

We want FDs with only one attribute in RHS
- Generate space of FDs

(a) Space of FDs.

(b) Candidate FDs pruned if $A \rightarrow C$ is valid
FD Validation

\[ \Pi_X = \{\{t_1\}, \{t_2, t_3\}, \{t_4\}\} \]
\[ \Pi_Y = \{\{t_1, t_2, t_3\}, \{t_4\}\} \]
\[ \Pi_{XY} = \{\{t_1\}, \{t_2, t_3\}, \{t_4\}\} \]

\[ X \rightarrow Y \] is a valid FD if and only if

\[ |\Pi_X| = |\Pi_{X \cup Y}| \]
**DC Discovery: Axioms**

**Triviality**

\[ \forall P_i, P_j, \text{ if } P_i \in \text{Imp}(P_j) \text{ then } \neg(\bar{P}_i \land P_j) \text{ is a trivial DC} \]

\[ \forall t_\alpha, t_\beta \in R, \neg (t_\alpha \cdot \text{SAL} = t_\beta \cdot \text{SAL} \land t_\alpha \cdot \text{SAL} > t_\beta \cdot \text{SAL}) \]

<table>
<thead>
<tr>
<th>( \phi )</th>
<th>=</th>
<th>( \neq )</th>
<th>&gt;</th>
<th>&lt;</th>
<th>≥</th>
<th>≤</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi )</td>
<td>( \neq )</td>
<td>=</td>
<td>≤</td>
<td>≥</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>\text{Imp}(\phi)</td>
<td>=, ≥, ≤</td>
<td>( \neq )</td>
<td>&gt;, ≥, ( \neq )</td>
<td>&lt;, ≤, ( \neq )</td>
<td>≥</td>
<td>≤</td>
</tr>
</tbody>
</table>
DC Discovery: Axioms

Augmentation

If $\neg(P_1 \land \ldots \land P_n)$ is valid, then $\neg(P_1 \land \ldots \land P_n \land Q)$ is also valid

Not Minimal

$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \land t_\alpha.ST \neq t_\beta.ST)$

$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \land t_\alpha.ST \neq t_\beta.ST \land t_\alpha.SAL < t_\beta.SAL)$
Transitivity  (more like composition)

If \( \neg(P_1 \land \ldots \land P_n \land Q_1) \), and \( \neg(R_1 \land \ldots \land R_m \land Q_2) \) are valid, and \( Q_2 \in \text{Imp}(Q_1) \), then
\( \neg(P_1 \land \ldots \land P_n \land R_1 \land \ldots \land R_m) \) is valid

\[
\forall t_\alpha, t_\beta \in R, \neg (t_\alpha \ ST = t_\beta \ ST) \land t_\alpha \ SAL < t_\beta \ SAL \land t_\alpha \ TR > t_\beta \ TR
\]

\[
\forall t_\alpha, t_\beta \in R, \neg (t_\alpha \ ZIP = t_\beta \ ZIP) \land t_\alpha \ ST \neq t_\beta \ ST
\]

\[
\forall t_\alpha, t_\beta \in R, \neg (t_\alpha \ ZIP = t_\beta \ ZIP \land t_\alpha \ SAL < t_\beta \ SAL \land t_\alpha \ TR > t_\beta \ TR)
\]
DC Discovery

Given a relational schema R and an instance I, find all non-trivial, minimal DCs that hold on I

Focus on DCs involving at most two tuples
FASTDC [Chu et al, VLDB 2013]

<table>
<thead>
<tr>
<th>TID</th>
<th>I(String)</th>
<th>M(String)</th>
<th>S(Double)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A1</td>
<td>A1</td>
<td>50</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A2</td>
<td>A1</td>
<td>40</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A3</td>
<td>A1</td>
<td>40</td>
</tr>
</tbody>
</table>

- **Employee**

- The space of predicates
  
  \[
  P_1 : t_\alpha.I = t_\beta.I \\
  P_2 : t_\alpha.I \neq t_\beta.I \\
  P_3 : t_\alpha.M = t_\beta.M \\
  P_4 : t_\alpha.M \neq t_\beta.M \\
  P_5 : t_\alpha.S = t_\beta.S \\
  P_6 : t_\alpha.S \neq t_\beta.S \\
  P_7 : t_\alpha.S > t_\beta.S \\
  P_8 : t_\alpha.S \leq t_\beta.S \\
  P_9 : t_\alpha.S < t_\beta.S \\
  P_{10} : t_\alpha.S \geq t_\beta.S \\
  P_{11} : t_\alpha.I = t_\alpha.M \\
  P_{12} : t_\alpha.I \neq t_\alpha.M \\
  P_{13} : t_\alpha.I = t_\beta.M \\
  P_{14} : t_\alpha.I \neq t_\beta.M \\
  \]

- Any combination of predicates constitutes a candidate DC
\overline{(P_i \land P_j \land P_k)} \text{ is a valid DC w.r.t. } I

\begin{itemize}
  \item For every tuple pair in I, \( P_i, P_j, P_k \) cannot be true together
  \item For every tuple pair in I, at least one of \( P_i, P_j, P_k \) is false
  \item For every tuple pair in I, at least one of \( \overline{P_i}, \overline{P_j}, \overline{P_k} \) is true
\end{itemize}

\( \overline{P_i}, \overline{P_j}, \overline{P_k} \) covers the \textit{set of true predicates for every tuple pair}
FASTDC

<table>
<thead>
<tr>
<th>TID</th>
<th>I(String)</th>
<th>M(String)</th>
<th>S(Double)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A1</td>
<td>A1</td>
<td>50</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A2</td>
<td>A1</td>
<td>40</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A3</td>
<td>A1</td>
<td>40</td>
</tr>
</tbody>
</table>

$E_{v_{1}}$

\(<t_2, t_3>, <t_3, t_2> \{P_2, P_3, P_5, P_8, P_{10}, P_{12}, P_{14}\}\)$

\(<t_2, t_1>, <t_3, t_1> \{P_2, P_3, P_6, P_8, P_9, P_{12}, P_{14}\}\)$

\(<t_1, t_2>, <t_1, t_3> \{P_2, P_3, P_6, P_7, P_{10}, P_{11}, P_{13}\}\)$

$\{P_{10}, P_{14}\}$ covers the set of true predicates for every tuple pair

\[\forall t_\alpha, t_\beta \in R, \neg(P_{10} \land P_{14})\]

\[\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.S < t_\beta.S \land t_\alpha.I = t_\beta.M)\] is a valid DC

$\{P_5, P_{10}, P_{14}\}$ covers the set of true predicates for every tuple pair

\[\neg(P_{10} \land P_{14} \land P_5)\] is a valid DC, but not minimal
FASTDC

\[ \text{Ev}_{1} \{P_2, P_3, P_5, P_8, P_{10}, P_{12}, P_{14}\} \]
\[ \{P_2, P_3, P_6, P_8, P_9, P_{12}, P_{14}\} \]
\[ \{P_2, P_3, P_6, P_7, P_{10}, P_{11}, P_{13}\} \]

\[ P_8 \in \text{Imp}(\neg P_6) \]
\[ P_{10} \in \text{Imp}(\neg P_6) \]
\[ P_{11} \in \text{Imp}(\neg P_{12}) \]
\[ P_{13} \in \text{Imp}(\neg P_{14}) \]

- : valid DC
- : pruned branch
- : invalid DC

Diagram with nodes representing different sets and conditions for each node.
**Key**: \{AC, PH\}  \quad \forall t_\alpha, t_\beta \in R, \neg (t_\alpha . AC = t_\beta . AC \land t_\alpha . PH = t_\beta . PH)

**Domain**: MS \in \{S, M\}  \quad \forall t_\alpha \in R, \neg (t_\alpha . MS \neq S \land t_\alpha . MS \neq M)

**FD**: ZIP \rightarrow ST  \quad \forall t_\alpha, t_\beta \in R, \neg (t_\alpha . ZIP = t_\beta . ZIP \land t_\alpha . ST \neq t_\beta . ST)

**CFD**: CT = Los Angeles \rightarrow ST = CA  \quad \forall t_\alpha \in R, \neg (t_\alpha . CT = Los Angeles \land t_\alpha . ST \neq CA)

**Check**: SAL \geq STX  \quad \forall t_\alpha \in R, \neg (t_\alpha . SAL < t_\alpha . STX)

**Business logic**: \forall t_\alpha, t_\beta \in R, \neg (t_\alpha . ST = t_\beta . ST \land t_\alpha . SAL < t_\beta . SAL \land t_\alpha . TR > t_\beta . TR)
Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

Error Type (What to detect?)
- IC
- Data deduplication
- FD
- CFD
- DC
- Others

Automation (How to detect?)
- Automatic
- Human guided

Analytics Layer (Where to detect?)
- Source
- Target
Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

- Error Type (What to detect?)
  - IC
  - Data deduplication
  - FD
  - CFD
  - DC
  - Others

- Automation (How to detect?)
  - Automatic
  - Human guided

- Analytics Layer (Where to detect?)
  - Source
  - Target
Holistic Error Detection

- **Vertex**: Cell in the database
- **Hyperedge**: A set of cells that violate a DC

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>85376</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>90012</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>85376</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>

Employee Table

Zip → ST

[Chu et al, ICDE 2013]
[Kolahi and Lakshmanan ICDT 2009]
Holistic Error Detection

- **Vertex**: Cell in the database
- **Hyperedge**: A set of cells that violate a DC

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>

∀t₁, t₂ ∈ Emp, ¬(t₁.ST = t₂.ST ∧ t₁.ROLE = “M” ∧ t₂.ROLE = “E” ∧ t₁.SAL < t₂.SAL)
Holistic Error Detection

- **Vertex**: Cell in the database
- **Hyperedge**: A set of cells that violate a DC

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>

Employee Table

```
<table>
<thead>
<tr>
<th>t₁.ROLE</th>
<th>t₁.ZIP</th>
<th>t₁.ST</th>
<th>t₁.SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₂.ROLE</td>
<td>t₂.ZIP</td>
<td>t₂.ST</td>
<td>t₂.SAL</td>
</tr>
<tr>
<td>t₃.ZIP</td>
<td>t₃.ST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Zip → ST**

\[ \forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \land t_\alpha.ROLE = "M" \land t_\beta.ROLE = "E" \land t_\alpha.SAL < t_\beta.SAL) \]
Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

Error Type (What to detect?)
- IC
- Data deduplication
- FD
- CFD
- DC
- Others

Automation (How to detect?)
- Automatic
- Human guided

Analytics Layer (Where to detect?)
- Source
- Target
CrowdER: [Wang et al, VLDB 2012]

- Human-Intelligence Task (HIT)

\[ O(n^2) \times \]
CrowdER: Batching Strategies

- Pair-based HIT

\[ O(n^2/k) \times \]
CrowdER: Batching Strategies

- Cluster-based HIT

$O(n^2/k^2) \times$
CrowdER: Workflow

(a) Remove the pairs whose likelihoods < 0.2

(b) Generate HITs to verify the pairs of records

(c) Output matching pairs
CrowdER: Workflow

Cluster-size size threshold k

Minimize the number of HITs

NP-Hard

(r₁, r₂, 0.90)
(r₄, r₆, 0.85)
(r₁, r₇, 0.82)
(r₃, r₄, 0.76)
(r₄, r₇, 0.70)
(r₈, r₉, 0.55)
(r₂, r₃, 0.45)
(r₂, r₇, 0.35)
(r₃, r₅, 0.31)
(r₄, r₅, 0.20)
(r₃, r₆, 0.15)
(r₁, r₃, 0.10)

...
Error Detection Techniques Taxonomy

Qualitative Error Detection Techniques

Error Type (What to detect?)
- IC
- Data deduplication
- FD
- CFD
- DC
- Others

Automation (How to detect?)
- Automatic
- Human guided

Analytics Layer (Where to detect?)
- Source
- Target
Decoupled in Space and Time

(1) In the same shop, the average salary for the managers (Grd=2) should be higher than the one for the staff (Grd=1)

(2) A bigger shop cannot have a smaller number of staff

(3) Phone number must have country code and local number

(4) S1.NAME is NOT NULL

(5) length(S3.NAME) < 30

\( t_\alpha.Shop = t_\beta.Shop \land \\
 t_\alpha.Avgsal > t_\beta.Avgsal \\
 \land t_\alpha.Grd < t_\beta.Grd \\
 \neg(t_\alpha.Size > t_\beta.Size \land \\
 t_\alpha.\#Emps < t_\beta.\#Emps) \)
### Calls for a New Solution

#### Error Fixing

<table>
<thead>
<tr>
<th></th>
<th>Target</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target</strong></td>
<td>Traditional Data Repair Algorithms</td>
<td><strong>Descriptive and Prescriptive Data Cleaning</strong></td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Dependency Propagation</td>
<td>Traditional Data Repair Algorithms</td>
</tr>
</tbody>
</table>

- **Constraints Declaration**
  - DBRx: [Chalamalla et al., SIGMOD 2014]
  - DataXRay: [Wang et al., SIGMOD 2015]
  - QOCO: [Bergman et al., VLDB 2015]
DBRx Architecture [Chalamalla et al, SIGMOD 2014]
Technical Challenges

- **Errors Propagation**
  - Blowup (e.g., Aggregates)
  - Propagation Level (violations vs Fixes)
  - Distributing Responsibilities

- **Source Error Identification**
  - Assign Weights based on Query and Error Semantics
  - Accumulate Evidences (different Violation Semantics)

- **Explain Errors**
Tracing the Sources of Errors

**SELECT** Shops.SId as Shop, Size, Emps.Grd, AVG(Emps.Sal) as AvgSal, COUNT(EId) as #Emps, ‘US’ as Region
FROM US.Emps JOIN US.Shops ON SId
GROUP BY SId, Size, Grd

Average salary of higher grade in the same shop should be higher!
## Error Contribution Scores

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>e4 [‘’,$\frac{1}{3}$]</td>
<td>91 [1,$\frac{1}{3}$]</td>
<td>1 [1,$\frac{1}{3}$]</td>
<td>NY1 [1,$\frac{1}{3}$]</td>
<td>[0,1]</td>
</tr>
<tr>
<td>$t_2$</td>
<td>e5</td>
<td>99 [0,’’]</td>
<td>2 [1,’’]</td>
<td>NY1 [1,’’]</td>
<td>[1,’’]</td>
</tr>
<tr>
<td>$t_3$</td>
<td>e7 [‘’,$\frac{1}{3}$]</td>
<td>93 [1,$\frac{1}{3}$]</td>
<td>1 [1,$\frac{1}{3}$]</td>
<td>NY1 [1,$\frac{1}{3}$]</td>
<td>[0,1]</td>
</tr>
<tr>
<td>$t_4$</td>
<td>e8 [‘’,$\frac{1}{3}$]</td>
<td>116 [1,$\frac{1}{3}$]</td>
<td>1 [1,$\frac{1}{3}$]</td>
<td>NY1 [1,$\frac{1}{3}$]</td>
<td>[1,1]</td>
</tr>
<tr>
<td>$t_5$</td>
<td>e11 [‘’,$\frac{1}{2}$]</td>
<td>89</td>
<td>1 [‘’,$\frac{1}{2}$]</td>
<td>NY2</td>
<td>[‘’,$0$]</td>
</tr>
<tr>
<td>$t_6$</td>
<td>e13</td>
<td>94</td>
<td>2</td>
<td>NY2</td>
<td>[‘’]</td>
</tr>
<tr>
<td>$t_7$</td>
<td>e14 [‘’,$\frac{1}{2}$]</td>
<td>91</td>
<td>1 [‘’,$\frac{1}{2}$]</td>
<td>NY2</td>
<td>[‘’,$0$]</td>
</tr>
<tr>
<td>$t_8$</td>
<td>e18</td>
<td>98</td>
<td>2</td>
<td>NY2</td>
<td>[‘’]</td>
</tr>
<tr>
<td>$t_9$</td>
<td>e14</td>
<td>94</td>
<td>2</td>
<td>LA1</td>
<td>[‘’]</td>
</tr>
<tr>
<td>$t_{10}$</td>
<td>e18</td>
<td>116</td>
<td>2</td>
<td>LA1</td>
<td>[‘’]</td>
</tr>
</tbody>
</table>

### $cs_v(c)$:
Contribution of this cell to the aggregate

<table>
<thead>
<tr>
<th>Shops</th>
<th>SId [CSV]</th>
<th>Size [CSV]</th>
<th>[RSV]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{12}$</td>
<td>NY1 [2,’’]</td>
<td>46 [‘’,$1$]</td>
<td>[1,1]</td>
</tr>
<tr>
<td>$t_{13}$</td>
<td>NY2</td>
<td>62 [‘’,$1$]</td>
<td>[‘’,$1$]</td>
</tr>
<tr>
<td>$t_{14}$</td>
<td>LA1</td>
<td>35</td>
<td>[‘’]</td>
</tr>
</tbody>
</table>

### $rs_v(t)$:
Removing $t_4$ eliminates the violations
Identifying Likely Errors

- Maximize a gain function of adding more source errors

\[
Gain(H_v) = \sum_{s \in H_v} c_v(s) - \sum_{1 \leq j \leq |H_v|} \sum_{j < k \leq |H_v|} D(s_j, s_k)
\]

\[
D(s_j, s_k) = |c_v(s_j) - c_v(s_k)|
\]

\[
c_v(s) = cs_v(s) + rs_v(s)
\]

<table>
<thead>
<tr>
<th>tid</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>0.67</td>
</tr>
<tr>
<td>s_2</td>
<td>0.54</td>
</tr>
<tr>
<td>s_3</td>
<td>0.47</td>
</tr>
<tr>
<td>s_4</td>
<td>0.08</td>
</tr>
<tr>
<td>s_5</td>
<td>0.06</td>
</tr>
<tr>
<td>s_6</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Gain = 1.08

<table>
<thead>
<tr>
<th>tid</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>0.67</td>
</tr>
<tr>
<td>s_2</td>
<td>0.54</td>
</tr>
<tr>
<td>s_3</td>
<td>0.47</td>
</tr>
<tr>
<td>s_4</td>
<td>0.08</td>
</tr>
<tr>
<td>s_5</td>
<td>0.06</td>
</tr>
<tr>
<td>s_6</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Gain = 1.28

<table>
<thead>
<tr>
<th>tid</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>0.67</td>
</tr>
<tr>
<td>s_2</td>
<td>0.54</td>
</tr>
<tr>
<td>s_3</td>
<td>0.47</td>
</tr>
<tr>
<td>s_4</td>
<td>0.08</td>
</tr>
<tr>
<td>s_5</td>
<td>0.06</td>
</tr>
<tr>
<td>s_6</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Gain = -0.08
## Error Explanation

### Likely Error Tuples

<table>
<thead>
<tr>
<th>Emps</th>
<th>EId</th>
<th>Name</th>
<th>Dept</th>
<th>Sal</th>
<th>Grd</th>
<th>SId</th>
<th>JoinYr</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>e4</td>
<td>John</td>
<td>S</td>
<td>91</td>
<td>1</td>
<td>NY1</td>
<td>2012</td>
</tr>
<tr>
<td>$t_2$</td>
<td>e5</td>
<td>Anne</td>
<td>D</td>
<td>99</td>
<td>2</td>
<td>NY1</td>
<td>2012</td>
</tr>
<tr>
<td>$t_3$</td>
<td>e7</td>
<td>Mark</td>
<td>S</td>
<td>93</td>
<td>1</td>
<td>NY1</td>
<td>2012</td>
</tr>
<tr>
<td>$t_4$</td>
<td>e8</td>
<td>Claire</td>
<td>S</td>
<td>116</td>
<td>1</td>
<td>NY1</td>
<td>2012</td>
</tr>
<tr>
<td>$t_5$</td>
<td>e11</td>
<td>Ian</td>
<td>R</td>
<td>89</td>
<td>1</td>
<td>NY2</td>
<td>2012</td>
</tr>
<tr>
<td>$t_6$</td>
<td>e13</td>
<td>Laure</td>
<td>R</td>
<td>94</td>
<td>2</td>
<td>NY2</td>
<td>2012</td>
</tr>
<tr>
<td>$t_7$</td>
<td>e14</td>
<td>Mary</td>
<td>E</td>
<td>91</td>
<td>1</td>
<td>NY2</td>
<td>2012</td>
</tr>
<tr>
<td>$t_8$</td>
<td>e18</td>
<td>Bill</td>
<td>D</td>
<td>98</td>
<td>2</td>
<td>NY2</td>
<td>2012</td>
</tr>
<tr>
<td>$t_9$</td>
<td>e14</td>
<td>Mike</td>
<td>R</td>
<td>94</td>
<td>2</td>
<td>LA1</td>
<td>2011</td>
</tr>
<tr>
<td>$t_{10}$</td>
<td>e18</td>
<td>Claire</td>
<td>E</td>
<td>116</td>
<td>2</td>
<td>LA1</td>
<td>2011</td>
</tr>
</tbody>
</table>

### Possible Explanations

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Recall</th>
<th>Precision</th>
<th>Concise</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Dept} = s$</td>
<td>Low</td>
<td>High</td>
<td>Concise</td>
</tr>
<tr>
<td>$\text{eid} = e_4 \lor \text{eid} = e_7 \lor$</td>
<td>High</td>
<td>High</td>
<td>Verbose</td>
</tr>
<tr>
<td>$\text{eid} = e_8 \lor \text{eid} = e_{14}$</td>
<td>High</td>
<td>Low</td>
<td>Concise</td>
</tr>
</tbody>
</table>
Data Repairing
Data Repairing Techniques Taxonomy

Data Repairing Techniques

Repair target
(What to repair?)
- Data
- Rules
- Both

Automation
(How to repair?)
- Automatic
- Human guided

Update model
(Where to repair?)
- In place
- Model based

Data Repairing Techniques Taxonomy

Data Repairing Techniques

Repairs target
(What to repair?)
- Data
- Rules
- Both

Automation
(How to repair?)
- Automatic
- Human guided

Update model
(Where to repair?)
- In place
- Model based

Piece-meal
Holistic
Repair Automation

- Most automatic repairing techniques adopt the “minimality” of repairs principle

- Repairing techniques in practice are predominantly manual and semi-automatic at best

- Will survey both
Data Repairing Techniques Taxonomy

Data Repairing Techniques

- **Repair target** (What to repair?)
  - Data
  - Rules
  - Both

- **Automation** (How to repair?)
  - Automatic
  - Human guided

- **Update model** (Where to repair?)
  - In place
  - Model based

- **Holistic**
  - Piece-meal

- **Update model** (Where to repair?)
  - In place
  - Model based
Data Repairing Techniques Taxonomy

Data Repairing Techniques

Repair target
(What to repair?)
- Data
- Rules
- Both

Automation
(How to repair?)
- Automatic
- Human guided

Update model
(Where to repair?)
- In place
- Model based

Piece-meal
Holistic
Data Repair by Value Update

- I is a **dirty** database if I \( \not\models \Sigma \), and I\(_j\) is a repair for I if I\(_j\) \( \models \Sigma \)
- For a repair I\(_j\), \( \Delta(I_j) \) is the set of changed cells in I\(_j\)

\[\begin{array}{ll}
\hline
& A & B \\
\hline
I & & \\
t_1 & 1 & 2 \\
t_2 & 1 & 3 \\
t_3 & 1 & 3 \\
t_4 & 4 & 5 \\
\hline
\end{array}\]

\[\begin{array}{ll}
\hline
& A & B \\
\hline
I_1 & & \\
t_1 & 1 & 3 \\
t_2 & 1 & 3 \\
t_3 & 1 & 3 \\
t_4 & 4 & 5 \\
\hline
\end{array}\]

\[\begin{array}{ll}
\hline
& A & B \\
\hline
I_2 & & \\
t_1 & 1 & 2 \\
t_2 & 1 & 2 \\
t_3 & 1 & 2 \\
t_4 & 4 & 5 \\
\hline
\end{array}\]

\( \Sigma = \{A \rightarrow B\} \)

\( \Delta(I_1) = \{t_1[B]\} \)

\( \Delta(I_2) = \{t_2[B], t_3[B]\} \)
Data Only Repairing

- **Cardinality-Minimal repairs**
  - Commonly used in obtaining a single repair automatically
  - Repairs with the minimum number of changes
  - $I_1$ is Card-Min iff $\not\exists I_2$ s.t. $|\Delta(I_2)| < |\Delta(I_1)|$

<table>
<thead>
<tr>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2</td>
<td>1 3</td>
<td>1 3</td>
<td>4 5</td>
</tr>
</tbody>
</table>

FD: $\{A \rightarrow B\}$

<table>
<thead>
<tr>
<th>$I_1$</th>
<th>$I_2$</th>
<th>$I_3$</th>
<th>$I_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B</td>
<td>A B</td>
<td>A B</td>
<td>A B</td>
</tr>
<tr>
<td>1 3</td>
<td>1 2</td>
<td>1 2</td>
<td>7 3</td>
</tr>
<tr>
<td>1 3</td>
<td>1 2</td>
<td>1 5</td>
<td>1 3</td>
</tr>
<tr>
<td>1 3</td>
<td>1 2</td>
<td>1 5</td>
<td>1 5</td>
</tr>
<tr>
<td>4 5</td>
<td>4 5</td>
<td>4 5</td>
<td>4 5</td>
</tr>
</tbody>
</table>
Data Repairing Techniques Taxonomy

Data Repairing Techniques

- **Repair target** (What to repair?)
  - Data
  - Rules
  - Both

- **Automation** (How to repair?)
  - Automatic
  - Human guided

- **Update model** (Where to repair?)
  - In place
  - Model based

- Piece-meal
  - Holistic
### FD Repairing [Bohannon et al, SIGMOD 2005]

<table>
<thead>
<tr>
<th>t</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>t₂</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₃</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₄</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**FD:** \{A \rightarrow B\}

**Building Equivalence Classes**

<table>
<thead>
<tr>
<th>t</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>t₂</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₃</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₄</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**Resolving Equivalence Classes**

<table>
<thead>
<tr>
<th>t</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₂</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₃</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>t₄</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**FD:** \{A \rightarrow B\}
Data Repairing Techniques Taxonomy

Data Repairing Techniques

- Repair target (What to repair?)
  - Data
  - Rules
  - Both

- Automation (How to repair?)
  - Automatic
  - Human guided

- Update model (Where to repair?)
  - In place
  - Model based

Piece-meal

Holistic
Holistic Data Repairing [Chu et al, ICDE 2013]

- **Vertex**: Cell in the database
- **Hyperedge**: A set of cells that violate a DC

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>85376</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>90012</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>85376</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>

**Employee Table**

\[
\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \land t_\alpha.ROLE = "M"
\land t_\beta.ROLE = "E" \land t_\alpha.SAL < t_\beta.SAL)
\]
Step 1: Minimal Vertex Cover

- A minimal set of vertices that are intersecting with every hyperedge

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>t₂</td>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>t₃</td>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>
A set of conditions that need to be satisfied to resolve all violations

Condition to resolve $e_1$ by changing $t_1.ST$:
$$t_1.ST = t_3.ST$$

Condition to resolve $e_2$ by changing $t_1.ST$:
$$t_1.ST \neq t_2.ST$$

$Zip \Rightarrow ST$

$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \land t_\alpha.ROLE = "M" \land t_\beta.ROLE = "E" \land t_\alpha.SAL < t_\beta.SAL)$$
Step 3: Get Updates

- A set of assignments satisfying the conditions, with minimal number of changed cells

<table>
<thead>
<tr>
<th>ID</th>
<th>FN</th>
<th>LN</th>
<th>ROLE</th>
<th>ZIP</th>
<th>ST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>Anne</td>
<td>Nash</td>
<td>E</td>
<td>85376</td>
<td>NY</td>
<td>110</td>
</tr>
<tr>
<td>211</td>
<td>Mark</td>
<td>White</td>
<td>M</td>
<td>90012</td>
<td>NY</td>
<td>80</td>
</tr>
<tr>
<td>386</td>
<td>Mark</td>
<td>Lee</td>
<td>E</td>
<td>85376</td>
<td>AZ</td>
<td>75</td>
</tr>
</tbody>
</table>

Gradually increase the number of cells that are going to be changed, until reach a solution

Suppose we only want to change t1.ST

t2.ST = NY

t3.ST = AZ
More Holistic Data Repairing [Fan et al, SIGMOD 2011]

<table>
<thead>
<tr>
<th>FN</th>
<th>LN</th>
<th>St</th>
<th>City</th>
<th>AC</th>
<th>Post</th>
<th>Phn</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>5 Wren St</td>
<td>Ldn</td>
<td>020</td>
<td>WC1H 9SE</td>
<td>3887644</td>
<td>watch</td>
</tr>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>5 Wren St</td>
<td>Ldn</td>
<td>020</td>
<td>WC1E 7HX</td>
<td>3887644</td>
<td>necklace</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FN</th>
<th>LN</th>
<th>St</th>
<th>City</th>
<th>AC</th>
<th>Zip</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert</td>
<td>Brady</td>
<td>5 Wren St</td>
<td>Ldn</td>
<td>020</td>
<td>WC1H 9SE</td>
<td>3887644</td>
</tr>
</tbody>
</table>

CFD: \( \text{Tran}( \text{AC} = 020 \rightarrow \text{City} = \text{Ldn}) \)

CFD: \( \text{Tran}( \text{FN} = \text{Bob} \rightarrow \text{FN} = \text{Robert}) \)

MD: \( \text{Tran}[\text{LN}, \text{City}, \text{St}, \text{Post}] = \text{card}[\text{LN}, \text{City}, \text{St}, \text{Zip}] \wedge \text{Tran}[\text{FN}] \approx \text{Card}[\text{FN}] \rightarrow \text{Tran}[\text{FN}, \text{Phn}] \Leftrightarrow \text{Card}[\text{FN}, \text{Tel}] \)

FD: \( \text{Tran}( \text{City}, \text{Phn} \rightarrow \text{St}, \text{AC}, \text{Post}) \)
Data Repairing Techniques Taxonomy

Data Repairing Techniques

Repair target (What to repair?)
- Data
- Rules
- Both

Automation (How to repair?)
- Automatic
- Human guided

Update model (Where to repair?)
- In place
- Model based

Piece-meal
Holistic
Data & Rules Repairing: Motivating Example

- Car Database
  - Model → Make was satisfied by Car databases till Mazda 323 was introduced (Conflicting with BMW 323)
  - Could be corrected to Model, Cylinders → Make

- US presidents Database
  - LastName, FirstName → StartYear, EndYear was satisfied till the election of George W. Bush
  - Should be corrected to LastName, MiddleInit, FirstName → StartYear, EndYear

[Chiang and Miller, ICDE 2011]
[Beskales et al, ICDE 2013]
Relative Trust

- In reality, both data and constraints (FDs) can be wrong
- The relative trust in data vs. FDs determines how we should repair data and FDs
### Example

<table>
<thead>
<tr>
<th>GivenName</th>
<th>Surname</th>
<th>BirthDate</th>
<th>Gender</th>
<th>Phone</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-213-1211</td>
</tr>
<tr>
<td>t₂</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-988-9211</td>
</tr>
<tr>
<td>t₃</td>
<td>Hong</td>
<td>Li</td>
<td>27 Oct 1972</td>
<td>Female</td>
<td>591-977-1244</td>
</tr>
<tr>
<td>t₄</td>
<td>Hong</td>
<td>Li</td>
<td>8 Mar 1979</td>
<td>Female</td>
<td>498-214-5822</td>
</tr>
<tr>
<td>t₅</td>
<td>Ning</td>
<td>Wu</td>
<td>3 Nov 1982</td>
<td>Male</td>
<td>313-134-9241</td>
</tr>
<tr>
<td>t₆</td>
<td>Ning</td>
<td>Wu</td>
<td>8 Nov 1982</td>
<td>Male</td>
<td>323-456-3452</td>
</tr>
</tbody>
</table>

Surname, GivenName $\rightarrow$ Income
Example (Trusted FD)

<table>
<thead>
<tr>
<th>GivenName</th>
<th>Surname</th>
<th>BirthDate</th>
<th>Gender</th>
<th>Phone</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-213-1211</td>
<td>120k</td>
</tr>
<tr>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-988-9211</td>
<td>120k</td>
</tr>
<tr>
<td>Hong</td>
<td>Li</td>
<td>27 Oct 1972</td>
<td>Female</td>
<td>591-977-1244</td>
<td>90k</td>
</tr>
<tr>
<td>Hong</td>
<td>Li</td>
<td>8 Mar 1979</td>
<td>Female</td>
<td>498-214-5822</td>
<td>90k</td>
</tr>
<tr>
<td>Ning</td>
<td>Wu</td>
<td>3 Nov 1982</td>
<td>Male</td>
<td>313-134-9241</td>
<td>95k</td>
</tr>
<tr>
<td>Ning</td>
<td>Wu</td>
<td>8 Nov 1982</td>
<td>Male</td>
<td>323-456-3452</td>
<td>95k</td>
</tr>
</tbody>
</table>

Surname, GivenName $\rightarrow$ Income
Example (Trusted Data)

<table>
<thead>
<tr>
<th></th>
<th>GivenName</th>
<th>Surname</th>
<th>BirthDate</th>
<th>Gender</th>
<th>Phone</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-213-1211</td>
<td>120k</td>
</tr>
<tr>
<td>t₂</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-988-9211</td>
<td>100k</td>
</tr>
<tr>
<td>t₃</td>
<td>Hong</td>
<td>Li</td>
<td>27 Oct 1972</td>
<td>Female</td>
<td>591-977-1244</td>
<td>90k</td>
</tr>
<tr>
<td>t₄</td>
<td>Hong</td>
<td>Li</td>
<td>8 Mar 1979</td>
<td>Female</td>
<td>498-214-5822</td>
<td>84k</td>
</tr>
<tr>
<td>t₅</td>
<td>Ning</td>
<td>Wu</td>
<td>3 Nov 1982</td>
<td>Male</td>
<td>313-134-9241</td>
<td>90k</td>
</tr>
<tr>
<td>t₆</td>
<td>Ning</td>
<td>Wu</td>
<td>8 Nov 1982</td>
<td>Male</td>
<td>323-456-3452</td>
<td>95k</td>
</tr>
</tbody>
</table>

Surname, GivenName, BirthDate, Phone \(\rightarrow\) Income
Example (Equally-trusted Data and FD)

<table>
<thead>
<tr>
<th></th>
<th>GivenName</th>
<th>Surname</th>
<th>BirthDate</th>
<th>Gender</th>
<th>Phone</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-213-1211</td>
<td>120k</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>Danielle</td>
<td>Blake</td>
<td>9 Dec 1970</td>
<td>Female</td>
<td>817-988-9211</td>
<td>120k</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>Hong</td>
<td>Li</td>
<td>27 Oct 1972</td>
<td>Female</td>
<td>591-977-1244</td>
<td>90k</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>Hong</td>
<td>Li</td>
<td>8 Mar 1979</td>
<td>Female</td>
<td>498-214-5822</td>
<td>84k</td>
</tr>
<tr>
<td>( t_5 )</td>
<td>Ning</td>
<td>Wu</td>
<td>3 Nov 1982</td>
<td>Male</td>
<td>313-134-9241</td>
<td>90k</td>
</tr>
<tr>
<td>( t_6 )</td>
<td>Ning</td>
<td>Wu</td>
<td>8 Nov 1982</td>
<td>Male</td>
<td>323-456-3452</td>
<td>95k</td>
</tr>
</tbody>
</table>

Surname, GivenName, \( \text{BirthDate} \) \( \rightarrow \) Income
Data Repair

- We repair instance I by modifying multiple cells and produce I’

- $\text{dist}_d(I,I’)$ is the number of different cells between I and I’
Repairing a set of FDs

- We repair an FD $X \rightarrow A$ by adding one or more attributes to the LHS.

- Let $w(Y)$ be a weight reflecting the penalty of adding attribute set $Y$ to $X$.
  - E.g., the number of attributes in $Y$, distinct values of $Y$ in $I$, entropy of $Y$.

- Let $\text{dist}_c(\Sigma, \Sigma')$ be the sum of $w(Y)$ across all changed FDs.
Relative Trust [Beskales et al, ICDE 2013]

\[ \text{dist}_{(I, I')} \]

\[ \text{dist}_{d}(I, I') \]

\[ \text{dist}_{c}(\Sigma, \Sigma') \]

\[ (I', \Sigma'): I' \models \Sigma' \]

Maximum number of cell changes (\(\tau\))
A Unified Cost Model [Chiang and Miller, ICDE 2011]

- Minimum description Length Principle
  - Find a model $M$ w.r.t. $\Sigma$ that can represent the data as much as possible

- $DL(M) = L(M) + L(I|M)$
  - $L(M)$: Length of the model
  - $L(I|M)$: Length of data given $M$
A Unified Cost Model: Data Repair

- **M:** empty
  - \(L(M) = 0\)
  - \(L(I|M) = 27\)
  - \(DL = 27\)

- **M:**
  - \(L(M) = 3 + 2 \times 6 = 15\)
  - \(L(I|M) = 0\)
  - \(DL = 15\)

**FD:** \(\{\text{District, Region} \rightarrow AC\}\)

<table>
<thead>
<tr>
<th>District</th>
<th>Region</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
<tr>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
</tbody>
</table>
A Unified Cost Model: FD Repair

- **M**: empty
  - \(L(M) = 0\)
  - \(L(I|M) = 36\)
  - \(DL = 36\)

- **M**: empty
  - \(L(M) = 12\)
  - \(L(I|M) = 0\)
  - \(DL = 12\)

**FD**: \{Municipal, District, Region \(\rightarrow\) AC\}

<table>
<thead>
<tr>
<th>Municipal</th>
<th>District</th>
<th>Region</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glendale</td>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Glendale</td>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Glendale</td>
<td>Brook</td>
<td>Granville</td>
<td>412</td>
</tr>
<tr>
<td>Guildwood</td>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Guildwood</td>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Guildwood</td>
<td>Brook</td>
<td>Granville</td>
<td>553</td>
</tr>
<tr>
<td>Moore</td>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
<tr>
<td>Moore</td>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
<tr>
<td>Moore</td>
<td>Brook</td>
<td>Granville</td>
<td>725</td>
</tr>
</tbody>
</table>
Data Repairing Techniques Taxonomy

Data Repairing Techniques

- **Repair target**
  - Data
  - Rules
  - Both

- **Automation**
  - Automatic
  - Human guided

- **Update model**
  - In place
  - Model based
Guided Data Repair (GDR) [Yakout et al, VLDB 2011]
### GDR: Generate Possible Repairs

<table>
<thead>
<tr>
<th>Name</th>
<th>SRC</th>
<th>STR</th>
<th>CT</th>
<th>STT</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1:</td>
<td>Jim</td>
<td>H1</td>
<td>REDWOOD DR</td>
<td>MICHIGAN CITY</td>
<td>MI</td>
</tr>
<tr>
<td>t2:</td>
<td>Tom</td>
<td>H1</td>
<td>REDWOOD DR</td>
<td>WESTVILLE</td>
<td>IN</td>
</tr>
<tr>
<td>t3:</td>
<td>Jeff</td>
<td>H2</td>
<td>BIRCH PARKWAY</td>
<td>WESTVILLE</td>
<td>IN</td>
</tr>
<tr>
<td>t4:</td>
<td>Rick</td>
<td>H2</td>
<td>BIRCH PARKWAY</td>
<td>WESTVILLE</td>
<td>IN</td>
</tr>
<tr>
<td>t5:</td>
<td>Mrk</td>
<td>H1</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
</tr>
<tr>
<td>t6:</td>
<td>Mark</td>
<td>H1</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
</tr>
<tr>
<td>t7:</td>
<td>Cady</td>
<td>H2</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
</tr>
<tr>
<td>t8:</td>
<td>Sindy</td>
<td>H2</td>
<td>SHERDEN RD</td>
<td>FT WAYNE</td>
<td>IN</td>
</tr>
</tbody>
</table>

**Suggested Updates:**
- Replace City "FORT WAYNE" with "Westville" in t5
- Replace Zip 46391 with 46825 in t5

**CFD₁**: \( ZIP \rightarrow CT, STT, \{46391 \mid \text{Westville, IN}\}\)

**CFD₂**: \( STR, CT \rightarrow ZIP, \{\_, \text{FortWayne } \mid \_\}\)
GDR: Group and Rank Repairs

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>SRC</th>
<th>STR</th>
<th>CT</th>
<th>STT</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Jim</td>
<td>H1</td>
<td>REDWOOD DR</td>
<td>MICHIGAN CITY</td>
<td>MI</td>
<td>46360</td>
</tr>
<tr>
<td>t2</td>
<td>Tom</td>
<td>H1</td>
<td>REDWOOD DR</td>
<td>WESTVILLE</td>
<td>IN</td>
<td>46360</td>
</tr>
<tr>
<td>t3</td>
<td>Jeff</td>
<td>H2</td>
<td>BIRCH PARKWAY</td>
<td>WESTVILLE</td>
<td>IN</td>
<td>46360</td>
</tr>
<tr>
<td>t4</td>
<td>Rick</td>
<td>H2</td>
<td>BIRCH PARKWAY</td>
<td>WESTVILLE</td>
<td>IN</td>
<td>46360</td>
</tr>
<tr>
<td>t5</td>
<td>Mrk</td>
<td>H1</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
<td>46391</td>
</tr>
<tr>
<td>t6</td>
<td>Mark</td>
<td>H1</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
<td>46825</td>
</tr>
<tr>
<td>t7</td>
<td>Cady</td>
<td>H2</td>
<td>BELL AVENUE</td>
<td>FORT WAYNE</td>
<td>IN</td>
<td>46825</td>
</tr>
<tr>
<td>t8</td>
<td>Sindy</td>
<td>H2</td>
<td>SHERDEN RD</td>
<td>FT WAYNE</td>
<td>IN</td>
<td>46774</td>
</tr>
</tbody>
</table>

**Contextual grouping for the suggested updates**

Update Group g₁: The city should be “Michigan City” for \{t₂, t₃, t₄\}.
Update Group g₂: The zip should be “46825” for \{t₅, t₈\}.

....
....
....
A Table of Soccer Players

FD: $B \rightarrow C$

- Automatic: Produce heuristic repairs
- GDR:
  - Rely on redundancy to detect errors
  - Require heavy human involvement

Proposal: Use external trustworthy information!

- KBs
- Crowd experts
KATARA Workflow

**INPUT**
- Table $T$
- Trusted KB $K$

**KATARA**

**Pattern Discovery**
- **Algorithm**: rank-join
- **Return**: candidate table patterns

**Pattern Validation**
- **Algorithm**: entropy based scheduling
- **Return**: a table pattern

**Data Annotation**
- **Algorithm**: Inverted list based approach
- **Return**: annotated data, new facts, top-k repairs

**OUTPUT**
- Crowd validated KB validated
- Possible repairs
- Table $T'$
- Enriched KB $K'$

**Example Graphs**

- **Before**:
  - $A$: person
  - $B$: country
  - $C$: Capital
  - $D$: football club
  - $E$: language
  - $F$: city

- **After**:
  - $A$: Pirlo
  - $B$: Italy
  - $C$: Madrid
  - $D$: Juve
  - $E$: Italian
  - $F$: Flero

**Enriched Knowledge Base**

- $K'$: Updated KB with validated data and repairs.
Pattern Discovery: Generate Candidates

Generate candidate types for every column:

\[ Q_{\text{types}} \]
\[
\text{select } ?c_i \\
\text{where } \{ ?x_i \text{ rdfs:label } t[A_i], \\
?x_i \text{ rdfs:type/rdfs:subClassOf* } ?c_i \}
\]

Generate candidate relationships for every column pair:

\[ Q^1_{\text{rels}} \]
\[
\text{select } ?P_{ij} \\
\text{where } \{ ?x_i \text{ rdfs:label } t[A_i], ?x_j \text{ rdfs:label } t[A_j], \\
?x_i \text{ ?P}_{ij}/\text{rdfs:subPropertyOf* } ?x_j \}
\]

\[ Q^2_{\text{rels}} \]
\[
\text{select } ?P_{ij} \\
\text{where } \{ ?x_i \text{ rdfs:label } t[A_i], \\
?x_i \text{ ?P}_{ij}/\text{rdfs:subPropertyOf* } t[A_j] \}
\]

**Type (B)**
- economy
- country
- location
- state
- ...

**Type (C)**
- City
- Capital
- whole
- artifact
- Person
- ...

**Relationship (B, C)**
- locatedIn
- hasCapital
Crowd Pattern Validation

Q₁: What is the most accurate type of the highlighted column?
(A, B, C, D, E, F, ...)
(Rossi, Italy, Rome, Verona, Italian, Proto, ...)
(Pirlo, Italy, Madrid, Juve, Italian, Flero, ...)
- country
- economy
- state

Q₂: What is the most accurate relationship?
(A, B, C, D, E, F, ...)
(Rossi, Italy, Rome, Verona, Italian, Proto, ...)
(Pirlo, Italy, Madrid, Juve, Italian, Flero, ...)
- B hasCapital C
- C locatedIn B
Data Annotation

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>Rossi</td>
<td>Italy</td>
<td>Rome</td>
<td>Verona</td>
<td>Italian</td>
<td>Proto</td>
<td>1.78</td>
</tr>
<tr>
<td>t₂</td>
<td>Klate</td>
<td>South Africa</td>
<td>Pretoria</td>
<td>Pirates</td>
<td>Afrikaans</td>
<td>P. Eliz.</td>
<td>1.69</td>
</tr>
<tr>
<td>t₃</td>
<td>Pirlo</td>
<td>Italy</td>
<td>Madrid</td>
<td>Juve</td>
<td>Italian</td>
<td>Flero</td>
<td>1.77</td>
</tr>
</tbody>
</table>

\[ t₁: \text{validated by KB} \]

\[ t₂: \text{validated by KB \& crowd} \]

\[ t₃: \text{Erroneous tuple} \]
## Data Repairing

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>Rossi</td>
<td>Italy</td>
<td>Rome</td>
<td>Verona</td>
<td>Italian</td>
<td>Proto</td>
<td>1.78</td>
</tr>
<tr>
<td>t₂</td>
<td>Klate</td>
<td>South Africa</td>
<td>Pretoria</td>
<td>Pirates</td>
<td>Afrikaans</td>
<td>P. Eliz.</td>
<td>1.69</td>
</tr>
<tr>
<td>t₃</td>
<td>Pirlo</td>
<td>Italy</td>
<td>Madrid</td>
<td>Juve</td>
<td>Italian</td>
<td>Flero</td>
<td>1.77</td>
</tr>
</tbody>
</table>

**G₁ has cost 1**

(a) Possible repair $G₁$

**G₂ has cost 5**

(b) Possible repair $G₂$
Data Repairing Techniques Taxonomy

Data Repairing Techniques

- Repair target (What to repair?)
  - Data
  - Rules
  - Both

- Automation (How to repair?)
  - Automatic
  - Human guided

- Update model (Where to repair?)
  - In place
  - Model based

Piece-meal
Holistic
One-Shot Data Cleaning

- Generate a single “trustworthy” instance

![Diagram of data cleaning process]

Queries → RDBMS → Deterministic Database → Single Clean Instance → Deterministic Data Cleaning → Unclean Database

Result

University of Waterloo
Model Based Approach

- Generate multiple possible clean instances

![Diagram showing the process of model-based approach for data cleaning.]

1. Unclean Database
2. Uncertain Database
3. Probabilistic Results
4. Queries
5. Uncertainty and Cleaning-aware RDBMS
6. Probabilistic Data Cleaning
7. Multiple Possible Clean Instances
8. Unclean Database
Model Based Approach Challenges

1. The space of all possible repairs is huge

2. How to efficiently generate, store and query the possible repairs
Two Example Model Based Approaches

- **Duplicate Detection** [Beskales et al, VLDB 2009]
  - Spaces of Possible Repairs
  - Generating and Storing Possible Repairs
  - Query Possible Repairs

- **Violations of Functional Dependencies** [Beskales et al, VLDB 2010]
  - Spaces of Possible Repairs
  - Sampling from a Meaningful Space of Repairs
Two Example Model Based Approaches

- **Duplicate Detection** [Beskales et al, VLDB 2009]
  - Spaces of Possible Repairs
  - Generating and Storing Possible Repairs
  - Query Possible Repairs

- **Violations of Functional Dependencies** [Beskales et al, VLDB 2010]
  - Spaces of Possible Repairs
  - Sampling from a Meaningful Space of Repairs
# Typical Data Deduplication

## Unclean Relation

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ZIP</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Green</td>
<td>51519</td>
<td>30k</td>
</tr>
<tr>
<td>P2</td>
<td>Green</td>
<td>51518</td>
<td>32k</td>
</tr>
<tr>
<td>P3</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P4</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P5</td>
<td>Gree</td>
<td>51519</td>
<td>55k</td>
</tr>
<tr>
<td>P6</td>
<td>Chuck</td>
<td>51519</td>
<td>30k</td>
</tr>
</tbody>
</table>

## Clean Relation

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ZIP</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Green</td>
<td>51519</td>
<td>39k</td>
</tr>
<tr>
<td>C2</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>C3</td>
<td>Chuck</td>
<td>51519</td>
<td>30k</td>
</tr>
</tbody>
</table>

## Compute Pair-wise Similarity

```
P1 0.9 P2
P5 0.3 P3 1.0 P4
P6
```

## Cluster Similar Records

```
P1 P2
P3 P4 P5
P6
```

## Merge Clusters

```
C1
C2
C3
```

---

102
A **possible repair** is a clustering (partitioning) of the input tuples.

### Person

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>ZIP</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Green</td>
<td>51519</td>
<td>30k</td>
</tr>
<tr>
<td>P2</td>
<td>Green</td>
<td>51518</td>
<td>32k</td>
</tr>
<tr>
<td>P3</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P4</td>
<td>Peter</td>
<td>30528</td>
<td>40k</td>
</tr>
<tr>
<td>P5</td>
<td>Gree</td>
<td>51519</td>
<td>55k</td>
</tr>
<tr>
<td>P6</td>
<td>Chuck</td>
<td>51519</td>
<td>30k</td>
</tr>
</tbody>
</table>

### Uncertain Clustering

- $X_1$: \{P1\}, \{P2\}, \{P3, P4\}, \{P5\}, \{P6\}
- $X_2$: \{P1, P2\}, \{P3, P4\}, \{P5\}, \{P6\}
- $X_3$: \{P1, P2, P5\}, \{P3, P4\}, \{P6\}

### Possible Repairs

- $X_1$: \{P1\}, \{P2\}, \{P3, P4\}, \{P5\}, \{P6\}
- $X_2$: \{P1, P2\}, \{P3, P4\}, \{P5\}, \{P6\}
- $X_3$: \{P1, P2, P5\}, \{P3, P4\}, \{P6\}
Generating Possible Repairs

Distance Threshold ($\tau$) →

Pair-wise Distance

Possible Thresholds
$[\tau_l, \tau_u]$
The probability of a repair is equal to the probability of the parameter range that generates such repair.
Storing Possible Repairs

- **U-Clean Relations**
  - Each cluster is stored once
  - We keep the “lineage” of each cluster

<table>
<thead>
<tr>
<th>Clustering 1</th>
<th>Clustering 2</th>
<th>Clustering 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>{P1}</td>
<td>{P1, P2}</td>
<td>{P1, P2, P5}</td>
</tr>
<tr>
<td>{P2}</td>
<td>{P3, P4}</td>
<td>{P3, P4}</td>
</tr>
<tr>
<td>{P3, P4}</td>
<td>{P5}</td>
<td>{P6}</td>
</tr>
<tr>
<td>{P5}</td>
<td>{P6}</td>
<td></td>
</tr>
<tr>
<td>{P6}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 ≤ τ < 1  1 ≤ τ < 3  3 ≤ τ < 10

### U-clean Relation Person$^C$

<table>
<thead>
<tr>
<th>ID</th>
<th>Income</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>31k</td>
<td>{P1, P2}</td>
<td>[1,3)</td>
</tr>
<tr>
<td>CP2</td>
<td>40k</td>
<td>{P3, P4}</td>
<td>[0,10)</td>
</tr>
<tr>
<td>CP3</td>
<td>55k</td>
<td>{P5}</td>
<td>[0,3)</td>
</tr>
<tr>
<td>CP4</td>
<td>30k</td>
<td>{P6}</td>
<td>[0,10)</td>
</tr>
<tr>
<td>CP5</td>
<td>39k</td>
<td>{P1, P2, P5}</td>
<td>[3,10)</td>
</tr>
<tr>
<td>CP6</td>
<td>30k</td>
<td>{P1}</td>
<td>[0,1)</td>
</tr>
<tr>
<td>CP7</td>
<td>32k</td>
<td>{P2}</td>
<td>[0,1)</td>
</tr>
</tbody>
</table>
Example: Projection Query

**Person**

<table>
<thead>
<tr>
<th>ID</th>
<th>...</th>
<th>Income</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td></td>
<td>31k</td>
<td>{P1,P2}</td>
<td>[1,3)</td>
</tr>
<tr>
<td>CP2</td>
<td></td>
<td>40k</td>
<td>{P3,P4}</td>
<td>[0,1)</td>
</tr>
<tr>
<td>CP3</td>
<td></td>
<td>55k</td>
<td>{P5}</td>
<td>[0,3)</td>
</tr>
<tr>
<td>CP4</td>
<td></td>
<td>30k</td>
<td>{P6}</td>
<td>[3,10]</td>
</tr>
<tr>
<td>CP5</td>
<td></td>
<td>40k</td>
<td>{P1,P2,P5}</td>
<td>[3,10]</td>
</tr>
<tr>
<td>CP6</td>
<td></td>
<td>30k</td>
<td>{P1}</td>
<td>[0,1)</td>
</tr>
<tr>
<td>CP7</td>
<td></td>
<td>32k</td>
<td>{P2}</td>
<td>[0,1)</td>
</tr>
</tbody>
</table>

**SELECT DISTINCT** Income
**FROM** Person

<table>
<thead>
<tr>
<th>Income</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>30k</td>
<td>{P1} v {P6}</td>
<td>[0,1) v [3,10]</td>
</tr>
<tr>
<td>31k</td>
<td>{P1,P2}</td>
<td>[1,3)</td>
</tr>
<tr>
<td>32k</td>
<td>{P2}</td>
<td>[0,1)</td>
</tr>
<tr>
<td>40k</td>
<td>{P3,P4} v {P1,P2,P5}</td>
<td>[0,1) v [3,10]</td>
</tr>
<tr>
<td>55k</td>
<td>{P5}</td>
<td>[0,3)</td>
</tr>
</tbody>
</table>
Big Data Cleaning Challenges

- **Volume**
  - Distributed Data Cleaning
  - Sample Clean

- **Velocity**
  - Incremental Data Cleaning

- **Variety**
  - Graph/JSON/RDF
  - Text
Distributed Data Deduplication [Chu et al, VLDB 2016]

- Data deduplication in data lake setting
  - A shared-nothing environment
  - Need to compare every tuple pair
- The goal is to minimizing
  - Largest communication cost
  - Largest computation cost
Conclusion and References

Error Detection

What (IC Languages and Discovery)

Conclusion and References

Error Detection

How (Human involvement)

Where (Analytics Layer)
Conclusion and References

Error Repairing

What (Data or Data & Rule)


How (Human Involvement)

Conclusion and References

- **Error Repairing**
  - **Where (Model-based)**

- **Taxonomy**