Data Quality Tools

Concepts and practical lessons from a vast operational environment

13/03/2018 @ ULB
Before we start…

• Although technical matter, hand in hand with application area specialists (« business » in the uncommercial sense)

• Each time iterations with application area specialists are crucial, logo on **upper right corner**:
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

1. Preventive and curative approaches: organization
2. The curative approach
3. DQ@Smals
4. Fitness for use
5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

- 1. Preventive and curative approaches: organization
- 2. The curative approach
- 3. DQ@Smals
- 4. Fitness for use
- 5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
1. Preventive and curative approaches: organization

- Business Logic
  - definitions (anomaly, ...),
  - business rules,
  - correction or standardization,
  - algorithms,
  - Δ Business Processes
  - Δ Applications,
  - Δ DB constraints, ...

- Applications

- Business

- DB A, B, C

- Documents

- Enrich (backtracking)

- Anomaly history

- Batch DQ project

- Validation

- Batch DQ project

- Result

- Document

- DQ Tools

- Online DQ

- DQTools, J2EE, SaaS, ...

- DQTeam

- Δ Business Processes

- Δ Applications,

- Δ DB constraints, ...

- Source: Dries Van Dromme
Contents

Introduction: DQ fundamentals

• 1. Preventive and curative approaches : organization
• 2. The curative approach
• 3. DQ@Smals
• 4. Fitness for use
• 5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
2. The curative approach

- Profiling: what’s happening into our data and metadata (if available)?
  - Investigate DQ and analyze (un)known anomalies
  - Measure when possible
2. The curative approach

- Profiling: what’s happening into our data and metadata (if available) ?

- Standardization: build and apply standards to our data
  - Formal or fundamental
  - Enriching with Knowledge DBs and/or Business Rules
2. The curative approach

• Profiling: what’s happening into our data and metadata (if available)?

• Standardization: build and apply standards to our data

• Matching: which records belong together?
  – Detect duplicates and inconsistencies: variable fuzziness
  – Deduplicate
  – Chose or build a « golden record »
  – \( \text{Performance} \)
2. The curative approach

- Profiling: what’s happening into our data and metadata (if available)?
- Standardization: build and apply standards to our data
- Matching: which records belong together?
2. The curative approach

- Profiling: what’s happening into our data and metadata (if available)?
- Standardization: build and apply standards to our data
- Matching: which records belong together?
- Dedicated tools, specific to one area or « all-in-one »
2. The curative approach: DQ tools
2. The curative approach: DQ tools

- Since 1980’s, initial core business: names and addresses
  - Ever-present issue
  - Ubiquitous: companies, client data, service providers, B2B, public administrations…
2. The curative approach: DQ tools

• Since 1980’s, initial core business: names and addresses
  – Ever-present issue
  – Ubiquitous: companies, client data, service providers, B2B, public administrations…

• Complex and changing standards
  – Knowledge bases built over time
  – Taking international context into account
  – Regular updates
2. The curative approach: DQ tools

- Since 1980’s, initial core business: names and addresses
  - Ever-present issue
  - Ubiquitous: companies, client data, service providers, B2B, public administrations…

- Complex and changing standards
  - Knowledge bases built over time
  - Taking international context into account
  - Regular updates

- Today, extended to all alphanum strings
  - Thousands of mature algorithms
  - Decades of optimizations
2. The curative approach: DQ tools

• Since 1980’s, initial core business: names and addresses
  – Ever-present issue
  – Ubiquitous: companies, client data, service providers, B2B, public administrations…

• Complex and changing standards
  – Knowledge bases built over time
  – Taking international context into account
  – Regular updates

• Today, extended to all alphanum strings
  – Thousands of mature algorithms
  – Decades of optimizations

• Adapted to DQ work nature
  – Iterations and drill-down
  – Constant business involvement (critical!)
  – Less time wasted in development: more efficient resource distribution
Contents

Introduction: DQ fundamentals

• 1. Preventive and curative approaches: organization
• 2. Technical approaches: Profiling, Standardization, Matching
• 3. DQ@Smals
• 4. Fitness for use
• 5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
3. DQ@Smals: projects (1)

- **2010**
  - **FOD Economie**: KBO Adreskwaliteit
  - **RSZ**: 30bis werfmeldingen – dubbeldetectie
  - **eHealth-platform**: inconsistency management (multiple DBs)
  - **FAGG**: Datamigratie Kadaster Officina
- **2011**
  - **FAGG**: Datamigratie Kadaster Officina
  - **SIGeDIS**: 2de pensioenpijler - preload KBO
  - **VAZG**: Datakwaliteit Vaccinnet
- **2012**
  - **eHealthPlatform** – opbouw van Validated Authentic Sources (VAS)
  - **RSZ** sociale-fraudebestrijding
- **2013**
  - **eHealthPlatform** – VAS (continued)
  - **RSZ** sociale-fraudebestrijding (adresmatching)
  - **RSZ** fuzzy matching Limosa-kadaster (foreign employees in Belgium)
3. DQ@Smals: projects (2)

- **2014-2015-2016**
  - 2015-2016: **RSZ** EDE (Dossier Electronique de l’Employeur)
  - **eHP** – VAS (continued)
  - **RSZ sociale-fraudebestrijding**: matching entities from various authentic sources (continued)
- **2017-2019**
  - **KBO – Repertorium**: comparative profiling
  - **RSZ** – Directie Risicobeheer: register matching enterprises from various authentic sources
  - **eHP** – VAS (continued)
  - **Fédération Wallonie-Bruxelles**: data quality management and integration from various financial and accounting databases
  - **Migration OSSOM – ONSS**: inconsistency detection and data migration
  - **FoLeEn**: repertory to identify foreign companies
Contents

Introduction: DQ fundamentals

• 1. Preventive and curative approaches: organization
• 2. The Technical approaches: Profiling, Standardization, Matching
• 3. DQ@Smals
• 4. Fitness for use
• 5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
4. Fitness for use

Good quality data

Bad data

Your job
4. Fitness for use: a typical situation

→ The same entity appears as 100s of different enterprises
4. Fitness for use: a typical situation

→ The same entity appears as 100s of different enterprises

« We need to view data as enterprises, not rows. »
4. Fitness for use: a typical situation

→ The same entity appears as 100s of different enterprises

« We need to view data as enterprises, not rows. »
But how do you do that?...
4. Fitness for use

• **Frequent use cases**
  – Creating a new repertory from external sources
  – Integration of IT systems and DBs
  – Fusions and migrations between administrations
  – Predictive analytics and statistical modeling
  – Etc.

• **Important financial impact** in Belgium (social security)
  – € 65 billion / year

• **…and elsewhere**
  – « $3,1 Trillions/year in the US, which is about 20 percent of the Gross Domestic Product. » - Redman T., *Getting in front on data*, Technics Publications, Denville (New Jersey, USA), 2016
## Introduction: DQ fundamentals

- 1. Preventive and curative approaches: organization
- 2. The Technical approaches: Profiling, Standardization, Matching
- 3. DQ@Smals
- 4. Fitness for use
- **5. How DQ tools work**

### Part 1: Data Profiling

### Part 2: Parsing, Standardization & Address enrichment (PSA)

### Part 3: Data matching and Window keys (performance)

### Conclusion & questions
4. How DQ tools work: locally
4. How DQ tools work: locally

NB: some tools do offer DB connectivity, mostly read-only, in addition to the common file import/export capabilities (e.g. OpenRefine)
4. How DQ tools work: locally

**E.g.** : OpenRefine, Trifacta Wrangler, Talend Data Preparation / Open Studio, etc.

**NB** : some tools do offer DB connectivity, mostly read-only, in addition to the common file import/export capabilities (e.g. OpenRefine)
4. How DQ tools work: client-server / multitier
4. How DQ tools work: client-server / multitier
4. How DQ tools work: client-server / multitier
4. How DQ tools work: client-server / multitier

- DQ TOOL SERVER
  - Client storage
  - Batch processing storage
  - ODBC
  - E.g.: Trillium, Talend Studio, Informatica, etc.

- Client - server / multitier

- Open or proprietary file format

- DB SERVER
  - DB
Contents

Introduction: DQ fundamentals

• 1. Preventive and curative approaches: organization
• 2. The Technical approaches: Profiling, Standardization, Matching
• 3. DQ@Smals
• 4. Fitness for use
• 5. How DQ tools work

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Introduction: DQ fundamentals

Part 1: Data Profiling

- 1. Main concepts
- 2. Drill-down into entities and attributes
- 3. Relations into the data
- 4. Business rules
- 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling
- 1. Main concepts
- 2. Drill-down into entities and attributes
- 3. Relations into the data
- 4. Business rules
- 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
1. Profiling: Main concepts

1. What is it?

2. Profiling with a DQ tool
1.1. Main concepts: what is profiling?

« The use of analytical techniques to discover the (...) structure, content and quality of data. »

1.1. Main concepts: what is profiling?

**Metadata**
(inaccurate, incomplete)

**Project Data Profiling**

**Metadata**
(corrected)

**Real data**
(complete, quality=?)

**Facts about data**
(quality = !)

**Data Quality Issues**
- lack of standardisation
- schema not respected
- rules not respected

**Large % of effort**
- getting access
- getting the metadata
- getting the data
- getting the data in the right (normal) form

**DQTools**

**Business people**
DQ Analyst team

DB-schema,
Constraints,
Business Rules,
Documentation, ...

Source: Dries Van Dromme
1.2. Profiling with a Data Quality Tool

- Automatic analysis upon data loading or « you seek it »

- Concretely, data about your data
  - « metadata »
  - Quantitative and qualitative
  - Fundamental and formal
  - != BI ; main focus = DQ
1.2. Profiling with a Data Quality Tool

• At the dataset level
  – Entity / table level profiling
1.2. Profiling with a Data Quality Tool

• At the dataset level
  – Entity / table level profiling

• Field per field
  – Attribute / column level profiling
1.2. Profiling with a Data Quality Tool

- At the dataset level
  - Entity / table level profiling
- Field per field
  - Attribute / column level profiling
- Relations into the data
  - Primary **Keys** analysis
  - Functional **Dependency analysis**
  - Referential constraints with **Join analysis**
1.2. Profiling with a Data Quality Tool

- At the dataset level
  - Entity / table level profiling
- Field per field
  - Attribute / column level profiling
- Relations into the data
  - Primary Keys analysis
  - Functional Dependency analysis
  - Referential constraints with Join analysis
- Consistency and business logic
  - Business rules analysis
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

- 1. Main concepts
- 2. Drill-down into entities and attributes
- 3. Relations into the data
- 4. Business rules
- 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
2. Drill-down into entities and attributes

- Browsing through data at various levels
  - Zomming on a specific observation: drill-down
  - Going back one layer: drill-up

- Full path: from entity/table metadata, through intermediary measures, to data
Drill down into entities and attributes.
2. Drill-down into entities and attributes

1. Entity / table level profiling

2. Attribute / column level profiling
2.1. Entity / table level profiling

• Summary about entity business rules

<table>
<thead>
<tr>
<th>Business Rules</th>
<th>1</th>
<th>The number of business rules defined for this entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Business Rules</td>
<td>1</td>
<td>The number of enabled business rules</td>
</tr>
<tr>
<td>Passing Business Rules</td>
<td>0</td>
<td>The number of passing business rules</td>
</tr>
<tr>
<td>Failing Business Rules</td>
<td>1</td>
<td>The number of failing business rules</td>
</tr>
</tbody>
</table>
2.1. Entity / table level profiling

- Summary about entity business rules

<table>
<thead>
<tr>
<th>Business Rules</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Business Rules</td>
<td>1</td>
</tr>
<tr>
<td>Passing Business Rules</td>
<td>0</td>
</tr>
<tr>
<td>Failing Business Rules</td>
<td>1</td>
</tr>
</tbody>
</table>

**Drill-down**

If country = Belgium then Postcode ~ NNNN
2.1. Entity / table level profiling

• Summary about entity business rules

<table>
<thead>
<tr>
<th>Business Rules</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Business Rules</td>
<td>1</td>
</tr>
<tr>
<td>Passing Business Rules</td>
<td>0</td>
</tr>
<tr>
<td>Failing Business Rules</td>
<td>1</td>
</tr>
</tbody>
</table>

- If country = Belgium then Postcode ~ NNNN

Drill-down to failing rows

Drill-down
2.2. Attribute / column level profiling

- Values counts and distinct measures

Drill-down, sort on length descending
2.2. Attribute / column level profiling

- **Datatype inference**

<table>
<thead>
<tr>
<th>Name</th>
<th>Strings</th>
<th>Strings Dist %</th>
<th>Decimals</th>
<th>Dec Dist %</th>
<th>Integers</th>
<th>Integer Dist %</th>
<th>Inferred Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce</td>
<td>72948</td>
<td>43.579</td>
<td>12</td>
<td>0.002</td>
<td>25630</td>
<td>55.441</td>
<td>Integer</td>
</tr>
<tr>
<td>O Tva Fe</td>
<td>80253</td>
<td>20.156</td>
<td>9</td>
<td>0.002</td>
<td>12400</td>
<td>2.537</td>
<td>String</td>
</tr>
<tr>
<td>T Busnbr Srce</td>
<td>3104</td>
<td>3.037</td>
<td>47</td>
<td>0.017</td>
<td>2043</td>
<td>8.317</td>
<td>Integer</td>
</tr>
<tr>
<td>T Housenbr Srce</td>
<td>4154</td>
<td>3.965</td>
<td>9</td>
<td>0.002</td>
<td>4097</td>
<td>65.379</td>
<td>Integer</td>
</tr>
<tr>
<td>T Street Srce</td>
<td>251504</td>
<td>99.646</td>
<td>1</td>
<td>0.001</td>
<td>416</td>
<td>0.346</td>
<td>String</td>
</tr>
</tbody>
</table>
2.2. Attribute / column level profiling

• Datatype inference

<table>
<thead>
<tr>
<th>Name</th>
<th>Strings</th>
<th>Strings Dist %</th>
<th>Decimals</th>
<th>Dec Dist %</th>
<th>Integers</th>
<th>Integer Dist %</th>
<th>Inferred Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce</td>
<td>72948</td>
<td>43.579</td>
<td>12</td>
<td>0.002</td>
<td>25630</td>
<td>55.441</td>
<td>Integer</td>
</tr>
<tr>
<td>O Tva Fe</td>
<td>80253</td>
<td>20.156</td>
<td>9</td>
<td>0.002</td>
<td>12400</td>
<td>2.537</td>
<td>String</td>
</tr>
<tr>
<td>T Busnhr Srce</td>
<td>3104</td>
<td>3.037</td>
<td>47</td>
<td>0.017</td>
<td>2043</td>
<td>8.317</td>
<td>Integer</td>
</tr>
<tr>
<td>T Housenbr Srce</td>
<td>4154</td>
<td>3.965</td>
<td>9</td>
<td>0.002</td>
<td>4097</td>
<td>65.379</td>
<td>Integer</td>
</tr>
<tr>
<td>T Street Srce</td>
<td>251504</td>
<td>99.646</td>
<td>1</td>
<td>0.001</td>
<td>416</td>
<td>0.346</td>
<td>String</td>
</tr>
</tbody>
</table>
2.2. Attribute / column level profiling

- Datatype inference

<table>
<thead>
<tr>
<th>Name</th>
<th>Strings</th>
<th>Strings Dist %</th>
<th>Decimals</th>
<th>Dec Dist %</th>
<th>Integers</th>
<th>Integer Dist %</th>
<th>Inferred</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce</td>
<td>72948</td>
<td>43.579</td>
<td>12</td>
<td>0.002</td>
<td>25630</td>
<td>55.441</td>
<td>Integer</td>
<td>Integer</td>
</tr>
<tr>
<td>O Tva Fe</td>
<td>80253</td>
<td>20.156</td>
<td>9</td>
<td>0.002</td>
<td>12400</td>
<td>2.537</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>T Busnbr Srce</td>
<td>3104</td>
<td>3.037</td>
<td>47</td>
<td>0.017</td>
<td>2043</td>
<td>8.317</td>
<td>Integer</td>
<td>Integer</td>
</tr>
<tr>
<td>T Housenbr Srce</td>
<td>4154</td>
<td>3.965</td>
<td>9</td>
<td>0.002</td>
<td>4097</td>
<td>65.379</td>
<td>Integer</td>
<td>Integer</td>
</tr>
<tr>
<td>T Street Srce</td>
<td>251504</td>
<td>99.646</td>
<td>1</td>
<td>0.001</td>
<td>416</td>
<td>0.346</td>
<td>String</td>
<td>String</td>
</tr>
</tbody>
</table>

Drill-down

<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
<th>Dist %</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am not payer</td>
<td>372</td>
<td>0.064</td>
</tr>
<tr>
<td>I am not payer VAT</td>
<td>259</td>
<td>0.044</td>
</tr>
<tr>
<td>iam not pay vat</td>
<td>45</td>
<td>0.008</td>
</tr>
<tr>
<td>I am not responsible to get VAT number.</td>
<td>10</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

• 1. Main concepts
• 2. Drill-down into entities and attributes
• 3. Relations into the data
• 4. Business rules
• 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
3. Relations into the data

1. Keys analysis
   Discover primary key candidates or check their validity

2. Dependencies analysis
   Discover or check functional dependencies

3. Joins analysis
   Check referential integrity and foreign keys
3.1. Keys analysis

• Looking for highly unique values

• In specific tools: while loading data, « keys discovery »
  – Analyzing a sample of rows (e.g. 10 000 rows)
  – Looking for atomic or composite (e.g. 2 attributes) keys
  – Keeping candidates that are above a certain uniqueness threshold
    • E.g. >= 98% unique
### 3.1. Keys analysis

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>AL32UTF8</td>
<td>Encoding used when loading the data</td>
</tr>
<tr>
<td>Permanent Keys</td>
<td>0</td>
<td>The number of permanent keys for this entity</td>
</tr>
<tr>
<td>Discovered Keys</td>
<td>5</td>
<td>The number of discovered keys for this entity</td>
</tr>
<tr>
<td>Permanent Deps</td>
<td>2</td>
<td>The number of permanent dependencies identified for</td>
</tr>
</tbody>
</table>
3.1. Keys analysis

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>AL32UTF8</td>
<td>Encoding used when loading the data</td>
</tr>
<tr>
<td>Permanent Keys</td>
<td>0</td>
<td>The number of permanent keys for this entity</td>
</tr>
<tr>
<td>Discovered Keys</td>
<td>5</td>
<td>The number of discovered keys for this entity</td>
</tr>
<tr>
<td>Permanent Deps</td>
<td>3</td>
<td>The number of permanent dependencies identified for this entity</td>
</tr>
</tbody>
</table>

**Entity = lim_20171029_old_ora_connector(64)**

<table>
<thead>
<tr>
<th>Lh Attrs</th>
<th>Status</th>
<th>Verified</th>
<th>Ref</th>
<th>Quality %</th>
<th>Keys</th>
<th>Duplicate Keys</th>
<th>Duplicate Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Source Key Unique</td>
<td>Discovered</td>
<td>No</td>
<td>8</td>
<td>100.000</td>
<td>10000</td>
<td>135</td>
<td>288</td>
</tr>
<tr>
<td>O Md5,T City Srce</td>
<td>Discovered</td>
<td>No</td>
<td>8</td>
<td>98.470</td>
<td>9712</td>
<td>164</td>
<td>352</td>
</tr>
<tr>
<td>O Md5,T Name Srce</td>
<td>Discovered</td>
<td>No</td>
<td>8</td>
<td>98.120</td>
<td>9648</td>
<td>164</td>
<td>352</td>
</tr>
<tr>
<td>O Md5 Wide,T City Srce</td>
<td>Discovered</td>
<td>No</td>
<td>8</td>
<td>98.470</td>
<td>9712</td>
<td>135</td>
<td>288</td>
</tr>
<tr>
<td>O Md5 Wide,T Name Srce</td>
<td>Discovered</td>
<td>No</td>
<td>8</td>
<td>98.120</td>
<td>9648</td>
<td>164</td>
<td>352</td>
</tr>
</tbody>
</table>
3.1. Keys analysis

Is this duplication normal? Application area specialists need to investigate.
3.2. Functional dependencies analysis

- One or more columns determine the value of others
  - Left1 + Left2 $\rightarrow$ Right
  - Street + Postcode + City $\rightarrow$ Country
3.2. Functional dependencies analysis

• One or more columns determine the value of others
  – Left1 + Left2 → Right
  – Street + Postcode + City → Country

• Checking if an expected dependency is met
  – Doubts if unnormalized data model
  – Analytic datasets (denormalized on purpose)
  – Pure data-level issues

• Discover unexpected dependencies
  – Issues in the data model
3.2. Functional dependencies analysis

- One or more columns determine the value of others
  - Left1 + Left2 → Right
  - Street + Postcode + City → Country

- Checking if an expected dependency is met
  - Doubts if unnormalized data model
  - Analytic datasets (denormalized on purpose)
  - Pure data-level issues

- Discover unexpected dependencies
  - Issues in the data model

- Drill down to conflicting values and rows
3.2. Functional dependencies analysis: results of a specific analysis

<table>
<thead>
<tr>
<th>Lh Attrs</th>
<th>Rh Attr</th>
<th>Quality %</th>
<th>Conflicting LH Values</th>
<th>Conflicting Rows</th>
<th>Verified Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce, T City Srce</td>
<td>C Nis Ctry Srce</td>
<td>99.858</td>
<td>1402</td>
<td>3393</td>
<td>2018/03/19 14:44:07</td>
</tr>
</tbody>
</table>
3.2. Functional dependencies analysis: results of a specific analysis

<table>
<thead>
<tr>
<th></th>
<th>Lh Attr</th>
<th>Rh Attr</th>
<th>Quality %</th>
<th>Conflicting LH Values</th>
<th>Conflicting Rows</th>
<th>Verified Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce, T City Srce</td>
<td>C Nis Ctry Srce</td>
<td>99.858</td>
<td>1402</td>
<td>3393</td>
<td></td>
<td>2018/03/19 14:44:07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58636</td>
<td>ISERLOHN</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>58636</td>
<td>ISERLOHN</td>
<td>134</td>
</tr>
<tr>
<td>7421</td>
<td>3620</td>
<td>LANAKEN</td>
<td>150</td>
</tr>
<tr>
<td>7</td>
<td>3620</td>
<td>LANAKEN</td>
<td>999</td>
</tr>
<tr>
<td>1</td>
<td>3620</td>
<td>LANAKEN</td>
<td>129</td>
</tr>
<tr>
<td>1</td>
<td>VG 1110</td>
<td>TORTOLA</td>
<td>486</td>
</tr>
<tr>
<td>1</td>
<td>VG 1110</td>
<td>TORTOLA</td>
<td>112</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>COLOGNE</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>COLOGNE</td>
<td>113</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>173</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>134</td>
</tr>
<tr>
<td>29</td>
<td>5001</td>
<td>BELGRADE</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>5001</td>
<td>BELGRADE</td>
<td>999</td>
</tr>
</tbody>
</table>
3.2. Functional dependencies analysis: results of a specific analysis

<table>
<thead>
<tr>
<th>Lh Attrs</th>
<th>Rh Attrs</th>
<th>Quality %</th>
<th>Conflicting LH Values</th>
<th>Conflicting Rows</th>
<th>Verified Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Zipcode Srce, T City Srce</td>
<td>C Nis Ctry Srce</td>
<td>99.858</td>
<td>1402</td>
<td>3393</td>
<td>2018/03/19 14:44:07</td>
</tr>
</tbody>
</table>

- **Frequency**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58636</td>
<td>ISERLOHN</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>58636</td>
<td>ISERLOHN</td>
<td>134</td>
</tr>
</tbody>
</table>

- **7421**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>3620</td>
<td>LANAKEN</td>
<td>150</td>
</tr>
<tr>
<td>1</td>
<td>3620</td>
<td>LANAKEN</td>
<td>999</td>
</tr>
<tr>
<td></td>
<td>3620</td>
<td>LANAKEN</td>
<td>129</td>
</tr>
</tbody>
</table>

- **1**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VG 1110</td>
<td>TORTOLA</td>
<td>486</td>
</tr>
<tr>
<td>1</td>
<td>VG 1110</td>
<td>TORTOLA</td>
<td>112</td>
</tr>
</tbody>
</table>

- **1**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50858</td>
<td>COLOGNE</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>COLOGNE</td>
<td>113</td>
</tr>
</tbody>
</table>

- **1**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>103</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>173</td>
</tr>
<tr>
<td>1</td>
<td>50858</td>
<td>KOLN</td>
<td>134</td>
</tr>
</tbody>
</table>

- **29**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>C Nis Ctry Srce</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>5001</td>
<td>BELGRADE</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>5001</td>
<td>BELGRADE</td>
<td>999</td>
</tr>
</tbody>
</table>
3.3. Joins analysis: referential integrity

- Join analysis between two or more entities

- Basic principle: Left x Right
  - Metadata on each side
  - Metadata on the intersection
  - Drill-down

- Possible to join on a processed column
  - join(Col)
  - join(ucase(Col))
3.3. Joins analysis: referential integrity

- Source 1: Authentieke bron(44)
  key: « Ind_nr »
- Source 2: Source_secondaire(60)
  fkey: « Identificatienummer »

→ Join Analysis Source 1 x Source 2
  86 values not found in source 1
  (in 669 records (so there are doubles))
3.3. Joins analysis: referential integrity
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

• 1. Main concepts
• 2. Drill-down into entities and attributes
• 3. Relations into the data
• 4. Business rules
• 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
4. Business rules (BRs)

- 1. Formalizing business constraints
- 2. BR threshold
- 3. Applying BRs
- 4. BRs library
4.1. Formalizing constraints

• Pinpointing business constraints
  ➔ « Postcodes should not contain values other than alphanumeric characters, dashes and spaces. »
4.1. Formalizing constraints

• Pinpointing business constraints
  → « Postcodes should not contain values other than alphanumeric characters, dashes and spaces. »

• Formalizing them:
  → \texttt{REGEXP("[^a-zA-Z0-9-\ ]","Postcode") = ""}
4.1. Formalizing constraints

• Pinpointing business constraints
  ➔ « Postcodes should not contain values other than alphanumeric characters, dashes and spaces. »

• Formalizing them:
  ➔ \texttt{REGEXP\left("[^a-zA-Z0-9\-\ ]",'Postcode'\right) = ""}

• Attribute-specific or inter-attribute
  ➔ \texttt{LENGTH\left(\textquoteleft Name\textquoteright \right) > 3}
  ➔ \texttt{LENGTH\left(\textquoteleft Name\textquoteright \right) > LENGTH\left(\textquoteleft Initials\textquoteright \right)}
4.1. Formalizing constraints

- **Any string**: names, dates, identifiers, etc.
- E.g.: two work sites declared separately must be more than 100m apart

\[
\text{PROXIMITY} \left( \text{`lat}_1`, \text{`lat}_2`, \text{`long}_1`, \text{`long}_2`, \text{"KM.000"} \right) > \text{"0.100"}
\]
4.2. Business rules threshold

- Passing threshold $T$
  - On rows
  - …or on values
4.2. Business rules threshold

- Passing threshold $T$
  - On rows
  - ...or on values

- Example:
  - Rule = 'Name' NOT LIKE "*&*"
  - Threshold = 50%

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>3</td>
<td>AXA</td>
</tr>
<tr>
<td>4</td>
<td>MERRILL LYNCH</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
</tbody>
</table>
4.2. Business rules threshold

- Passing threshold $T$
  - On rows
  - …or on values

- Example:
  - Rule = 'Name' NOT LIKE "*\&*"
  - Threshold = 50%
    - On rows: 33% passing < T → fail
    - On values: 75% passing ≥ T → pass

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>3</td>
<td>AXA</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>4</td>
<td>MERRILL LYNCH</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
<tr>
<td>2</td>
<td>STANDARD &amp; POOR’S</td>
</tr>
</tbody>
</table>
4.3. Applying BRs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Name</th>
<th>Threshold</th>
<th>Result</th>
<th>Passing Fraction</th>
<th>Created By</th>
<th>Date Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Name Srce</td>
<td>Length more than 3</td>
<td>100</td>
<td>failed</td>
<td>99.192</td>
<td>ganha</td>
<td>2017/11/21 11:0..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No HTML patterns</td>
<td>100</td>
<td>failed</td>
<td>99.999</td>
<td>ganha</td>
<td>2017/11/21 13:4..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No special character</td>
<td>100</td>
<td>failed</td>
<td>90.227</td>
<td>ganha</td>
<td>2017/11/21 11:0..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>Not only num</td>
<td>100</td>
<td>failed</td>
<td>99.743</td>
<td>ganha</td>
<td>2017/11/21 17:0..</td>
</tr>
</tbody>
</table>

BRs can run individually or as sets
4.3. Applying BRs

BRs can run individually or as sets

A threshold can be used to allow a margin of tolerance, which is 0% here
4.3. Applying BRs

BRs can run individually or as sets

A threshold can be used to allow a margin of tolerance, which is 0% here

Measured results

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Name</th>
<th>Threshold</th>
<th>Result</th>
<th>Passing Fraction</th>
<th>Created By</th>
<th>Date Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Name Srce</td>
<td>Length more than 3</td>
<td>100</td>
<td>failed</td>
<td>99.192</td>
<td>ganha</td>
<td>2017/11/21 11:0..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No HTML patterns</td>
<td>100</td>
<td>failed</td>
<td>99.999</td>
<td>ganha</td>
<td>2017/11/21 13:4..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No special character</td>
<td>100</td>
<td>failed</td>
<td>90.227</td>
<td>ganha</td>
<td>2017/11/21 11:0..</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>Not only num</td>
<td>100</td>
<td>failed</td>
<td>99.743</td>
<td>ganha</td>
<td>2017/11/21 17:0..</td>
</tr>
</tbody>
</table>
### 4.3. Applying BRs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Name</th>
<th>Threshold</th>
<th>Result</th>
<th>Passing Fraction</th>
<th>Created By</th>
<th>Date Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Name Srce</td>
<td>Length more than 3</td>
<td>100</td>
<td>failed</td>
<td>99.192</td>
<td>ganha</td>
<td>2017/11/21 11:00</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No HTML patterns</td>
<td>100</td>
<td>failed</td>
<td>99.999</td>
<td>ganha</td>
<td>2017/11/21 13:40</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No special character</td>
<td>100</td>
<td>failed</td>
<td>90.227</td>
<td>ganha</td>
<td>2017/11/21 11:00</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>Not only num</td>
<td>100</td>
<td>failed</td>
<td>99.743</td>
<td>ganha</td>
<td>2017/11/21 17:00</td>
</tr>
</tbody>
</table>

BRs can run individually or as sets

A threshold can be used to allow a margin of tolerance, which is 0% here

Measured results

Audit info
4.3. Applying BRs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Name</th>
<th>Threshold</th>
<th>Result</th>
<th>Passing Fraction</th>
<th>Created By</th>
<th>Date Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Name Srce</td>
<td>Length more than 3</td>
<td>100</td>
<td>failed</td>
<td>99.192</td>
<td>ganha</td>
<td>2017/11/21</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No HTML patterns</td>
<td>100</td>
<td>failed</td>
<td>99.999</td>
<td>ganha</td>
<td>2017/11/21</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>No special character</td>
<td>100</td>
<td>failed</td>
<td>90.227</td>
<td>ganha</td>
<td>2017/11/21</td>
</tr>
<tr>
<td>T Name Srce</td>
<td>Not only num</td>
<td>100</td>
<td>failed</td>
<td>99.743</td>
<td>ganha</td>
<td>2017/11/21</td>
</tr>
</tbody>
</table>

Drill-down
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

• 1. Main concepts
• 2. Drill-down into entities and attributes
• 3. Relations into the data
• 4. Business rules
• 5. Profiling report and iterating with business

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
5. Profiling report and iterating with business

• **Iterating with data users and application area specialists is key**

• Interpreting profiling results
  – What is not an issue
  – What is an issue
  – Setting priorities
  – Comparing sources

• Follow-up and monitoring can be supported with a profiling report
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)
- 1. Main concepts
- 2. Conditional operations
- 3. Parsing-enabled standardization
- 4. Validating and enriching addresses

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

• 1. Main concepts
• 2. Conditional operations
• 3. Parsing-enabled standardization
• 4. Validating and enriching addresses

Part 3: Data matching and Window keys (performance)

Conclusion & questions
1. Standardization: Main concepts

1. What is it?

2. How DQ tools process data
1.1. Main concepts: what is data standardization?

- Building standards: unambiguous conventions for a correct formal representation of data based on simple business rules
1.1. Main concepts: what is data standardization?

- Building standards: unambiguous conventions for a correct formal representation of data based on simple business rules
  - Eg: « All mobile numbers should be represented as:
    \[+NN NNN NNN NNN\]
    without /, -, (), and with spaces each 3 chars ».

- Conforming the representation of data to the agreed standard
  - Profiling can help discover standardization issues
1.1. Main concepts: what is data standardization?

- Building standards: unambiguous conventions for a correct formal representation of data based on simple business rules
  - Eg: « All mobile numbers should be represented as:
    
    +NN NNN NNN NNN
    
    without /, -, (), and with spaces each 3 chars ».

- Conforming the representation of data to the agreed standard
  - Profiling can help discover standardization issues

(http://www.smalsresearch.be/?wpfb_dl=85).

1.1. Main concepts: what is data standardization?

- Solving the lack of standardization *per se*:
1.1. Main concepts: what is data standardization?

- Solving the lack of standardization per se:
  - in one data source
  - across databases
    - solving inconsistencies in the (re-)use of data concepts
    - transversal data management, Master Data Management
    - requires breaking down siloes, and governance
  - across institutions
    - Inter-institutional Master Data Management, even more governance
1.1. Main concepts: what is data standardization?

- Solving the lack of standardization *per se*:
  - in one data source
  - across databases
    - solving inconsistencies in the (re-)use of data concepts
    - transversal data management, Master Data Management
    - requires breaking down siloes, and governance
  - across institutions
    - Inter-institutional Master Data Management, even more governance

- Or as an intermediary step in *fuzzy matching*
  - standardization = best practice
  - greatly improving reliability of matching results
1.2. Main concepts: How DQ tools process data

• Unlike profiling, with standardization (and then matching):
  – We modify entity / table schemas
    • Create, delete, merge, rename columns
    • Join or split tables
  – We transform the data itself
    • Cleansing, concatenating, splitting…
    • Validating, enriching
    • Merging rows
    • Etc.

• Thus, we need to understand how data gets processed in a DQ tool
1.2. Main concepts: How DQ tools process data – Spreadsheet-like interfaces

• Data is almost permanently shown on-screen
  – Possibly with some statistics
  – Most often, only a sample for performance reasons

• Transformations are done “in-place”
  – Can be recorded as a script for later re-use
  – One final file as a result

• Lightweight, great for:
  – Quick fixes
  – Reasonable datasets
  – Modest budgets
1.2. Main concepts: How DQ tools process data – Spreadsheet-like interfaces

- Data is almost permanently shown on-screen
  - Possibly with some statistics
  - Most often, only a sample for performance reasons

- Transformations are done “in-place”
  - Can be recorded as a script for later re-use
  - One final file as a result

- Lightweight, great for:
  - Quick fixes
  - Reasonable datasets
  - Modest budgets

E.g.: OpenRefine, Trifacta Data Wrangler

- The interface is usually “IDE”-like
  - Central panel: development / data area
  - Side panel(s): processes palette, entities / tables, projects
  - Bottom panel: console / logs

- Data from process(es) to process(es)
  - Input(s) → Process(es) → Output(s)
  - Intermediary files are available
  - Designed to be

- Higher flexibility - Higher complexity
  - Processes are dedicated to specific tasks
  - Each process is a tool by itself
  - Data routing freedom
  → Steeper learning curve

- The interface is usually “IDE”-like
  - Central panel: development / data area
  - Side panel(s): processes palette, entities / tables, projects
  - Bottom panel: console / logs

- Data from process(es) to process(es)
  - Input(s) → Process(es) → Output(s)
  - Intermediary files are available
  - Designed to be

- Higher flexibility - Higher complexity
  - Processes are dedicated to specific tasks
  - Each process is a tool by itself
  - Data routing freedom
  → Steeper learning curve

E.g.: Trillium, Talend Studio, Informatica, etc.

E.g.: Trillium, Talend Studio, Informatica, etc.

Data table (with profiling metadata)

Data table (no profiling metadata yet)

E.g. : Trillium, Talend Studio, Informatica, etc.
1.2. Main concepts: How DQ tools process data

- As a rule of thumb: Every change in extra attributes
  - Original data never overwritten
  - Comparable and reversible changes
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment
  • 1. Main concepts
  • 2. Conditional operations
  • 3. Parsing-enabled standardization
  • 4. Validating and enriching addresses

Part 3: Data matching and Window keys (performance)

Conclusion & questions
2. Conditional operations

- Built manually
  - Various languages/scripts: Java, GREL, etc.
  - Executed if a condition is met

- **Character-level recodings**
  - Managing invisible characters (e.g. control chars)
  - Untypable, targeted with their hex value
  - E.g. CR LF: 0x0D 0x0A (Windows), Sub: 0x1A (certain OSes)
2. Conditional operations

- Built manually
  - Various languages/scripts: Java, GREL, etc.
  - Executed if a condition is met

- In-attribute changes
  - Moves, substitutions or deletions inside an attribute
2. Conditional operations

• Built manually
  – Various languages/scripts : Java, GREL, etc.
  – Executed if a condition is met

• Join-based recodings
  – Substitutions, deletions, enrichments, with a « From → To » file

Recode Table (datamask.csv)

<table>
<thead>
<tr>
<th>Original Mask</th>
<th>Recode Mask</th>
<th>N = Numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/N/NNNN,0N-0N-NNNN</td>
<td>N/N/NNNN,0N-NN-NNNN</td>
<td>NN/N/NNNN,NN-0N-NNNN</td>
</tr>
<tr>
<td>NN/N/NNNN,NN-0N-NNNN</td>
<td>NN/N/NNNN,NN-NN-NNNN</td>
<td>N/NNNNNN,0N-NN-NNNN</td>
</tr>
</tbody>
</table>

Source: Trillium interactive documentation
2. Conditional operations

• Built manually
  – Various languages/scripts: Java, GREL, etc.
  – Executed if a condition is met

• Processing / enriching values with webservice calls
  – E.g. geocoding

<table>
<thead>
<tr>
<th>Address</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVENUE FONSNY 20 1060 BRUXELLES</td>
<td></td>
</tr>
</tbody>
</table>

Geocoding API

<table>
<thead>
<tr>
<th>Address</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVENUE FONSNY 20 1060 BRUXELLES</td>
<td>50.835827;4.3382999</td>
</tr>
</tbody>
</table>
2. Conditional operations

• Built manually
  – Various languages/scripts: Java, GREL, etc.
  – Executed if a condition is met

• Function-based operations
  – In- or inter-attribute changes
  – Versatile: Anything that can be the output of a function

Here: filling extra working attribute (TSQ_NAME) with original name value (‘T Name Srce’) trimmed and fully converted to upper case.
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment
- 1. Main concepts
- 2. Conditional operations
- 3. Parsing-enabled standardization
- 4. Validating and enriching addresses

Part 3: Data matching and Window keys (performance)

Conclusion & questions
3. Parsing-enabled standardization

1. Basic principles

2. Data parsing in a DQ tool
3.1. Parsing-enabled standardization: basic principles

• Processing attributes according to their nature
  – Parsing values into categories
  – Attribute X is a PERSON_NAME
  – Attribute Y is an APPARTMENT_NUMBER

• Knowledge bases
  – Standardization definitions and patterns
  – Specific to each parsing category

• Formal validity of data
### 3.1. Parsing-enabled standardization: basic principles

<table>
<thead>
<tr>
<th>FROM</th>
<th>PARSED AS</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betty</td>
<td>First name</td>
<td>Elizabeth</td>
</tr>
<tr>
<td>Lizzy</td>
<td></td>
<td>ASBL</td>
</tr>
<tr>
<td>Beth</td>
<td>Legal form</td>
<td></td>
</tr>
<tr>
<td>asbl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V. Z.W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assoc. Sans But Lucratif</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0485/123 456</td>
<td>Mobile</td>
<td>+32 485 123 456</td>
</tr>
<tr>
<td>(0032) 485.123.456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+32485123456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24/01/17</td>
<td>Date</td>
<td>2017-01-24</td>
</tr>
</tbody>
</table>
### 3.1. Parsing-enabled standardization: basic principles

<table>
<thead>
<tr>
<th>FROM</th>
<th>PARSED AS</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betty</td>
<td>First name</td>
<td>Elizabeth</td>
</tr>
<tr>
<td>Lizzy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>asbl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V. Z.W</td>
<td>Legal form</td>
<td>ASBL</td>
</tr>
<tr>
<td>Assoc. Sans But Lucratif</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0485/123 456</td>
<td>Mobile</td>
<td>+32 485 123 456</td>
</tr>
<tr>
<td>(0032) 485.123.456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+32485123456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24/01/17</td>
<td>Date</td>
<td>2017-01-24</td>
</tr>
</tbody>
</table>

**IN EXTRA ATTRIBUTES**

**NOT OVERRIDING ORIGINAL DATA!**
3.2. Parsing-enabled standardization: Data parsing in a DQ tool

- Various **manual approaches**:  
  - Conditional operations  
  - Regexes

- Some tools go further, providing **pre-built, language-specific**:  
  - Context-free grammar  
  - Context-sensitive grammar  
    ...and the ability to edit / expand them
3.2. Parsing-enabled standardization: Data parsing in a DQ tool – Grammar-based approach

- Standardize according a set (50 000+) of
  - Rules:

    'GASTRONOMIE' | NAME DEF ATT=BUS
    = If I see « Gastronomie » in a name field, I’ll consider it as part of a Business name.

    'GEERT' | NAME DEF ATT=GVN-NM GEN=M
    = « Geert » is a first name for a male individual.

    'POB' | STREET DEF ATT=POBOX REC='POSTBUS'
    = If I see « POB » in a street attribute, I’ll consider that address as a postbox Indicator and I’ll replace it with « POSTBUS ».

- And Patterns:

    'ALPHA STR-NM HSNO 1ALPHA'
    PATTERN STREET DEF
    REC='STR-NM STR-NM HSNO APT'
    = If I see a pattern made of an alphabetic string, a street name string, a house number and 1 alphabetic character, it’s a street and I’ll recode it to a street name, house number and appartment.

E.g.: « Fonsny Street 20 B »
3.2. Parsing-enabled standardization: Data parsing in a DQ tool – Product data

<table>
<thead>
<tr>
<th>P/N</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1774-5674</td>
<td>TUBE, CENTRIFUGE POLY S 15ML (CS/500)CONICAL-BOTTOM</td>
</tr>
<tr>
<td>1774-5675</td>
<td>TUBE, CENTRIFUGE PPL 15ML (CS/500)CONICAL-BOTTOM</td>
</tr>
<tr>
<td>1774-4532</td>
<td>TUBE, CENTRIFUGE PPL 50ML (CS/500)CONICAL-BTTMPCK 25/RACK</td>
</tr>
<tr>
<td>1774-4538</td>
<td>TUBE, CENTRIFUGE POLY S 50ML (CS/500)CONICAL-BTMPK 25/RACK</td>
</tr>
<tr>
<td>645-4556</td>
<td>PIPET, CLEAR SEROLOGICAL 2ML (CASE/500)</td>
</tr>
<tr>
<td>195-7934</td>
<td>NUT, LOCK RH,11”</td>
</tr>
<tr>
<td>3324-7955</td>
<td>VIAL, WHEATON 33* CLEAR 4ML (CS/144)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P/N</th>
<th>ITEM NAME</th>
<th>MATERIAL</th>
<th>SIZE</th>
<th>UOM</th>
<th>DESCRIPTOR</th>
<th>PACKAGE</th>
<th>PACK METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1774-5674</td>
<td>CENTRIFUGE TUBE</td>
<td>POLYSTERENE</td>
<td>15ML</td>
<td>ML</td>
<td>CONICAL</td>
<td>CASE/500</td>
<td>BOTTOM PACKED</td>
</tr>
<tr>
<td>1774-5675</td>
<td>CENTRIFUGE TUBE</td>
<td>POLYPROPILENE</td>
<td>15ML</td>
<td>ML</td>
<td>CONICAL</td>
<td>CASE/500</td>
<td>BOTTOM PACKED</td>
</tr>
<tr>
<td>1774-4532</td>
<td>CENTRIFUGE TUBE</td>
<td>POLYPROPILENE</td>
<td>50ML</td>
<td>ML</td>
<td>CONICAL</td>
<td>CASE/500</td>
<td>BOTTOM PACKED 25/RACK</td>
</tr>
<tr>
<td>1774-4538</td>
<td>CENTRIFUGE TUBE</td>
<td>POLYSTERENE</td>
<td>50ML</td>
<td>ML</td>
<td>CONICAL</td>
<td>CASE/500</td>
<td>BOTTOM PACKED 25/RACK</td>
</tr>
<tr>
<td>0645-4556</td>
<td>SEROLOGICAL PIPET</td>
<td></td>
<td>2ML</td>
<td>ML</td>
<td>CLEAR</td>
<td>CASE/500</td>
<td></td>
</tr>
<tr>
<td>0195-7934</td>
<td>LOCK NUT</td>
<td></td>
<td>11IN</td>
<td>IN</td>
<td>RIGHT HAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3324-7955</td>
<td>WHEATON VIAL</td>
<td></td>
<td>4ML</td>
<td>ML</td>
<td>CLEAR</td>
<td>CASE/144</td>
<td></td>
</tr>
<tr>
<td>Contents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Introduction: DQ fundamentals</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Part 1: Data Profiling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Part 2: Parsing, Standardization &amp; Address enrichment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 1. Main concepts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 2. Conditional operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 3. Parsing-enabled standardization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 4. Validating and enriching addresses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Part 3: Data matching and Window keys (performance)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Conclusion &amp; questions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Validating and enriching addresses

1. Basic principles

2. Address validation in a DQ tool

3. Parsing and postal validation hand in hand
3.1. Validating and enriching addresses: basic principles

• **Fundamental validity** of data
  – != « this looks like a correct address » (parsing)
  – = « this is a correct address »

• **Currently, knowledge-based approaches**
  – Addresses are very volatile concepts
  – Few standards exist (EU : Inspire ; BE : Best Address)
  – Ubiquitous and strategic problem (clients DBs, public administrations, B2B...)

ULB MASTIC Smals ICT for society
3.2. Validating and enriching addresses: address validation in a DQ tool

- **The local database approach**
  - DB stored in the DQ server
    - Provided by the DQ tool editor
    - Theoretically, could be self-provided by the user
  - The server admin will need to push updates to the DB
  - The tool will match input data in batch against this DB

- **The webservice approach**
  - No local access to the DB itself
  - The service provider pushes updates himself
  - The tool will call an API for each address/in small batches

- **Typical results**
  - Validating or correcting addresses (or error code if not possible)
  - Statistics about address issues
3.3. Parsing and postal validation hand in hand
Let’s try it!
3.3. Parsing and postal validation hand in hand
Let’s try it!

ASBL SmalS v.z.w.
Av Fny 20
Bxl
3.3. Parsing and postal validation hand in hand
Let’s try it!

ASBL SmalS v.z.w.
Av Fny 20
Bxl

Unstandardized
- Denomination is inconsistent
- Wrong street
- No postcode
- City abbreviation
Let's try it!
Input parsed into multiple attributes and standardized to upper case.
Input parsed into multiple attributes and standardized to upper case.

Name standardized, moving legal forms.

City « Bxl » recoded to « BRUXELLES ».
Street has been corrected
Street has been corrected

Postcode, municipality and region have been added (enrichment).
Street has been corrected. Postcode, municipality, and region have been added (enrichment). Address translated to the language of our choice (here, Flemish).
Extra: A problem to solve
Extra: A problem to solve

- Running a data flow, we noticed a big issue in the postal validation of our data

<table>
<thead>
<tr>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23506</td>
<td>Records Failed At State/City Level.</td>
</tr>
<tr>
<td>496712</td>
<td>Records Failed At Street Name Level.</td>
</tr>
<tr>
<td>4332</td>
<td>Records Failed At The House Number Level.</td>
</tr>
<tr>
<td>401</td>
<td>Records Failed At The Aggregate Components Level.</td>
</tr>
<tr>
<td>0</td>
<td>Records Failed Even Though They Matched, Because Of</td>
</tr>
<tr>
<td>0</td>
<td>Records Passed But Directory Had Partial Or No Str</td>
</tr>
</tbody>
</table>

Postal Directory Date: JAN-2018
Address Accuracy Match: 4.6%
Extra: A problem to solve

- Running a data flow, we noticed a big issue in the postal validation of our data

<table>
<thead>
<tr>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23506</td>
<td>Records Failed At State/City Level.</td>
</tr>
<tr>
<td>496712</td>
<td>Records Failed At Street Name Level.</td>
</tr>
<tr>
<td>4332</td>
<td>Records Failed At The House Number Level.</td>
</tr>
<tr>
<td>401</td>
<td>Records Failed At The Aggregate Components Level.</td>
</tr>
<tr>
<td>0</td>
<td>Records Failed Even Though They Matched, Because</td>
</tr>
<tr>
<td>0</td>
<td>Records Passed But Directory Had Partial Or No Str</td>
</tr>
</tbody>
</table>

Postal Directory Date: JAN-2018
Address Accuracy Match: 4.6%
Extra: A problem to solve

• Drilling down into the data, we see:

<table>
<thead>
<tr>
<th>Tsq Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUE DE CHIEVRES(T) 17</td>
</tr>
<tr>
<td>PLACE SAINTE-ANNE(COM) 21</td>
</tr>
<tr>
<td>PLACE COMMUNALE(LL) S/N</td>
</tr>
<tr>
<td>PLACE COMMUNALE(LL) 1</td>
</tr>
<tr>
<td>GRAND'PLACE(R) 1</td>
</tr>
<tr>
<td>GRAND'PLACE(L) 12</td>
</tr>
<tr>
<td>GRAND PLACE(BT) 11</td>
</tr>
<tr>
<td>RUE SAINT-PAUL(BIN) 14</td>
</tr>
<tr>
<td>GRAND'PLACE(CH) 13</td>
</tr>
<tr>
<td>PLACE ALBERT 1ER(FRO) 38</td>
</tr>
<tr>
<td>PLACE ALBERT 1ER(FRO) 38</td>
</tr>
<tr>
<td>GRAND-PLACE (MGS) 1</td>
</tr>
<tr>
<td>RUE SAINT MARTIN(MIC) 71</td>
</tr>
</tbody>
</table>
Extra: A problem to solve

• Drilling down into the data, we see:

<table>
<thead>
<tr>
<th>Tsq Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUE DE CHIEVRES(T) 17</td>
</tr>
<tr>
<td>PLACE SAINTE-ANNE(COM) 21</td>
</tr>
<tr>
<td>PLACE COMMUNALE(LL) S/N</td>
</tr>
<tr>
<td>PLACE COMMUNALE(LL) 1</td>
</tr>
<tr>
<td>GRAND'PLACE(R) 1</td>
</tr>
<tr>
<td>GRAND'PLACE(L) 12</td>
</tr>
<tr>
<td>GRAND PLACE(BT) 11</td>
</tr>
<tr>
<td>RUE SAINT-PAUL(BIN) 14</td>
</tr>
<tr>
<td>GRAND'PLACE(CH) 13</td>
</tr>
<tr>
<td>PLACE ALBERT 1ER(FRO) 38</td>
</tr>
<tr>
<td>PLACE ALBERT 1ER(FRO) 38</td>
</tr>
<tr>
<td>GRAND-PLACE (MGS) 1</td>
</tr>
<tr>
<td>RUE SAINT MARTIN(MIC) 71</td>
</tr>
</tbody>
</table>

> 500 000 occurrences !!
Extra: A problem to solve

Conditional operation
Extra: A problem solved – Address correction: 4%  →  96%

• After re-running the flow starting with the Transformer
  – No pattern issues anymore in the Parser
  – Spectacular rise in address validation

<table>
<thead>
<tr>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2129</td>
<td>Records Failed At State/City Level.</td>
</tr>
<tr>
<td>15727</td>
<td>Records Failed At Street Name Level.</td>
</tr>
<tr>
<td>1251</td>
<td>Records Failed At The House Number Level.</td>
</tr>
<tr>
<td>310</td>
<td>Records Failed At The Aggregate Components Level.</td>
</tr>
<tr>
<td>0</td>
<td>Records Failed Even Though They Matched, Because</td>
</tr>
<tr>
<td>0</td>
<td>Records Passed But Directory Had Partial Or No St</td>
</tr>
</tbody>
</table>

Postal Directory Date: JAN-2018

Address Accuracy Match: 96.5%
Extra: A problem solved – Address correction: 4% → 96%

- Prevalence → Structural issue?

- Investigation
  - Application?
  - Public servants?
  - Certain cities or villages?
Industry-grade DQ Tools come with
- 10’s of thousands of rules for Parsing
- knowledge bases for address enrichment, often region-sensitive
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment (PSA)

Part 3: Data matching and Window keys (performance)

Conclusion & questions
## Contents

**Introduction: DQ fundamentals**

**Part 1: Data Profiling**

**Part 2: Parsing, Standardization & Address enrichment**

**Part 3: Data matching and Window keys (performance)**

- 1. Main concepts
- 2. Matching algorithms
- 3. Data matching in a DQ tool
- 4. Performance and window keys
- 5. Running a project

**Conclusion & questions**
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment

Part 3: Data matching and Window keys (performance)

• 1. Main concepts
• 2. Matching algorithms
• 3. Data matching in a DQ tool
• 4. Performance and window keys
• 5. Running a project

Conclusion & questions
1. Data matching: main concepts

1. What is data matching?

2. Attribute level and Record level

3. Deterministic vs Probabilistic data matching
1.1. What is data matching?

- **Linking between records**
  - within one data source (duplicates detection)
  - across multiple sources (reference matching, detection of duplicates and inconsistencies)
  - even with different data models (data integration)

- **…and deduplicating if needed**
  - “Golden record” picking or commonization

- **Some use cases**
  - Creating a new repertory from external sources
  - Fusion between administrations
  - Integration of IT systems and DBs
  - Statistical modeling / datamining mixing referential and transactional data
  - Etc.

- **~ Relationship linking, entity matching, record linkage, entity resolution**
1.1. What is data matching?

- **Dealing with fuzziness**
  - typographical errors, inaccuracies, lack of standardization
  - \(!=\) exact duplicates

- **Agility critical task**
  - definitions not clear from the start:
    - what may or may not be considered as ‘double’ or ‘inconsistent’
  - many iterations with business are necessary

- **Performance critical task**
  - Support many iterations and application-critical deadlines
  - Esp. with millions of records

- **Link with Anomaly Management**
  - AM needs clear and formal definition (validated by business owners) of detection rules and treatment process
1.2. Attribute level and Record level matching

- Matching on two levels
  - Attribute-per-attribute: comparison algos

---

1.2. Attribute level and Record level matching

- Matching on two levels
  - Attribute-per-attribute: comparison algos
  - Then decision per record (aggregation)

Source: Dries Van Dromme

1.2. Attribute level and Record level matching

- Matching on two levels
  - Attribute-per-attribute: comparison algos
  - Then decision per record (aggregation)

Deterministic or probabilistic

Database A
- Record A:
  - nom
  - prenom
  - rue
  - numero
  - cp
  - date_naiss

Database A|B
- Record B:
  - name
  - 1st_name
  - street
  - housenbr
  - postcode
  - gender

Match
Non-match

1.3. Deterministic vs Probabilistic matching
1.3. Deterministic vs Probabilistic matching

Deterministic: **match patterns** approach

<table>
<thead>
<tr>
<th>Lastname</th>
<th>Firstname</th>
<th>Street</th>
<th>Housenb</th>
<th>Postcode</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td><strong>Match</strong></td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td><strong>Suspect</strong></td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td><strong>Fail</strong></td>
</tr>
</tbody>
</table>
1.3. Deterministic vs Probabilistic matching

Deterministic: **match patterns** approach

<table>
<thead>
<tr>
<th>Lastname</th>
<th>Firstname</th>
<th>Street</th>
<th>Housenb</th>
<th>Postcode</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
1.3. Deterministic vs Probabilistic matching

Deterministic: **match patterns** approach

<table>
<thead>
<tr>
<th>Lastname</th>
<th>Firstname</th>
<th>Street</th>
<th>Housenb</th>
<th>Postcode</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>Match</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Suspect</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Probabilistic: **weighted attributes** approach (**very simplified!** here)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lastname</td>
<td>0.40</td>
</tr>
<tr>
<td>Firstname</td>
<td>0.20</td>
</tr>
<tr>
<td>Street</td>
<td>0.35</td>
</tr>
<tr>
<td>Housenb</td>
<td>0.05</td>
</tr>
<tr>
<td>Postcode</td>
<td>0.10</td>
</tr>
</tbody>
</table>

\[
\text{Threshold match} = 0.8
\]

\[
\sim 0.4 - 0.2 + 0.35 - 0.05 + 0.1 = 0.6
\]

Threshold **suspect**
1.3. Probabilistic matching pros & cons

(+) Simplified human intervention (set weights, still w/ business)
(+) Native unmatch probability

(-) Can be difficult to understand or justify a match
   → Danger when dealing with e-gov data:
      considerable impacts (human, legal, financial…)

(-) Weights and thresholds still imply a part of determinism
   (training or estimating)
1.3. Deterministic matching pros & cons

(+) Able to justify every step if legal requirements*
(+) Finer grain control and tuning

(-) Time needed for human iterations (business x IT)
(-) No native unmatch scoring

* “Deterministic” does not imply exact “==” matching. It simply means the decision (match vs non-match) is rule-based.
Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment

Part 3: Data matching and Window keys (performance)
  • 1. Main concepts
  • 2. Matching algorithms
  • 3. Data matching in a DQ tool
  • 4. Performance and window keys
  • 5. Running a project

Conclusion & questions
2. Matching algorithms

- Any character strings
  - Names, streets, numbers, geographic coordinates… wherever there is fuzziness

- “Match”:
  - Does not mean exact match
  - Is entirely depending on the algorithm:
    - Smals \textasciitilde{} Société de Mécanographie pour l’Application des Lois Sociales
    - Smals \textasciitilde{} Smals
    - Smals \textasciitilde{} SMALs
    - Smals \textasciitilde{} Smallz
    - Smals VZW \textasciitilde{} VZW Smals

- Thousands of existing algorithms, always new ones
  - Generic or specific
  - Language agnostic or not
  - Called “comparison routines”, “clustering methods”, “matching functions”, etc.

- Valid in Deterministic AND Probabilistic approaches
2. Matching algorithms: families

- Booleans / Classifiers
  - Rules & predicates
  - Phonetics

- Similarity-based
  - Word-based
  - Token-based
2. Matching algorithms: families

- Booleans / Classifiers
  - Rules & predicates
  - Phonetics

- Similarity-based
  - Word-based
  - Token-based
2. Matching algorithms - Boolean family: Rules & predicates

- Boolean: they output Y or N (2 classes)
- Other classifiers: > 2 discrete output classes

- Typically
  - Generic conventions (law, grammar, etc.)
  - Custom / domain-specific rules
  - Attribute B is <predicate> of Attribute A

- Boolean matching
  
  if (rule(\text{Attribute\_A}, \text{Attribute\_B}))
  then \text{Attribute\_A} “=” \text{Attribute\_B}
## 2. Matching algorithms - Boolean family: Rules & predicates (examples)

<table>
<thead>
<tr>
<th>Attribute record A</th>
<th>Attribute record B</th>
<th>Algorithm</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate</td>
<td>Translator</td>
<td>Stemming</td>
<td>Y</td>
</tr>
<tr>
<td>Vereniging zonder winstooogmerk</td>
<td>VZW</td>
<td>Initials</td>
<td>Y</td>
</tr>
</tbody>
</table>
2. Matching algorithms: families

- Booleans / Classifiers
  - Rules & predicates
  - Phonetics

- Similarity-based
  - Word-based
  - Token-based
2. Matching algorithms - Boolean family: Phonetics

- Useful for:
  - Transcription errors: oral → written
  - Typical pronunciation confusions (e.g. in Fr, “p”-“b”, “an”-“on”)
  - Post office counter, Call centers,…

- E.g. Mr “Dupont”:
  - Dupond
  - Dubont
  - Dubond
  - Dupant
  - …

- Phonetic matching:
  If \( \text{phon}(\text{Attribute}_A) = \text{phon}(\text{Attribute}_B) \)
  then \( \text{Attribute}_A \ “=” \text{Attribute}_B \)
2. Matching algorithms - Boolean family: Phonetics (examples)

- Russel Soundex Algorithm (1918)
  1. Keep first character
  2. Delete a, e, h, i, o, u, w, y
  3. Recode:
     - “1”: B, F, P, V
     - “2”: C, G, J, K, Q, S, X
     - “3”: D, T
     - “4”: L
     - “5”: M, N
     - “6”: R
  4. Retain first 4 characters (padding with 0’s if necessary)
2. Matching algorithms - Boolean family: Phonetics (examples)

- Russel Soundex Algorithm (1918)

1. Keep first character

2. Delete a,e,h,i,o,u,w,y

3. Recode:
   - “1”: B,F,P,V
   - “2”: C,G,J,K,Q,S,X
   - “3”: D,T
   - “4”: L
   - “5”: M,N
   - “6”: R

4. Retain first 4 characters (padding with 0’s if necessary)

- Example

<table>
<thead>
<tr>
<th>Dupont</th>
<th>Dubond</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. D</td>
<td>1. D</td>
</tr>
<tr>
<td>2. DPNT</td>
<td>2. DBND</td>
</tr>
<tr>
<td>3. D153</td>
<td>3. D153</td>
</tr>
</tbody>
</table>
2. Matching algorithms - Boolean family: Phonetics (examples)

• And many others:
  – Metaphone (no length limit)
  – Double Metaphone (language specificities)
  – NYSIIS (US English names)
  – Daitch-Mokotoff
    • Slavic & Yiddic languages
    • 54 entries
  – Fonem
    • French-oriented
    • 64 rules
  – Phonex
  – etc.
2. Matching algorithms - Boolean family: Phonetics (examples)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Algorithm</th>
<th>Algo(A)</th>
<th>Algo(B)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard &amp; Poor’s</td>
<td>Standard de Liège</td>
<td>Soundex</td>
<td>S353</td>
<td>S353</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metaphone</td>
<td>STNTRTPRS</td>
<td>STNTRTTLJ</td>
<td>N</td>
</tr>
</tbody>
</table>
## 2. Matching algorithms - Boolean family: Phonetics (examples)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Algorithm</th>
<th>Algo(A)</th>
<th>Algo(B)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard &amp; Poor’s</td>
<td>Standard de Liège</td>
<td>Soundex</td>
<td>S353</td>
<td>S353</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metaphone</td>
<td>STNTRTPRS</td>
<td>STNTRTTLJ</td>
<td>N</td>
</tr>
</tbody>
</table>
2. Matching algorithms - Boolean family: Phonetics (examples)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Algorithm</th>
<th>Algo(A)</th>
<th>Algo(B)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard &amp; Poor’s</td>
<td>Standard de Liège</td>
<td>Soundex</td>
<td>S353</td>
<td>S353</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metaphone</td>
<td>STNTRTPRS</td>
<td>STNTRTTLJ</td>
<td>N</td>
</tr>
<tr>
<td>McBride</td>
<td>MacBrigge</td>
<td>Metaphone</td>
<td>MKBRJ</td>
<td>MKBRK</td>
<td>N</td>
</tr>
</tbody>
</table>
2. Matching algorithms - Boolean family: Phonetics (examples)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Algorithm</th>
<th>Algo(A)</th>
<th>Algo(B)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard &amp; Poor’s</td>
<td>Standard de Liège</td>
<td>Soundex</td>
<td>S353</td>
<td>S353</td>
<td>Y</td>
</tr>
<tr>
<td>Metaphone</td>
<td></td>
<td>STNTRTPRS</td>
<td></td>
<td>STNTRTLJ</td>
<td>N</td>
</tr>
<tr>
<td>McBride</td>
<td>MacBrigge</td>
<td>Metaphone</td>
<td>MKBRJ</td>
<td>MKBRK</td>
<td>N</td>
</tr>
<tr>
<td>NYSIIS</td>
<td></td>
<td>MCBRAG</td>
<td>MCBRAG</td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>
2. Matching algorithms: families

- Booleans / Classifiers
  - Rules & predicates
  - Phonetics

- Similarity-based
  - Word-based
  - Token-based

• Also called “distance algorithms”
  – Same principle
  – Inverted approach: instead of measuring similarity, you measure differences

• Useful for typos, errors in OCR, etc.

• Output: continuous score
  • More granularity than boolean algos
  • Integer / Float between -1 and 1, 0 and 1, 0 and 100, etc…
  Implementation specific

- Matching based on a distance algorithm:
  - if (distance(Attribute_A, Attribute_B) <= k)
    then Attribute_A “=” Attribute_B
2. Matching algorithms – Similarity family: Word-based (examples)

- **Levenshtein distance**: min. number of «operations» to transform Attribute_B into Attribute_A
  - Insertion (I)
  - Deletion (D)
  - Substitution (S)
  - (Damereau-Levenshtein: Transposition (T))

- **Example**
  - Attribute_A = Smals, Attribute_B = Smallz
  - Smalllz (D « l »)
  - Smals (S « z » → « s »)

⇒ 2 operations
2. Matching algorithms: families

- Booleans / Classifiers
  - Rules & predicates
  - Phonetics

- Similarity-based
  - Word-based
  - Token-based
2. Matching algorithms – Similarity family: Token-based

• Like word-based algos, output: continuous score

• Specific use of token-based approach:
  – Token = “atomic unit of language” (mostly, “words”)
  – Comparing tokens != comparing whole string
  – Most often, word order does not count
  – Possibly take discriminative power of tokens into account
    • If rare token matches, weighs more than another token match (e.g. TF-IDF)

• Token-based matching
  If (token(Attribute_A, Attribute_B) >= thresh)
  then Attribute_A = Attribute_B
2. Matching algorithms – Similarity family: Token-based (examples)

- Jaccard index
  - Given Attribute_A, Attribute_B, Jaccard(Attribute_A, Attribute_B):
    \[
    \frac{|\text{Attribute}_A \cap \text{Attribute}_B|}{|\text{Attribute}_A \cup \text{Attribute}_B|}
    \]
2. Matching algorithms – Similarity family: Token-based (examples)

- Jaccard index
  - Given Attribute_A, Attribute_B, Jaccard(Attribute_A, Attribute_B):

\[
\frac{|\text{Attribute}_A \cap \text{Attribute}_B|}{|\text{Attribute}_A \cup \text{Attribute}_B|}
\]
2. Matching algorithms – Similarity family: Token-based (examples)

• Jaccard index
  – Given Attribute_A, Attribute_B, Jaccard(Attribute_A, Attribute_B):

\[
\text{Jaccard index} = \frac{|\text{Attribute}_A \cap \text{Attribute}_B|}{|\text{Attribute}_A \cup \text{Attribute}_B|}
\]

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Jaccard index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smals VZW ASBL</td>
<td>2/3</td>
</tr>
<tr>
<td>Smals VZW</td>
<td></td>
</tr>
</tbody>
</table>
Matching algorithms – Similarity family: Token-based (examples)
## 2. Matching algorithms: families

<table>
<thead>
<tr>
<th>Rules &amp; predicates</th>
<th>Phonetics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smals “=”</td>
<td>Dupont “=”</td>
</tr>
<tr>
<td>Société de Mécanographie pour l’Application des Lois Sociales</td>
<td>Dubond</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Booleans / Classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smals ASBL “=” VZW Smals</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Similarity-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smals “=” Smlas</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>VZW Smals ASBL “=” ASBL Smals</td>
</tr>
</tbody>
</table>
2. Matching algorithms: families

- Booleans / Classifiers
- Rules & predicates
- Phonetics
- Similarity-based
- Word-based
- Token-based

/!lackbox
software
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment

Part 3: Data matching and Window keys (performance)
  • 1. Main concepts
  • 2. Matching algorithms
  • 3. Data matching in a DQ tool
  • 4. Performance and window keys
  • 5. Running a project

Conclusion & questions
3. Data matching with DQ tools

1. Matching: rules and patterns

2. Interpreting matching results

3. Advanced: multi-matching and transitivity

4. Commonization: the “golden record”
### 3.1. Matching: rules (deterministic approach)

- Example: setting algorithms, columns and score thresholds

<table>
<thead>
<tr>
<th>Description</th>
<th>Score A</th>
<th>Score B</th>
<th>Score C</th>
<th>Score D</th>
<th>Comparison Routine</th>
<th>Propagation</th>
<th>Field Name 1</th>
<th>Field Name 2</th>
<th>Routine Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>business_name</td>
<td>99</td>
<td>69</td>
<td>84</td>
<td></td>
<td>busname</td>
<td></td>
<td>PR_BUSNAME_RECODED_01</td>
<td>PR_BUSNAME_RECODED_01</td>
<td>SORT</td>
</tr>
<tr>
<td>business_sort</td>
<td>99</td>
<td>68</td>
<td>90</td>
<td></td>
<td>busname</td>
<td></td>
<td>PR_BUSNAME_RECODED_01</td>
<td>PR_BUSNAME_RECODED_01</td>
<td>SQUISH</td>
</tr>
<tr>
<td>business_squish</td>
<td>99</td>
<td>69</td>
<td>84</td>
<td></td>
<td>spelling</td>
<td></td>
<td>TS_STREET_NAME</td>
<td>TS_STREET_NAME</td>
<td></td>
</tr>
<tr>
<td>business_substr</td>
<td>99</td>
<td>68</td>
<td>90</td>
<td></td>
<td>substring</td>
<td></td>
<td>PR_BUSNAME_RECODED_01</td>
<td>PR_BUSNAME_RECODED_01</td>
<td></td>
</tr>
<tr>
<td>street_name</td>
<td>99</td>
<td>68</td>
<td>84</td>
<td></td>
<td>streets</td>
<td></td>
<td>TS_HOUSE_NUMBER</td>
<td>TS_HOUSE_NUMBER</td>
<td></td>
</tr>
<tr>
<td>house_number</td>
<td>99</td>
<td>69</td>
<td>90</td>
<td></td>
<td>houseno</td>
<td></td>
<td>PR_BOX1_NUMBER</td>
<td>PR_BOX1_NUMBER</td>
<td></td>
</tr>
<tr>
<td>box_number_incl blanks</td>
<td>95</td>
<td>69</td>
<td>90</td>
<td></td>
<td>aptino</td>
<td></td>
<td>PR_BOX1_NUMBER</td>
<td>PR_BOX1_NUMBER</td>
<td></td>
</tr>
<tr>
<td>box_number_excl blanks</td>
<td>75</td>
<td>69</td>
<td>90</td>
<td></td>
<td>partial1</td>
<td></td>
<td>PR_BOX1_NUMBER</td>
<td>PR_BOX1_NUMBER</td>
<td></td>
</tr>
<tr>
<td>prevent_between [un]...</td>
<td>100</td>
<td>69</td>
<td>84</td>
<td></td>
<td>partial1</td>
<td></td>
<td>Tmp</td>
<td>Tmp</td>
<td></td>
</tr>
</tbody>
</table>
3.1. Matching: patterns (deterministic approach)

- The real power and flexibility of matching: patterns

- Matching pattern:
  - Combination of the rule scores we just saw
  - A pattern is either a passing, suspect or failing match
- Top to bottom
- All thresholds should be met for the pattern to pass.
- If a pattern does not pass, the next one is evaluated.
- If a Failure pattern (e.g. n°999) is hit, comparison of the two current rows stops.
3.2. Interpreting matching results

• When two rows match, a common cluster ID (“match ID”, “group ID”, …) is generated (typically an integer ID)

• The matching pattern ID is also provided
  – Understanding why it matched
  – Fine-tuning
  – Justifying a match if needed

• Generating groups != merging data automatically (unlike OpenRefine)
### 3.2. Interpreting matching results

<table>
<thead>
<tr>
<th>ID</th>
<th>NOM</th>
<th>ADRESSE</th>
<th>CODE POSTAL</th>
<th>LOCALITE</th>
<th>MATCH_ID</th>
<th>MATCH_PATTERN</th>
</tr>
</thead>
<tbody>
<tr>
<td>6885</td>
<td>INST.ST.DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>SCHAERBEEK</td>
<td>00000001053</td>
<td>110</td>
</tr>
<tr>
<td>6885</td>
<td>INST.ST.DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>SCHAERBEEK</td>
<td>00000001053</td>
<td>110</td>
</tr>
<tr>
<td>23117</td>
<td>INSTITUT SAINT-DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>BRUXELLES</td>
<td>00000001053</td>
<td>135</td>
</tr>
</tbody>
</table>
### 3.2. Interpreting matching results

<table>
<thead>
<tr>
<th>ID</th>
<th>NOM</th>
<th>ADRESSE</th>
<th>CODE POSTAL</th>
<th>LOCALITE</th>
<th>MATCH_ID</th>
<th>MATCH_PATTERN</th>
</tr>
</thead>
<tbody>
<tr>
<td>6885</td>
<td>INST.ST.DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>SCHAERBEEK</td>
<td>00000001053</td>
<td>110</td>
</tr>
<tr>
<td>6885</td>
<td>INST.ST.DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>SCHAERBEEK</td>
<td>00000001053</td>
<td>110</td>
</tr>
<tr>
<td>23117</td>
<td>INSTITUT SAINT-DOMINIQUE</td>
<td>RUE CAPORAL CLAES 38</td>
<td>1030</td>
<td>BRUXELLES</td>
<td>00000001053</td>
<td>135</td>
</tr>
</tbody>
</table>
3.2. Interpreting matching results: drill-down

<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
<th>Dist %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000001</td>
<td>3</td>
<td>0.016</td>
</tr>
<tr>
<td>0000000002</td>
<td>3</td>
<td>0.016</td>
</tr>
<tr>
<td>0000000003</td>
<td>2</td>
<td>0.011</td>
</tr>
<tr>
<td>0000000004</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>0000000005</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>0000000006</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>0000000007</td>
<td>2</td>
<td>0.011</td>
</tr>
<tr>
<td>0000000008</td>
<td>2</td>
<td>0.011</td>
</tr>
<tr>
<td>0000000009</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>0000000010</td>
<td>1</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Cluster IDs
3.2. Interpreting matching results: drill-down

- Sort on frequency desc
- Drill-down
- Cluster IDs
3.3. Advanced : multi-matching and transitivity
### 3.3. Advanced : multi-matching and transitivity : results across 3 DBs

- Zoom from previous slide

<table>
<thead>
<tr>
<th>Company</th>
<th>Code</th>
<th>Zip</th>
<th>Street/Location</th>
<th>City</th>
<th>Code/Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Electric International Inc.</td>
<td>116</td>
<td>17</td>
<td>Clonshaugh Ind Est</td>
<td>Dublin</td>
<td>0083656V</td>
</tr>
<tr>
<td>General Electric Int. Inc.</td>
<td>116</td>
<td></td>
<td>BN33FH Clonshaugh Industrial Estate</td>
<td>Dublin</td>
<td></td>
</tr>
<tr>
<td>General Electric</td>
<td>112</td>
<td></td>
<td>G81 8BW HYDEPARK STREET</td>
<td>89 GLASGOW</td>
<td></td>
</tr>
<tr>
<td>General Electric Int. Inc.</td>
<td>112</td>
<td></td>
<td>G3 8BW 2 Central Quay, 89 Hydepark Street</td>
<td>Glasgow</td>
<td></td>
</tr>
<tr>
<td>GENERAL ELECTRIC INTERNATIONAL INC</td>
<td>103</td>
<td>60313</td>
<td>BLEICHSTRABE</td>
<td>64 66 FRANKFURT AM MAIN</td>
<td></td>
</tr>
<tr>
<td>General Electric International Inc.</td>
<td>103</td>
<td>60313</td>
<td>Bleichstraße 64-66</td>
<td>Frankfurt</td>
<td></td>
</tr>
</tbody>
</table>
3.4. Commonization: the “golden record”
3.4. Commonization: the “golden record”

- If deduplication is needed, it is possible to build a “golden record”

- The “golden record” is the result of the best parts of each record in a matching group

- Choosing parts of different records is called commonization

- \(∥\) if deduplication \(\rightarrow\) keep history of previous records!
### 3.4. Commonization: the “golden record”

#### Specific matching routines
- Date
- First Name
- Last Name
- Ignore Punctuation
- Absolute

#### Distinct survivorship routines
- Most Recent
- Complete
- Most Common
- Most Recent
- Best Source

<table>
<thead>
<tr>
<th>Date</th>
<th>First</th>
<th>Last</th>
<th>Phone</th>
<th>Email</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/00</td>
<td>Art</td>
<td>Barrios</td>
<td>908-845-1234</td>
<td><a href="mailto:bigwheels@hotmail.com">bigwheels@hotmail.com</a></td>
<td>WEB</td>
</tr>
<tr>
<td>12/02/2005</td>
<td>A. Barros</td>
<td>908-845-1234</td>
<td><a href="mailto:abarrios@accen.com">abarrios@accen.com</a></td>
<td>CRM</td>
<td></td>
</tr>
<tr>
<td>6/17/2003</td>
<td>Arthur Barrios</td>
<td>(902)-845-4417</td>
<td><a href="mailto:abarrios@accen.com">abarrios@accen.com</a></td>
<td>SAP</td>
<td></td>
</tr>
</tbody>
</table>
3.4. Commonization: the “golden record”
Example of real rules

- Address: the most valid
  - Totally valid
    - If not: problem in house number
      - If not: problem in street name
        » If not: problem in city name
  - If tied: the most frequent address
  - If still tied: the longest
3.4. Commonization: the “golden record”

- Documenting the decisions is key and sometimes even required by law
  – Eg Registre National (NISS and BISS number) ; source : Isabelle Boydens

“BAUDOIN, Roi des Belges,
A tous présents et à venir, Salut.
[...]
Vu l’urgence;
Art. 5 Si le jour ou le mois de naissance d’une personne ne sont pas connus, la date de naissance est composée comme suit : [...]
Si l’année de naissance d’une personne n’est pas connue, [...]
Art. 6 Un numéro d’identification qui a déjà été utilisé ne peut être attribué à nouveau ni avant qu’un délai de cent ans ne se soit écoulé depuis la date de naissance du titulaire précédent, ni avant que celui-ci soit décédé depuis trente ans au moins.
[...]
Art 8. Si deux ou plusieurs numéros d’identification sont attribués à une même personne, un seul numéro d’identification est retenu. Les autres numéros sont détruits. Pour déterminer le numéro retenu, il est donné priorité, en ordre décroissant, au :
- numéro d’identification attribué en exécution du présent arrêté, dont on ne peut déduire la date de naissance, ou une partie de celle-ci, ainsi que le sexe;
- numéro d’identification attribué en exécution du présent arrêté, dont on peut uniquement déduire la date de naissance ou une partie de celle-ci;
- numéro d’identification attribué en exécution du présent arrêté, dont on peut uniquement déduire le sexe;
- numéro d’identification attribué en exécution du présent arrêté, ayant le numéro d’ordre le plus élevé.
Art. 9. Un numéro d’ordre attribué conformément au présent arrêté n’est pas modifié lorsque, après attribution du numéro, les données y reprises relatives à la date de naissance ou au sexe de la personne s’avèrent inexactes [...]

# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction: DQ fundamentals</td>
<td></td>
</tr>
<tr>
<td>Part 1: Data Profiling</td>
<td></td>
</tr>
<tr>
<td>Part 2: Parsing, Standardization &amp; Address enrichment</td>
<td></td>
</tr>
<tr>
<td>Part 3: Data matching and Window keys (performance)</td>
<td>• 1. Main concepts</td>
</tr>
<tr>
<td></td>
<td>• 2. Matching algorithms</td>
</tr>
<tr>
<td></td>
<td>• 3. Data matching in a DQ tool</td>
</tr>
<tr>
<td></td>
<td>• 4. Performance and window keys</td>
</tr>
<tr>
<td></td>
<td>• 5. Running a project</td>
</tr>
<tr>
<td>Conclusion &amp; questions</td>
<td></td>
</tr>
</tbody>
</table>
4. Performance and blocking / windowing

• Matching = comparisons. Naive approach:

\[ N = n^2 \]

• (A bit) Less naive approach: not comparing R with itself

\[ N = n^2 - n \]

- \( n \): number of rows to compare
- \( N \): total number of comparisons
4. Performance and blocking / windowing

- Optimal approach (comparisons are not directional):
  \[ N = \frac{n^2 - n}{2} \]
  - \( n \): number of rows to compare
  - \( N \): total number of comparisons

- Thus time complexity remains \( \sim O(n^2) \)
4. Performance and blocking / windowing

- \( N \) comparisons in practice
  - \( n = 3 \) rows \( \Rightarrow N = 3 \)
  - \( n = 6 \) \( \Rightarrow N = 15 \)
  - \( n = 10\,000\,000 \) \( \Rightarrow N = 49\,999\,995\,000\,000 \)

- For each comparison (pair of rows)
  * \( p \): from one up to dozens of patterns to test
    * \( a \): multiple attributes to process per pattern per row
      * \( t \): from one up to dozens of transformations per attribute (comparison algorithms)
In total, $N_p \cdot (2\alpha t + 1)$ logical operations.

- For 10 million rows:
  \[ N_p \cdot (2\alpha t + 1) = 80999991900000000 \]

- Assuming a common situation where $p = 20$, $\alpha = 4$, and $t = 10$. 

80999991900000000
In total, \( Np \times (2at + 1) \) logical operations.

- For 10 million rows:
  \[
  Np \times (2 \times 4 + 1) = 80\,999\,991\,900\,000\,000
  \]

- Assuming a common situation where \( p = 20 \), \( a = 4 \), and \( t = 10 \).
4. Performance and blocking / windowing

• In total, \( N_p \times (2at + 1) \) logical operations
  – For 10 million rows:

\[
N_p \times (2at + 1) = 80 \, 999 \, 991 \, 900 \, 000 \, 000
\]

• Assuming a common situation where \( p = 20 \), \( a = 4 \), and \( t = 10 \).
4. Performance and blocking / windowing: principle

- Derive « **keys** » to split data rows in **subgroups** (windows)
  - /!!\ quality of the source attributes that are used → business!

- E.g. key based on 1 rule (many other possible choices):
  - 4 char Soundex*(T_NAME_SRCE)
  - Needs to be tuned iteratively

<table>
<thead>
<tr>
<th>T Name Srce</th>
<th>T Street Srce</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>Window Key 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAD ROESELARE</td>
<td>BOTERMARKT</td>
<td>8800</td>
<td>ROESELARE</td>
<td>S336</td>
</tr>
</tbody>
</table>
4. Performance and blocking / windowing: principle

- Comparisons for matching happen **only within each window** (e.g., here in 2 dimensions)

- Size of each window = determinant for feasibility
  - Windows around 500 to 1000 records are a sweet spot
  - Time performance vs. completeness (recall) of matching
4. Performance and blocking / windowing: Multimatching

- Using multiple window keys per dataset to improve matching results
  - Several data sources
  - Several window keys per source
    - Eg, for enterprises: postal code, NACE code (activity category), etc.
  - Several matching processes per source
  - Several matching processes between sources over time

<table>
<thead>
<tr>
<th>T Name Srce</th>
<th>T Street Srce</th>
<th>C Zipcode Srce</th>
<th>T City Srce</th>
<th>Window Key 01</th>
<th>Window Key 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAD ROESELARE</td>
<td>BOTERMARKT</td>
<td>8800</td>
<td>ROESELARE</td>
<td>S336</td>
<td>88B365</td>
</tr>
<tr>
<td>ROESELARE STAD</td>
<td>BOTER MARKT</td>
<td>8800</td>
<td>ROESELARE</td>
<td>R246</td>
<td>88B365</td>
</tr>
</tbody>
</table>
4. Performance and blocking / windowing: performance gains

- Finding appropriate window keys requires analysis and iterations. Worth it.

- After optimizing window keys for the flow we just saw:
  - 59 processes, >6 million rows * 49 attributes avg, 10GB initial srce, 4 matching processes
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment

Part 3: Data matching and Window keys (performance)
  • 1. Main concepts
  • 2. Matching algorithms
  • 3. Data matching in a DQ tool
  • 4. Performance and window keys
  • 5. Running a project

Conclusion & questions
5. Running a project

• Spreadsheet-like approach:
  – Apply modifications in-place
  – Export modified dataset
  – Export modification script for later re-use

• Data flow approach:
  – Build the project through a GUI client
  – Run the project / a sample through the client
  – Export to batch (typically : big piece of Java / Bash / … code)
  – Possibly schedule runs / wait for third party server orders
5. Running a project: DB read/write in batch

**DB server**

<table>
<thead>
<tr>
<th>O_MD5</th>
<th>T_NAME_SRCE</th>
<th>T_STREET_SRCE</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O_MD5</th>
<th>T_NAME_SRCE</th>
<th>T_STREET_SRCE</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DQ server**

<table>
<thead>
<tr>
<th>O_MD5</th>
<th>T_NAME_SRCE</th>
<th>T_STREET_SRCE</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O_MD5</th>
<th>T_NAME_SRCE</th>
<th>T_STREET_SRCE</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O_MD5</th>
<th>T_NAME_SRCE</th>
<th>T_STREET_SRCE</th>
<th>TQ_GOUT_STREET_NAME</th>
<th>LEV1_MATCHED</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Contents

Introduction: DQ fundamentals

Part 1: Data Profiling

Part 2: Parsing, Standardization & Address enrichment

Part 3: Data matching and Window keys (performance)

Conclusion & questions
Conclusion

• To summarize:
  – Technical approach, very evolutive
  – Profiling: data & metadata audit
  – Standardization
    • Parsing
    • Validation & enrichment for some fields
  – Matching and optimizing performance
    • 4 algo (eg. Soundex, Levenshtein) families
    • Blocking
    • Golden record (w/ business)
Conclusion

• General takeaways
  – Never-ending iteration
    • Business owners
    • Methodological approach: going to the source of DQ problems
  – When DQ has strategic impact
    • Changing data usage (migration, integration; evolving anomalies)
    • Business inefficiency
    • Costs
  – Decades of optimizations + performance =
    • Focusing on logic instead of code
    • Dealing with huge datasets in reasonable timespans
    • Easy collab
Conclusion: two complementary approaches – continuity and recursivity

First: identify business priorities, «fitness for use», budget and «cost-benefits»

Preventive approaches

Curative approaches
Conclusion

• Future: “machine learning” or other technical approaches
  – No big breakthrough yet
    • Some tools (e.g. Talend) offer basic ML functionalities
  – Caution around
    • Operational results involving real and validated business case studies
    • The “explainability” of results
Documentation

• Smals Research about data quality tools
  – https://www.smalsresearch.be/?wpfb_dl=62
  – https://www.smalsresearch.be/?wpfb_dl=85