Guidelines of Data Quality Issues for Data Integration in the Context of the TPC-DI Benchmark

Qishan Yang[1], Mouzhi Ge[2] and Markus Helfert[1]
[1]*Insight Centre for Data Analytics, School of Computing, Dublin City University, Dublin, Ireland*
[2]*Department of Computing, Main University, MySecondTown, MyCountry*
qishan.yang@insight-centre.org, markus.helfert@dcu.ie,

Keywords: Data Quality, Data Integration, TPC-DI Benchmark, ETL

Abstract: Nowadays, many business intelligence or master data management initiatives are based on regular data integration, since data integration intends to extract and combine a variety of data sources, it is thus considered as a prerequisite for data analytics and management. More recently, TPC-DI is proposed as an industry benchmark for data integration. It is designed to benchmark the data integration and serve as a standardisation to evaluate the ETL performance. There are a variety of data quality problems such as multi-meaning attributes and inconsistent data schemas in source data, which will not only cause problems for the data integration process but also affect further data mining or data analytics. This paper has summarised typical data quality problems in the data integration and adapted the traditional data quality dimensions to classify those data quality problems. We found that data completeness, timeliness and consistency are critical for data quality management in data integration, and data consistency should be further defined in the pragmatic level. In order to prevent typical data quality problems and proactively manage data quality in ETL, we proposed a set of practical guidelines for researchers and practitioners to conduct data quality management in data integration.

1 INTRODUCTION

The data warehouse, as the organizations' data repository, is a subject-oriented, integrated, non-volatile and time-variant collection of data in support of management's decision (Inmon et al., 2010). The links and relationships among the Extract-Transform-Load (ETL), data warehouse and data quality were denoted by Kimball and Caserta (2011): ETL systems extract data from the source data, enforce data quality and consistency standards, and conform data, which enable the separate sources to be used together and finally deliver data in a data warehouse with the presentation-ready format.

Recently, a more comprehensive acronym DI (data integration) replaced the ETL. The process of the ETL can be described by DI which extracts and combines data from source data with a variety of formats, transforms the data into a unified data model representation and populates it into a data repository (Poess et al, 2014).

When building a data warehouse, ETL tools are the bridge for the data migration from data sources to destinations. Even though, it is invisible to end users and a black room activity, it could cost 70 percent of the resources needed for the data warehousing implementation and maintenance (Kimball and Caserta, 2011). Data integration systems manipulate and examine data streams to avoid rubbish data in for a data warehouse and rubbish out for decision-making or presentation systems. Hence, DI benchmark plays an vital role to evaluate ETL tools when there are several ETL candidates to choose. It could also provide data and a schema to benchmark ETL tools and build a ETL evaluation-oriented data warehouse respectively.

The TPC-DI[1] is designed as the first benchmark to evaluate Data Integration systems(Poess et al., 2014). The data used in the TPC-DI benchmark for testing and data warehouse populating is generated by a (fictitious) brokerage firm's operating system along with other sources of data. This benchmark

[1] http://www.tpc.org/tpcdi/

also designs the source and destination data models, data transformations and implementation rules (TPC, 2016).

Data quality issues appear frequently in the stage of the data integration when ETL tools extract data from resources, migrate and populate data into data repositories. Hence, data quality is an important aspect in the data integration process (Kimball and Caserta, 2011). Data quality has become a critical concern to the success of organisations. Numerous business initiatives have been delayed or even cancelled, citing poor-quality data as the main reason. Previous research has indicated that understanding the effects of data quality is critical to the success of organisations (Ge et al. 2011). A high quality of data provides the foundation for the data integration.

Most initial data quality frameworks have considered all the data quality dimensions are equally important (Knight and Burn, 2005). More recently, as Fehrenbacher and Helfert (2012) stated, it is necessary to prioritise certain data quality dimensions for data management. However, as far as we know, there is not yet work to prioritise data quality diemensions in ETL. Furthermore, there is limited research in guiding the data quality management in the data integration process.

Therefore, in this paper we intend to find out which data quality dimensions are crucial to data integration and also attempt to derive the guidelines for proactive data quality management in data integration. The contribution of this paper are two folds, first, we found that some typical data quality problems exist in data integration process. We have specified those data quality problems and related them to different data quality dimensions. It can be seen that certain data quality diemsnions need to be further refined, and more dimenions towards operational squence and data uniqueness should be used in the data quality management in ETL. On the other hand, in order to proactively manage data quality in data integration, we have derived a set of data quality guidelines that can be used to avoid data quality pitfalls and problems when integrating data and using the TPC-DI Benchmark.

The remainder of the paper is organised as follows. Section 2 reviewes the related work of data quality and data integration. Section 3 describes the research methodology used to conduct our research. Then in Section 4 we list the data quality problems in data integration process and classify those data quality problems into different data quality dimensions in section 5. Section 6 describes the guidelines for data quality management in data integration. Finally Section 7 concludes the paper and outlines the future research.

2 RELATED WORK

In order to manage data quality, Wang (1998) proposed the Total Data Quality Management (TDQM) model to deliver high quality information products. This model consists of four continuous phases: define, measure, analyse and improve, in which the measurement phase is critical, because one cannot manage information quality without having measured it effectively and meaningfully (Batini and Scannapieco 2016). In order to measure data quality, data quality dimensions must be determined. To this end, Wang and Strong (1996) used an exploratory factor analysis to derive 15 data quality dimensions, which are widely accepted in the following data quality research. Based on the 15 proposed dimensions, data quality assessment has been applied in different domains such as Healthcare (Warwicka et al., 2015), Supply Chain Management (Ge and Helfert, 2013), and Smart City Applications (Helfert and Ge, 2016).

Among the application domains, DI or ETL systems have been emerging as an important field that requires data quality management. The goal of the data integration system denoted by Doan et al. (2012) is decreasing the effort of users to acquire high-quality answers from a data integration system. They also defined a data warehouse in two tasks: (1) implementing the centralised database schema and physical design, (2) defining a batch of ETL operations. Hence, the DI or ETL system is the groundwork of the data warehousing in order to provide synthesized, consistent and accurate data. The ETL system manages some procedures specifically in (1) revising or removing mistakes and missing data, (2) offering confident documented measures in data, (3) safekeeping the captured data flow of transactions, (4) calibrating and integrating multiple sources data to be leveraged collaboratively, (5) structuring data to be usable by end-user tools (Kimball and Caserta, 2011). It is not only just extracting data from source systems, but also as a combination of traffic policemen and garages for the motorway of data flows in the data warehousing architecture.

Due to the importance of data quality management in ETL systems, previous research has been conducted to study the data quality problems in ETL systems. Singh and Singh (2010) attempted to tabulate possible data quality issues appearing in the

process of the data warehousing (the data source, data integration and data profiling, data staging, ETL and database schema). In this research, there were totally 117 data quality problems demonstrated in four tables for each data warehousing phases respectively. Nearly half of them (52) data quality flaws were contributed from the data sources stage, 36 issues were listed at the stage of ETL tools, and rest of them occupied 29 data quality problems. By reviewing the previous research, we found that there is lack of clearly defining the data quality problems and matching the data quality problems to data quality dimensions. Moreover, as far as we know, there is still no study that focuses on the data quality problems in the ETL process that aligns with the TPC-DI benchmark.

Before TPC-DI, there were some self-defined measurements to benchmark ETL systems, such as DWEB (Darmont et al., 2005) and Efficiency Evaluation of Open Source ETL Tools (Majchrzak et al. 2011). However, there was a lack of industry standardised ETL benchmarks which can be used to evaluate performances of ETL tools (Wyatt et al., 2009). The TPC-DI was the first industry benchmark to fill this gap regarding ETL evaluations (Poess et al., 2014). The TPC-DI benchmark was released by the Transaction Processing Performance Council (TPC) which is a non-profit corporation founded to define transaction processing and database benchmarks. This standardised measurement is characterised by (1) operating and populating large volumes of data, (2) multiple-sources data sets and a variety of different data formats, (3) manipulations in fact and dimensional tables' creation and maintenance, (4) a myriad of transformations incorporating data validation, key lookups, conditional logic, data type conversions, complex aggregation operations, etc., (5) historical and incremental Data Warehousing population loadings, (6) guaranteeing trustable and correct data results in integration processes under consistency requirements. It also provides a standard specification for the TPC-DI benchmark, in which 14 clauses have been given to deeply explain data sources, data warehousing schema, transformations, description of the system under test, execution rules & metrics, pricing etc. (TPC-DI, 2016). The code for data sets generation can be downloaded and executed under JDK. The data set size can be controlled by configuring the scale factor parameter. There are three batches of data sets, the Batch 1 is for the historical loading, the Batch 2 and 3 are aimed at incremental loadings.

Poess et al. (2014) summarised and explained the components of the TPC-DI including the source and target data models, characteristics and technical details for the generation of the data sets, the transformations of the DI workload, the execution rules, metric and a performance study. The TPC-DI source data came from five different data sources, which needed to be integrated into a decision support system. The data warehousing architecture and workflow were pictured hierarchically and divided into the SUT (system under test) and out of SUT parts. The SUT part should be benchmarked, while the out of SUT should be ignored in the process of the evaluation. The relationships and structure of fact, dimension and reference tables were depicted to better demonstrate the target schema, which would be useful in processes of constructing and populating the data warehouse.

Since benchmarking is critical for data integration (Vassiliadis, 2009) and TPC-DI is the first industrial standard benchmark for data integration (Poess et al., 2014), it is thus valuable to study how to manage data quality in data integration that is aligned with TPC-DI benchmark. Therefore, based on the previous research we have not only identified the data quality problems in the TPC-DI context, but also classified those problems to data quality dimensions, which could be used for data quality management.

3 RESEARCH METHODOLOGY AND SCENARIO

In this section, we describe the data integration process that is aligned with the TPC-DI benchmark. Along with this process, we present a typical scenario herein for the data integration. We frame our research in this scenario and derive guidelines accordingly.

The data integration process with the TPC-DI benchmark usually begins from the source data files generated by DIGen which is built on top of the Parallel Data Generation Framework (PDGF). The capabilities of the PDGF are extended to create data sets accompanied by the specific characteristics required by this benchmark. The DIGen is required to be executed under Java environment and the PDGF needs to be placed in the same directory (TPC-DI, 2106). After the data sources are generated by the DIGen, the data will be delivered into the Data Staging Area. This process is just the migration of the source data from outside to SUT (system

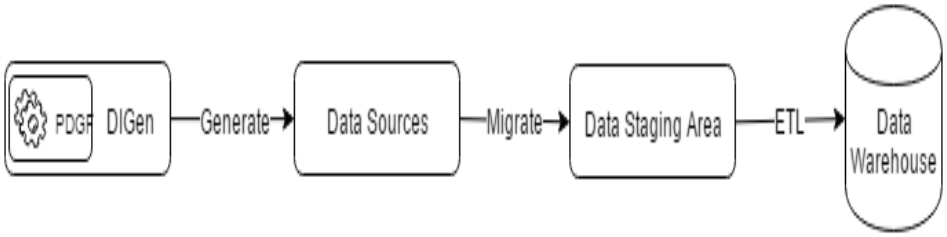

Figure 1: The outline of the process in the data warehousing architecture

under test) and no data cleansing operations. In the Data Staging Area, discovered data quality issues need to be addressed and the data quality management is conducted. Afterwards the ETL will be carried out to import the data to the data warehouse. The outline of the data integration workflow is depicted in figure 1.

In practice, it is common to extract source data firstly into flat files rather than transport data from data resources to data warehouses directly. It is sometimes necessary to obtain or purchase external data from outside free source data or third-party companies. In this case, a retail brokerage data warehouse is built using source data provided by TPC-DI. During this process, some data quality issues appeared in the source data, since the data was collected from internal and external data resources.

In our scenario, the data is aggregated from five sources, which are the Trading Database, Human Resource Database, Customer Prospect List, Financial Newswire and Customer Management System. In the data warehouse, there are some tables which need to be emphasised because they are involved herein as the data quality management examples. The DimCustomer dimension table stores customer records and DimAccount dimension table archives customers' account details. A new customer must accompany a new account, but existed customers can open more than one account. In some scenarios, these two tables need to be looked up.

When analysing the key customers or a quarter or annual trades made by customers via their accounts, we need to obtain the records from DimTrade table, and join corresponding entities from DimCustomer and DimAccount tables. The DimCompany table contains companies' ID, name, CEO, address etc. The DimSecurity table incorporates securities issued by companies. The Financial table gathers all the financial data of companies. All data for these three tables is provided by the FINWIRE files. When reviewing the market history or rating the companies with finance, the Fact MarketHistory table would be retrieved, and the DimCompany, DimSecurity and Financial tables would be looked up.

4 DATA QUALITY PROBLEMS IN DATA INTEGRATION

In this section, we describe the data quality problems investigated in the data sets provided by TPC-DI when doing the data integration process in this case. For each type of data quality problem, we define the data quality problem and provide typical examples to describe the data quality problems based on our scenario. Afterwards, we also classify the data quality problems into different data quality dimensions. Thus, we are able to identify which data quality dimensions are important for data quality management in data integration.

4.1 Missing Values

There are mainly two types of missing value problems in this data integration process. First, the data in one field appears to be null or empty, we define this type of missing value as *direct incompleteness*, which means this can be directly detected by rule-based query. On the other hand, the data can be missing because of the data operations such as data update. We define this type of missing value as *indirect incompleteness*. We describe the two types of missing value in details as follows.

4.1.1 The Missing Value in a Field

The Missing Value in a field indicates there is no non-null requirement or no compulsory value needed in some specific fields in a table. In our scenario, the DimCompany table's data is obtained from FINWIRE files, some values are missing in the field of the FoundingDate which shows when a

company has been created with the granularity of the date.

Even this field can be empty in the DimCompany table, but the missing values would influence the further Data Mining or data analysis jobs (e.g. the company reputation assessment). Even through the DimCompany table has a field named Sparting for standard & poor company's rating, but it would be revised associated with other attribute of values (e.g. FoundingDate), so in this situation, the value of the FoundingDate attribute might be considered to re-rate the value in the Sparting attribute.

4.1.2 The Updating Record with Missing Values

When updating a record, only new values are given to revise the old values in the record, other fields which are unnecessary to update are not provided in updating records. As a typical feature in the data warehouse, the update is not directly carried out in the record, instead, the data warehouse will maintain and mark this record as a legacy record and create a new record for the updated values.

In our scenario, in the process of the DimCustomer update, a record may only provide customerID, address or phone values to update, the rest fields are empty. The customerID is the Customer identifier to uniquely identify a certain customer, which is the primary key in the Customer table. According to TPC-DI and the dimension tables' characteristic of the data warehouse, when updating the record, the new fresh records will be inserted and the legacy records will still be maintained rather than be deleted. Moreover, the fields for updating in the records may be disparate as some records only need to update address, while some only need to update email etc. The generalised samples from the TPC-DI source data are tabulated below:

Table 1: The updating records with missing values.

Customer ID	Address	Email	Action Type
956	XXX	X@X.X	New
956	NULL	Y@Y.Y	Update
956	YYY	NULL	Update

The updating records could not be inserted into the dimension tables directly. Errors may be thrown by a database system because there is a violation to insert a null or empty value into non-null-allowed fields.

4.2 The Conflict of Entities

In this paper an entity is defined an object which is stored in dimension table as a record. The reason why we differentiate entity and record is that one record may contain different entities, and sometimes a record is an entity. The conflicts of entities mean that there are more than one valid or active record with the same identifier in a table. The records in tables need to agree with each other and no conflicts.

In our scenario, when we are inserting a record in the DimSecurity table, a lookup needs be performed to check whether the same ID already exists, if existed, the IsCurrent field of old record should be modified to false firstly, and then the following inserting operation continues. However, it is typical to use a batch to insert and update a list of records. In order to speed up the process, several threads may carry out the inserting and updating operations in parallel. If inserting and updating for a certain entity in flat files are very close, updating this entity could be executed before inserting the record. Thus, the lookup job would return not found and the old record's IsCurrent field is still true.

The situation above appeared in our experiment when loading data with big cache. the old record's status would be still valid all the time even it has already been updated. If this case is ignored or solved improperly, there could be more than one entity which have the same identifier and active status but different surrogate keys. When querying this kind of entities, which are current or valid, more than one entity would be given with the same entity identifier because of the conflict.

4.3 Format Incompatibility

This issue is very frequently appearing for the Date format in data resources. The Data format conflicts are mainly triggered by the inconsistent styles between the data resource and data warehouse.

In our scenario, in some dimension tables of the data source, the field of EffectiveDate is the beginning of effective date range of a certain record. The date retrieved from source data is a String with the format of the YYYY-MM-DDTHH24:MI:SS which contains date and time split by the capital T. Using a data warehouse in the Oracle database system as an example, the date format is DD-Mon-YY HH.MI.SS.000000000 AM/PM which has different date and time formats compared with the formats in source date. Two EffectiveDate samples from source data and the Oracle data warehouse are given in table 2.

Table 2: The samples of format incompatibility

Date Format	Place
2007-07-07T04:28:56	In the data resource
07-JUL-07 04.28.56.000000000 AM	In the data warehouse

If the original data with the date format in the data resource is inserted into the data warehouse without format transformations, the error would be thrown as the format violation. Therefore, the original date values need to be reformatted to match the data warehouse date style.

4.4 Multi-Resource or Mixed Records

In the raw data resource, a record may contain more than one table's entities. The entities in this record normally have referential or dependent relationships.

The number of entities in the raw data record depends on the planned data operations. For example, in the CustomerMgmt.xml, a record may contain two dimension tables' entities (DimCustomer and DimAccount tables). An account must belong to a certain customer, while a customer could have more than one account (One-to-Many Relationship). For each record, there is a planned operation, named as ActionType in Table 3. When we insert or update an account, we need to know this account belongs to which customer, thus this record contains two entities, which are customer and account. On the other hand, when we only update the customer information, the ActionType is filled with "UPDCUST" which means Update Customer. In this case, the record only contains one entity. In practice, there might be more entities in one record.

As such, when we carry on the data operations with raw data sources, we could either firstly differentiate the entities and extract the data operation or firstly extract the data operation and then base on the data operation to differentiate the entities. We found that it is time-consuming to first differentiate the entities, since the data operation may not use all the differentiated entities.

4.5 Multi-Table Files

In the data resource, there are some files that contain more than one table's records. This situation may happen when records in the tables are collected from the one system.

In our scenario, one file may contain three tables' data: CMP, SEC and FIN. The CMP records are related to DimCompany table; the SEC belongs to DimSecurty table; the FIN denotes to the Financial table. The three records in Table 4 come from the data source.

Based on the record type, the data stream extracted from this data source file is divided into several branches. Each branch may have sub-branches for different purposes such as status can be further split into different sub-branches (ACTV and INAC). Then there are several branches and sub-branches need to be considered in the process of loading data into (ACTV and INAC). If there are dependencies among the tables, the sequence of loading the data into table needs to be refined as some table may depend on other tables in terms of foreign keys. If other tables are not loaded, then there could trigger an error that the foreign keys are not found.

4.6 Multi-Meaning Attributes

In the data source, an attribute or a field may allow to contain different types of data which could have different meanings, while it could be difficult to avoid ambiguous and inseparable identifications.

Table 3: The samples in mixed records

Customer ID	Account ID	Action Type	Other Customer Info.	Other Account Info.
1	1	New	...	...
1	Null	UPDCUST	...	Null
1	1	UPDACCT	...	Null

Table 4: The samples in multi-table files

Posting date & time	Record Type	Status	Other Information
19670401-065923	FIN	NULL	Other Financial Information
19670425-114814	SEC	ACTV	Other Security Information
19670425-083141	CMP	ACTV	Other Company Information

Table 5: The samples of multi-meaning attributes

Posting date & time	Record Type	CoNameOrCIK	…
19670401-065923	FIN	1836200100000000056	...
19670403-194201	FIN	501026396GKXARCFbFebKiAlLUJXKJgRjmqXdA QcnJFJAKTzRouxMxMVkXQMjtVZu	...

In our scenario, in the data resource files there is a field named CoNameOrCIK that can carry the company identification code (10 chars) or company name (60 chars). In table 5, the first row is using a company identification code and the second one is using a company name.

In the Financial table, there is an attribute called SK_CompanyID which is the primary key of DimCompany as well as foreign key of Financial. Thus, when we insert the two records into the Financial table, from the data source, We could use either the company identification code or company name to look up the DimCompany table to find the primary key and then insert it into the Financial table as a foreign key.

However, in practice, the different type data can be very similar but have different meanings. In our example, the company identification code and company name could be very similar and hard to differentiate. If the program cannot differentiate the data types, there will be a "not found" error that means we are using the wrong data to locate the primary key.

5. CLASSIFY DATA QUALITY PROBLEMS INTO DQ MODEL

In order to facilitate the data quality management in data integration, we have classified the data quality problems investigated in this experiment into the classic data quality dimensions proposed by Wang and Strong (1996). The last two data quality problems are not totally fitting into the proposed data quality dimensions and we have proposed new dimensions for the data quality problems.

Table 6: Data quality dimensions in data integration (new data quality dimensions in ETL are marked with *)

Data quality dimension	Data quality problem
Completeness	Missing Value
Timeliness	Conflict of Entities
Consistency	Format Incompatibility
Operational Sequence*	Multi-Resource or Mixed Records
	Multi-Table Files
Uniqueness*	Multi-Meaning Attributes

In the context of data integration, we could see that not all the data quality dimensions are equally important. This has been confirmed in other data quality studies such as Fehrenbacher and Helfert (2012). For data quality management in ETL, we propose to initially focus on the dimensions of completeness, timeliness and consistency. This small set of dimensions not only point out the key focus of data quality management in data integration but also provide a foundation for data cleansing in data integration.

Moreover, some data quality dimensions need to be further refined. For example, representational consistency in data integration is not enough. We need to align the definitions of the data rather than only align the names. Therefore the consistency can be further refined into syntactic, pragmatic and semantic levels.

Accuracy is always considered as the most important data quality dimensions in data quality management. However, in the data integration, it is usually lack of the ground truth for the data. Therefore, wrong value is not included in our data quality problems. As an initial step in data quality management, we recommend to focus on the tangible set of data quality dimensions.

Not all the data quality problems can be classified into classic data quality dimensions, especially the problems about the sequence of the data operations. A correct sequence of data operation can increase the process efficiency and avoid data quality errors. For example, we could use the different type of operations to determine which entities are involved, or use table dependency to define the sequence of loading the data.

Furthermore, as Dakrory et al. (2015) has stated, uniqueness is one of the important data quality dimensions in ETL. We also found that apart from the classic data quality dimensions, data uniqueness is a critical indicator to differ the data meaning in order to avoid possible data ambiguity.

6. GUIDELINES FOR DATA QUALITY MANAGEMENT

In order to prevent the data quality problems in ETL and proactively manage data quality, we propose the following guidelines to help researchers and

practitioners to avoid data quality pitfalls and guide effective data quality management process. Specifically, guideline 1 and 2 tackle the missing value problems; Guideline 3 can be used to prevent the entity conflicts; Guideline 4 deals with the format incompatibility; Guideline 5 is for optimising mixed records and multi-table files in ETL and guideline 6 intends to solve the problem of multi-meaning attributes.

6.1 Guideline 1

In order to manage the possible effects of missing values after ETL, one can use business logic to derive the field dependency, and then pay attention especially to the fields that are involved in the field dependency and meanwhile allow null or empty values.

After we have finished the ETL, there can be certain fields that allow null or empty value in the data warehouse. Those fields may not cause errors in the ETL process but when those fields are used in the data analytics or some business operations, this type of field may play as an independent variable and can be used to determine other fields or values. It will then cause a problem because of the missing value.

6.2 Guideline 2

In the data quality management for ETL, the dimension of completeness should be further refined, since there can be direct incompleteness such as missing value in the record or indirect incompleteness that are caused by data operations.

Completeness is one of the well-known dimensions in data quality management. Managing data completeness is especially important during ETL, since it is usually a straightforward problem one can foresee, whereas in the meantime there might be certain incompleteness pitfall that people will overlook. As the example given in Section 4.1.2, the new data for updating and the original data to be updated are both complete. Only when carrying out the update operation, the updated records can turn to be incomplete without lookup. Therefore, to deal with the indirect incompleteness caused by update, it is necessary to use lookup to get the values that do not need to be updated.

6.3 Guideline 3

During ETL, when insert and update records appear together in the batch operation, the sequence of data operations in the batch needs to be designed to avoid entity conflicts.

In the ETL, batch operations are typically used to perform the data CRUD operations. In order to accelerate this process, in practice, distributed operations are usually conducted in parallel to process the data. Thus, for the same entity, it is necessary to avoid for example update or delete before the insert operation. One of the best practices is to separate the CRUD operations into different batches. Inside the separated batch, one can use the parallel operations.

6.4 Guideline 4

For ETL, assuring format consistency in the syntactic (representational) level is not enough. Data format consistency between data source and data warehouse should be aligned in a pragmatic level.

Data format consistency cannot be only confirmed by the format name. With the same data format name (syntactic level), there might be different real usages or different definitions (pragmatic level) for the same data format name. One of the prevalent format inconsistency is the Date format unconformity. Thus before carrying out the ETL, practitioners should especially look into what certain format means and whether the definitions and data types of the format are aligned between data source and data warehouse.

6.5 Guideline 5

Optimizing the sequence of data operations can increase the efficiency of the ETL process and avoid data quality problems.

In the ETL process, the CRUD data operations can be mixed together with the data entities. We recommend firstly extracting the data operation and based on the data operation to differentiate the data entities. In this way, we can avoid to look up the entities that are not used in the data operation. This will largely increase the efficiency when many entities are mixed in one record. Moreover, when we load the data source to various tables, optimising the loading sequence can avoid the errors triggered by table dependencies.

Table 7: The summary of guidelines

Data quality problems	Guidelines	Proposed proactive actions for data integration
Missing Values	Guideline 1	Field dependencies and indirect incompleteness caused by data operations should be specified.
	Guideline 2	
Conflict of Entities	Guideline 3	The sequence of data operations in the batch needs to be properly designed to avoid entity conflicts.
Format Incompatibility	Guideline 4	Representational and pragmatic consistency should be both examined before ETL.
Multi-Resource or Mixed Records	Guideline 5	The sequence of data operations can be optimised by firstly extracting the types of data operations and then differentiating the entities.
Multi-Table Files		
Multi-Meaning Attributes	Guideline 6	Data uniqueness should be included in the data quality management in data integration.

6.6 Guideline 6

Data uniqueness is an important dimension in data quality management. A complete logic should be used to identify the data.

In ETL, regular expressions are usually used to identify certain type of data. However, they are not always enough to differentiate the data, for example, when different letters or letter combinations have different meanings, it can be difficult for regular expressions to separate the meanings. Therefore, we recommend deriving a set of comprehensive conditional logic that can be used to categorise the data to their semantics.

To summarise the typical data quality problems in ETL and the corresponding proactive actions, we have used the Table 7 to provide an overview.

7 CONCLUSION

In this paper, we have investigated the data integration process in line with the TPC-DI Benchmark, which is the first and well known industry data integration benchmark. We have found a set of typical data quality problems that can occur in the data integration process. For each data quality problem, we have defined the problem and provided examples to demonstrate the problem trigger and possible effects. In order to facilitate the data quality management in data integration, we have classified the data quality problems into different data quality dimensions. This result indicates which data quality dimensions are important in data integration. These important dimensions can help researchers and practitioners to set the focus in data quality management and reduce the unnecessary cost and time. In addition, we found that operational sequence and data uniqueness are two critical data quality dimensions beyond the common data quality dimensions. Moreover, we have proposed a set of guidelines to avoid the data quality pitfalls and problems and construct proactive data quality management during data integration.

As future works, we plan to carry out the data improvement experiment to examine which data quality dimensions can be improved and how to coordinate the trade-offs between the data quality dimensions. The evaluation of this experiment needs to be enhanced as regards the effect of guidelines for data quality issues. Furthermore, the effects of data quality in the data integration process can be further studied. In addition, we also plan to further investigate the data quality problems in Big Data.

ACKNOWLEDGMENTS

This publication was supported by Science Foundation Ireland grant SFI/12/RC/2289 to Insight-Centre for Data Analytics (www.insight-centre.org).

REFERENCES

Batini, C. and Scannapieco, M., 2016. Erratum to: Data and Information Quality: Dimensions, Principles and Techniques. In Data and Information Quality. Springer International Publishing.

Dakrory, S.B., Mahmoud T.M., Ali A.A., 2015. Automated ETL Testing on the Data Quality of a Data Warehouse, International Journal of Computer Applications, 131(16) pp.9-16

Darmont, J., Boussaid, O. and Bentayeb, F., 2005. Dweb: A data warehouse engineering benchmark. In

proceedings of Data Warehousing and Knowledge Discovery 2005, volume 3589, pp. 85-94.

Doan, A., Halevy, A. and Ives, Z., 2012. Principles of data integration. Elsevier.

Fehrenbacher, D. and Helfert, M., 2012. Contextual factors influencing perceived importance and trade-offs of information quality, Communications of the Association for Information Systems. 30(8).

Ge, M., Helfert, M., and Jannach, D., 2011. Information Quality Assessment: Validating Measurement Dimensions and Process, in proceedings of 19th European Conference on Information Systems, Helsinki, Finland, 2011.

Ge, M. and Helfert, M., 2013. Impact of information quality on supply chain decisions, Journal of Computer Information Systems, 53 (4), 2013.

Helfert, M. and Ge, M., 2016. Big data quality-towards an explanation model in a smart city context. In proceedings of 21st International Conference on Information Quality, Ciudad Real, Spain, 2016.

Inmon, W.H., Strauss, D. and Neushloss, G., 2010. DW 2.0: The architecture for the next generation of data warehousing: The architecture for the next generation of data warehousing. Morgan Kaufmann.

Kimball, R. and Caserta, J., 2011. The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data. Publisher: Wiley.

Knight, S.A. and Burn, J.M., 2005. Developing a framework for assessing information quality on the World Wide Web. Informing Science: International Journal of an Emerging Transdiscipline, 8(5), pp.159-172.

Majchrzak, T.A., Jansen, T. and Kuchen, H., 2011. Efficiency evaluation of open source ETL tools. In Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 287-294).

Poess, M., Rabl, T., Jacobsen, H.A. and Caufield, B., 2014. TPC-DI: the first industry benchmark for data integration. In proceedings of the VLDB Endowment, 7(13), pp.1367-1378.

Singh, R. and Singh, K., 2010. A descriptive classification of causes of data quality problems in data warehousing. International Journal of Computer Science Issues, 7(3), pp.41-50.

Stvilia, B., Gasser, L., Twidale, M.B. and Smith, L.C., 2007. A framework for information quality assessment. Journal of the American society for information science and technology, 58(12), pp.1720-1733.

TPC-DS. (2016) TPC-DS Available at: http://www.tpc.org/tpcds/ [Accessed 16 December 2016].

TPC-DI. (2016) TPC-DI. Available at: http://www.tpc.org/tpcdi/ [Accessed 16 December 2016].

Vassiliadis, P. (2009). A Survey of Extract-Transform-Load Technology, International Journal of Data Warehousing & Data Mining, 5(3), 1-27, 2009

Warwicka, W., Johnsona, S., Bonda, J., Fletchera, G. and Kanellakisa, P., 2015. A framework to assess healthcare data quality. European Journal of Social & Behavioural Sciences, 13(2), pp.1730.

Wang, R.Y., 1998. A product perspective on total data quality management. Communications of the ACM, 41(2), pp.58-65.

Wang, R.Y. and Strong, D.M., 1996. Beyond accuracy: What data quality means to data consumers. Journal of management information systems, 12(4), pp.5-33.

Wyatt, L., Caufield, B. and Pol, D., 2009, August. Principles for an ETL Benchmark. In Technology Conference on Performance Evaluation and Benchmarking (pp. 183-198). Springer Berlin Heidelberg.