Data Cleaning and Transformation - Tools

Helena Galhardas
DEI IST
Agenda

- ETL/Data Quality tools
  - Generic functionalities
  - Categories of tools
- The AJAX data cleaning and transformation framework
Typical architecture of a DQ system

- **Data Extraction**
- **Data Transformation**
- **Data Loading**
- **Metadata**
- **Dictionaries**
- **Schema Integration**

**SOURCE DATA**

**TARGET DATA**

- Human Knowledge
- Human Knowledge
- Data Analysis
Existing technology

- How to perform data transformation and cleaning to achieve high quality data
  - Ad-hoc programs
    - Programs difficult to optimize and maintain
  - RDBMS mechanisms to guarantee the verification of integrity constraints
    - Do not address important data instance problems
  - Data transformation scripts using an ETL or data quality tool
(ETL and) Data Quality tools
Generic functionalities (1/2)

- **Data sources** – extract data from different and heterogeneous data sources
- **Extraction capabilities** – schedule extracts; select data; merge data from multiple sources
- **Loading capabilities** – multiple types, heterogeneous targets; refresh and append; automatically create tables
- **Incremental updates** – update instead of rebuild from scratch
- **User interface** – graphical vs non-graphical
- **Metadata repository** – stores data schemas and transformations
Generic functionalities (2/2)

- **Performance techniques** – partitioning; parallel processing; threads; clustering
- **Versioning** – version control of data quality programs
- **Function library** – pre-built functions
- **Language binding** – language support to define new functions
- **Debugging and tracing** – document the execution; tracking and analyzing to detect problems and refine specifications
- **Exception detection and handling** – set of input rules for which the execution of data quality program fails
- **Data lineage** – identifies the input records that produced a given data item
## Classification of tools (2005)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Data sources</th>
<th>Extraction</th>
<th>Loading</th>
<th>Incremental updates</th>
<th>Interface</th>
<th>Metadata repository</th>
<th>Performance</th>
<th>Versioning</th>
<th>Function library</th>
<th>Language binding</th>
<th>Debugging</th>
<th>Exceptions</th>
<th>Data lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commercial</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrus Merge/Purge</td>
<td>DB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ChoiceMaker</td>
<td>DB, FF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Data Integrator</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>DataBlade</td>
<td>Informix</td>
<td>-</td>
<td>Informix</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>DataFusion</td>
<td>DB</td>
<td>Y</td>
<td>DB</td>
<td>Y</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DataStage</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DeDupe</td>
<td>DB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dfPower</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DoubleTake</td>
<td>ODBC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ETL*Data Cleanser</td>
<td>Several</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ETLQ</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Firstlogic</td>
<td>DB, FF</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hummingbird ETL</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>M</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Identity Search Server</td>
<td>DB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Informatica ETL</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MatchIT</td>
<td>DB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Merge/Purge Plus</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Migration Architect</td>
<td>Several</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NaDIS</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>G</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>QuickAddress Batch</td>
<td>ODBC</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>G</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Sagent</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>Y</td>
<td>N, SQL</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SQL Server 2000 DTS</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>G</td>
<td>X</td>
<td>-</td>
<td>Y</td>
<td>N</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>SQL Server 2005</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sunopsis</td>
<td>DB, FF</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>SQL</td>
<td>-</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Trillium</td>
<td>Several</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WizRule</td>
<td>DB, FF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WizSame</td>
<td>DB, FF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WizWhy</td>
<td>DB, FF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Research</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ajax</td>
<td>DB, FF</td>
<td>Y</td>
<td>DB</td>
<td>X</td>
<td>NG</td>
<td>X</td>
<td>Y</td>
<td>X</td>
<td>Y</td>
<td>JAVA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Arktos</td>
<td>JDBC</td>
<td>-</td>
<td>JDBC</td>
<td>-</td>
<td>G</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Clio</td>
<td>DB, XML</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Flamingo Project</td>
<td>DB</td>
<td>-</td>
<td>DB</td>
<td>-</td>
<td>NG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>X</td>
</tr>
<tr>
<td>FraqL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>NG</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IntelliClean</td>
<td>DB</td>
<td>-</td>
<td>DB</td>
<td>-</td>
<td>NG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ken State University</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>NG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Potter's Wheel</td>
<td>Y</td>
<td>-</td>
<td>ODBC</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transcm</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Categories of data quality tools
Categories of data quality tools

Data analysis – statistical evaluation, logical study and application of data mining algorithms to define data patterns and rules
Categories of data quality tools

Data profiling – analysing data sources to identify data quality problems
Categories of data quality tools

Data transformation– set of operations that source data must undergo to fit target schema
Categories of data quality tools

Data cleaning—detecting, removing and correcting dirty data
Duplicate elimination– identifies and merges approximate duplicate records
Categories of data quality tools

Data enrichment – use of additional information to improve data quality
<table>
<thead>
<tr>
<th>Categories</th>
<th>Data Analysis</th>
<th>Data Profiling</th>
<th>Data Transformation</th>
<th>Data Cleaning</th>
<th>Duplicate Elimination</th>
<th>Data Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ajax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Arktos</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Clio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flamingo Project</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>FraQL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>IntelliClean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Kent State University</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Potter's Wheel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Transcm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The AJAX data transformation and
Problems of ETL/data quality solutions (1)

Data transformations

App. Domain 1
App. Domain 2
App. Domain 3
Problems of ETL/data quality solutions (1)

The semantics of some data transformations is defined in terms of their implementation algorithms.
Problems of ETL/data quality solutions (2)
Problems of ETL/data quality solutions (2)

- Clean data
- Rejected data

Data Cleaning and Transformation

Dirty Data
Problems of ETL/data quality solutions (2)

Dirty Data → Data Cleaning and Transformation → Clean data and Rejected data

Dirty Data
Problems of ETL/data quality solutions (2)

Clean data  Rejected data

Data Cleaning and Transformation

Dirty Data
There is a lack of interactive facilities to tune a data cleaning application program.
Motivating example (1)
Motivating example (1)

DirtyData(paper:String)
Motivating example (1)

Publications(pubKey, title, eventKey, url, volume, number, pages, city, month, year)

Events(eventKey, name)

Authors(authorKey, name)

PubsAuthors(pubKey, authorKey)

DirtyData(paper: String)
Motivating example (2)


DirtyData
Modeling an ETL/data quality process

A data quality process is modeled by a directed acyclic graph of data transformations.
AJAX features

- An extensible **data cleaning and transformation (quality) framework**
  - Logical operators as extensions of relational algebra
  - Physical execution algorithms

- A **declarative language** for logical operators
  - SQL extension

- A **debugger facility** for tuning a data cleaning program
AJAX features

- An extensible **data cleaning and transformation (quality) framework**
  - Logical operators as extensions of relational algebra
  - Physical execution algorithms

- A **declarative language** for logical operators
  - SQL extension

- A **debugger facility** for tuning a data cleaning program
Logical level: parametric operators
Logical level: parametric operators

- View
- Cluster
- Map
- Match
- Merge
- Apply
Logical level: parametric operators

- **View**: arbitrary SQL query
- **Map**: iterator-based one-to-many mapping with arbitrary user-defined functions
- **Match**: iterator-based approximate join
- **Cluster**: uses an arbitrary clustering function
- **Merge**: extends SQL group-by with user-defined aggregate functions
- **Apply**: executes an arbitrary user-defined algorithm
Logical

Duplicate Elimination

DirtyTitles... DirtyEvents

Extraction

Map

Standardization

Map

Cities Tags

Formatting

Map

DirtyData

Authors

Merge

Cluster

Match
Logical

Duplicate Elimination

DirtyTitles...  DirtyEvents  DirtyAuthors

Extraction

Map

Standardization

Map

Cities Tags

Formattin

Map

DirtyData

Authors

Duplicate Elimination

Map

Map

Map
Match

- Input: 2 relations
- Finds data records that correspond to the same real object
- Calls distance functions for comparing field values and computing the distance between input tuples
- Output: 1 relation containing matching tuples and possibly 1 or 2 relations containing non-matching tuples
Example

Duplicate Elimination
Example

Duplicate Elimination
CREATE MATCH MatchDirtyAuthors
FROM DirtyAuthors da1, DirtyAuthors da2
LET distance = editDistance(da1.name, da2.name)
WHERE distance < maxDist
INTO MatchAuthors

Duplicate Elimination
**Example**

```
CREATE MATCH MatchDirtyAuthors
FROM DirtyAuthors da1, DirtyAuthors da2
LET distance = editDistance(da1.name, da2.name)
WHERE distance < maxDist
INTO MatchAuthors

Input:
```

*Duplicate Elimination*
Implementation of the match

\[ \forall s_1 \in S_1, s_2 \in S_2 \]

\[ (s_1, s_2) \text{ is a match if } \]

\[ \text{editDistance} (s_1, s_2) < \text{maxDist} \]
Nested loop

- Very expensive evaluation when handling large amounts of data
Nested loop

- **Very expensive** evaluation when handling large amounts of data

⇒ **Need alternative execution algorithms for the same logical specification**
A database solution

CREATE TABLE   MatchAuthors AS
    SELECT authorKey1, authorKey2, distance
    FROM (SELECT a1.authorKey authorKey1,
                a2.authorKey authorKey2,
                editDistance (a1.name, a2.name) distance
    FROM DirtyAuthors a1, DirtyAuthors a2)
    WHERE distance < maxDist;
A database solution

CREATE TABLE MatchAuthors AS
  SELECT authorKey1, authorKey2, distance
  FROM (SELECT a1.authorKey authorKey1,
            a2.authorKey authorKey2,
            editDistance (a1.name, a2.name) distance
  FROM DirtyAuthors a1, DirtyAuthors a2)
WHERE distance < maxDist;

- No optimization supported for a Cartesian product with external function calls
Window scanning
Window scanning

S

\[ n \]
Window scanning

$s$

---

---

---

---

---

---

---

---

---

\[ n \]
Window scanning

\[ S \]

- May lose some matches
String distance filtering

$S_1$

$\overline{\text{John Smith}}$

$S_2$

$\begin{array}{l}
\overline{\text{John Smit}} \\
\overline{\text{Jogn Smith}} \\
\overline{\text{John Smithe}}
\end{array}$

maxDist = 1
String distance filtering

$S_1$

$S_2$

$\text{editDistance}$

$\text{maxDist} = 1$
AJAX features

- An extensible **data cleaning and transformation (quality) framework**
  - Logical operators as extensions of relational algebra
  - Physical execution algorithms

- A **declarative language** for logical operators
  - SQL extension

- A **debugger facility** for tuning a data cleaning program
DEFINE FUNCTIONS AS
Choose.uniqueString(OBJECT[]) RETURN STRING THROWS CiteSeerException
Generate.generateId(INTEGER) RETURN STRING
Normal.removeCitationTags(STRING) RETURN STRING (600)

DEFINE ALGORITHMS AS
TransitiveClosure
SourceClustering(STRING)

DEFINE INPUT DATA FLOWS AS
TABLE DirtyData
(paper STRING (400)
);
TABLE City
(city STRING (80),
citysyn STRING (80)
)
KEY city,citysyn;

DEFINE TRANSFORMATIONS AS
CREATE MAPPING mapKeDiDa
FROM DirtyData Dd
LET keyKdd = generateld(1)
{SELECT keyKdd AS paperKey, Dd.paper AS paper
KEY paperKey
CONSTRAINT NOT NULL mapKeDiDa.paper
}
DEFINE FUNCTIONS AS
Choose.uniqueString(OBJECT[]) RETURN STRING
THROWS CiteSeerException
Generate.generateId(INTEGER) RETURN STRING
Normal.removeCitationTags(STRING)

DEFINE ALGORITHMS AS
TransitiveClosure
SourceClustering(STRING)

DEFINE INPUT DATA FLOWS AS
TABLE DirtyData (paper STRING (400))
TABLE City (city STRING (80), citysyn STRING (80))
KEY city, citysyn;

DEFINE TRANSFORMATIONS AS
CREATE MAPPING mapKeDiDa FROM DirtyData Dd
LET keyKdd = generateId(1)
{SELECT keyKdd AS paperKey, Dd.paper AS paper
KEY paperKey
CONSTRAINT NOT NULL mapKeDiDa.paper
}
AJAX features

- An extensible **data cleaning and transformation (quality) framework**
  - Logical operators as extensions of relational algebra
  - Physical execution algorithms

- A **declarative language** for logical operators
  - SQL extension

- A **debugger facility** for tuning a data cleaning program
Management of exceptions
Management of exceptions

- **Problem**: to mark tuples not handled by the cleaning criteria of an operator
Management of exceptions

- **Problem**: to mark tuples not handled by the cleaning criteria of an operator

- **Solution**: to specify the generation of exception tuples within a logical operator
  - exceptions are thrown by external functions
  - output constraints are violated
Debugger facility

- Supports the (backward and forward) data derivation of tuples wrt an operator to debug exceptions

- Supports the interactive data modification and, in the future, the incremental execution of logical operators
Architecture

Program Specification

SPEC GUI

Analyzer

Program internal representation

Optimizer

Optimization

Execution

EXEC GUI

Execution engine

optimal plan

user call

data modification

algorithm invocation

function calls

JDBC calls

Testing & Debugging

DEBG GUI

Explainer

RDBMS

FILES

JDBC calls

Library of algorithms

Library of functions

Library Specification
References

References

AJAX features

- An extensible **data cleaning and transformation (quality) framework**
  - Logical operators as extensions of relational algebra
  - Physical execution algorithms

- A **declarative language** for logical operators
  - SQL extension

- A **debugger facility** for tuning a data cleaning program
Annotation-based optimization

- The user specifies types of optimization
- The system suggests which algorithm to use

Ex:

```
CREATE MATCHING MatchDirtyAuthors
FROM DirtyAuthors da1, DirtyAuthors da2
LET dist = editDistance(da1.name, da2.name)
WHERE dist < maxDist
% distance-filtering: map= length; dist = abs %
INTO MatchAuthors
```