Data quality in ETL process: A preliminary study

Manel Souibgui, Faten Atigui, Saloua Zammali, Samira Si-Said Cherfi, Sadok Ben Yahia

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 676–687

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Data quality in ETL process: A preliminary study

Manel Souibgui[a,b,*], Faten Atigui[b], Saloua Zammali[a], Samira Cherfi[b], Sadok Ben Yahia[a]

[a] *University of Tunis El Manar, Faculty of Sciences of Tunis LIPAH-LR11ES14, Tunis, Tunisia*
{*manel.souibgui,saloua.zammali*}*@fst.utm.tn, sadok.benyahia@fst.rnu.tn*
[b] *Conservatoire National des Arts et Métiers CEDRIC-CNAM, Paris, France*
{*faten.atigui,samira.cherfi*}*@cnam.fr*

Abstract

The accuracy and relevance of Business Intelligence & Analytics (BI&A) rely on the ability to bring high data quality to the data warehouse from both internal and external sources using the ETL process. The latter is complex and time-consuming as it manages data with heterogeneous content and diverse quality problems. Ensuring data quality requires tracking quality defects along the ETL process. In this paper, we present the main ETL quality characteristics. We provide an overview of the existing ETL process data quality approaches. We also present a comparative study of some commercial ETL tools to show how much these tools consider data quality dimensions. To illustrate our study, we carry out experiments using an ETL dedicated solution (Talend Data Integration) and a data quality dedicated solution (Talend Data Quality). Based on our study, we identify and discuss quality challenges to be addressed in our future research.

Keywords: Business Intelligence & Analytics; ETL quality; Data and process quality; Talend Data Integration; Talend Data Quality

1. Introduction

Business Intelligence & Analytics (BI&A) is defined as ”a broad category of applications, technologies, and processes for gathering, storing, accessing, and analyzing data to help business users make better decisions” [37]. Data warehouses (DW) stand as the cornerstone of BI&A systems. Inmon [21] defines the DW as ”a subject-oriented, integrated, time-variant, non-volatile collection of data in support of managements decision-making process”. Figure A.1 shows BI&A architecture where data is gathered from company operational databases and external data. Gathered data is heterogeneous, and has different types and formats. Before being loaded into the DW, this data is transformed and integrated using the ETL process [34]. The latter performs three basic functions: (i) extraction from data source; (ii) data transformation where the data is converted to be stored in the proper format or structure for the purposes of

* Corresponding author. Tel.: +216-26-767-794
E-mail address: manel.souibgui@fst.utm.tn

1877-0509
10.1016/j.procs.2019.09.223

querying and analysis (e.g., data cleansing, reformatting, matching, aggregation, etc.); (iii) the resulting data set is loaded into the target system, typically the DW. A data mart (data cube) is the access layer of the DW environment that is used to get data out to the decision-makers.
Lavalle et al. [25] conducted a study based on 3 000 business executives and managers. This survey showed that 50% of the respondents consider improvement of data management and BI&A as their priority. It also revealed that 20% of them cited concerns with data quality as a primary obstacle to BI&A systems. Data analytics has been used for many years to provide support for business decision making.
Several authors stressed out that poor data quality has direct and indirect impacts on the underlying business decisions [4, 36]. According to Redman [28], at least three proprietary studies have provided estimates of poor data quality costs between 8 and 12% of revenue range.

In order to properly identify quality-related issues, in the literature, Data Quality (DQ) is recognized as multi-dimensional to better reflect its facets and influences. Each dimension is associated to a set of metrics allowing its evaluation and measurement. The quality dimensions are organized into four categories according to Wang et al. [35] namely: *Intrinsic*, *Contextual*, *Representational* and *Accessibility*. The *Intrinsic* quality dimensions are *accuracy*, *reputation*, *believability* and *provenance*. They rely on internal characteristics of the data during evaluation. The *Contextual* quality is more information than data oriented, since it refers to attributes that are dependent to the context in which data is produced or used. It comprises *amount of data*, *relevance*, *completeness* and *timeliness* quality dimensions. *Representational* quality however, is more related to the way data is perceived by its users and relies on *understandability*, *consistency* and *conciseness* quality dimensions. Finally, *Accessibility* allows measuring the ease with which data could be accessed and covers *accessibility* and *security* dimensions.

Ensuring data quality in the data warehouse and in data cube relies on the quality of the ETL phase which is considered as the sine qua non condition to a successful BI&A system. In this paper, we explore the different facets of quality within an ETL process. We carry out a literature review to gather the different approaches that deal with quality problem in the ETL process. Through our study, we have demonstrated that authors tackle ETL related DQ problems from two main perspectives: *(i)* process centred and *(ii)* data centered. Also, we have shown that both ETL and DQ tools still have DQ limits. These limits are highlighted through Talend Data Integration (TDI) [14] and Talend Data Quality (TDQ) [15]. We refer to the following study as preliminary because we make no claim of completeness.
The remainder of this paper is structured as follows: Section 2 shows a classification of ETL process related DQ problems. In Section 3, we present existing ETL DQ approaches. In Section 4, we present a comparative study based on four ETL tools to show how much these tools consider DQ dimensions. In Section 5, we carry out experiments using an ETL solution, i.e., TDI and a data quality dedicated solution, i.e, TDQ in order to highlight DQ limits of these tools. Finally, Section 6 outlines the limits of the surveyed approaches and tools dealing with DQ problems in the ETL process and presents a set of open issues for research in ETL quality management.

2. Data quality defects within the ETL process

Many reasons stand behind the need of data integration phase within the decision system: (i) heterogeneous formats; (ii) data format can be difficult to be interpreted or ambiguous; (iii) legacy systems using obsolete databases; and (iv) data source's structure is changing over time. All these characteristics of data sources make DQ uncertain. A variety of studies were conducted in the sake of identifying different quality issues within the data integration process. The majority of them agree that DQ faces different challenges. Indeed, ETL is a crucial part in the data warehousing process where most of the data cleansing and curation are carried out. Hence, we propose a classification of typical DQ issues according to ETL stages, i.e., extract, transform and load. As depicted in Table 1, each one of these stages is prone to different quality problems in both schema and instance level.
DQ issues found in the ETL process are the focus of DQ improvement phase. In practice, the improvement phase is often a prerequisite for DQ assessment. The process of integrating DQ into the ETL process is an indicator of the gap between the quality obtained and that expected. Furthermore, overcoming all the DQ problems is still challenging. In the rest, we classify the pioneering approaches that integrate and improve DQ in the ETL process.

Table 1. Examples of ETL data quality problems

		Problems	Descriptions
E	**Schema**	Lack of integrity constraints [27]	Rule that defines the consistency of a given data or dataset in the database (e.g., Primary key, uniqueness). **Example of uniqueness violation:** Two customers having the same SSN number customer 1= (name="John", SSN="12663") , customer 2= (name="Jane", SSN="12663").
		Poor schema design	Imperfect schema level definition [27, 16]. **Example 1:** Attributes names are not significant: "FN" stands for First Name and "Add" stands for Address **Example 2:** Source without schema: "John;Doe;jd@gmail.com;USA".
		Embedded values	Multiple values entered in one attribute [16] . **Example:** name=" John D. Tunisia Freedom 32".
	Instance	Duplicate records	Data is repeated [9]. Misspellings, different ways of writing names and even address changes over time can all lead to duplicate entries [18]. Another form of duplication is the conflicts of entities when inserting a record having the same id as an existing record [39].
		Missing values	Yang et al. have classified missing values into two types: Data in one field appears to be null or empty [39, 19] (i.e., Direct incompleteness) and missing values caused by data operations such as update (i.e., Indirect incompleteness)[39].
T	**Schema**	Variety of data types	Different data types between the source and the target schema.
		Naming conflicts	If we have two data sources which have two synonymous attributes (e.g., gender/sex) then the union of the aforementioned sources requires schema recognition [19, 27, 18].
	Instance	Syntax inconsistency (Structural conflicts)	The date retrieved from the source hasn't the same format as the DW's date [39]. There are a different syntactic representations of attributes whose type is the same [9]. **Example 1:** French date format (i.e., dd/mm/yyyy) is different from that of the US format (i.e., mm/dd/yyyy). **Example 2:** Gender attribute is represented differently in the two data sources, e.g., 0/1, F/M.
L		Wrong mapping of data	Linking a data source to the wrong destination results in the spread of wrong data.
		Wrong implementation of the slowly changing dimension	Problem with versioning of data after every load and update operation [19].

3. Data quality approaches in the ETL process

The dedicated literature reveals a variety of approaches dealing with DQ issues in the ETL process. Since there are many quality problems as aforementioned, several efforts have been provided to tackle DQ challenges. To summarize our literature study, we identified two main streams for DQ management: *(i)* process centered approaches that tackle DQ through ETL process quality; and *(ii)* data centered approaches that consider DQ defaults.

3.1. Process oriented approaches

Quality characteristics are known to be non-orthogonal. Moreover, the perception of quality and the acceptable levels for each characteristic may differ from one user to another. Hence, taking into consideration user's need in term of prioritizing quality characteristics seems important to achieve the desired result. Focusing on user-defined importance of each quality characteristic, authors in [31] have proposed a functional architecture of user centered ETL process, in which DQ characteristics are incorporated. In fact, the user selects among quality characteristics that are of interest in order to receive quality assessment as a feedback. This step enables the user to evaluate how well they fit with the already set strategic goals. Subsequently, the quality evaluation helps users to decide which quality characteristics should be improved in order to reach a process version that fit with users goals.
Theodorou et al. [32] proposed an automatic data generator called *Bijoux*. Based on an ETL process logic model, this tool extracts the semantics of data operations and analysis the constraints they imply over input data. Roughly speaking, *Bijoux* allows users to generate a set of data to test the different execution scenarios. To cover all test scenarios, an algorithm is proposed to enumerate different paths existing in the ETL process which is modeled as a graph. For every path, *Bijoux* analysis the different constraints applied to ETL operations in order to generate the testing

datasets. This approach has been shown to be useful for measuring the performance of different implementations of the same ETL process.
On the other hand, they proposed a tool called *POIESIS* that generates alternative flows by adding flow patterns to the initial flow. This is based on user preferences on different quality characteristics. Each generated flow is accompanied by DQ and performance measures. It's up to the user to decide which one of them is suitable to his quality needs. Among quality characteristics cited in literature, authors have proposed flow component patterns related to some of them such as *performance*, *reliability*, *deduplication* and *filtering of null values*.

3.2. Data oriented approaches

Data oriented approaches tackle quality defaults at data level and try to resolve them. We propose to classify the existing approaches into *(i)* syntactic; *(ii)* semantic; and *(iii)* contextual points of view.

3.2.1. Syntactic oriented approaches

An important quality default is related to data duplication. If a company has different customer databases related to subsidiaries and would like to combine them, it would be an actual challenge to identify matching records, since the same data is presented in different ways. This problem known as *Entity resolution* problem, is defined as the process of identifying and merging records judged to represent the same real-world entity. In [5], Benjelloun et al. proposed three algorithms based on the comparison of features in order to identify matching records.
Likewise, Term Frequency-Inverse Document Frequency (TF-IDF) metric has been used for duplicates detection. Moyer et al. [18] have proposed a new method that extends TF-IDF to address not only duplicates but also missing entries. Indeed, missing entries could hamper the identification of duplicates. For instance, both of these following records: ("John Bruin", " ", "male") and ("John Bruin, "CA", " ") are similar but applying TF-IDF wouldn't be sufficient to have a correct result.
Data loaded from various sources into the DW often has different format and coding (e.g,. date format). Since the user may enter various data formats, authors in [19] have proposed an open source tool called MassEETL that manages data conversion in addition to naming conflicts and missing values handling. Standard format is defined based on a data format mapping rule.

3.2.2. Semantic oriented approaches

Understanding the semantics of data is essential for data integration. Bergamaschi et al. [6] proposed a tool supporting the extraction process that allows a semi-automatic semantic mappings between data source and DW's attributes. To do so, this tool identifies the parts of the data sources schemata that are related to the data warehouse schema and uses syntactic and semantic similarity measures. This work operates in three phases: *(i)* Extraction of the schema description from the sources using wrapper for different types of data sources (RDF, XML, etc.); *(ii)* Annotation using WordNet which is the process of associating the most suitable concept of a reference ontology; *(iii)* Creation of thesaurus of relationships between the schema elements. For the transformation phase, the authors introduced an algorithm to identify the suitable transformation function at the mapping level using the result of the semantic enrichment in order to get data types and the generated thesaurus.
A wide variety of anomaly types may exist in the data, e.g., conflicts between the names of manipulated objects such as attributes that could be homonyms or synonyms. Those problems are caused by the lack of semantics. Moreover, data enrichment by semantic information allows a better cleansing. For example, there is no method to compute the similarity distance allowing the approximation between Pekin and Beijing. The approach in [30] consists in a semantic recognition of the data schema by proposing a semantic name to each column of the source. It is the fact of identifying category and language for values associated to each column using regular expressions and data dictionary. Additionally, a semantic profiling algorithm is proposed to check the syntactic and semantic validity of values in the data sources. Given the set of regular expressions, data dictionary, syntactic and semantic indicators, and the data source, the algorithm returns a set of tables containing the profiling results.

3.2.3. Context oriented approaches

To the best of our knowledge, only few contributions have addressed DQ assessment considering the context of use. Besides, context coverage has been added as a quality in use characteristic according to the quality standard ISO/IEC 25010 [1]. Authors in [38] showed that it is necessary to simultaneously examine both contextual and objective quality attributes. They have proposed a theoretical model to highlight the importance of integrating contextual DQ assessment processes. Bertossi et al. [7] have taken a first step to formalize and investigate the notion of DQ assessment and DQ query answering as context dependent activities using datalog and first-order predicate logic. Another solution is described in [23] where authors have demonstrated how a quality plan can be systematically created for evaluating usability of test cases. Furthermore, they introduced a new set of context factors that might influence DQ evaluation. However, most of the previous studies do not take into account context in DQ evaluation in the ETL process nor emphasize its important role to influence DQ measures.

4. Comparative study: ETL tools and data quality dimensions

Our objective in this section is to study the progress of practitioners. We present a comparative study of ETL tools with the goal of verifying the integration of DQ in ETL components. In fact, extensive surveys in the literature have been carried out to compare leading ETL tools that are available in the market [31, 26]. They analyse data accessibility, performance and functionalities offered by these tools. It is worth mentioning that our comparative study differs from the existing ones as we identify the extent to which ETL tools components incorporate DQ with respect to DQ dimensions. To this end, we have considered four major ETL tools: *(i)* two open source: Talend Data Integration (TDI) and Pentaho Data Integration (PDI) [13] two commercial tools: Informatica Data Integration (IDI) [11] and Microsoft SQL Server Integration Services (SSIS) [12]. The result is summarized in Table 2.

The choice of the considered quality dimensions is justified by their relevance within the ETL process [31]. Practically, the comparison criteria that we consider in the following are the most common quality dimensions that have been considered to handle DQ integration problem:

- **Duplication**: data deduplication is a challenging goal for DQ integration in the ETL tools. The aforementioned tools provide different solutions for data deduplication. The components provided by TDI and PDI remove duplicates based on one or more columns using exact matching. As an example, we detail the TDI `tUniqueRow`

[1] https://www.iso.org/standard/35733.html

Table 2. ETL tools comparison according to data quality dimensions

	Talend Data Integration	Pentaho Data Integration	Microsoft SQL Server Integration Services	Informatica Data Integration
Duplication	**Component**: tUniqRow	**Component**: Unique rows	**Component**: Fuzzy grouping	**Mapping**: filter duplicate records using aggregator transformation that removes duplicates using group by the primary key
Completeness	**Component**: tFilterRow	**Component**: javaFilter	**Component**: Conditional Split	**Transformation**: filter transformation using the IsNull function
Accuracy	**Component**: tVerfiyEmail	**Component**: Mail Validator	-	**Mapplet**: Address validation Email verification
Interpretability	-	-	-	Data domain discovery
Representational conciseness	**Component**: tAddressRowCloud	-	**Component**: Address Parse	**Mapplet**: Address parsing
Representational consistency	-	-	-	-

component in Section 5. As for SSIS, the `Fuzzy grouping` component identifies duplicates after selecting: *(i)* Input columns: columns used to identify duplicates; *(ii)* Matching type: the matching between two values can be either exact matching or fuzzy matching. The fuzzy matching groups rows that have approximately the same values using similarity threshold. On the other hand, IDI provides a *Mapping*, i.e., set of reusable components to filter duplicate records using aggregator transformation. To remove duplicate records, it uses Group By the primary key which will groups all unique records together.

- **Completeness**: completeness is a contextual DQ dimension. To this end, it is not only defined as the degree of absence of missing values [31] but the extent to which all required data are available and of sufficient scope for the context of use [35]. Handling missing data is crucial for further data analysis, otherwise it can rise a biased results. From the technical side, all ETL tools, mentioned in Table 2, provide components that filter the input rows by setting one or more conditions on the selected columns (e.g., tFilterRow can be used to filter rows that contain missing values using relational routines `!Relational.ISNULL(row1.Column1)` if the considered column type is *Integer* or by using the expression `!input_row.Column0.equals("")` whenever the considered column type is *String*). However, none of these tools have considered techniques (e.g., Imputation) that handle missing values according to the context of use.
- **Accuracy**: the extent to which data are correct, reliable, and certified error-free [35]. This criterion assesses the percentage of data without data errors [31]. After delving the considered ETL tools, we noticed that most of them provide components that check the validity of email addresses. Unlike the rest of the studied tools, Informatica *Mapplet* (i.e., a set of reusable components) is not only reduced in structural validity of email address using regular expression, but also ensuring email addresses validity at the domain level. Furthermore, it validates an address against a set of reference address databases.
- **Interpretability**: degree to which user can understand data that they get [31]. To enhance DQ according to this dimension, it is absolutely necessary to integrate semantic recognition operations to the ETL process. This quality dimension is deemed in Informatica by providing a data domain discovery. Since many important data remains undiscovered in data sources or it does not have a significant column name, the data domain can be a predefined or a user-defined repository based on the semantics of columns.
- **Representational conciseness**: the extent to which data are compactly represented (i.e., brief in presentation, yet complete and to the point) [35]. In fact, data sources are likely to have embedded values which lead to have overwhelming data. This problem is often related to address field in the data source, where we can have multiple address elements in the same field. The considered ETL tools in Table 2 have achieved a first step towards the enhancement of the DQ according to this dimension. They provide components for address verification and parsing, and yet, it is important to consider other cases where we can have multi-valued fields.
- **Representational consistency**: the extent to which data are always presented in the same format and are compatible with previous data [35]. To fulfill this dimension, ETL tools have to provide solutions to resolve problems like naming conflicts and syntax inconsistency. In practice, none of the mentioned tools have yielded operators to guarantee the representational consistency while integrating data coming from various data sources.

From the above comparative study, we conclude that ETL tools components improve to a limited extent the DQ according to DQ dimensions. These tools do not take in deeper consideration each DQ dimensions. Although, they are almost similar in terms of their relevance, they differ in their contributions to the enhancement of DQ according to DQ dimensions. We shed light in the following section on some ETL operational limits through illustrative examples.

5. Analysis by examples: Limits of Talend Data Integration and Talend Data Quality

ETL tools are basically based on: *(i)* Connectors for retrieving and transferring data; *(ii)* Operations for complex data manipulation; *(iii)* Mapping the data source schema to the target schema. The phase of data collection and preliminary preparation was generally underestimated. In this section, we present three major DQ problems related to *(i)* TDI operations, i.e, deduplication, union and aggregation; *(ii)* matching analysis technique proposed by TDQ. We illustrate through examples how TDI and TDQ tools tackle those problems.

Table 3. List of components considered in the test examples of Talend Data Integration

Operation type	Talend Component	Description
Extraction	tMysqlInput	Extracts data from MySQL database based on query.
	tFileInputXML	Reads an XML structured file and extracts data row by row.
Transformation	tUniqRow	Belongs to DQ component family. It compares entries and removes duplicate ones from the input flow.
	tMap	Transforms and routes data from one or multiple data sources to single or multiple destinations. TMap can be parametrized to perform different kind of operations (i.g., Join).
	tUnit	Merges data from various sources, based on a common schema.
	tAggregateRow	Performs a calculation on a set of values based on one or more columns.
Loading	tFileOutputDelimited	Creates a delimited file that holds data organized according to the defined schema.
	tLogRow	Exhibits results or data in the running console.

5.1. Talend Data Integration components limits

In order to test whether each component intends to ensure DQ, we carried out several experiments on TDI components. The choice of Talend is motivated to the fact that it is an open source tool, evaluated by Gartner Magic Quadrant[2] as a leader and is supposed to incorporate DQ components. In the following, we unveil through illustrative examples the main weaknesses of tht three TDI components: `tUniqRow`, `tUnit`, and `tAggregateRow`. We classified the used components in our experiments according to the operation type: *(i)* extraction; *(ii)* transformation and *(iii)* loading as depicted in Table 3.

5.1.1. tUniqRow

Our objective is to check to which extent DQ is achieved in TDQ components. We split our example into two scenarios named *S1* and *S2*. For both of them, we consider an excerpt of an ETL process wherein we perform a join operation using TDI components as shown in Figure B.2. In our example, data are extracted from XML file and MySQL table. The XML file holds data about customers that are matched to their orders in the MySQL table through an inner join on the customer ID. We can see that among customers data in the XML file, there are two different elements having the same primary key (i.e., *customer_Id* equals 2). In the sake of ensuring DQ, TDQ tool offers a DQ component that compares entries and removes duplicate one from the input flow. Subsequently, we used `tUniqRow` taking as input customers data stored in the XML file. In the basic setting of this component, the user selects one or more columns on which (s)he wants the deduplication to be carried out. Firstly, in the scenario *S1*, we select the *customer_Id* column. Then, we run the Talend job to perform the join operation using the `tMap` component. The output of the running example is stored on a the tFileOutputDelimeted. We point out that the customer *Adara* has been removed.

In the second scenario *S2*, we consider the above example and we modify the settings of the quality component `tUniqRow` as depicted in Figure B.2. Hence, in addition to the primary key we have selected customer first name. As expected, we can see in the output file that `tUniqRow` keeps duplicate records having the same primary key (i.e., 2). Consequently, the duplication problem is not resolved yet. As a first sight, the problem seems to be a simple user mistake rather than a problem of DQ itself. However, this become more complex if we tackle thousands of inaccurate records. In the above mentioned experiments, we unveiled limits in the `tUniqRow` component which leads to a loss of information. Thus loosing information about one customer leads certainly to wrong analytics results.

5.1.2. tUnit

We performed a unit operation using `tUnit` component (see Figure B.3) which have two input tables stored in a MySQL database. The source tables hold data about customers (i.e., *phone*, *email* and *gender*). They have the same column names but a different column order. In the output file created after running the Talend Job, we point out that data in the second column *email* of the first table is unified with the second column *phone* of the second table.

[2] https://fr.talend.com/resources/2018-gartner-magic-quadrant-data-integration-tools/

Additionally, *Gender* column is united with the *email* column, etc. Hence, merging data from various sources based on common schema but having different column order produces certainly an incorrect result. Moreover, Talend user must be alerted so that (s)he can prevent the spread of wrong data throughout the integration process. Thus, the above mentioned example proves that this component needs a further semantic enrichment in order to detect that columns order of the two data sources is different. So that, it helps the user to fix the problem before its spread in the rest of the data process.

5.1.3. tAggregateRow

In our example illustrated through Figure B.4, we intend to aggregate the students thorough scores based on students names. Thus, we create a Talend Job using the `tAggregateRow` component as an intermediary step, since it requires input and output. The input data about students and their associated scores are extracted from a table stored in a MySQL database. The output data is diplayed in the Talend console using the `tLogRow` component. The view setting of the `tAggregateRow` component offers the option to edit the type of the output data. In our example, the score type is an Integer, whereas the Max column type in the output is Double. In this case, running the Talend Job rises an exception as depicted in Figure B.4. The incompatibility of input and output types presents a major limit related to the `tAggregateRow` component. Moreover, the user will face this problem each time (s)he tries to aggregate data to have Max or Min values. In fact, when we set the *score* attribute type in the input data to Double instead of Integer, running the Talend Job produces a correct result without throwing any exception.

5.2. Talend Data Quality limits

There are a number of usage scenarios where data profiling techniques have an important role to play, most commonly related to DQ assessment and metadata discovery. Obviously, understanding the properties of data sources is of paramount importance to create correct schema mappings and data transformations, and to correctly standardize and cleanse the data [2]. Profiling tools like TDQ, and Informatica Data Quality [3] focus on several analyzes that elucidate the structure and content of data and establish inductions on that data using statistics related to duplicate data, invalid address format, etc. Usually, in this kind of tools, no guidelines are provided to explain how the profiling reports can be used to identify actionable DQ requirements [29]. To the best of our knowledge, all those data quality tools do not evaluate DQ according to different quality dimensions.

By way of example, TDQ, which is an open source profiling tool, generates statistics about data sources and creates matching analysis. The latter is dedicated to analyze duplicates. Since the elimination of duplicate records requires similarity detection, TDQ creates a matching analysis which is based on computing similarity distance using different methods (e.g., Jaro Winkler, Soundex). In the following, we illustrate through an example how TDQ performs a partial duplicate detection. We used a data source stored in MySQL database that contains 5 records. In the latter, we have four duplicate records and one unique records. The first duplicate ones have the same city names but different employee names which are spelled differently (i.e., ”Marhta” and ”Martha”). For the other duplicate records, the names are written differently (i.e., ”John Doe” and ”John D”) and the cities are the same except that they are not written in the same language (i.e., ”Pékin” and ”Beijing”). To carry out a matching analysis, the user starts firstly by choosing the matching key on which (s)he wants to apply the matching algorithm. Secondly, the user specifies the type of the matching algorithm and the threshold above which two values are considered as duplicates. The rest of the configuration includes the setting of the confidence weight that gives importance to certain columns and the confident match threshold.

The result of the matching analysis indicates the existence of 40% duplicates and 60% of unique data. Hence, those statistics are wrong, since Talend offers algorithms that compute only syntactic similarity distance between values. For this reason, the duplication of the employees city (”Pékin” and ”Beijing”) is not detected. A key limitation of this solution is that it does not address semantic similarity which is very important to ensure DQ. For instance, the two

[3] https://www.informatica.com/fr/products/data-quality.html

countries or city names (”Pékin”,”Beijing”) and (”Switzerland”,”Suisse”) are syntactically different but semantically similar.

6. Discussion & Future Research Directions

The quality of multidimensional analysis, relies on the quality of the extracted, transformed and loaded data. In this paper, we have studied DQ approaches in ETL process. We have classified existing approaches into process and data oriented approaches. Among the fifteen quality dimensions shown in [35] that could be relevant to ensure better quality in ETL process, only three dimensions (*performance*, *reliability* and *deduplication*) have been considered in existing works. Furthermore, the absence of semantic analysis while handling incoming data is deemed to be the origin of several DQ problems. For instance, the resolution of *deduplication* problem relies on syntactic similarity without considering the semantic one. We believe that a semantic understanding and comprehension of the input data should increase the reliability of analyses.

On the other hand, we have presented a comparative study based on four ETL tools to show how much these tools consider DQ dimensions. We noticed that ETL tools components improve partially the DQ according to DQ dimensions. We have studied TDI and TDQ as well-known examples of open source tools to evaluate their assistance to quality. We conclude that these tools have many DQ limits that may affect the accuracy of the final result. For instance, the Talend data quality component `tUniqRow`, which is dedicated to remove duplicate entries, can lead to a loss of information whenever the data source is a semi-structured file, e.g., XML and JSON files. In case of two different records having the same *ID*, the `tUniqRow` removes the second one.

The study presented in this paper has shown that the problem of quality in ETL process still requires a lot of efforts, especially if the BI&A system is based on non-relational data sources, e.g., NoSQL databases [1] . With the emergence of big data, data analytics is gaining increasing attention in BI&A solutions that become Data-Driven rather than Business and intuition driven. Predefined processes and analysis models may not be able to provide a clear explanation on why a given phenomenon happens. The solution requires often, extra data that is not included in the corporate systems. Consequently, here is an obvious need to be able to cope with high technology standards to face big data specific complexity (Volume, Variety, Velocity, etc.) [8, 10].
Today, Data Management Systems based on non relational models as key-value, column-oriented, document-oriented and graph-oriented, which adopt a schemaless representation are widely accepted as efficient systems to store data. To store heterogeneous data with non predefined schemas, schemaless databases could be a promising issue. In operational applications, the absence of unique schema grants flexibility, but adds complexity to analytical applications [17]. With this kind of data sources, going on a traditional vision of BI&A solutions implies re-modelling the data warehouse, uploading, transforming and testing the new data [20]. Other issues related to data arise such as DQ, data understanding and availability [33, 24]. These problems related to data will definitely generate performance degradation going against the expected reactiveness. Since these sources are extremely heterogeneous and evolving, an ETL framework needs to be on-demand and extensible [3].

The result of our studies drives us to propose in future works a new approaches within the scope of quality-driven solutions to extract-transform and load data including the semantic integration. In fact, understanding data semantic is of paramount importance in a data integration context [22]. From the outcome of our investigation of ETL tools, it is clear that further integration of semantic recognition would be of interest to resolve DQ problems (e.g., Naming conflicts and duplication). The second issue to tackle is outlier detection. Indeed, mainly we are studying the outlier detection in JSON based structure stream of documents. The implementation of this approach is in progress. It consists on real-time outlier detection in JSON document stream. In fact, with the huge amount of data generated by billions of devices (e.g., Internet of Things), stream processing has become a potential and a major requirement for big data analytics.

Appendix A. BI&A architecture

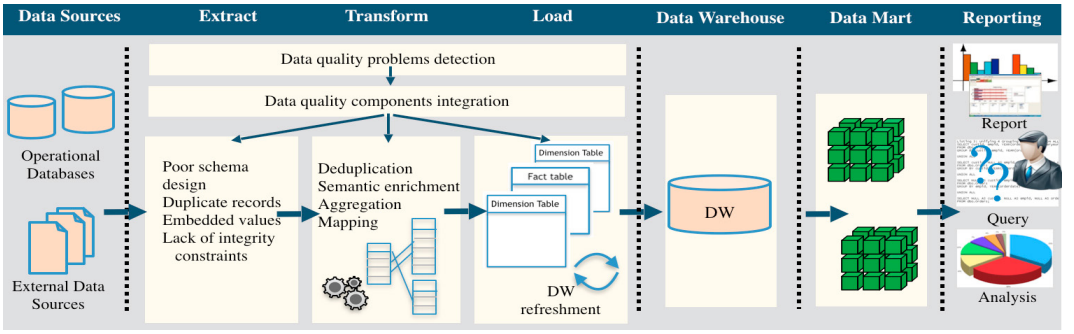

Fig. A.1. Business Intelligence & Analytics system architecture

Appendix B. Limits of Talend Data Integration and Talend Data Quality tools: Examples

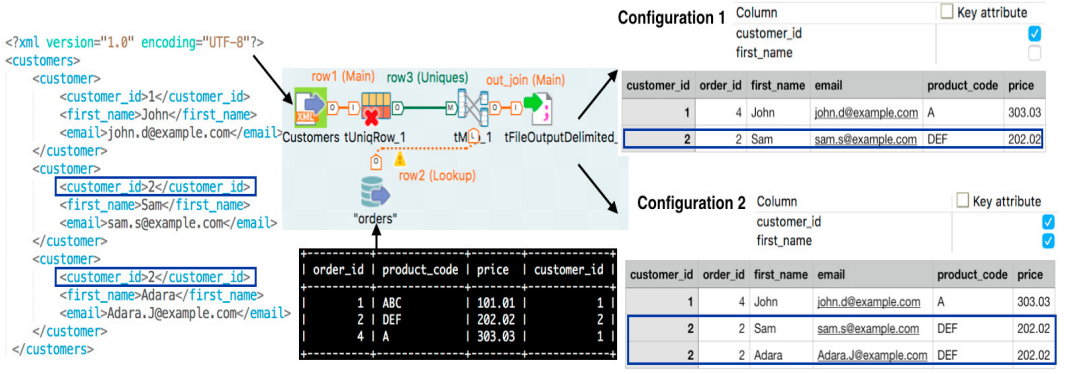

Fig. B.2. Example of information loss and duplicate records produced by Talend tUniqRow component

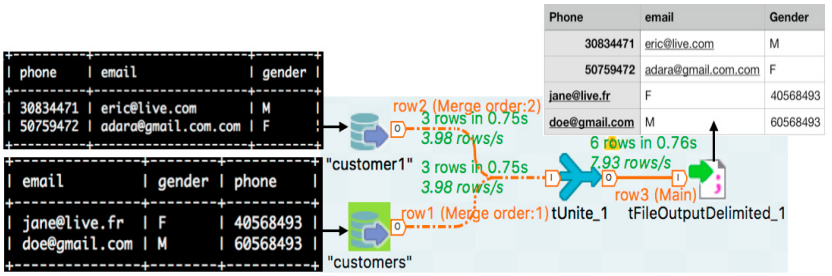

Fig. B.3. Example of incorrect result after using the tUnit component

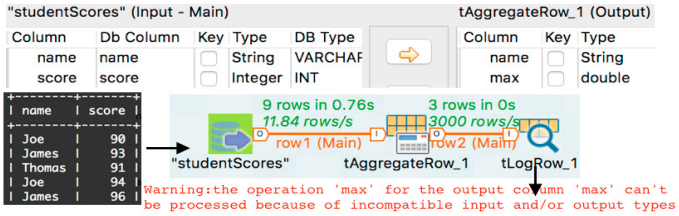

Fig. B.4. Example of incompatibility types problem in Talend tAggregateRow component

References

[1] Abdelhédi, F., Brahim, A.A., Atigui, F., Zurfluh, G., 2017. Mda-based approach for nosql databases modelling, in: Big Data Analytics and Knowledge Discovery - 19th International Conference, DaWaK 2017, Lyon, France, August 28-31, 2017, Proceedings, pp. 88–102.

[2] Abedjan, Z., Golab, L., Naumann, F., 2015. Profiling relational data: a survey. The International Journal on Very Large Data Bases VLDB 24, 557–581.

[3] Ali, S.M.F., 2018. Next-generation ETL framework to address the challenges posed by big data, in: Proceedings of the 20th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data EDBT/ICDT, Vienna, Austria.

[4] Batini, C., Cappiello, C., Francalanci, C., Maurino, A., 2009. Methodologies for data quality assessment and improvement. The journal of ACM computing surveys CSUR 41, 16.

[5] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J., 2008. Swoosh: a generic approach to entity resolution. The International Journal on Very Large Data Bases VLDB 18, 255–276.

[6] Bergamaschi, S., Guerra, F., Orsini, M., Sartori, C., Vincini, M., 2011. A semantic approach to ETL technologies. Journal of Data Knowledge Engineering DKE 70, 717–731.

[7] Bertossi, L.E., Rizzolo, F., Jiang, L., 2010. Data quality is context dependent, in: Proceedings of the International Workshop on Business Intelligence for the Real-Time Enterprise BIRTE, Berlin, Heidelberg. pp. 52–67.

[8] Bizer, C., Boncz, P.A., Brodie, M.L., Erling, O., 2011. The meaningful use of big data: four perspectives - four challenges. Journal of Special Interest Group on Management of Data SIGMOD 40, 56–60.

[9] Boufares, F., Salem, A.B., 2012. Heterogeneous data-integration and data quality: Overview of conflicts, in: Proceedings of the 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications SETIT, Sousse, Tunisia. pp. 867–874.

[10] Chen, H., Chiang, R.H.L., Storey, V.C., 2012. Business intelligence and analytics: From big data to big impact. Journal of Management Information Systems MIS 36, 1165–1188.

[11] Corporation, I., 2018a. Informatica Data Integration Hub. https://www.informatica.com/content/dam/informatica-com/global/amer/us/collateral/data-sheet/data-integration-hub_data-sheet_2473.pdf.

[12] Corporation, M.S.S.I.S., 2017. Fuzzy Grouping Transformation. https://docs.microsoft.com/en-us/sql/integration-services/data-flow/transformations/fuzzy-grouping-transformation?view=sql-server-2017.

[13] Corporation, P., 2018b. Pentaho Data Integration. https://www.hitachivantara.com/en-us/products/big-data-integration-analytics/pentaho-data-integration.html?source=pentaho-redirect.

[14] Corporation, T., 2019a. Talend data integration. https://www.talend.com/resources/l-application-integration-tool/.

[15] Corporation, T., 2019b. Talend data quality. https://fr.talend.com/products/talend-open-studio/data-quality-open-studio/.

[16] Debbarma, N., Nath, G., Das, H., 2013. Analysis of data quality and performance issues in data warehousing and business intelligence. The International Journal of Computer Applications IJCA 79.

[17] Gallinucci, E., Golfarelli, M., Rizzi, S., 2018. Variety-aware OLAP of document-oriented databases, in: Proceedings of the 20th International Workshop On Design, Optimization, Languages and Analytical Processing of Big Data DOLAP, Vienna, Austria.

[18] van Gennip, Y., Hunter, B., Ma, A., Moyer, D., 2018. Unsupervised record matching with noisy and incomplete data. International Journal of Data Science and Analytics IJDSA 6, 109–129.

[19] Gill, R., Singh, J., 2014. "An Open Source ETL Tool - Medium and Small Scale Enterprise ETL MaSSEETL". International Journal of Computer Applications IJCA 108, 15–22.

[20] Golfarelli, M., Rizzi, S., 2018. From star schemas to big data: 20+ years of data warehouse research, in: A Comprehensive Guide Through the Italian Database Research. Springer International Publishing. volume 31 of *Studies in Big Data*, pp. 93–107.

[21] Inmon, W.H., 1992. Building the Data Warehouse. QED Information Sciences, Inc., Wellesley, MA, USA.

[22] Issa, S., Paris, P., Hamdi, F., Cherfi, S.S.S., 2019. Revealing the Conceptual Schema of RDF Datasets, in: 31st International Conference on Advanced Information Systems Engineering, CAiSE,, Italy. pp. 1–15.

[23] Jovanovikj, I., Narasimhan, V., Engels, G., Sauer, S., 2018. Context-specific quality evaluation of test cases, in: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, Funchal, Madeira, Portugal.

[24] Kim, M., Zimmermann, T., DeLine, R., Begel, A., 2018. Data scientists in software teams: state of the art and challenges, in: Proceedings of the 40th International Conference on Software Engineering, ICSE, Gothenburg, Sweden. p. 585.

[25] LaValle, S., Lesser, E., Shockley, R., Hopkins, M.S., Kruschwitz, N., 2011. Big data, analytics and the path from insights to value. MIT sloan

management review Journal 52, 21.
[26] Majchrzak, T.A., Jansen, T., Kuchen, H., 2011. Efficiency evaluation of open source ETL tools, in: Proceedings of the 2011 ACM Symposium on Applied Computing SAC, TaiChung, Taiwan. pp. 287–294.
[27] Rahm, E., Do, H.H., 2000. Data cleaning: Problems and current approaches. Journal of IEEE Data Engineering Bulletin 23, 3–13.
[28] Redman, T.C., 1998. The impact of poor data quality on the typical enterprise. Communications of the ACM Journal 41, 79–82.
[29] Sadiq, S., Indulska, M., 2017. Open data: Quality over quantity. International Journal of Information Management IJIM 37, 150–154.
[30] Salem, A.B., Boufares, F., Correia, S., 2014. Semantic Recognition of a Data Structure in Big-Data. Journal of Computer and Communications JCC 2, 93–102.
[31] Theodorou, V., Abelló, A., Lehner, W., Thiele, M., 2016. Quality measures for ETL processes: from goals to implementation. Journal of Concurrency and Computation: Practice and Experience CCPE 28, 3969–3993.
[32] Theodorou, V., Jovanovic, P., Abelló, A., Nakuçi, E., 2017. Data generator for evaluating ETL process quality. Journal of Information Systems JIS 63, 80–100.
[33] Thota, S., 2017. Big Data Quality. Springer International Publishing, Cham. pp. 1–5.
[34] Vassiliadis, P., 2009. A survey of extract-transform-load technology. International Journal of Data Warehousing and Mining IJDWM 5, 1–27.
[35] Wang, R.Y., Strong, D.M., 1996. Beyond Accuracy : What Data Quality Means to Data Consumers. Journal of Management Information Systems JMIS 12, 5–33.
[36] Warth, J., Kaiser, G., Kügler, M., 2011. The impact of data quality and analytical capabilities on planning performance: insights from the automotive industry, in: Proceedings of the Wirtschaftsinformatik, Zurich. p. 87.
[37] Watson, H.J., 2009. Business intelligence: Past, present and future, in: Proceedings of the 15th Americas Conference on Information Systems AMCIS, San Francisco, California, USA. p. 153.
[38] Watts, S., Shankaranarayanan, G., Even, A., 2009. Data quality assessment in context: A cognitive perspective. Journal of Decision Support Systems DSS 48, 202–211.
[39] Yang, Q., Ge, M., Helfert, M., 2017. Guidelines of Data Quality Issues for Data Integration in the Context of the TPC-DI Benchmark, in: Proceedings of the 19th International Conference on Enterprise Information Systems, Portugal. pp. 135–144.