OULUN YLIOPISTO
UNIVERSITY of OULU

Effects of Automatically Updated Database Documentation on the Work Tasks of IS Professionals and End-users

University of Oulu
Faculty of Information Technology and Electrical Engineering / Degree program of Information Processing Science
Master's Thesis
Aliisa Värttö
2.6.2020

Abstract

This study analysed the effects of automatically generated database documentation on the work tasks of IS professionals and end-users. The effects were analysed with case study approach in one Finnish software company that provides customer information system to their customer companies. The availability of the consistent database documentation was a new situation for both IS professionals in the case company and for the end-users in the customer companies. The case company wished to understand, how the documentation would affect the work tasks of the two stakeholder groups.

Database documentation is used for database design and analysis tasks. Database documentation is created in the development phase of an IS and maintained afterwards. Yet, the documentation can end up stagnated and the documentation maintenance is commonly considered a burden for the IS professionals. Database reverse-engineering allows automatic generation of database documentation, thus erasing the manual maintenance of database documentation.

The research data was gathered using two separate questionnaires for the end-users and IS professionals. Both qualitative and quantitative data were gathered. The questionnaire's open questions were analysed using content analysis and closed questions were considered descriptive statistics.

According to the findings, database documentation supports both IS professionals' and end-users' in their work tasks. Database documentation was considered important for the efficiency of their work tasks. Documentation aids the database users to gain knowledge of the database structure and prevents false interpretations. Database documentation also allows the users to plan work better and to conduct tasks with fewer increments. Additionally, documentation allows a better degree of independent work and gives courage for the employees to familiarize themselves to new parts of the database and information system.

The consistent, up-to-date database documentation has positive effects on the work tasks of both IS professionals and end-users. Lower workload, better understanding of the system and less false interpretations indicate that software companies would benefit from documenting their databases in more detail and more consistently, for example, using database reverse-engineering. Consistent database documentation benefits all of the database users.

Keywords
Database, Relational database, Database documentation, Database reverse-engineering, ER diagram, Metadata

Supervisor
PhD, Adjunct Professor and University Lecturer Raija Halonen

Foreword

It has been the most exciting and peculiar spring. Although COVID-19 changed our daily lives in 2020 by forcing us to work from home and develop new methods of collaboration, I managed to get my master's thesis ready right on the verge of summer. I have multiple people that I wish to thank for enabling this achievement for me.

I would like to thank the case organization and its representatives for allowing this master's thesis to be written. The guidance I received from the wise and supportive case company developers was crucial for the success of this study and helped me to enhance the thesis time and time again. I want to express my gratitude to the people who gave valuable answers for the questionnaires and thus made this thesis possible.

I would like to thank PhD Raija Halonen for supervising my master's thesis. They gave me plenty of guidance regarding scientific communication and practices. Additionally, I am most grateful for PhD Jukka Mononen for their commitment to the research process. They were like a bedrock of this study, initially endorsing the theme of the research to me and giving most valuable guidance along the way. Thank you, MSc Anneli Karjula for all the feedback you gave.

Speculative, interesting discussions with Sanna Kemppainen regarding the thesis topic and case were a power that motivated me to strike while the iron was still hot. Thank you, for the wonderful, inspiring conversations. Thank you, Jukka Uhlgren and Ilpo Kekäläinen, for providing very valuable feedback on my thesis. I would also like to thank all of the peer reviewers of my thesis, you have been most helpful.

Lastly, I would like to express my utmost gratitude to my whole family for supporting me in my studies through these years and for giving me valuable advice of life. My father once said that writing a thesis is "nothing more, than to sit and to write", and that is what I did. I have learned that hard work, positivity and resilience pay off in the end and I thank my mother for teaching me the skills that I have needed to meet my goals.

Aliisa Värttö

Oulu, June 2, 2020

Abbreviations

ACID	Atomicity (A), consistency (C), isolation (I) and durability (D) of data
CAP	Consistency (C) and availability (A) in case of distributed system partition (P)
CIS	Customer Information System
BASE	Basically available (BA), soft state (S), eventually consistent (E)
DB	Database
DBMS	Database Management System
DBRE	Database Reverse Engineering
ER	Entity-Relationship (model)
IS	Information System
JDBC	Java Database Connectivity
ORM	Object-Role Modeling
SW	Software
UML	Unified Modeling Language

Contents

1. Introduction

Visual database descriptions are commonly seen as tools for the design phase of an information system (IS). Database (DB) and its structure is ideally documented in graphical diagrams in the design phase of an IS and the documentation is updated throughout the software lifecycle (Gregersen & Jensen, 1999). In addition to the benefits the developers gain from the documentation, DB representations given to the users influence the successfulness of their database use and affect the system learnability and usage in the future (Leitheiser & March, 1996). Yet DB documentation might face documentation-stagnation in the next phases of the software lifecycle, which means that it might be missing, out-of-date, incomplete or inaccurate (Steinberger & Prakash, 2011). This phenomenon cumulates from the developer's perception of DB documentation being more arduous than source code documentation (Steinberger & Prakash, 2011). Database reverse engineering (DBRE) makes it possible to automatically generate DB documentation and avoid the stagnation (Mfourga, 1997). This study analysed the effects of automated database documentation in a case company and on its customers. The purpose was to describe how the availability and consistency of visual database structure descriptions and metadata overviews influence the daily work of both stakeholder groups.

It is rather common for DB documentation to face stagnation (Steinberger & Prakash, 2011) and such was the case in this study as well. The case company is a Finnish software company which functions in the business-to-business market and provides Customer information system (CIS) to their customers. The manually updated database documentation the company had was inevitably outdated with some columns, relationships and metadata missing and it only described production DB structure, not the reporting DB. There was no clear process for DB documentation maintenance in the case company and even the employees slowly adapted to the era of inaccessible or inaccurate DB documentation. Developers quickly familiarised themselves with the DB structures through queries to the database and with the aid of co-workers.

Yet, the customers had started requesting updated documentation of their database structure and metadata. With the earlier solution the database documentation that could be passed to the customers was rather basic and partly inadequate. One option was to automate the process, requiring more work in the development process but decreasing the cumulative use of resources in the future. In the late 2019, the company decided to automate their DB documentation process.

As a result from the automation development, the company had up-to-date, comprehensible and consistent DB documentation that could be created anytime with a few simple steps. The automated solution was a script that interpreted the DB structure and created the documentation with the help of open source tools. The documentation consisted of modified ER diagrams and overview of DB metadata. The initial documentation received a positive response from the customers. After the automation was created and first documents sent to the customers, the case company's developers started wondering how this rapid change in documentation availability would affect them and their customers, including end users of the CIS.

Prior research has studied the relational model of databases proposed by Codd (1970) and created multiple different notations to document database structure. Common notations are, for example, the entity-relationship (ER) model by Chen (1976) and the modified class diagram (Halpin & Morgan, 2010). Additionally, database metadata can be documented in addition to the visual models (Fowler, 2004). Although database structure visualisations are used in design phase of IS development and mainly by developers (Gregersen & Jensen, 1999), the users can benefit from the documentation by gaining more understanding of the system (Leitheiser & March, 1996). Prior research has studied the automatic generation of DB visualisations through DBRE (Blaha, 1998; Chiang & Barron, 1995; Ndefo, 2017).

Developers of the case company wished to understand their customer's work in more detail. They did not know the effects the up-to-date, consistent database documentation has on their customer's work tasks and whether it would make some of their work easier. Similarly they were interested in understanding the effects on their own work and to other IS professionals' work in the case company. The research problem was formatted into the following research question:

RQ: How does (automated) DB documentation affect the DB users' work tasks?

Two separate stakeholder groups were recognised to possibly face changes due to the automated database documentation. First recognised stakeholder group was the customers of the case company that act as end-users of the database. The second recognised stakeholder group was the personnel of the case company that work in multiple IS related positions. To study the effects of automated database documentation separately on the two stakeholder groups, the research questions 1 and 2 were formatted:

RQ1: How does (automated) DB documentation affect the end-users' work tasks?

RQ2: How does (automated) DB documentation affect the IS professionals' work tasks?

To answer for the main research question, findings of the sub questions 1 and 2 were analysed and combined to provide broader insight into the effects of automated database documentation on the stakeholder groups' work tasks. Automation itself was not analysed in this study. The effects the automation had on the stakeholder groups was analysed in another master's thesis (Kemppainen, 2020).

Case study was chosen as the research strategy for this study. Questionnaire was used as a qualitative research method for both sub questions. Sub question 1 was studied using questionnaire for the case company's customers and end-users that worked with the reporting database. Customer questionnaire was sent to the participants as part of a bigger questionnaire that contained questions from two other Master's thesis studies (Kemppainen, 2020; Uhlgren, 2020). Participants in the customer questionnaire were asked how they perceived the change that was caused by the automatically generated DB documentation. Sub question 2 was studied with qualitative questionnaires that were sent to the case organization's IS professionals from multiple teams. The IS professionals were asked how they had benefitted or could benefit from automatically generated DB documentation in their work tasks.

Prior literature is discussed in Chapter 2, divided into three categories. The first theme is the effects of automation, the second is the basics of relational databases, and finally the relational database documentation and DBRE are presented. Research methods are introduced in Chapter 3. Implementation of the questionnaires and the analysis of data is described in Chapter 4. Chapter 5 describes the findings that can be derived from the analysis of the questionnaires. Chapter 6 contains discussion of the findings and comparing these to the prior literature. Chapter 7 concludes the study.

2. Prior Literature

This study analysed the automatically created and updated database documentation and its effects on the work tasks of developers and end users. Prior research has studied automation processes and outcomes, relational databases and the database documentation. The first theme in prior research is the overview of automation and its effects in Section 2.1. The second theme is the history of relational databases and their nature in Section 2.2. The last theme in Section 2.3 is the relational database documentation including modelling and reverse engineering a database into a visual format.

2.1 Effects of automation

IT industry is one frontier of manual task automation. Some companies in the software (SW) industry automate when their manual tasks are considered too time-consuming and interfering. Carrying out other processes might suffer when employee's attention is pinned to the manual tasks at hand. Automating the manual tasks is a good option if resources are wished to be freed from the consecutive tasks to the more important ones, to lower costs and gain higher profitability. (Krishnan & Ravindran, 2017.)

If the goal is to make human's work simpler the core question should not be "what can be automated"; more human-centered question would be "what kind of automation supports the human operators and the human-machine cooperation the best" (Sarter, Woods, & Billings, 1997). Computer algorithms can be written only if it is possible to define strict and well-defined rules for it (Henning & Kutscha, 1994). Automated systems cannot be 'human', performing reckless and intuitive decision making, because each step of the process that is the subject of automation has to be definable (Tschiersch & Brandt, 1996).

Janssen, Donker, Brumby and Kun (2019) divided human-automation interaction and system automation into three overlapping categories. Automated systems are either expert systems or for non-professional users and they can be time-sensitive and safety critical. Certain systems fall into multiple categories. As an example Janssen et al. (2019) pointed out that automated cars are used by non-professional users, they contain automated expert systems and require high standards for safety procedures.

Janssen et al. (2019) listed three lasting themes in automation research. First is the persistent discussion of how automation of work tasks changes the workload of humans. Second is the debate on superiority of either humans or computers in certain tasks. Third lasting theme of automation research is the so-called 'Irony of automation', as described by Bainbridge (1983). The irony lies in the attempt to erase the need of human manual control of processes through automation, resulting in more complex systems requiring more experienced supervisors and maintainers. According to Bainbridge (1983), this paradox leads to a conclusion: both technical and human factors are as important for automated systems.

This understanding of the nature of automation is visible in the human-centered automation approach which supports the requirements and operations of the users. Its counterpart is technology-centered approach, which aims for technological applicability and effectiveness in the automated process. The main difference is that human-centered automation does not aim to erase the human from the process but to make the automation support humans and vice versa. (Sarter et al., 1997.)

According to Sarter et al. (1997), user-centered approach to automation considers firsthand the impacts the automation has on the roles of the people using the system and additionally how the automated piece functions with the other pieces of the surrounding system. They also introduced a list of eight characteristic, assumed benefits of automation and their possible side-effects when pursued as listed in Table 1.

Table 1. Designers' view of assumed benefits of automation with the contradicting experiences and side-effects (Sarter et al., 1997).

Assumed benefits	Experiences of effects
Better results with same system (substitution)	Practices transform, roles change
Less resources 1: Offloads work	New disruptive cognitive work tasks
Less resources 2: Focus user attention on the right answer	New threads to follow, tracking changes and staying aware harder
Less knowledge required	New knowledge and skills required
Autonomy of the system	Criticality of good human-automation interaction rises
Same feedback as before	New roles require more precise feedback
Generic flexibility	Explosion of new features with options, modes and errors - more failures
Reduce human error	Issues of human-machine interaction produce new types of errors and breakdowns

Table 1 lists the contradiction of assumed change the automation provides and the changes the newly automated process could create. Automation tends to accrue unexpected changes. For example, automation is expected to free work resources from the previously manual tasks and ease decision making but as an effect the new process requires more refined knowledge from the user. Observation and maintenance of the automated solution could create new work tasks that are cognitively exhausting and require high awareness and knowledge of the system surrounding the process. (Sarter et al., 1997.)

2.2 Relational databases

Codd (1970) introduced the relational model of data which is the basis of all relational databases. With Codd's (1970) relational model of data, model's users have it easier to represent and navigate their data with less details. Relational model provides means to handle derivability, redundancy and consistency in a database. In relational model, the database consists of multiple relations (tables) containing varying amount of tuples (rows) which consist of attributes (columns) and specifically, their values. The data included in the attributes of a tuple form a description of a predefined set of attributes linked to one real life entity, as showcased in Figure 1 that follows the Codd's (1970) relational model notation. In the example given in Figure 1, tables 'Info', 'Name' and 'Address' all have their own primary keys that make it possible to find relations between the tables. Using the data in the 'Info' table, it is possible to find both customer name and address using the foreign keys 'Name ID' and 'Address ID' for all of the customers.

Info

Customer ID	Customer Num	Name ID	Address ID
1	'1'	2	3
4	'2'	5	6
7	'3'	8	9

Name

Name ID	First Name	Last Name
2	'Petunia'	'Dursley'
5	'Sherlock'	'Holmes'
8	'Bruce'	'Wayne'

Address

Address ID	Street	Street Num	House	Apartment	City	Country
3	'Privet Drive'	4	NULL	NULL	'Surrey'	'United Kingdom'
6	'Baker Street'	22	'B'	1	'London'	'United Kingdom'
9	'Mountain Drive'	1007	NULL	NULL	'Gotham'	'United States'

Figure 1. Example of a relational model of database structure containing customer data.

Relational databases should be normalized using non-redundant primary keys that make each tuple of each relation unique in the whole database. Relationships between tuples of the same entity are created in other database tables using foreign keys, for example, linking data from table A to the data of table B using the primary key of table B as a foreign key in the table A, as seen in Figure 1. Cross referencing with unique keys is a user-oriented approach to database management and it provides the possibility to describe data with its own natural structure. (Codd, 1970.)

The data of a relational database can be better understood using 'semantic data modeling', independent of the database implementation. Relational model captures the basics of relational database. Semantic data of a database contains descriptions of database features including domains, keys and dependence between relations. Figure 2 presents an example of unconditional generalization. 'Organization' and 'Person' are subclasses of 'Customer' and the attribute 'Customer#' is used to recognise separate entities. (Codd, 1979.)

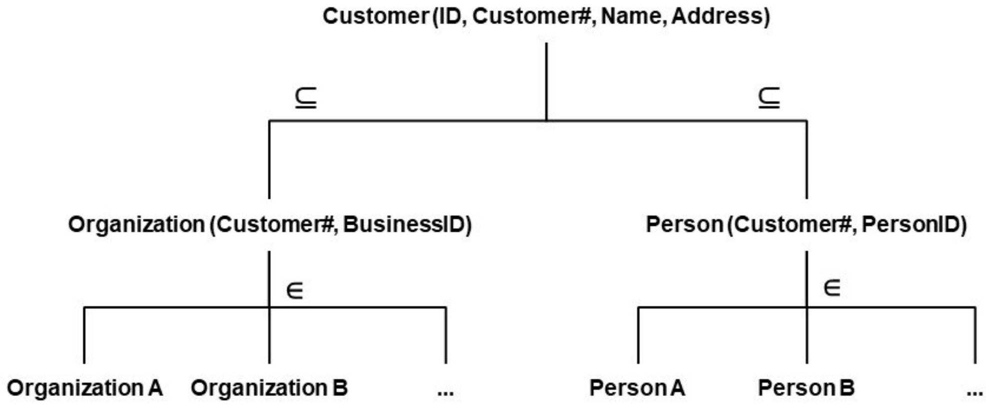

Figure 2. Unconditional generalization.

Many database management systems used are relational (Gregersen & Jensen, 1999) but additionally the so called NoSQL databases have penetrated the marked by providing more flexibility and better performance than relational databases. NoSQL DB cannot be queried using SQL language because they are not relational. NoSQL DB are commonly divided into two main categories which are storages for key-values or documents (Stonebraker, 2010.) Although NoSQL has its benefits in scalability and efficiency (Leavitt, 2010), organizations with high requirements for accuracy of data might require the consistency and reliability of relational databases.

As described in CAP theorem companies have to make decision whether to pick consistency (C) or availability (A) in case of distributed system partition (P). In case of node malfunction, the system handles write-transactions differently depending on the system design. System that emphasizes the availability carries out the transaction to functioning nodes, creating temporary and sometimes permanent inconsistency in the database nodes. System that demands consistency of data rollbacks the transaction fully, not allowing changes to happen while the partition is active, making the system thus unavailable. Two opposite approaches are the consistent ACID properties and the available BASE approach. (Brewer, 2012.)

ACID principle attempts to ensure the atomicity (A), consistency (C), isolation (I) and durability (D) of data in case of database recovery (Haerder & Reuter, 1983). NoSQL databases do not necessarily pass the ACID restraints of data. Obeying the ACID principles is often necessary for the high quality of organizational data, possibly describing sensitive information of individuals, for example, medical records or payment behavior. (Leavitt, 2010.) According to Brewer (2012), systems that follow the ACID principle choose consistency over availability in case of system partition as described in CAP theorem.

In systems that follow the BASE principle the data can be “basically available, soft state, eventually consistent” (Pritchett, 2008). BASE allows write-transactions also in case of system partition, therefore creating temporal inconsistency of data in the different nodes but enabling high read and write performance. Consistency can be later acquired when the transaction is carried out to the other nodes after the system partition has ended. (Vogels, 2009.) Systems that follow BASE ideology choose availability over consistency when considering the CAP theorem and are more often NoSQL databases (Brewer, 2012).

2.3 Relational database documentation

Chen (1976) introduced an entity-relationship (ER) model to highlight how entities in databases mirror the natural world and real-life entities and relationships and lists four levels on which the data can be described:

1. Information about entities and their relationships,
2. Information structure,
3. Access-path-independent data structure,
4. Access-path-dependent data structure.

Gregersen and Jensen (1999) pointed out that ER model can be used for different tasks of database design and analysis. Especially the design tasks are commonly supported by ER diagrams, but afterwards analysis tasks can benefit from the diagrams as well. The original ER model includes the means to describe entities and entity sets, attributes, values and value sets and lastly the conceptual information structure. The primary keys of Codd (1970) can be exploited to visualise the database structure and contents into a form visible in Figure 3 that follows the examples of Chen (1976). In Figure 3, entity 'Car' can have a relation of type 'Car-Buyer' to a maximum of one entity 'Buyer', whereas 'Buyer' can have a relation to multiple 'Car' entities. ER diagrams have not been standardized and this has led to numerous different new and extended versions, using varying notations and considering more characteristics of the data (Schmieder, Plimmer, & Dobbie, 2009; Halpin & Morgan, 2010).

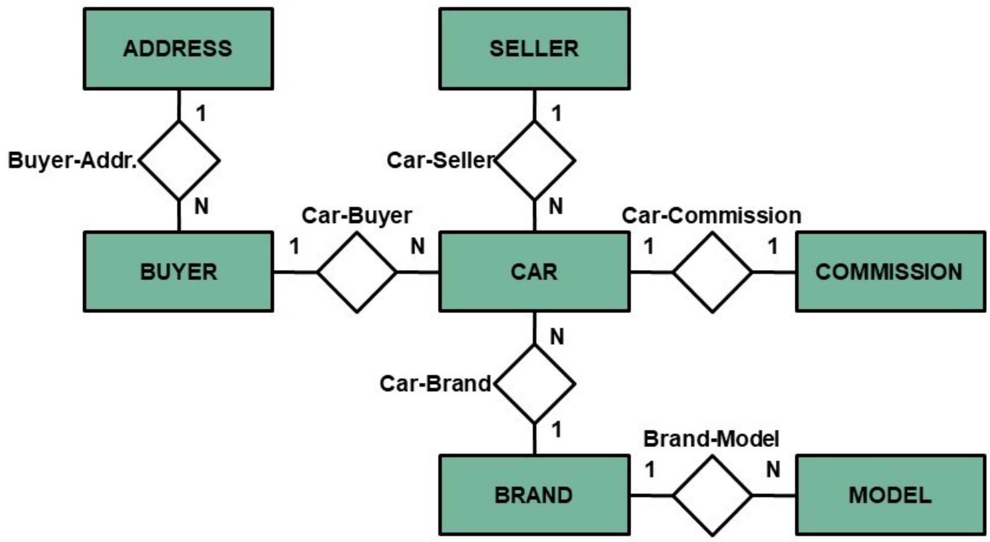

Figure 3. Example of an Entity-relationship diagram.

Figure 3 visualises the database structure and cardinalities between database tables. ER models also highlight the different types of relationships between database tables, as visible in Figure 3. An ER diagram (Figure 3) can be used when designing a database with four steps (Chen, 1976):

1. Identify the entity and relationships sets,
2. Identify semantic information in the relationship sets,
3. Define the value sets and attributes,
4. Organize data into entity/relationship relations and decide primary keys.

ER model is useful when specifying a database with its constraints to ensure data integrity. Constraints define what types of values are acceptable for each attribute (column) in a database. These can be, for example, if a value is nullable or not, if a value must be of type string and shorter than 30 characters or if it has to be a number between 0 and 10. After the database has been designed and possibly implemented, the ER model also aids when trying to understand the meaning of attributes or structure of the database. (Chen, 1976.)

ER modeling is often used for designing and analysing databases. ER diagrams are commonly used in the DB development process in its first phases of DB design because it is considered easy-to-use (Schmieder, Plimmer, & Dobbie, 2009). It has become very popular in the research community and industry due to its simplicity, learnability, understandability and capability to describe the database structure in comprehensible form (Gregersen & Jensen, 1999). Gaining understanding of the meaning of data is an endless process (Codd, 1979). It might be unnecessary and impossible to document all of the available knowledge of real-life entities and relationships (Chen, 1976).

Database structure can be visualised with different modelling notations to make it more tangible, provide additional knowledge of the database and to summarize metadata (Fowler, 2004). Metadata is ‘data that provides information about other data’ (Merriam-Webster, n.d.), and more simply, ‘data about the data’ (Graefe, 1993). In relational databases metadata can be saved in certain database tables. For example, metadata can describe namespaces and access-control information (Ghermawat, Gobioff & Leung, 2003), document types and publication dates (Sebastiani, 2002) or categories (Yee, Swearingen, & Hearst, 2003). Metadata can be used to verify and ensure data validity (Graefe, 1993). In Oracle databases, metadata is saved in read-only data dictionaries in the DB’s SYSTEM table space and they include descriptions of schema objects (e.g., tables, views and indexes), allocated space for objects, default values, integrity constraints and many more (The Data Dictionary, n.d.). In PostgreSQL databases, schema metadata is saved in system catalogs as regular tables, thus allowing the use of write-operations as well. PostgreSQL catalogs contain, for example, constraints, descriptions and indexes (System Catalogs, n.d.), similar to the Oracle dictionaries.

ER model by Chen (1976) is not the only option for DB modelling: For example, different versions of ER diagram, modified unified modeling language (UML) class diagram and object-role model (ORM) are used to document the database structure in IS design tasks (Halpin & Morgan, 2010). ER model has been enhanced many times during its years and multiple broadened or more precise versions have been published to handle aspects not included in the original ER model by Chen (1976). Such aspects are, for example, temporality and time dimension that are not adequately supported in the standard ER model or even in the extended ER model (Gregersen & Jensen, 1999). Although standardized, variants of UML diagram notations exist and different designers and scientists have their own preferences (Purchase, Colpoys, McGill, Carrington, & Britton, 2001). For example, differences and similarities of ER, ORM and UML class diagrams are presented in Figure 4 following the examples of Halpin and Morgan (2010). Examples describe a database structure which handles car ownership.

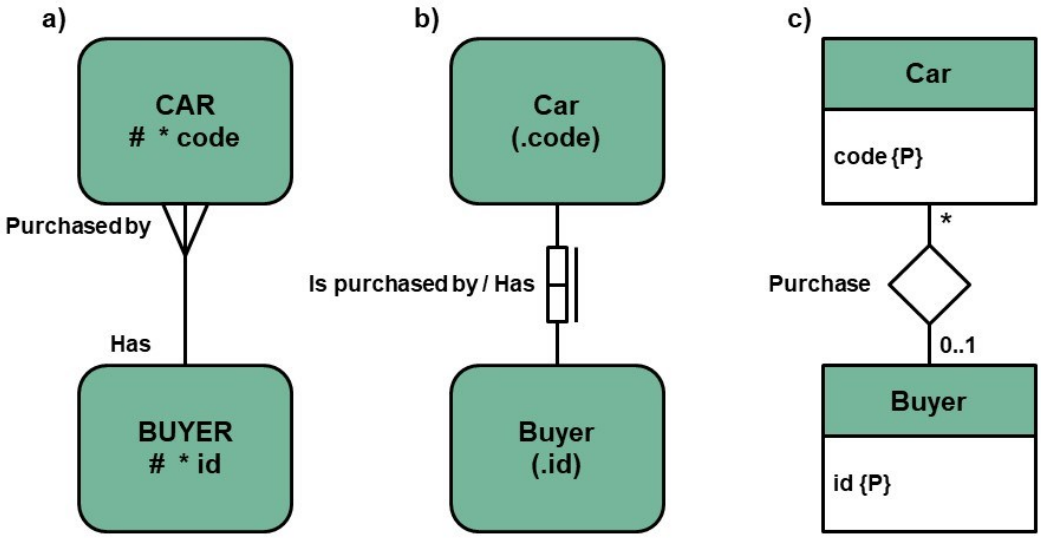

Figure 4. The Same database structure described in three different modelling notations: a) ER diagram, b) ORM diagram and c) UML Class diagram.

Figure 4 demonstrates how three different modelling notations describe the same simple database structure. The notations used are ER, ORM and UML class modelling. All of the three diagrams contain two entities (Car, Buyer) with attributes 'code' and 'id'. Diagrams also describe the relationships and cardinalities between the entities. Notations have differences, for example, UML class diagram describes entities as two-story boxes, whereas included variants of ER and ORM models describe them as rounded rectangles. In contradiction, Figure 3 of Chen's (1976) ER model describes entities as boxes. Other difference is the cardinality and relationship: example of an ER diagram in Figure 4 uses so-called crow's foot notation to describe that 'entity Buyer can have multiple Cars', whereas ORM does not describe cardinality in the diagram and UML class diagram uses notations '0..1' and '*'. Although the database structure behind the diagrams may be the same, all diagrams look different and their interpretation might vary due to their characteristics. As Purchase et al. (2001) stated, the decision on which notation to use is dependent on the decision maker and their preferences. Additionally, Halpin and Morgan (2010) pointed out that the working environment and existing culture might influence the selection of modelling notation.

Keeping database documentation up-to-date prevents 'documentation-stagnation'. Stagnated documentation might mean that the database documentation is either missing, incomplete, inaccurate or it does not mirror the current structure of the database. For example, new IS versions can include added database tables, columns or relationships and these changes should be added to the existing documentation to prevent stagnation. One undocumented version can multiply the required time of document updates in the future: locating and tracing the changes and understanding why they were implemented takes more time after time has passed. Furthermore, outdated documentation can be considered more harmful than no documentation at all, since false assumptions of the database and code can lead to wasted resources. (Steinberger & Prakash, 2011.)

First versions of database documentation can be hand-drawn sketches, but it is recommended to formalise them soon after creation (Schmieder, Plimmer, & Dobbie, 2009). Formalising sketched ER diagrams automatically might prove to be hard, since ER diagrams are not standardised (Schmieder, Plimmer, & Dobbie, 2009; Halpin &

Morgan, 2010) and the systems might not recognise or handle the different notations correctly (Schmieder, Plimmer, & Dobbie, 2009). DB documentation is often perceived more difficult and laborious than source code documentation: Steinberger and Prakash (2011) listed the lack of good documentation tools as one reason for the outdated database documentation. They also stated that human attention and observation is required for the successful use of most DB documentation tools.

Comprehensive database documentation does not only profit developers and other IS professionals, but it also makes the system use easier: End-users that have read-only rights to a database spend a lot of time searching for the right schemas and tables to write their own queries (Yasir, Swamy, & Reddy, 2012). End-users of the IS can also gain advantages of database documentation. System learnability and successfulness is significantly increased when the end-user understands the database structure and has basic knowledge of query languages. (Leitheiser & March, 1996.)

Different approaches might be taken to avoid manual creation of DB documentation, for example, commercially available documentation tools (Gregersen & Jensen, 1999) or open source tools (Steinberger & Prakash, 2011). Especially legacy systems are commonly poorly documented, and this poses as an issue for many companies: Instead of coping with inadequate documentation and generating more undocumented changes, it is possible to update the documentation through reverse engineering (Mfourga, 1997). Reverse engineering takes an artefact, tries to see through its implementation and to generate description of its conceptual structure (Blaha, 1998). Database reverse engineering (DBRE) can be seen as a phase of IS maintenance, providing understanding of the system in case of lacking documentation (Chiang & Barron, 1995) or as part of software re-engineering (Blaha, 1998). In DBRE, database is studied to define how it has been structured and organized and what relationships exist inside it (Blaha, 1998). In general, DBRE includes the tasks of collecting semantics of the existing database and then visualising these in a conceptual format (Ndefo, 2017).

Hainaut (1991) divided DBRE into two phases:

1. Data structure extraction
2. Data structure conceptualization

Blaha (1998) divided DBRE into three phases:

1. Implementation recovery
2. Design recovery
3. Analysis recovery

In the Hainaut (1991) process, the database structure is first studied and gathered and then conceptualised into diagrams. In Blaha's process (1998) DB documentation is generated throughout the process where the first task is to get to know the system at hand and to make it easier to interpret the models, the second task is to study the primary and foreign key relations in the database and the last phase is to study the draft of the DB model and refine it, possibly by adding abstraction (Blaha, 1998). The process of extracting the underlying database structure is commonly referred to as semantic acquisition and it is conducted to gain broader understanding of the database domain. After the semantic acquisition is done the elements are classified and conseptualized using chosen notations, for example, ER diagrams, UML class diagrams or ORMs. The transformation process between the phases of semantic acquisition and conceptualization often rely on some extent of heuristic rules. (Ndefo, 2017.)

Blaha (1998) encouraged to use automation in the DBRE process. To obtain high level of automation in DBRE three quality issues must be discussed. These are the methodology for DBRE, the extent of semantics to gather and the desired performance of the the process. (Chiang & Barron, 1995). There does not exist one standardised relational database schema representation (Vukovic et al., 2017). Many DBRE approaches are not fully or at all automated and they require some extent of involvement or validation from humans (Ndefo, 2017).

There are multiple tools for DBRE that make the automation easier. Most tools support only one type of a database management system (DBMS) but some are capable of interpreting multiple database types. Both open source and commercial tools for DBRE have been developed. For example, SchemaSpy, SchemaCrawler and MySQL Workbench are open source tools that are capable of reverse-engineering a database structure into a diagram. Both SchemaSpy and SchemaCrawler are capable of interpreting multiple database types that utilise Java Database Connectivity (JDBC) drivers (SchemaSpy, n.d.; SchemaCrawler, n.d.). JDBC drivers translate the requests from Java applications to the relational DBMSs (Java SE Technologies - Database, n.d.) such as Oracle, PostgreSQL or DB2 (SchemaCrawler, n.d.). Additionally, DBMS dependent tools for DBRE exist, for example, MySQL Workbench that is compatible with MySQL databases (MySQL Workbench, n.d.). It is possible to combine multiple tools to gain higher level of automation: Vukovic et al. (2017) used three different tools for the DBRE process. These were the graphical user interface for the automation process operations, the DB structure interpreter and the UML generator.

All of the three examples of DBRE tools (SchemaSpy, SchemaCrawler and MySQL Workbench) create modified ER diagrams of the database structure (SchemaSpy, n.d.; SchemaCrawler, n.d.; MySQL Workbench, n.d.). For example, SchemaSpy produces modified, navigable ER diagrams of the database structure according to an XML file. The XML file includes the relationships between the primary and foreign keys of the database tables. SchemaSpy utilises Java 8 and is a single executable jar-file. The XML can be automatically generated of the whole database structure or separately created manually or through automation. Specified XML file can be used to define smaller part of the database, thus limiting excess visibility. Documentation is created in HTML format and it can be opened in browsers and shared with stakeholders (SchemaSpy, n.d.). Yamashita (2015) stated that SchemaSpy is a very cost-effective, comprehensive tool for data gathering, pointing out potential anomalies and creating metadata reports at the same time with ER diagrams. An example of database documentation generated by SchemaSpy version 6.1 is provided in Figure 5.

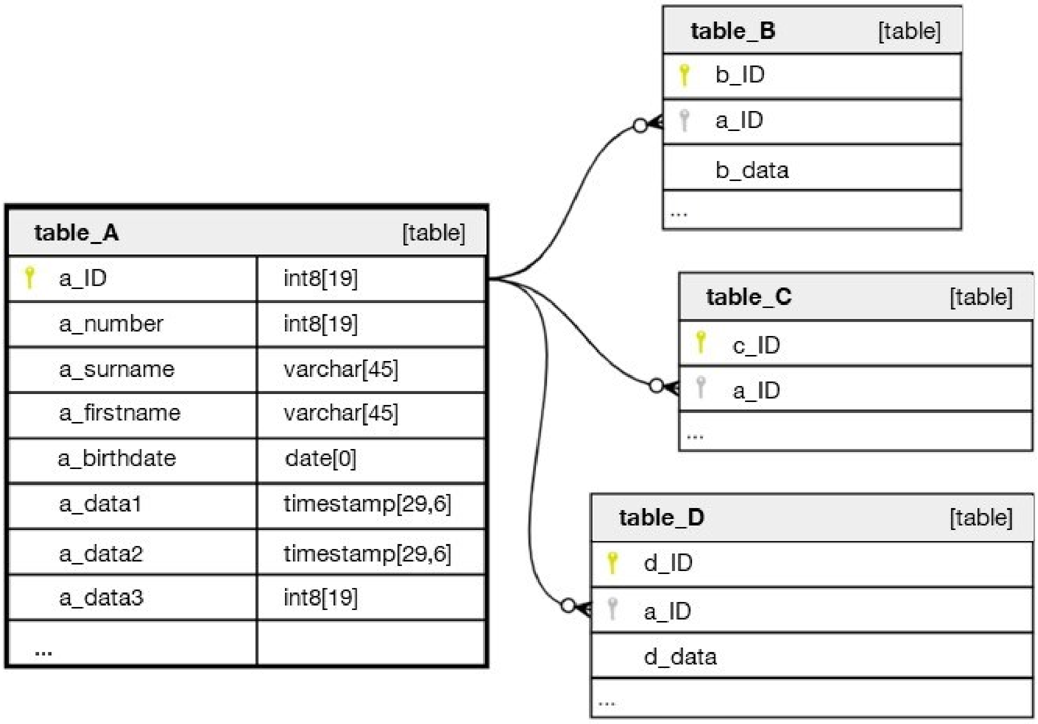

Figure 5. Example of a database documentation generated using SchemaSpy version 6.1.

SchemaSpy generates modified ER diagrams based on the interpreted database structure, as presented in Figure 5. Database tables are described as rectangular small tables: the table name is visible in the first row with the type of the database object, for example, 'table' or 'view'. Column names are listed with possible additional metadata about the column. For example, column 'a_surname' of 'table_A' in Figure 5 is of type variable character (varchar) and accepts up to 45 characters to the field. Primary keys are pointed out with yellow key icons and foreign keys as grey key icons. Tables B, C and D all have rows that are foreign keys to table A: primary key 'a_ID' of table A is saved in the B, C and D tables in corresponding rows. Relationship between the primary and foreign keys are described using crow's foot notation with an additional circle. This indicates that each 'a_ID' may be used in either zero, one or multiple rows in that table. In other words, entity described in table A must not have rows in tables B, C and D, but it may have one or many in all of them. Further examples of SchemaSpy documentation can be accessed via SchemaSpy website (SchemaSpy, n.d.).

3. Research Approach

Research methods that were used in this study are presented in this chapter. Research strategy was case study and both qualitative and quantitative methods were used to gather data and gain benefits of triangulation. Qualitative methods include questionnaires for the IS professionals and for the end-users and customers. The questionnaires contained both open questions to gather qualitative data and closed questions to gather quantitative data. Section 3.1 introduces case study and in Section 3.2 the case in this study is presented. Section 3.3 describes qualitative and quantitative methods and triangulation.

3.1 Case study

A case is a snapshot of a larger phenomenon. The primary goal of a research case is to either cumulate a new theory or to test an existing theory, thus verifying or disproving it in the case at hand. Generation of new theories in case studies is more common than verification of existing theories, especially if the topic of the case has gained less interest from the research community previously. (Myers, 2019.) Exploration is not the only reason of using case study. They also provide explanations for phenomena and further prove or disprove hypotheses. (Benbasat et al., 1987.) Themes that have already gained more attention from the research community can be tested using case study approach. In addition to testing the theories, case study enables comparison of multiple theories or exploring causalities it the case. (Myers, 2019.)

Benbasat, Goldstein and Mead (1987) defined that case study analyses a phenomenon in its natural context, combining different data gathering methods. They further define that the information is gathered from single or few entities, for example, people, teams or whole companies. Myers (2019) described that case study research gathers and delivers empirical evidence from one or many organizations. They add that the subject of the study should be analysed in context.

The phenomenon can be studied in a single case or in multiple cases. The case description can give further tools to define the nature of the phenomenon, thus enabling generalization of the conclusions. (Myers, 2019.) A single case or a few cases might not represent the other existing cases in adequate depth. Choosing multiple cases for the research might represent other cases better than single one. Yet, a case study is not a sampling research: this means that the case is not studied only to describe other cases but to study the phenomenon in the case at hand. (Stake, 1995.) Yin (1984) suggested that single case studies are suitable if (1) the research on phenomenon has not been previously accessible for the research community, (2) the case is crucial and beneficial for development of new theories or (3) it is a unique case.

Ideally a case study describes a current phenomenon and delivers something new for the research community and has an input to the scientific knowledge. (Myers, 2019.) A case study commonly contains multiple methods for data gathering to allow the benefits of triangulation, mainly the support for the researcher’s conclusions. The data that is to be collected should be defined before any site visits or other similar events such as interviews. One goal of case research is to gather enough data to make the formation of contextual complexity easier. (Benbasat et al., 1987.)

A case study should introduce enough evidence to describe the case for the reader (Myers, 2019). The evidence should be introduced and derived so the reader finds it easy to follow the flow of the study. This also adds to the reliability of the data and analysis. Suitable types of evidence for a case research are, for example, documents describing the case, statistic records, interviews, observations and physical artefacts. (Yin, 1984.) Adding citations from interviews is a suitable method to describe cases in text. Additionally, everything relevant to the phenomenon at hand should be presented in adequate extent. Case studies should be honest to real life, and facts that do not necessarily support the theory should also be presented and discussed. (Myers, 2019.)

Benbasat et al. (1987) defined 11 key characteristics of a case study:

1. Phenomenon is studied in its natural environment
2. Multiple methods of data collection
3. One or few entities as a unit of analysis
4. Detailed study of the unit complexity
5. Researcher has a positive attitude towards explorative studies
6. No manipulation or experimental controls
7. No predefined independent and dependent variables
8. The researcher is responsible of forming the derived results
9. Changes to the site selection possible if considered crucial
10. Answers well for the ‘why’ and ‘how’ research questions
11. Contemporary events are in the scope of the study

Benbasat et al. (1987) discussed the use of case study in IS research and provided suggestions for the conduction and evaluation of the studies. They stated that the case research strategy adequately seizes the knowledge of the stakeholders and allows the development of theories from the understanding. Yin (2018) listed five aspects that should be considered in a case study:

1. Research questions
2. Possible propositions
3. Unit(s) of analysis
4. The relationship between the data and the propositions
5. Interpretation criteria

The unit of analysis can be, for example, individuals, groups, projects, decisions or organizations. The research questions can be analysed to derive the suitable unit of analysis. (Benbasat et al., 1987.) A positivist approach to case study such as Yin’s (2018) contains the use of propositions and hypotheses in the research and the emphasizing of the quality assessment by defining the validity and reliability of the study. Additional approaches of case study are, for example, interpretive and critical case studies, which do not necessarily require the use of propositions or hypotheses and do not use the words validity and reliability in their quality assessment. (Myers, 2019.)

3.2 Case description

In 2019 a Finnish IS organization had started receiving requests from their business customers. The customer organizations stated that the documentation they had of their CIS’s reporting database was inadequate, outdated and partly missing entirely. They requested updated, consistent and comprehensible documentation of their database, including metadata summaries and ER diagrams of the reporting database structure. The developers of the IS company knew it was time for decisions. Database documentation had stagnated a long time ago and further database tables and columns were added version deployment after another. The first option was that the developers rolled their sleeves and somehow updated their work processes to enhance the documentation in the future. Manual process of documentation updates had cumulated to be too laborious - not to speak about the debt of hours the developers would have to work to make up for the undocumented changes. However, the developers knew of another option and chose to chase it: automation.

In the fall of 2019 the company started a project to redesign their reporting database through automation. In addition, the database documentation was simultaneously generated in the automation process. The new reporting database views and tables were automatically constructed by interpreting the CIS's database structure while following strict data security policies. In the same process, open source tool Schemaspy was used to generate metadata descriptions and ER diagrams of the new reporting database. The company decided to use Schemaspy because they had positive experiences of using it prior to the project in 2019.

The automated solution was initially designed to be run either manually by developers after each version deployment or autonomously with supportive tools such as Jenkins. When the new solution was introduced to the customers they stated they were very pleased with the new metadata overview and the diagrams and that these were just what they needed and everyone seemed pleased. What the customers needed the documentation so eagerly for was not entirely common to the developers. The newly automated solution for database documentation was considered beneficial for both the case company and the customer organizations. The developers started wondering how exactly the existence of updated documentation would affect all stakeholder groups' work – including the customers and the developers.

3.3 Qualitative and quantitative methods

In addition to the case study strategy, other qualitative research methods are, for example, ethnography, field studies and interview studies. Research design in qualitative research emphasizes the phenomenological nature the study has - describing the phenomenon in its contexts. In comparison, quantitative research methods focus on measuring the phenomena, aiming for generalization of the results. The distinction between qualitative and quantitative methods and data are strongest in the data gathering process, the data recording policies and analysis. (Brannen, 2017.)

The quantitative research methods collect measurable data that is presented in numerical form. Statistical methodologies can be used to analyse the data or to create different types of presentations. (Creswell & Creswell, 2017.) This is acquired using, for example, sampling methods, variable control and randomization (Newman, Benz, & Ridenour, 1998). The qualitative research methods collect data that describes the phenomenon: focuses are the context, the humans and their experiences. In qualitative research these are not compressed into numeric form. (Taylor, Bogdan, & DeVault, 2015.) Research validity is increased if methodologies of both quantitative and qualitative research can be used and most importantly if the right approach is chosen for the right research question. (Newman, Benz, & Ridenour, 1998.)

Case studies commonly utilise only qualitative methods in data gathering process but quantitative methods such as questionnaires are also possible as the main method of data gathering or as an additional method (Dubé & Paré, 2003). Questionnaires gather data, for example, about the required demographics of people and their attitudes, feelings and behaviour (De Vaus, 2002). Questionnaire questions can be either open or closed which means that the respondent can either answer freely using text or by choosing the most appropriate option of the predefined ones. Open format of questions provides possibility of finding new insights to the phenomenon but are more laborious to answer for as the respondent and to analyse as a researcher. Closed questions on the other hand provide simplicity of analysis and are faster to answer to as a respondent but they do not provide lots of new insights to the phenomenon. (Armstrong, 2009.) For example, Likert scale measures attitudes of participants. Sometimes a Likert scale is analysed as an interval scale, although it is more suitable to consider them ordinal scales. (Edmondson, 2005.) Likert scales consist of statements regarding certain phenomena, either experienced or considered hypothetically by the participants. Participants choose whether they disagree or agree with the statements and to what extent, for example, they could “strongly disagree” or “somewhat disagree” with a statement. (Singh, 2006.)

4. Research Implementation and Analysis

Two separate stakeholder groups were identified from the perspective of the DB documentation automation and the possible users of DB documentation in the case company. These were the case company's customers and the case company personnel. The research question was:

RQ: How does (automated) DB documentation affect the DB users' work tasks?

The research question was analysed by studying the effects on work tasks of the two recognised stakeholder groups in the case. The main research question was answered through the analysis of the sub questions 1 and 2 that studied the experiences of stakeholder groups separately. Qualitative data was analysed with content analysis. Section 4.1 describes the customer questionnaire implementation and Section 4.2 the personnel questionnaire implementation. Section 4.3 describes the used analysis methods in detail for each questionnaire.

4.1 Customer questionnaire

Questionnaire for the customers and end-users gathered data for the RQ1. Sub question of the main RQ was:

RQ1: How (automated) DB documentation affects the end-users' work tasks?

Open questions were used to gather more understanding of the work tasks of the end-users, especially what work tasks were supported partly or fully with the database documentation and how often they reference it. Closed questions were used to gather data of the known effects the database documentation has based on the prior literature. The closed questions of the questionnaire consisted of Likert statements that were answered by choosing the most suitable number between 1 and 5:

1: Strongly Disagree
2: Disagree
3: Neither Agree nor Disagree
4: Agree
5: Strongly Agree

Questionnaire was heuristically piloted by two case organization representatives, who interpreted the questionnaire from the point of view of customers. Minor corrections were made according to the findings of the heuristic evaluation. All final customer questionnaire questions are listed in Appendix A: Customer Questionnaire. Questionnaire was in Finnish because each of the possible participants were native Finnish speakers.

The customer questionnaire of this study was sent to the participants as part of a bigger questionnaire that contained questions from two other Master's thesis studies by (Kemppainen, 2020; Uhlgren, 2020). Therefore the overall length of the questionnaire was over 20 questions, whereas questions regarding the DB documentations effects on work tasks consisted of only seven questions. Five of the seven questions were open questions where the participants could answer freely. Two of the seven questions contained 11 Likert statements per each question. Same statements were asked to be valuated regarding both the prior documentation the customers had in their disposal and the new documentation they had after the automation.

The customer questionnaire link was sent to all customer organization representatives who had prior experience of the reporting DB and the new DB documentation solution. Invitation to take part in the questionnaire was sent to 23 possible participants through email on May 8, 2020. The questionnaire was created using Webropol service. The participants were given a week to submit their answers. Reminders of the questionnaire were sent twice. The first reminder was sent on May 13, 2020 and the second on May 14, 2020. The questionnaire was closed on May 15, 2020.

4.2 Personnel questionnaire

The questionnaire for the IS professionals of the case company gathered data for the RQ2. Sub question of the main RQ was:

RQ2: How does (automated) DB documentation affect the IS professionals' work tasks?

The questionnaire was piloted by one case organization representative. Minor corrections were made according to the findings of their evaluations and answers. Questionnaire consisted of multiple open questions and a few closed questions. All 13 questionnaire questions are listed in Appendix B: Personnel Questionnaire. The questionnaire was delivered in both Finnish and English and the participants could choose which language to use.

The participants were first asked three simple questions that gathered demographic data. In the first question the participant chose the most suitable option to describe the team they work in. In the second question participants answered either "Yes" or "No" to whether they are accustomed to the type of database documentation SchemaSpy creates. Similarly in the third question the participants notified whether they had used DB documentation as a supporting tool in their work or not. Last 10 questions were open and the participants could describe freely how the documentation has affected or could affect their work tasks. The questions were planned to steer the participant to assess the effects in comprehensible way. For example, the participants could compare which of their work tasks could become easier, harder, faster or slower and whether the up-to-date, consistent database documentation could erase some of their work tasks completely. Additionally, the participants described how important DB documentation is for their work.

The personnel questionnaire was sent internally to multiple mailing lists through email on May 14, 2020. Additional notification of the study was done in internal communication tool at the same day. Invitation to participate in a study was sent to every employee who were expected to work with databases in some extent in their position. The participants were given a link to organizational wiki containing examples of SchemaSpy DB documentation and another link to the questionnaire. Wiki page contained example pictures of metadata overviews and ER diagrams from open source tool SchemaSpy. The questionnaire was created using Webropol service. The participants were given four days to answer for the questionnaire. Reminder of the questionnaire was sent through email on May 15, 2020. The questionnaire was closed on May 17, 2020.

4.3 Analysis methods

Content analysis is a data analysis method for either qualitative or quantitative data (Krippendorff, 2018). The content analysis focuses on different types of documents, varying from example written interview reports, questionnaire answers, video, pictures or simple communication files in text. Content analysis is a systematic approach to study possible patterns that can be found from the different document types. (Bell, Bryman, & Harley, 2018.)

Studying the documents is done systematically and the significant pieces of content are 'coded'. The codes are assigned to contents to highlight the importance of certain pieces of content and to group similar contents together. Although some minor differences might exist between different academic fields regarding the use of techniques in content analysis, all follow similar procedures of studying the documents carefully and coding their contents. (Hodder, 1994.) Every content analyst should pay attention to the nature of the analysed documents and their suitable context, what data has been gathered, from whom and how it has been measured. Additionally, a content analyst should keep in mind their research questions and limit their analysis to matters that concern their research. (Krippendorff, 2018.)

There exists both quantitative and qualitative content analysis but their usual approaches differ (Krippendorff, 2018). Quantitative content analysis can discuss, for example, prevalence of words or images in the data set (Kracauer, 1952). Quantitative content analysis is a deductive approach to data analysis, which means that it has a hypothesis and the codes have been defined prior to the analysis process. Qualitative content analysis is done with an inductive approach, meaning that there is no hypothesis. The data is analysed with the aid of the research questions and an analyst defines the codes in the process of the analysis. The inductive approach takes more time to conduct that deductive approach: The inductive approach might require multiple iterations where codes are reworked before the analysis is ready. (White & Marsh, 2006.) The quantitative content analysis focuses on finding the manifest meanings behind data (Kracauer, 1952) whereas the qualitative content analysis focuses on the latent meanings of data (White & Marsh, 2006).

In this study both sets of qualitative questionnaire data were analysed using inductive content analysis. The qualitative data was recorded in written format and the content analysis was considered applicable and suitable for the interpretation of the open question answers. Inductive approach was chosen for both data sets to allow recognition of patterns and phenomena in the stakeholders' answers without predefined codes or themes. The inductive content analysis also enables the exploration of latent meanings from the data.

The customer questionnaire contained statements that collected Likert data. Depending on the amount of gathered data the experiences can be analysed using either qualitative or quantitative methodologies. In this case study the Likert data was considered descriptive statistics and it was interpreted as part of the qualitative content analysis due to the small sample size. This allowed exploration of patterns from the experiences of the customers.

4.3.1 Analysis of customer questionnaire data

The customer questionnaire contained nine questions of which two were demographic questions, five were open and two consisted of 11 Likert items which participants evaluated based on their own experiences. The first Likert scales items concerned the prior DB documentation the customer might have had in their disposal and the second Likert scale items concerned only the new, automated DB documentation. It was possible for participants to not evaluate either of the two last questions if they do not have experience of its use.

The five open questions were analysed using content analysis. The first phase was to go through the data and gain an overview to the experiences of the end-users. The questionnaire was sent to 23 possible participants of whom six responded to the questionnaire. All of the six data sets were considered acceptable for further analysis. Data was first labelled, then the labels were organized into a table. These labels were further analysed, reworked and combined. After this, the combined labels were compared with the original labels and their meaning to make sure their consistency.

After content analysis of the first open questions the Likert scales were analysed. Because the sample size was small, the answers were considered descriptive statistics and analysed qualitatively and inductively to find patterns and latent meanings behind the data, similar to content analysis.

4.3.2 Analysis of personnel questionnaire data

The content analysis of the personnel questionnaire was started by going through the gathered data and validating it. 16 participants from the case company submitted their answers for the questionnaire and gave their consent to take part in the research. Two of these participants answered only for the three first demographic questions and not to any of the open questions. Their input was left out of the analysis and only 14 sets of answers were used in the next phases of analysis process.

The answers of each individual participant were collected into one single file per each employee. The first phase of content analysis was coding the contents, keeping in mind the research question. After their initial use the same codes were used for the next documents as well until all 14 answer documents had been coded. The patterns that were recognised in multiple documents were coded and additional points of interest were marked up for further analysis. Multiple increments were required to unify the used coding scheme for each document.

In the next phase the separate documents were unified. In the personnel questionnaire the participants had chosen the most suitable team they worked in. Knowledge of the teams was used to group codes under each corresponding team, thus combining multiple answers. Physical paper notes were used to first list all of the unique, recognised codes for each team. Redundant notes were not written: if multiple participants had brought the same theme up, this was marked into the physical notes as well. The codes were further combined and clarified before each team's codes were recorded into a spreadsheet.

In the next phase of content analysis the team division was erased and the codes were unified further to form overview of the effects the DB documentation has on the work tasks of IS professionals. The codes were also labelled under certain themes that represent the codes under them. Lastly the coding in the original answer documents was inspected and compared to the unified codes and themes to make sure the codes represented the data and that the codes had not evolved their meaning too much.

5. Findings

The findings of the data analysis are presented in this chapter. Section 5.1 presents data from the questionnaire that the case company's customers responded on. Section 5.2 presents data that was gathered from the case company's IS professionals through a questionnaire. The questionnaire questions are listed in Appendix A: Customer Questionnaire and in Appendix B: Personnel Questionnaire.

5.1 Effects on end-users' work tasks

Six respondents of the possible 23 participated in the customer survey. The list of respondents and their demographic data and experience of the prior and new DB documentation solutions are listed in Table 2. The participants work in different customer companies and their experiences of the new and prior solution varies between deployment situations in the organizations. For example, some participants have used both solutions, whereas some have used only the new solution. Additionally, one participant had not yet tried the new solution themselves. The substance field of the case company's customers was not published in this study. The knowledge of the substance field and its processes, and other similar ISs might be useful for the end-users to fully benefit from the DB documentation but the knowledge is not necessarily required.

Table 2. Demographics of respondents.

Respondent	Work experience from the substance field (Years)	Work experience of working with databases (Years)	Experience of the prior DB documentation	Experience of the new DB documentation
R1	3	13	No	Yes
R2	6	3	Yes	Yes
R3	12	20	No	Yes
R4	5	3	Yes	Yes
R5	5	30	Yes	No
R6	12	20	No	Yes

As listed in Table 2, the respondents R1, R3, R5 and R6 had over ten years of experience from working with databases and less experience of working in the customer organizations' substance field. The respondents R2 and R4 had shorter experience of working with databases that they had on the substance field. Only R2, R4 and R5 had used the prior database documentation in their work. R5 is the only respondent who had not used the new DB documentation in their work tasks yet. R2 and R4 were the only ones with experiences on both prior and the new DB documentation. The experiences of the prior DB documentation experiences are summarized in Table 3.

Table 3. Experiences of the prior database documentation.

Statement	R2	R4	R5
Prior database documentation is up-to-date	Strongly Disagree	Strongly Disagree	Strongly Disagree
Prior database documentation includes all the information I need	Strongly Disagree	Strongly Disagree	Disagree
Prior database documentation includes incorrect information	Neither Agree nor Disagree	Strongly Disagree	-
No erroneous interpretations happen when I utilise the database documentation	Neither Agree nor Disagree	Neither Agree nor Disagree	-
The database is easy to use	Agree	Neither Agree nor Disagree	Neither Agree nor Disagree
Getting to know the database structure is easy	Agree	Strongly Disagree	-
Documentation supports me in my database use	Disagree	Strongly Disagree	Strongly Disagree
Using the documentation I quickly find the database schema, table or view I want to	Neither Agree nor Disagree	Strongly Disagree	-
It is easy to make database queries to the database	Agree	Strongly Agree	Agree
Database documentation's diagrams are easy to interpret	Agree	Neither Agree nor Disagree	-
It is easy to see the connections between the tables / views from the database documentation's diagrams	Disagree	Strongly Disagree	-

When comparing the experiences from Table 3 regarding the answers of R2, R4 and R5 about the prior DB documentation, certain dissatisfaction is visible. All respondents consider that the DB documentation was outdated, it did not include all of the information they need in their work tasks, it did not support them in their DB use and the DB structure is hard to interpret from the documentation. Although both R2 and R4 have similar experience of DB use, they experienced the DB use in different ways. R2 considered it easy to interpret DB structures from the prior documentation whereas R4 completely disagreed. Creating DB queries was considered easy by all respondents using the prior documentation.

Although only three of the six respondents had used the prior DB documentation, five of them had used the new solution. Experiences of the new reporting DB documentation are summarized in Table 4.

Table 4. Experiences of the new database documentation.

Statement	R1	R2	R3	R4	R6
New database documentation is up-to-date	Agree	Agree	Disagree	Strongly Agree	Disagree
New database documentation includes all the information I need	Strongly Disagree	Strongly Agree	Disagree	Strongly Agree	Disagree
New database documentation includes incorrect information	Neither Agree nor Disagree	Neither Agree nor Disagree	Strongly Disagree	Strongly Disagree	Neither Agree nor Disagree
No erroneous interpretations happen when I utilise the database documentation	Disagree	Disagree	Neither Agree nor Disagree	Agree	Neither Agree nor Disagree
The database is easy to use	Disagree	Agree	Agree	Agree	Agree
Getting to know the database structure is easy	Disagree	Agree	Neither Agree nor Disagree	Disagree	Agree
Documentation supports me in my database use	Disagree	Strongly Agree	Disagree	Strongly Agree	Agree
Using the documentation I quickly find the database schema, table or view I want to	Disagree	Agree	Disagree	Strongly Agree	Neither Agree nor Disagree
It is easy to make database queries to the database	Agree	Agree	Strongly Agree	Strongly Agree	Neither Agree nor Disagree
Database documentation's diagrams are easy to interpret	Agree	Strongly Agree	Disagree	Agree	Agree
It is easy to see the connections between the tables / views from the database documentation's diagrams	Agree	Strongly Agree	Neither Agree nor Disagree	Strongly Agree	Neither Agree nor Disagree

The respondents had rather varying perceptions of whether the new reporting DB documentation is adequate, as visualized in Table 4. For example, whether DB documentation is up-to-date, if it includes all the information the customer needs or if it is easy to get to know the DB structure. Only statements where respondents R1, R2, R3, R4 and R6 are rather likeminded are that it is easy to do queries to the database and that it is easy to interpret the ER diagrams for relationships between tables and views in the DB. The experiences of customers seems to be diverse.

The versatility of the answers could have many reasons. Firstly, not all respondents who have used the new documentation have used the prior version. This might influence their interpretations of how they see the new solution. Secondly, the respondents might compare the current documentation to other formats of documentation they might have used in their previous work places, delivered by different service providers. Additionally, it is impossible to say, if the documentation they compare the new DB documentation company with has been created by hand to serve the customer needs in detail or whether automation has been used. Lastly, at the time of the questionnaire the different customers had had the new documentation for shorter and longer times, some might have only seen it once. Some respondents might have familiarized themselves with the documentation format in more detail than others. Thus, all answers regarding the new solution are not fully comparable.

The only respondents who had used both the prior and the new DB documentation to support their work tasks were R2 and R4. Comparing their experiences between the prior and the new solution provides insight into whether the documentation has been of use to the end-users. The comparison is presented in Table 5.

Table 5. Combined experiences of the prior and the new database documentation.

Statement	R2		R4	
	Prior	New	Prior	New
Database documentation is up-to-date	Strongly Disagree	Agree	Strongly Disagree	Strongly Agree
Database documentation includes all the information I need	Strongly Disagree	Strongly Agree	Strongly Disagree	Strongly Agree
Database documentation includes incorrect information	Neither Agree nor Disagree	Neither Agree nor Disagree	Strongly Disagree	Strongly Disagree
No erroneous interpretations happen when I utilise the database documentation	Neither Agree nor Disagree	Disagree	Neither Agree nor Disagree	Agree
The database is easy to use	Agree	Agree	Neither Agree nor Disagree	Agree
Getting to know the database structure is easy	Agree	Agree	Strongly Disagree	Disagree
Documentation supports me in my database use	Disagree	Strongly Agree	Strongly Disagree	Strongly Agree
Using the documentation I quickly find the database schema, table or view I want to	Neither Agree nor Disagree	Agree	Strongly Disagree	Strongly Agree
It is easy to make database queries to the database	Agree	Agree	Strongly Agree	Strongly Agree
Database documentation's diagrams are easy to interpret	Agree	Strongly Agree	Neither Agree nor Disagree	Agree
It is easy to see the connections between the tables / views from the database documentation's diagrams	Disagree	Strongly Agree	Strongly Disagree	Strongly Agree

There exists a significant pattern for both R2 and R4 experiences in the Table 5. The prior solution was perceived outdated and inconsistent and it additionally lacked some information. It was not considered to support the DB use of the end-users and that the structure of the database and its primary and foreign key relationships were hard to interpret. Correspondingly, the new solution was considered to perform positively in all of previous cases. The only significant differences between the answers of R2 and R4 are that R4 thought that false interpretations do not happen when using the DB documentation whereas R2 states that they do happen. This on the other hand can be a result of wrong interpretation of the statement or their prior experience of DB use. There also exists few statements that both respondents have answered similarly for both prior and the new solution, for example, they stated that querying a DB is as easy with both of the solutions. In summary, the R2 and R4 both valuate the new DB documentation to support them in their work tasks either better than the prior DB documentation or in similar manner.

In addition to the experiences analysed from the Likert scales, the respondents also gave open answers regarding their DB documentation usage and in what ways they are using it. After the analysis of each open question answers the results of content analysis are showcased in Table 6.

The respondents utilise DB documentation in varying frequencies. R2 stated that they would use DB documentation approximately two times a week. R1 would use DB documentation whenever they are doing queries to the DB. R3 points out that the need for DB documentation decreases as knowledge of the database grows. They state that especially in the beginning of reporting DB use they required lots of support from the documentation. Also R6 required more support from the DB documentation when starting to use the reporting DB but has needed it less after they got more and more experience. R3 and R6 have rather respectable amount of experience from working with databases, so this might have an impact on their perception. R4 states that they did not yet have documentation of the DB but the case company had introduced and familiarized the customers with the new solution. R4’s company was waiting for the documentation at the time of the questionnaire but they had answered according to their impressions of the solutions. Also R5 uses DB documentation very scarcely because they did not have the new documentation solution yet. R5 added that they had had to contact customer support to gain knowledge of the DB since the prior documentation was not up-to-date. Whereas R3 and R6 prefer learning the DB structures through work, both R1 and R2 would use documentation rather often. The usage might be very dependent of the precise nature of each employee’s work tasks.

The end-users also listed in which work tasks they used the support of documentation. R4 has used DB documentation to gain knowledge of the DB structure. R1 used DB documentation for the needs of third parties. These include, for example, other IS providers where interfaces need to be studied. R3 and R4 use DB documentation in different data reporting tasks. Also R5 uses DB documentation for reporting tasks but additionally for modelling and documenting themselves. Reporting is also mentioned by R6 who additionally had used the documentation in integration work tasks. R2 uses DB documentation in their projects containing both testing and integration support. Thus, the most common tasks that DB documentation is utilised in are tasks of reporting and integration. Although only R4 mentioned tasks of gaining understanding of the database, both reporting and integration planning contain tasks of interpretation and gaining understanding.

The DB documentation was considered necessary or useful support for multiple work tasks. For R1 the DB interpretation tasks conducted for the needs of third party IS providers require DB documentation. R3 pointed out that for the reporting tasks the DB documentation is necessary support and R6 considers it useful support. Also R5 named reporting as one of the tasks that require DB documentation but in addition to being helpful, documenting the database and modelling it also require the DB documentation. R4 requires DB documentation to use it when supporting other users in the customer organizations.

The customers had utilised DB documentation for interface design and interpretation too. Contacts from the case company's CIS to other ISs have to be carefully planned and it must be possible to trace and solve possible problems - DB documentation is a good support for these tasks. R1 and R5 admitted that the provided DB documentation has been useful in the design and interpretation of interfaces to other systems and reporting services. R2 also stated that DB documentation makes it easier to design interfaces regarding separate systems and the integration between then. R6 has also used the documentation to plan interfaces with third parties. Although R4's company does not yet have DB documentation, they believed it would be useful if, for example, some interface requires further development and whether this would be solvable using the reporting DB.

According to the respondents the DB documentation has enabled use of external tools, SW and ISs. R1 noted that the DB documentation had enabled them to use an external analytics systems for their data analysis needs and R3 states that DB documentation has allowed the use of their integration tool. R4 believed that DB documentation would in cases of certain SW malfunctions allow the customers to generate temporal solutions to the problem themselves. R5 had used the prior DB documentation in migration tasks, although they stated that this has sometimes been arduous due to the inconsistency of the documentation. Combined results of both content analysis and descriptive statistics are presented in Table 6.

Table 6. Effects of DB documentation on the work tasks of customers and end-users.

Category	Effect
Work task support	Supports integration planning, testing & support
	Supports interface design and interpretation
	Easier to make queries to the DB
	Enables reporting tasks
	Enables creation of modified and augmented DB documentation
Knowledge of DB structure	Easier to interpret relationships in the DB
	Enhances knowledge of the DB

Table 6 contains the work tasks and effects on DB knowledge the documentation was analysed to have in content analysis process. DB documentation was considered to make interpretation of DB structure easier by allowing exploration of DB tables, views, columns and relationships. Especially primary and foreign key relationships were considered to be easier to interpret and analyse for other work tasks. Additionally, the availability of the documentation was considered to enhance the customers understanding of the DB structure and thus the functions of the whole system.

Especially the work tasks regarding integration and interfaces were considered to require up-to-date, consistent documentation as their support. The documentation was reported to be useful support for work tasks of reporting. The customers also experienced that making queries to the database had become easier trough the availability of consistent DB documentation and this also enables them to create their own modified DB documentation, specifically tailored for the customers' internal needs.

5.2 Effects on IS professionals' work tasks

16 employees of the case company participated in the personnel survey and 14 sets were selected for further analysis. From the predefined teams only customer support and product development received multiple answers. Thus, teams were rearranged to enable anonymized analysis for the other data as well. Other participants' work tasks were considered to be more similar to product development than customer support and were thus combined with the first. Two separate analysis teams were formed: "Customer Support" with four participants and "Development, DevOps and Product Management" or shortly "Combined Development team" with 10 participants. Demographics of each analysis team are listed in Table 7.

Table 7. Demographics by Analysis group.

Analysis team	Number of participants	Was familiar with database documentation	Had used database documentation in work tasks	Importance of database documentation
Customer Support	4	Yes: 3 No: 1	Yes: 4 No: 0	Not important: 0 Important: 1 Very important: 3
Development, DevOps and Product Management	10	Yes: 9 No: 1	Yes: 6 No: 4	Not important: 1 Important: 6 Very important: 3
All participants	14	Yes: 12 No: 2	Yes: 10 No: 4	Not important: 1 Important: 7 Very important: 6

As presented in Table 7, two of the 14 respondents stated that they were not accustomed to the format of database documentation, including ER diagrams and metadata overviews. Four respondents from the combined development team had not used database documentation to support them in their work tasks. All respondents from customer support had used database documentation to support their work tasks whereas one of them was not familiar with the SchemaSpy generated metadata overviews and ER-diagrams. One respondent from the combined product development team was similarly not accustomed to the format of the DB documentation.

13 respondents viewed the DB documentation to be either somewhat important or very important for them. Six of the 14 respondents stated that DB documentation would be beneficial for their work and it is thus important for them. Seven respondents valuated consistent, up-to-date DB documentation very important or crucial for their work tasks. Only one respondent stated that up-to-date and consistent DB documentation would not have any effects on their work tasks and is thus unimportant for them.

One participant from customer support noted that consistent database documentation would erase one of their occasional work tasks. According to them, customers might ask for reports regarding primary and foreign key relationships in the database. Certainly such tasks would be erased, especially with the documentation provided to the customers as well. One participant from combined development team stated that some problem tracing tasks could be erased from larger processes with the support of documentation. Another thought that documentation would erase work tasks where they search and combine knowledge from multiple sources. Other respondents did not notice whether their work tasks would be erased

The participants from the customer support did not consider that database documentation would make any of their work tasks harder or slower whereas three participants from the combined development team had some concerns regarding the possible additional work tasks the generation of DB documentation would accrue. These concerns will be discussed later in this chapter.

The customer support participants would be using consistent, up-to-date database documentation in their work occasionally and when needed. Two of the participants further specified they could use it weekly. Participants in the combined development team would be using DB documentation more diversely. Two of them considered they would be using documentation almost daily, three almost weekly and the other five more scarcely and irregularly, approximately once a month or less.

The customer support considered provided examples of automatically generated Schemaspy DB documentation easy to interpret. One of them added that the format of ER diagrams was common for them from the time of their studies. Similarly the combined development team perceived documentation rather easy to interpret. Two participants from the combined development team pointed out that ER diagrams are familiar to them from some management tools they have used before. One respondent noted that the ER diagrams are also used in DB design process and are thus easy to interpret.

The respondents from the customer support stated they would use DB documentation to support their work tasks if consistent, up-to-date DB documentation would be available for them. Although four of the respondents in combined development team had not used DB documentation in their work tasks, all of the 10 participants considered they would use documentation to support their work, given it is consistent and up-to-date. The specific tasks that were considered to benefit from DB documentation are listed in Table 8.

Table 8. Work tasks that would benefit from database documentation.

Group	Description of specific task	Respondents (of 14)
Customer Support	Scripting	3
	Database queries	3
	Data updates	2
	Finding foreign and primary key relations	1
Development, DevOps and Product Management	Software design	3
	Software development	3
	Database changes	2
	Bug fixes	2
	Data model design	1
	Specifications	1
	Software architecture design	1
	Database maintenance	1
	Database and system compatibility	1
	Optimization	1

The groups of work tasks that would benefit from consistent, up-to-date DB documentation were recognised for both analysis groups as listed in Table 8. The participants from customer support stated that DB documentation would benefit them in many of their daily tasks. In script writing process the documentation would provide certainty and erase multiple iterations from the process because script could be planned in more precision. Additionally, support personnel stated that documentation would help them in creating complex DB queries and erasing the simple queries to check database structure. Furthermore, data updates would similarly benefit from the faster process of finding primary and foreign key relationships from the database.

Development, DevOps and Product Management team's most common tasks that would benefit from DB documentation was the tasks of SW design and SW development, including bug fixes and DB changes. Additionally, the combined development team pointed out that, for example, data model and SW architecture design, specification work, DB maintenance, optimization and DB-system-compatibility checks would benefit from the documentation. One respondent also added that any work task that is considered complex in the context of their work could benefit from the documentation.

A few opinions regarding the usefulness of metadata overviews emerged. One respondent from the product development team stated, that they see the metadata overviews not as useful in their work tasks as they see the ER diagrams. Other respondent from the customer support team highlighted how useful the availability of consistent metadata overviews would be for their work tasks concerning scripting:

> *"Also knowing the metadata would help because one could allocate enough characters for example, into some scripts that use varchar."*

One respondent also pointed out that DB documentation would affect positively their understanding of the CISs structure and therefore they could focus more on the most important tasks:

> *"Consistent documentation would provide a chance to focus on the problem instead of learning the database structure and its use."*

Consistency of the DB documentation is considered to have positive effects throughout the responses. One participant pointed out that customers' first impressions of the database are affected by the documentation they are provided. According to the respondent, customers have gotten very positive impressions of the SchemaSpy documentation. Two respondents also point out that if they could fully trust the DB documentation this would be beneficial for their work. Additionally, delivering comprehensive documentation to the customers could increase transparency and create stronger bonds between the provider and the customer.

Although many respondents see only positive effects the automated DB documentation could have, some negative effects had been recognised as well. For example, infrastructure maintenance might become harder in some aspects and additional tools the documentation possibly requires could create more complexity to some processes. Additionally, environment creation might get slower. Some concerns rose about the documentation maintenance, although this was automated. Two respondents noted that higher standards for database metadata documenting and documentation maintenance might increase workload of some development work tasks.

What seems to recur in the answers is the courage the trustworthy DB documentation gives to an employee, for example:

> *"It would be easier to get started with tasks you are less accustomed to if there exists database documentation and you can trust it."*

> *"-- If I would have to do something new or bigger operations in the database, then I would [use database documentation]"*

Database documentation was considered stagnated at the time of the questionnaire. Most participants saw only opportunities the new documentation could create and how it would change their work processes into better. One respondent from the combined development team also pointed out that although it would be nice to have documentation, they have survived without it too. Lack of documentation was something they were accustomed to. Three participants told that they did DB queries and inquired their co-workers to gather the knowledge that would otherwise be quickly viewable in the DB documentation. One respondent added that DB documentation could also provide possibilities to try new work tasks:

> *"[Database documentation] would make it easier to move over to, for example, coding or testing or to such tasks where a quick glance into the database structure would be needed"*

Two main categories of effects were recognised when analysing the questionnaire data from both analysis teams. The first main category is effects on workload and the second is effects on the use of databases. The recognised effects are presented in Table 9. The effects were divided into two categories: effects on workload and effects on database use.

Table 9. Main effects of database documentation on work tasks.

Effect category	Effect description	Respondents (of 14)
Workload	Easier to conduct work tasks	10
	Supports problem solving	8
	Faster to conduct work tasks	7
	Lowers cognitive burden	6
	Enables independent work	5
	Lowers the amount of contacts from the customers	4
	Enables more precise planning of work tasks	2
Use of databases	Less false interpretations of database structure	9
	More knowledge and understanding of the database structure	8
	Less queries to the database required	8
	Interpreting new database structure becomes easier	4

The summary of the DB documentations effects on the work tasks of IS professionals is provided in Table 9. The most notified effect on workload according to the respondents was that DB documentation would make their work tasks easier to conduct. 10 of the 14 respondents stated, that DB documentation would make at least some steps of their work tasks easier by erasing uncertainty and providing one source of DB and system knowledge they could use. Additionally, eight of the 14 respondents considered that the DB documentation would support their problem solving in general, since they could interpret documentation and search the most appropriate information with less effort, less database queries and less go-through of source code. Like work tasks would become easier and problem solving would be supported, seven respondents considered that multiple tasks would also be possible to carry out faster due to the smaller amount of required investigation. Two respondents also pointed out that DB documentation would allow them to plan their work tasks more precisely, thus saving their time and effort.

Six of the 14 respondents pointed out that DB documentation would lower their cognitive burden. Without consistent and updated DB documentation the employees often try to remember as much of the DB structure they could. This phenomenon is probably consequence of the long period of stagnated documentation where fastest way to check information is to remember as many DB's table names, columns and relationships as possible and try them out while conducting work tasks. One respondent summarizes the cognitive issue:

> *"Although most of them [Database tables or views] are named reasonably, no one can know the purpose of each. -- There are too many data items for any human to remember. Documentation partly exists, also table comments regarding the new tables but not in all of them, some have no [Documentation] anywhere"*

Five of the 14 respondents considered that consistent DB documentation would lower the workload of many other employees as well in addition to them. Respondents pointed out that they commonly communicate to their co-workers inside their own team and in other teams to gather knowledge of the certain DB structure they would like to know better. Inquiring co-workers was considered an excess burden for both the one asking around and the one that is questioned. One respondent even called the process "causing trouble for the fellow workers". Underlying reason for this is that employees consider their own information gathering process a burden for both them and their co-workers. According to the five respondents, consistent DB documentation would allow them to work more independently and not interrupt their co-workers tasks. Further discussion of the DB documentations impacts were done by four participants who pointed out that by providing the customers the consistent DB documentation the workload of some manual tasks would be freed since the customers could check the DB structure themselves.

The second main category of effects was on the use of databases. Nine of the 14 participants stated that without DB documentation they make false interpretations of the database structure. This means that some of their work tasks had to be done in multiple small increments where the falsely interpreted structure is located and fixed. False interpretations cause issues, for example, in database operations, script writing and bug fixing. The respondents considered that the DB documentation would erase these false interpretations because the structure could be checked before starting the work task. Furthermore, eight respondents noted that they would have to do less queries to the database. Although the amount of queries done by the employees is probably not an issue for the performance of the database, knowledge of the DB structure would avoid some of the cases where queries become malformed or too massive. Similarly, eight respondents pointed out that their understanding of the whole database structure behind the CIS would be enhanced. This would be acquired by studying otherwise somewhat hidden knowledge of the connections inside the DB. In addition to broader understanding of the DB structure, four participants also stated that getting to know new parts of DB structure would become easier, enabling the employee to gain more understanding in shorter time.

6. Discussion

In this chapter the findings are analysed and compared with the knowledge from the prior literature. The research question is answered by first discussing the two sub questions. The two sub questions discussed the effects the DB documentation automation case had on the recognised stakeholder groups. These were the end-users of the CIS and the IS professionals of the case company. The sub question 1 regarding the effects on end-users' work tasks is answered in Section 6.1 and the sub question 2 regarding the effects on IS professionals' work tasks in Section 6.2. Main research question is answered in Section 6.3 by combining the discussions from Sections 6.1 and 6.2.

6.1 Effects on end-users' work tasks

The first of the two recognised stakeholder groups was the end-users of the CIS. To answer the question regarding the effects on customers' work tasks, a questionnaire was sent to 23 possible participants to gather their experiences and perceptions regarding the DB documentation's effects on their work tasks. Six customers took part in the questionnaire. The first sub question was:

RQ1: How does (automated) DB documentation affect the end-users' work tasks?

According to the findings, DB documentation supports end-users' work tasks and enhances their understanding of the DB structure. DB documentation is called stagnated when it is inaccurate, incomplete or out-of-date (Steinberger & Prakash, 2011). The customers considered that the prior DB documentation solution that was provided to them lacked some crucial information and it was outdated, thus stagnated. Leitheiser and March (1996) pointed out that the documentation can benefit the users by enhancing their knowledge of the system and its features. The new solution was considered to be consistent and up-to-date, thus supporting DB and system use of the customers.

Documentation allows the end-users to interpret more easily the DB structure including table and column names, metadata and relationships for their own needs. With stagnated documentation the customers have to make multiple assumptions regarding the DB structure. False interpretations take time and create frustration while conducting work tasks. While allowing easier interpretation of the DB structure, at the same time the customers' knowledge of the DB structure is enhanced and strengthened. This allows the customers to carry out more complex work tasks with the DB and the system with less errors while also lowering their workload.

System learnability and successfulness of work tasks in the system can be enhanced by DB documentation (Leitheiser & March, 1996) and the customers considered that the DB documentation supports them in their work. Customers' work tasks can be very different from another, thus requirements for the DB documentation for each position might be very different. Some might require documentation of only certain DB tables but in detail while some customers might require more broad insight to the system interfaces.

End-users' work tasks do not only include interpreting the DB for their system use, but for design tasks as well. The organizational customers of the case company use multiple other systems in addition to the CIS. The customers highlighted that multiple work tasks regarding interface design, interpretation or third-party integrations require consistent DB documentation to be carried out efficiently.

The case company's customers had read-only rights to the reporting DB. The DB is designed to provide the customers the means to conduct business reporting tasks and interpret the DB structure and metadata. The DB documentation supports end-user to more efficiently query the DB, since DB queries are a part of the customers' tool box as well. Without documentation the customers require lots of excess time to plan their queries, searching for right DB objects (Yasir, Swamy, & Reddy, 2012). The new automated DB documentation served the customers an easy way to interpret the primary and foreign key relationships in the DB. ER diagrams are considered easy to learn and simple to interpret (Gregersen & Jensen, 1999) and customers were also accustomed to the notation of ER diagrams generated by SchemaSpy 6.1. The case company's customers did consider ER diagrams simple to interpret and thought that the DB structure regarding the reporting DB views was comprehensively visualised to allow further analysis.

With consistent and up-to-date DB documentation the customers can carry out their statistical and organizational reporting more efficiently. Consistent DB documentation allows the representatives of the customer organizations to create their own augmented DB documentation for their own company's internal needs. For example, the augmented documentation can be used to define one segment of DB structure for certain employees or to support new specialised customer company's employees on their familiarization process.

6.2 Effects on IS professionals' work tasks

The second recognised stakeholder group was the IS professionals of the case company. To allow the analysis of the effects on work tasks, an invitation to participate in a questionnaire was sent in-house to all employees that worked with databases in some extent. 14 employees described how up-to-date, consistent DB documentation had affected or could affect their work tasks. The second sub question was:

RQ2: How does (automated) DB documentation affect the IS professionals' work tasks?

According to the findings, DB documentation affects the workload of IS professionals and additionally allows DB use to be more efficient. Employees stated that available, trustworthy documentation would allow them to conduct their work tasks faster while also providing possibility to plan work tasks beforehand. In addition, work tasks would become easier with the DB documentation's support to problem solving process. Consistent DB documentation enables the employee to work more independently regarding DB structures not common to them and lowers cognitive burden by erasing the need to memorize as many DB objects as possible. Trainees and new employees can also familiarize themselves with the DB structure and the CIS with less effort when DB documentation is provided to them. Furthermore, by providing documentation to the customers and end-users, employees considered that some of the contacts from customer side would be left out, thus lowering workload especially in customer support and customer projects.

DB use of the IS professionals would become more efficient and the amount of errors that result in faulty updates and changes would be lowered. Although DB documentation is commonly used by SW developers (Gregersen & Jensen, 1999) also other IS professionals from the case organization, for example, customer support and DevOps, benefit from it in their work tasks. The employees considered they would make less false interpretations regarding the DB structure if they had trustworthy documentation that they could check when conducting work tasks. In addition successful DB use would require less queries by erasing trial and error increments of the process. While using DB documentation to support work tasks, the employees would gain deeper understanding of the DB structure and the CIS's features in the long run. Familiarizing with the new schemas, tables and columns of the DB would become easier for the professionals because interpretation of the DB structure is supported by the documentation. This means that employees can more easily try new work tasks and develop their own knowledge regarding other parts of the system they have had limited knowledge of. This also benefits other professionals by allowing more liquid conduct of similar work tasks by multiple employees instead of certain ones with lots of prior knowledge about the specific DB structure.

The developers are more eager to document code than databases (Steinberger & Prakash, 2011) and thus the case company employees had become accustomed to working without consistent DB documentation. The available DB documentation was considered stagnated and the IS professionals of the case company stated that availability of consistent DB documentation would enhance their working processes. The availability of the consistent DB documentation was assessed important for employees working with the CIS and its DB, both in customer support and in different development and maintenance tasks.

Stagnated documentation can lead to wasted resources by forcing the professionals to make assumptions of the DB (Steinberger & Prakash, 2011). Work tasks that benefit from consistent DB documentation are not only the more complex ones but the ordinary work tasks as well. For example, locating sources of problems, bug fixing, scripting and tasks of SW design were considered to benefit from documentation. The benefit comes from less false interpretations and less time to understand problems, thus making tasks easier and faster and freeing time for other work tasks – alleviating productivity and efficiency of individual employees.

Gregersen and Jensen (1999) pointed out that ER modelling is used not only for design tasks of a DB but for analysis tasks as well. Interpreting a DB structure is in itself analysis of the DB. According to Chen (1976) the ER model allows the user to understand meaning of DB objects. IS professionals considered ER diagrams a comprehensible notation for DB documentation. ER diagrams were familiar to the employees from their studies, from their work experience of SW design tasks or from tools they have used before, similar to Gregersen's and Jensen's (1999) remark regarding the popularity of ER diagrams over two decades ago. Notation of the DB documentation created with SchemaSpy 6.1 was considered comprehensible and easy-to-use.

6.3 Effects of automated database documentation

The findings from both end-users and IS professionals provide overall insight into how DB documentation affects the work tasks of both stakeholder groups. The main research question was:

RQ: How does (automated) DB documentation affect the DB users' work tasks?

Consistent and up-to-date DB documentation affects the understanding of DB structure, therefore allowing more complex work to be carried out more efficiently and with less false interpretations for both end-users and IS professionals. Secondly, availability of DB documentation allows more precise planning and conduct of work tasks, therefore enhancing work quality by lowering workload and cognitive pressure of the DB users. If the user has opportunities to interpret the DB structure whenever needed, their knowledge of the DB and the system is higher than without documentation. Further use of documentation incrementally strengthens the individual's knowledge of the DB structure, thus enabling new work tasks and safer and easier conduct of the prior tasks. With the aid of documentation or the previous knowledge the user has, the individual can plan their work in greater detail and conduct tasks that would otherwise require support from other stakeholders, co-workers or service provider. The cumulated knowledge of the user, the availability of the DB documentation for further analysis and possibilities in planning work tasks form a circle that strengthens itself. All aspects influence the rate of benefits from other aspects. In figure 6, the circle of DB structure knowledge and its effects on work tasks are visualized regarding any DB user.

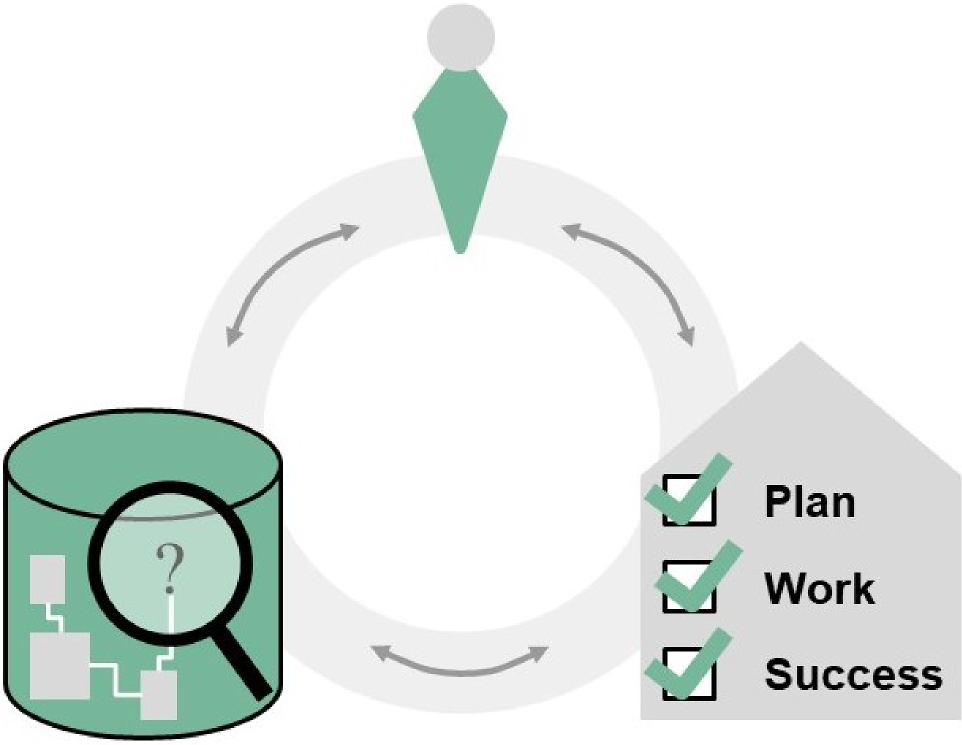

Figure 6. Knowledge of DB structure affects the employee's work and DB knowledge by allowing planning, lowering error-proneness and enhancing successfulness of work tasks

The work tasks of end-users of the system and the developers and other IS professionals of the system provider differ greatly. Yet, both stakeholder groups require DB documentation for similar reasons. DB documentation provides the stakeholders a possibility to easily interpret DB structure, including tables, columns, metadata and primary and foreign key relationships. Possibility to analyse the structure from diagrams which follow a notation that is common to the interpreter, for example, ER diagrams benefit the DB user's knowledge of the DB and thus the knowledge of the functionalities of the system. The gathered knowledge further cumulates while conducting work tasks and analysing the DB structure more deeply.

The customers use the system and require the DB documentation to support their system understanding, ensure compatibility with their other systems and develop their business further. Providing the customers with DB documentation that is designed to fulfil their needs can enhance customer satisfaction. The customers appreciate transparency from the service provider’s side. Furthermore, supporting the customers to develop their interfaces rather require is always better for the customer organizations than service provider making it harder by not giving the required documentation.

IS professionals that develop and maintain the systems require the documentation to familiarize themselves with DB structure behind it. This allows them to plan their work tasks in better detail, thus erasing excess interpretation time otherwise required. Although the use of DB documentation affects directly an individual employee, the effect can be considered also collectively. If consistent, up-to-date DB documentation including ER diagrams and metadata overviews are provided for the employees, the workload of multiple employees is lowered due to the saved time, the capability to work independently and the courage to try more complex and critical work tasks.

7. Conclusion

Automatically generated consistent and up-to-date DB documentation was considered important for both end-users' and IS professionals' work tasks. DB documentation enhances the DB users' knowledge of the DB structure and the system functionalities. This allows the DB users to avoid excess false interpretations of DB structure, which lowers their workload and cognitive burden. DB documentation allows the end-users of a system to interpret and design interfaces and integration between their other other systems more effortlessly. IS professionals working in different SW development and maintenance tasks benefit from DB documentation because it eases the independent problem solving process of the employees. Additionally, IS professionals can feel more courageous to try work tasks not previously familiar to them with the aid of documentation. Utilising DB documentation to support the employees in their work tasks also saves them from conducting error-prone changes or updates to the system or DB too often.

The results of this analysis indicate that DB documentation is an efficient support for both IS professionals' and end-users' work tasks. The findings highlight the importance of DB documentation to the individual DB users and thus for the whole organizations. The positive effects found in the study suggest that the DB documentation is maintained properly and provided for the users. This eases the workload of stakeholders and allows them to understand the DB better. Although DB documentation is generated in the development process of a system, the documentation tends to stagnated through the multiple changes to the DB during the development cycles. DBRE enables generation of consistent and up-to-date documentation in case of incomplete documentation. To allow the stakeholder groups to get the support to their work tasks from DB documentation, DBRE can be a cost-effective way to update the documentation.

This study analysed the effects of DB documentation in one case company. The company was a Finnish IS provider, providing a CIS for their customer companies. Therefore the findings of the study describe effects of DB documentation only in the case and the findings cannot be directly generalized to concern other cases. Precise impacts the DB documentation depend on the extent of the system, the expertise of the customers and the teams of the case organization, including which tasks each employee conducts. The case DB was a reporting DB, to which the customer have read-only rights. If the stakeholder groups have different rights to the DB the results might vary. Additionally, the sample of customers was considered small and additional responses could have brought up more detailed picture of the work tasks the customers carry out using DB documentation.

Future research could study similar DBRE cases to allow comparative analysis of the results. This study did not primarily discuss the effects on system learnability and use when the end-users are provided DB documentation. DB documentation effects on system use could be studied in more detail. Future research could also try to seize, if there is a more strict division of different DB documentation users and what work tasks they use DB documentation for and what content it must contain to allow their work tasks. ER diagrams and metadata overviews generated with SchemaSpy 6.1 were used in this case as the DB documentation. Whether DB users would favour some other notation and whether organizations have strict guidelines to DB documentation notation was not discussed in this study.

References

Armstrong, W. B. (2009). Planning your Survey, A Brief Primer on Strategies and Approaches. *Student Research & Information*, University of California-San Diego. Retrieved from http://studentresearch. ucsd. edu/_files/assessment/workshops/2009_Planning YourSurvey. pdf.

Bainbridge, L. (1983). Ironies of automation. In *Analysis, design and evaluation of man–machine systems* (pp. 129-135). Pergamon.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS quarterly*, 369-386.

Bell, E., Bryman, A., & Harley, B. (2018). *Business research methods*. Oxford university press.

Blaha, M. (1998, October). On reverse engineering of vendor databases. In *Proceedings Fifth Working Conference on Reverse Engineering* (Cat. No. 98TB100261) (pp. 183-190). IEEE.

Brannen, J. (Ed.). (2017). *Mixing methods: Qualitative and quantitative research*. Routledge.

Brewer, E. (2012). CAP twelve years later: How the" rules" have changed. *Computer*, 45(2), 23-29.

Chen, P. P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1, 1 (March 1976), 9–36. DOI:https://doi-org.pc124152.oulu.fi:9443/10.1145/320434.320440

Chiang, R. H. L., & Barron, T. M. (1995, June). Quality issues in database reverse engineering: an overview. In *Proceedings for Operating Research and the Management Sciences* (pp. 185-189). IEEE.

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 13 (6): 377–387. doi:10.1145/362384.362685

Codd, E. F. (1979). Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems*, 4, 4 (December 1979), pp. 397–434. DOI:https://doi-org.pc124152.oulu.fi:9443/10.1145/320107.320109

Creswell, J. W., & Creswell, J. D. (2017). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications.

De Vaus, D. (2002). *Surveys in Social Research* (5th eds). University College London Press, London.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS quarterly*, 597-636.

Edmondson, D. R. (2005, April). *Likert scales: A history*. In Proceedings of the 12th conference on historical analysis and research in marketing (CHARM) (pp. 127-133).

Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

Ghemawat, S., Gobioff, H., & Leung, S. T. (2003, October). The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles* (pp. 29-43).

Graefe, G. (1993). Query evaluation techniques for large databases. *ACM Computing Surveys (CSUR),* 25(2), 73-169.

Gregersen, H., & Jensen, C. S. (1999). Temporal Entity-Relationship models-a survey. *IEEE Transactions on knowledge and data engineering*, 11(3), 464-497.

Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM computing surveys (CSUR)*, 15(4), 287-317.

Hainaut, J. L. (1991, November). Database Reverse Engineering: Models, Techniques, and Strategies. In *ER* (pp. 729-741).

Halpin, T., & Morgan, T. (2010*). Information modeling and relational databases*. Morgan Kaufmann.

Henning, K., & Kutscha, S. (1994). *Informatik im Maschinenbau*. Berlin etc: Springer.

Hodder, I. (1994). *The interpretation of documents and material culture*. Sage biographical research, 1.

Janssen, C. P., Donker, S. F., Brumby, D. P., & Kun, A. L. (2019). History and future of human-automation interaction. *International Journal of Human-Computer Studies*, 131, pp. 99-107.

Java SE Technologies. (n.d.) Retrieved on April 16, 2020, from: https://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html

Kemppainen, S. (2020). *The Impacts of Automating Manual Processes in Software Maintenance: A Case Study* (Unpublished master's thesis). University of Oulu.

Kracauer, S. (1952*). The challenge of qualitative content analysis*. Public opinion quarterly, 631-642.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.

Krishnan, G., & Ravindran, V. (2017). IT service management automation and its impact to IT industry. *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)* (pp. 1-4). IEEE.

Leavitt, N. (2010). Will NoSQL databases live up to their promise?. *Computer*, 43(2), 12-14.

Leitheiser, R. L., & March, S. T. (1996). The influence of database structure representation on database system learning and use. *Journal of Management Information Systems*, 12(4), pp. 187-213.

Merriam-Webster. (n.d.). Metadata. In *Merriam-Webster.com dictionary*. Retrieved April 6, 2020, from https://www.merriam-webster.com/dictionary/metadata

Mfourga, N. (1997, October). Extracting entity-relationship schemas from relational databases: a form-driven approach. In *Proceedings of the Fourth Working Conference on Reverse Engineering* (pp. 184-193). IEEE.

Myers, M. D. (2019). *Qualitative research in business and management*. Sage Publications Limited.

MySQL Workbench. (n.d.) Retrieved on April 16, 2020, from: https://www.mysql.com/products/workbench/

Ndefo, N. (2017, September). Formalization of Database Reverse Engineering. In *European Conference on Advances in Databases and Information Systems* (pp. 410-421). Springer, Cham.

Newman, I., Benz, C. R., & Ridenour, C. S. (1998). *Qualitative-quantitative research methodology: Exploring the interactive continuum*. SIU Press.

Pritchett, D. (2008). Base: An acid alternative. *Queue*, 6(3), pp. 48-55.

Purchase, H. C., Colpoys, L., McGill, M., Carrington, D., & Britton, C. (2001, December). UML class diagram syntax: an empirical study of comprehension. In *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation*, 9, pp. 113-120. Australian Computer Society, Inc.

Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation surprises. *Handbook of human factors and ergonomics*, 2, 1926-1943.

SchemaCrawler. (n.d.). Retrieved on April 16, 2020, from: https://www.schemacrawler.com/

SchemaSpy. (n.d.). Retrieved on April 14, 2020, from: http://schemaspy.org/

Schmieder, P., Plimmer, B., & Dobbie, G. (2009). Sketching ER diagrams. In *Proceedings of the Tenth Australasian Conference on User Interfaces*, 93, pp. 53-60.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1), 1-47.

Singh, Y. K. (2006). *Fundamental of research methodology and statistics*. New Age International.

Stake, R. E. (1995). *The art of case study research*. Sage.

Steinberger, J., & Prakash, A. (2011). ModelDoc: Auto-generated, Auto-regenerated Wiki-Based Database Documentation.

Stonebraker, M. (2010). *SQL databases v. NoSQL databases*. Communications of the ACM, 53(4), 10-11.

System Catalogs. (n.d.) PostgreSQL: Documentation. Retrieved on April 6, 2020, from https://www.postgresql.org/docs/9.1/catalogs.html

Taylor, S. J., Bogdan, R., & DeVault, M. (2015). *Introduction to qualitative research methods: A guidebook and resource*. John Wiley & Sons.

The Data Dictionary. (n.d.) Oracle9i Database Concepts. Retrieved on April 6, 2020, from https://docs.oracle.com/cd/B10501_01/server.920/a96524/c05dicti.htm

Tschiersch, I., & Brandt, D. (1996). Chaos and Automation. *IFAC Proceedings Volumes*, 29(1), pp. 778-783.

Uhlgren, J. (2020). *Database Structure Extension – a Case Study using Design Science Research* (Unpublished master's thesis). University of Oulu.

Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, 52(1), 40-44.

Vukovic, D., Brdjanin, D., & Maric, S. (2017, November). A UML-based approach to reverse engineering of relational databases. In *2017 25th Telecommunication Forum (TELFOR)* (pp. 1-4). IEEE.

White, M. D., & Marsh, E. E. (2006). *Content analysis: A flexible methodology*. Library trends, 55(1), 22-45.

Yasir, A., Swamy, M. K., & Reddy, P. K. (2012). Exploiting schema and documentation for summarizing relational databases. *International Conference on Big Data Analytics* (pp. 77-90). Springer, Berlin, Heidelberg.

Yamashita, A. (2015). Experiences from performing software quality evaluations via combining benchmark-based metrics analysis, software visualization, and expert assessment. *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 421-428). IEEE. doi: 10.1109/ICSM.2015.7332493

Yee, K. P., Swearingen, K., Li, K., & Hearst, M. (2003, April). Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 401-408).

Yin, R. K. (1984). *Case Study Research, Design and Methods*. Sage Publications.

Yin, R. K. (2018). *Case study research and applications: design and methods*. Sage Publications.

Appendix A. Customer Questionnaire

1. Estimate your work experience in substance field (in years)

2. Estimate, how long you have worked with databases (For example, creating queries or interpreting the DB structure)

3. Describe how often you utilise the reporting database documentation

4. In which of your work tasks you use database documentation?

5. For which of your work tasks database documentation is a necessary or a useful support?

6. Has database documentation been useful when designing or interpreting interfaces? If yes, in which interface's design or interpretation?

7. Has database documentation enabled the use of some external tools, software or information systems? If yes, which external tools, software or information systems?

8. Prior reporting database and its documentation: Answer for the next statement (a-k) according to your own experience by choosing a number between 1 and 5. Do notice that the next statements discuss only the prior reporting solution. If you have not had the prior solution at your disposal move to the next question.

   (1: Strongly Disagree, 2: Disagree, 3: Neither Agree nor Disagree, 4: Agree, 5: Strongly Agree)

   a. Prior database documentation is up-to-date
   b. Prior database documentation includes all the information I need
   c. Prior database documentation includes incorrect information
   d. No erroneous interpretations happen when I utilise the database documentation
   e. The database is easy to use
   f. Getting to know the database structure is easy
   g. Documentation supports me in my database use
   h. Using the documentation I quickly find the database schema, table or view I want to
   i. It is easy to make database queries to the database
   j. Database documentation's diagrams are easy to interpret
   k. It is easy to see the connections between the tables / views from the database documentation's diagrams

9. New reporting database and its documentation: Answer for the next statement (a-k) according to your own experience by choosing a number between 1 and 5. Do notice that the next statements discuss only the new reporting solution. If you have not had the new solution at your disposal move to the next question.

(1: Strongly Disagree, 2: Disagree, 3: Neither Agree nor Disagree, 4: Agree, 5: Strongly Agree)

a. New database documentation is up-to-date
b. New database documentation includes all the information I need
c. New database documentation includes incorrect information
d. No erroneous interpretations happen when I utilise the database documentation
e. The database is easy to use
f. Getting to know the database structure is easy
g. Documentation supports me in my database use
h. Using the documentation I quickly find the database schema, table or view I want to
i. It is easy to make database queries to the database
j. Database documentation's diagrams are easy to interpret
k. It is easy to see the connections between the tables / views from the database documentation's diagrams

Appendix B. Personnel Questionnaire

1. Your team (Select the most suitable option)

    a. Customer Projects
    b. Customer Support
    c. DevOps
    d. Product Development
    e. Product Management
    f. Testing

2. Is database (DB) documentation familiar to you (ER diagrams and metadata overviews)? (Yes / No)

3. Have you used DB documentation as a supporting tool in your work? (Yes / No)

4. How important is up-to-date, consistent DB documentation for your work tasks and why?

5. How would up-to-date, consistent DB documentation affect your knowledge of the CIS's database structure?

6. Which of your work tasks would benefit from the DB documentation?

7. Do you think that up-to-date, consistent DB documentation could make some new work tasks possible? Which work tasks?

8. Would the DB documentation erase some of your work tasks? Which work tasks?

9. Would the DB documentation make some of your work tasks easier or harder? Which work tasks?

10. Would the DB documentation make some of your work tasks faster or slower? Which work tasks?

11. How often would you use DB documentation in your work tasks? Would this happen consistently?

12. Do you consider the provided example of generated Schemaspy documentation easy to understand and interpret? Why?

13. Would you use the DB documentation to support your work tasks? Why?