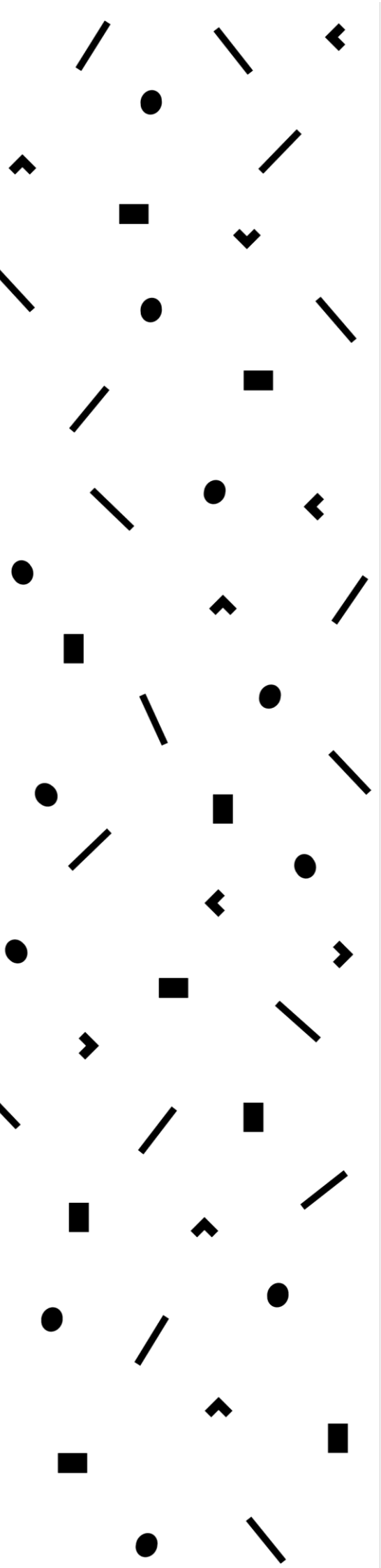

LAMK Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

DATA QUALITY IN DATA WAREHOUSES

LAHTI UNIVERSITY OF APPLIED SCIENCES
Master's Degree Programme in Information and Communications Technology
Master´s Thesis
Spring 2018
Jere Aunola

Lahti University of Applied Sciences
Degree Programme in Information and Communications Technology

AUNOLA, JERE: Data quality in data warehouses

Master's Thesis in ICT, 82 pages, 3 pages of appendices

Spring 2018

ABSTRACT

The purpose of this thesis was to research data quality and its effects on data warehouses. The definition of data quality was examined and how to measure and analyze data quality was studied. The thesis presents the most common data profiling methods and analysis methods. This is followed by data warehousing on a general level and the effects of data quality from the data warehouse point of view. In addition, it is examined how data quality analysis can be combined with the data warehouse. Semi-structured interviews were conducted with the data warehouse supplier personnel and client-side personnel. The interviews were used to gather information about the views of the interviewees about data quality and how it affects data warehouses.

The data quality always depends on the usage. The same data can be good-quality data for one use and poor-quality data for some other use. Before the data quality can be measured, the data must be profiled. Data profiling examines the data as it is, and it gathers statistics about the data. Data gained from data profiling is used to define the data quality rules. Data quality rules are always defined from the business point of view and from the user perspective. These rules are executed, and the results are used to measure the current quality of data. The result from the rules is simple ratios between acceptable records from all of the records.

Data usually comes into data warehouses from multiple sources and in different formats. Data warehouses are used to combine the data and to produce a uniform layer to support reporting and business intelligence. For this reason, it is very important that the data quality is good in data warehouses. Based on the interviews, it can be stated that the good data quality is a requirement for the entire data warehouse.

Keywords: data quality, data warehouse, DW, data profiling, poor data, good data, quality, business intelligence, BI

Lahden ammattikorkeakoulu
Degree Programme in Information and Communications Technology

AUNOLA, JERE: Tiedon laatu tietovarastoissa

Master’s Thesis in ICT, 82 sivua, 3 liitesivua

Kevät 2018

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli tutkia tiedon laatua ja sen vaikutuksia tietovarastoille. Tiedon laadusta selvitettiin sen määritelmä ja se, miten tiedon laatua voidaan mitata ja analysoida. Opinnäytetyössä käsitellään yleisimmät tiedon profilointi menetelmät ja yleisimmät tiedon laadun analysointi menetelmät. Tämän jälkeen käydään läpi tietovarastointia yleisellä tasolla ja tarkastellaan tiedon laadun vaikutuksia tietovaraston näkökulmasta. Lisäksi tarkasteltiin miten tiedon laadun analysointi olisi mahdollista yhdistää tietovarastointiin. Opinnäytetyössä suoritettiin myös semi-strukturoidut haastattelut sekä tietovarastojen toimittajan henkilöillä, että asiakkaan puolen henkilöille. Haastatteluilla tarkasteltiin toimittajan ja asiakkaan puolen näkemyksiä tiedon laadusta ja sen vaikutuksista tietovarastoille.

Tiedon laatu riippuu aina tiedon käyttötarkoituksesta. Sama tieto voi olla hyvää laadultaan toiseen tarkoitukseen ja toiseen taas laadultaan heikkoa. Ennen kun tiedon laatua voidaan mitata, täytyy tietoa profiloida. Profiloinnilla tarkastellaan tietoa sellaisenaan ja kerätään siitä statistiikkaa. Profiloinnin tuottaman tiedon avulla voidaan määritellä tiedon laadun säännöt. Säännöt määritellään aina tiedon käyttäjän näkökulmasta ja bisneksen näkökulmasta. Näitä sääntöjä ajamalla saadaan kuva tämän hetken tiedon laadusta, kun verrataan hyväksyttyjen tietueiden määrää tietueiden kokonaismäärään.

Tietovarastoihin tuodaan yleensä tietoja useista läheteistä ja erilaisissa formaateissa. Tietovarastoja käytetään tietojen yhdistämisessä ja tuottamaan yhtenäisen kerroksen tukemaan raportointia ja business intelligenceä. Tästä syystä onkin tärkeää, että tietovaraston tiedon laatu on hyvää. Haastattelujen perusteella voidaankin todeta, että tiedon laatu on koko tietovaraston edellytys.

Asiasanat: tiedon laatu, tietovarasto, DW, tiedon profilointi, laatu, datan laatu, datan profilointi, business intelligence, BI

CONTENTS

1 INTRODUCTION

Data quality plays an important role nowadays when there are more and more data coming from different sources. Data quality means that the data fulfills the requirements and needs of the usage. Data can be good quality for one use and poor quality for another use. Chapter 2 examines the definition of data quality and how it can be measured.

Today, almost every company has a data warehouse. Bigger companies can even have multiple data warehouses. The purpose of the data warehouse is to combine and unify the data from operative systems in a way that the business gets the maximum benefit from it. The data warehouse can consist of one or more source systems. The basic principles of data warehousing are presented in chapter 3. It explains in more detail the need and implementation of data warehouse and also the importance of data quality in data warehousing.

1.1 Objective

The objective of this thesis is to study data quality and the common data quality issues encountered in data warehouses. Focus on the supplier perspective and how to incorporate automated data quality analysis with data warehouse projects. Crucial goal is to identify most common data quality issues and how to detect, measure and score those issues. Other goals include the analysis of how feasible it is to automate the data quality analysis of these issues in data warehouses.

1.2 Research questions

The target was to answer the following questions:

1. What are the common data quality issues in data warehouses from supplier perspective and how these issues impact data warehouse projects?

2. What are the common data quality issues and impacts in data warehouses from clients' perspective?
3. How to measure and score data quality?
4. Would there be any need and markets for data quality analysis and data profiling?

This thesis analyses how feasible it is to automate data quality analysis in data warehouses by answering to these questions. Question 1 and 2 also rises an additional sub-question which is also answered when comparing the results of the empirical study:

5. How the clients and suppliers' perspectives on data quality issues in data warehouses differ from each other?

1.3 Research methods and workflow

Workflow of the research is divided into four main phases. The first phase is the gathering of the required knowledge by using literature review. The second phase is to carry out the interviews and refine the results of the interviews. The third phase is to analyze the refined results and use those to identify common data quality issues and select required metrics for automated data quality analysis. The last phase examines the results and conclusions are made.

1.3.1 Theoretical framework

Theoretical framework consists of gathering the required information about data quality and data warehousing. This includes defining the common metrics and scoring of data quality. It also contains the gathering of the information about how to combine the data quality analysis and data warehouse. Knowledge gathered from this part is used to lay the basis for the empirical study.

1.3.2 Empirical study

Research is done to gather information from two different perspectives about the data quality in data warehouses and encountered data quality issues. This study is done by semi-structured interviewing of persons from the supplier and client side that form the two perspectives. Two different interview templates are used, one for each perspective. Data collected by the interviews are compared and analyzed. Both qualitative and quantitative methods are used. Content analysis was done to the transcribed interviews.

2 DATA QUALITY

Data quality is important to organizations for many reasons. Bad quality data can harm operation and performance of the organization's daily activities. In worst case scenarios it can disturb making of important decisions or prevent to see the true status of the organization. (Herzog et al. 2007, 10.) Data quality is also meaningful cost maker for organizations. Costs are coming from efforts to correct data, lost customers, and wrong decisions. Data in itself is nowadays very valuable asset for organizations, and the volume is increasing all the time. (Olson 2003, 3-4.) This chapter focuses on investigating the most common data profiling methods and data quality rules.

2.1 Defining data and data quality

To understand data quality, we first need to understand what data is. Defining data can be complicated and not as simple as it sounds. One way is to divide data to the data model and data values to explain data. Data usually reflects a real-world object or concept which is called an entity. These can be for example a person, a car or a customer. Attributes are used to describe these more closely and to describe what makes the entity. Attributes for car entity could be for example the manufacturer of the car, model of the car, year manufactured and color of the car. For the person entity, these attributes could be a birth date, gender, name, height, and weight just to name a few examples. Data values are the stored values of the attributes. For example, the car entity’s data values could be Toyota for the manufacturer attribute and 2007 for the year of manufacturing attribute. (Redman 2001, 71.) Another commonly used unit for data is data element. It is the smallest unit of data and can be seen same as the attribute above (Lee et al. 2006, 125).

In the book Data quality, the accuracy dimension (2003) Jack Olson defines data quality as follows:

> *Data has quality if it satisfies the requirements of its intended use. It lacks quality to the extent that it does not satisfy the requirement (Olson 2003, 24).*

This means that quality of data depends on the usage of the data and in the data itself. Accuracy, timeliness, relevance, completeness, understandability, comparability and reliability are the most common attributes that are used to define quality. (Olson 2003, 24.)

The accuracy of data means that how many records of data is correct. The percentage of the acceptable amount of non-accurate records depends completely on the case which the date is being used. All the records do not have to be always accurate, in some cases it is completely acceptable that for example, 80% of the records are accurate. (Olson 2003, 24.)

Data timeliness means that data is available at the moment it is needed. Defining the timeliness of data also heavily depends on the usage of data. In other words, timeliness means how fast the data is available after the data is inputted into the system. (Olson 2003, 25.)

Data must be relevant to the need it is being used to. It also might bring additional info to other uses, or it might be used in completely different need. If the data does not have any relevance to the need that it is collected it is considered to be bad quality data. Data can be bad quality for one need and same time exceptionally good for another use.

Completeness means for example that there are no missing records in the database nor records have missing attributes. Missing records may cause serious issues in many cases. (Olson 2003, 26.)

Data must be understandable. If the data does not make sense to the users or if the users create false assumptions about the data it is considered to be bad quality. (Olson 2003, 26.)

Data must be trusted, meaning that the users must be able to trust the data. When data cannot be trusted and thus be used it is considered to be

poor quality data. Sometimes this can happen even if the data is accurate and otherwise good quality data. (Olson 2003, 26-27.)

All dimension of data quality is visible in figure 1 below. It contains nine main categories. Most common dimensions usually covered are Availability, Security, Comprehensiveness, Appropriate use, Clear definition, Source, Relevancy, Accuracy, Ease of interpretation, Measurement, Early warning, Help, Documentation, Naming and Unit cost. Some of these were briefly covered in this section. (Redman 2001, 106.)

Dimension	***Dimension***
Accessibility/Delivery	***Presentation Quality***
Availability	Appropriateness
Protocol	Format Precision
Security	Use of Storage
Quality of Content	***Flexibility***
Attribute Granularity	Portability
Comprehensiveness	Representation Consistency
Essentialness	Null Values
Flexibility	Formats
Appropriate Use	Language
Areas Coveresd	Ease of Interpretation
Homogeneity	***Improvement***
Naturalness	Feedback
Obtainability	Measurement
Precision of Domains	Track Record
Robustness	***Privacy***
Semantic Consistency	Consumer Privacy
Structural Consistency	Privacy of Others
Simplicity	Security
Clear Definition	***Commitment***
Identifiability	Early Warning
Source	Help
Relevancy	Special Requests
Quality of Values	Commitment
Accuracy	***Architecture***
Completenss	Library/Documentation
Timeliness	Logical Structure
Consistency	Physical Structure
	Naming
	Rules
	Redundancy
	Unit Cost

FIGURE 1. All dimensions of data quality (Redman 2001, 106)

2.2 Causes of bad data

For successful data quality assessment, it is important to understand where and how poor data comes into databases. Commonly reasons for poor quality data can be traced back into following sources: Initial data entry, data decay, moving data and restructuring of data and finally data usage. The last, data usage, causes problems in the reports created from data. The first three creates issues in the database. From this, we can conclude that the data quality is at its poorest when we use it. Still, organizations use it to create reports and to make decisions. (Olson 2003, 43.)

In the article, A Taxonomy of Dirty Data (2003) Won et al. suggests taxonomy for dirty data that is partially shown in table 1 below. The full table can be seen in appendix 1. Suggested taxonomy assumes that dirty data is manifested in three different ways; missing data, not missing data but wrong and not missing and not wrong but otherwise unusable. The last one of these three occurs when multiple databases are integrated into one or when there are no common representation rules used when inputting data. This taxonomy is hierarchical decomposition of these three ways of dirty data manifestation in databases. It also only contains primitive datatypes, and it does not consider composite types of dirty data. (Won et al. 2003, 83.) They also introduce a taxonomy of techniques to prevent, check and correct these types of dirty data identified in table 1. (Won et al. 2003, 92).

TABLE 1. Taxonomy of dirty data (Won et al 2003, 84-85)

1. Missing data
    1.1 Missing data where there is no Null-not-allowed constraint
    1.2 Missing data where Null-not-allowed constraint should be enforced
2. Not-missing data
    2.1 Wrong data, due to
        2.1.1 Non-enforcement of automatically enforceable integrity constraints
            2.1.1.1 Integrity constraints supported in relational database systems today
                2.1.1.1.1 User-specificable constraints
                2.1.1.1.2 Integrity guaranteed through transaction management
            2.1.1.2 Integrity constraints not supported in relational database systems today
                2.1.1.2.1 Wrong categorical data (e.g. out of category range data)
                2.1.1.2.2 Outdated temporal data (e.g. person's age or salary not having been updated)
                2.1.1.2.3 Inconsistent spatial data (e.g. incomplete shape)
        2.1.2 Non-enforceability of integrity constraints
            2.1.2.1 Data entry error involving a single table/file
                2.1.2.1.1 Data entry error involving a single field
                2.1.2.1.2 Data entry error involving multiple fields
            2.1.2.2 Inconsistency across multiple tables/files (e.g. the number of employee in the Employee table and the number of employee in the Department table do not match)
    2.2 Not wrong, but unusable data
        2.2.1 Different data for the same entity across multiple databases
        (e.g. different salary data for the same person in two different tables or two different databases)
        2.2.2 Ambiguous data, due to
            2.2.2.1 Use of abbreviation (Dr. For doctor or drive)
            2.2.2.2 Incomplete context (homonyms; and Miami, of Ohio or Florida)
        2.2.3 Non-standard conforming data, due to
            2.2.3.1 Different representations of non-compound data
                2.2.3.1.1 Algorithmic transformation is not possible
                2.2.3.1.2 Algorithmic transformation is possible
            2.2.3.2 Different representations of compound data
                2.2.3.2.1 Concatenated data
                2.2.3.2.2 Hierarchical data

2.2.1 Data entry

The most common reason for inaccurate data is the initial data entry into the system by a human being. A person entering the data makes a simple misspelling, inserts a correct value into the wrong field or chooses the wrong item from a drop-down field. Data in operational systems comes from a person who enters the data into the system. People make mistakes

all the time. It is almost impossible that someone could enter data correctly into hundreds and hundreds of forms continuously. (Olson 2003, 44.)

Data entry into the system usually starts with completing some kind of a form. The form can be traditional paper form or form that is completed electronically with a computer. The design and implementation of the form play a major factor about how likely the person entering the information makes mistakes or input invalid values. The person entering the data into the form should be selected so that the same person enters data using the form frequently. This is so that usually the person filling the form first time are more insecure how and how to fill the form. Nowadays almost all the forms are available to be filled on the Internet, and this decreases the need for experienced data entry persons. For this particular reason, it is crucial to design the forms to be clear, understandable and logical. (Olson 2003, 44-45.)

One of the most common issue when entering data through forms is so-called null problem issue. It is common that person entering the data into the form does not know all the values. Forms do not usually have a field that could be used to tell that the person entering the data does not know the value. For this reason, the field is usually left blank. When reviewing the data entered it is impossible to know why the field was left blank, because it was not applicable or because the value was not known? In a way, it would be good that in this kind of situations there would be an option in the form to tell that the value is not known or that the field is not applicable. Data would be accurate, and no room for questing would be left. (Olson 2003, 46.)

It is also good to remember so-called considered mistakes in the forms. These are usually caused by one of the three reasons: Person does not know the correct value, a person does not want to tell the correct value or the person benefits from entering an incorrect value. Persons input wrong value to the form if the value is not known to them, but the field is mandatory. It is also possible that the person inputting data knows the correct value but does not want it to end up into the system or the person

gains some kind of benefit by inputting wrong value to the field. (Olson 2003, 47.)

2.2.2 Data decay

Data that is accurate when inputted may lose its accuracy in the database trough time. In other words, data values do not change, but the accuracy of the data does. All the attributes are not vulnerable to the accuracy lost by time. For example, personal data can become inaccurate quite fast in a database. Person move, change their last name or change their phone numbers. Organizations do not commonly identify this issue that they can have data that loses its accuracy trough time and that the data should frequently be updated. The importance is to identify this kind of data in the database and make a plan to check the accuracy of the data regularly. (Olson 2003, 50-51.)

2.2.3 Moving and restructuring data

Data accuracy is usually affected by moving the data or when restructuring the data. Moving and restructuring of data is common in data warehousing. Data is loaded from source systems and moved to the data warehouse. This step is commonly underestimated when finding out what and where causes the inaccuracies in data. Moving of the data into a data warehouse is commonly done using ETL-process (Extract, Transform and Load). This process is done utilizing separate packaged tools or in-house created scripts and programs. When using this kind of tools only in very rare cases, the tool is responsible for inaccurate data, but the usual reason for inaccurate data is the definitions that are used to load the data. (Olson 2003, 52.)

It also rare that there is up-to-date documentation available about the structure of the database or description about the fields in it. Operational systems are changing all the time, and it is common these changes are not documented. For example, the use for the field might change to completely different than it was originally designed for. For this reason, the

documentation can tell the original meaning of the field instead of the new meaning if the documentation is not updated correctly. (Olson 2003, 54.)

2.2.4 Data usage

Data can be accurate, but if the user using it does not understand it, then it can be inaccurate (Olson 2003, 62). If we remember that the definition for data quality is how it fits for the purpose. This means that data can be accurate for one use but inaccurate for another use. For this reason, new uses for data can have an impact on data quality even if the data itself remains unchanged. (Maydanchik 2007, 20.)

2.3 Profiling data and data quality rules

Data profiling is important. It provides a comprehensive look at what the data actually looks like. More deeply the data profiling is, better the results are, and more precise data quality rules can be defined. In other words, data profiling tells us how the data looks like and data quality assessment tells how good it is. Data is usually profiled with following methods: attribute profiling, relationship profiling, state-transition profiling and dependency profiling. (Maydanchik 2007, 49-50.)

Rules to measure data quality are the most important part when assessing the data quality. Rules set constraints for data. In general, the more there are rules the better are the results. In practice defining these rules can be considered to be quite hard and it should be done systematically. Creation of the rules is not hard nor the most time-consuming part. Defining the rules is. Rules must be programmed and any of the rules should be possible to run at any given time if necessary. After creation of the rules it is common that there is a need to fine-tune the rules to be more precise and to minimize false positives. (Maydanchik 2007, 50-53.)

2.3.1 Attribute rules

Attribute rules affect the smallest partition of data, the data values. Attributes describes the object. Invalid values for attributes is usually easy to identify. For example, we know that Car objects attribute manufacturing year cannot have value of 1600 or that Human object cannot have value 1700cm in its height attribute. Attribute rules are the most common and simplest of the rules. (Maydanchik 2007, 63.)

Attribute rules are used to constrain the values that attribute can have. This can be achieved by allowing only given values, set of values or range of values for attribute to have. These rules can commonly be defined quite effectively from the results of attribute profiling. (Maydanchik 2007, 65.)

Attribute profiling is used to inspect single attributes and it produces three types of results; basic aggregated statistics, most common values and distribution of values (Maydanchik 2007, 65). Figure 2 shows results of attribute profiling for database table Persons attribute birth_date.

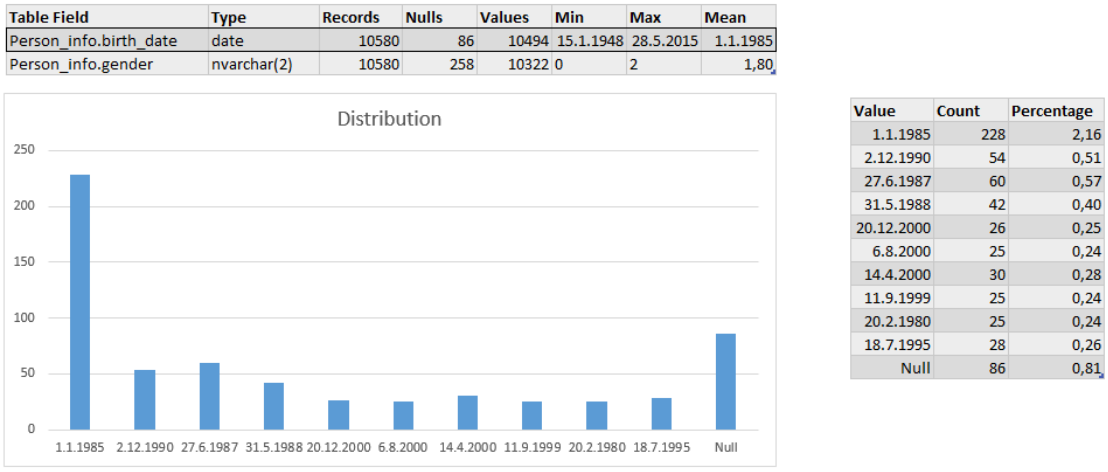

FIGURE 2. Attribute profiling (Maydanchik 2007, 65)

In the top of figure 2 basic statistics about the attribute is shown. It describes the data type, count of null values, total row count, minimum value, maximum value, the mean value and some additional info about the attribute. The table in the right shows the most common values of the

given attribute and how many percents it is from the total count of rows. The graph shows the distribution of the most common values. (Maydanchik 2007, 65-66.)

Optionality constraint is the easiest and the simplest to identify. Meaning of it is to prevent attributes to get empty or null values. It is common that relational tables define what attributes can have null values and which attributes cannot. To identify these via attribute profiling the first step is to compare the count of null values against the total count of records. If there are a very low number of null values, then it can be concluded that the attribute value is required. Sometimes default values are used in the databases to get around the not null constraint. When looking from the data quality point of view then the default values do not differ from the missing values. (Maydanchik 2007, 66-67.)

Default values can be identified with attribute profiling by looking the most common values of the attribute. Figure 3 shows most common values for City and Weekly work hours attributes. Rows that are marked as red in the figure are identified as default values. Usually, the values that occur more often than others are good candidates to be default values. (Maydanchik 2007, 68-69.)

City

Value	Count	Percent
Espoo	101	1,97
Tampere	32	0,62
NOT DEFINED	28	0,55
Lahti	15	0,29
Pori	8	0,16
Tallinn	6	0,12
Stockholm	3	0,06

Weekly work hours

Value	Count	Percent
37,5	3546	69,22
40	546	10,66
0	135	2,64
30	64	1,25
15	8	0,16
25	5	0,10

FIGURE 3. Example of default values in database tables

Precision constraint is rarely defined in the data model even though some attributes may have defined precisions. Precision constraint requires that

attribute values have the same precision. For numeric values, this usually means that count of decimals is defined or there is defined a rule for rounding. Data profiling can be used to create a distribution of precision. The result of this kind of profiling for Salary attribute is shown in figure 4 below. Most of the values are whole numbers and most of these are dividable by ten, hundred and thousand. Small partition of values contains decimals and are probably erroneous values. Results of the profiling can't be straightly used to detect correct precision, support from the business is always needed to determine the correct precision. Precision constraints can be used for numeric and time data. For numeric data the precision constraint can be used to require a certain number of decimals and for time data it can be used to set precision in forms of the month, day, hour, minute and second for example. Granularity and unit of measurement constraints are similar data quality constraints. (Maydanchik 2007, 74-76.)

Salary

Value	Count	Percent
0,01	28	0,56
0,1	95	1,9
1	378	7,56
10	1648	32,96
100	2197	43,94
1000	654	13,08

FIGURE 4. Example salary precisions (Maydanchik 2007, 75)

Format constraints are used to set the desired format for attribute values. Generally, format constraints are used for legacy systems where attributes are not necessarily strongly typed but instead for example dates are stored as text. Format constraints usually are represented with so-called value masks, like DD.MM.YYYY for date attribute for example. This mask states that first two numbers are representing the date. After that two more numbers for month separated by one character. Last four numbers are for the year and again separated from month by one character. Format constraints are mainly used for text attributes. Length and format

constraints are common for these attributes. For example, Finnish social security number could have following format constraint: first six characters should be numbers, next character should be +, - or A. After these there should be four characters where three first should be numbers and last one should be either number or character. Freeform text fields that have more than one word are especially hard to validate and require modern text analysis and parsing software.

He continues that many attributes have limited set of possible values and that there are rules for these kinds of attributes. Valid value constraints purpose is to limit the permitted values into this kind of set. To determine this set, all the values and occurrences of the values are required for given attribute. Most common values are not enough in this case. Then the valid values are determined from the list of values gathered earlier and the list of valid values is created. For numeric and date/time values the list would be too large and for this reason it is common to restrict them with domain constraints instead. Example of this kind of domain constraint could be constraint stating that salary should be greater than 0 or that the employees age should be at least 16 years. In some cases, it is hard to determine correct limits like when limiting employees age, it would be hard to set maximum limit. (Maydanchik 2007, 69-72.)

2.3.2 Relational integrity rules

In relational databases it is common to have data that has some kind of relation in different tables and relate to each other by using relationships. For example, database table Car contains data about the car and it can have relationship to table Manufacturer that contains data about manufacturers. Relationship cardinality in this example means that Manufacturer can have one or more cars. Car can only have one manufacturer. Relational databases usually use foreign keys to implement relationships between tables. Relational integrity rules are rules that places constraints to these relations. (Maydanchik 2007, 79-82.)

Identity key is information that is used to identify real-world entities from each other in the data. Identity rule is used to make sure that for every record there is only one real-world entity and that no two records maps to single real-world entity. Usually there are as many identity rules as there is entities in the data model. (Maydanchik 2007, 82.) In data warehousing term business key is used commonly instead of identity key. Business key means combination of one or more attributes that individualizes the record from other records. These attributes should be as static as possible meaning that the values do not change. This means that commonly the business keys are also natural keys of the entity. It is also preferred that business keys have meaning in the business point of view. Business keys usually are not surrogate keys which are commonly used as identity keys in operational databases. (Linstedt et al. 2016, 95-97.) In databases it is common that tables have primary keys, but those keys rarely identify the entity. That is because usually primary keys are surrogate keys. (Maydanchik 2007, 83.)

Figure 5 shows records from database table Users. Basic information about users is stored in the system and the table contains UserId surrogate key column as a primary key. Primary key guarantees that each row can be identified as individual in the system. It does not guarantee proper individualization of users in the table. User Maija Meikäläinen for example has two records in the table with identical social security numbers. This means that the primary key of the table is not true identity key. Identity rule compares the real identity key to other keys. All duplicates are erroneous. Unique identity key validation might still not be enough to truly identify all identity key violations. (Maydanchik 2007, 83-84.)

UserId	FirstName	LastName	BirthDate	SocialSecurityNumber	Gender
1563	Maija	Meikäläinen	12/06/1985	120685-112A	F
3141	Maija	Meikäläinen	12/06/1985	120685-112A	F
6297	Matti	Virtanen	25/09/1970	250970-255Z	M
12609	Matti	Virtanen	17/01/1983	170183-1035	M
25233	Ville	Korhonen	29/04/1987	290487-179C	M
50481	Ville	Korhonen	04/11/1968	041168-3111	M
100977	Seppo	Meikäläinen	15/05/1990	150590-121V	M

FIGURE 5. Identity rule violation in the user table

Reference rule is used to guarantee that reference from one entity to another one can be resolved. Foreign keys are used in relational databases to represent reference rules. Foreign key consists of one or more attributes of the entity and it references to another entity's primary key. (Maydanchik 2007, 85.)

Order status history

OrderId	HistoryDate	OrderStateCode	OrderState
1004	26.5.2017	S	SUBMITTED
1004	26.5.2017	PP	PAYMENT PENDING
1004	28.5.2017	PR	PAYMENT RECEIVED
1005	2.6.2017	S	SUBMITTED
1005	2.6.2017	PP	PAYMENT PENDING
1006	3.6.2017	S	SUBMITTED

Order Status

OrderId	OrderStatusCode	OrderStatus
1003	S	SUBMITTED
1005	PR	PAYMENT RECEIVED
1006	PP	PAYMENT PENDING

FIGURE 6. Reference rule violation

Figure 6 consists of two database tables; Order Status and Order Status History. Order Status History table holds the history of order status. Order Status table contains the current statuses of orders. History table references Order Status table with a value in OrderId attribute. Data quality reference rule says that for each history rows there should be one

current row in the Order Status table. By looking the data presented in the figure 6 it can be seen that for OrderId 1004 there are three records in history table but none in the parent table Order Status. Reasons, why there is no row in the table, can be for example that the rows are erroneous, and they belong to another order. It also might be that the order with id 1004 has been removed erroneously in some point of time. (Maydanchik 2007, 85-86.)

Cardinal rules define the cardinality of relations. Where reference rules defined that referenced entity should be present in the referenced table the cardinal rules define how many records should be found via reference. Count of cardinal rules can be easily determined by counting the references. Two cardinal rules per reference, one for both directions. Example cardinal rule by using tables and data represented in figure 7 could be that for each product there should be one and only one product group and one product group should have zero or more products. Record in Product table with value 20152 in its ProductId attribute violates this rule, it does not belong to any of the Product Groups. A common problem in the databases is that they do not completely support defining of cardinality for foreign keys. (Maydanchik 2007, 86-88.)

Product

ProductId	ProductGroupId	Name	Description
20151	1001	Small Table	Small kitchen table...
20152	*null*	Nightstand	White nightstand...
20155	1002	Basic Couch	Basic Couch...
20156	1002	Premium Couch	Premium Couch...
20157	1002	Floor Lamp	Bronze colored floo...
20187	1001	Knife Set	Set of 5 stainless st...

Product Group

ProductGroupId	Name	Description
1001	Kitchen	Kitchen furniture
1002	Living Room	Living room furniture
1003	Bedroom	Bedroom furniture

FIGURE 7. Cardinal rule violation

To determine true cardinality, it is important to carry out relationship cardinality profiling for the data. This profiling technique calculates the true frequency of each relation. Results can be visualized by a graph like one shown in figure 8 below. The graph shows how many related records are for parent record. In this case, it shows how many records are found in the product table that references the parent table product group. It can be seen that most groups contain more than three products but still there are two groups that do not have any products and some groups only have few products. (Maydanchik 2007, 89-90.)

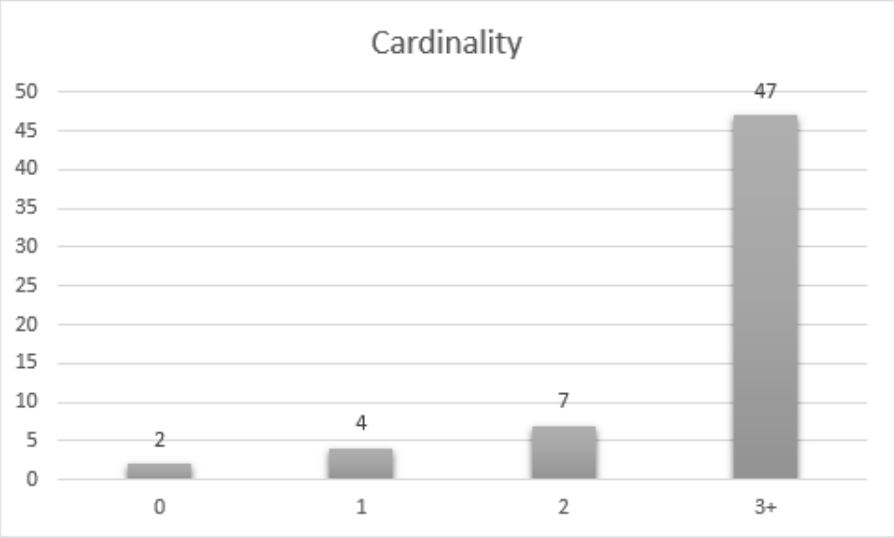

FIGURE 8. Cardinality chart of product and product group relation

2.3.3 Historical data rules

Most of the attribute values of real-world objects change over time. For example, the weight of human being changes and varies throughout its lifetime. In databases that contain these kinds of time-dependent attributes, it is important to take it into account. In some cases, only the recent values are important. For example, Customers latest email address could be one of these kinds of attributes. Still, there probably would be

some attributes about customers that should be stored with value history. (Maydanchik 2007, 93.)

The major part of data stored in operational systems and in data warehouses is time-dependent. This kind of data is prone to errors. Databases usually have timestamps that describe when the data is effective. This can be implemented by saving at least the effective from timestamp for given record. Usually the effective to timestamp is also present, even though it is not required. It is possible to determine the end time from the next records from time by subtracting one unit from it. (Maydanchik 2007, 93-94.)

Retention rule is used to set rules for how many rows of history there should be available or how far back in time there should be history available. Currency rule, in turn, is used to limit how old the newest record can be. History record values can form predictable patterns which can be used to create data quality rules if needed. (Maydanchik 2007, 95-96.)

Example of records in employee salary history table is shown in figure 9. EmployeeID is used to identify employees from each other. In addition, the table contains From- and ToDate fields for each record which are used to identify the time period when the record was active. YearlySalary attribute reflects the yearly salary of an employee in a given period by the date fields. This kind of data that is aggregated to a time period is called as accumulator history. There are more constraints for accumulator history data than there are for time series data. In this case, one constraint is that the timespan for each record should be exactly one year and any of the employees should not have overlapping periods in the data. (Maydanchik 2007, 96-97.)

EmployeeId	FromDate	ToDate	YearlySalary
512	1.1.2011	31.12.2011	46000
512	1.1.2012	31.12.2012	48200
512	1.1.2013	31.12.2013	50800
512	1.1.2014	31.12.2014	52600
513	1.1.2012	31.12.2012	40600
513	1.1.2013	31.12.2013	43500
513	1.1.2014	31.12.2014	46400

FIGURE 9. Example of employee salary history (Maydanchik 2007, 97)

Currency rule is used to guarantee the freshness of history data. It can be implemented by finding the most recent record for the entity and comparing the effective date to predetermined limit value. Another way is to limit the age of the most recent record. Currency rules may change at the object level. It can be that some of the records represent data that is no more updated and should be handled differently. (Maydanchik 2007, 98-99.)

EmployeeId	FromDate	ToDate	YearlySalary
512	01/01/2011	31/12/2011	46000
512	01/01/2012	31/12/2012	48200
512	01/01/2013	31/12/2013	50800
512	01/01/2014	31/12/2014	52600
513	01/01/2012	31/12/2012	40600
513	01/01/2013	31/12/2013	43500
513	01/01/2014	31/12/2014	46400
513	01/01/2015	31/12/2015	48000
513	01/01/2016	31/12/2016	50500
513	01/01/2017	31/12/2017	51000

FIGURE 10. Example of currency rule violation in the employee salary history table (Maydanchik 2007, 98)

Figure 10 shows yearly salary history for two employees. Used currency rule states that each employee should have salary data for last full calendar year, in this case for the year 2017. By looking the data, we can see that employee with id 513 has data for the year 2017 meaning that its data is current and fills the requirements. An employee with id 512 is

lacking data and only has history up to the year 2014 and thus breaking the currency rule. (Maydanchik 2007, 98.)

Where currency rule sets limitations for the freshness of data, the retention rule sets limits on how much there should be data available for events. It can be implemented by setting limits how many history records there should be available for an object or how far back there should be history available. These rules are common to be based on authority demands. It can be that some data should be kept predetermined time before it can be deleted. It is also important to consider that not all records necessarily have a history before a set date. This kind of example is presented in figure 11 below. We have set retention rule stating that each employee should have data five years back. An employee with id 512 has only data four years back and does not fulfill this set rule. However, it is possible that the employee was hired in 2011 and for this reason does not have any salary history before that time. This means that it is important to note the maximum timespan where data is available can change per record. In this example, this means that when creating the retention rule for salary history we also should check the hire date for an employee to determine the correct rule. (Maydanchik 2007, 99-100.)

Employee	FromDate	ToDate	YearlySalary
512	01/01/2011	31/12/2011	46000
512	01/01/2012	31/12/2012	48200
512	01/01/2013	31/12/2013	50800
512	01/01/2014	31/12/2014	52600
513	01/01/2012	31/12/2012	40600
513	01/01/2013	31/12/2013	43500
513	01/01/2014	31/12/2014	46400
513	01/01/2015	31/12/2015	48000
513	01/01/2016	31/12/2016	50500
513	01/01/2017	31/12/2017	51000

FIGURE 11. Example of retention rule violation in the employee salary history table (Maydanchik 2007, 100)

Accumulator history type of data which is aggregated it is common to set continuity and granularity rules. Granularity rules restrict the time span to be the same for all records. For example, the timespan for each record should be exactly one month. In previous examples of employee yearly salary, the granularity rule was used to limit the time span to be exactly one year. Continuity rule is used to enforce that the timespans do not overlap and that it does not contain any gaps. This means that the effective date immediately follows the end date of the previous record. Figure 12 contains examples of continuity and granularity rules for employee yearly salary history table. It shows an example of wrong granularity, the gap in time span and overlapping intervals. Continuity and granularity rules are not used for data containing measurement data that are taken in points in time. These are only used for aggregated data where records are valid for some period of time. (Maydanchik 2007, 101.)

For history data, it is common to set more complex data quality rules like timeline pattern rules and value pattern rules. Timeline pattern rules are used to restrict that the measurements are taken by given period between each other. For example, every tenth day of the month or every day six o'clock in the morning. Value patterns restrict the upcoming future values based on the past values. These value pattern rules can be used to restrict the future values to be greater than the current value for example. Another example would be a rule that sets restrictions that future value should be inside of subset of values based on the current value. (Maydanchik 2007, 102-104.)

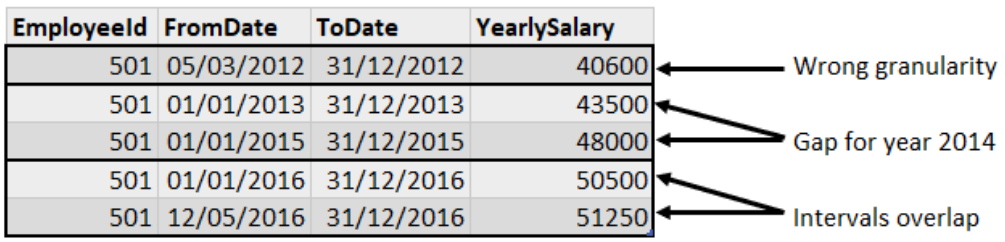

FIGURE 12. Continuity and granularity example (Maydanchik 2007, 101)

2.3.4 State-dependent object rules

State-dependency means objects that have a state and the state changes over time. Example for this kind of object would be common ordering process in the web store. First, the order is placed by the customer into the system. Orders state is now: created. When the shop takes the order for handling its state changes to processing. Next step could be: shipped and the last step could be: received. It is important to measure the quality of this kind of data with state-dependent profiling and rules. (Maydanchik 2007, 113.)

Figure 13 presents states of the order in a timeline. Order moves through a sequence of different states in the system in its lifetime. This kind of objects is called as state-dependent objects. Order state changes from received to processing and from shipped to completed. However, all changes to the states are not permitted, for example, the order cannot change its state from received to received again. When determining the data quality rules for state-dependent objects it is essential to identify the allowed and not allowed state changes. State change models are used to describe restrictions for this kind of objects with the definition of states and actions. (Maydanchik 2007, 114-115.)

States means all the states that object can be in. The object must be in one of these states and it can be only in one state at a time. The term for the first and last state of the object is a terminator. Actions mean the actions that triggered the change of the state in the object and they can have constraints which must be filled before the action can be performed. (Maydanchik 2007, 115.) These changes in states are called as transitions. State-dependent objects in programming terms are state machines and they can be visualized using state machine diagrams. (Fowler 2010,107-109.) From the states identified in figure 13, the terminator states can be identified to be Received and Completed. The first state that the order can have is Received and the last that it can have is Completed. (Maydanchik 2007, 115.)

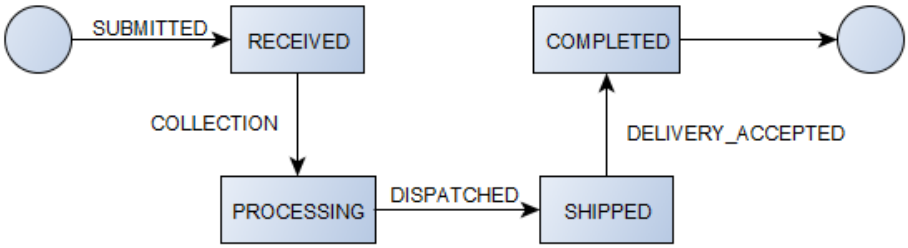

FIGURE 13. Example of order status states and actions

State-dependent objects can be identified in the database by examining the attributes of entities. If an entity has timestamps that represent the effective time of the record, there is a good chance that it is a time-dependent object. Next step would be to examine if there are any attributes that represent a state like, for example StatusCode, State, etc. It is also important to notice that not all state dependent objects have timestamps. It is possible that they are ordered by some other way like with sequence numbers. For state-dependent objects it is common that they are possible to order chronologically via the timestamps or some other way. Figure 14 shows web stores database table for order status history. It contains state-dependent objects whole lifecycle. Attribute FromDate can be used to order the statuses chronologically and attributes ActionCode and OrderStatusCode represents the state and action which led to that given state. (Maydanchik 2007, 117-118.)

OrderStatusHistory

OrderID	FromDate	ToDate	ActionCode	OrderStateCode
1121	01.01.2018	04.01.2018	SUBMITTED	R
1121	05.01.2018	08.01.2018	COLLECTION	P
1121	09.01.2018	14.01.2018	DISPATCHED	S
1121	15.01.2018	*null*	DELIVERY_ACCEPTED	C

FIGURE 14. Example of order status history table

By profiling these state-dependent objects it is possible to determine state changes models and this kind of profiling is called as state changes model profiling. This profiling method holds a different kind of profiling methods to analyze state dependent objects and to get real data about the order of states and actions and information about the duration of the states. First of these methods is to perform state and terminator profiling which can be used to determine all the possible states and possible terminator states. Figure 15 shows the results of this kind of profiling performed to the order status history table that was presented in figure 14. Profiling shows how many times object is first seen in any of the states. In other words, it shows how many times given state is seen as a terminator. It is clearly seen in figure 15 that the submitted state is the correct terminator state. (Maydanchik 2007, 119-120.)

Terminator	Frequency	Percentage
SUBMITTED	15000	98.68%
COLLECTION	188	1.24%
DISPATCHED	12	0.08%
DELIVERY_ACCEPTED	0	0.00%
TOTAL	**15201**	**100%**

FIGURE 15. Results of terminator profiling

The second phase is to perform state transition profiling for the objects. It is used to determine what state the object is coming to another state and how often that transition occurs. This information can be used to determine the allowed state transitions. Results of state transition profiling for the order state history objects Completed state is shown in figure 16. By analyzing the results, it can be concluded that the only correct state from where the object can come to Completed state is Shipped state. (Maydanchik 2007, 121.)

From State	To State	Frequency	Percentage
Shipped	Completed	14441	95.00%
Processing	Completed	631	4.15%
Received	Completed	129	0.85%
Completed	Completed	0	0.00%
TOTAL		**15201**	**100%**

FIGURE 16. Results of state transition profiling

Action profiling is used to examine how many times given action results in certain state change. Figure 17 shows results for action profiling where all the actions are gathered that led to Submitted terminator state. By examining the results, it is obvious that only correct terminator action is RECEIVED action. Other two actions are either erroneous or they are partially missing history from the order history table. Action profiling is used to examine all the actions and the state changes they cause in the data. (Maydanchik 2007, 122.)

Terminator	Action	Frequency	Percentage
Shipped	DISPATCHED	14441	95.00%
Shipped	COLLECTION	631	4.15%
Shipped	SUBMITTED	129	0.85%
TOTAL		**15201**	**100%**

FIGURE 17. Results of action profiling for terminator state Shipped

Three data quality rules can be used to ensure the validity of states, actions, and terminators. State and action constraints are basically regular attribute domain constraints. State domain constraint is used to limit states into allowed states. Most common errors in the states are misspellings in other way correct records and the correct state can usually be determined from the value of the action. Action domain constraint limits the possible values of the action into allowed values and like with the states the most common errors with actions are also misspellings and the correct action can be determined from the value of the state. Terminator domain constraint limits the terminator states into allowed terminator states. As

mentioned earlier terminator state is a state where the object when it is seen first and last. Errors in the terminator states are mostly caused by missing records in objects lifespan. (Maydanchik 2007, 125.)

State-Transition constraints are used to limit state changes into allowed changes. State-transitions are generally presented in matrix-like one in figure 18. Rows showing the states where the object comes from and columns showing the destination states. Cross in the intersection means that the state-transition is allowed and blank means that the state-transition is not allowed. For this example, there is additional state Canceled added to show more interesting matrix. (Maydanchik 2007, 126-127.)

		TO STATE				
		Submitted	Processing	Shipped	Canceled	Completed
FROM STATE	Submitted		X		X	
	Processing			X	X	
	Canceled					
	Shipped					X
	Completed					

FIGURE 18. State transition matrix

State-action constraint is used to validate consistent changes in objects state by given action. This means that resulting state change in action should be always the same correct state for the object. Figure 19 shows an example of data in an OrderStatusHistory table where there is a violation of transition constraint. State code for order with id 1121 is not changed when its action is COLLECTION. Only the state PROCESSING is a valid state for COLLECTION action. (Maydanchik 2007, 126-127.)

OrderStatusHistory

OrderID	FromDate	ToDate	ActionCode	OrderStateCode
1121	01.01.2018	04.01.2018	SUBMITTED	R
1121	05.01.2018	08.01.2018	COLLECTION	R
1121	09.01.2018	14.01.2018	DISPATCHED	S
1121	15.01.2018	*null*	DELIVERY_ACCEPTED	C

FIGURE 19. Order status history table showing a state-action violation.

2.3.5 Attribute dependency rules

Attribute dependency means that some attribute value dependences on some other attributes value. This kind of dependency is the simplest one. One attribute value affects other. Usually, the dependency is not this simple. It can have multiple attributes whose values affect the value of the attribute. (Maydanchik 2007, 144.)

Redundant attributes are data values that represent the same attribute of the real-world object. In databases, it is common to discourage repetition of data because it makes maintainability much harder since changes to data need to be done to more than one location. Still, it is common, particularly in legacy systems and also in some of the modern-day applications are repeating same data. In most cases the repletion is intentional, and the most common reasons are that it might give better query performance, or it makes fetching of the data more easily for presentation. Repetition of data can also occur in data warehouses where it is commonly inevitable. Same data comes from different source systems. In these situations, it is common to use redundant attribute rules. For example, same employee information can come to the data warehouse from different systems like time entry system and finance system. Both systems probably at least contain the name of the employee. (Maydanchik 2007, 144.)

Figure 20 shows example data of two tables: Order and OrderHistory. Order table contains information about the order and OrderHistory table contains events of the order in atomic level. By examining the figure, we can see that the Order table contains attribute OrderDate that can also be

determined from the Date attribute of the OrderHistory table. Also, the OrderCompletedDate can be determined from the history. This kind of attribute redundancy is common in operative systems. Simple redundant attribute rule can be used in this kind of situations. The rule is simple, and it says that *Attribute1 = Attribute2*. (Maydanchik 2007, 145.)

Order

OrderID	OrderDate	OrderCompletedDate
1101	12.01.2018	26.01.2018

OrderHistory

OrderID	OrderStatus	Date
1101	RECEIVED	12.01.2018
1101	SHIPPED	20.01.2018
1101	COMPLETED	26.01.2018

FIGURE 20. Examples of Order and Order History table

Derived attributes are calculated attributes which values depends on other attributes. These attributes are common for complex calculation rules which depend on several records or several entities. One example of derived attributes is the time entry systems projects total amount of hours done attribute. It is aggregated from the time entries done in the system for a project or its tasks. It is derived attribute which is made from the sum of the atomic entries. (Maydanchik 2007, 146-147.)

Partially dependent attributes are attributes which allowed values are limited by another attribute. The limitation is not absolute, instead, it limits the number of allowed values into subsets from all of the values. One simple example of this kind of attribute is seen in figure 20. The OrderCompletedDate attribute is a partially dependent attribute. The following rule can be determined from it: OrderDate < OrderCompletedDate. This rule can also be presented in another way: OrderCompletedDate – OrderDate > 0. It can be said that the value of one attribute restricts the values that other attribute or attributes can have into a smaller subset. Conditional optionality means that the value of attribute

restricts another attribute to have a value or not to have value. It means that the other attribute dictates if another attributes value should be null or not null. These attributes always are also partially dependent attributes. (Maydanchik 2007, 148-149.)

2.4 Data sampling strategies

Data sampling is an important step of data quality assessment. In many cases, it is not feasible to analyze all data. There can be simply too much data, analysis takes too much time, or it would cost too much. Operational databases can have millions of records in one table and data warehouses can have much more records. (Lee et al. 2006, 67.)

The first step is to think how to take the samples. There are few good methods for taking samples: Simple random sample, systematic sample, stratified random sample and cluster sample. (Lee et al. 2006, 69.) Most common of these methods is the random sample method (Lee et al. 2006, 71).

The simple random sample is the easiest and simplest method of taking samples. It consists of taking x number of rows from a table containing 1 to N rows where *x* is the sample size. The rows are selected by generating *x* number of random numbers between 1-N which are used to get the corresponding row from the table. Resulted rows form the sample. (Lee et al. 2006, 70.)

Systematic sample resembles the random sample. The random number is generated (*x*) to determine the starting point of the sample. After this, every *y*th row from a table is selected starting from the *x*th row. *Y* is determined by the ratio between row count of the table and the sample size. (Lee et al. 2006, 70.)

The stratified random sample is used when it is known that parts of data are more devoted to errors. Data is first divided into subsets so that every subset should contain a uniform amount of rows devoted to errors. Rows

are then randomly selected from these subsets. This method ensures that the sample has rows from all of the subsets. (Lee et al. 2006, 70.)

In cluster sample, the data is divided into groups with some rule. Some of the groups are then randomly selected. All of the rows from selected groups are then selected or just a portion of rows are selected randomly. Practically this method is very useful in data warehouses if data from multiple operational systems is integrated into a single table. In that kind of situation, data can be clustered based on the operational system. This method can also be used to select random amount of tables from all of the tables in the data warehouse. (Lee et al. 2006, 71.)

To determine the sample size the formula shown in figure 21 can be used. It is the common formula used when determining the sample size and it is meant to be used in situations where population size is large or unknown. *z* is the confidence level. (Smith 2013, 2).

$$Sample\ Size = (z)^2 p(1-p) \,/\, (e)^2$$

FIGURE 21. The formula to determine sample size (Smith 2013, 3)

TABLE 2. Most common z-values (Smith 2013, 3)

Percentage	z value
90%	1.645
95%	1.96
99%	2.326

The most common confidence levels are shown in table 2 above with percentages and the corresponding *z* values. *p* is standard deviation which means that how much variance there is expected to be in the results. This is usually unknown and then the value should be *0.5* which

ensures high enough sample size. The last value *e* is the margin of error in percentages. Figure 22 shows an example calculation with a confidence value of *99%*, standard deviation *0.5* and margin of error *5%*. The resulting sample size is 541 units. (Smith 2013, 2-3.)

$$Sample\ Size = (2.326)^2 \times 0.5(1 - 0.5) \,/\, (0.05)^2$$

$$Sample\ Size = (5.410276 \times 0.25) \,/\, 0.0025$$

$$Sample\ Size = 1.352569 \,/\, 0.0025$$

$$Sample\ Size = 541.0276$$

FIGURE 22. Example calculation of sample size (Smith 2013, 3)

In general, it is more important to determine how to take the sample than what is the size of the sample (Maydanchik 2007, 208).

2.5 Measuring and scoring data quality

Results from data quality rules are used to form the reports of data quality assessment. Reports are used to view precise information about data quality. By selecting different combinations from results it is possible to form various aggregated scores. These aggregated scores can measure data quality for different uses or data quality of different source systems. Defining the aggregated scores is a very important step. Without aggregated scores interpretation of the results from rules is very hard. (Maydanchik 2007, 243.)

Aggregated scores describe high-level estimate of data quality. Every score aggregates result from data quality rules into one number. This number shows the percentage of good data records among all of the records. It is possible to create and build multiple different aggregated scores by selecting different groups from the data. Aggregated scores are meant to provide clean and understandable measures from the huge amount of error reports created by the data quality rules. There are few

important aggregated scores: Impact of bad data, Sources of bad data, Location of bad data and record- and subject level scores. (Maydanchik 2007, 244.)

Because the data quality is defined as fitness for the use it is important to create aggregated scores for different uses of data. Aggregated score always measures the percentage of good records among all of the records. Aggregated score for the data usage uses only the partition of data and data quality rules that are meaningful for the given data usage. (Maydanchik 2007, 245.)

Data comes from different sources into the database. Some data comes from manual entry and other data might come from electronic sources. Aggregated scores by source represent the data quality of different sources. These aggregates are created by selecting the records and data quality rules that affect the given source in question. These aggregated scores are important because it can be used to get improvements to data quality. When we know the sources of bad data it is quite easy to intervene and make corrective actions to these sources. Another this kind of aggregate is to create the aggregated scores by diving records with time. This allows creating scores that show how the quality of data is changed among the records by time. Is the quality of data been better in the past or is it the way around. (Maydanchik 2007, 246.)

Commonly the data quality errors do not divide equally among the database. Some tables have more errors than others and some records have more errors than others. Locations of the errors can be measured with different aggregated scores. Database score aggregate measures the errors in all of the records in the database. This aggregate is not so important but it is easy and simple to implement. Entity score aggregate measures errors in one table. This is also easy to implement and understand. With this score, it is possible to make more assumptions where the errors are, and this measure can be used to determine which tables to select for data cleansing. In addition to these scores, it is important to create aggregated scores for different kind of subjects. These

subject aggregate scores provide very important information about where the errors are. Subject aggregate inspects data in different subjects. For example, dividing employee information to subjects by daughter companies and creating an aggregated score for each subject. With this kind of aggregated scores, it is possible to see if one of the subsidiaries is providing data with more errors than others. (Maydanchik 2007, 247.)

Record level score measures bad data records among all of the records. Where subject level score measures the percentage of subjects that have one or more errors. These two scores complete each other and are an important part when scoring data quality. (Maydanchik 2007, 247-248.)

Data quality is commonly measured with simple ratios. In these simple ratios *1* means most desired result and *0* means completely undesired result. Figure 23 shows the formula for simple ratio. This ratio is used to measure the ratio between count undesirable values and the total count of values. (Pipino et al. 2002, 213.)

$$Rating = 1 - \left(\frac{Number\ of\ undesirable\ outcomes}{Total\ outcomes}\right)$$

FIGURE 23. The formula of the simple ratio (Lee et al. 2006, 54)

Many data quality measures utilize these simple ratios. One example is the *free-of-error* rating. This measure can be used in many ways and in many different contexts. For example, the context can be table, record or field. The *free-of-error* measure requires definition for what is considered to be the unit and what is considered to be an error. Another measure utilizing the simple ratio is the *completeness* measure. It represents the ratio of not-complete units to all units. (Pipino et al. 2002, 213.) These both measures are shown in figure 24 below.

$$free\ of\ error\ rating = 1 - (\frac{Number\ of\ data\ units\ in\ error}{Total\ number\ of\ data\ units})$$

$$Completeness\ rating = 1 - (\frac{Number\ of\ incomplete\ items}{Total\ number\ of\ items})$$

FIGURE 24. Free-of-error and completeness rating formulas (Lee et al. 2006, 55-56)

Min and *Max* operators are common for data quality dimensions that require aggregation of more than one data quality metric. It calculates the minimum or maximum value from a group of individual normalized data quality metrics. Example for usage of *Min* operator is the *appropriate amount of data* dimension. It is calculated by taking the minimum ratio from two ratios. The first ratio is the ratio of available units to the required units. The second ratio is the ratio of required units to the available units. This formula is presented in figure 25 below. (Pipino et al. 2002, 213-214.)

$$Appropriate\ amount\ of\ data = min\left[\frac{Number\ of\ data\ units\ needed}{Number\ of\ data\ units\ provided}\quad\frac{Number\ of\ data\ units\ provided}{Number\ of\ data\ units\ needed}\right]$$

FIGURE 25. An appropriate amount of data measure (Lee et al. 2006, 58)

3 DATA WAREHOUSE

Before Data Warehouses existed, users created reports and analysis by straight queries into the operative systems. Data in these systems usually is in relational databases which serve the needs of the operative system in question. There is an advantage when the database is directly queried, the data is real-time data. However straight queries can cause problems. Analysis of data requires large amounts of data from the operative system. This can cause serious issues to the performance of the operative system. (Linstedt et al. 2016, 2-3.) Another problem is that the data required for reporting is probably distributed between two or more operative systems and it is not easy to combine to form uniform data. For example, customer data can be stored in different systems for same or different uses in the organization as seen in figure 26. This makes it hard to create common report form this kind of data. (Hovi et al. 2009, 5.)

Customer (Source 1)

cutomer_key	customer_id	name	phone
1	10025	Organization Ab	000-123123
2	10026	Company Ltd	000-321321
3	10015	Nikon Marjakauppa Oy	

Customer (Source 2)

customer_id	name	address
10025	Organization Ab	Highway 1
10026	Company Ltd	Road 123
10015	Nikon Marjakauppa Oy	Ajotie 6

Customer (Source 3)

id	name	address	city
10025	Organization Ab	Highway 1	Stockholm
10015	Nikon Marjakauppa Oy	Ajotie 6	Vantaa

FIGURE 26. Data in different source systems (Hovi et al. 2009, 5)

Data warehouse is designed to support reporting and making analytics. Most common use for the data warehouse is to support needs of business

intelligence. Data warehouse combines multiple operational databases into one and offers uniform database. (Hovi et al. 2009, 14.)

Data sources for data warehouse usually are organizations operational systems, external systems and non-structured systems. Organizations operational systems are the most common data source and consist of ERP and CRM systems for example. External sources can be open data services or other services that are not managed by the organization itself. Non-structured data sources are sources that contain data that is not structured. This includes data like emails, texts or images for example. (Hovi et al. 2009, 18.)

The *Single version of truth* is one common requirement for data warehouses. It means that the organization has unified look into its data. In data warehouses there possibly is a need for more than one version of truth depending on the requirements and what is considered to be the truth by different departments in the organization. Good example about *single version of truth* is the previous example of different customer data in different operative systems. In this case, the single version of truth means that in the data warehouse there is only one customer even if it exists in multiple operative systems with different data. Customer data is cleansed, and the leading system is selected that creates the single version of truth as shown in figure 27 below. (Linstedt et al. 2016, 5-6.)

Customer

customer_id	name	address	city	phone
10025	Organization Ab	Highway 1	Stockholm	000-123123
10026	Company Ltd	Road 123		000-321321
10015	Nikon Marjakauppa Oy	Ajotie 6	Vantaa	

FIGURE 27. Combined customer data (Hovi et al. 2009, 17)

Purpose of the data warehouse is to serve the organizations reporting and analysis needs. Data in data warehouse must be easily accessible, understandable and trustable to the user. (Kimball et al. 2002, 2-4.)

3.1 Business need for data warehouse

Data warehouse offers many advantages for organizations. Different operational systems can be integrated into one place. This can be used to examine the different parts of the organization in a unified way. Data warehouse is also independent of business processes. There can be calculated, or derived information created to support reporting needs. This ensures that everyone is using a unified and single version of key figures. Information is also available easily and from a single point. (Hovi et al. 2009, 14-15.)

Data warehouse also supports the quick creation of reports that can contain data from multiple operative systems without causing load to them. Data is structured clearly and described so that it is easy to understand the data. The user does not have to be a technical person. Most common data warehouse users are business persons instead of IT –persons. (Hovi et al. 2009, 9.)

3.1.1 Business Intelligence

Business Intelligence (BI) means that the data is represented in a way that decision-makers understand it easily and it can be defined as delivering accurate usable information for decision makers in time to support effective decision making. Effective decision making is important in organizations and for this reason, they need business intelligence. Decision makers at a higher level in the organization needs to see the bigger picture and their role is to set long-term goals for the organization. Decision makers need a wide view into their area of responsibility. (Larson 2009, 16.)

The final goal for the data warehouse is that it is used in business intelligence. To achieve the goals of business intelligence the data in the data warehouse is analyzed and trends and patterns are looked from the data. These are then used to make decisions. BI applications show results of the analysis with tables and in graphical representations like charts and

maps. (Kozietski et al. 2009, 7.) Figure 28 shows a screenshot from a report created with Microsoft Power BI application.

FIGURE 28. Stock portfolio report created in Microsoft Power BI (Ferrari et al. 2016, 139)

3.2 Architecture overview

Implementation of the data warehouse can be done with three different architectures: one or more data marts, centralized enterprise data warehouse (EDW) or creating unified data marts. (Hovi et al. 2009, 26.)

In data mart implementation there are one or multiple different data marts over one or few operational systems. This can be seen in figure 29. This kind of implementation usually is small and used only for specific purpose. These purposes can be human resources (HR) or finance. Data mart architectures advantage is fast implementation. Disadvantages are that they are separate, and they do not support uniform reporting.

Organizations commonly drift into this kind of architecture instead of deciding to do so. (Hovi et al. 2009, 26.)

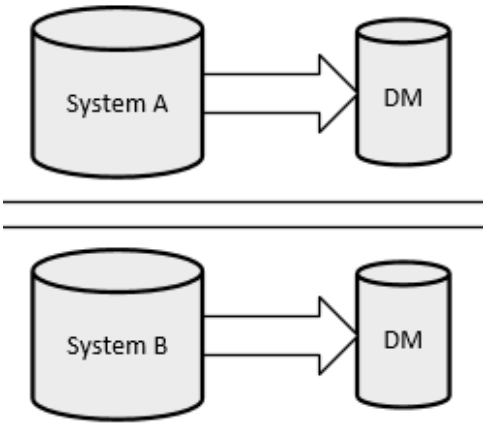

FIGURE 29. Different datamart architecture (Hovi et al. 2009, 26)

In centralized enterprise data warehouse (EDW) architecture the idea is to combine and integrate originations operational systems together. Its purpose is to gather data from different sub-sets of business and present it in a uniform fashion. Data in EDW is viewed at organization level over operational systems or organization limits. Queries for reporting are rarely done straight from EDW. Instead, data marts are built on top of the EDW to support reporting needs. Data marts can then contain subsets of the data from EDW and have additional calculations or derived information available for reporting. Architecture overview can be seen in figure 30. (Hovi et al. 2009, 27.)

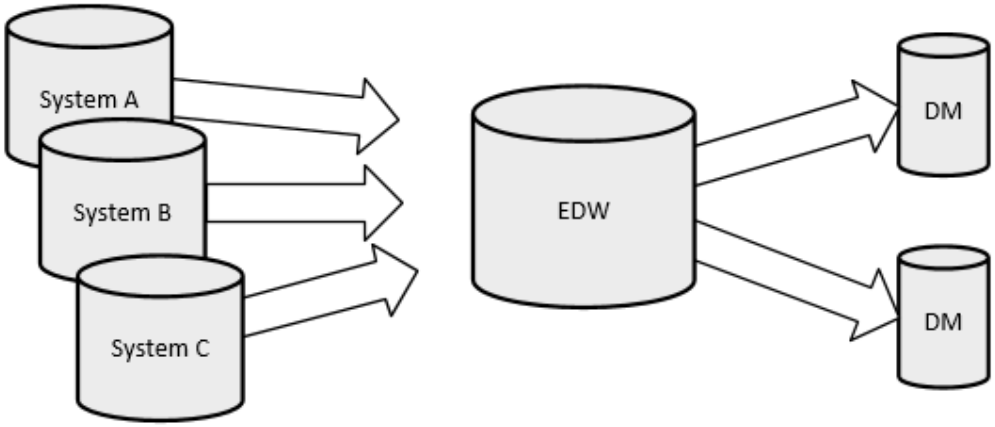

FIGURE 30. Enterprise Data Warehouse architecture overview

3.3 The development process of data warehouse

Data warehouse development project is like any other operative system development project. It is a process which includes projects, maintenance and follow-up development of the system. It is common that there will be changes during the development of the data warehouse caused by the rising possibilities of the data warehouse. For this reason, it is recommended to use the iterative model when developing the data warehouse system. The iterative model makes it possible to define the requirements before starting a new iteration. At the beginning of the data warehouse project, it is important to consider the bad quality data and the availability of the data in the operational systems. This is a common reason for delays and increased costs in data warehouse projects. (Hovi et al. 2009, 130.)

In the book Tietovarastot ja business intelligence (2009) Hovi et al. suggests following model as seen in the 31. Model is based on the model created by Ralph Kimball. When organizing the project, it advised keeping in mind that there might be many parties involved in the development process. Nowadays it is common that the operative systems are developed and maintained by third parties instead of the organization itself. For this reason, one expert from each of these parties is required for the data warehouse project. (Hovi et al. 2009, 130.)

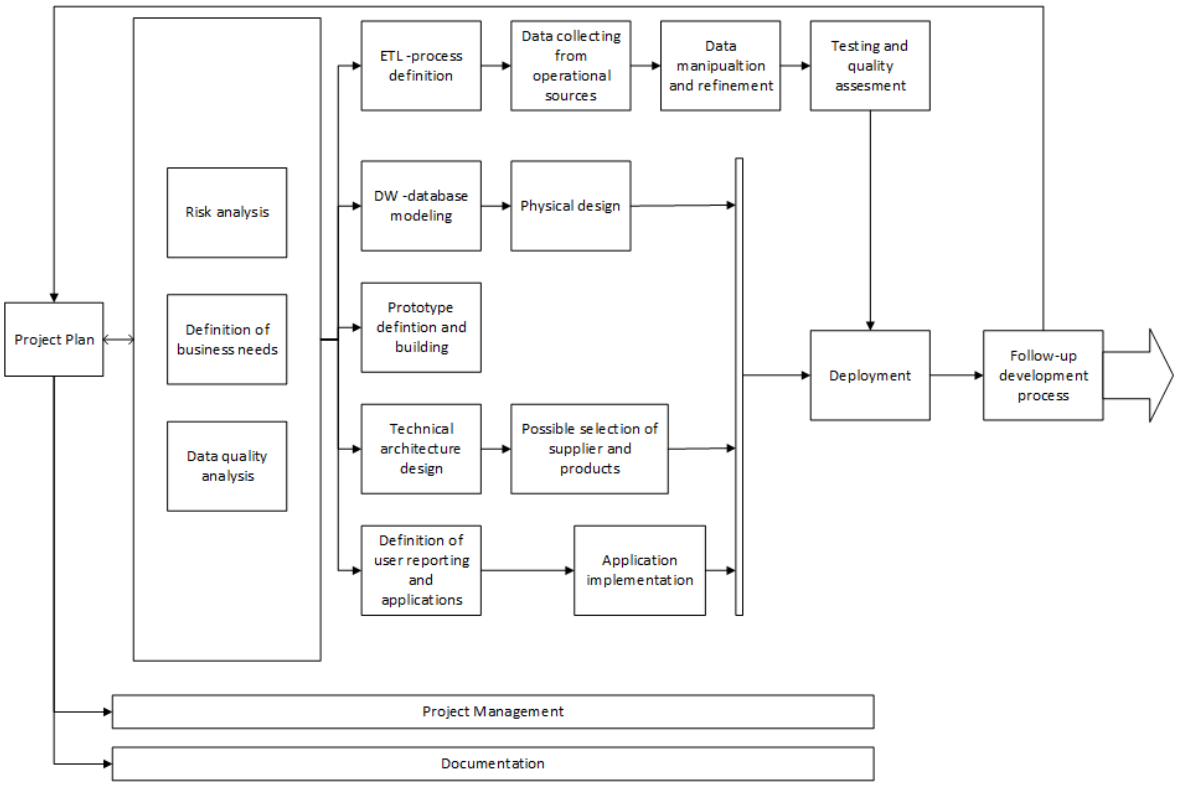

FIGURE 31. Data warehouse development model (Hovi et al. 2009, 131)

This model is suited for small and large data warehouse projects and it contains the whole lifecycle of the data warehouse. The model consists of areas that can be developed simultaneously making it possible to make the duration of the project shorter. It also favors iterative development methods which are important in high price data warehouse projects. It allows getting visible results faster. (Hovi et al. 2009, 131.)

In incremental development model, the data warehouse project is split into sections. These sections can, for example, be Human resources, finance, and production. First is developed one section and the systems and tools are tested at the same time. Then other sections follow one by one. This continues until there are no sections left and the data warehouse project is ready. (Hovi et al. 2009, 132.)

An iterative method is a key concept for successful data warehouse project, unlike traditional Big Bang method. Dividing the implementation into phases gives two advantages for end-users. Firstly, uncertainty and

insecurity are lowered when business problems are solved one by one instead of solving all of the problems at once. Secondly, it gradually raises knowledge and believability in the organization instead of offering a lot of untested information at once for the organization. In addition, iteration supports project management in many ways, like learning from the previous iteration and to see earlier the benefits gained by the customer from the system. This helps the development team to believe in their work. It is advised that the project should be defined as ambitious as possible and after that, it should be split up into smaller manageable pieces. Figure 32 shows an overview of the iteration process and common phases in the data warehousing projects. (Dijcks 2004, 18-20.)

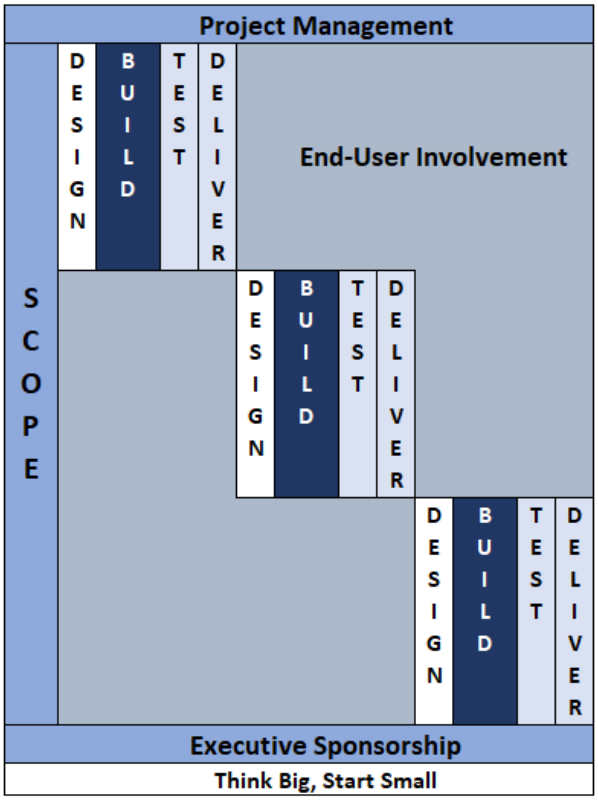

FIGURE 32. Iteration and common phases of data warehouse project (Dijcks 2004, 20)

One of the major issues in data warehouse development is how to select what data to load into the data warehouse. It is common that the developers guess what data is needed in the data warehouse. To support

this guessing the developers map out the requirements by interviewing the clients. For this reason, it is common that when the data warehouse has been completed that it lacks some information for some needs and it probably has some information which is never used. Disk capacity is cheap nowadays and thus does not limit the amount of information loaded into data warehouse however unneeded fields and tables slow down the loading, processing, and querying of the data warehouse. This issue could be avoided by collecting all the queries from all the applications that use the data warehouse. From collected queries, the used tables and fields could then be extracted. This would provide the list of required data. In reality, this is difficult to do before creating the data warehouse. Nevertheless, this data could be used to fine-tune the data warehouse afterward by removing the unnecessary data from it. (Kim 2002, 43.)

3.4 Data quality in data warehouse

Data quality has a big impact on data warehouse even though the data warehouse is rarely the reason for the bad quality of data. Operative systems are the most common reason for bad quality data in data warehouses. Data quality should be thought about before the start of data warehousing project. Challenge is to get the time and resources required to do any data quality research at this stage of the project. This means that additional section for data quality evaluation should be added to project plan before the start of planning. This is commonly entirely left out from the data warehousing project and it affects the later phases of the project causing delays to deliver erroneous data for the end-user. (Dijcks 2004, 21.)

Data quality issues rise in the testing phase of the data warehouse at the latest. Data in the report is wrong and cannot be published. It is common that the missing data is data that is not required to input in the operational system and thus usually skipped by the person responsible for inputting it. Data that is not inputted can’t be shown in reports nor can the wrong figure become true in the data warehouse. Data warehouse project has come a

long way when it hits the testing phase and if it is then when the issues of data quality are first encountered it will usually cause delays for the whole project and possibly more costs. For this reason, it is advised that data quality of the source systems should be surveyed before the data warehouse project. Data in a data warehouse should be never changed. It should be the same data that is in the operational system. If the data is changed, then the data in data warehouse would be different than the data in the operational system. (Hovi et al. 2009, 68-69.)

Some stages of the data warehouse are more susceptible to data quality issues than other stages. These stages are responsible for the final data quality of the data warehouse. Figure 33 shows the stages of a data warehouse that are susceptible to data quality issues. (Singh et al. 2010, 42.)

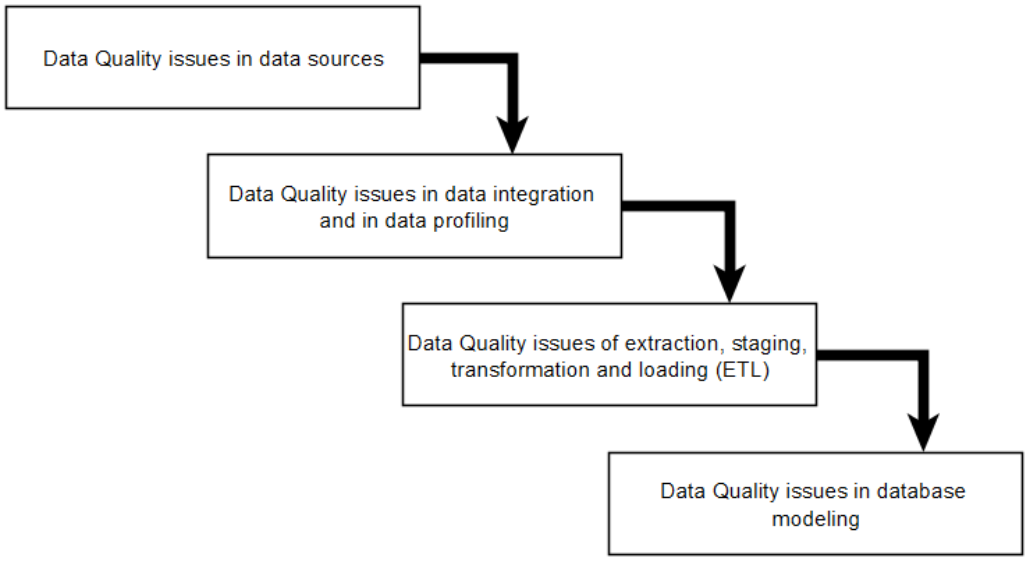

FIGURE 33. Stages of data warehouse susceptible to data quality issues (Singh et al. 2010, 42)

Loading of wrong or bad quality data from the source system is one of the most common reasons which leads to failure of the data warehouse project. Source systems have multiple different ways to store information, some of these are more cooperative than others. This diversity causes that

different source systems have a different kind of data quality issues. Some legacy systems do not store any kind of metadata that would describe them, for example. So-called dirty data coming from source systems usually originates from erroneous data inputted by human or from erroneous data update by the application. It should also be noted that some of the data to data warehouse comes from text files and from Excel files. It is almost certain that some of these files are manually created by combining multiple files. (Singh et al. 2010, 43-44.)

To solve these data quality issues in data warehousing projects there are two possible places where it can be done, in the source system or in the data warehouse. Generally, it is common that the development team does not have any means to fix data quality issues in the source system, instead, they have complete access into the data warehouse. How the data quality issues are solved depends on available technologies and resources. Before solving these issues, the required business rules to solve them needs to be defined. It also should be noted that the fixing of data quality issues Is not a nonrecurring procedure but instead a continuous process. Data quality changes over time and it can be affected for example by new data sources. (Dijcks 2004, 22.)

Data profiling is an essential part of data warehousing even though it is usually not done and by doing so the data quality of data warehouse is compromised. Fast data profiling of source system should be done immediately after data from the source system is identified required and it is needed to be loaded into the data warehouse. Staging and ETL -phase is crucial regarding data quality in the data warehouse and it is the key place for validating the data quality of source systems. (Singh et al. 2010, 46.)

3.5 Implementing data quality into the data warehouse

In the development of the data warehouse, the designers need to take into account all the data quality requirements of different stakeholders. For this reason, the development team must understand the data quality

perspectives which are relevant for each of the stakeholders. Table 3 lists roles of the stakeholders and the possible data quality issues for them. (Kumar et al. 2013, 62.)

TABLE 3. Data quality issues for stakeholders (Kumar et al. 2013, 62)

Stakeholder	Role	Data Quality Issues
Decision Makers	Final users who uses reporting tools, OLAP, Data mining to get answers to their questions	Overall quality, ease of access, reports in desired format and timeliness
Data Warehouse Administrator	Keeps data warehouse properly operating	Error reporting, timeliness and metadata accessibility
Data Warehouse Designer	Designs of data warehouse architecture	Schema design, metadata quality design and software quality design
Data Warehouse Programmer	Developes actual data warehouse applications	Implementation quality, overall software quality, metadata quality
Executive Manager	Concerned with financial information regarding data warehouse	Keeps a check on costs, benefits and return on investments

They continue by suggesting a conceptual framework for managing data quality in data warehouse systems. This framework can be seen in figure 34 below.

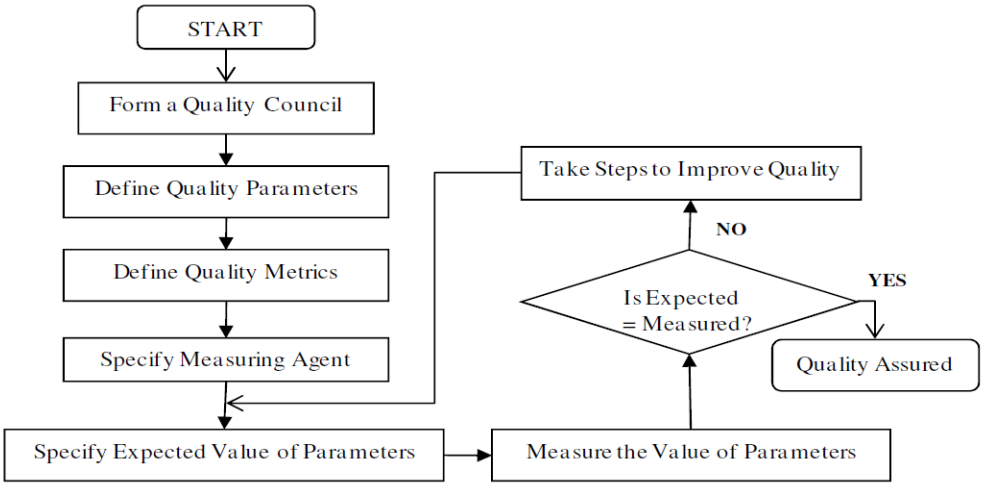

FIGURE 34. A conceptual framework for data quality measurement in the data warehouse (Kumar et al. 2013, 62)

In data warehouses, the data flows always from operational systems through staging area into the data warehouse. Data quality is always in danger when data is extracted, integrated, cleansed, changed and loaded into the data warehouse. All these stages are potential sources for data quality issues and for this reason, all of these stages should be monitored to catch the potential data quality issues. (Kumar et al. 2013, 63.)

They continue with proposing a metadata-based quality model which is shown in figure 35. In the model, there is so-called *quality goal* for each of the stakeholders. These quality goals are abstract requirements defined on data warehouse objects and they are documented for a purpose in which the stakeholders are interested in. The model contains *quality dimensions* which are used to abstract the different aspects of quality. Quality goals are associated with one or more *quality queries*. Quality query defines if the goal is reached or not. The quality query is defined for the *quality metric* which in turn reflects the measurement of the quality and it is defined for specific data warehouse object. The quality metric also defines the interval of expected values within the domain and it also includes the actual value for given point in time. Simple software agent measures the values of the quality metrics. (Kumar et al. 2013, 67.)

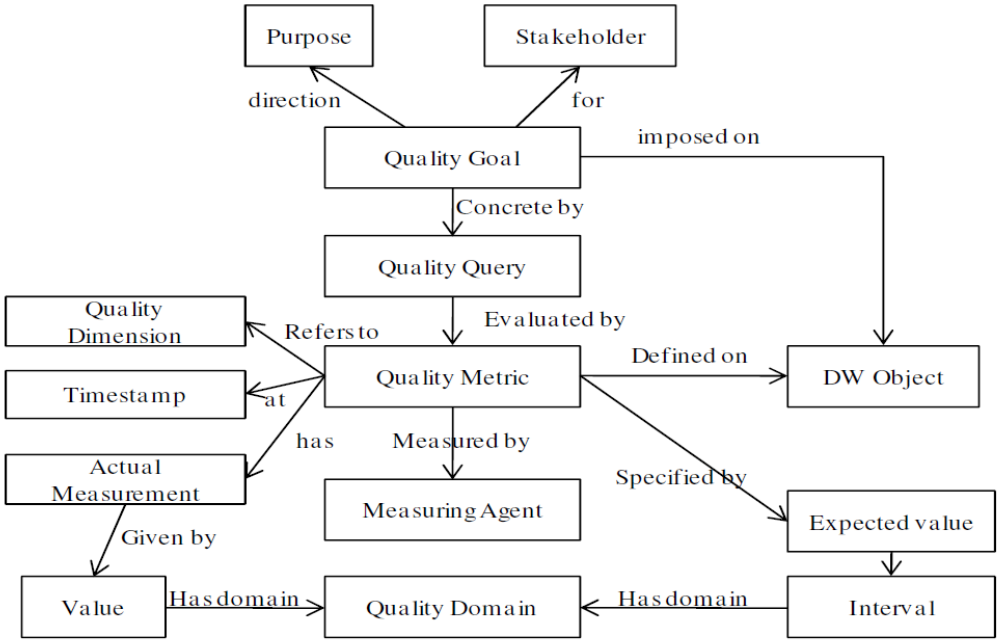

FIGURE 35. Quality meta-model framework proposed by Kumar et al. (67)

Helfert et al. propose an architecture for a metadata-based data quality systems in their paper Proactive Data Quality Management for Data Warehouse Systems. This architecture is shown in figure 36. The system covers the whole data warehouse from operational systems all the way into analytical applications. It measures the data quality while the data flows through the data warehouse system. Metadata is the key aspect of the system and the metadata of transformations, processes, and data schemes are most important. Most crucial part of the concept is the integrated metadata management component which stores all the information regarding data quality. (Helfert et al. 2002, 4-5.)

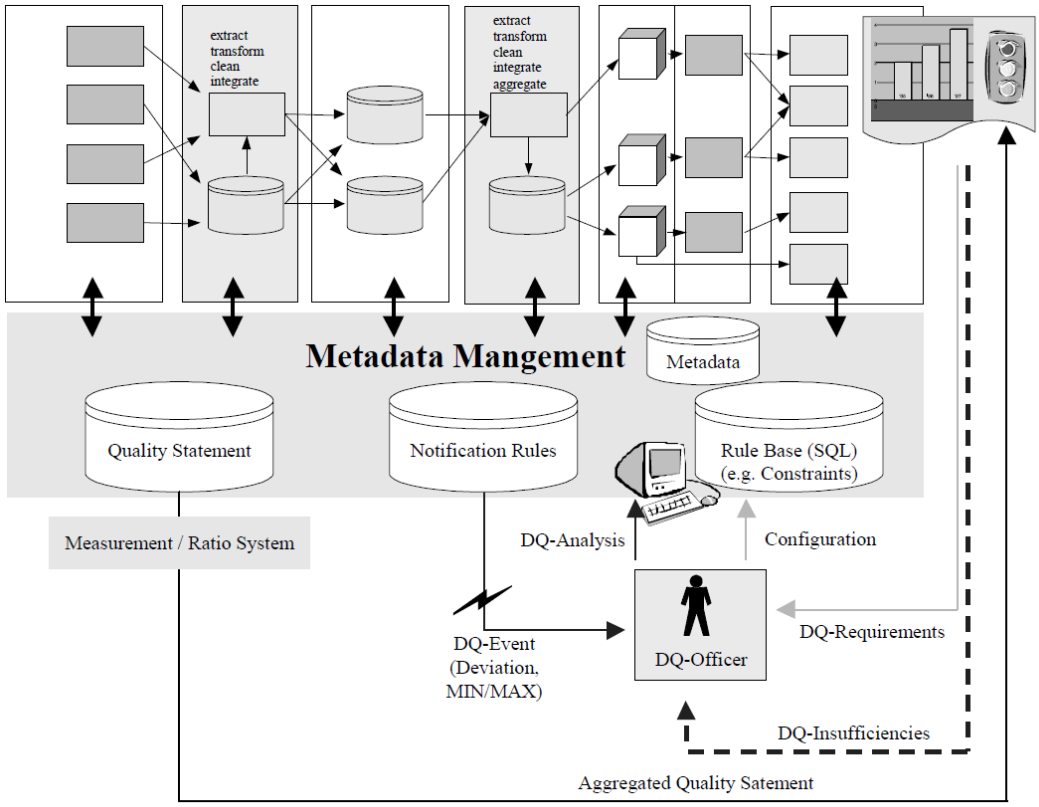

FIGURE 36. Architecture for metadata-based data quality system (Helfert et al. 2002, 5)

4 INTERVIEWS

4.1 Semi-structured interviews

Semi-structured interviews were conducted for client and supplier. A major part of the interviews was done via Skype calls. Few of the interviews were held as face-to-face interviews. All interviews were recorded, and every interviewee was informed beforehand about the recording and confidentiality of the interview. There were separate interviews for persons in client and supplier sides. These two interview templates were similar, and the only major difference was the different point of view in them. Both interview templates can be seen in appendices 2 and 3.

Interviewees were mainly selected from already known people, some snowballing was used to get more candidates for the interviews. Snowballing means that the interviewees suggest persons for interviewer for candidates. (Hirsjärvi et al. 2016, 59-60.) When doing the interviews, it was noted that when the person was not known before contacting, they rarely replied anything to the request for an interview.

Almost all of the recorded interviews were transcribed on the same day. Few that was not was transcribed at the next day at the latest. The questions were tested beforehand by conducting test interviews for both interview templates as suggested by Hirsjärvi et al. (2016, 72-73). Changes were done to the interviews if some issues or need was identified in the test interviews. Test interviews were also used to estimate the length of the interviews and the time needed to transcribe them. (Hirsjärvi et al. 2016, 72-73.)

In both interviews the interviewees were asked about their current role and how many years they have been working in a similar role. These questions turned out to be irrelevant. Reason for this was that the roles of interviewees differed a lot which means that the years of experience in the role is not comparable in any way. Answers to these two questions are not

analyzed. A better question would have been: how many years you have been working with data warehouses, for example.

The length of supplier interviews were 12 minutes on average and the length of client interviews was 16 minutes on average. Transcribing the interviews took roughly twice the time of the interview.

Content analysis was conducted to the transcribed material and the material was coded.

4.2 Supplier interview results

There were in total 15 interviews for people working in supplier side. The interviewees were from four different companies. All of the interviewees were working with data warehousing and business intelligence projects. Roles of the interviewees varied from Business intelligence consultant to Sales manager. Three (20%) of the interviewees were females and 12 (80%) were males.

Each of the interviewees was asked to approximate the number of data warehousing projects that they have been part of. Figure 37 shows the approximated count of data warehousing projects.

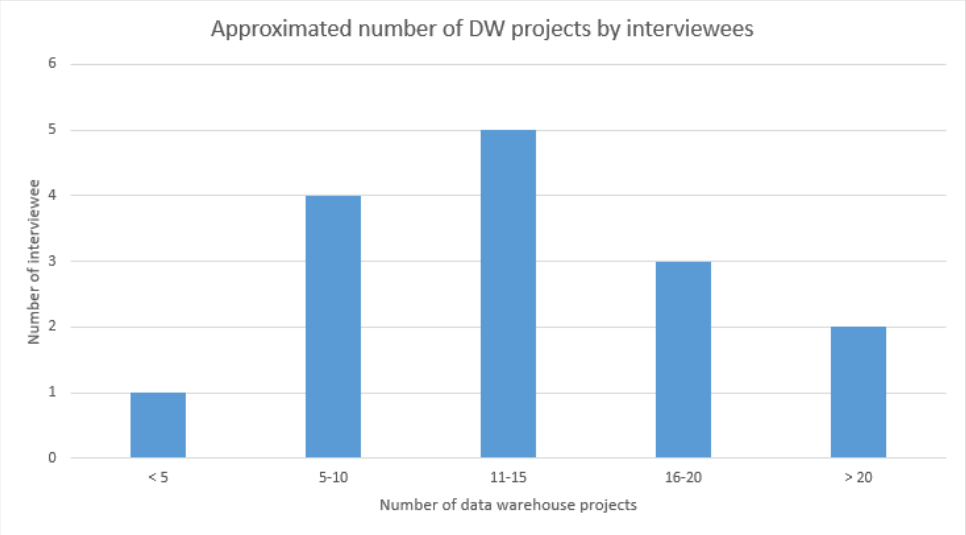

FIGURE 37. Approximated count of data warehousing projects

The interview consisted of 11 questions and these are presented next. Questions were grouped into following groups:

- Meaning of data quality
- Data quality impact to data warehouse projects
- Data quality issues
- Data profiling and data quality analysis
- Client aspect

4.2.1 Meaning of data quality

Interviewees were asked what data quality means for them. Two of the most common answers was that the data is correct and that the data is trusted. Four interviewees from 15 (26,7%) said that it means that data is correct and four from 15 (26,7%) said that it means that data is trusted.

> *"First that it brings to my mind is the correctness of data. More correct the data is, better the quality and it is more usable in the future."*

> *"Trust, it is what comes first into my mind about data quality. On the other hand, there are many kinds of data quality, is the technical quality and the quality of data content. Trust applies to both."*

The second most common answer was that data must be usable. This attribute was said three times.

> *"The fact that data, in general, is usable, it must have a certain level of quality to be usable."*

> *"It means that the data can be used to make decisions."*

Generally, the interviewees had a good understanding of the meaning of data quality. It should be also noted that almost every interviewee said more than one attribute when describing data quality. Figure 38 below shows all the identified attributes and their occurrences.

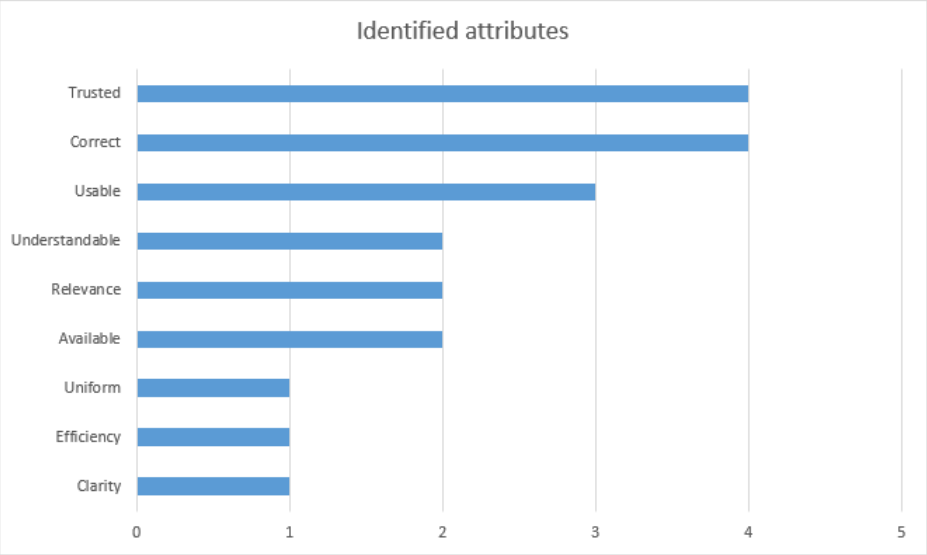

FIGURE 38. Identified data quality attributes

4.2.2 Data quality impact to data warehouse projects

When interviewees were asked about the impact that data quality has to data warehousing projects the most common answer was that the data quality is the foundation of the data warehouse. Seven interviewees from 15 (46,7%) said this. Other interviewees focused more to describe the impacts to the project itself when data quality is bad or when it is good. When interviewees talked about the effects of bad data quality to data warehousing projects the most common effect was that it increases the workload, six (40%) interviewees noted this. The second most common effect was that it lengthens the project and it compromises the whole data warehouse project. These both were said three times (20%) by the interviewees.

> *"Data quality is one requirement for the success of the project. If you put bad data into the machine, then you get bad results. Without quality, you don't need to do this business very long."*

> *"It is the most significant thing. If bad data goes in it can't be made any better."*

> *“I'll see it in a way that if the data quality is not good and this is not considered when estimating the workload or that the data quality comes as a surprise it will in a long sight also lead into customer dissatisfaction and even into that the project is not profitable.”*

One interviewee noted that the data quality itself does not necessarily have meaning in data warehouse projects. It all depends on how well the client understands the current state of the data quality:

> *“It depends very much on how well the client is aware of their data quality. In fact, the data quality is not necessarily a good or bad thing. Of course, if everything is good the life is much easier for everyone, but if there are problems with data quality, and there usually is, then the questions is if the client understands the condition of their data and if the objectives for the project are realistic.”*

It is good to note also here that many of the interviewees listed more than one impact that data quality causes to the data warehousing project. In figure 39 below there are listed all the effects that were mentioned by the interviewees to the data warehousing project when data quality is poor.

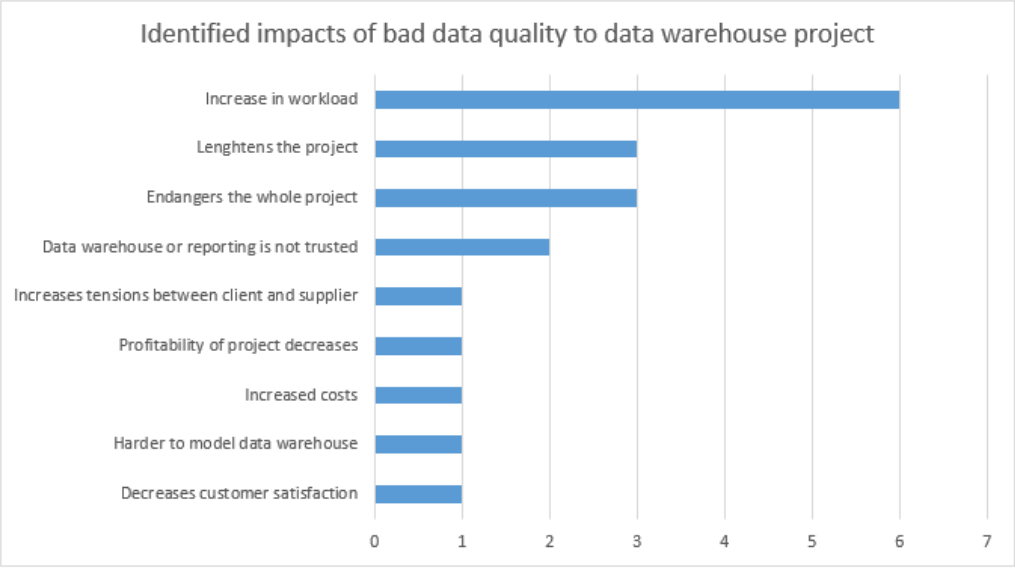

FIGURE 39. Effects of poor data quality on data warehousing project

Six (40%) of the interviewees did mention effects when data quality is good. The most common effect that interviewees told that good quality causes were that it eases the development which was mentioned by five (33,3%) of the interviewees. One (6,7%) interviewee told that good data quality minifies the maintenance required by the data warehouse.

> *“If quality is good then the work is straightforward. Everything is easier and smoother.”*
>
> *“Of course, if the data is complete it is easier to make these projects.”*
>
> *“Sometimes there has been really good quality data and everything went smoothly forward.”*

4.2.3 Data quality issues

Interviewees were asked if they had faced any data quality issues when developing data warehouse or been part of data warehouse project. All interviewees (100%) had faced some sort of data quality issues. Then the interviewees were asked to list common data quality issues that they have faced in data warehouse projects. Most common data quality issue was the erroneous user input in source systems, nine (60%) interviewees said this.

The second most common issue faced was missing or not complete data. Six (40%) of interviewees mentioned this. Also, in here it was common that one interviewee mentioned more than one issue. Figure 40 below shows all issues that interviewees mentioned they had faced in data warehouse projects and the occurrences of these issues.

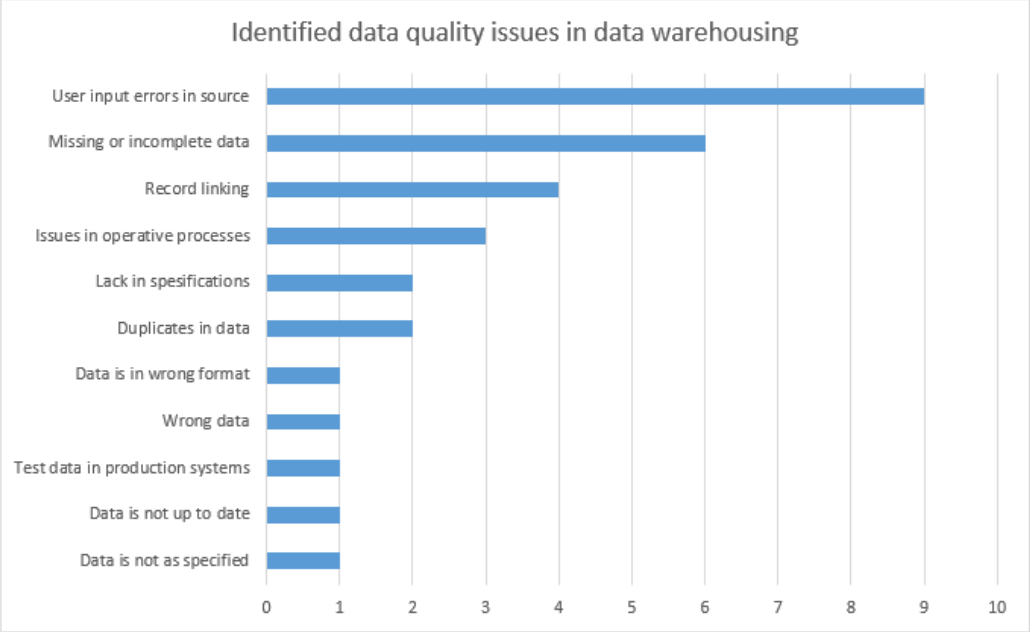

FIGURE 40. Identified data quality issues mentioned by interviewees

"Sometimes the data is missing and sometimes it is inputted in the wrong format. Someone might have thought that I'll just put a question mark in place of a number when the correct number is not known, and it can be that thing which crashes the whole execution when the system awaits a number, but the value contains characters instead."

"The excessive manual work that one would think should already be get rid of in this millennium and in this century. It is a bummer that in many cases there are elegant data warehouse and reporting system but still one single badly managed manual process at the start can ruin the whole pipeline."

"In support tasks especially almost, every issue is somehow related to that the data is distorted or data quality is poor, somehow different than specified. It can be as simple that the data type is completely wrong and differs from which is agreed. Data that is someway different what is agreed about is the very common reason for issues."

“Incomplete data, some columns have data others does not.”

Interviewees were also asked to tell which of the data quality issues cause the most impact to the data warehousing project. The most common answer was recorded linking from multiple sources is the issue that causes the biggest impact, this was mentioned by four (27%) interviewees. All the mentioned issues with the most impact and their occurrences are visible in figure 41 on the next page.

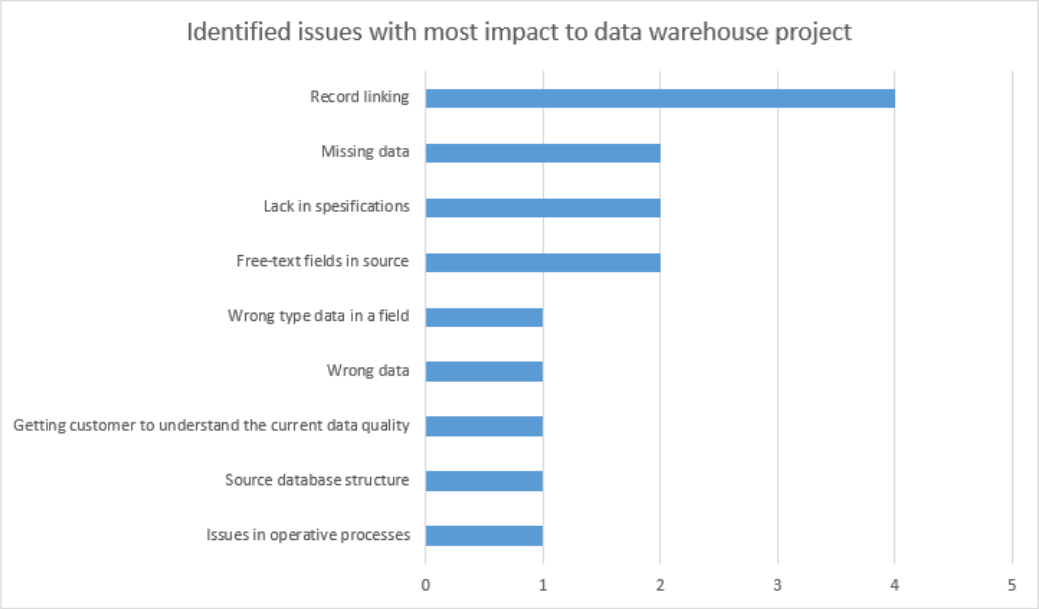

FIGURE 41. Identified issues with most impact

“Quality issues affects the schedule and workload. The third is that the data reliability in reports suffers and if these issues cannot be fixed it causes that the users stop using the reports because they cannot trust them.”

“If the belief to own data is so strong that they do not understand that the data is incorrect then it takes an awful amount of time to get everyone to understand the current state of data quality.”

4.2.4 Data profiling and data quality analysis

Interviewees were asked what their thoughts were about basic data profiling before or at the beginning of data warehouse project. A short explanation about what the data profiling means was also given by the interviewer to the interviewee. 14 (93,3%) interviewees kept the idea of data profiling as a good thing and that it would help the project. One (6,6%) interviewee did not find it as a good thing but did not also find it as a bad thing.

> *“Is important and I have done it. In one project we did not have documentation about data and we did not know which fields identify one record, so we started to profile it. After all, there were no fields that could be used to identify one record.”*
>
> *“As such a good thing, but is it realistic, that I cannot say.”*
>
> *“It would be damn good. It would give a perception of the current data quality quickly. I could see myself using this kind of tool in the start of every project and it should be included in the project work estimates.”*

Next, the interviewees were asked what they think about simple data quality analysis before or at the beginning of data warehousing project. Again, a small explanation what the data quality analysis means was given to the interviewee by the interviewer. All (100%) of the interviewees kept this as a good thing. Seven (46,7%) interviewees did bring up potential issues with the most common issue being the challenging implementation of data quality analysis.

> *“Yes, this would be very helpful. Simply it would give a reference to what should be done before actually doing anything.”*
>
> *“The viewpoint is excellent. I do not see it as a bad thing, but it might be hard to implement?”*

“It would likely find more issues relating to reporting which is the most important layer for end-user. They do not give a cent about what is in the data warehouse. If the value is wrong in the report it does not matter what it is in the data warehouse.”

“Not bad idea at all. It would allow us to caught common issues at that point and we could react and fix the data before it is being received by the data warehouse. It would solve a lot of problems before the actual development work.”

4.2.5 Client aspect

Question about if any data profiling or data quality analysis were offered to clients was asked from interviewees. Nine (60%) told that nothing like this was ever offered to clients that they know of.

“As far as I know we have not offered any data profiling.”

Two (13,3%) said that sometimes something little bit like this has been offered to the client. Four (26,7%) of the interviewees did not know or did not answer this question.

“Yes, one project. We did pre-project investigation where we looked for the data required and at the same time we tested relations and did investigate the data. We did it at the same time but not consciously.”

“Yeah would be probably pretty good and actually something similar was done in one projects. We did these reconciliation reports that were not related to business but instead they were used to analyze the data quality and data uniformity”

In addition, interviewees were queried if there has been any talking about data quality with clients before the start of data warehousing project to

which nine (53,3%) said that no or rarely. Rest seven (46,7%) told that data quality is always or often discussed with the client.

> *“It is spoken and questioned out loud if the clients’ data is in shape. Also, tried aloud to bring up what kind of project model should be used. Clients usually want fixed price or target priced projects, we try to argue against it because for example if there is uncertainty about data quality, it increases the risks when considering these kinds of contracts.”*

> *“Not in the projects that I have been in. Data is given like, here is the data study and load it into the system.”*

> *“No, almost in every offer we make the offer contains closure that problems related to data quality are not within the scope of the project. But in practice it cannot be completely ignored, if correct values are not available because the data is awry then the project has to take stand in some way.”*

The last question asked from the interviewees was that what kind of view the client had about their data quality in which 10 (66,7%) of interviewees answered that the client usually has a good view about the quality of their data, but not necessarily the correct view. Four (26,7%) interviewees told that the client is always surprised about the actual data quality or that they do not have a uniform view of the quality of their data. Two (13,3%) told that the clients have optimistic view of the quality of their data.

> *“What I know usually the client have had good knowledge and understanding of their data quality. The reporting system is also used to assist in finding the problems.”*

> *“Client usually have assumption if the data is okay or not. Most honest ones say straightly if the data is not in order. Few of the clients know the actual level of data quality in the end. There is probably a gut feeling about the data quality but rarely any facts about it. Profiling would help in this.”*

“Probably client does not have an appropriate view on their data. They do not necessarily even have tools which they could use to look the data and examine the data quality.”

“Yes, well they do have usually knowledge what the quality of data is, although not always. Usually pretty good understanding.”

4.3 Client interview results

There were in total 10 semi-structured interviews for people working on the client side. The interviewees were from seven different companies. Interviewees worked in many different roles from mathematician all the way to financial manager. Interviewees all worked with data warehouses as a user or as an organizations data warehouse developer. Four of the interviewees were female and rest six were male.

Semi-structured interviewees consisted of 13 questions and these are presented next in this chapter. Questions were grouped into following groups:

- Meaning of data quality
- Data quality issues
- Data quality impact on data warehouse projects
- Data profiling and data quality analysis

4.3.1 Data quality

Interviewees were asked what data quality means to them. The most common answer was that the data must be trusted. Five (50%) of interviewees mentioned it. Next common answer was that the data must be correct which was said by four (40%) interviewees.

“Quality means that the data is correct. It is also the foundation for everything that the data can be trusted and when the data is good quality data then it can be trusted.”

> *“This reporting is like a trust business in a way that what is shown in there it should never be wrong. If it is wrong, it quickly overturns the credibility of the whole system if it has even a few problems and hence the data quality is the backbone of these kinds of systems.”*

> *“Data is correct, and it is in a form that can be easily used and that it is up to date.”*

The third most common answer given was that the data must be up-to-date. It was mentioned by three (30%) interviewees.

> *“In a common level data quality means that the data is up-to-date and correct. Correct in that way that it fulfills the definitions what has been done regarding it.”*

Client-side interviewees also had a good understanding what data quality means and they usually told more than one attribute to describe the data quality. Figure 42 below shows all the identified data quality attributes from the interviews.

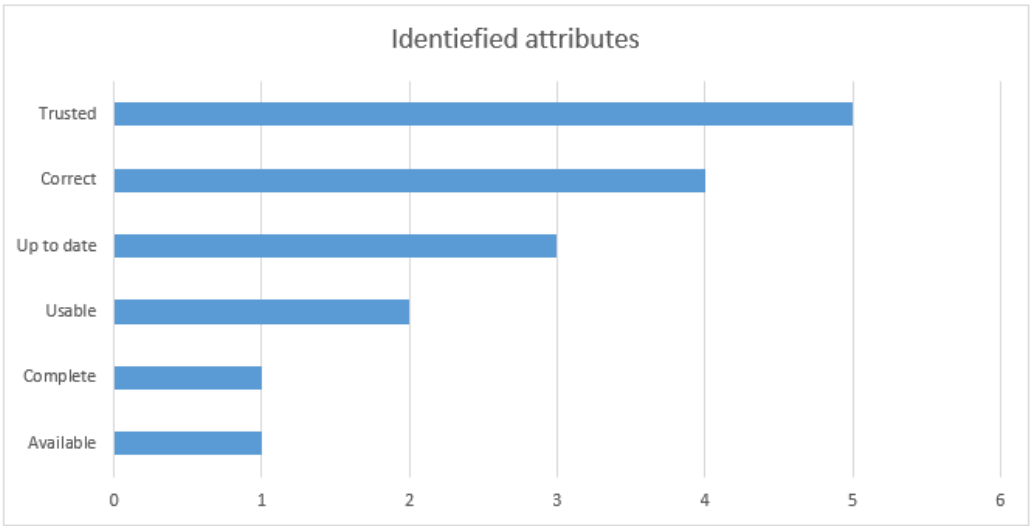

FIGURE 42. All identified data quality attributes

4.3.2 Data quality issues

When interviewees were asked if their organization has any data quality issues in their data warehouse or in other systems, nine (90%)

interviewees told that there are issues. One interviewee did not know of any issues currently in their systems.

> *“Yes, there is”*
>
> *“Well, there is probably always some problems”*
>
> *“Nothing comes to mind, I myself am not working at system level every day.”*
>
> *“Well, yes, I must admit that there are issues currently”*

Interviewees were asked to list issues which they have faced. Most common of these issues were technical issues which were mentioned by four (40%) interviewees. Technical issues were mainly issues caused by hardware, issues with metering devices. Issues with limitations in databases that caused issues were also counted as technical issues.

> *“The challenge is that when you begin to combine data from the operative systems there is a bit of so-called inaccuracy in measurements which is caused by old kind measurements.”*
>
> *“If the information is transferred from one system to another, the format changes so that the comparison is then a bit of hassle when the format needs to be changed between the comparisons.”*
>
> *“For example, coping with that the numbers have been entered incorrectly and the basic systems agree to accept those incorrect inputs and data. Some difficult data types, but it's not a quality problem.”*

The second most common response was missing data. It was mentioned by three (30%) interviewees. Figure 43 shows all the issues faced by the interviewees.

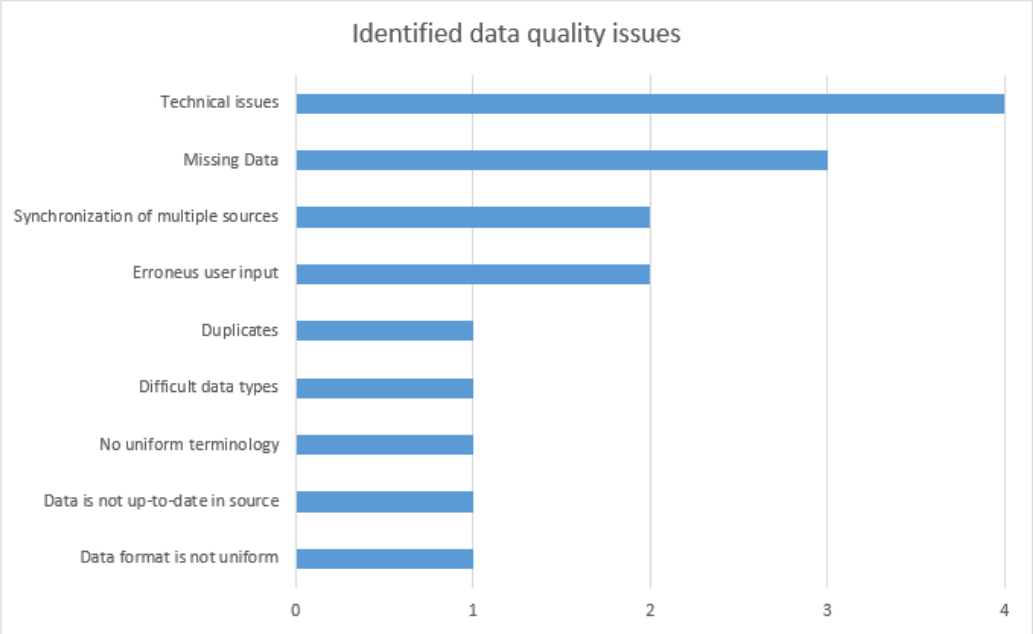

FIGURE 43. All identified data quality issues

Next, the interviewees were asked which of these issues caused the biggest impact to their work or to work of others. Technical issues were mentioned most, three (30%) of interviewees mentioned it. The second was missing data and manual correction of data and validation. These two were both mentioned two (20%) times.

> *"Older measurement techniques or that the reading of measurement is failed. But mainly the problems with systems like that there are spikes in the data time to time that are unrealistically large."*

> *"Well, maybe this missing data is the biggest thing. Many systems also accept that some data has not been entered. Sure, for its own system, it seems that everything is ok, but then when we try to combine and use the data more widely, then there are problems."*

> *"Data types, timestamps, and dates. Now that they are not in the same format in the data warehouse and in the production system. Now, however, when these must be used in parallel, it is that every program has to be coded up to two different versions."*

Figure 44 shows all the issues that cause the most impact to interviewees work or to others work.

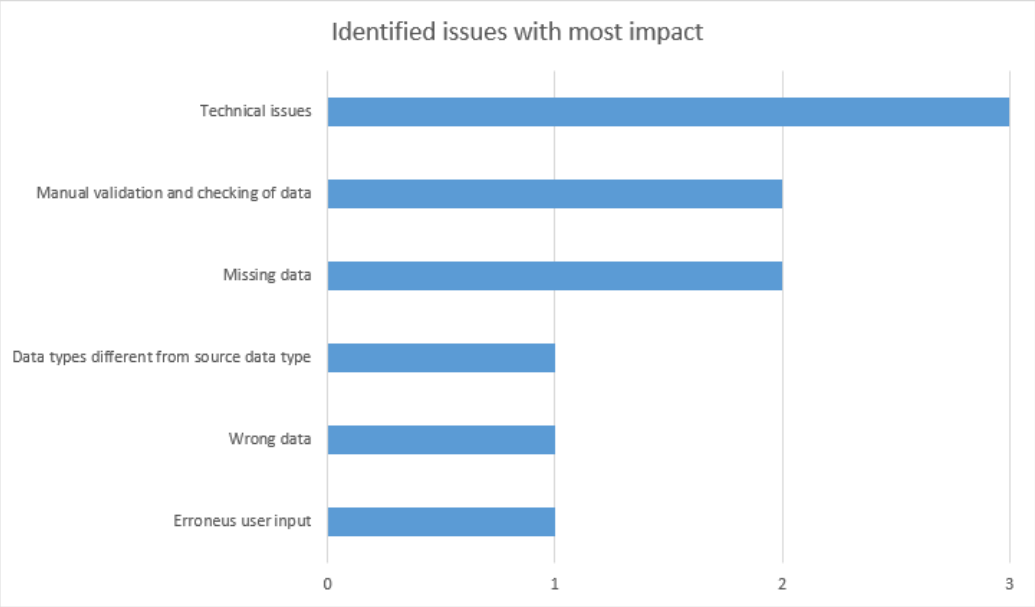

FIGURE 44. Identified data quality issues with the most impact to client

Interviewees were questioned about what they think causes these data quality issues. The most frequent answer was again the technical issues which were answered five (50%) times. Second frequent answer were issues in organizations processes and in information flow between organization units. This issue was mentioned three (30%) times.

> *“As a single big issue is some resource issues and timeout problems.”*
>
> *“Typically, there has been a break in information, so it has not come to us to know that something has been changed and it has not been taken into account. The course of information is certainly the main reason for such situations.”*
>
> *“The basic reason is that over the silo's occurring processes, so then comes these data quality problems.”*

One of the interviewees mentioned that the supplier has been at least once the cause of data quality issues.

> *"Yes, these problems caused the supplier quite obviously. It is difficult to understand why the data cannot be in the same format as in the source system."*

In figure 45 below all the causes that the interviewees mentioned are shown.

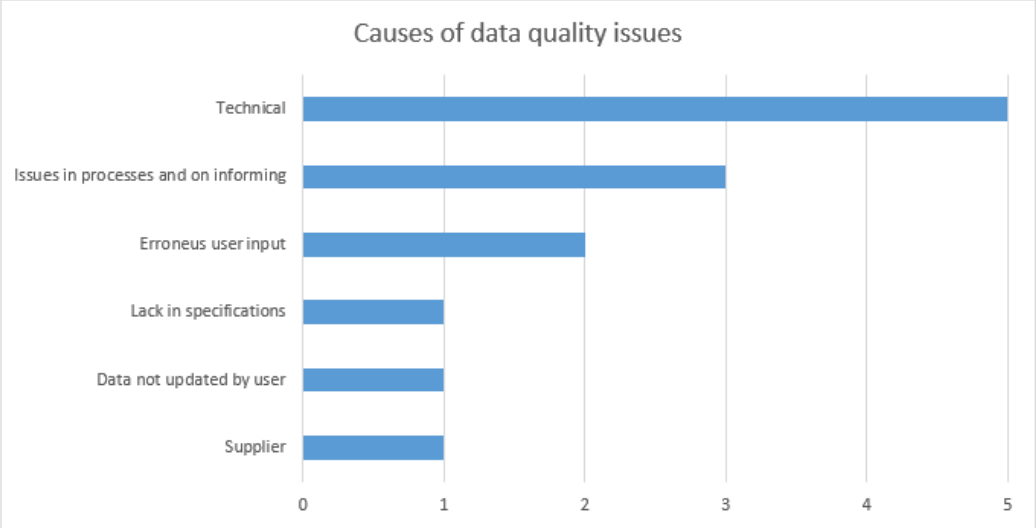

FIGURE 45. Identified causes of data quality issues

4.3.3 Data quality impact on data warehouse projects

Interviewees were asked how they see that data quality affects data warehousing projects. The most frequent answer was that the data quality is an absolute condition for the data warehouse. Five (50%) of the interviewees thought this. The second most frequent answer was that bad data quality increases the workload on their side (client side). This was mentioned two (20%) times.

> *“Of course, everything that makes use of data in data warehouse, problems that affect the quality of data and the data quality, so that is a really big deal! That is, our other systems that use data in data warehouse, so they trust that the data is correct and that it is reliable and unchanged.”*

> *“I think it's absolutely unconditional. The user must be able to rely on it enough if not 100% at least 99% or more.”*

> *“For me, it means quite a lot of integration and patching work.”*

> *“Well, at least we have the effect that when data that is in our systems is transferred to the data warehouse and if the data is rubbish in the source system, it will immediately affect the quality of the reports.”*

Other mentioned impacts were the increase in costs and that the data warehouse does not fulfill the expectations or that it will not be trusted.

> *“It has an impact on costs and the way that the data warehouse solution is not trusted.”*

Figure 46 shows all mentioned impacts of bad data quality to data warehouse projects.

FIGURE 46. Impacts of bad data quality to data warehouse projects

4.3.4 Data profiling and data quality analysis

The interviewees were asked if they do any data quality analysis in their organization in which eight (80%) answered that they do some sort data quality analysis. Two (20%) that they do not do any regular data quality analysis.

> *"In a way, it is done in the sense of surveying the production databases and searching for potential problems through data warehouses. Testing that the information in the production systems is correct so that it does not flow forward to end-users or reports."*

> *"Yes, we do. We compare data in source systems against data in the data warehouse."*

> *"No, nothing regular that would create something that could be compared."*

When interviewees were asked their thoughts about simple data profiling before or at the start of data warehouse project. Eight (80%) interviewees kept it as a good idea and the rest two (20%) did not see it as helpful but did not see it as a bad thing either. The interviewer told briefly what the

data profiling means to the interviewees.

> *“It would be good, and it should be done at the beginning of every project. Trusting that data as such as someone imagines it in a table or file is not really enough.”*
>
> *“Definitely, it would be good and necessary”*
>
> *“It depends on the project that would this help anything.”*

Next, the interviewees were asked what they think about simple data quality analysis before or at the start of data warehouse project. All (100%) of the interviewees kept it as a good thing. Three (30%) interviewees noted some potential issues like hard implementation of the data quality analysis.

> *“Some data field has long been used in a certain way or left unused, so it causes problems if afterward it is decided that some data is mandatory, so it may be difficult to get the missing data into operative systems.”*
>
> *“Yeah, it would be good. It would show if it is worth to start the project now or maybe use the resources to fix the data.”*
>
> *“Certainly, quite functional, maybe then what comes to mind is that it depends on the amount of data and the number of data field”*

After these, the interviewees were asked what their thoughts were if the data profiling and data quality analysis would be continuous and automatic and would provide a weekly/monthly report. All (100%) of interviewees thought that it would be good. One interviewee noted one a problem in this that it would depend on how much the business would give it value.

> *“Supports the reliability of the reports when it comes to understanding what it lacks, so it is understood what is shown in a report.”*
>
> *“Well, it could work better and that's what we basically do monthly.*

That is, check what is prohibited. Such work is done, but we do it ourselves."

"Well, that's a little bit like a control report, that at least for myself would be really useful report yes. You can take a little look at how reliable data in a data warehouse would be for specific uses."

"Yeah, basically, we have such a thing about checking and analyzing data in operative systems. From the point of view of information management, yeah, but how much business does put weight into it?"

The last question to the interviewees was if anyone has ever offered data profiling or data quality analysis services to their organization. Nine (90%) answered that no, nothing like this that they know of. One interviewee did not know.

4.4 Combined analysis

The combined analysis section contains combined results from both interviews.

4.4.1 Data quality

When we examine all the attributes identified from all interviews the most common attribute is trust. Nine (36%) of the interviewees mentioned trust when they described data quality. Second most common attribute mentioned was the correctness of data, it was mentioned eight (32%) of interviewees. Figure 47 shows all mentioned data quality attributes and their frequencies.

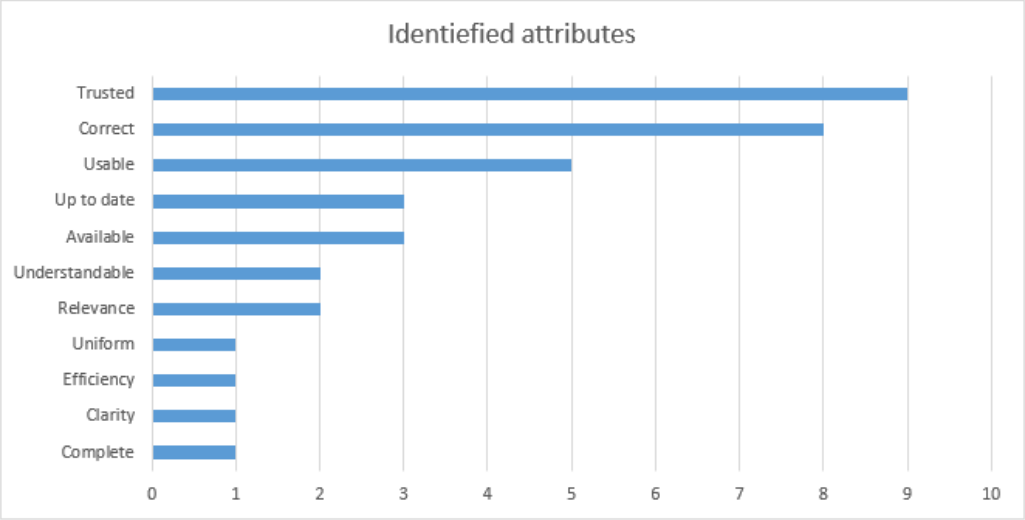

FIGURE 47. Identified data quality attributes

Most common mentioned data quality issue was the erroneous input by the user. It was mentioned 11 times (44%) by the interviewees. Next most common issue mentioned was missing or incomplete data. It was mentioned by nine (36%) of the interviewees. Figure 48 below shows all data quality issues mentioned by the interviewees.

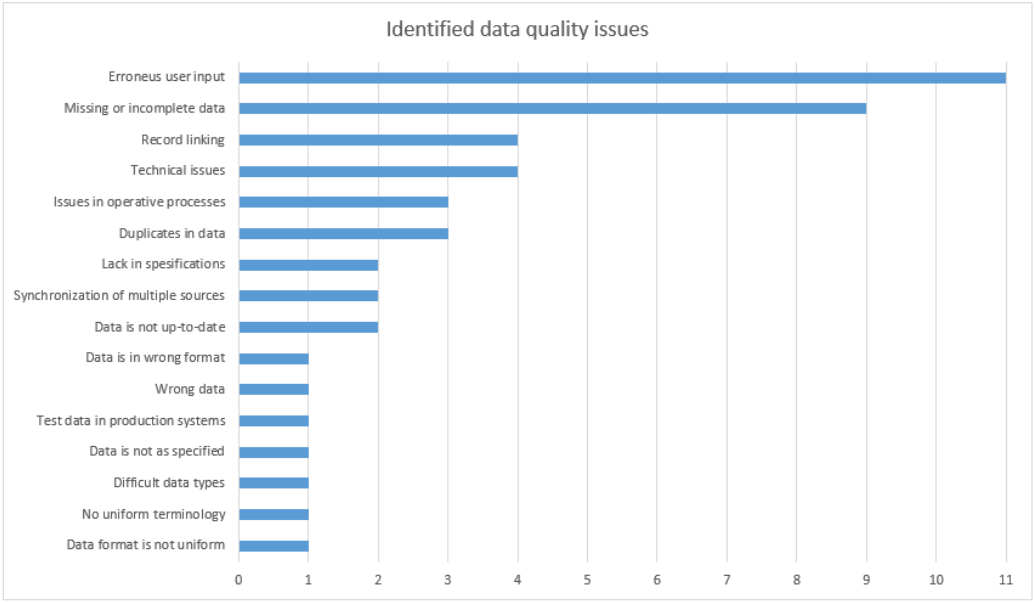

FIGURE 48. All identified data quality issues

By examining the impacts of poor data quality to the data warehouse project mentioned by all of the interviewees the most common impacts are that the data warehouse is not usable and that it increases workload. These both were mentioned eight (32%) times. Figure 49 shows all mentioned impacts that poor data quality can cause to data warehouse project.

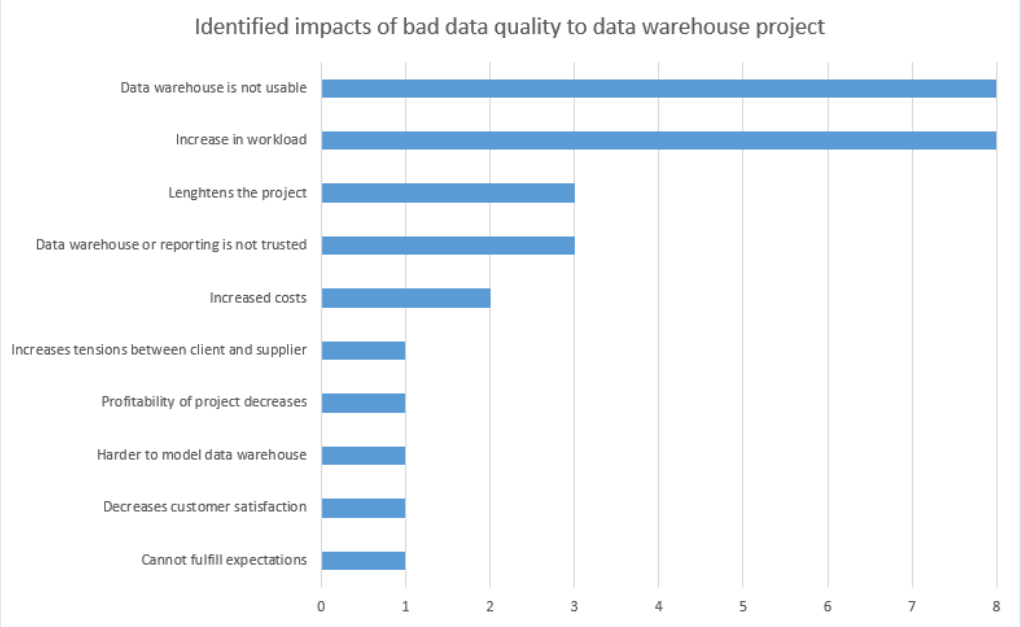

FIGURE 49. All identified impacts of poor data quality to data warehouse projects

4.4.2 Data profiling and data analysis

22 (88%) from all of the interviewees thought that data profiling would be a good thing to do at the beginning of the data warehouse project. Three (12%) of the interviewees did not see it that important but on the other hand, they did not keep it as a bad thing neither, these are counted as neutral opinions. Figure 50 below shows the overall opinion about data profiling.

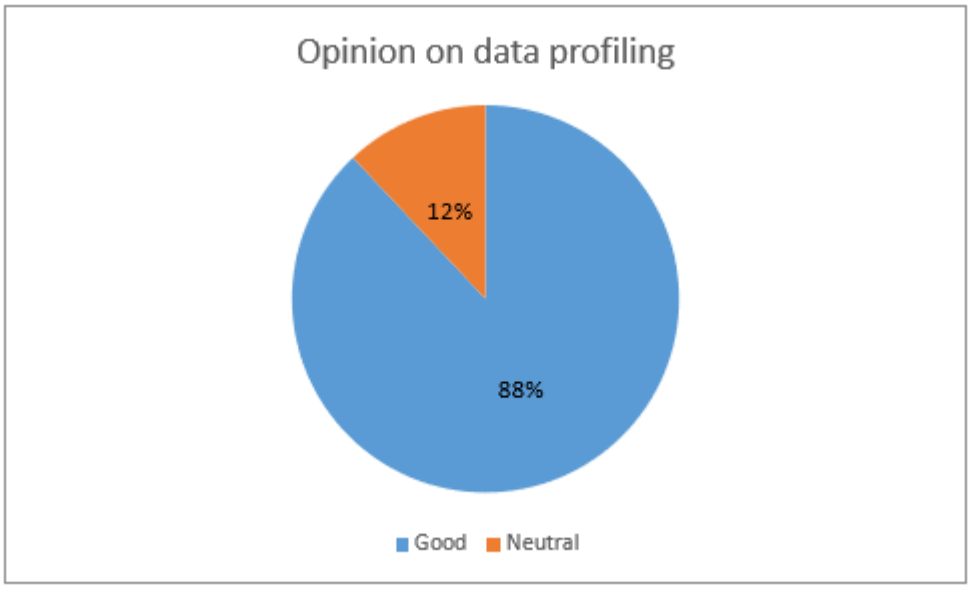

FIGURE 50. Interviewees opinions about data profiling

All interviewees (100%) kept data quality analysis at the beginning of the data warehouse project as a good thing. Ten (40%) of the interviewees, however, did bring up challenges about its implementation and the work effort that it would require.

5 CONCLUSIONS

The goal of this thesis was to identify data quality meaning in data warehousing from the clients and suppliers point of views. The aim was also to identify common data quality issues that the clients and suppliers had faced when working with data warehouses. To achieve these goals the theoretical research was used to gather necessary knowledge about data warehousing and data quality.

5.1 Research questions

Knowledge gathered in the theoretical part of this thesis (chapter 2) answers to the following research question:

How to measure and score data quality?

Simple data profiling can be used to find some of the data quality issues in the data. There are many kinds of data quality analysis for different kinds of data content. It also always requires the knowledge about the business which is used to create the appropriate data quality rules.

Interviews were conducted to get knowledge about what kind of data quality issues clients and suppliers faces in data warehouses. Most common data quality issues were erroneously inputted data and the second most common issue was missing or incomplete data (chapter 4.4.1).

Supplier side interviews showed that the record linkage is the most common data quality issue in data warehouse projects (chapter 4.2.3) and that poor-quality data increases the workload and lengthens the project (chapter 4.2.2). Supplier side interviews answers to the following research question:

What are the common data quality issues in data warehouses from supplier perspective and how these issues impact data warehouse projects?

From the client's point of view, most common data quality issues were technical and the second most common was missing data (chapter 4.3.2). Clients thought that the data quality is a condition for data warehouse and that poor-quality data increases the workload of the project (chapter 4.3.3). These interview questions answers to the research question:

> *What are the common data quality issues and impacts in data warehouses from clients' perspective?*

Interviewed clients said that they do not have been offered any data profiling or data analysis services (chapter 4.3.4). Few of the suppliers interviewed said that only rarely something similar was offered to clients. But the most common answer was that nothing like this was offered to clients (chapter 4.2.5).

Both the supplier personnel and the client personnel said that data profiling would be good and kept data quality analysis mainly a good idea (chapter 4.4.2). It can be concluded that there are markets for data profiling and data quality analysis if they could be done with reasonable workload and effort. When conducting the interviews, it was also important to tell interviewees what data profiling and data quality analysis means. It would also be important when marketing this kind of services, so the client would understand the meaning of these and the benefits they bring. This answers to the following research question:

> *Would there be any need and markets for data quality analysis and data profiling?*

The viewpoints of supplier and client about data quality does not differ much. Both highlighted mainly most issues and described data quality with same attributes. Clients highlighted more technical issues and issues related to usability than supplier personnel. Technical issues were hardware or other way software caused issues or problems. This answers to the sub-question:

How the clients and suppliers perspectives on data quality issues in data warehouses differ from each other?

5.2 Discussion

Interviews show clearly that almost all interviewees kept data profiling and data quality analysis as a good thing. One can only wonder why these are not done? Is it really too laborious or expensive?

There are a lot of books about data quality analysis and nearly all of them offer big and heavy processes to implement it. Basically, in data warehousing generally simple and swiftly implemented solution would be enough and it would catch most of the issues that cause the most problems to data warehouse projects. Data profiling would also support developers creating these data warehouse systems by giving them a cross-section of the data. The simple data quality analysis would then support the client to see the current state of their data and how it changes over time. The analysis could also increase trustiness of the data warehouse because it would show where the issues are and where there are no issues. End users could use this information, so they know what they are looking at the reports when they could acknowledge these issues. This analysis would also support client when fixing these issues and to validate the functionality of processes.

In order to get data profiling and data quality analysis to be profitable and worthwhile, it should be as cost-effective as possible. This could be achieved by automating most of the work required. Data profiling is mostly quite simple to automate. Databases can be queried to get so-called metadata about its contents. This metadata can then be used to generate required queries to execute the profiling. It should be noted that some profiling methods requires knowledge about the data contents and this information must be inputted into the system manually.

It is also possible to automate some parts of data quality analysis, but it will always require manual work in form of defining the data quality rules. After defining these rules, they can be used to automatically execute the analysis and to automatically create the reports from results. The defining of these rules should be as simple as possible, and the application should guide the user all the time. This would allow adding and modification of the rules when new source systems are added or when new uses for existing data arises.

Both the data profiling and data quality analysis should also be possible to do for each layer in data warehouse if necessary. Data is generally loaded as is into the staging area of the data warehouse and thus it is a good place to perform first data profiling. It is hard to execute data profiling or data quality analysis straight on top of the source systems because the source system can be almost anything. They range from flat files to old mainframe systems. It is also not a good practice to cause additional load to operative systems.

5.3 Follow-up development

Next step is to develop a tool for data profiling and data quality analysis. It should be simple and easily expandable. The tool would make use of the metadata available in databases and with the metadata user could select database objects to include in the profiling. Later the tool could be used to define more complex profiling techniques for selected objects. In first version, only the simplest of data quality rules would be implemented.

This kind of simple tool could then be used to test markets and the benefits. Once the knowledge has been gained, we can decide on further development.

REFERENCES

Dijcks, J-P. 2004. Integrating Data Quality into Your Data Warehouse Architecture. Business Intelligence Journal, Spring 2004, 9, 2. 18-26. [accessed 23 March 2018]. Available in: https://search-proquest-com.aineistot.lamk.fi/business/docview/222639489

Eskola, J. & Suoranta, J. 2014. Johdatus laadulliseen tutkimukseen

Ferrari, A. & Russo, M. 2016. Introducing Microsoft Power BI

Fowler, M. 2010. UML Distilled Third Edition, A brief guide to the standard object modeling language

Helfert, M. & Clemens, H. 2002. Proactive data quality management for data warehouse systems. DMDW. Vol. 2002. [accessed 25 March 2018]. Available in: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-58/herrmann.pdf

Herzog, T. N., Scheuren, F. J. & Winkler, W. E. 2007. Data Quality and Record Linkage Techniques

Hirsjärvi, S. & Hurme, H. 2016. Tutkimushaastattelu, Teemahaastattelun teoria ja käytäntö

Hovi, A., Hervonen, H. & Koistinen, H. 2009. Tietovarastot ja business intelligence

Kim, W. 2002. On Three Major Holes in Data Warehousing Today. Journal of Object Technology, 1.4, 39-47. [accessed 25 March 2018]. Available in: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.7826&rep=rep1&type=pdf

Kimball, R. & Ross, M. 2002. The Data Warehouse Toolkit, second edition

Kumar, V. & Reema, T. 2013. A Simplified Approach for Quality Management in Data Warehouse. [accessed 23 March 2018]. Available in: https://arxiv.org/ftp/arxiv/papers/1310/1310.2066.pdf

Kozielski, S. & Wrembel, R. 2009. New Trends in Data Warehousing and Data Analysis

Larson, B. 2009. Delivering Business Intelligence with Microsoft SQL Server 2008

Lee, Y. W., Pipino, L. L., Funk, J. D. & Richard, Y. W. 2006. Journey to Data Quality

Linstedt, D. & Olschimke, M. 2016. Building a Scalable Data Warehouse with Data Vault 2.0

Maydanchik, A. 2007. Data Quality Assessment

Olson, J. E. O. 2003. Data Quality the Accuracy Dimension

Pipino, L. L., Yang, W. L. & Richard, Y. W. 2002. Data Quality Assessment. Communications of the ACM, 45.4. 211-218. [accessed 25 March 2018]. Available in: http://0374288.netsolhost.com/pdf/MIT-pipleewang.pdf

Redman, C.T. 2001. Data Quality the Field Guide

Smith, M. S. 2013. Determining Sample Size. [accessed 23 February 2018]. Available in: https://www.researchgate.net/profile/Ambarish_Rai/post/How_do_you_select_sample_size_in_relation_to_population_size/attachment/59d61f1779197b807797d810/AS:282588074790914@1444385655855/download/Determining-Sample-Size.pdf

Singh, R. & Singh, K. 2010. A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. International Journal of Computer Science Issues, Vol 7, Issue 3, No 2. 41-50. [accessed 20 January 2018]. Available in: https://pdfs.semanticscholar.org/a0b3/03955e1d05f8338b426111392ee749ac1a60.pdf

Won, K., Byoung-ju, C., Eui-kyeong, H., Soo-kyung, K. & Doheon, L. 2003. A Taxonomy of Dirty Data. Data Mining and Knowledge Discovery, 7, 81-99. [accessed 18 October 2017]. Available in: http://dx.doi.org/10.1023/A:1021564703268

APPENDIX 1.

Complete taxonomy of dirty data. (Won et al. 2003, 84-85.)

1. Missing data
    1.1 Missing data where there is no Null-not-allowed constraint
    1.2 Missing data where Null-not-allowed constraint should be enforced
2. Not-missing data
    2.1 Wrong data, due to
        2.1.1 Non-enforcement of automatically enforceable integrity constraints
            2.1.1.1 Integrity constraints supported in relational database systems today
                2.1.1.1.1 User-specifiable constraints
                    2.1.1.1.1.1 Use of wrong data type (violating data type constraint, including value range)
                    2.1.1.1.1.2 Dangling data (violating referential integrity)
                    2.1.1.1.1.3 Duplicated data (violating non-null uniqueness constraint)
                    2.1.1.1.1.4 Mutually inconsistent data (action not triggered upon a condition taking a place)
                2.1.1.1.2 Integrity guaranteed through transaction management
                    2.1.1.1.2.1 Lost update (due to lack of concurrency control)
                    2.1.1.1.2.2 Dirt read (due to lack of concurrency control)
                    2.1.1.1.2.3 Unrepeatable data (due to lack of concurrency control)
                    2.1.1.1.2.4 Lost transaction (due to lack of proper crash recovery)
            2.1.1.2 Integrity constraints not supported in relational database systems today
                2.1.1.2.1 Wrong categorical data (e.g. out of category range data)
                2.1.1.2.2 Outdated temporal data (e.g. person's age or salary not having been updated)
                2.1.1.2.3 Inconsistent spatial data (e.g. incomplete shape)
        2.1.2 Non-enforceability of integrity constraints
            2.1.2.1 Data entry error involving a single table/file
                2.1.2.1.1 Data entry error involving a single field
                    2.1.2.1.1.1 Erroneous entry (e.g. age mistyped as 26 instead of 25)
                    2.1.2.1.1.2 Misspelling
                    2.1.2.1.1.3 Extraneous data (e.g. name and title, instead of just the name)
                2.1.2.1.2 Data entry error involving multiple fields
                    2.1.2.1.2.1 Entry into wrong fields (e.g. address in the name field)
                    2.1.2.1.2.2 Wrong derived-field data (due to error in functions for computing data in a derived field)
            2.1.2.2 Inconsistency across multiple tables/files (e.g. the number of employee in the Employee table and the number of employee in the Department table do not match)
    2.2 Not wrong, but unusable data
        2.2.1 Different data for the same entity across multiple databases (e.g. different salary data for the same person in two different tables or two different databases)
        2.2.2 Ambiguous data, due to
            2.2.2.1 Use of abbreviation (Dr. For doctor or drive)
            2.2.2.2 Incomplete context (homonyms; and Miami, of Ohio or Florida)
        2.2.3 Non-standard conforming data, due to
            2.2.3.1 Different representations of non-compound data
                2.2.3.1.1 Algorithmic transformation is not possible
                    2.2.3.1.1.1 Abbreviation (ste for suite, hwy for highway)

2.2.3.1.1.2 Alias/nick name (e.g. Bill Clinton, President Clinton)

2.2.3.1.2 Algorithmic transformation is possible

2.2.3.1.2.1 Encoding formats (ASCII, EBCDIC,…)

2.2.3.1.2.2 Representations (including negative number, currency, date, time, precision, fraction)

2.2.3.1.2.3 Measurement units (including date, time, currency, distance, weight, area, volume,...)

2.2.3.2 Different representations of compound data

2.2.3.2.1 Concatenated data

2.2.3.2.1.1 Abbreviated version (John Kennedy for John Fitzgerald Kennedy)

2.2.3.2.1.2 Uses of special characters (Space, no space, dash, parenthesis)

2.2.3.2.1.3 Different orderings (John Kennedy for Kennedy, John)

2.2.3.2.2 Hierarchical data

2.2.3.2.2.1 Abbreviated version

2.2.3.2.2.2 Uses of special characters

2.2.3.2.2.3 Different orderings (City-state, state-city)

APPENDIX 2.

Supplier interview template.

#	Question
1	What is your current role in the organization?
2	How many years have you worked in your current job?
3	In how many data warehousing projects or in similar projects have you been part of?
4	What does data quality mean to you?
5	What effects if any does data quality cause for the data warehousing project?
6	Have you encountered problems with data quality when developing data warehouses? If yes, then what kind of issues?
7	Which of these issues has the biggest impact in your opinion?
8	What do you think about basic data profiling before the start or at the beginning of data warehouse project?
9	How about basic data quality analysis before the start or at the beginning of data warehouse project?
10	Have there been any talk about data quality with customers before project start or has your company offered any data profiling/data quality analysis to the clients?
11	Has the customer been interested in the data quality or has the client had an opinion on the data quality of their systems at the beginning of the project?

APPENDIX 3.

Client interview template.

#	Question
1	What is your current role in the organization?
2	How many years have you worked at your current job?
3	What does data quality mean to you?
4	Do you do any kind of data quality analysis?
5	Are there any issues with data quality in your organization's data warehouses or other systems?
6	Do you face any data quality issues in your work? if yes, then in what kind of issues?
7	Which of these issues has the biggest impact on your work or the work of others?
8	What in your opinion causes these data quality issues?
9	How do you think that data quality affects data warehouse projects or similar projects?
10	What do you think about basic data profiling before the start or at the beginning of data warehouse project?
11	How about basic data quality analysis before the start or at the beginning of data warehouse project?
12	What do you think about automatic and continuous data quality analysis and data profiling utilizing the data in the data warehouse?
13	Has your company been offered this kind of services?