Data Quality Problems in TPC-DI Based Data Integration Processes

Chapter · July 2018

CITATIONS
2

READS
227

3 authors:

Qishan Yang
Dublin City University
4 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Markus Helfert
National University of Ireland, Maynooth
311 PUBLICATIONS **1,582** CITATIONS
SEE PROFILE

Mouzhi Ge
Deggendorf Institute of Technology
97 PUBLICATIONS **1,582** CITATIONS
SEE PROFILE

Data Quality Problems in TPC-DI Based Data Integration Processes

Qishan Yang[1(✉)], Mouzhi Ge[2], and Markus Helfert[1]

[1] Insight Centre for Data Analytics, Dublin City University, Dublin, Ireland
qishan.yang@insight-centre.org, markus.helfert@dcu.ie
[2] Faculty of Informatics, Masaryk University, Brno, Czech Republic
mouzhi.ge@muni.cz

Abstract. Many data driven organisations need to integrate data from multiple, distributed and heterogeneous resources for advanced data analysis. A data integration system is an essential component to collect data into a data warehouse or other data analytics systems. There are various alternatives of data integration systems which are created in-house or provided by vendors. Hence, it is necessary for an organisation to compare and benchmark them when choosing a suitable one to meet its requirements. Recently, the TPC-DI is proposed as the first industrial benchmark for evaluating data integration systems. When using this benchmark, we find some typical data quality problems in the TPC-DI data source such as multi-meaning attributes and inconsistent data schemas, which could delay or even fail the data integration process. This paper explains processes of this benchmark and summarises typical data quality problems identified in the TPC-DI data source. Furthermore, in order to prevent data quality problems and proactively manage data quality, we propose a set of practical guidelines for researchers and practitioners to conduct data quality management when using the TPC-DI benchmark.

Keywords: Data quality · Data integration · TPC-DI Benchmark ETL

1 Introduction

The data warehouse, as an organization's data repository, is a subject-oriented, integrated, non-volatile and time-variant collection of data in support of management decisions [10]. A data warehouse may need an Extract-Transform-Load (ETL) system to collect and integrate data. A ETL system extracts data from data sources, enforces data quality standards, and conforms data, which gathers the separate sources and finally delivers data in data warehouses with a presentation-ready and unified format [11]. Even though a ELT system is invisible to end users as a black box, it could cost 70% of the resources in the data warehousing implementation and maintenance [11]. The process of the ETL can

S. Hammoudi et al. (Eds.): ICEIS 2017, LNBIP 321, pp. 57–73, 2018.
https://doi.org/10.1007/978-3-319-93375-7_4

be described by data integration (DI) which extracts, transforms and populates data into a data repository [14]. When building a data warehouse, a DI system is the bridge for the data migration from data sources to the destination.

The TPC-DI[1] is designed as the first benchmark to evaluate DI systems [14]. This benchmark provides the source and destination data models, data transformations and implementation rules. It allows people to evaluate DI systems in order to choose a suitable one that meets their requirements. It can also be leveraged to assess the performance of a legacy DI system for further improvements.

Data quality problems appear frequently in the stage of DI when extracting, migrating and populating data into data repositories. Data quality is considered an important aspect that influences the DI process [11]. Previous research indicates that understanding the effects of data quality is critical to the success of organisations [6]. Numerous business initiatives have been delayed or even cancelled, citing poor-quality data as the main reason. Most initial data quality frameworks consider that data quality dimensions are equally important [12]. More recently, as [5] states, it is necessary to prioritise certain data quality dimensions for data management. However, as far as we know, there is limited research on prioritising data quality dimensions and guiding the data quality management in the DI process. As far as we know, there is still no study that focuses on the data quality problems aligning with the TPC-DI benchmark.

Therefore, in this paper, we intend to find out which data quality dimensions are crucial to DI and also attempt to derive the guidelines for proactive data quality management in DI. The contributions of this paper are shown in three parts. The TPC-DI processes are investigated based on the data flow from different data sources to a data warehouse. Then we demonstrate some typical data quality problems which should be considered in the DI process. We specify these data quality problems and classify them into different data quality dimensions. Finally, in order to proactively manage data quality in DI, we derive a set of data quality guidelines that can be used to avoid data quality pitfalls and problems when using the TPC-DI Benchmark.

The remainder of the paper is organised as follows. Section 2 reviews the related work about data quality and DI. The processes of the TPC-DI benchmark is explained in Sect. 3. Section 4 describes a scenario used to conduct our research. Then we investigate the data quality problems in the DI process in Sect. 5 and classify these problems into different data quality dimensions in Sect. 6. Section 7 proposes the guidelines for data quality management in DI. Finally, Sect. 8 concludes the paper and outlines the future research.

2 Related Work

In order to manage data quality, Wang [22] proposes the Total Data Quality Management (TDQM) model to deliver high-quality information products.

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

This model consists of four continuous phases: the definition, measurement, analysis and improvement. The measurement phase is critical as information quality cannot be managed without being measured effectively and meaningfully [1]. In order to measure data quality, data quality dimensions need to be determined. [9] investigates the data quality management in the process of the data warehousing aligned with a world-leading financial services company. A practically oriented concept is illustrated in order to manage the data quality in large data warehouse systems. It also shares experiences of developing a data quality strategy for data quality planning and controlling.

[16] lists possible data quality issues appearing in the different stages, such as the data source, DI and data profiling, data staging, ETL and database schema. 117 data quality problems are demonstrated. Nearly half of them (52) data quality flaws are contributed by the data source stage, 36 issues are derived from the stage of ETL tools, and rest occupies 29 data quality problems. Wang and Strong [23] use an exploratory factor analysis to derive 15 important data quality dimensions from initial 179 attributes, which are widely accepted in the following data quality research. Based on these proposed dimensions, data quality assessments are applied in different domains such as the Healthcare [21], Supply Chain Management [7], and Smart City Applications [8].

[4] declares the goal of the DI system which is to decrease the effort of users in acquiring high-quality answers. The DI 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, (5) structuring data for end-user tools [11]. Hence, the DI system is the foundational work of the data warehousing in order to provide synthesized, consistent and accurate data for further analysis.

Before the TPC-DI, there are some self-defined benchmarks, such as Efficiency Evaluation of Open Source ETL Tools [13] and the Data Warehouse Engineering Benchmark (DWEB) [3]. However, there is a lack of an industrial standardised ETL benchmark, which can be used to evaluate performances of ETL tools [24]. The TPC-DI is the first industrial benchmark to fill this gap regarding ETL evaluations [14]. It is released by the Transaction Processing Performance Council (TPC) which is a non-profit corporation founded to define transaction processing and database benchmarks.

Poess et al. [14] explain the components and characteristics of the TPC-DI, such as the source and target data models of a data warehouse, technical details for the generation of the data sets, the transformations of the DI workload, etc. The TPC-DI data comes from different data sources, which need to be integrated into a data warehouse. The data warehousing architecture and work flow are hierarchical and divided into the SUT (system under test) and out of SUT parts. The SUT part will be benchmarked, while the out of SUT is not covered in the process of evaluation. This standardised benchmark provides a standard specification for usage of the TPC-DI benchmark, in which 14 clauses are given to explain data sources, data warehousing schema, transformations,

description of the system under test, execution rules & metrics, pricing etc. [18]. The code for data set generation can be downloaded and executed in JDK environment. The data set size can be controlled by configuring the scale factor parameter.

3 TPC-DI Processes

Since benchmarking is critical for DI evaluation [20], it is thus valuable to study how TPC-DI benchmark works. In this section, this benchmark is investigated from the process flow perspective.

3.1 Data Generation

DI processes of the TPC-DI benchmark usually begin from the data source 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 with the specific characteristics required for this benchmark [18]. PDGF is implemented at the University of Passau, which is suitable for cloud scale data generation with highly parallel and very adaptable characteristics. This framework is configured using two XML files for data description and distribution, which facilitates the generation of different distributions of specified data sets. The implementation expends much effort on performance and extensibility. Hence, it is easily doable to generate new domains derived from PDGF [15]. This framework uses a peculiar methodology for the seed to exploit the pseudo random number generators in parallel. It keeps track of the random number sequences for each value in the data set, which gives this framework highly scalable ability on multi-core, multi-socket, and multi-node systems [14].

DIGen is a specific generator based on PDGF to create data sources and audit information for this benchmark. It is required to be executed in a Java environment and PDGF needs to be placed in the same directory [18]. There are some regulations for usage of DIGen according to the Standard Specification of the TPC-DI version 1.1.0 [19]: (1) The data source must be created using DIGen for this benchmark; (2) It is not allowed to modify DIGen; (3) The version of the specification and DIGen must match; (4) PDGF should be used; (5) Errors in a compliant DIGen version is deemed to be in compliance with the specification; (6) DIGen should be used to create the data source based on a minimum of Java SE 7; (7) Test sponsors need to ensure DIGen execution correctly in their environments; (8) The issue submission should contact the TPC administrator with the document including the exact issue and proposed fix.

3.2 Data Sources

According to [19], the data is comprised of five sources and a small number of reference files, which come from the online transaction processing (OLTP) database, Human Resource (HR) Database, Customer Prospect List, Financial Newswire

Table 1. The data sets.

System/Reference	File	Format	Historical	Incremental
OLTP	Account.txt	CDC		Y
	Customer.txt	CDC		Y
	Trade.txt	CDC	Y	Y
	TradeHistory.txt	CDC	Y	
	CashTransaction.txt	CDC	Y	Y
	HoldingHistory.txt	CDC	Y	Y
	DailyMarket.txt	CDC	Y	Y
	WatchItem.txt	CDC	Y	Y
HR DB	HR.csv	CSV	Y	
Represent List	Prospect.csv	CSV	Y	Y
FINWIRE	FINWIRE	Multi	Y	
Customer Management System	CustomerMgmt.xml	XML	Y	
Reference	Date.txt	DEL	Y	
	Time.txt	DEL	Y	
	Industry.txt	DEL	Y	
	StatusType.txt	DEL	Y	
	TaxRate.txt	DEL	Y	
	TradeType.txt	DEL	Y	

and Customer Management System. The OLTP database contains data for transactional information about securities market trading and the relevant entities. The HR Database manages data for the employees and their reporting hierarchy. The Prospect List is extracted daily from an external data provider. It includes customers of the brokerage and potential customers' name, contact and demographic information. The Financial Newswire contributes to the FINWIRE file for quarterly historical data feeding of the two dimension tables and a reference table: the DimCompany table, DimSecurity table and Financial table. The Customer Management System manipulates new and updated customer and account information. The reference file provides data or complementary information to support other tables. There are 18 files from these five sources and reference files, which are summarised and tabulated in Table 1 based on the TPC-DI standard specification [19]. After data is generated by DIGen, it will be sorted into three different directories: the Batch 1 for the historical load; the Batch2 for the Incremental Update 1; the Batch3 for the Incremental Update 2.

3.3 Data Warehouse

After data sources are generated, they are finally delivered into a data warehouse. This data warehouse is based on dimensional modeling with the ability

to efficiently reply to business questions [19]. The fact tables and dimension tables are the fundamental components of a data warehouse. In this research, six fact tables, seven dimension tables and five reference tables are created to build this warehouse. Based on the document [19], the main relationships of these tables in this data warehouse are depicted in Fig. 1. As can be seen, tables are linked by foreign keys and most links are built between dimension tables and fact tables. For instance, the FactWatches table has relationships with three dimension tables (DimCustomer, DimSecurity and DimDate). However, there

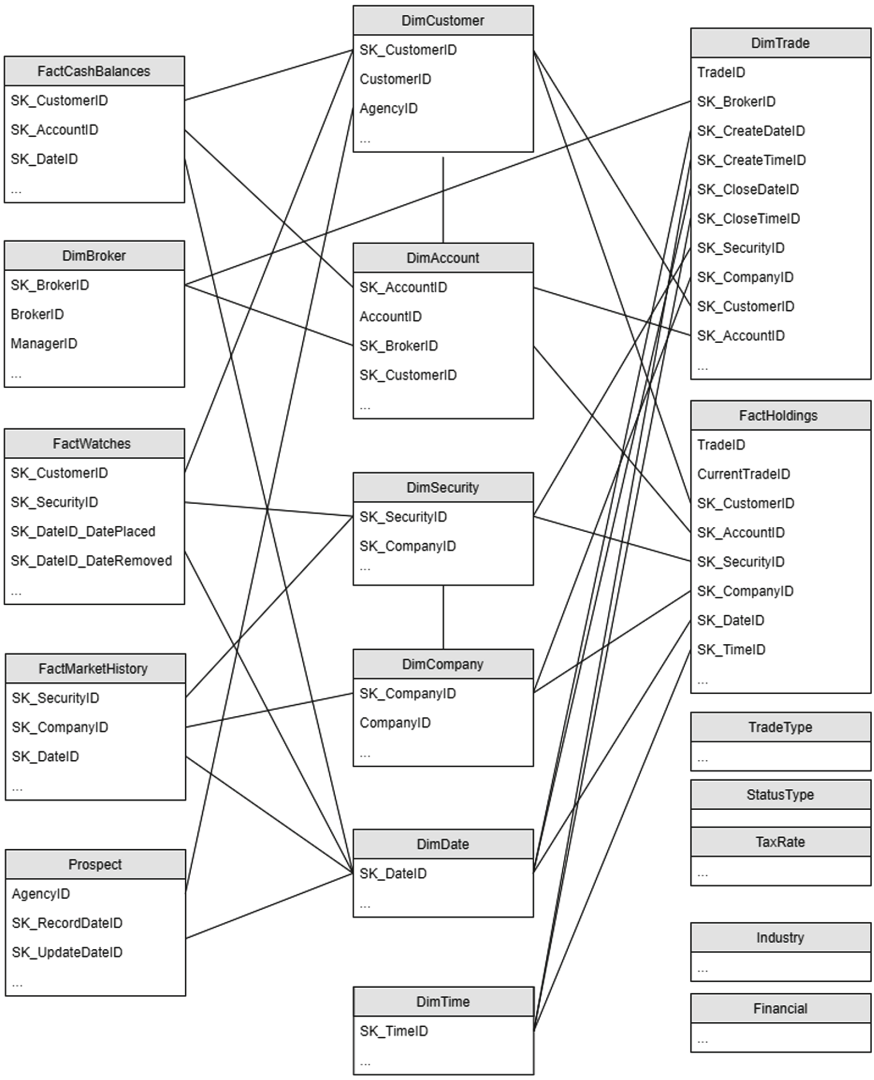

Fig. 1. The main relationships among tables.

are some links which appear among dimension tables, such as the link between the DimCustomer table and DimAccount table. Reference tables support other tables when more or detailed information is required. In addition, the DimTrade table can be seen as a dimension table or fact table depending on its functions in different situations.

3.4 Data Flow

The data flow in this research has several steps from the data generator to destination. After the data is generated by DIGen, it is delivered into the data staging area. This process is just the migration of data sources from outside to system under test (SUT) and no data transformations and cleansing operations are executed. After data is sent to the staging area, data quality issues should be solved before loading into the data warehouse. The data flow is depicted in Fig. 2. The data flow before the staging area is out of the testing scope, the rest of data flow is under the test for evaluations.

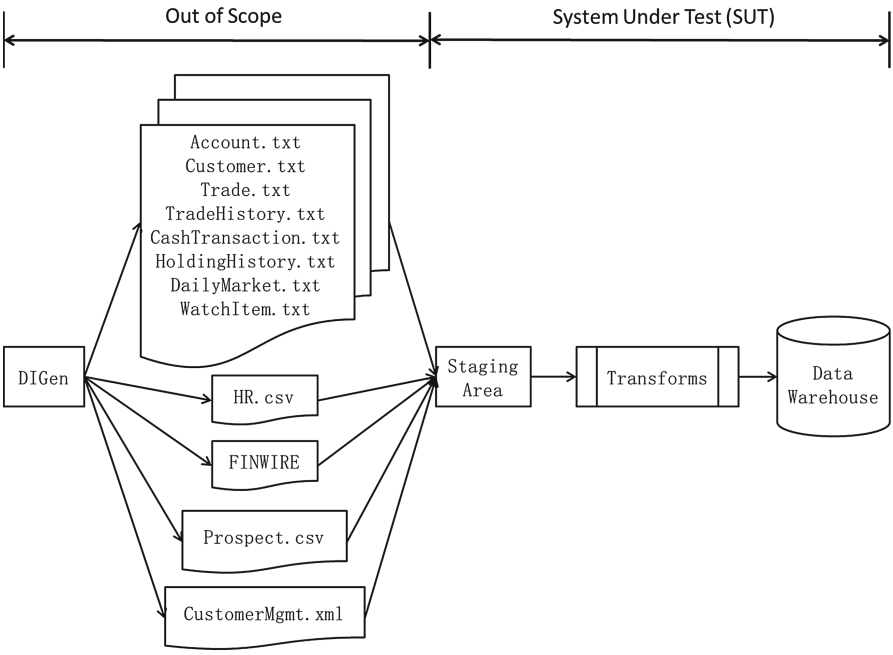

Fig. 2. The data flow.

4 Scenario

In order to investigate the data quality in DI aligned with the TPC-DI, a typical scenario is set up to frame our research. In practice, it is common to extract data firstly into flat files rather than transport data from data resources to a

data warehouse directly. It is sometimes necessary to obtain or purchase external data from outside free data sources or third-party companies. In this case, a retail brokerage data warehouse is built and fed by using the TPC-DI data. During this process, some data quality problems appear in the process of DI, which are explained concretely in the following sections.

In our scenario, some tables are emphasised herein, since they are involved as the data quality issue 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 existing customers can open more than one account. When analysing the key customers or trades via 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 companies' financial data. All data for these three tables is fed by the FINWIRE files. When reviewing the market history or rating the companies from a finance perspective, the MarketHistory table would be retrieved, and the DimCompany, DimSecurity and Financial tables would be looked up to acquire information if necessary.

5 Data Quality Problems in Data Integration

In this section, we describe and define the data quality problems in the TPC-DI data sets when conducting this scenario. Some typical examples are given to describe these problems. Afterwards, we classify these problems into different data quality dimensions. Thus, we are able to identify which data quality dimensions are important for data quality management in DI aligned with the TPC-DI.

5.1 Missing Values

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

The Missing Values in Fields. The Missing Values in fields indicate there is no non-null requirement or no compulsory values in some specific fields. In our scenario, the DimCompany table's data is obtained from FINWIRE files, some values are missing in the field of the FoundingDate which show the creation time of companies.

Even this field can be empty in the DimCompany table, but these missing values would influence the further data mining or data analysis (e.g. the company reputation assessment). The DimCompany table has a field named Sparting for standard & poor company's rating which would be influenced by values of the FoundingDate. So in this situation, if the missing values of the FoundingDate could be given or found, they might be leveraged to re-rate the value in the Sparting field.

The Updating Record with Missing Values. When updating some records, only new values are given to revise the old values, while other fields are missing because they are unnecessary to be updated. As a typical characteristic of the data warehouse, the update is not directly changing the old values of a record, instead, the data warehouse maintains and marks the old record as a legacy or inactive record and create a new record with a surrogate key for updating.

The records for updating may be nonuniform in the DimCustomer table. For example, some records only have new addresses, while some records only have emails, because they only need to update addresses or emails. A generalised example of these records is described in Table 2. Based on the characteristic of the data warehouse, when updating a record, a new record will be created and the legacy record will still be maintained rather than be deleted.

Table 2. The updating records with missing values.

Customer ID	Address	Email	Action type
56	XXX	X@X.X	New
56	YYY	NULL	Update
56	NULL	Z@Z.Z	Update

Updating these records could not be inserted into the dimension tables directly. Otherwise, errors may be thrown by a database system, because of inserting null or empty values into non-null-allowed fields.

5.2 The Conflict of Entities

In this paper, the definition of an entity is a record or object stored in a table. The reason why we differentiate the entity and record is that a record may contain several entities outside tables, while sometimes a record is an entity. The conflicts of entities mean that there is 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 avoid conflicts between them.

In our scenario, when inserting a record into the DimSecurity table, a lookup need to be conducted to check whether an ID already exists. If it is existed, the IsCurrent field of the old record should be modified to false or inactive firstly, and

then the new record is inserted into the table. However, in order to speed up the process, caches and several threads may be leveraged for creating and updating operations in parallel. If operations of creating and updating a certain entity are allocated in different caches or threads, the updating could be executed before the creating. The lookup would return the "not exist" ID for updating. This may result in more than one record's IsCurrent shows active or true. But only a record should be active or true for an ID.

The situation above appears in our experiment when using big caches and several threads to load data. Some records would be active or true all the time even they have already been updated. If this issue is ignored or resolved improperly, some entities may have the same ID and active status but different surrogate keys. Conflicts occur when querying these entities with IDs.

5.3 Format Incompatibility

This issue appears frequently in the date format. Date format conflicts are mainly triggered by inconsistent date styles between the data resource and data warehouse.

In our scenario, the field of EffectiveDate means the effective date of a certain record. The date retrieved from the data source is the string with the format "YYYY-MM-DDTHH24:MI:SS" which includes the date and time split by the capital T. If the data warehouse is built in the Oracle database system as an example, the date format is "DD-Mon-YY HH.MI.SS.000000000 AM/PM" which shows different date and time formats compared with the format in the data source. Two formats of an EffectiveDate example in the data source and the Oracle data warehouse are given in Table 3.

Table 3. Examples of format incompatibility.

Date format	Place
2007-05-08T07:21:56	The date format in the data resource
07-MAY-08 07.21.56.000000000 AM	The date format in the data warehouse

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

5.4 Multi-resource or Mixed Records

In raw data sources, a record may contain more than one table's entities. These entities in a record normally have referential or dependent relationships. 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 and a customer could have more than one account (One-to-Many Relationship). For each record, there is a field named a Action Type, which shows the purpose of this record. When we insert a record to create or update a new account, the value of the Action Type is "New" or "UPDACCT" respectively. This record includes two entities which contain customer and account's information. Whereas, when we only update the customer information, the value of Action Type is "UPDCUST". In this case, the record only contains one entity. The Table 4 illustrates these three kinds of records.

Table 4. Examples of mixed records.

Customer ID	Account ID	Action type	Other customer info.	Other account info.
137	165	New	...	...
137	Null	UPDCUST	...	Null
137	165	UPDACCT	...	Null

As such, when carrying out the data operations, there are two options: (1) differentiating the entities then identifying the purpose; (2) identifying the purposes then differentiating the entities if necessary. We find that the option 1 is slower compared to the option 2, since some records do not need to be differentiated, but all purposes of records need to be identified.

5.5 Multi-table Files

In raw data sources, some files contain more than one table's records. This situation may happen when records in tables are collected from the one system. In the TPC-DI data sets, a file may contain three tables' data: CMP, SEC and FIN. The CMP records are related to the DimCompany table; the SEC is for the DimSecurty table; the FIN feeds the Financial table. Table 5 gives three examples of these types.

Table 5. Examples in multi-table files.

Posting date and time	Record type	Status	Other information
19860502-082315	FIN	NULL	Other financial information
19760713-103826	SEC	ACTV	Other Security information
19850826-113217	CMP	ACTV	Other Company Information

Based on types of records, the data extracted from this data source file can be divided into several branches. Each branch may have sub-branches for different

purposes as they can be further split into different sub-branches (e.g. ACTV and INAC). Then there are several branches and sub-branches need to be considered in the process of loading data. If dependencies and links exist among these tables, the sequence of loading the data into tables needs to be prioritised as some tables may depend on other tables via foreign keys. If ignoring this sequence, errors would be triggered as the foreign keys are not found.

5.6 Multi-meaning Attributes

In data sources, an attribute or a field may allow containing different types of data which could have different meanings. It could be difficult to avoid ambiguities with improper differentiation.

In our scenario, a attribute named the CoNameOrCIK that may carry the company identification code (10 chars) or company name (60 chars). The Table 6 shows two records as examples from data sources. The first row uses a company identification code, while the second row uses a company name which may be encrypted. In the Financial table, there is an attribute called SK_CompanyID which is the primary key of the DimCompany table as well as the foreign key of the Financial table. Thus, when inserting a record into the Financial table, we could either use 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.

Table 6. Examples of multi-meaning attributes.

Posting date and time	Record type	CoNameOrCIK
19790911-082315	FIN	18362001000000123456
19830512-091241	FIN	501026396HBGSKDFbFe bKiJHFLSJIEFgRjmqXd AQcnYFGETDHzRouxMx JHURQIjtVZu

If the company identification code and company name are very similar and hard to be identified, the program could not differentiate the meaning and type of a value. Errors could occur when loading these misunderstood or incorrect values into tables.

6 Classify Data Quality Problems

In order to facilitate the data quality management in DI, we classify these data quality problems into the classic data quality dimensions proposed by Wang and Strong [23]. The last two data quality problems are not totally fitted into proposed data quality dimensions and we propose new dimensions for the data quality problems, which are marked with *. The details is tabulated in Table 7.

Table 7. Data quality dimensions in data integration [25].

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 DI, not all data quality dimensions are equally important for data quality management in ID. We propose initially focusing 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 DI but also provide a foundation for data cleansing in DI.

Moreover, some data quality dimensions need to be further refined. For example, representational consistency in DI 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 one of the most important data quality dimensions in data quality management. However, in DI, 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 focusing 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 can identify different types of operations or use table dependencies to define the sequence of loading the data.

Furthermore, as Dakrory et al. [2] state that uniqueness is one of the important data quality dimensions in DI. We find 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 ambiguities.

7 Guidelines for Data Quality Management

In order to prevent these data quality problems in DI 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. Specifically, guideline 1 and 2 tackle the missing value problems; Guideline 3 can be used to prevent entity conflicts; Guideline 4 deals with format incompatibility; Guideline 5 is for optimising mixed records and multi-table files in DI and Guideline 6 intends to solve the problem of multi-meaning attributes.

To summarise the typical data quality problems in DI and the corresponding proactive actions, Table 8 is provided as an overview.

7.1 Guideline 1

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

There are certain fields that allow null or empty values in the data warehouse. Those fields may not cause errors in DI processes. But when these fields are used in the data analytics or some business operations, they may play as an independent variable and can be used to determine other fields or values. The Missing values would then cause a problem.

7.2 Guideline 2

In the data quality management for DI, the dimension of completeness should be further refined, since there can be direct incompleteness such as the missing value in a record or indirect incompleteness such as the missing value in the update process.

Completeness is one of the well-known dimensions in data quality management. Managing data completeness is especially important during DI, since it is usually a straightforward problem which can be foreseen, whereas in the meantime there might be certain incompleteness pitfalls that people will overlook. As the example given in the Sect. 5.1, when carrying out the update operation, the updated records can turn out to be incomplete without a lookup. Therefore, to deal with the indirect incompleteness caused by the update, it is necessary to look up and find the values that do not need to be updated.

7.3 Guideline 3

When both types of inserting and updating records appear in batch operations, the sequence of data operations in a batch can avoid entity conflicts.

Batch operations are typically used to speed up the data creation, read, update and deletion (CRUD) operations. In practice, distributed operations are also usually conducted in parallel to expedite the processing of data. Thus, for an entity, it is necessary to avoid update or deletion before the insert operation. One of the best practices is to separate the CRUD operations into different batches and rank them. Parallel operations could be conducted inside a separated batch.

7.4 Guideline 4

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

Table 8. The summary of guidelines [25].

Data quality problems	Guidelines	Proposed proactive actions for data integration
Missing values	Guideline1 Guideline 2	Field dependencies and indirect incompleteness caused by data operations should be specified
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 Multi-Table files	Guideline 5	The sequence of data operations can be optimised by firstly extracting the types of data operations and then differentiating the entities
Multi-Meaning attributes	Guideline 6	Data uniqueness should be included in the data quality management in data integration

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

7.5 Guideline 5

Optimising the sequence of data operations can increase the efficiency of DI processes and avoid data quality problems.

In DI processes, data entities may be mixed together in a record. We recommend firstly to identify the purpose of this record, then separate the data entities if necessary. Moreover, when we load a data source with various tables, optimising the loading sequence can avoid the errors triggered by table dependencies.

7.6 Guideline 6

Data uniqueness is an important dimension in data quality management. Complete logic should be used to identify the data. In DI, regular expressions could be used to identify certain types of data. However, they are not always enough

to differentiate the data. For example, when different letters or letter combinations have different meanings, it could be difficult for regular expressions to identify the meanings. Therefore, we recommend deriving a set of comprehensive conditional logic that can be used to categorise their semantics.

8 Conclusion

This paper investigates processes of DI allied with the TPC-DI benchmark. There is a set of typical data quality problems that may occur in the DI processes when using the TPC-DI. We define these problems and provide examples to demonstrate problem triggers and possible effects. These problems are further classified based on traditional data quality dimensions, which can be used to indicate that what data quality dimensions are important in DI. These dimensions can help researchers and practitioners to set the focus on data quality management, and reduce the cost and time to identify data quality dimensions. In addition, we propose a set of guidelines that can be used to avoid data quality problems when using the TPC-DI benchmark.

In the future, we will conduct experiments to examine which data quality dimensions can be improved and how to coordinate the trade-offs between the data quality dimensions. The different DI scenarios should be taken into account for further verifying the utility of the guidelines. In addition, as data is exploding and many organisations are building data warehouses in the context of Big Data, we will investigate DI and data quality problems allied with the TPC-DI in Big Data.

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

References

1. Batini, C., Scannapieco, M.: Erratum to: data and information quality: dimensions, principles and techniques. Data and Information Quality. DSA, p. E1. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-24106-7_15
2. Dakrory, S.B., Mahmoud, T.M., Ali, A.A.: Automated ETL testing on the data quality of a data warehouse. Int. J. Comput. Appl. **131**(16), 9–16 (2015)
3. Darmont, J., Bentayeb, F., Boussaïd, O.: DWEB: a data warehouse engineering benchmark. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 85–94. Springer, Heidelberg (2005). https://doi.org/10.1007/11546849_9
4. Doan, A., Halevy, A., Ives, Z.: Principles of Data Integration. Elsevier, Amsterdam (2012)
5. Fehrenbacher, D., Helfert, M.: Contextual factors influencing perceived importance and trade-offs of information quality. Commun. Assoc. Inf. Syst. **30**(8) (2012)
6. Ge, M., Helfert, M., Jannach, D.: Information quality assessment: validating measurement dimensions and process. In: Proceedings of 19th European Conference on Information Systems, Helsinki, Finland (2011)
7. Ge, M., Helfert, M.: Impact of information quality on supply chain decisions. J. Comput. Inf. Syst. **53**(4), 59–67 (2013)

8. Helfert, M., Ge, M.: 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)
9. Helfert, M., Herrmann, C.: Introducing data quality management in data warehousing. In: Wang, R.Y. , Madnick, S.E., Pierce, E.M., Fisher, C.W. (eds.) Information Quality. Advances in Management Information Systems, vol. 1, pp. 135-150. M.E. Sharpe, Armonk, NY (2005)
10. Inmon, W.H., Strauss, D., Neushloss, G.: DW 2.0: The Architecture for the Next Generation of Data Warehousing: The Architecture for the Next Generation of Data Warehousing. Morgan Kaufmann, Massachusetts (2010)
11. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley, Hoboken (2011)
12. Knight, S.A., Burn, J.M.: Developing a framework for assessing information quality on the World Wide Web. Inf. Sci. Int. J. Emerg. Transdiscipl. **8**(5), 159–172 (2005)
13. Majchrzak, T.A., Jansen, T., Kuchen, H.: Efficiency evaluation of open source ETL tools. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 287–294 (2011)
14. Poess, M., Rabl, T., Jacobsen, H.A., Caufield, B.: TPC-DI: the first industry benchmark for data integration. Proc. VLDB Endow. **7**(13), 1367–1378 (2014)
15. Rabl, T., Frank, M., Sergieh, H.M., Kosch, H.: A data generator for cloud-scale benchmarking. In: Nambiar, R., Poess, M. (eds.) TPCTC 2010. LNCS, vol. 6417, pp. 41–56. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18206-8_4
16. Singh, R., Singh, K.: A descriptive classification of causes of data quality problems in data warehousing. Int. J. Comput. Sci. Issues **7**(3), 41–50 (2010)
17. Stvilia, B., Gasser, L., Twidale, M.B., Smith, L.C.: A framework for information quality assessment. J. Am. Soc. Inform. Sci. Technol. **58**(12), 1720–1733 (2007)
18. TPC-DI. http://www.tpc.org/tpcdi/. Accessed 16 Dec 2016
19. TPC Benchmark DI: TPC Benchmark DI Stand Specification Version 1.1.0. Transaction Processing Performance Council (2014)
20. Vassiliadis, P.: A survey of extract-transform-load technology. Int. J. Data Warehous. Data Mining **5**(3), 1–27 (2009)
21. Warwicka, W., Johnsona, S., Bonda, J., Fletchera, G., Kanellakisa, P.: A framework to assess healthcare data quality. Europ. J. Soc. Behav. Sci. **13**(2), 1730 (2015)
22. Wang, R.Y.: A product perspective on total data quality management. Commun. ACM **41**(2), 58–65 (1998)
23. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. J. Manag. Inf. Syst. **12**(4), 5–33 (1996)
24. Wyatt, L., Caufield, B., Pol, D.: Principles for an ETL benchmark. In: Nambiar, R., Poess, M. (eds.) TPCTC 2009. LNCS, vol. 5895, pp. 183–198. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10424-4_14
25. Yang, Q., Ge, M., Helfert, M.: Guidelines of data quality issues for data integration in the context of the TPC-DI benchmark. In: 19th International Conference on Enterprise Information System (2017)