NOVA IMS Information Management School

MGI

Master Degree Program in

Information Management

Enterprise Data Warehouse based on Data Vault 2.0 sourced by a Data Lake: A Banking Industry Use Case

Inês de Oliveira Margarido

Dissertation

presented as partial requirement for obtaining the Master Degree Program in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Enterprise Data Warehouse based on Data Vault 2.0 sourced by a Data Lake: A Banking Industry Use Case

By

Inês de Oliveira Margarido

Master Thesis presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Knowledge Management and Business Intelligence.

Supervisor: José Henrique Pereira S. Mamede

Co-Supervisor*:* Vitor Manuel Duarte dos Santos

July 2023

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Inês de Oliveira Margarido

Lisboa, 4 de julho de 2023

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents for all the hard work and sacrifices they made so that I could get a good education, and for always reminding me that I am capable of achieving anything I set my mind to. Without them, none of this would have been possible.

A huge thank you to my Thesis Supervisor, Professor Henrique Mamede, for his constant availability and calming presence, especially throughout the most challenging phases of this work. During moments of uncertainty, it was nice to have someone knowledgeable and experienced by our side, with whom we could freely share our concerns.

To my closest friends and boyfriend for all the encouraging words, jokes, hugs, and for listening to me complain during this past year, a heartfelt thank you. Without you, this journey would have been much lonelier.

To everyone who impacted my academic path in any way, partners in group projects, housemates, all the friends I made during these past five years that passed by so quickly, thank you for all the memories and lessons that will stay with me forever.

ABSTRACT

The increasing volume, speed and heterogeneity of data has led companies to invest in Big Data technology, such as Data Lakes, which are central data repositories capable of ingesting structured and unstructured data in a large scale. However, Data Lakes are still not suitable for typical Business Intelligence use cases and analyses as data is stored without a defined schema, which is why most companies still want to keep their existing Enterprise Data Warehouses (EDW). Regarding architectures that combine a Data Lake and an EDW, there are no defined best practices for data storage, metadata management, and data loading from the Data Lake into the EDW, particularly into those based on Data Vault 2.0. There is also a need to understand the impact that a Delta Lake layer can have in optimizing said data loading. This dissertation aims to fill these gaps in the literature and provide the scientific community and banking industry with an efficient architecture for a Data Lake that sources an EDW, and an EDW model based on Data Vault 2.0.

KEYWORDS

Data Lake; Data Warehouse; Architecture; Data Vault; Delta Lake; Metadata

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

EDW	Enterprise Data Warehouse
BI	Business Intelligence
DW	Data Warehouse
SLR	Systematic Literature Review
GL	Grey Literature
DL	Data Lake
RQ	Research Question
PICOC	Population, Intervention, Comparison, Outcome, Context
QA	Quality Assessment
DBMS	Database Management System
UML	Unified Modeling Language
OWL	Web Ontology Language
RDF	Resource Description Framework
SQL	Standard Query Language
XML	Extensible Markup Language
JSON	JavaScript Object Notation
DV	Data Vault
ETL	Extract, Transform, Load
ELT	Extract, Load Transform
EL	Extract, Load
OLAP	Online Analytical Processing
PCF	Process Control Framework
ACID	Atomicity, Consistency, Isolation, Durability
ORC	Optimized Row Columnar
CDF	Change Data Feed
DML	Data Manipulation Language

DSR	Design Science Research
BK	Business Key
HK	Hash Key
PK	Primary Key
AK	Alternate Key
GDPR	General Data Protection Regulation

1 INTRODUCTION

The emergence of Big Data, marked by the increasing volume, velocity, and variety of data, created many opportunities for companies, however, it also meant that they needed to start adapting their data analytics platforms to ingest, store and process these large amounts of heterogeneous data (Nambiar & Mundra, 2022). The commonly used Data Warehouses are no longer enough to deal with all this data, as they are purpose-built for certain downstream Business Intelligence (BI) applications by applying a pre-defined schema to the data, don't support unstructured data and are costly to scale (Armbrust et al., 2021; Ravat & Zhao, 2019a). To solve these challenges, Data Lakes were created, which are low-cost big data repositories that ingest structured, semi-structured and unstructured data in raw format, providing direct access to it. However, this architecture leads to more data quality and governance problems, as the data does not conform to a specific schema, which is why the two-tier architecture is now dominant in the industry, where a small part of the data is loaded from the Data Lake into a Data Warehouse (DW) for important decision support and BI activities that require clean and secured data (Armbrust et al., 2021; Nambiar & Mundra, 2022).

The Data Vault 2.0 methodology is a quite recent approach to Data Warehouse modelling that is very adaptable to changes and based on business concepts. Although it has been gaining popularity, there is a need to understand how a Data Warehouse based on Data Vault 2.0 should work together with a Data Lake as a source and if there are any industry practices in place for this matter. The Delta Lake is also a recent technology that is being used to provide metadata management and transactional capabilities to the Data Lake, this way solving some of the issues mentioned above, regarding data quality and governance. We want to study if this concept can be fully or partially applied to two-tier architectures.

Given the importance that Data Lakes are gaining in the industry, the fact that companies are extending their existing DW architecture to include a Data Lake (Zaloni, 2016), and the emergence of new methodologies and concepts such as Data Vault 2.0 and Delta Lake, we want to study how these can work together in order to provide a scalable architecture. This research is conducted within the scope of an ongoing effort by a company in the banking industry to restructure their data architecture and integrate a Data Lake with an EDW based on Data Vault 2.0. It aims to find guidelines and best practices for implementing and managing a Data Lake in this environment and propose a conceptual model for the EDW based on Data Vault 2.0 using a small subset of data. The research question to be answered is: How can we model and integrate a Data Vault 2.0 EDW in a Data Lake architecture, using Delta Lake concepts? The associated objectives related to the research question are: (i) to present a Data Vault 2.0 model for the EDW, using the company's metadata, (ii) to identify best practices for the integration of the EDW into the Data Lake architecture, (iii) to discover if and how Delta Lake concepts can impact the efficiency of data loading from the Data Lake to the Enterprise Data Warehouse.

The Design Science Research Methodology will be used to conduct the research, following the six phases proposed by Peffers et al. (2007). To obtain the existent literature in an unbiased and repeatable way, a Systematic Literature Review will be performed, following the guidelines proposed by Kitchenham (2004). Because of the lack of formal literature on this topic, as shown by a previous literature review, a Grey Literature Review will also be performed, following the guidelines by Garousi et al. (2019) and Adams et al. (2017). For the Design and Development phase, an artifact will be produced in the form of a Data Lake conceptual architecture and a Data Vault 2.0 model, which will be

the focus of this work. The latter will then be validated in the Evaluation phase, through semi-structured interviews with domain experts.

This study collects and synthesizes all existent literature on Data Lake architectures sourcing a Data Warehouse based on Data Vault 2.0, and the impact Delta Lake can have in a data architecture of this type. Additionally, a Data Vault 2.0 model is proposed for the banking company's EDW, as well as the underlying Data Lake architecture in which the pilot Data Warehouse model will be implemented. Due to time and data access constraints, the model was not fully implemented or tested with real data. To evaluate the model and understand its contribution, interviews were conducted with experts from the company. The proposed model serves as a proof of concept of the usage of Data Vault 2.0 in the banking industry and the research provides the scientific community with a complete state-of-the-art regarding Data Lake architectures, Data Vault 2.0, and Delta Lake concepts.

The remainder of this document is structured as follows. In Section 2, the Theoretical Background for this research is presented, as well as a Systematic Literature Review, including both formal and grey literature. Section 3 describes how each phase of the Design Science Research methodology will be carried out. In Section 4, the problems and motivation for this research are presented, as well as the objectives that the solution aims to achieve. In Section 5, the process of designing and developing the artifact is thoroughly explained. Section 6 constitutes a demonstration of the artifact to showcase its adaptability in some use cases. In Section 7, the artifact is evaluated by analysing qualitative data collected in interviews with experts. The results and discussion are presented in Section 8. Finally, in Section 9, this research is concluded, presenting the main findings related to the defined research objectives, its contribution to the literature, the limitations of the research and the recommendations for future works.

2 LITERATURE REVIEW

2.1 INTRODUCTION

A systematic literature review (SLR) gathers, evaluates, and synthesizes all existent literature regarding a particular topic in a thorough, unbiased, and repeatable way, by following a previously defined review protocol. The guidelines by Kitchenham (2004) will be followed to perform an SLR for Formal Literature, which derive from existing guidelines in the medical field that were adapted for research in the Software Engineering field. Peer-reviewed studies are more rigorous, but they may also be behind on some topics that are more recent and very used in the industry but are still not widely researched by academics. Because of this and the evident lack of relevant literature found in the Formal Literature Review, a Grey Literature (GL) Review will also be performed, following the guidelines proposed by Garousi et al. (2019) and Adams et al. (2017). These guidelines allow for a review of grey literature that is as systematic and scientifically valuable as possible, considering its non-scientific nature.

Both reviews are divided into the three phases proposed in Kitchenham (2004): Planning, Conducting and Reporting. In the Planning phase, the review protocol is defined, including the research question(s), the search string, the PICOC (Population, Intervention, Comparison, Outcome, Context) of the research, sources to be searched, selection criteria, quality assessment checklist and data extraction form are defined. Once the protocol is thoroughly defined, the Conducting phase starts, which is when the search string is used in each source, results are collected and filtered using the previously defined selection criteria and their quality is assessed using the quality assessment questions (some of them can be disregarded if they do not surpass a defined cut-off score). After this process, the content of the remaining studies is fully read and analysed. Finally, the findings are reported by research question, this way presenting the answers in a logical and organized way.

The Data Vault 2.0 methodology will be used to design the conceptual model for the Data Warehouse, following the guidelines proposed by Linstedt & Olschimke (2016). Data Vault 2.0 emerged from the evolution of Data Vault 1.0, which only contemplated the modelling aspect. This new methodology not only provides modelling techniques, but also follows Scrum and Agile best practices, considers NoSQL (non-relational) databases and Big Data systems, and provides implementation guidelines. Because Data Vault 2.0 is platform-independent, it can integrate any type of database, structured or unstructured. In this case, the staging area for the Data Vault 2.0 EDW will be a Data Lake. While the EDW is modelled using Data Vault, containing historical raw data, the information marts that derive from it for end-user access follow the Kimball model, containing data products that are subject-oriented and prepared for reporting. Regarding Data Vault modelling, it is based on 3 core concepts/tables: Hubs, Links and Satellites. The Hubs represent the core business concepts, which vary from industry to industry and company to company, the Links are tables that represent a relationship between two or more Hubs, which is always a many-to-many relationship, and the Satellites contain the descriptive and historical information of the Hubs and/or Links.

2.2 Formal Literature Review

2.2.1 Research Background

The increasing volume, speed and heterogeneity of data has challenged the data extraction, storage, and processing capabilities traditional data management systems, which has led to companies adopting new solutions fit for Big Data (P. Sawadogo & Darmont, 2021). Financial institutions are no exception, as their data typically comes from hundreds of data sources (Sienkiewicz & Wrembel, 2021), but they are still lagging behind other sectors in terms of analytics tools and architectures (Pisoni et al., 2021). The Data Lake (DL) is an architecture that has been gaining popularity because it allows the storage, management, and analysis of huge amounts of data in any format, this way addressing the aforementioned issues. Although knowledge has evolved on this topic, most design approaches for Data Lakes provide few theoretical and implementation details (P. Sawadogo & Darmont, 2021).

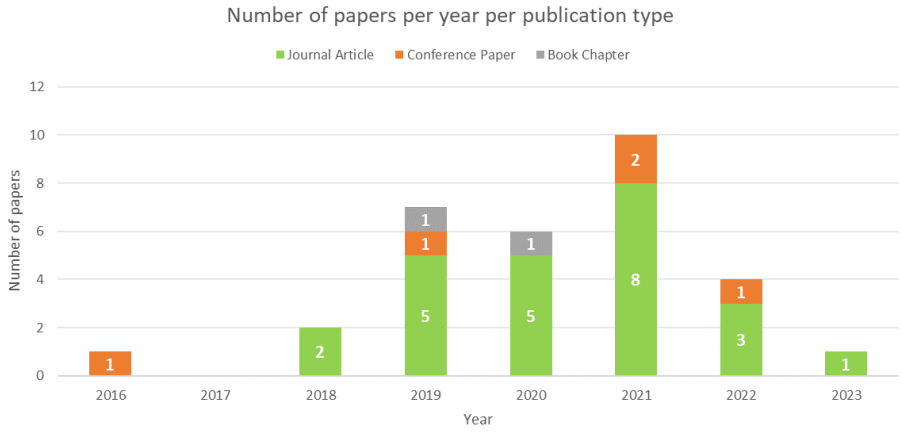

Figure 1 - Number of accepted papers per year and type

Data Lakes can easily become data swamps (B. Inmon, 2016), which can be prevented with efficient metadata management, but there is no generic system defined for that purpose (Ravat & Zhao, 2019b). Furthermore, Llave (2018) recognizes Data Lakes as a complement to Data Warehouses and mentions the lack of research on this architecture. The Data Vault methodology can be used to model a DL's structured data, but there is hardly any research on this topic, as stated by Giebler et al. (2019). The number of publications regarding the topics covered by this systematic review (Figure 1) hasn't been growing consistently and the years with more publications were 2019 and 2021. Most of the accepted studies are journal articles.

2.2.2 Planning

2.2.2.1 Research Questions

The aim of this systematic literature review is to provide the state-of-the-art regarding Data Lake architectures, more specifically in the case where the data will be loaded into an Enterprise Data Warehouse based on Data Vault 2.0. Furthermore, we are also interested in finding out the impact that Delta Lake concepts may have on this kind of architecture. To do that, and to fill the research gaps

mentioned in the Research Background, this study plans to answer the following research questions (RQ):

- RQ1: What are the best practices for storing data in a Data Lake?
- RQ2: What are the current best practices for storing metadata in the Data Lake?
- RQ3: How is Data Vault design methodology being used to model structured data of a Data Lake?
- RQ4: Are there any proposed practices for storing data in a Data Lake, which will be loaded onto a Data Warehouse based in Data Vault 2.0?
- RQ5: How can a Delta Lake layer impact a Data Lake architecture?

2.2.2.2 Data Sources and Search Strategy

The review is divided into three phases: Planning, Conducting and Reporting. The Planning phase and some parts of the Conducting phase, namely importing the studies and study selection, were done using the *parsif.al* tool.

In the Planning phase, the PICOC (Population, Intervention, Comparison, Outcome and Context), research questions, search string, sources, selection criteria, quality assessment checklist and data extraction form were defined. The PICOC is defined in Table 1.

Table 1 - PICOC definition

Population	Delta records coming from the source systems of the company to the Data Lake architecture
Intervention	Data Lake, Delta Lake, Data Warehouse, Data Vault 2.0, Metadata, Data Architecture
Comparison	Not applicable
Outcome	State-of-the-art knowledge about efficient data storage in Data Lake architectures; Known best practices to store data that will be loaded from a Data Lake into a Data Warehouse based in Data Vault 2.0
Context	Banking Industry, Data Management

An initial exploratory search in only three academic databases revealed a low number of results, so other sources were progressively added to obtain a significant number of results for our review. The sources that were searched to collect relevant literature were Scopus, ACM, EBSCO, IEEE, B-On, and Science@Direct. This mixture of aggregation services (EBSCO, B-On) with primary sources (Scopus, ACM, IEEE, Science) has been deliberately done to amplify and enlarge the search, since a preliminary search brought practically no results. The search string as well as the number of results per source is shown in Table 2.

Table 2 - Search string and number of results per source

Search String: ("Data Lake" OR "Data Lakehouse") AND "Data Warehouse" AND "Architecture" AND ("Data Vault" OR "Delta Lake" OR "Metadata")			
SOURCE	**FIELDS**	**NR OF RESULTS**	**OBSERVATIONS**
Scopus	All Fields	121	
ACM	All Fields	45	

EBSCO	All Fields	15	
IEEE	All Fields	2	
B-On	Title; Abstract; Keywords	7	Too many results using "All Fields" (>1300)
Science@Direct	All Fields	121	

2.2.2.3 Selection Criteria

First, the duplicate papers will be removed. After that, at least the Title and Abstract of the remaining papers will be read. In some cases, the Introduction, Conclusion, and a general overview of the full text can be needed to determine whether the study should be accepted or rejected, according to the defined criteria. The inclusion and exclusion criteria are defined in Table 3.

Table 3 - Inclusion and exclusion criteria

INCLUSION CRITERIA	**EXCLUSION CRITERIA**
Research on Data Lake architectures with a Delta Lake layer	Does not mention Data Lake/Lakehouse architectures
Research focused on Data Lake architectures and Data Vault Modelling	In the context of or refers to Data Lake/Lakehouse architectures, but does not discuss the actual architecture and data storage
Addresses a Data Lake architecture combined with a Data Warehouse	Addresses Data Lake architectures but not together with a Data Warehouse
Research on metadata management in a Data Lake/Lakehouse architecture	Addresses both Data Lakes and Data Warehouses but as separate approaches
	Full text not in English language
	Full text unavailable
	Published before 2016
	Not peer-reviewed
	Invalid imported reference

2.2.2.4 Quality Assessment

To assess the quality of the accepted papers, questions related to the content, methodology and novelty of the studies were adapted from Garousi et al. (2019). For each study, all the questions will be answered as “Yes”, “Partially” or “No”, with the corresponding values of 1, 0.5 and 0, respectively. The methodology and novelty questions have a weight of 0.5, unlike the content questions which have a weight of 1, because they are less relevant to identify the usefulness of a study. Then, the weighted sum of the values will be calculated for each study, as well as the normalized sum. The cut-off score is set to 1.5, being that studies with a QA (Quality Assessment) score lower than 1.5 will be removed from the analysis. The QA questions used are defined in Table 4.

Table 4 - Quality assessment checklist

CRITERIA	**QUESTIONS**	**WEIGHT**
Content	Does the study focus on how to efficiently store data in Data Lake architectures?	1
	Does the study discuss the storage and loading of the data from a Data Lake to an EDW?	1
	Does the study discuss the impact of a Delta Lake layer on the Data Lake architecture?	1

	Does the study discuss the Data Vault / Data Vault 2.0 methodology in the context of DLs or EDWs?	1
	Is metadata management in a Data Lake discussed?	1
Methodology	Was a comprehensive literature review performed?	0.5
Novelty	Does the study add something new? (Propose a framework, guidelines, model, etc.)	0.5
Cut-off Score: Weighted sum < 1.5		

2.2.3 Conducting

In the Conducting phase, studies are selected based on defined inclusion and exclusion criteria. Once the final papers are selected, they are extracted, their quality is assessed, and the information is synthesized.

After applying the search string to all the selected sources, 311 papers were collected and imported into the *parsif.al* tool. Then, duplicates were removed (26 papers), some automatically (using the "Find Duplicates" functionality) and others manually. For the remaining studies, the selection criteria were applied, which led to a final number of 39 selected papers. Because the final number of papers is small, a Grey Literature Review has been performed, as described in Section 2. After performing the Grey Literature Review, two of the accepted articles were included in this review as they were peer-reviewed scientific papers. After performing the quality assessment, a total of 31 papers remained for analysis (Figure 2).

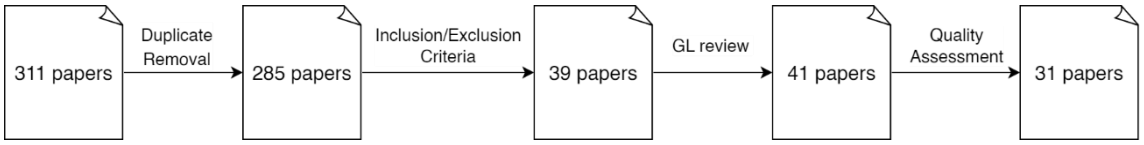

Figure 2 - Study selection process

2.2.3.1 Inclusion and exclusion criteria

In total, 246 from the 285 unique papers were excluded using the exclusion criteria. The remaining 39 papers were accepted because they complied to, at least, one of the inclusion criteria. To accept or reject the remaining papers after duplicates removal, at least the Title and Abstract were read. Some invalid imported references were also eliminated (6). The inclusion and exclusion criteria, as well as the number of articles accepted/rejected per criterion, are defined in Table 5.

Table 5 - Number of included and excluded papers by the criteria

#	INCLUSION CRITERIA	EXCLUSION CRITERIA	#
2	Addresses a Data Lake architecture with a Delta Lake layer	Does not mention Data Lake/Lakehouse architectures	174
3	Addresses Data Lake architectures and Data Vault Modelling	In the context of or refers to Data Lake/Lakehouse architectures, but does not discuss the actual architecture and data storage	23

13	Addresses a Data Lake architecture combined with a Data Warehouse	Addresses Data Lake architectures but not together with a Data Warehouse	27
21	Addresses metadata management in a Data Lake/Lakehouse architecture	Addresses both Data Lakes and Data Warehouses but as separate approaches	7
		Full text not in English language	1
		Full text unavailable	4
		Published before 2016	0
		Not peer-reviewed	4
		Invalid imported reference	6
39			246

Because the number of relevant papers revealed itself to be extremely low, papers as old as 2016 were accepted. Also, four books were excluded based on the criterion "Not peer-reviewed", which can be used in a future Grey Literature review. Information about the number of studies per journal can be found in Table 6.

Table 6 - Journal articles

PUBLICATION, PUBLISHER	NUMBER OF PUBLICATIONS
Communications in Computer and Information Science, Springer	5
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag	4
Procedia Computer Science, Elsevier	2
International Journal of Advanced Computer Science and Applications, Science, and Information Organization	2
Lecture Notes on Data Engineering and Communications Technologies, Springer International Publishing AG	2
CEUR Workshop Proceedings	2
Procedia CIRP, Elsevier	1
E3S Web of Conferences, EDP Sciences	1
Procedia Manufacturing	1
ACM International Conference Proceeding Series (ICPS), ACM	1
Data & Knowledge Engineering, Elsevier	1
Journal of Intelligent Information Systems, Springer	1
Journal of Big Data, SpringerOpen	1
Frontiers in Big Data, Frontiers Media S.A.	1
International Journal of Knowledge Management Studies, Inderscience Enterprises Ltd.	1
Proceedings of the VLDB Endowment	1
Applied System Innovation, MDPI AG	1
Baltic Journal of Modern Computing, University of Latvia – Institute of Mathematics and Informatics	1
Future Generation Computer Systems, Elsevier	1
Annals of DAAAM and Proceedings of the International DAAAM Symposium	1
Big Data and Cognitive Computing	1

Table 7 shows the conferences in which the accepted papers were published and the number of publications per conference.

Table 7 - Conferences

CONFERENCE	NUMBER OF PUBLICATIONS
2022 45th Jubilee International Convention on Information, Communication and Electronic Technology, MIPRO 2022	1
2021 17th International Conference on Network and Service Management: Smart Management for Future Networks and Services, CNSM 2021	1
ACM SIGMOD International Conference on Management of Data	1
2021 44th International Convention on Information, Communication and Electronic Technology, MIPRO 2021	1
2021 IEEE International Conference on Big Data, Big Data 2021	1
21st International Conference on Enterprise Information Systems, ICEIS 2019	1
SMC '19: The Second Conference of the Moroccan Classification Society Kenitra Morocco	1

Table 8 includes information about the book chapters included in this review, as well as their publisher.

Table 8 - Book chapters

BOOK, CHAPTER, PUBLISHER	NUMBER OF PUBLICATIONS
Data Lakes, Chapter 5, Wiley	1
Data Architecture, Chapter 2.1, Elsevier	1

The authors mentioned in more than one paper are presented in Table 9.

Table 9 - Authors with more than one citation in the accepted papers

AUTHORS	NUMBER OF CITATIONS
Sawadogo, P.N.	5
Darmont, J.	5
Ravat, F.	3
Zhao, Y.	3
Oukhouya, L.	2
El Haddadi, A.	2
Er-raha, B.	2
Asri, H.	2
Hai, R.	2
Quix, C.	2
Hlupic, T.	2
Orescanin, D.	2
Solodovnikova, D.	2
Niedrite, L.	2
Giebler, C.	2
Schwarz, H.	2
Gröger, C.	2
Mitschang, B.	2

2.2.3.2 Quality Assessment

The 39 accepted papers plus 2 that were included from the Grey Literature review (total of 41) were carefully assessed using the QA questions. In the end, 10 papers were excluded in the Quality Assessment phase, as they had a score lower than 1.5, leaving only 31 papers for analysis.

2.2.4 Reporting

2.2.4.1 RQ1: Data Lake Architectures

There are multiple approaches in the literature for Data Lake architectures. The most used and reviewed ones are the Data Pond architecture, the Zaloni Zone architecture and the Data Lakehouse. Some papers also review other architectures that aren't used as much now, as they evolved to the Data Pond and Zone architectures, e.g., mono-zone, two-layered and multi-layered architectures (Megdiche et al., 2020; Hlupic et al., 2022; Ravat & Zhao, 2019a).

The Data Pond architecture, as proposed initially by (Inmon, 2016) is considered to be a hybrid architecture (P. Sawadogo & Darmont, 2021), as it divides the data lake into ponds, based on function but also maturity of the data (raw, analog, application, textual and archive data ponds). The biggest drawback of this architecture is the fact that after being loaded to the designated data pond, raw data is lost. Also, the division in ponds for different types of data, including the archival pond for rarely used data, does not ensure the availability of all raw data at any time (Megdiche et al., 2020).

The zone architecture overcomes this issue by ingesting and storing data temporarily in a transient landing zone, which is then stored permanently in the raw zone in its original form. Then, data is integrated and structured in the refined zone. The trusted zone stores all the cleansed data, applies governance rules and is normally the source for the Data Warehouse (Oukhouya et al., 2023). Some approaches also consider a Sandbox or Access zone for exploratory analysis. It is important to note that the designations of each zone may not always be the ones mentioned, but the main idea of having a zone for raw storage, zone(s) for processing and a zone for data access and analysis persist in all approaches. A 5-zone approach is also presented by Nambiar & Mundra (2022) which, apart from the raw, cleansed and application data zones, also has two optional zones: standardized, where the appropriate format for data cleansing is selected, and sandbox, which is dedicated to exploratory analysis performed by data scientists and analysts on the raw data.

The Data Vault based zone architecture, as described in Hlupic et al. (2022), is similar to the Zaloni zone architecture, but the harmonized and distilled zones (the ones that store structured data; counterparts of the trusted and refined zones of the zone architecture) are modelled using Data Vault. Unlike the zone architecture, the data storage layer should have defined raw data storage models and a metadata repository.

Priebe et al. (2021) mentions three types of Big Data architectures. The Logical Data Warehouse, where a Data Lake and a Data Warehouse (modelled using Data Vault) are used for storage and virtualization layer is used for data integration. The other two architectures – Data Fabric and Data Mesh – are more generic, not restricted to architecture archetypes such as the Data Lake or the Data Warehouse and incorporate ideas from the Lambda and Kappa architectures. Data Fabric focuses mainly on metadata,

which consists of a Data Catalog and a Knowledge Graph with semantically linked metadata. Data Mesh focuses on business domains and data products.

The Lambda architecture (Solodovnikova & Niedrite, 2020b; Wieder & Nolte, 2022) is proposed as a way to enhance the capabilities of Data Lakes by processing streaming data in real-time instead of only performing batch-processing of the ingested data with time delay. This architecture implements two parallel processing streams – speed layer (real-time) and batch layer – and a serving layer, that combines the output of the two streams and provides a batch view of the data.

Solodovnikova & Niedrite (2020b) also mentions other types of architectures for Big Data that are not based on Data Lakes: Virtual and Polystore architectures. Table 10 identifies the main types of architectures and the studies that explore them.

Table 10 - Types of Big Data architectures

ARCHITECTURES	**SOURCES**
Zone architecture	Megdiche et al. (2020); Hlupic et al. (2022); Ravat & Zhao (2019a); Pisoni et al. (2021); Solodovnikova & Niedrite (2020b); P. Sawadogo & Darmont (2021); Wieder & Nolte (2022); Oukhouya et al. (2023); Nambiar & Mundra (2022)
Pond architecture	Hlupic et al. (2022); W. H. Inmon et al. (2019); P. Sawadogo & Darmont (2021)
Data Vault based zone architecture	Hlupic et al. (2022)
Data Lakehouse	Orescanin & Hlupic (2021); Wieder & Nolte (2022)
Lambda architecture	Solodovnikova & Niedrite (2020b); Wieder & Nolte (2022)
Logical Data Warehouse	Priebe et al. (2021)
Data Fabric	Priebe et al. (2021)
Data Mesh	Priebe et al. (2021)
Virtual architecture	Solodovnikova & Niedrite (2020b)
Polystore architecture	Solodovnikova & Niedrite (2020b)

In terms of just Data Lake architectures, disregarding the Big Data architectures that follow a different paradigm and/or do not have concrete implementation approaches, we can compare them based on the most discussed features in the literature. In Table 11, we analyse the architectures according to: (i) how they organize the data (by degree of processing/refinement or by type), (ii) if they store raw data persistently, (iii) if they have a transient loading area, (iv) a cold storage area and/or (v) a data discovery area, (vi) if they state the necessity of a metadata repository for all areas/zones, (vii) if they define a storage model for the raw data, (viii) if the zones are independent between them, (ix) if the data access has to be through a virtualization layer (because different types of data are stored in different places), and (x) if all data is stored in the same environment. We can conclude that, according to these analysed features, the Data Vault-based Zone architecture is the most complete out of the four, as it is an adaptation of the Zone architecture, possessing its benefits, while additionally having more control over data models, especially for structured data (modelled using Data Vault), and metadata.

Table 11 - Data Lake architectures and respective features

Architecture / Features	Zone Architecture	Pond Architecture	Data Vault-based Zone architecture	Data Lakehouse
Division by degree of processing	X		X	X
Division by type		X		
Raw data persistence	X		X	X
Transient landing area	X		X	X
Cold storage area		X		
Data discovery area	X		X	
Metadata repository available for all areas/zones			X	
Requires definition of storage model for raw data			X	
Zones are independent between them			X	
Data access through a virtualization layer		X		X
All data is stored in the same environment	X	X		

2.2.4.2 RQ2: Metadata in Data Lakes

2.2.4.2.1 Metadata classification

In terms of metadata classification, the one that is most commonly adopted among the reviewed studies is the division of metadata in three categories: inter-object metadata, intra-object metadata, and global metadata (P. N. Sawadogo et al., 2021; P. N. Sawadogo, Scholly, et al., 2019; P. N. Sawadogo, Kibata, et al., 2019; P. Sawadogo & Darmont, 2021; P. N. Sawadogo, 2019), where global metadata is not associated with a particular object, but concerns the whole data lake. In these studies, the three categories of metadata are divided into subtypes. Intra-object metadata can contain metadata properties, versions, and representations (updates and refining operations) or a previsualization (summary) of a certain object. Inter-object metadata can be object groupings, similarity links and parenthood links (save data lineage). Global metadata can be semantic resources (e.g., ontologies), indexes and logs.

Other studies adopt the classification into only inter-metadata and intra-metadata (Oukhouya Lamya et al., 2021; Francia et al., 2021; Ravat & Zhao, 2019b; Megdiche et al., 2020). According to Megdiche et al. (2020) and Ravat & Zhao (2019b), intra-metadata can be data characteristics, definitional metadata (semantic and schematic), navigational metadata (location of data), lineage metadata (data lifecycle), access metadata, quality metadata and security metadata. Inter-metadata can correspond to dataset containment, partial overlap, provenance (dataset produced from another dataset), logical clusters (e.g., different versions of the same dataset) and content similarity.

The rest of the studies don't specify the classification or have a non-generic one adapted to the underlying architecture. However, the majority considers that, at least, data item-specific metadata and relationships between metadata should be modelled. In terms of the most granular data item with

metadata information, some studies consider it to be a “dataset” and others replace it by “object”. The adoption of one over the other typically has to do with how generic the model is, because the object is normally used to encompass all types of data (structured or not). Eichler et al. (2021) considers that categorizing metadata does not help in any way in building a generic metadata model.

Table 12 summarizes the two main metadata classifications and the studies that support each of them.

Table 12 - Metadata classifications

METADATA TYPES	SOURCES
Intra-metadata; Inter-metadata	Oukhouya Lamya et al. (2021); Francia et al. (2021); Ravat & Zhao (2019b); Megdiche et al. (2020)
Intra-metadata; Inter-metadata; Global metadata	P. N. Sawadogo et al. (2021); P. N. Sawadogo, Scholly, et al. (2019); P. N. Sawadogo, Kibata, et al. (2019); P. Sawadogo & Darmont (2021); P. N. Sawadogo (2019)

2.2.4.2.2 Metadata representation and modelling

In terms of metadata representation and modelling, some studies consider a graph-based approach to represent metadata and links between data and metadata (Ziegler et al., 2020; P. N. Sawadogo et al., 2021; Francia et al., 2021; P. N. Sawadogo, Scholly, et al., 2019; Eichler et al., 2021; Alrehamy & Walker, 2018; P. N. Sawadogo, 2019) and others use a relational DBMS (Oukhouya Lamya et al., 2021; Solodovnikova & Niedrite, 2020a). One study uses a graph DBMS to store logical links between metadata, but also uses a relational DBMS and a filesystem for other data (P. N. Sawadogo, Kibata, et al., 2019).

Some papers recognize Data Vault as a design methodology for metadata models, stating that it supports schema and data source evolution, as evolving the Data Vault over time mainly implies adding satellites, links or hubs, and obsolete entities can be identified by their timestamp (Nogueira et al., 2018; P. Sawadogo & Darmont, 2021; Wieder & Nolte, 2022). Nogueira et al. (2018) suggests four main hubs – *hub_title, hub_location, hub_date* and *hub_category* – with many satellites, each specific to a source, and a *link_document* that allows association between all hubs. P. N. Sawadogo, Kibata, et al. (2019) also mentions Data Vault modelling for metadata representation, but combined with a graph representation, to manage changing number and form of inter-dataset metadata.

Both Megdiche et al. (2020) and Ravat & Zhao (2019b) implement a metadata model in a graph DBMS and a relational DBMS. They conclude that graph databases are more flexible and scalable, which is good to support schema evolution, however, they don’t have a standardized query language, which increases costs, and some don’t provide security mechanisms yet. Both implementations have pros and cons, so the final decision should be made considering the environment of the metadata management system. Table 13 synthesizes the metadata modelling approaches, indicating the studies that mention them.

Table 13 - Metadata modelling approaches

METADATA MODELING APPROACHES	SOURCES
Graph	Ziegler et al. (2020); P. N. Sawadogo et al. (2021); Francia et al. (2021); P. N. Sawadogo, Scholly, et al. (2019); Eichler et al. (2021); Alrehamy & Walker (2018); P. N. Sawadogo (2019)
Relational	Oukhouya Lamya et al. (2021); Solodovnikova & Niedrite (2020a)
Data Vault	Nogueira et al. (2018); P. Sawadogo & Darmont (2021); Wieder & Nolte (2022)
Graph + Relational	Megdiche et al. (2020); Ravat & Zhao (2019b)
Data Vault + Graph	P. N. Sawadogo, Kibata, et al. (2019)

2.2.4.2.3 Metadata key features

Many studies reflect on the key functionalities that a metadata management system should have. A recent study by Oukhouya Lamya et al. (2021), which proposes a metadata system for a Data Lake to be used as a source for a DW, defines nine key features of metadata systems, which I believe to be the most complete set: (i) semantic enrichment, (ii) data polymorphism, (iii) data versioning, (iv) usage tracking, (v) categorization, (vi) similarity links, (vii) metadata properties, (viii) multiple granularity levels and (ix) schema evolution. Apart from these features, Solodovnikova & Niedrite (2020a) and Wieder & Nolte (2022) mention data provenance as being a key feature of metadata systems to track datasets across transformations and P. Sawadogo & Darmont (2021) mentions data indexing as an important feature of metadata for efficient data retrieval from the DL (e.g., through keywords). Because data provenance is usually embedded in the metadata properties, only the latter feature will be considered. Table 14 summarizes the features supported by the metadata management models presented by each study.

Table 14 - Features supported by the metadata models

Features / Studies	Semantic enrichment	Data polymorphism	Data versioning	Usage tracking	Categorization	Similarity links	Metadata properties	Multiple granularity levels	Schema evolution	Data indexing
(Alrehamy & Walker, 2018)	X					X	X		X	
(Eichler et al., 2021)	X	X		X	X	X	X	X		X
(Francia et al., 2021)	X	X	X	X	X	X	X	X	X	
(Hai et al., 2016)	X						X	X	X	X
(Megdiche et al., 2020)	X		X	X	X	X	X			X
(Nogueira et al., 2018)	X		X				X		X	X
(P. N. Sawadogo et al., 2021)	X	X	X	X	X	X	X			X

(P. N. Sawadogo, 2019)	X	X	X	X	X	X	X			
(P. N. Sawadogo, Kibata, et al., 2019)	X	X			X	X	X		X	X
(P. N. Sawadogo, Scholly, et al., 2019)	X	X	X	X	X	X	X			X
(Ravat & Zhao, 2019b)	X	X	X	X	X	X	X			X
(Solodovnikova & Niedrite, 2020a)	X	X					X	X	X	
(Ziegler et al., 2020)	X					X	X			X
Oukhouya Lamya et al. (2021)	X	X	X	X	X	X	X	X	X	X

From the table above, we can see that the most complete model is the one proposed by Oukhouya Lamya et al. (2021), as it supports each of the analysed features.

2.2.4.2.4 Ontologies

Some studies consider the use of ontologies to represent metadata, as they allow the formal modelling of knowledge and use semantic search technology to uncover meaningful information from data (Holom et al., 2020).

In Oukhouya Lamya et al. (2021), the UML formalism is used to design the conceptual metadata model but, because it is semantically ambiguous, it is later transformed into an OWL ontology, using transformation rules for each UML element. *SemLinker*, the ontology-based integration system proposed by Alrehamy & Walker (2018), includes a global schema layer, which is also modelled as an OWL ontology. Local schemas, containing physical schemas of the data sources and semantics of their data, are stored as RDF graphs, and mapped to the corresponding concepts in the global schema.

Holom et al. (2020) also employs an ontology-based approach, but based on RDF and Spark SQL, which has some of the benefits of traditional ontologies, while allowing the team to keep the current database infrastructure, not requiring high knowledge in semantic technology, and allowing queries in standard SQL.

2.2.4.2.5 Data formats for metadata

Metadata can be associated with data in different ways: (i) embedded in the resource, (ii) stored in a file linked to a resource, (iii) stored in an individual repository, independent from the resource (Megdiche et al., 2020; P. N. Sawadogo, Kibata, et al., 2019).

Most studies do not specify the data formats used to store metadata. Some of the most generic models consider different data formats for metadata depending on how structured the data is: relational tables for structured data and XML, RDF, JSON, etc. for semi-structured data (Solodovnikova & Niedrite, 2020a; Hai et al., 2016) . The JSON file format is commonly used by the graph and data vault implementations, except for P. N. Sawadogo, Kibata, et al. (2019), which uses XML manifests to store metadata in its Graph + Data Vault model implementation.

Holom et al. (2020) uses two intermediate formats for metadata storage: Avro and Salad. Avro is used to define data schemas in JSON and represent syntactic metadata. Because Avro does not support the definition of direct relationships, Salad schema language is used to describe linked data through annotations and define rules for pre-processing, structural validation, and link checking. The integration of these formats with Spark SQL, allows data to be connected and have context, while being able to be queried using standard SQL.

In Alrehamy & Walker (2018), local schemas (physical schema and data semantics of data from a certain data source) are extracted from data automatically and stored in RDF graphs. Each local schema is then mapped to the semantically corresponding concept in the global schema, which is modelled as a global ontology using OWL.

2.2.4.2.6 Underlying architecture

Most of the analysed metadata management systems are implemented in a Data Lake based in zones (zone architecture). However, the metadata system proposed by (P. N. Sawadogo, Kibata, et al., 2019) for textual documents (only unstructured) data is implemented in a DL with a pond architecture. We can conclude that the most used underlying architecture for metadata systems encompassing all types of data is the zone architecture of Data Lakes.

2.2.4.3 RQ3: Data Vault Modelling

As mentioned by Giebler et al. (2019), there are currently no insights or practical experiences on how to use Data Vault in the context of Data Lakes.

There are only three papers that mention Data Vault modelling in the context of Data Lake architectures or architectures that include a Data Lake. Hlupic et al. (2022) describes a Data Vault based zone architecture for Data Lakes, where each zone holds data with different degrees of processing and in a format specific to their intended use, similarly to the Zaloni zoned architecture. The difference is that this architecture has more zones – landing zone, raw zone, harmonized zone, distilled zone, delivery zone and explorative zone –, and two of them are modelled using Data Vault (harmonized and distilled zones). While the harmonized zone contains raw structured data, the distilled zone contains

the same data but with business logic applied over it. Priebe et al. (2021) analyses Big Data Architectures, with the Logical Data Warehouse being one of those. This architecture integrates many existing approaches, such as a Data Lake and Data Warehouse for storage, as well as Data Virtualization for integration and interoperability. Data Vault is considered as a modelling approach for the Data Warehouse of this architecture. However, none of the aforementioned papers provide guidance on how to adapt data storage in the Data Lake, knowing that the structured data will then be modelled using Data Vault, or even discuss the benefits of using Data Vault instead of, for example, dimensional modelling. Giebler et al. (2019) considers DV (Data Vault) for modelling the structured data of a Data Lake as an attempt to solve data quality and integration issues that derive from the architecture's lack of structure. This paper recognizes three key characteristics of Data Vault – flexibility (links always represent many-to-many relationships), loading efficiency (tables can be loaded in parallel in DV 2.0) and auditability –, and presents real use cases where some practical issues and solutions for modelling are proposed. The remaining papers that mention Data Vault do it in the context of metadata modelling, which was already mentioned in the previous RQ. Table 15 shows the context in which Data Vault modelling is used in the studies that mention it.

Table 15 - Context in which Data Vault modelling is used

CONTEXT IN WHICH DV MODELING IS USED	SOURCES
To model structured data of the Data Lake	(Hlupic et al., 2022); Giebler et al. (2019)
To model a DW that is used together with a DL for storage	Priebe et al. (2021)
Model Data Lake metadata	Nogueira et al. (2018); P. Sawadogo & Darmont (2021); Wieder & Nolte (2022); P. N. Sawadogo, Kibata, et al. (2019)

2.2.4.4 RQ4: From the Data Lake to a Data Warehouse based on Data Vault

Ravat & Zhao (2019a) considers that Data Lakes should coexist with Data Warehouses, because they have different objectives and users, however, that kind of system has not yet been studied or implemented. The authors also raise some questions on this architecture (DL feeding a DW): (i) from which zone is the data extracted (process zone or access zone)? (ii) how is the data integrated in the DW? (iii) should the data extracted go through an ETL, ELT or only EL process before entering the DW? (iv) is the data transformed in real time, near real time or batch? (v) what is the type of refreshment (never, complete, or incremental)? (vi) what technical architecture should be used to ensure power and reliability of the data flows between the DL and the DW?

Two papers that review the state-of-the-art of Big Data architectures discuss the combination of Data Lakes and Data Warehouses. P. Sawadogo & Darmont (2021) considers two approaches: Data Lake sourcing a Data Warehouse and Data Warehouse as a component of the Data Lake. In the first approach, the Data Lake is used as a staging area for the DW before the ETL process takes place, allowing for OLAP analyses in the DW and ad-hoc analyses directly from the DL, over the same data. Because, in this case, DLs and DWs are functionally separated, a data siloing issue arises, which can be leveraged by the second approach, where all data is managed in a single platform, allowing for better data lineage control. Wieder & Nolte (2022) considers a Data Lakehouse architecture, with a Delta Lake layer, which facilitates the separation of the processing and storage of data, however, limits the file formats and possible use cases. There are some papers that propose or discuss an architecture that

uses a Data Lake as a source for a Data Warehouse, but most of them don't focus specifically on how data should be stored and loaded in this type of architecture. Also, none of the architectures presented considers Data Vault for modelling the Data Warehouse.

In the financial sector, Pisoni et al. (2021) presents an architecture that shows symbiosis between a Data Lake and a DW but does not show how the two integrate and work together. Sienkiewicz & Wrembel (2021) presents the architecture of a cloud data repository (Data Lake) which includes internal and external data sources and a DW, that is fed by those sources through ETL/ELT processes. The authors raised some challenges regarding this architecture, such as designing efficient ETL/ELT processes (tasks may need to run in parallel and be reorganized) and handling the evolution of data sources (can lead to incomparable snapshots of data, when extracting deltas), but did not provide practical solutions.

Orescanin & Hlupic (2021) implements a Data Lakehouse architecture, where a Data Lake and a Data Warehouse coexist and are managed by a Process Control Framework (PCF), and suggest two ways for combining the existing systems: establishing a virtualization layer over the DL and the DW, or using a metadata-driven querying engine that combines the query results of the DL and DW. In this architecture, the DW is fed by the foundation layer of the DL (containing all historical data with changes preserved) and can also offload data into that same area, when it is not being needed for analysis.

Contrarily to the Data Lakehouse, Saddad et al. (2020), Solodovnikova & Niedrite (2020b), Oukhouya et al. (2023) and Nambiar & Mundra (2022) propose architectures where the DL and the DW are clearly separated, only the first serves as a source for the latter. Both Solodovnikova & Niedrite (2020b) and Oukhouya et al. (2023) consider the Data Lake as a "data highway" where the first level/zone stores the raw data and the last has all the integrated and aggregated data ready to be loaded into the DW, through ELT processes. Both of them also consider dimensional modelling for the DW, and Solodovnikova & Niedrite (2020b) suggests the integration of surrogate keys from the DW's dimensions into the DL's data, to support data provenance tracking and evolution. However, this research is not interested in architectures where the DW is modelled using facts and dimensions, but rather using Data Vault. Nambiar & Mundra (2022) also mention the possibility of using the Data Lake to offload heavy processes from the DW as well as store archive data, to free up storage and bandwidth in the DW. Lastly, Saddad et al. (2020) proposes an architecture where the data fed into the DW comes from the Delta Lake layer, where the refined tables are stored. The Data Lake in this architecture is only responsible for raw data storage.

We can conclude that almost all studies mention ELT processes and the need for a schema-on-read approach in this type of architectures, since schemas cannot be enforced into semi-structured and unstructured data. Data should be loaded into the DW first, and only then transformed. However, there is clearly a lack of practical solutions and best practices for data loading, especially catered to DW modelled using Data Vault. Table 16 synthesizes the diverse ways a DL can interact with a DW in the same architecture.

Table 16 - Relationship between the DL and the DW in the architecture

RELATIONSHIP BETWEEN DL AND DW	SOURCES
Data Lake sourcing a Data Warehouse	Ravat & Zhao (2019a); P. Sawadogo & Darmont (2021); Pisoni et al. (2021); Sienkiewicz & Wrembel (2021); Saddad et al. (2020); Solodovnikova & Niedrite (2020a); Oukhouya et al. (2023)
Data Warehouse sourcing a Data Lake	Ravat & Zhao (2019a)
Data Warehouse as a part of the Data Lake	P. Sawadogo & Darmont (2021)
Data Lakehouse	Wieder & Nolte (2022); Orescanin & Hlupic (2021)
Data Lake and Data Warehouse in the same architecture but used separately (one for unstructured data and the other for structured data)	Autarrom et al. (2022)
Data Lake sourcing a DW and DW offloading data/processes into the DL	Nambiar & Mundra (2022)

2.2.4.5 RQ5: Delta Lake Layer

Delta Lake concepts are still very recent and quite undocumented in the literature. However, from the analysed papers, we can conclude that Delta Lake layers are being used mainly in the context of Data Lakehouses and Data Lakes that serve as a source for a Data Warehouse (Orescanin & Hlupic, 2021; Wieder & Nolte, 2022; Saddad et al., 2020). Delta Lake is mostly mentioned as a metadata layer for the Databricks Lakehouse Platform that gives reliability to the Data Lake, with its advantages being: (i) allowing for both streaming and batch processing, (ii) providing ACID transactions and guaranteeing the consistency of data in the Data Lake, (iii) providing access to previous versions of the data (data versioning), (iv) supporting schema evolution in the Delta Tables, (v) allowing data updates without having to traverse the entire Data Lake, (vi) handling late-arriving data, by processing data as it arrives, (vii) provides a metadata layer, allowing SQL-like access to tables using transaction logs in the Parquet format.

Saddad et al. (2020) is the only paper proposing an actual implementation of a Delta Lake layer on top of a Data Lake architecture based on Apache Spark. The Delta Lake has two layers: (i) the atomic layer, storing “Silver” tables containing refined data, (ii) the departmental layer, storing “Gold” tables, containing aggregated data. This refined data from the Delta Lake can then be used as input to a Data Warehouse. The Apache Parquet format is used to store all tables, which can then easily be translated into Delta Tables. Table 17 summarizes the benefits of the Delta Lake layer.

Table 17 - Benefits of Delta Lake

BENEFITS OF DELTA LAKE	SOURCES
Allowing for both streaming and batch processing	Saddad et al. (2020); Ren et al. (2021)
Providing ACID transactions	Saddad et al. (2020); Ren et al. (2021); Wieder & Nolte (2022)
Supporting data versioning	Saddad et al. (2020)
Supporting schema evolution	Saddad et al. (2020)
Allowing data updates without having to traverse the entire Data Lake	Saddad et al. (2020)
Handling late-arriving data	Saddad et al. (2020)
Providing scalable metadata processing	Orescanin & Hlupic (2021); Saddad et al. (2020); Ren et al. (2021); Wieder & Nolte (2022)

2.2.5 Discussion

This research sought to investigate current best practices and/or guidelines for metadata management, data storage, access and loading from a Data Lake to a Data Warehouse based on Data Vault 2.0, as well as the impact of a Delta Lake layer on this type of architecture. A systematic literature review was performed and information regarding each research question was synthesized, together with tables summarizing the main concepts with the sources that mentioned them.

Based on the thorough synthesis and analysis of the collected studies, we can affirm that there is no single study that discusses an architecture where a Data Lake serves as a source for a DW based on Data Vault, let alone Data Vault 2.0. Nonetheless, some general conclusions can be extracted regarding Data Lake zones, metadata modelling and management, Data Vault modelling in the context of Data Lakes, and Delta Lake layers.

In terms of the Data Lake structure, the zone architecture is the most used and suited for Data Lakes, especially for those that feed a Data Warehouse, because unlike the pond architecture, all the cleansed data is stored in one place (trusted/distilled zone), which makes it easier to load into the EDW. At least 3 zones are always present: one for raw data, other for structured data and a final zone for aggregated data, ready to for consumption. Some papers also consider the existence of another zone between the structured and access zone, for standardized data which is just raw data in an optimized format for cleaning. Another zone is sometimes adopted, the sandbox zone, for exploratory data analysis (e.g., machine learning, data science). One paper also proposes a Data Vault-based Data Lake architecture, where the zones that store structured data are modelled using Data Vault.

A Data Lake sourcing a Data Warehouse is a way to support increasing volumes of heterogeneous data while maintaining a trusted source for structured data and separating storage and computing. However, most works that present an architecture of this type consider a dimensional (Kimball) model for the Data Warehouse. There are also three papers that mention a Data Lake sourcing a DW based on Data Vault specifically, but don't explain how to integrate the two or how to organize data inside the Data Lake to make the process of loading into the EDW more efficient.

Metadata is pointed out as crucial to prevent Data Lakes from turning into data swamps by almost every study, especially in the case where more than one repository is involved, as there is a need to keep track of data provenance and evolution. There are multiple approaches in the literature to model metadata – relational, graph, and Data Vault-based –, with various pros and cons, but the choice ultimately depends on the environment and data architecture. Metadata modelling using Data Vault is an approach that allows easy and flexible schema evolution, although it falls behind in other features. In terms of content, at least metadata regarding the data itself and its relationships with other data needs to be stored, to establish connections between data and extract valuable insights.

Finally, regarding Delta Lake, its refined tables – Gold tables – can be used as a source for the Data Warehouse, as mentioned in one paper. However, the study does not elaborate on the benefits of that solution or how to load the gold tables into the DW.

A lot of open questions and problems still need to be addressed. There is still a need for a generic metadata management model that provides thorough implementation guidelines and that considers metadata in all areas of the Data Lake, while supporting schema and data source evolution. There is

also an incredible lack of research on architectures that take advantage of the Data Lake as a source for the Data Warehouse, as these are mainly used as separate approaches or as a single architecture (e.g., Data Lakehouse). The studies that mention a Data Lake sourcing a DW, only present the architecture without providing implementation guidelines/best practices or are review papers that comment on the lack of literature on this topic, while raising open research questions. In terms of Data Vault modelling, there is also very little research, as it is mentioned as an approach for modelling structured data and metadata in Data Lakes, but without any guidance on how it can be done. Delta Lake concepts are tightly coupled to the Delta Lake storage layer by *Databricks* and, because of that, there aren't many published articles on how to integrate it into specific architectures, like the one we are focused in. However, we were able to discover some of its benefits and found an example of a Data Lake architecture using a Delta Lake layer to feed a DW.

2.3 Grey Literature Review

2.3.1 Research Background

Even when it comes to Grey Literature, there is a reduced number of articles that discuss architectures comprising a Data Lake that serves as a source for a DW based on Data Vault 2.0 and/or discuss the impact of the Delta Lake concept on this type of architecture. However, there is a substantial amount of works that consider the Data Lake and DW as complementary solutions instead of mutually exclusive (Baumann, 2022; Chu, 2020; Coates, 2017; Etse, 2022; Golec, 2019; Kyslyi, 2021; Olschimke, 2022; Zaloni, 2016). The importance of folder organization inside the Data Lake is also mentioned in the Grey Literature, unlike in academic papers ("An Efficient Data Lake Structure," 2019; Coates, 2017; Mitruś, 2021). Nonetheless, there is an evident lack of consensual implementation practices for two-tier architectures, although they are the most common in the industry today (Armbrust et al., 2021; Golec, 2019; Mike, 2022; Zaloni, 2016) and few to no research on Delta Lakes used together with a DW. We can see an increase in the number of accepted works throughout the years, especially in the years 2021 and 2022, with Blog Posts as the most common type of work (Figure 3).

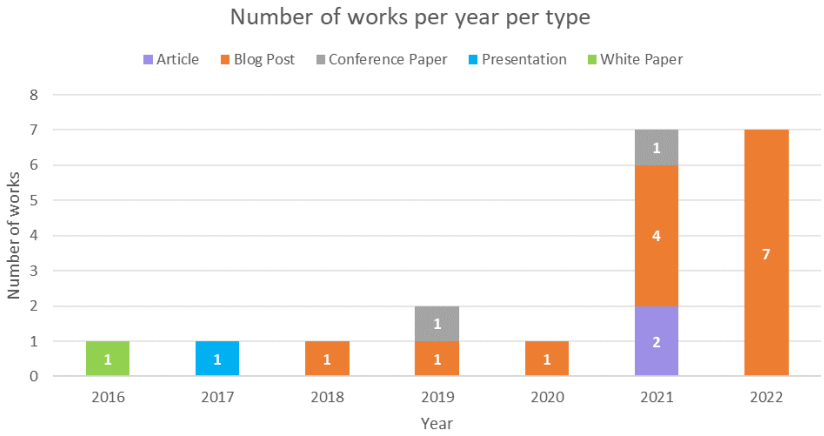

Figure 3 - Number of accepted works per year and type

2.3.2 Planning

2.3.2.1 Motivation

The decision on whether to perform a grey literature (GL) review was made systematically according to criteria adapted from Garousi et al. (2019), where one or more "Yes" answers implies the need for a GL review (Table 18).

Table 18 - Criteria for performing a GL review

QUESTION	ANSWER
Is the subject "complex" and not solvable by considering only the formal literature?	Yes
Is there a lack of volume or quality of evidence, or a lack of consensus of outcome measurement in the formal literature?	Yes
Is the contextual information important to the subject under study?	No
Is it the goal to validate or corroborate scientific outcomes with practical experiences?	Yes
Is it the goal to challenge assumptions or falsify results from practice using academic research or vice versa?	Yes
Would a synthesis of insights and evidence from the industrial and academic community be useful to one or even both communities?	Yes

As we can see from the answers provided above, there is a need to perform a grey literature review, mainly to overcome the gaps in academic research and complement the findings of the formal literature review (SLR) conducted previously.

2.3.2.2 Research Questions

The aim of this literature review is to provide the state-of-the-art regarding Data Lake architectures, more specifically in the case where the data is to be loaded into an Enterprise Data Warehouse based on Data Vault 2.0. Furthermore, we are also interested in finding out the impact that Delta Lake concepts may have on this kind of architecture. To do that, and to fill the same research gaps as the ones mentioned for the formal literature review, this study plans to answer the following questions:

- RQ1: What are the best practices for storing data in a Data Lake?
- RQ2: What are the current best practices for storing metadata in the Data Lake?
- RQ3: How is Data Vault 2.0 design methodology being used to model structured data of a Data Lake?
- RQ4: Are there any proposed practices for storing data in a Data Lake, which will be loaded onto a Data Warehouse based in Data Vault 2.0?
- RQ5: How can a Delta Lake layer impact a Data Lake architecture?

2.3.2.3 Data Sources and Search Strategy

In order to conduct a grey literature review that is as systematic, repeatable, and unbiased as possible, the guidelines by Garousi et al. (2019) and Adams et al. (2017) will be followed. This review is divided into three phases, similarly to the SLR: Planning, Conducting and Reporting.

In the Planning phase, the objectives and motivation of the review, research questions, search string, sources, selection criteria, stopping criteria, types of GL, quality assessment checklist and a data extraction form were defined. To maintain consistency between the two reviews, the same keywords for the search will be used, only with some adjustments in the operators (AND, OR), to get the best results possible out of the non-academic literature. The only source to be used is Google, as a previous exploratory search revealed that other sources (e.g., big technology company websites, such as IBM, Oracle, Databricks and Microsoft) had little to no relevant articles and were very biased towards their own software solutions. Table 19 shows the search string, source and stopping criteria used.

Table 19 - Search strategy

Search String: "Data Lake" AND "Architecture" AND ("Data Warehouse" OR "Data Vault" OR "Delta Lake" OR "Metadata")		
SOURCE	**STOPPING CRITERIA**	**OBSERVATIONS**
Google	Top 200 hits / Evidence Exhaustion	Ads will be removed

The stopping criteria used is effort bounded (top 200 results) and based on evidence exhaustion (search stops when quality and availability of evidence greatly declines), whichever comes first. The types of GL that will be accepted are website articles/pages, forum/blog posts, and any type of non-peer-reviewed paper (peer-reviewed papers will be added to the Formal Literature Review). In each of the sources, snowballing can be performed, i.e., other websites and sources can be consulted because they were referenced in the obtained works.

2.3.3 Conducting

In the Conducting phase, works are selected based on defined inclusion and exclusion criteria. Once the final papers are selected, they are extracted, their quality is assessed, and the information is synthesized.

After applying the search string to Google, 167 results were collected plus 3 using forward snowballing (170 results). The unique results ended after 180 hits and the last few were completely out of the scope of the RQs. Because of that, the search stopped before reaching the end of the results (evidence exhaustion). After the selection process using the inclusion and exclusion criteria, which led to 39 selected works, each of them was carefully read and its quality was assessed. Only 20 works remained after performing the quality assessment (Figure 4).

Figure 4 - Work selection process

2.3.3.1 Inclusion and Exclusion Criteria

The content of each work is briefly read to determine whether it complies to any of the inclusion or exclusion criteria. If a work does not pass any of the inclusion criteria, it is excluded based on one of the exclusion criteria. From the collected works, only 39 were accepted based on the defined criteria. Two of the results were peer-reviewed papers, so they will be included in the formal SLR instead. The total number of works accepted/rejected per criteria is described in Table 20.

Table 20 - Number of works included and excluded by the criteria

#	INCLUSION CRITERIA	EXCLUSION CRITERIA	#
3	Addresses Data Lake architectures and Data Vault 2.0 Modeling	In the context of/refers to Data Lake/Lakehouse architectures, but does not discuss the architecture in detail	57
24	Addresses a Data Lake architecture combined with a Data Warehouse	Addresses Data Lake architectures but not together with a Data Warehouse	9
1	Addresses metadata management in a Data Lake/Lakehouse architecture	Addresses both Data Lakes and Data Warehouses but as separate approaches	30
11	Addresses the effect of a Delta Lake layer on a Data Lake/Lakehouse architecture	Full text not in English language	0
		Unavailable	3
		Published before 2016	0
		Duplicated	0
		Already reviewed in the academic SLR	6
		Out of the scope of the RQs	20
		Integral book	1
		Considers a combination of a DL and DW only as a Data Lakehouse	3
		Peer-reviewed study (move to SLR)	2
39			131

2.3.3.2 Quality Assessment

To assess the quality of the works, a more thorough assessment needs to be made, when compared to formal, peer-reviewed literature that is published in academic sources. The QA questions defined in Table 21 were adapted from Garousi et al. (2019) with some alterations. The Content criteria were added, to match the QA for the Formal Literature Review, as we want to ensure that the papers are still relevant to the research questions. Regarding the Authority of the producer, Methodology, Objectivity, Novelty and Impact criteria, some questions were removed as they were, in most cases, very difficult or even impossible to answer with the information available in the sources. To evaluate the expertise of the authors, their resumes, *Linkedin* profiles or personal websites were consulted, to determine whether they have been working in the field for some time, if they have somewhat contributed to research on the subject and/or if they are associated with a reputable organization.

For each study, all the questions were answered as “Yes”, “Partially” or “No”, with the corresponding values of 1, 0.5 and 0, respectively. In the case of the Outlet type question, the Yes=1/Partially=0.5/No=0 measures will be used as the measures for the 1st, 2nd, and 3rd Tiers, respectively. The questions related to the Content and Outlet type have a weight of 1, unlike the rest, which have a weight of 0.5 in the final sum. Then, the weighted sum of the values was calculated for each study, as well as the normalized sum. The cut-off score used to determine whether a work is

considered for further analysis was 4.5/10 (works with a score of 4.5 were still included in the analysis). In the end, only 20 works were accepted by the quality assessment process.

Table 21 - Quality assessment checklist

CRITERIA	QUESTIONS	WEIGHT
Content	Does the study discuss the storage and loading of the data from a Data Lake to an EDW?	1
	Does the study discuss the impact of Data Vault 2.0 modeling in the context of a Data Lake architecture?	1
	Does the study discuss the impact of a Delta Lake layer on the Data Lake/Lakehouse architecture?	1
	Is metadata management discussed?	1
Authority of the producer	Is the publishing organization reputable?	0.5
	Is an individual author associated with a reputable organization?	0.5
	Does the author have expertise in the area?	0.5
Methodology	Does the source have a clearly stated aim?	0.5
	Is the source supported by authoritative, contemporary references?	0.5
Objectivity	Is there no vested interest of a particular company or technology?	0.5
Date	Does the item have a clearly stated date?	0.5
Position w.r.t. related sources	Have key related GL or formal sources been linked to / discussed?	0.5
Novelty	Does it enrich or add something unique to the research?	0.5
Impact	Does the work have backlinks?	0.5
Outlet type	What is the tier of the GL work? (1st Tier: High outlet control / High Credibility; 2nd Tier: Moderate outlet control / Moderate Credibility; 3rd Tier: Low outlet control / Low Credibility)	1
Cut-off Score: Weighted sum < 4.5		

2.3.4 Reporting

2.3.4.1 RQ1: Data Lake Architectures

2.3.4.1.1 Zone-based architectures

The most common Data Lake architecture found in the Grey Literature is the one divided in zones based on the level of processing of the data. *Blueprint* (2021) also mentions a division based on the processing stage of the data and further on business dimensions (e.g., Country).

Mitruś (2021) considers 3-5 zones for Data Lake architectures, based on the company's experience. In fact, all analysed works consider a minimum of 3 and maximum of 5 zones. At least one zone for raw data storage, one for cleaned, standardized data and other for curated, aggregated data ready for consumption and possibly with security and governance measures applied. The designation for these zones varies between articles, but their purpose is mostly the same. Some articles also consider a transient/temporary/landing zone that exists to store data from the sources temporarily before it is persistently stored in the raw data zone (*Blueprint*, 2021; Coates, 2017; Kyslyi, 2021). Apart from the 3 main zones, a standardized data layer that precedes the cleansed data zone can also exist, as well as an application data layer for cleansed data with enforced business logic (Mitruś, 2021). Some works also consider a sandbox zone for explorative analytics directly on the raw data, which can be considered optional (Mitruś, 2021; Mike, 2022; Golec, 2019; Mikhailouskaya, 2018; Coates, 2017;

Kyslyi, 2021). In terms of Data Lakehouse architectures, as they combine a Data Lake and Data Warehouse in the same architecture, their layers also go from raw data ("Bronze" tables) to standardized and cleaned data ("Silver" tables) to aggregated data ("Gold" tables) that can be served to BI and analytics applications (Kukreja, 2021). Table 22 summarizes the types of zone-based architectures found in the literature.

Table 22 - Zone-based Data Lake architectures

ARCHITECTURE	SOURCES
Lakehouse (Bronze, Silver, Gold)	Kukreja (2021)
5 Zones (Raw, Standardized, Cleansed, Application, Sandbox)	Mitruś (2021)
3 Zones (Raw/Landing, Processed/Development, Curated/Production	Chu (2020); Mike (2022)
4 Zones (Transient/Temporary/Landing/Loading, Raw/Staging, Curated/Refined, Sandbox)	Mikhailouskaya (2018); Coates (2017); Kyslyi (2021)

2.3.4.1.2 Organization in the Data Lake

In order to prevent turning the Data Lake into a data swamp, its storage should be properly organized into a reliable folder structure (Mitruś, 2021). The ease of access to the Data Lake storage downstream (e.g., loading into an EDW) depends on how efficiently stored the data is. The Data Lake should be structured according to the metadata of the source systems ("An Efficient Data Lake Structure," 2019). According to Mitruś (2021) and Coates (2017), the organization of the Data Lake can be based on a variety of factors: (i) time partitioning, (ii) data load patterns (real-time, streaming, incremental, full load, one time), (iii) subject area/source, (iv) security boundaries, (v) downstream app/purpose, (vi) owner/stewardship, (vii) retention policies (temporary, permanent, time-fixed), (viii) business impact (high, medium, low, critical), (ix) confidential classification (public information, internal use only, personally identifiable information, sensitive, etc.), (x) probability of data access (recent, historical, etc.).

Coates (2017) and Mitruś (2021) recommend an initial generic folder structure for raw data storage – Subject Area/Data Source/Object/Year/Month/Day/File(s). For cleansed/curated data and application data, Mitruś (2021) recommends a Purpose/Type/Files structure, and Coates (2017) adds a Snapshot Date after "Type" – Purpose/Type/Snapshot Date/Files. With this division, there are no distinct security or organizational boundaries, which can be solved by adding an "Organizational Unit" folder in the highest level of the hierarchy (before "Subject Area" and "Purpose"), however, that can also lead to siloed and duplicated data across organizational units (Coates, 2017). "An Efficient Data Lake Structure" (2019), which considers the Data Lake simply as a persistent staging area before loading data into a Data Vault 2.0-based EDW, recommends a different folder structure – Source System/Connection/Schema Name/File Name/Load Date Timestamp, where the connection represents one of the possible connections to the source system, the schema name represents the structure of the database and the file itself also has a Load Date Timestamp attribute and a sub sequence number.

2.3.4.2 RQ2: Metadata in Data Lakes

Metadata is the key to easily finding data in the Data Lake and deriving value from it with confidence (Zaloni, 2016). However, this research revealed a substantial lack of information on how to store metadata in the Data Lake. The majority of collected works that mentioned metadata as a way to better govern the Data Lake and prevent its data from becoming unusable, did not further develop on best practices or guidelines for metadata storage.

Zaloni (2016) recommends the division of metadata in three types – technical (structure of data), operational (lineage, quality, profile, and provenance of data) and business (semantic) – to get the most complete view of the data. Metadata is also discussed in Armbrust et al. (2021) as a layer that enables Lakehouses to support ACID transactions and other warehouse-like features. Operations are not atomic in Data Lakes, as a table's data can be spread through multiple files. Because of that, management layers that track which objects are part of a table in a transaction log (in Parquet/ORC format) and generate a table format (e.g., Delta Lake) started to be developed. These table formats have proven to have similar or better performance than raw Parquet/ORC files and add useful features apart from ACID transactions, such as data versioning and being able to make changes to tables without affecting the underlying data. These metadata layers are easy to implement on top of an already existing Data Lake and also support schema enforcement/evolution, set constraints on the ingested data, and can be used to implement governance features, such as access control and audit logging. However, this is still a very recent topic with room for improvement. As of right now, the metadata layers can only support transactions one table at a time, which could be improved by storing the transaction logs in a faster storage system, as opposed to just using the Data Lake's object storage.

2.3.4.3 RQ3: Data Vault Modelling

Companies tend to choose between implementing a Data Lake or a Data Warehouse, however, Data Vault 2.0 can be used as a way to integrate the advantages of both solutions (Olschimke, 2022). *Scalefree*, in the blog post by Olschimke (2022), recommends the implementation of a hybrid architecture where the Data Vault 2.0 is integrated with a Data Lake, with the latter serving as a staging area. The Data Vault part is composed by the Raw Data Vault, which integrates and produces versions of the data coming from the DL while breaking it down into business keys (*hubs*), relationships (*links*) and descriptive data (*satellites*), the Business Data Vault, which enforces business logic and security measures on the data, and finally the Information Marts, which deliver the final data for consumption in a target schema. This model allows the management of data and business logic as in a typical Data Warehouse but following a schema-on-read approach as in Data Lakes, as the target schema is only applied in the last layer of the Data Vault (Information Marts), which makes it easy to load data from the DL into the Raw Vault. It is also scalable and adapts easily to business rule or structure changes, which is also an advantage since the Data Lake is constantly receiving large amounts of data.

Chu (2020) presents a reference Data Lake + EDW architecture that considers Data Vault modelling for the EDW, however the data entering the EDW does not come from the Data Lake, but from the original sources, passing through an intermediate staging area. The Data Lake and the Data Warehouse are used for separate use cases, although communication between the two is possible. Because of this,

using the Data Vault methodology for DW design does not pose any constraints to the storage and organization of the Data Lake.

Arnold (2021) showcases the benefits of using Data Vault in a modern data platform that can include a Data Lake as a persistent staging area before loading the data into the Raw Vault, where the main Data Vault model is stored (hubs, links, and satellites). The main advantage of using Data Vault is that it enables repeatable and consistent patterns for data loading into the EDW. The pros of using Data Vault in a Data Lake architecture, as stated by Arnold (2021) are: (i) it's insert only, (ii) supports historical record tracking, (iii) provides auditability, (iv) can be built incrementally, (v) is adaptable to changes without re-engineering, (vi) enables parallel data loads, (vii) is technology-agnostic, (viii) has fault-tolerant ingestion pipelines. On the other hand, the cons are: (i) the models can be complex, (ii) teams need specific training, (iii) needs high amounts of storage to keep historical data, (iv) data isn't immediately user-ready (needs to pass through the business vault and ultimately information marts for business logic to be enforced).

2.3.4.4 RQ4: From the Data Lake to a Data Warehouse based on Data Vault

There are many mentions in the grey literature to data architectures that include a Data Lake sourcing a Data Warehouse, however there is still no common designation for it. Some works that consider this type of architecture call it a Cloud Data Warehouse (Mike, 2022), a Cloud Data Lake (Mazumdar, 2022), Two-tier architecture (Etse, 2022; *Data Lakehouse*, 2021; Armbrust et al., 2021), Hybrid architecture (Baumann, 2022; "An Efficient Data Lake Structure," 2019; Kyslyi, 2021: Olschimke, 2022) or Modern Data Warehouse (Coates, 2017). Regardless of its designation, the two-tier architecture gained popularity because Data Lakes alone have difficulties in providing data quality and governance downstream (Armbrust et al., 2021; Etse, 2022). By loading data from the Data Lake into a Data Warehouse, we can have data with a defined schema, that can support BI applications, while raw data can still be accessed by advanced analytics workloads (Kyslyi, 2021) directly in the Data Lake, which is why it is the dominant architecture in the industry today (Armbrust et al., 2021). Zaloni (2016) considers that Data Warehouse augmentation, which corresponds to the integration of a Data Lake with an existing DW, is a smart first step for companies looking to add flexibility and speed to data processing and capturing, while freeing up bandwidth in the DW for BI analytics and cutting costs on storage by leveraging the cheap commodity hardware of the Data Lake. The main advantages of this architecture are: (i) cheap data storage of the Data Lake, (ii) support of various frameworks for machine learning workloads while still having a subset of curated data in the DW for BI applications, (iii) support for all structured, unstructured and semi-structured data, (iv) cost-efficient solution, (v) decoupled storage and compute, (vi) allows offload of time-consuming ETL processes to the DL, (vii) allows offload of archive data to the DL, reducing storage costs of the DW, (viii) fast data loading into the DL, with parallel batch processing and schema-on-read approach, which enables faster analytic insights, (ix) better control over data usage and compliance, as sensitive data can be rerouted directly to the DW without entering the DL. Table 23 summarizes the advantages of the two-tier architecture.

Table 23 - Advantages of the two-tier architecture

ADVANTAGES OF THE TWO-TIER ARCHITECTURE	SOURCES
Cheap data storage	Etse (2022); Zaloni (2016); Armbrust et al. (2021); Kyslyi (2021)
Simultaneous support for ML workloads and BI applications	Etse (2022); Zaloni (2016); "An Efficient Data Lake Structure" (2019); Mazumdar (2022); Armbrust et al. (2021); Kyslyi (2021)
Support for all types of data	Etse (2022); Mazumdar (2022); Coates (2017)
Decoupled storage and compute	Etse (2022); Mazumdar (2022); Armbrust et al. (2021)
Offloading time-consuming processes to the DL	Mitruś (2021); Baumann (2022); Coates (2017)
Offloading archive data from the DW to the DL	Mitruś (2021); Coates (2017)
Fast data collection	Kyslyi (2021)
Better control over data usage and compliance	Kyslyi (2021)

However, some problems with the two-tier architecture are identified in the literature, which were what ultimately led to the appearance of Data Lakehouse architectures. Some of those are: (i) The fact that BI analysts only have access to the DW part of the architecture, relying on data engineers for data structuring and metadata cataloguing, (ii) coarse-grained access control to data (as opposed to fine-grained), as all data is stored and managed as files, (iii) difficulty managing file versions, (iv) the high cost of maintaining two storage systems, (v) data duplication and inconsistencies, (vi) high complexity for users, (vii) high risk of delays and failures in the ETL process between the DL and the DW, (viii) files may not be optimized for downstream applications, (ix) data staleness, as data from the DL may take days to load, (x) limited support for advanced analytics, as the raw data in the DL is not ACID-compliant and does not support data versioning and indexing, (xi) lack of transaction support, and (xii) poor scalability of the DW when compared to the DL. Table 24 summarizes the potential problems of the two-tier architecture.

Table 24 - Problems with the two-tier architecture

PROBLEMS WITH TWO-TIER ARCHITECTURE	SOURCES
BI analysts have limited access to data	Etse (2022)
Coarse-grained access control to data	Etse (2022)
Difficulty managing file versions	Etse (2022)
High cost of maintaining two systems	Etse (2022); *Data Lakehouse (*2021); Armbrust et al. (2021)
Data duplication and inconsistencies between the DL and DW	Etse (2022); Mazumdar (2022); Armbrust et al. (2021); Mike (2022)
Delays and failures	Etse (2022); Mazumdar (2022); Armbrust et al. (2021)
High complexity	Etse (2022); *Data Lakehouse* (2021)
Files may not be optimized for downstream apps	Mazumdar (2022)
Data staleness	Etse (2022); Armbrust et al. (2021)
Limited support for advanced analytics	Armbrust et al. (2021)
Lack of transaction support	Mike (2022); Armbrust et al. (2021)
Poor scalability of the DW	Etse (2022)

According to Armbrust et al. (2021), the fairly new Data Lakehouse architecture is able to solve the problems mentioned above by integrating Data Lake and Data Warehouse capabilities in the same

architecture together with a metadata layer, providing transactional views of the Data Lake, enabling DW-like management features, such as data versioning and schema enforcement and improving data reliability and consistency.

After a critical analysis of the works, it is possible to conclude that an architecture which has a Data Lake sourcing a Data Warehouse is different from a Data Lakehouse, however, the line between the two is still fuzzy in some articles. Most of them characterize the Data Lakehouse as a monolithic architecture that combines the benefits of a Data Lake (schema-on-read, storing unstructured data, scalable, cheap storage, etc.) with the benefits of a Data Warehouse (ACID transactions, data indexing and versioning, schema enforcement, etc.). Nonetheless, there are still some works that define the Data Lakehouse simply as a Data Lake integrated with a separate Data Warehouse (Kukreja, 2021), which is not aligned with the definition of the Lakehouse, provided in the white paper by Armbrust et al. (2021) and other sources (*Data Lakehouse*, 2021; Etse, 2022; Mazumdar, 2022), where the two-tier architecture is clearly distinguished from the Data Lakehouse architecture.

Although the problems of the two-tier architecture can be solved by implementing a Lakehouse, the first architecture is still a viable option for companies who are trying to take a step further into big data while keeping their existing Data Warehouse, instead of implementing a new solution from scratch. Some of the covered works tackle the specific case of a an architecture where the Data Lake serves as a persistent staging area for a Data Warehouse based on the Data Vault/Data Vault 2.0 methodology ("An Efficient Data Lake Structure," 2019; Arnold, 2021; Olschimke, 2022). As seen in the summary of RQ3: Data Vault Modelling, this specific methodology can improve the two-tier architecture, because it follows a schema-on-read approach as the Data Lake, meaning that it is easy to load raw data into the Raw Data Vault (first layer of the DW) because the target schema is only applied in the last layer. Moreover, it addresses the scalability problems of the DW, since Data Vault modelling is very adaptable to schema and business changes.

2.3.4.5 RQ5: Delta Lake Layer

The concept of Delta Lake is tightly coupled with Data Lakehouse in the literature, as the first is commonly referred to as an implementation of a Lakehouse by *Databricks*. However, the Lakehouse is a broader concept, it is a data architecture, which is comprised of a storage layer, usually referred to as the Data Lake itself, raw file formats (Apache Parquet, ORC, etc.) in which data is stored, table formats (e.g., Delta Lake), which provide an abstraction layer over the file formats, keeping track of which files constitute a table and its changes over time (in a transaction log), query engines and finally, the downstream applications that use the data (Mazumdar, 2022). As the Delta Lake is the most important part of the Lakehouse, being the metadata layer that enables ACID transactions and multiple management features, its benefits are mostly in line with the ones of the Lakehouse. The advantages of the Delta Lake and similar formats are: (i) ACID transactions, allowing only one change at a time to preserve consistency, (ii) data versioning, enabled by the transaction log kept in delta tables and CDF (Change Data Feed), which tracks row-level changes in tables, (iii) zero-copy cloning, which means Delta Lake can convert a collection of Parquet files into delta tables just by adding a transaction log and without copies, (iv) schema enforcement and evolution, (v) access control and audit logging, (vi) high SQL performance on top of the object storage by using data caching, auxiliary files, indexing, etc., (vii) data clustering and compaction, to solve the small file problem, as querying many small files is

more time-consuming that querying less larger files, (viii) scalable metadata handling, (ix) unified batch and streaming data processing, as streaming and batch data can be merged into the same table, (x) data sharing protocol to secure data sharing between organizations, (xi) no vendor lock-in, as it is based on open formats, (xii) allows DML (Data Manipulation Language) operations directly on the distributed files. Table 25 summarizes the benefits of the Delta Lake layer.

Table 25 - Benefits of Delta Lake

BENEFITS OF A DELTA LAKE LAYER	**SOURCES**
ACID transactions	Armbrust et al. (2021); Etse (2022); Tondak (2022); Baumann (2022); Späti (2022); *Data Lakehouse (*2021); Mazumdar (2022); Mike (2022)
Data versioning	Armbrust et al. (2021); Baumann (2022); Späti (2022); *Data Lakehouse (*2021); Mazumdar (2022); Mike (2022)
Zero-copy cloning	Armbrust et al. (2021); *Data Lakehouse (*2021)
Schema enforcement and evolution	Armbrust et al. (2021); Etse (2022); Baumann (2022); Späti (2022); *Data Lakehouse (*2021); Mazumdar (2022); Mike (2022)
Access control and audit logging	Armbrust et al. (2021); Etse (2022)
High SQL performance	Armbrust et al. (2021); Etse (2022); Baumann (2022); Mike (2022)
Compacts file size for faster queries	Etse (2022); Baumann (2022); Späti (2022); Mazumdar (2022)
Scalable metadata handling	Armbrust et al. (2021); Etse (2022); Tondak (2022); Späti (2022)
Unified batch and streaming data processing	Armbrust et al. (2021); Tondak (2022); Späti (2022); Mike (2022)
Data sharing	Späti (2022)
No vendor lock-in	Mazumdar (2022)
DML operations on distributed files	Späti (2022); Mike (2022)

Delta Lake/Data Lakehouse was introduced as an attempt to solve the issues with the two-tier architecture (Armbrust et al., 2021), as it provides the features of a Data Warehouse to the Data Lake's data and, because of that, having a DW on top of this architecture would be redundant (Etse, 2022). However, one work considers the possibility of having a cloud DW for some BI use cases (*Data Lakehouse*, 2021). Additionally, because the Lakehouse allows concurrent streaming and batch processing, it makes the Lambda architecture obsolete (Mike, 2022; Späti, 2022; Tondak, 2022). Moreover, the Lakehouse architecture also has some problems and open issues, some of which are described in Table 26.

Table 26 - Problems with Lakehouse architectures

PROBLEMS WITH LAKEHOUSE ARCHITECTURES	**SOURCES**
Fairly recent architecture	Etse (2022); *Data Lakehouse* (2021)
Can only query delta lake tables and not external tables	Etse (2022)
Using notebooks for data analysis can be more complex than the interface of DWs	Etse (2022)
Monolithic structure of the Lakehouse can be difficult to build and maintain	*Data Lakehouse* (2021)
Lakehouse capabilities go much further than the current technology needs of most companies	*Data Lakehouse* (2021)

Storage system for metadata can be improved (currently stored in the same object storage as the data)	Armbrust et al. (2021)
Maximum number of tables processed at the same time in transactions is currently just one	Armbrust et al. (2021)

In *Data Lakehouse* (2021), it is also mentioned that the value brought by the Data Lakehouse has been questioned by critics, as some claim that the two-tier architecture can be just as efficient if managed properly and combined with automation tools.

2.3.5 Discussion

This literature review sought to investigate current best practices and/or guidelines for metadata management, data storage, access and loading from a Data Lake to a Data Warehouse based on Data Vault 2.0. A systematic literature review was attempted, following the guidelines by Garousi et al. (2019), although it is much more difficult to provide unbiased and repeatable results when collecting and analysing grey literature, which is very diverse in nature.

After a thorough analysis and synthesis of the accepted works, 51 in total, from both the Formal Literature Review and the Grey Literature Review, we can now establish the state-of-the-art regarding Data Lake architectures, especially when used together with a Data Warehouse, the use of the Data Vault methodology in the context of Data Lakes, metadata management in the Data Lake and also the impact of a Delta Lake layer on this type of architecture. Additionally, we can conclude that there is a substantial lack of literature that interconnects all the keywords selected for this research. The works that do mention an architecture comprising a Data Lake and a Data Warehouse based on Data Vault 2.0, do not go into much detail on implementation practices.

Regarding Data Lake architectures, the zone-based architecture is the most widely adopted architecture for Data Lakes, with 3-5 data zones going from a low degree of processing to a high degree of processing, and allowing the storage of cleansed, curated data all in the same place, ready to be loaded into a Data Warehouse. There is at least one zone for persistent raw storage, one for standardized/cleansed data and other for aggregated, consumable data.

It is important to organize the Data Lake in folders according to, at least, subject area and/or source as well as the time (year, month, day) the data was collected. While data moves towards more cleansed and structured zones, it's useful to separate it according to purpose and type of files, making it easier to load it into specific downstream applications.

Metadata management, whether included in a metadata layer such as the Delta Lake, or through a metadata management system, is essential to keep track of data provenance and evolution in the Data Lake and prevent it from turning into a data swamp. At least intra and inter-metadata, representing the properties and relationships between data respectively, should be stored for easy data management and querying. Apart from that, no best practices, or guidelines for storing and managing metadata in the Data Lake were found.

Investing in a Data Lake to augment an existing Data Warehouse architecture can be a smart first step for companies looking to take advantage of unstructured data while keeping part of their existing

architecture. Two-tier architectures are currently the most common in the industry, as they support persistent storage of raw data and access to that data while maintaining a trusted source for structured application data. However, they can create some problems down the line, related to the fact that there are too many storage layers overall, those being the zones of the DL, the DW and the downstream applications (e.g., BI tools), which adds complexity, costs, potential inconsistencies with data in different layers and failure modes.

The Data Lakehouse was created as an alternative to the two-tier architecture mentioned above, but is still not widely adopted, although it is assumed to solve many issues of the two-tier architecture.

The Data Lake can be used as a persistent staging area for a Data Warehouse based on the Data Vault 2.0 methodology, as both follow a schema-on-read approach (raw data can be directly consumed by the Raw Data Vault) and are adaptable to business and data structure changes. The Data Vault 2.0 methodology can also be used to model metadata, allowing for flexible schema evolution, which is important in Data Lakes, because of the speed and volume in which data arrives.

The Delta Lake layer combines capabilities of the Lambda architecture and DW-like features in a single metadata layer on top of the Data Lake, that allows a transactional view of the Data Lake in open table formats. There are very few occurrences in the literature of an architecture where a Data Lake with a Delta Lake layer is used together with a DW. Some sources consider that the use of a Delta Lake layer eliminates the need for a DW (it is redundant), and others consider that the Delta Lake's Gold tables can be used as a source for a DW.

Apart from these findings, there are still some open issues. Overall, we can safely conclude that there is a substantial lack of research and industry best practices on how to efficiently load data from the Data Lake into the Data Warehouse, let alone a DW based on Data Vault 2.0.

There is also a need for thorough implementation guidelines of a generic metadata system that supports schema and data source evolution while covering the whole data lifecycle inside the Data Lake.

The research revealed that two-tier architectures are the most common architecture for companies adopting Data Lakes, but there are no generic guidelines on how to integrate a Data Lake with a Data Warehouse and how to manage data and metadata efficiently in this type of architecture. Lakehouse architectures, although being mentioned as an alternative for this architecture, are recent and still have many open questions and issues.

Data Vault 2.0 is only mentioned by two company websites as a methodology fit for a DL+DW architecture and none of the articles goes into much detail on how data should be handled in the Data Lake to be loaded into a DW that is modelled using this methodology.

Although some sources consider the implementation of a Delta Lake layer to be redundant in a two-tier architecture, this approach hasn’t been properly researched, implemented, and demonstrated yet.

These findings, as well as the open issues, support the next phase of this research, the design of a Data Vault 2.0 model for an EDW sourced by a Data Lake, which aims to answer the question: How can we model and integrate a Data Vault 2.0 EDW in a Data Lake architecture, using Delta Lake concepts?

3 METHODOLOGY

The methodology that will be used to conduct this research is Design Science Research (DSR), following the activities proposed by Peffers et al. (2007): (i) Problem identification and motivation, (ii) Definition of the objectives for a solution, (iii) Design and development, (iv) Demonstration, (v) Evaluation, (vi) Communication. Activities (i) to (iv) can serve as research entry points (Figure 5).

The DSR methodology was chosen because the project that this research is inserted in requires the development of an artifact– a scalable DW model to maximize efficiency and scalability. Both the results of the Systematic Literature Review and the requirements of the company have triggered the development of the artifact, making DSR with a problem-centred approach the right choice of methodology.

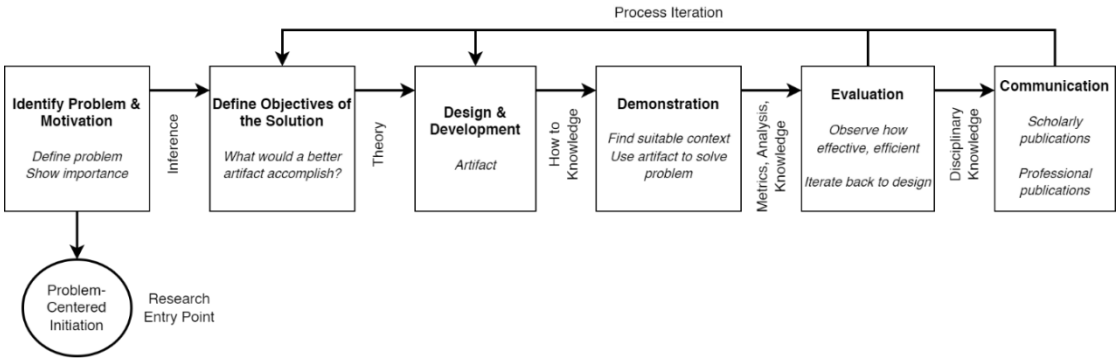

Figure 5 - DSR Model Process (Adapted from Peffers et al., 2007)

In the first phase, the problems that motivated this research are identified and linked to the findings of the literature review, and the context of the replatforming project is explained. In the second phase we present the underlying Data Lake architecture where the Data Vault model will be implemented and define the main objectives for this solution. In the third phase, the solution is presented, and the decisions made in the model designing process are explained. In the Demonstration phase, the flexibility of the model is tested by presenting a use case where new business requirements need to be supported by the model. In the Evaluation phase, qualitative data will be collected through interviews with selected domain experts, to assess the adequacy and usability of the model in the company's environment. Finally, in the Communication phase, this research will be presented to the company, which in turn can use these findings to fully implement a DV 2.0 model and test its performance in a broader environment.

The remainder of this work will be structured according to the phases of the DSR methodology.

4 PROBLEM IDENTIFICATION AND MOTIVATION

In this section, the first phase of the DSR Methodology, Problem Identification and Motivation, is presented.

This research was conducted within the scope of an effort by a company in the banking industry to restructure their data architecture. The company is implementing a brand-new Data Lake to store their high volumes of heterogeneous data, but still want to keep the already existing Enterprise Data Warehouse to store specific information by theme or structure and feed downstream applications. Additionally, the company would highly benefit from a methodology such as Data Vault 2.0 to model the EDW, as it is highly scalable and can support the dynamic and ever-changing nature of the data kept in the company. These needs are what triggered the development of a solution – a DV 2.0 model for an Enterprise Data Warehouse (problem-centred approach).

The Systematic Literature review revealed a significant lack of guidelines and best-practices for two-tier architectures (Data Lake sourcing an EDW) and the use of Data Vault 2.0 and Delta Lake on this type of architectures, which further motivates the proposal of an EDW model that works efficiently in this type of environment. The research question to be answered, associated with the research problem is: How can we model and integrate a Data Vault 2.0 EDW in a Data Lake architecture, using Delta Lake concepts?

4.1 DEFINITION OF OBJECTIVES

The solution to be developed is a Data Vault 2.0 model for the banking company, that fits into the Data Lake architecture presented in the next Section, in the Enriched zone, receiving data directly from the Data Lake and structuring it around the core business concepts of the company. This model will focus on two broad business domains for simplicity – “Customers” and “Accounts” –, specifically current accounts and credit card accounts, and associated concepts. This pilot version of the DW model intends to demonstrate the benefits and adequacy of Data Vault 2.0 to the company’s requirements while keeping it simple and easy to understand, by focusing on a specific subset of the data.

The objectives of this solution are: (i) to reflect the business in an accurate way, taking advantage of the business-oriented modelling approach of Data Vault 2.0, (ii) to easily scale and adapt when new business rules and requirements are introduced, which is the main benefit of Data Vault modelling, (iii) to help Data Warehouse modelers of the company in their ongoing Data Warehouse implementation, (iv) to improve the performance of Data Warehouse loading processes, by taking advantage of Data Vault 2.0 recommended practices, (v) presenting an easy to use model that can be easily improved in the future and fully implemented, (vi) serving as a proof of concept for the use of Data Vault 2.0 in a Data Warehouse that is fed from a Data Lake.

5 DESIGN AND DEVELOPMENT

This section constitutes the Design and Development phase of the DSR Methodology. First, the Data Lake architecture in which the Data Vault 2.0 model will be integrated is explained, based on the findings of the Systematic Literature Review. Then, the tools and metadata used to design the Data Vault 2.0 model are presented. Some theoretical concepts associated with the Data Vault 2.0 methodology are presented before introducing the model. Regarding the design of the model itself, all entities – hubs, links, satellites, and reference tables – are presented and later integrated into the full, complete model.

5.1 DATA LAKE AS A SOURCE FOR THE DATA WAREHOUSE

From the findings of the Systematic Literature Review, we were able to define an architecture for the Data Lake zones to be implemented, together with domain experts of the company. As we have seen, at least 3 zones are required in a Data Lake: a zone for persistent raw storage, a structured/standardized zone, and a processed/aggregated zone. Because the Data Lake will be integrated with a Data Warehouse based on Data Vault 2.0, the processed/aggregated zone won't be needed, as the Data Vault is already responsible for implementing a structure for the cleaned and transformed data, enriched with business logic. When Data Vault is used to model the DW, the Data Lake mostly serves as a repository for raw storage ("An Efficient Data Lake Structure," 2019; Arnold, 2021; Olschimke, 2022).

The Data Lake itself will have 3 zones: Landing, Raw and Structured. The Landing zone will ingest and temporarily store the data in raw format, the Raw zone will store the data from Landing persistently, and in the Structured zone the files will be transformed into different formats that will optimize the cleaning process. Then, two zones that serve as the Data Warehouse will be implemented: Enriched and Processed. The Enriched zone will store data enriched with business logic in a Data Vault, and the Processed zone will support the implementation of Data Mart models to serve applications. Figure 6 showcases the conceptual model of the implemented Data Lake architecture in the company.

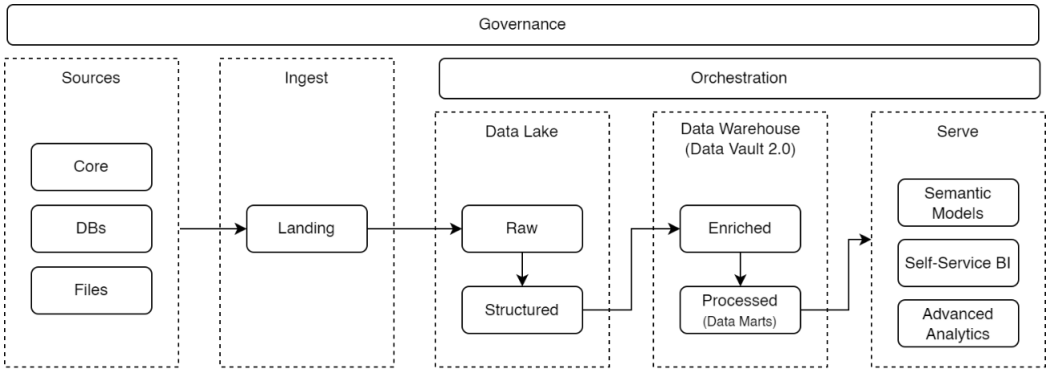

Figure 6 - Generic model of the implemented Data Lake architecture

As demonstrated by the literature review, there are no practical guidelines for Data Warehouse modelling, using Data Vault 2.0, when the data is sourced from a Data Lake, as long as the Data Vault 2.0 guidelines are respected. Additionally, regarding the research on Delta Lake layers used in this type of architecture, we concluded that, although there are some implementations documented, it does not make sense in this case, as the Data Warehouse based in Data Vault 2.0 is, theoretically, able to provide all of the capabilities of a Delta Lake layer. The purpose of the Delta Lake, as we have seen, is to provide Data Warehouse-like capabilities to the Data Lake, transforming the Data Lake into an almost "all-in-one" solution. However, as the Data Vault 2.0 model will be implemented in the Enriched zone of the Data Lake, transforming the data into Delta tables before loading it into the DW just adds unnecessary complexity and is, as we have seen in the literature review, redundant.

5.2 Metadata and Tools

In order to develop the Data Vault model, metadata from the company's raw data was consulted, specifically regarding clients, accounts, cards, and other associated tables. Transactional data was occasionally consulted on premises, and meetings with data experts from the information management division were needed to better understand the business, the sources and limitations that needed to be reflected on the model. The Data Vault model was designed following the guidelines of the Data Vault 2.0 methodology and the book by Linstedt & Olschimke (2016).

The first step was to extract metadata from the available applications and data dictionaries, using *SAS Enterprise Guide*, and export it to *Microsoft Excel*. With information about the tables, respective columns, their sizes and descriptions, business keys, etc., it was possible to start designing the model. Weekly meetings with the team also provided useful insights into the business and allowed us to explore the actual data without having direct access.

The software used to design the model was *PowerDesigner* (Wang, 2020), as it supports multiple types of diagrams and modelling languages, having the necessary tools for each. It also allows the generation of an SQL script directly from the physical diagram of a relational model (containing all primary and foreign keys, and relationships between tables), which can create all tables, relationships, constraints, procedures, and so on, in a database just by running that script.

The focus of this document will be primarily on individual customers, enterprise customers, credit card accounts and respective relationships with the customers and credit cards. The relationships between customers and current accounts were the focus of another team member's work, thus will not be as thoroughly explained.

5.3 Data Vault Entities and Hash Keys

A Data Vault 2.0 model, at its core, is based on three types of tables or entities: Hubs, Links and Satellites.

Source systems refer to business objects using a business key (BK) or a combination of business keys. These business keys are crucial to the business because it is through them that we can search for any

business object and obtain a unique result. For that reason, in Data Vault 2.0, they are separated from the rest of the attributes and stored in tables that only store the business key(s) of one business object – the Hub (Linstedt & Olschimke, 2016, p. 91).

Business objects do not exist alone and are always connected to other business objects through business processes that traverse multiple areas of the company. In Data Vault 2.0, this connection is always represented as its own table – a Link –, which establishes an association between two or more business keys (or hubs) (Linstedt & Olschimke, 2016, p. 91).

Having the BKs stored in hubs and the connections between them in links is not enough, as business objects and relationships between business objects need to have information that gives them context. This information is stored in satellite entities, which apart from storing the descriptive attributes of hub and link entities, also store their history by adding each change as a new row (Linstedt & Olschimke, 2016, p. 93).

One thing that all of the entities mentioned above have in common is a unique identifier of the table – hash key (HK) –, which is generated from the business key(s) of the table using a hash algorithm, such as MD5. The hash key of a hub is its primary key (PK) and is a foreign key in its connected Links and Satellites (Linstedt & Olschimke, 2016, p. 98). Apart from the hash keys, all Data Vault entities contain other relevant metadata attributes, namely: record source (stores the source system of the data), and load timestamp (stores the date and time at which the data was loaded into the table).

Because Data Vault 2.0 requires more tables than, for example, a relational model, the performance of joins needs to be optimized as much as possible. When checking if a business key already exists in a Hub, the *lookup* operation can be much slower if the BKs have varying lengths. The hash key solves this problem as it has a fixed length and is usually shorter in the case of long BKs (Linstedt & Olschimke, 2016, p. 98).

5.4 HUBS DEFINITION

The first step when modelling in Data Vault is to define the Core Business Concepts of the company, which will be represented as Hub tables (Linstedt & Olschimke, 2016, p. 93). Each hub corresponds to a business concept and stores, among other standard attributes, only the business key(s) associated with that concept and a fixed-length hash key generated from the business key(s). The hubs also contain a record source attribute that stores the source system of the data and a load timestamp, to record the date and time at which that piece of data was first loaded into the hub (Linstedt & Olschimke, 2016, p. 99). All Hub tables will be represented in the color blue and prefixed by “H_” for consistency and readability.

The hubs that were defined according to the available metadata were:

- H_CUSTOMER
- H_CURRENT_ACCOUNT
- H_CREDIT_CARD_ACCOUNT
- H_CREDIT_CARD
- H_BENEFICIARIES

The company has 3 different types of customers, but all of them come from the same system with the same business key. Although they have unique characteristics and are part of different processes, we decided to keep them all in the same hub – H_CUSTOMER –, as the BKs are the same and they represent the same core business concept. In the future, with the addition of other business concepts and relationships, it might make sense to separate them. Currently, each customer, regardless of their type, is identified by a unique *customer_number*.

Regarding the chosen accounts to model, they were separated into two different hubs – H_CURRENT_ACCOUNT (for current accounts) and H_CREDIT_CARD_ACCOUNT (for credit card accounts) –, because each of them has specific relationships with certain business concepts that the other may not have. They are substantially different, especially if we consider future improvements of the model, which further justifies this decision. Each account, regardless of their type, is identified by 4 fields – product_code (identifies the type of account), *branch_code* (where the account was opened), *account_number* and *check_digit* (security code for the account) –, which are separated into 4 different columns from the source, forming a composite key. However, in practice, the business does not use these 4 fields separated to identify each account. Instead, a string is used that concatenates the 4 columns, separating the fields with "." – *account_id* –, which is also present in the hub and serves as a unique BK as well (Linstedt & Olschimke, 2016, p. 99). This makes the "hashing" process of the composite BK easier, as it is stored in a single field (*account_id*).

The H_CREDIT_CARD hub stores the business keys of the physical credit cards that are associated with the corresponding credit card accounts. The only business key of this hub is the *card_number*.

H_BENEFICIARIES is a "keyed-instance hub". This concept is not explored in the Data Vault 2.0 official documentation, and it is not part of the recommended practices in (Linstedt & Olschimke, 2016). However, it has been used and implemented in practical, industry cases by Data Vault 2.0 certified practitioners, and it available in Data Vault certification courses (Hultgren, 2015). This concept emerged due to the need to avoid link-on-link structures in Data Vault models. Link-on-link structures exist when two Links are directly connected and are dependent on one another. As expected, this does not scale well and requires exponential work in loading efforts (Linstedt & Olschimke, 2016, p. 128). The "keyed-instance hub" solves this issue by creating a hub entity that has a 1:1 relationship with a Link and inheriting its primary key. This way, when contextual information (satellites) or other Links need to be connected to this Link, meaning that it is an important focal point of the business, all of that can be done though this new Hub. It is essentially an instance of the link.

In this case, H_BENEFICIARIES was necessary because the Link it was “generated” from had a dependent child key (Linstedt & Olschimke, 2016, p. 111), which is an additional key that is necessary to uniquely identify each record of the relationship (this concept will be exemplified later on), and it needed to be connected to another link. Because link-on-link structures are not recommended, with the “keyed-instance hub” it is possible to “materialize” the relationship in a hub and connect it to any entities (Figure 7).

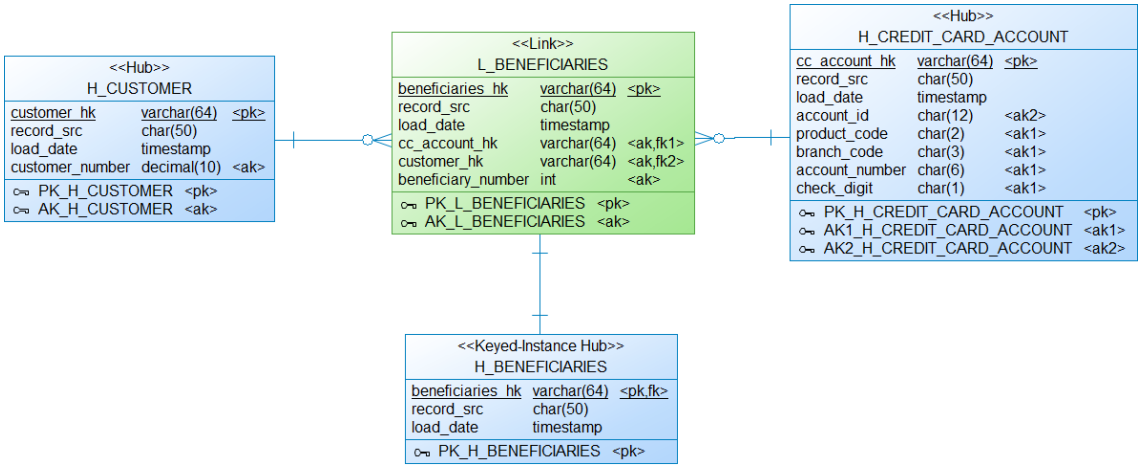

Figure 7 - Beneficiaries Keyed-Instance Hub

All hubs with respective hash keys, business key(s), metadata attributes, column data types, and primary keys are represented in Figure 8. The alternate keys (AK) identified below represent the attributes or group of attributes that can also uniquely identify each record of the hub. In this case, all AKs are business keys of the hubs.

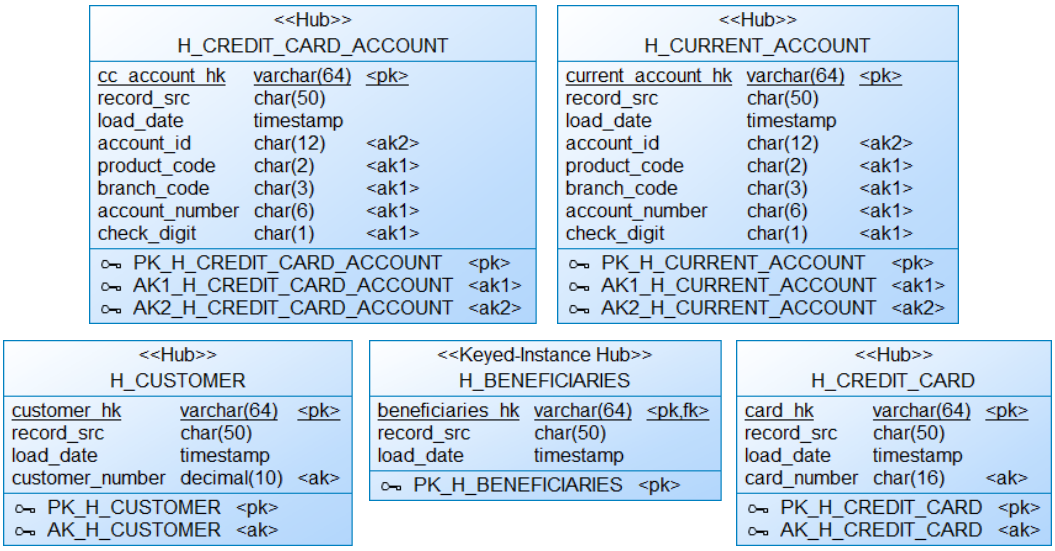

Figure 8 - Hubs with respective attributes, column data types and keys

5.5 LINKS DEFINITION

The Link tables establish relationships between the core business concepts (hubs) by joining their hash keys in a table that corresponds to a "many-to-many" relationship. This way, if the business changes and a "one-to-many" relationship is now a "many-to-many" relationship, the model can accommodate that without the need to reengineer the model and add a new table, because a link between two tables is always represented as its own table in Data Vault, regardless of the type of relationship (Linstedt & Olschimke, 2016, p. 105). Link tables are identified by their own hash key, which is generated from the combination of the business keys, not hash keys, of the referenced hubs. The hash key reduces the number of joins needed during ETL jobs, because the hubs do not need to be consulted, as the Link hash key already comprises all of their BKs combined (Linstedt & Olschimke, 2016, p. 111). Apart from the hash key, the Link table also contains the record source and load timestamp attributes, similarly to the Hub tables, and may contain other attributes such as a "dependent child key", which is needed in combination with the other keys of the referenced Hubs in order to uniquely identify each record of the relationship and define its grain and is also part of the link's primary key (Linstedt & Olschimke, 2016, p. 111). All Link tables will be represented in the color green and prefixed by "L_" for consistency and readability.

There are many types of links in Data Vault with different purposes, such as same-as links (for business objects that can be identified by more than one business key), hierarchical links (to model parent-child relationships between objects), non-historized links (for transactional data that cannot be updated), non-descriptive links (links without satellites connected to them storing descriptive data), and so on. These specific links allow for a better fit between the model and the actual business.

The link tables that were modeled based on the available metadata were:

- L_CUSTOMER_RELATIONSHIP
- L_CUSTOMER_CURRENT_ACCOUNT
- L_CUSTOMER_CREDIT_CARD_ACCOUNT
- L_BENEFICIARIES
- L_BENEFICIARIES_CURRENT_ACCOUNT
- L_CREDIT_CARD_ACCOUNT_CARD

Because this work will be mainly focused on credit card accounts and associated tables, only the relationships between customers and credit card accounts and between credit card accounts and credit cards will be further explained – L_CUSTOMER_CREDIT_CARD_ACCOUNT, L_BENEFICIARIES, L_BENEFICIARIES_CURRENT_ACCOUNT, L_CREDIT_CARD_ACCOUNT_CARD.

Both the current accounts and the credit card accounts have an important relationship with the customer. These relationship between customers and credit card accounts is represented by the L_CUSTOMER_CREDIT_CARD_ACCOUNT link. Its hash key is derived from the business keys of the referenced hubs – *customer_number* and *account_id* –, and contextual information on this relationship is stored in satellites connected to the link.

L_CREDIT_CARD_ACCOUNT_CARD is the link table that establishes the relationship between a credit card account and the physical credit card(s) associated with it. Its hash key joins the business keys of

H_CREDIT_CARD_ACCOUNT and H_CREDIT_CARD – *account_id* and *card_number* –, and it also contains the standard attributes “record source” and “load timestamp”.

Because not all beneficiaries of credit cards are customers of the bank, the link that associates a customer with a credit card account also needs to have another key, a “dependent child key”, so that the same customer-account pair can have multiple beneficiaries (e.g., family members of the customer that is the owner of the account) (Linstedt & Olschimke, 2016, p. 111). This link, L_BENEFICIARIES, apart from the record source and timestamp, contains a hash key that is calculated from the aggregation of the business keys of the referenced hubs – *customer_number* and *account_id* –, and the dependent child key – *beneficiary_number* –, which indicates the number of the beneficiary in the account (first, second, third, etc.).

Current accounts are the most important type of accounts in the bank. All other accounts, such as credit card or loan accounts, need to be connected to a current account. For example, when a customer makes a purchase using their credit card, the amount is charged to their credit card account. In the end of the billing cycle, when a statement with all performed transactions is issued, the customer can settle the credit card balance by making a payment directly from their current account. In this case, because a credit card account can have multiple beneficiaries, the previously mentioned relationship happens between the L_BENEFICIARIES link, which already contains information on the customer, credit card account and associated beneficiaries, and the H_CURRENT_ACCOUNT hub. Introducing a link between these two entities would create a link-on-link structure, which we want to avoid, because it presents multiple performance problems, as discussed previously. Instead, a link between the H_BENEFICIARIES “keyed-instance hub” (explained in subsection 7.2) and H_CURRENT_ACCOUNT is established – L_BENEFICIARIES_CURRENT ACCOUNT (Figure 9). The hash key of this link will be composed by the following keys: *customer_number*, *account_id* (of the credit card account), *beneficiary_number*, *account_id* (of the current account).

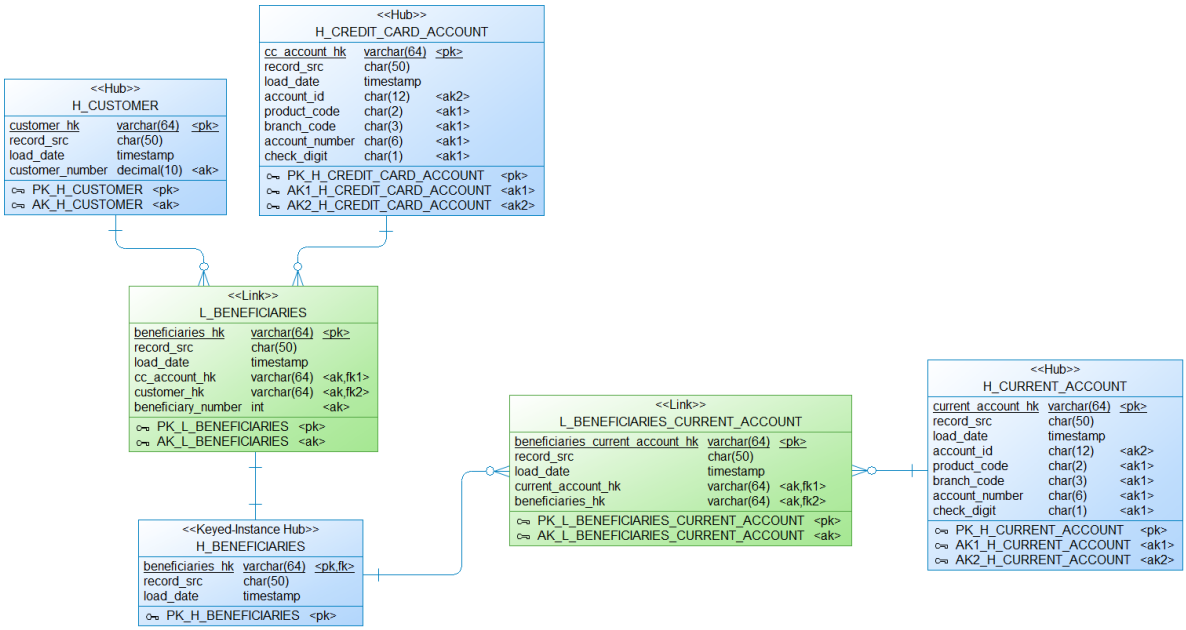

Figure 9 - Link between Beneficiaries Keyed-Instance Hub and Current Accounts Hub

All links with respective hash keys, metadata attributes, column data types, foreign keys of the referenced hubs, and dependent child keys (if applicable) are represented in Figure 10. The alternate keys identified below represent the attributes or group of attributes that can also uniquely identify each record of the hub. In this case, all AKs are the group of foreign keys of the referenced hubs.

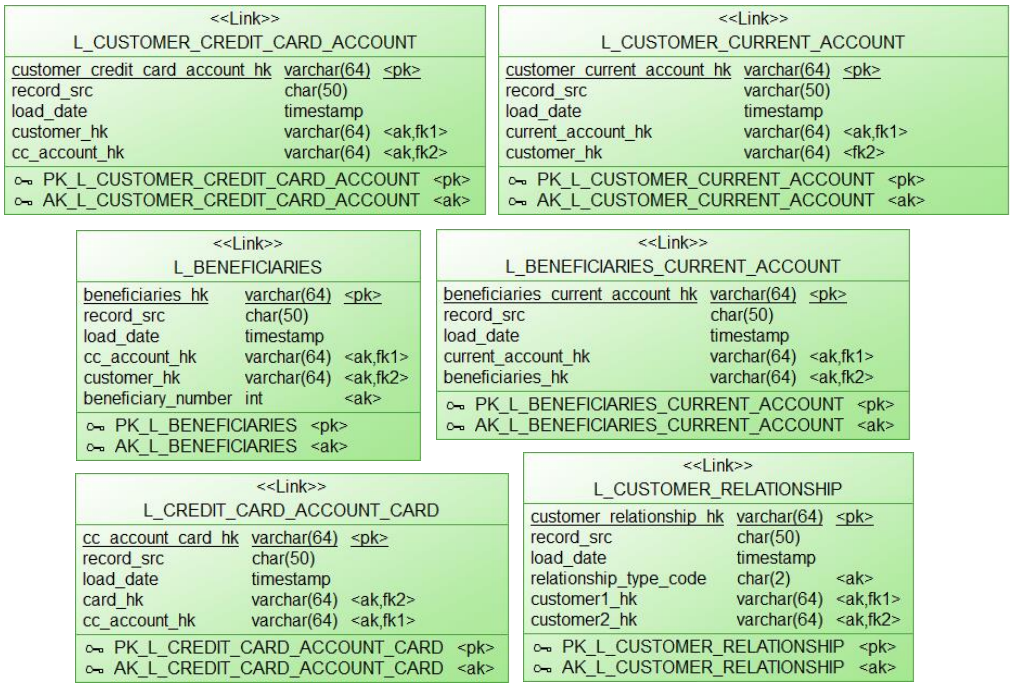

Figure 10 - Links with respective attributes, column data types and keys

5.6 SATELLITES DEFINITION

Satellite tables store descriptive information associated with business objects, which can be Hubs or Links. They keep track of changes over time on that descriptive data as well (Linstedt & Olschimke, 2016, p. 112). Because they are dependent on their parent Hub or Link, their primary key is composed by the key(s) of the associated Hub or Link together with a load timestamp, to track change (Linstedt & Olschimke, 2016, p. 116). Additionally, they should always have a load end date, to indicate when the records become invalid, and a record source attribute. They can also have two optional fields: a "hash difference" attribute, which is a hashed value of all data inside the satellite to help compare records when loading, and an "extract date", which stores the date in which the data was extracted from the source (Linstedt & Olschimke, 2016, p. 117). In this case, we decided to include the "hash difference" attribute in all satellites, as it speeds up the process of loading new records. All Satellite tables will be represented in the color yellow and prefixed by "S_" for consistency and readability.

Usually, Satellite data is split according to its rate of change (e.g., changed daily, monthly, rarely, etc.) or according to the source system (Linstedt & Olschimke, 2016, p. 114). In this case, the business has multiple objects which share the same business key from the same system but have substantially different attributes (e.g., customers), so splitting by type/functionality of business object is suggested (Linstedt & Olschimke, 2016, p. 93). For example, if we keep descriptive data of enterprise customers

in a dedicated satellite, separated from private customers data, when we load new customers of this type, we only have to load the hub and the respective satellite, instead of multiple satellites.

There are multiple types of satellites such as multi-active satellites (multiple active records at the same time for the same business object), status tracking satellites, effectivity satellites (to track if a relationship between two objects is active or not), and so on. The satellites present on the complete model are:

- S_CUSTOMER_BASICS
- S_CUSTOMER_ENTERPRISE
- S_CUSTOMER_PRIVATE
- S_CUSTOMER_RELATIONSHIP
- S_CURRENT_ACCOUNT_STATIC
- S_CURRENT_ACCOUNT_BALANCE
- S_CURRENT_ACCOUNT_CHECKS
- S_CREDIT_CARD_ACCOUNT
- S_CREDIT_CARD_ACCOUNT_STATIC
- S_CREDIT_CARD
- S_CUSTOMER_CURRENT_ACCOUNT
- MAS_CURRENT_ACCOUNT_OWNERSHIP
- S_CUSTOMER_CREDIT_CARD_ACCOUNT
- MAS_CREDIT_CARD_ACCOUNT_OWNERSHIP
- S_BENEFICIARIES
- S_BENEFICIARIES_CURRENT_ACCOUNT

As previously, the focus will be on satellites associated with customers, credit card accounts and credit cards.

The H_CUSTOMER hub, which stores the business keys of all customers in the company, has three satellites connected to it: S_CUSTOMER_BASICS, S_CUSTOMER_ENTERPRISE and S_CUSTOMER_PRIVATE. S_CUSTOMER_BASICS contains all common attributes of private and enterprise customers, such as name, document information, address information, professional information, and so on. S_CUSTOMER_ENTERPRISE contains the attributes specific to enterprise customers, such as establishment date, company type, contact information, and others. S_CUSTOMER_PRIVATE contains all information specific to private or individual customers, such as birth date, education level, job information, sex, and others. All of them have a primary key composed by the hash key of the H_CUSTOMERS hub (*customer_hk*) and the load timestamp attribute (Figure 11).

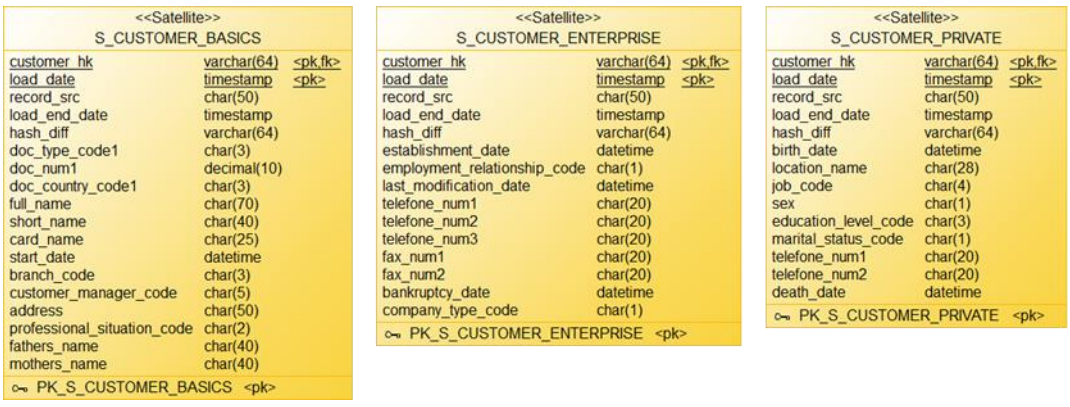

Figure 11 - Satellites of the Customer Hub

The H_CREDIT_CARD_ACCOUNT hub has two satellites: S_CREDIT_CARD_ACCOUNT_STATIC and S_CREDIT_CARD_ACCOUNT. S_CREDIT_CARD_ACCOUNT_STATIC stores the attributes of the credit card account that are rarely or never changed, such as the card brand, the product code of the account, authorization insertion date, and others. This way, when information regarding the other attributes needs to be changed, this satellite does not need to be accessed, saving time. S_CREDIT_CARD_ACCOUNT stores all other attributes that are frequently or sometimes changed, such as the number of beneficiaries, the payment method, the credit limit, the cancelation date, and so on. The primary key of both satellites is composed by the hash key of the H_CREDIT_CARD_ACCOUNT hub (*cc_account_hk*) together with the load timestamp attribute (Figure 12).

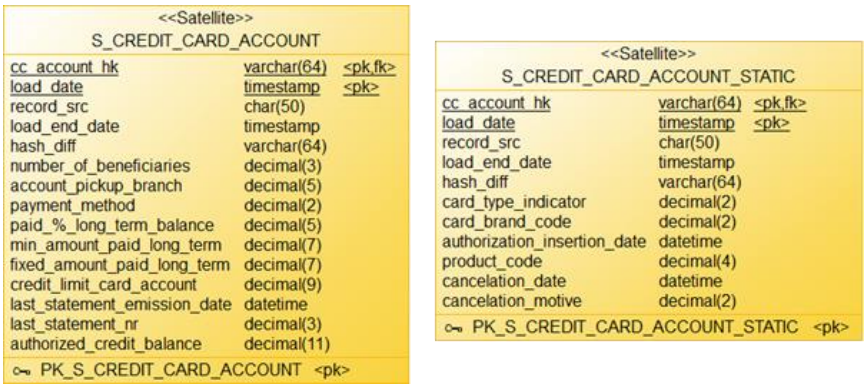

Figure 12 - Satellites of the Credit Card Account Hub

S_CREDIT_CARD is the satellite that is connected to the credit card hub (H_CREDIT_CARD). It stores information about the physical credit cards, such as number, validity date, where it will be shipped to, and so on. Its primary key consists of the H_CREDIT_CARD hash key (*card_hk*) and the load timestamp (Figure 13).

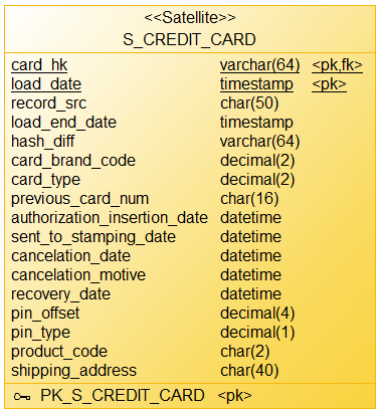

Figure 13 - Satellite of the Credit Card Hub

The relationship between customers and credit card accounts, represented by L_CUSTOMER_CREDIT_CARD_ACCOUNT, has two satellites: S_CUSTOMER_CREDIT_CARD_ACCOUNT and MAS_CREDIT_CARD_ACCOUNT_OWNERSHIP (Figure 14).

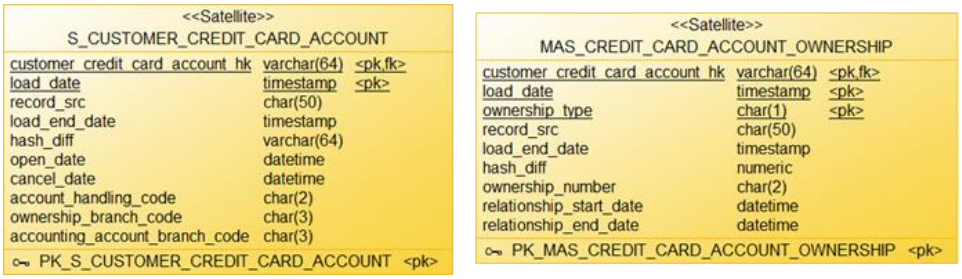

Figure 14 - Satellites of the Link between Customers and Credit Card Accounts

S_CUSTOMER_CREDIT_CARD_ACCOUNT stores information about the relationship between the customers and their respective credit card accounts, such as the opening date of the account, the branch where the account was opened, the cancelation date, etc. Its primary key is composed by the hash key of the L_CUSTOMER_CREDIT_CARD_ACCOUNT link (*customer_credit_card_account_hk*) and the load timestamp. MAS_CREDIT_CARD_ACCOUNT_OWNERSHIP is a multi-active satellite (Linstedt & Olschimke, 2016, p. 141), which allows us to store more than one record at a time per customer and credit card account pair, since the same customer may have different types of ownership of the same account at the same time (e.g., owner, tutor). To do this, its unique identifier is composed by three fields: the hash key inherited from the associated link table (L_CUSTOMER_CREDIT_CARD_ACCOUNT), the load timestamp and an additional *ownership_type* key, which differentiates the different types of ownership a customer can have in the same account, changing the grain of the data. Apart from this, an *ownership_number* field is needed because when the *ownership_type* is “owner”, we need to specify if it is the first, second, third owner or other. Apart from these attributes, there are also *relationship_start_date* and *relationship_end_date* attributes, as we need to know when each ownership relationship started and ended. This satellite also exists for the link between customers and current accounts (L_CUSTOMER_CURRENT_ACCOUNT) and any other accounts that will be added to the model in the future, as all of them require ownership information.

The S_BENEFICIARIES satellite (Figure 15) is attached to the H_BENEFICIARIES hub, which is a “keyed-instance hub”. It stores personal information about the beneficiaries of the credit card accounts (even if they are not customers of the bank), cancelation date, balance information, etc.

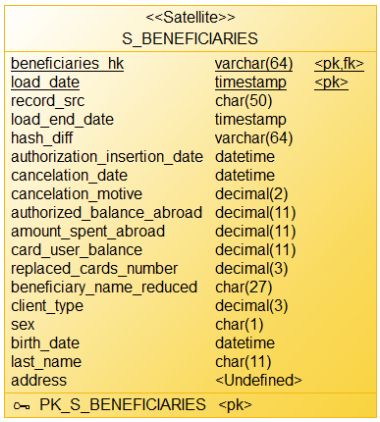

Figure 15 - Satellite of the beneficiaries Hub

L_BENEFICIARIES_CURRENT_ACCOUNT, which establishes the relationship between beneficiaries of credit card accounts, the actual credit card accounts, and the associated current account, has one satellite: S_BENEFICIARIES_CURRENT_ACCOUNT. This satellite contains auxiliary information, such as the insurance plan of the beneficiary, temporary card limits, fees associated with the account, and so on. Its primary key is composed by the hash key of the associated link (*beneficiaries_current_account_hk*), as well as a load timestamp (Figure 16).

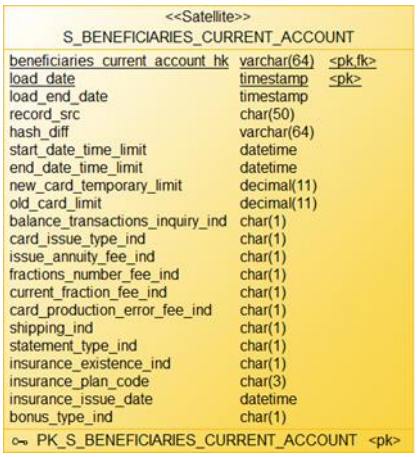

Figure 16 - Satellite of the Link between Beneficiaries and Current Accounts

All satellites also contain the record source (*record_src*), load end date (*load_end_date*) and hash difference (hash_diff) attributes, apart from the ones that compose the primary key and were already mentioned for each specific case. Some attributes given by the metadata were omitted in the satellites for simplicity.

The complete set of all satellites in the model, including those concerning current accounts and relationships between customers, can be found in Appendix A.

5.7 REFERENCE TABLES DEFINITION

Apart from the main tables of the core architecture – Hubs, Links and Satellites –, another type of table can also be implemented to store information about objects that are needed for the business but do not constitute a core business concept themselves – Reference Tables (Linstedt & Olschimke, 2016, p. 160). They do not store historical data and their purpose is to give context to other business keys in the model. They also do not have a hash key, instead, Reference Tables only have a descriptive business key as PK and other simple attributes, such as a brief description of the objects to be stored (Linstedt & Olschimke, 2016, p. 162).

In this case, we need Reference Tables for every "code" attribute stored in the model (e.g., country codes, product codes, branch codes) that does not yet exist as a hub on its own. However, in the future, with the expansion of the model, some of this reference tables will most likely turn into hubs, as they are more than just "codes" in other contexts (e.g., products and branches). In Figure 17, two reference tables are presented to provide a description for the *product_code* and *branch_code* attributes, which are part of the composite keys of current and credit card accounts.

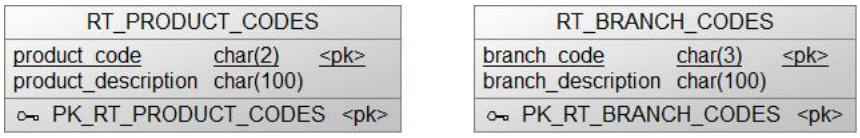

Figure 17 - Reference Tables for product codes and branch codes

Figure 18 constitutes the full proposed Data Vault 2.0 model, including all entities described in this section, with all respective attributes, primary and foreign keys, alternative keys, column data types according to the metadata, and relationships between tables.

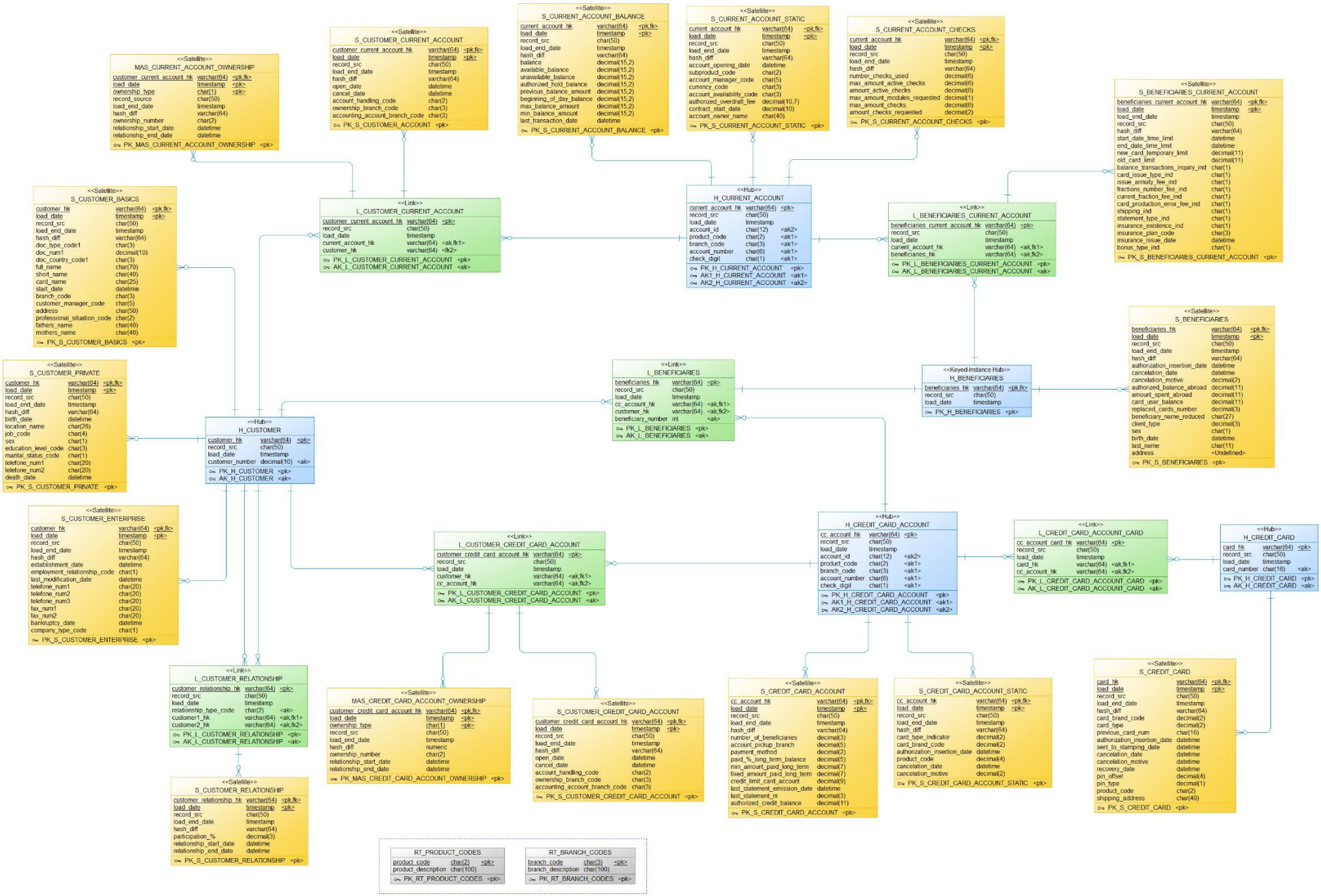

Figure 18 - Proposed Data Vault 2.0 model

6 DEMONSTRATION

This section presents the Demonstration phase of the DSR Methodology. In this phase, we aim to prove that the solution works, before conducting a formal evaluation, by demonstrating how the model can adapt to a set of new business requirements introduced by the company and this way, solve hypothetical problems (Peffers et al., 2007, p. 55). The solution for these problems will be demonstrated by changing the diagram of the Data Vault model to incorporate the requirements. This way, we intend to show how a Data Vault model can easily adapt to changes without the need to restructure the current model. These requirements were defined in collaboration with employees of the company and some of them are based on real problems and scenarios.

6.1 REQUIREMENT 1: NEW SOURCE OF CUSTOMERS

The current model is being loaded with data from one unique source – the newly implemented Data Lake –, which has replaced the legacy Operational Data Store. However, there is another bank that belongs to the same enterprise group, fully dedicated to enterprise customers, which has not been integrated into the Data Lake yet. Since the Data Lake's implementation is still ongoing, and there is a need to make it operational as soon as possible, some source systems, such as this one, which do not receive new data as fast as the main sources, will only be integrated later. While it is not integrated into the Data Lake, this company's customer data needs to be loaded into the Data Warehouse for analysis and to produce reports for the Marketing Department. For that reason, the Data Warehouse will temporarily have two sources – the Data Lake and the other bank's transactional system.

Requirement 1: New enterprise customers of another company belonging to the same enterprise group, which are also uniquely identified by a customer number, need to be loaded into the model and integrated with the existing customers.

Proposed Solution: Modeling data from different sources in the same hub is not a problem in Data Vault 2.0, as the *record_source* attribute, present in all hubs, links and satellites, stores information on the original source of the data, allowing us to distinguish which records came from which source. Because the customers in the new source are also identified by a *customer_number* BK with values that do not overlap with the customer numbers from the current source, it is possible to load them all into the same hub – H_CUSTOMER. However, the same customers may be clients of both banks, in which case they will have two different customer numbers, despite corresponding to the same entity. Data Vault 2.0 has a way to map the business keys of one source to the other, this way identifying the BKs that correspond to the same customer, using a Same-As Link (Linstedt & Olschimke, 2016, p. 125). The Same-As Link is attached to the customer hub, referencing it two times – each foreign key obtained from the hub corresponds to the customer hash key of one source (Linstedt & Olschimke, 2016, p. 129). The hash keys in the column *BaseCustomer_hk* correspond to the customers from the main source – the Data Lake –, and the hash keys in *OtherSourceCustomer_hk* correspond to the customers from the new source system. A mapping between the two sources should be generated, based on the Taxpayer Identification Numbers ("NIF – Número de Identificação Fiscal" in Portugal), which identifies each customer uniquely, even if they have two different business keys in each

source. This way, by consulting the SAL_CUSTOMER same-as link, it is possible to identify any customer business key of the new system that corresponds to business key stored in the Data Lake (Figure 19).

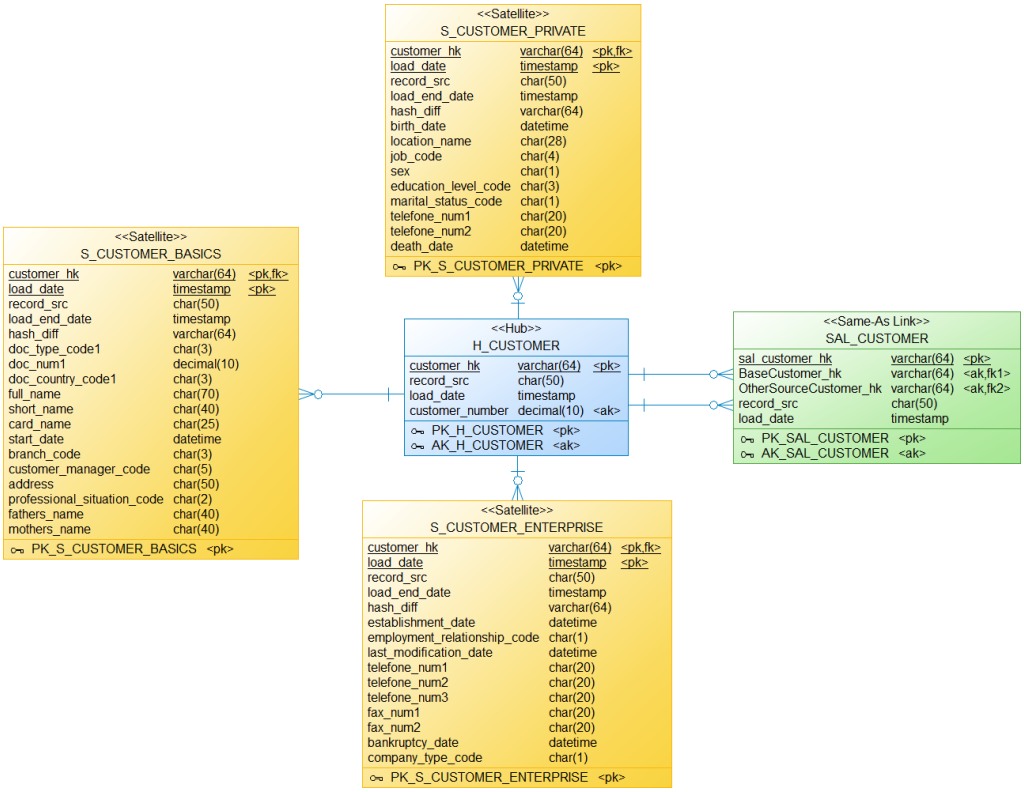

Figure 19 - Same-As Link for the Customer Hub

6.2 Requirement 2: New Term Deposit Account

The company is looking the expand the current model and add a new type of account that is also widely used by customers who are interested in investing their money. The term deposit accounts are a safe option for investment because the deposited funds are protected by the bank and the customer receives a fixed interest rate on the deposit for the entire duration of the deposit term. They are commonly used for short to medium-term savings goals, where the customer knows exactly how much interest they will earn over a specific timeframe, and that the funds are safe and growing steadily until the money is needed. The term deposit account, apart from being a new business concept to be introduced into the model, has relationships with the already modelled concepts, such as the customers and the current accounts. The customer may use a current account to transfer the funds into the term deposit account, to credit the interest earned or even to transfer funds from the term deposit account into the current account if, for example, unexpected expenses arise.

Requirement 2: A new term deposit account needs to be integrated to the model and linked to the appropriate existing business concepts.

Proposed Solution: Similarly to the other modelled accounts, the term deposit account constitutes its own business concept and has a unique key composed by the same 4 fields – *product_code*,

branch_code, *account_number* and *check_digit* –, which are concatenated into one field, apart from existing separately – the *account_id*. This way, it will be modelled as a hub – H_TERM_DEPOSIT_ACCOUNT. Because it has many descriptive attributes and some are altered more frequently than others, they were separated into two different satellites, one for static information (mainly dates and information that is only loaded once, when the account is created) – S_TERM_DEPOSIT_ACCOUNT_STATIC –, and other for information that is changed rather frequently – S_TERM_DEPOSIT_ACCOUNT. In terms of the relationship with the customers, a new link was added – L_CUSTOMER_TERM_DEPOSIT_ACCOUNT – to represent the usual relationship between a customer and any type of account, containing two satellites connected to it. S_CUSTOMER_TERM_DEPOSIT_ACCOUNT stores information on the opening and closing dates of the account, the owner branch code, etc. and MAS_TERM_DEPOSIT_ACCOUNT_OWNERSHIP contains information on the types of ownership a customer can have with an account. Because no other metadata was provided regarding the relationship between customers and term deposit accounts, this was the only link modelled for that relationship. In terms of the relationship with the current account, a new link was introduced – L_TERM_DEPOSIT_CURRENT_ACCOUNT – to represent the relationship between these two accounts. From the metadata provided, we could see that a term deposit account can be linked to different current accounts for different purposes. For example, one current account for creditor interest, other for withdrawals, other for support, and a main one. Because Data Vault models all relationships as "many-to-many", it is possible to model all of these relationships with different current accounts in the new link (Figure 20).

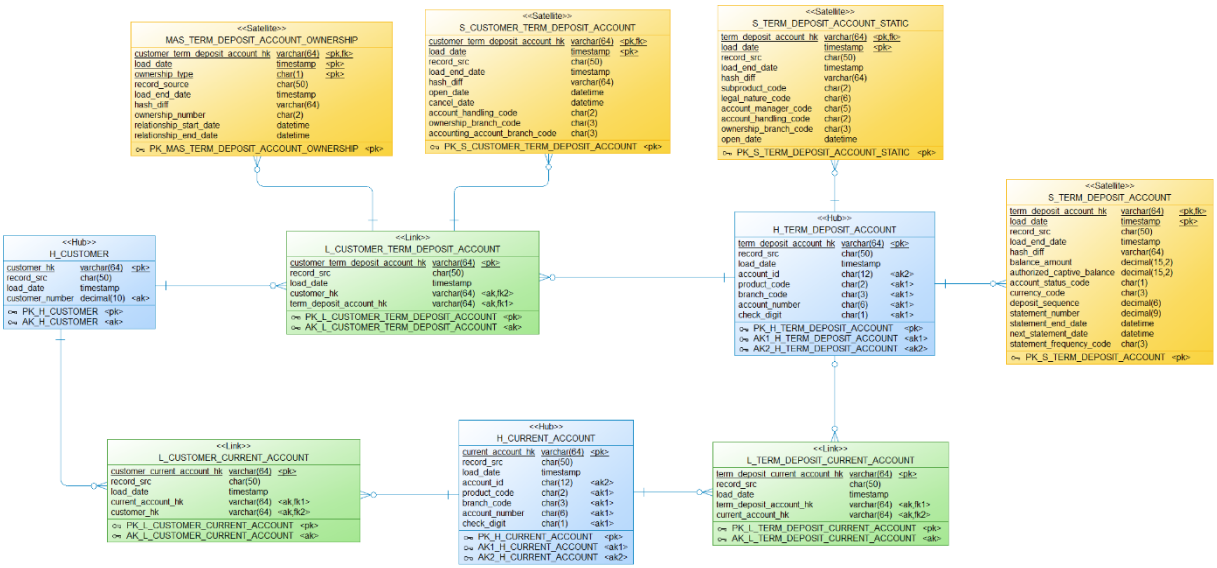

Figure 20 - New Term Deposit hub and link to the current accounts

6.3 REQUIREMENT 3: CUSTOMERS' RIGHT TO BE FORGOTTEN

The company wants to implement some measures to comply with the General Data Protection Regulation (GDPR). The first steps into this implementation include the integration of the Article 17 of the General Data Protection Regulation (Regulation (EU) 2016/679), which states that individuals have the "right to erasure" or "right to be forgotten" (European Parliament & Council, 2016), in the company's Data Warehouse. However, complete deletion of a customer from the company's Data

Warehouse is not feasible, as it would disrupt data integrity and defeat the purpose of the Data Warehouse, which is to record historical data of the company persistently. Apart from that, banks are subject to several legal and regulatory requirements that require them to keep data during, at least, a certain retention period, which means that they cannot simply be erased. Banks also undergo auditing and compliance assessments frequently, and receive requests of data from important national institutions, such as the court of justice. This means that, erasing data, especially without appropriate documentation and auditing, could raise problems in future regulatory inspections.

Requirement 3: The customers' right to erasure ("right to be forgotten"), according to the Article 17 of the General Data Protection Regulation (Regulation (EU) 2016/679), needs to be implemented in the Data Warehouse.

Proposed Solution: As mentioned previously, data cannot simply be erased from the Data Warehouse, so, for a first implementation of this regulation, it would not be appropriate to propose a full deletion of records. Instead, a new satellite appended to the customer hub – H_CUSTOMER – could be introduced to store all data and requests from the customers related to their rights, according to the GDPR – S_CUSTOMER_GDPR. In order to implement the "right to be forgotten", three records were introduced: a "flag" attribute, which can assume two values (0 or 1), to register if a customer asked for their right to be forgotten in the context of the company; a date for when this request was made; and a date for when the right was revoked. This allows the company to know which customers can be included in reports and other data products that are not subject to auditing and regulatory duties. Figure 21 shows a possible implementation of these attributes.

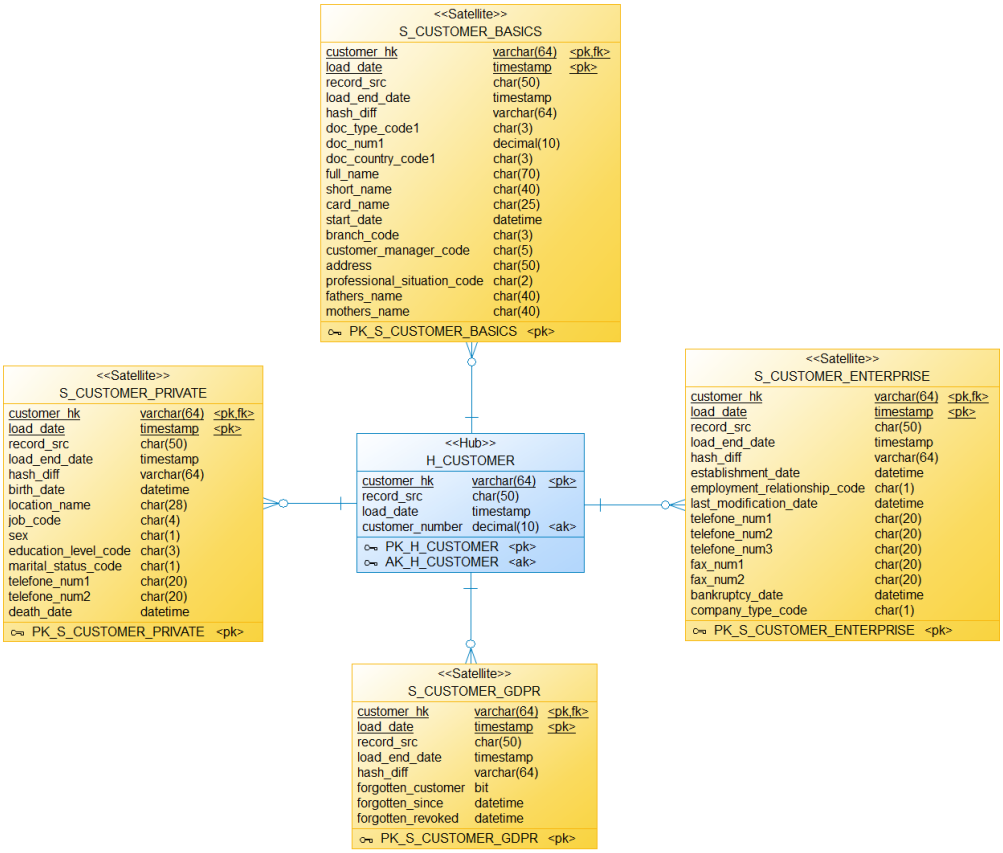

Figure 21 - Customer satellite for GDPR attributes

6.4 Requirement 4: Solving performance issues when joining Customers and Credit Cards

One of the biggest drawbacks of Data Vault modelling is the fact that the number of tables can grow substantially fast, as all relationships are modelled as tables, and the descriptive attributes are separated in multiple satellites. Although this isn't problematic for loading Data into the Data Warehouse, in terms of querying, it can represent big performance issues (Linstedt & Olschimke, 2016, p. 158). A situation where this problem can occur is when querying the Data Vault to consult the credit cards of a certain set of customers. In this case, multiple tables need to be joined, as customers are only directly linked to the credit card accounts. The credit card account is, in turn, linked to credit card(s). Even if the query is limited to certain customers, accounts, or cards by using the WHERE clause in the SQL statement, the five tables that need to be joined hold a huge number of records and, consequently, the query will take a lot of time to be performed.

Requirement 4: Query performance issues need to be addressed when consulting the credit cards of certain customers, at a given time.

Proposed Solution: Data Vault 2.0 presents a solution for this query performance problem, with the introduction of Bridge Tables, which are a part of the Business Vault, an intermediate layer on top of the Raw Vault, that represents an extension of the already modelled data with business rules applied (Linstedt & Olschimke, 2016, p. 28). The Bridge Table connects multiple tables – hubs or links –, containing all of their hash keys as a primary key (Linstedt & Olschimke, 2016, p. 158). Additionally, it has a "snapshot date" attribute containing the date on which the records were loaded, and optionally business keys of the tables it spans, as well as computed fields, for recurrent computations that usually take a lot of time (Linstedt & Olschimke, 2016, p. 159). In this case, the Bridge Table will contain the hash keys of the customer, credit card account and credit card hubs – H_CUSTOMER, H_CREDIT_CARD_ACCOUNT and H_CREDIT_CARD, respectively –, as well as the hash keys of the link between customer and credit card account and the link between credit card account and credit card – L_CUSTOMER_CREDIT_CARD_ACCOUNT and L_CREDIT_CARD_ACCOUNT_CARD, respectively. Additionally, it will contain the *snapshot_date* attribute to store the date on which each record was loaded into the table. This way, the five tables do not need to be consulted every time there is a need to map customers to their respective credit card (Figure 22). If this is a usual operation, the bridge table can be consulted instead.

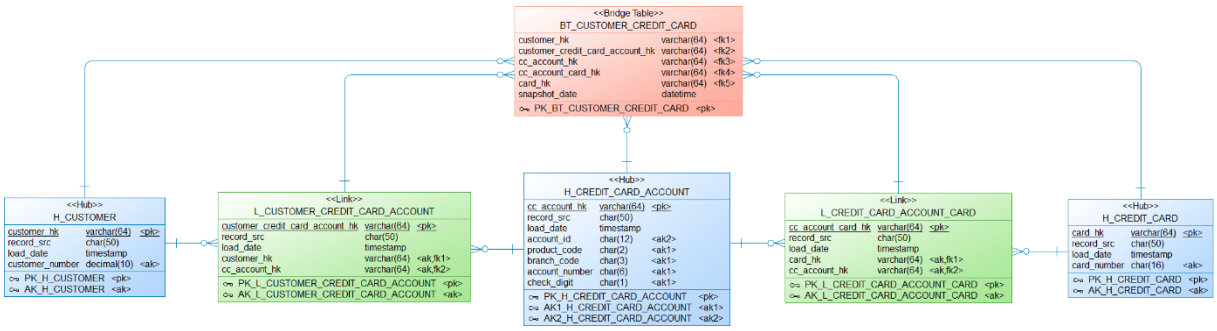

Figure 22 - Bridge Table for Customers and Credit Cards

6.5 Requirement 5: A beneficiary of a Credit Card Account can now be associated with multiple Current Accounts

Currently, the company only allows for a credit card account of a beneficiary to be associated with one current account. This current account is typically the one charged for the transactions made on the credit card account and the payment is settled by transferring funds from the current account to the credit card account or by setting up automatic payments at the end of each billing cycle. The company now wants to allow a beneficiary of a credit card account to be able to settle payments from more than one current account, to accommodate situations where the customer has a joint, family or business credit card account and may want to link it to many current accounts for cases where one of them doesn't have any available balance, so the funds should be transferred from the other. For example, the case where a certain beneficiary of a joint credit card account wants to have the credit card payments deducted from their own current account, but, if there is no available balance, their spouse's current account should be used.

Requirement 5: A beneficiary of a credit card account should now be able to be associated with more than one current account.

Proposed Solution: The fact that a beneficiary of a credit card account can now be associated with multiple current accounts implies that the relationship between H_BENEFICIARIES and H_CURRENT_ACCOUNT, which used to be "one-to-many", is now "many-to-many". This demonstrates one of the main benefits of Data Vault 2.0 regarding its flexibility, because all relationships are modelled as "many-to-many" regardless of their nature in the context of the business. The two hubs are already connected through a link table, representing a "many-to-many" relationship, so this change in the business logic has no impact in the existing structure of the Data Vault 2.0 model and does not violate any primary key constraints (Figure 23).

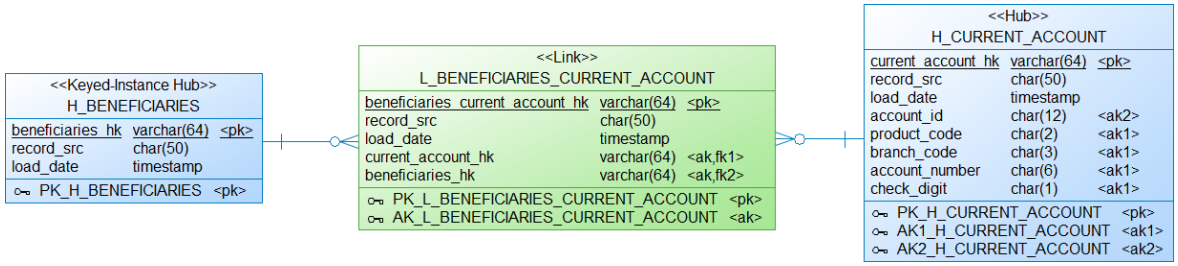

Figure 23 - Many-to-many relationship between Beneficiaries and Current Accounts

7 EVALUATION

The Evaluation phase of the DSR methodology aims to measure the effectiveness of the artifact in terms of solving the identified problem and fulfilling the objectives determined in the beginning.

To determine how to evaluate the model, the framework by Pries-Heje et al. (2008) was followed. It divides evaluation strategies in 4 quadrants by distinguishing between "ex ante" and "ex post", and artificial and naturalistic evaluation. "Ex ante" evaluations take place before the artifact is proposed (the artifact is evaluated based on its design specifications) and is used when the purpose of the evaluation is to determine whether to adopt or develop a certain technology. On the other hand, "Ex post" evaluation happens after the artifact is constructed. In terms of the setting in which the evaluation is performed, it can be artificial or naturalistic. While artificial evaluation uses an unreal setting, which can be translated in unreal users, systems or problems, the naturalistic approach evaluates a solution in its real environment. In this case, since we want to evaluate the adequateness of the model according to a company's requirements, we will follow an "ex post" and naturalistic approach.

Venable et al. (2012) proposes a DSR Evaluation Method Selection Framework, where evaluation methods for the 4 evaluation strategies (quadrants) are suggested (Table 27). For the "ex post" naturalistic approach, methods such as case studies, focus groups and surveys (quantitative or qualitative) are proposed. In this case, because we want to collect as much qualitative, insightful feedback as possible from practitioners, we will conduct interviews instead of surveys.

Table 27 - DSR Evaluation Method Selection Framework (Venable et al., 2012)

DSR EVALUATION METHOD SELECTION FRAMEWORK	EX ANTE	EX POST
NATURALISTIC	Action Research Focus Group	Action Research Case Study Focus Group Participant Observation Ethnography Phenomenology Survey (qualitative or quantitative)
ARTIFICIAL	Mathematical or Logical Proof Criteria-Based Evaluation Lab Experiment Computer Simulation	Mathematical or Logical Proof Lab Experiment Role Playing Simulation Computer Simulation Field Experiment

7.1 INTERVIEW PLANNING AND STRUCTURE

In this section, the structure of the interviews conducted as a part of this study is presented. The aim of these interviews is to gather rich data regarding the proposed artifact and provide valuable perspectives from people that work on the company's data and are familiar with the modelled business

concepts. The interviews were conducted in a semi-structured format. Although the participants are asked predefined questions, they are all open ended, encouraging an open discussion on the topics addressed. The interviewer can also perform follow-up questions and ask for clarification, enabling a fluid and spontaneous conversation (Galletta, 2013).

An interview script was carefully developed beforehand, to maintain consistency between interviews despite their semi-structured nature. The questions were designed to address certain specificities of the proposed model and how they relate to the business, and also gather an overall perception of the artifact's completeness, understandability, ease of use, robustness, pertinence, and utility, especially in the context of the organization. The full interview script can be found in Appendix B.

To ensure consistency and optimize the productivity of the interviews, all participants were also provided with a detailed documentation of the model beforehand, including descriptions of some parts of the model, Data Vault 2.0 concepts that were modelled, the tables from which metadata was extracted, and a demonstration of the flexibility of the model using the requirements and respective solutions presented in the last section. Nonetheless, the model and the provided materials were also presented in the beginning of each interview, to ensure the same level of understanding of the artifact between all participants.

All interviews were conducted via *Microsoft Teams*, due to location and time constraints. Each session was video recorded, using the platform's screen recording feature, as it is the most accurate form of capturing the conversations and other non-verbal cues. All participants were asked for consent to record the interview before it started. The interviews were transcribed, ensuring the anonymity of the participants and the confidentiality of company sensitive data, which was appropriately suppressed. Because the company is Portuguese and all the participants felt more comfortable being interviewed in Portuguese, the interviews were conducted in Portuguese. The full transcript of each interview can be found in the Appendixes section (Appendixes C, D, E, F, G, H, I). The interviews were transcribed in non-naturalistic language, omitting some interjections and confidential information, and in a linear scheme, following the guidelines provided in Azevedo et al. (2017).

The transcriptions were carefully analyzed to identify common patterns and themes, which were organized and summarized to provide the main insights gathered from each question, as well as the overall sentiment of the participants.

All respondents consented to participate in this study and their anonymity was ensured throughout the entire process. The personal data collected is not sufficient to uniquely identify any participant, all mentions to the company, its employees, data, or projects were carefully omitted, and the participants were identified by the letter "P", followed by the number of the interview, in the order they were conducted (e.g., the participant of the first interview is identified in the transcriptions as P1).

7.2 THE PARTICIPANTS

The respondents are employees of the company, in the data and analytics field, that work on and/or make use of the company's data and are familiar with the business. They were selected based on their knowledge and experience related to the company's data as well as its data sources.

After presenting the model in the beginning of the interviews and clarifying any doubts that may have arisen, the participants were asked a few introductory questions about themselves, which do not compromise their anonymity. The profiling of the respondents was made based on: (i) age, (ii) sex, (iii) job function, (iv) department/division, (v) years of experience in the banking industry, (vi) years of experience in the data and analytics field.

There were seven participants interviewed, four women and three men. The ages ranged between 37 and 56 years. All participants have several years of experience in the banking industry, between 9 and 32 years, with the average being 22 years of experience. In terms of experience in the data and analytics field, the years ranged between 2 and 29, with the average being 14 years of experience. The job function of the majority of the interviewees is "Technician", which is the formal designation inside the company for technical people, and there are multiple job levels for this function (level I, II and III). Two external consultants (developers) were also interviewed, as well as the Chief Data Officer. Regarding the field/department the participants work in, two of them work in the field of Data Governance, two in the "Projects" field, two in the Data Governance and Quality field and one in the Data Quality field. Table 28 describes the profile of all participants.

Table 28 - Profile of the Participants

PARTICIPANT #	AGE	SEX	JOB FUNCTION	FIELD/ DEPARTMENT	EXPERIENCE IN BANKING	EXPERIENCE IN THE DATA & ANALYTICS FIELD
P1	49	M	Technician level III	Data Quality	29 years	29 years
P2	55	F	Technician level II	Data Governance and Quality	12 years	19 years
P3	37	F	External Consultant – Analyst/Developer	Projects	9 years	9 years
P4	56	M	External Consultant – Analyst/Developer	Projects	32 years	15 years
P5	51	F	Technician level III	Data Governance	22 years	5 years
P6	48	F	Technician level II	Data Governance	25 years	2 years
P7	54	M	Chief Data Officer	Data Governance and Quality	23 years	18 years

7.3 INTERVIEW QUESTIONS

Based on the hierarchy of criteria for Information Systems artifact evaluation provided in Prat et al. (2014), five criteria of the environment, structure, and evolution dimensions were chosen to evaluate the artifact: (i) homomorphism (fidelity to modelled phenomena) , (ii) completeness, (iii) consistency with people (understandability, ease of use and utility), (iv) robustness, and (v) consistency with organization (fit with organization). Each criterion will have an associated question, meant to gather qualitative data about the artifact (Table 29). Two generic questions that don't correspond to any criteria were also added to gather additional comments and recommendations the participants may have.

All questions are open-ended and are meant to evoke natural conversations about each topic. The respondents are encouraged to elaborate and go beyond the main topic of each question.

Table 29 - Interview questions and criteria

#	CRITERIA	QUESTIONS
Q1	Homomorphism (Fidelity to modelled phenomena)	In terms of the beneficiaries of credit card accounts and associated relationships, does the model reflect the business in an accurate way?
Q2	Homomorphism (Fidelity to modelled phenomena)	In terms of account ownership, does the model reflect the business in an accurate way?
Q3	Homomorphism (Fidelity to modelled phenomena)	Overall, does the model reflect the business in an accurate way, considering the represented business concepts and associated relationships, while complying with the recommended practices of Data Vault 2.0?
Q4	Completeness	Considering only the business domains modeled, how do you classify the artifact in terms of completeness?
Q5	Consistency with people (Understandability; Ease of use)	In your opinion, is the proposed artifact simple to understand and use?
Q6	Robustness	How would you classify the robustness of the artifact in its ability to adapt to both the presented business requirements and those that may arise in the future?
Q7	Robustness	In your opinion, does the model represent a good proof of concept for a future implementation?
Q8	Consistency with organization (Fit with organization)	In what way do you consider the existence of the proposed artifact pertinent and/or important in the context of the organization?
Q9	Consistency with people (Utility)	In your opinion, can the proposed artifact be useful for data architects, engineers, analysts of the organization?
Q10	(Generic)	What recommendations/suggestions would you give to improve the artifact?
Q11	(Generic)	What other comments can you provide about the proposed artifact?

7.4 ANALYSIS OF THE RESULTS

In this section, the transcriptions of the interviews will be analyzed to discover patterns within the responses, determine the overall opinion of the participants regarding the topics discussed, and

summarize the most common suggestions and comments. A summary of the main findings and overall sentiment will be provided for each question asked in the interview.

7.4.1 Q1: Beneficiaries of credit card accounts and associated relationships

All participants except for one (P4, male, Analyst/Developer) agreed that, according to the assumptions made when designing the model and based on the metadata that was provided, the artifact reflects the business in an accurate way in terms of beneficiaries of credit card accounts and associated relationships. Participant P4 shared that, from his knowledge of the data and metadata, the beneficiaries part of the model was not as complex as in our proposal, and that it could be simplified.

One participant (P2, female, Technician level II) revealed to us some preoccupations she had regarding the beneficiaries, but confessed she had little knowledge on that specific part of the business. She emphasized how it wouldn't make sense for the relationship between credit card accounts and current accounts exist only through the beneficiaries in the case where the beneficiaries do not need to be customers of the company. However, she was not sure if all beneficiaries coming from the source are registered as customers of the bank.

Another participant (P3, female, Analyst/Developer) referred that the model would benefit from an identification of the multiple entities of the company and suggested the creation of an "entities hub" to model the multiple entities – customers or not – that play a part in the business. Nonetheless, she mentioned that, based on her knowledge, the beneficiaries and associated accounts were well modelled.

Two participants (P5, P6) said that the beneficiaries and associated accounts were modelled correctly according to the business and what was presented in the session but shared that they are not familiar enough with this part of the business to elaborate on the matter.

One participant (P7, male, Chief Data Officer), apart from stating that the model accurately reflects the business in terms of beneficiaries and correlated accounts, commented positively on how the model allows for incorrect data to be integrated into the Data Warehouse, because that happens in a lot of sources, and can actually be a good thing, if the data is marked as incorrect. He added that the company is currently trying to do that when integrating other companies of the same group.

7.4.2 Q2: Account ownership

In terms of account ownership, whether it is between a customer and a credit card account or between a customer and a current account, all participants agreed that, overall, the model reflected the business in an accurate way, according to the assumptions made while modelling. However, one of them answered positively with some reservations (P4, male, Analyst/Developer), as they did not see the model fully implemented or look at the data carefully.

The majority of participants (P2, P3, P5, P6, P7) answered with no reservations that the model reflected the ownership of accounts accurately. One participant (P3, female, Analyst/Developer) emphasized that the multi-active satellite for ownership between a customer and an account can be adapted to

any scenario (any other type of account) and another participant (P6, female, Technician level II) added that it was a good and innovative way of modelling the intervention of customers in accounts.

7.4.3 Q3: Model accuracy in representing the business and complying with Data Vault 2.0

Overall, all participants agreed that the model, within the scope and modelled concepts, reflects the business in an accurate way, while complying with the Data Vault 2.0 recommended practices.

Two participants (P1, P3) mentioned the importance of an enterprise/entity code to identify the multiple enterprises, aside from the main bank, that are also part of the business, which were not modelled. However, they also understood that this information was not provided in the metadata that was accessed to design the model. One of them (P3, female, Analyst/Developer) also mentioned that there are some fields that need to be decoded in reference tables, similarly to the product code and branch code ones. Another participant (P4, male, Analyst/Developer) mentioned that, apart from the beneficiaries question, the model represented the business relatively well, but added that the loan accounts are much more complex than what was presented in the demonstration, as they are an application on their own.

The majority of the participants said that the model accurately reflected the business without any reservations, considering the addressed business concepts (P2, P5, P6, P7). One of them (P2, female, Technician level II) added that the main benefit of this solution is the separation of the satellites by rate of change, as the usual models treat all tables and attributes the same way when it is rarely the case. This way, only the tables that suffered updates are accessed, saving time, and improving performance.

7.4.4 Q4: Completeness of the artifact

Almost all participants (P2, P3, P4, P5, P6, P7) considered the model really complete, especially given the information that was provided about the data. One participant (P4, male, Analyst/Developer) referred that the most important thing is to have customers, accounts, and the relationship between them well defined, and that part was well represented. Another participant (P2, female, Technician level II) added that the model has two “entries”, current accounts, and term deposit accounts, two “exits”, loan accounts, and credit cards, and the account that manages them, so, in that sense, the model is very complete. One participant (P7, male, Chief Data Officer) stated that the model not only covered all the most important themes but also those that are “trickier” and less obvious.

One participant (P1, male, Technician level III) revealed some concerns on the limitations of the model, as it lacks some basic concepts associated to the ones that were modelled, such as enterprise codes and subproducts, but recognized that, given the available metadata and lack of access to the whole company structure, it is considerably complete.

7.4.5 Q5: Simplicity of the artifact

The majority of the participants found the model simple to understand and use, especially given the way it was presented (P2, P3, P6, P7). Two participants found it even easier than the popular relational or dimensional models (P2, P7). One of them (P2, female, Technician level II) shared that, because the Data Vault model separates the keys, descriptive attributes, and relationships, it is even easier to understand. The other (P7, male, Chief Data Officer) focused on how no model is difficult to understand when there is logic and reasoning behind the choices made. He also added that if we define the hubs clearly and start from there, we can have a full enterprise model with just hubs and links, while the satellites are only needed to answer specific questions about the data.

Three participants agreed that the model is not as easy to understand (P1, P4, P5). One of them (P1, male, Technician level III) considers that Data Vault models are never easy to understand and need a lot of documentation and tools to simplify the model for it to be easier to understand. However, the same participant added that the Bridge Table and other comparable solutions are a way to optimize queries, especially in his field, which is Data Quality, and stated that overall, apart from his personal opinion on Data Vault as a modelling approach, the model was simple to understand. The other two participants (P4, P5) consider that the model has an intermediate level of simplicity, especially due to the fact that it has too many tables.

7.4.6 Q6: Robustness and flexibility of the artifact

All participants referred to the Data Vault model as being completely adaptable, especially regarding the presented examples of requirements. In terms of the capacity to accommodate future requirements, some participants (P1, P4) had some reservations as to how it would perform in real life scenarios. One participant (P4, male, Analyst/Developer) emphasized the need to test the model with real data in order to fully understand its flexibility.

One participant (P2, female, Technician level II) stated that the model is robust and is a great option for storing data. She added that Data Vault in general is a great modelling approach to store data, however, for data extraction and manipulation there are better, more advanced tools that should be used to complement the Data Vault model.

Another participant (P7, male, Chief Data Officer) recognized the adaptability of the model but revealed that, in his opinion, the demonstration lacked scenarios in which the model needs to be restructured, as Data Vault is also very permissive in terms of changes in its structure without losing the existent information and context.

7.4.7 Q7: Proof of concept for future implementation

All participants considered the artifact a good starting point for a future implementation. There were very few additional comments on this question. One participant (P1, male, Technician level III) referred that the artifact has some interesting modelling approaches but is still lacking a lot of business concepts for a potential implementation. However, he considered it a simple yet effective proof of concept. Another participant (P2, female, Technician level II) shared that the model represents a good proof of

concept for future implementation, especially for a large-scale Data Warehouse of a company with a lot of acquired companies and, consequently, data sources.

7.4.8 Q8: Pertinence and importance of the artifact in the context of the organization

All participants agreed that, overall, the artifact is important and pertinent, especially considering that the bank is currently working towards the first steps of implementation of the new Data Warehouse. Some also mentioned that the model has interesting ideas and solutions that could be used and adapted to the company's current Data Warehouse implementation efforts (P1, P5, P6).

One participant (P4, male, Analyst/Developer) mentioned that the model is very pertinent and important, especially in terms of improving performance. He added that, although data access is more difficult in Data Vault, there seem to be many solutions to improve performance even in this case. Another participant (P2, female, Technician level II) also shared that the Data Vault model is easier to implement than, for example, relational and dimensional models, however, only if it is implemented from scratch, as migrating models is exceedingly difficult.

Two participants (P3, P7) spoke on the adaptability of the model as its biggest strength. One of them (P3, female, Analyst/Developer) shared that, currently, the process of altering and including new attributes is very difficult and that implementing a Data Vault model in the company's Data Warehouse would help in this sense. The other participant (P7, male, Chief Data Officer) compared the Data Vault model to its dimensional counterpart and explained how, in dimensional Data Warehouses, all of the dimensions need to be well defined from the start, otherwise, future changes can be very difficult. In the context of the company, some dimensions, such as clients, are always changing and, in that sense, this type of Data Vault model would help a lot, as it is great in terms of evolution. He added that, for example, if a certain business concept is still not fully implemented, reference tables can be used to decode some fields, such as the examples demonstrated in the presentation. In his opinion, Data Vault should be the Data Warehouse model and dimensional models should only be used to deliver data products.

7.4.9 Q9: Usefulness of the artifact for data architects, engineers, and analysts

Overall, all participants found that the artifact can be useful for data architects, engineers, and analysts of the organization. One participant (P2, female, Technician level II) specified that it can be very useful for Data Warehouse modelers, however, for data analysts, it can be more difficult to adapt to this new modelling approach, because there are too many tables. She added that, with a Data Vault Data Warehouse, data can be delivered to analysts faster, as new changes to the model are introduced faster as well.

7.4.10 Q10: Additional recommendations and suggestions

Most participants revealed that they did not have any other suggestions apart from the ones included in the previous questions (P1, P4, P5, P6, P7). One participant (P2, female, Technician level II) suggested

that the modelers of data repositories should always be accompanied by the technicians responsible for data applications and other people that fully understand the business. In her opinion, the model should always be linked to a specific industry, in this case, the banking industry. Apart from that, she mentioned that the products and branches go far beyond reference tables, but that would be something to improve in the future. Another participant (P3, female, Analyst/Developer) also touched on the topic of the reference tables, and stated that, apart from her previous suggestions, the reference tables should be integrated into the model in some way, instead of not being linked to any tables.

One participant (P7, male, Chief Data Officer), despite not having any other recommendations, emphasized the fact that the customers, private and enterprise, were modelled in a simple and direct way. He added that the model does not have to be perfect before implementation and that, even in the case of problems that Data Vault is unable to solve directly, other solutions were explored and well presented.

7.4.11 Q11: Other comments

Most participants had no other comments to provide (P1, P3, P5, P6, P7). Participant P2 (female, Technician level II) stated that Data Vault provides a solution for almost everything and advises the modeler whether a certain solution may hurt performance. She added that the Bridge Table is an effective way of facilitating the transition between data storage and data extraction. Participant P4 (male, Analyst/Developer) shared that the only thing missing is a practical implementation and demonstration with a subset of real company data.

8 RESULTS AND DISCUSSION

8.1 MODEL ASSESSMENT

Upon collecting qualitative data on the proposed artifact through interviews with domain experts, we are now able to accurately assess the model in multiple angles and discuss its implications in the context of the company, strengths, weaknesses, and possible improvements.

After carefully summarizing the opinions of seven practitioners on the proposed model, we can separate the answers according to more generic and practical criteria: accuracy in reflecting the business, completeness, simplicity, robustness, importance, and utility. We can also integrate the additional recommendations and comments made by the interviewees into these criteria to better integrate the data collected and discuss the final results of the evaluation of the artifact. Table 30 summarizes the overall opinion, feedback, and suggestions by analyzed criteria.

Table 30 - Summary of the feedback from the interviews

CRITERIA	OVERALL OPINION	ADDITIONAL FEEDBACK		SUGGESTIONS
		POSITIVE	NEGATIVE	
Accuracy in reflecting the business (Q1, Q2, Q3)	The model reflects the business in an accurate way, considering the business concepts modelled	The model allows the integration of incorrect data that can be marked as such; Ownership multi-active satellite is easily adapted to other scenarios; Innovative; Separation of satellites by rate of change; Customers (private and enterprise) are modelled in a simple and direct way	Lack of an "enterprise code" for other banks of the group; Beneficiaries part is too complex; Lack of decoding for some code fields	The modeler should work closely with the people responsible for the data sources; Model products and branches as hubs; Reference Tables should be integrated in the model
Completeness (Q4)	The model is really complete, given the information available	The most important parts of the business (clients and accounts) are well modelled; 2 "entries" and 2 "exits" in terms of business flow – well balanced; Tricky cases were well modelled	Lack of basic concepts such as enterprise codes and subproduct codes	
Simplicity (Q5)	The model is relatively simple to understand and use, especially	Easier to understand than relational or dimensional models; Clear separation of	Too many tables; Needs a lot of documentation	

	given the way it was presented	business keys from attributes and relationships; Bridge Tables are a good way of optimizing the model		
Robustness (Q6, Q7)	The model is very adaptable and a good starting point for future implementation	Great for storing data	Not as good for data extraction; Needs to be tested with real data; Demonstration lacks scenarios where the structure of the model needs to be altered	
Importance (Q8)	The model is very pertinent and important, as the company is moving towards DW implementation	Some modelling approaches presented can be adapted and used by the company; Good performance; Solutions to improve performance in data access; Adaptability; Easy to implement from scratch; Allows incremental implementation	Difficult to migrate from other models to DV	
Utility (Q9)	The model is very useful for data engineers and analysts, but especially for data architects (modelers)	Data is delivered to analysts faster	Difficult for data analysts to adapt to it, because of the high number of tables	

We can conclude that, regarding all criteria evaluated, the interviewees' responses were generally very positive. Overall, not much concrete feedback and suggestions, beyond the asked questions, was given. This was mainly due to two reasons: the participants did not feel comfortable answering about certain topics that they were not proficient in, and the fact that the model was not fully implemented, loaded with real data, and tested for performance. Nonetheless, we can identify some suggestions for improvement: (i) model products and branches as their own hubs (business concepts), (ii) introduce some sort of "enterprise code" to identify the companies, (iii) create reference tables for the other existing codes, (iv) model subproduct codes, (v) integrate reference tables in the model.

8.2 IMPLEMENTATION OF SUGGESTIONS

Modelling products and subproducts as business concepts was outside of the scope of this project, as we did not have access to their metadata, so they were modelled simply as codes, although, in the real context of the organization, they have historical attributes of their own. In order to model them, we will assume that products and subproducts behave in the same way and have similar attributes. This way, it only makes sense to load them all into the same hub – H_PRODUCT –, as they even have the same key length. The only thing left to model is the relationship between products and subproducts. In Data Vault 2.0, this is the classic example of a hierarchical link, which is a link with two references to the same hub, where parent-child relationships are modelled (Linstedt & Olschimke, 2016, p. 129). This link behaves as a normal Data Vault link, containing a unique hash key, record source and load date timestamp attributes and the hash keys of the referenced hubs (which come from the same hub, in this case). It just serves as a way to model which subproducts are a subtype of a certain product, and vice versa.

The accounts originally have the product code as part of their business key but now that code will need to be removed as it is part of its own hub. However, the accounts' composite key is no longer unique without the product code. To simplify and make the model more general and permissive of other sources and data formats, we will remove the composite key of the accounts and leave only the *account_id* attribute, which is always unique. Now, each account can be linked to the product hub, to identify the type of account it is, through a link table, as shown in Figure 24. The prefix "HAL" in HAL_PRODUCT_SUBPRODUCT stands for hierarchical link.

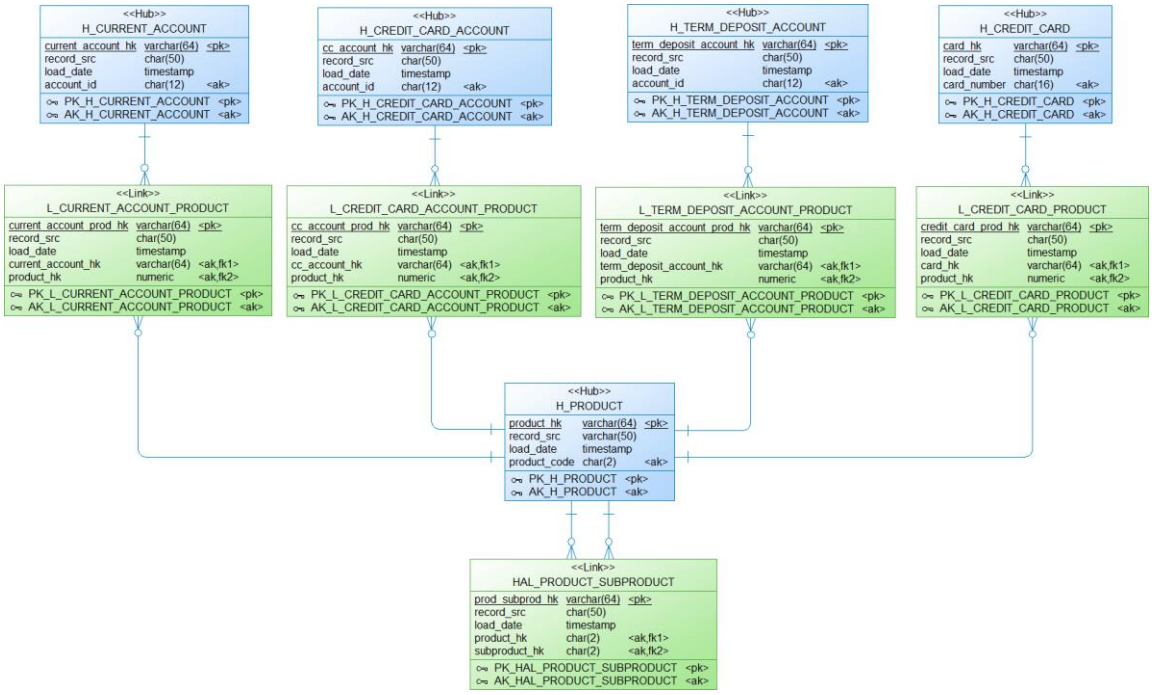

Figure 24 - Products, subproducts and accounts

This diagram only showcases the tables that are affected by the introduction of the product hub in the model, to allow for better readability. Apart from that, H_PRODUCT and HAL_PRODUCT_SUBPRODUCT

do not have any satellites, as there was no provided metadata with this information. Only hubs and links are presented.

In terms of branches, as they are part of a much bigger internal organization composed by departments, areas, etc., that we have no information on, it is not possible to model them in a sufficiently accurate way. The same applies to suggestion (ii), which has to do with the multi-enterprise nature of the company. The company group has several enterprises, each of them with a unique enterprise code. Currently, the model only reflects the customers and accounts of a single bank, the main one. To make the model multi-enterprise, as two interviewees suggested, a new hub for the entities would be needed or, at least, a reference table containing a description of each enterprise code and a new attribute in S_CUSTOMER_BASICS with the enterprise code identifying the bank the customer is registered in. However, these possible implementations are purely based on assumptions and the brief feedback given on the interviews. Transforming the model into one that reflects the business of multiple enterprises would mean restructuring it completely. Figure 25 illustrates the simplest solution for the inclusion of enterprise codes in the model, in the context of the customers, without altering the scope of the project, and given that we do not have access to metadata of other enterprises that belong to the group.

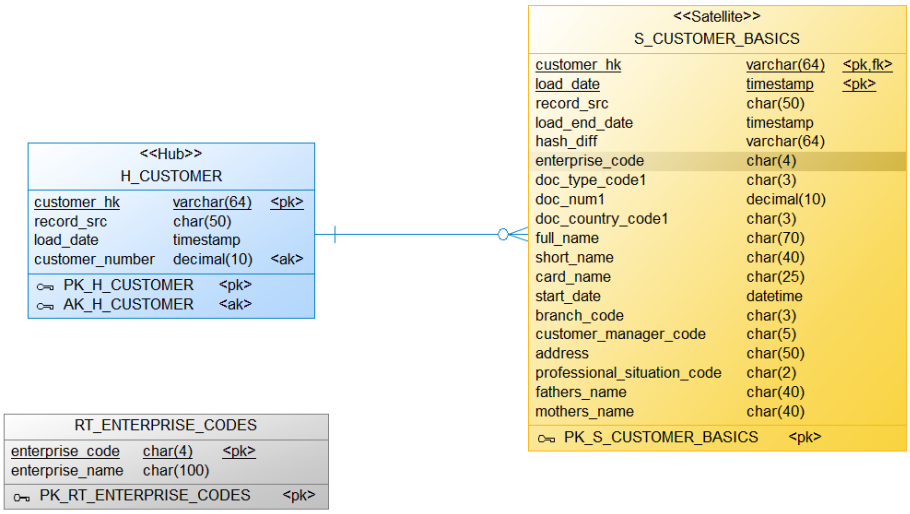

Figure 25 - Inclusion of enterprise codes in the model

Regarding suggestion (iii), the other fields in the model that need decoding are: (i) document type code, (ii) customer manager code, (iii) professional situation code, (iv) education level code, (v) marital status code, (vi) employment relationship code, (vii) company type code, (viii) card brand code, (ix) insurance plan code, (x) account manager code, (xi) currency code, (xii) account availability code, (xiii) account handling code. Apart from these, there are other indicator fields that need contextual information. The solution would be to simply create reference tables similar to the product and branch code ones, with the codes as primary and business key, and an attribute containing the descriptive information that explains that code. Because reference data has no impact on the model as it is not part of it, there are too many codes, and the reference tables all have the same structure, we only modelled products and branches to exemplify.

Suggestion (iv) is already implemented in the sense that subproduct codes are the same as subproduct codes in the scope of this project. Both are stored in the product codes reference table – RT_PRODUCT_CODES –, as both are two characters long and have a description of the code as contextual information. Because we had no access to any other information on product/subproduct codes, they were only modeled as reference data and included in the business keys of accounts, in the case of product codes, and satellites of certain accounts, in the case of subproduct codes (e.g., term deposit account).

Regarding suggestion (v), it does not make sense to integrate the reference tables into the model, as we do not want to keep invalid data from entering the Data Warehouse. By introducing foreign key constraints between the reference table business keys and the correspondent codes in other tables of the model, we are enforcing referential integrity between codes. The Data Warehouse should allow incorrect data and report it, to discover what went wrong in the process of entering the data and maintain historical records of that (Hultgren, 2012, p. 225).

8.3 DISCUSSION

These alterations further prove that the Data Vault 2.0 model easily adapts to changes in the business without needing to redesign the whole existing structure of the Data Warehouse, which is, indubitably the most prominent advantage of the model. The banking company could benefit a lot from using a model such as this one, as they have huge volumes of data being loaded into the DW every day, and their data is always changing. By implementing a Data Vault 2.0 DW model, they can have a working solution in each increment of the implementation project, which allows them to continue delivering data products and testing the model while the implementation is still in progress.

Additionally, by comparing the originally defined objectives of the solution to the results of the demonstration and evaluation phase (interviews), we can conclude that the artifact adequately met the objectives.

Regarding the first objective of reflecting the business in an accurate way, we can confidently say that this was achieved, according to the feedback from the participants and the demonstration performed. Considering the limited scope of the model, every business concept that was presented and modelled was according to the business and provided metadata. The model's ability to scale and adapt to new requirements has also been demonstrated many times through examples of new requirements, alterations made using feedback from the interviews, and was confirmed by the participants in the interviews. The model can also be considered as helpful for DW modelers and users of the organization, as some participants mentioned that some ideas and approaches from the Data Vault 2.0 model can and should be integrated in their ongoing replatforming project. In terms of improving DW processes performance, because the model was not fully tested and implemented with real company data, it is impossible to define whether this objective was met. However, recommended Data Vault 2.0 practices for streamlining performance were used and some interviewees mentioned that they were innovative and could work well in real scenarios. The participants that were familiar with Data Vault modelling considered the model easy to use and understand. Finally, it was consensual between all practitioners interviewed that the artifact constitutes a good proof of concept of a Data Vault 2.0 model for future implementation in the company.

9 CONCLUSIONS

This study summarized the best practices for modelling a Data Warehouse using the Data Vault 2.0 methodology, considering that it is loaded from a Data Lake, and studied the impact of a Delta Lake layer on this type of architecture. Additionally, a Data Vault 2.0 model was proposed, based on company metadata, and focusing on a subset of business concepts, to integrate the company's Data Lake architecture. Ending this research, the defined objectives were achieved: (i) to present a Data Vault 2.0 model for the EDW, using the company's metadata, (ii) to identify best practices for the integration of the EDW into the Data Lake architecture, (iii) to discover if and how Delta Lake concepts can impact the efficiency of data loading from the Data Lake to the Enterprise Data Warehouse.

The Systematic Literature Review allowed for a complete and unbiased review of existent literature, which revealed a lack of guidelines and best practices regarding architectures that combine a Data Lake and a Data Warehouse. Nonetheless, it revealed the most common approach to divide the Data Lake into zones, the types and number of zones that are usually modelled, the basic folder structure that is recommended to store files in the raw zone and in structured zone, the advantage of using a Data Lake to source a Data Vault model as they are both schema-on-read, and the fact that it is unnecessary and redundant to implement a Delta Lake layer in an architecture that contains a Data Warehouse. All of these findings contributed to a more informed, holistic view of data architectures, modelling techniques and methodologies, which further contributed to the design of the Data Lake and Data Vault 2.0 model as part of the same architecture. The fact that the Data Vault 2.0 model is loaded from a Data Lake ultimately has no impact in the modelling process, it would only impact the loading of the data into the model. The Delta Lake layer also ended up not having a positive impact in this architecture, at least from a theoretical standpoint, as it aims to mimic DW capabilities and, in this case, would only make the architecture unnecessarily more complex.

Although the model was not physically implemented, we were able to assess it according to experts' opinions, which are employees of the company and have a considerable understanding of the business and the data itself. The model was evaluated according to many criteria, such as accuracy, utility, ease of use, understandability, robustness, etc., and the feedback was very positive. The demonstration phase was also important to showcase the flexibility of the model and its adaptability to hypothetical requirements from the company. Apart from positive feedback, some suggestions were made by the participants, which were implemented if they were inside of the scope or did not violate Data Vault 2.0 guidelines and recommended practices.

This document constitutes the communication phase of the research, as it rigorously presents and explains the artifact, its importance, its advantages and utility for researchers and industry practitioners. A review paper was also published in a renowned international Journal, with the contents of the SLR. Finally, the results of the SLR were also communicated formally in an online session for employees of the company working in the data and analytics field.

This study successfully provided the scientific community with a complete state-of-the-art regarding Data Lakes sourcing Data Warehouses, Data Vault 2.0, and Delta Lake layers. It also contributed with a proof of concept of Data Vault 2.0 models in the context of the banking industry and large-scale Data Warehouses, using real data and use cases, which can be used for future implementations and developments in the field.

9.1 LIMITATIONS

As the company's replatforming project evolved, some decisions were made regarding the structure of the Data Lake, platforms and modelling approaches that were beyond our control and impacted the predicted course of this study. Although the findings of the literature review had some influence on the design of the Data Lake zones, most decisions were made by consultants and service providers of the chosen cloud platform to host the Data Lake. For that reason, the focus of this research ended up being on the Data Vault 2.0 model and proving it was a fit for a large-scale DW and for this banking company in particular.

Due to time, management, and security constraints inside the company, it was impossible to have access to a "data box" in the Enriched zone of the Data Lake, specifically prepared to implement the Data Vault 2.0 model. Without the credentials to access the Data Lake, the model could not be implemented and loaded to data in order to test its performance. Because of this, the model presented is purely conceptual and was only evaluated subjectively, in terms of its structure, capacity to adapt to business changes, utility, simplicity and completeness.

To design the model, metadata from the source system was provided. Because the company's business is very complex, involves many entities and entails a very deep understanding of the banking industry, the model's scope was limited to customers, current accounts, credit card accounts, credit cards and all relationships between them. Involving more business concepts would exponentially increase the complexity of the model and compromise its understandability and readability. Additionally, accessing and understanding the metadata was very difficult, which led to a need for support from various people from the data governance department. Because real company data was only accessed on premises through employees of the company, and not provided to this research for legal reasons, it was difficult to understand relationships between the data and decode the meaning and behavior of some attributes.

Finally, access to the company's *Microsoft Office* account was limited, which complicated communication with some people that were crucial to the understanding of the business and subsequent design of the DW model.

9.2 RECOMMENDATIONS FOR FUTURE RESEARCH

There are many opportunities for future research based on the work of this dissertation, related to the limitations mentioned above.

The proposed Data Vault 2.0 model should be implemented in the future, in order to properly assess its performance. Related to this, after a pilot implementation to test the model, it should be enriched with more business concepts that are crucial to the functioning of the company. Ideally, it would be scaled to represent the whole business. Additionally, its performance could be compared to the existent relational DW model, testing only the modelled business concepts of course, to further validate the advantages of Data Vault 2.0 over other modelling approaches.

It would also be interesting to try different approaches for file and metadata organization inside the Data Lake and see how it impacts data loading into the Data Warehouse.

Finally, Delta Lake files could be implemented from the Structured zone forward, as a way to replace the Data Vault 2.0 EDW and structure data for utilization in Data Marts, which would be modelled according to the Kimball methodology. These two approaches could be compared in terms of timeliness of data products' delivery, query performance, and efficiency in loading data from the Delta tables to the Data Marts.

BIBLIOGRAPHICAL REFERENCES

Adams, R. J., Smart, P., & Huff, A. S. (2017). Shades of Grey: Guidelines for Working with the Grey Literature in Systematic Reviews for Management and Organizational Studies. *International Journal of Management Reviews*, *19*(4), 432–454. https://doi.org/10.1111/ijmr.12102

Alrehamy, H., & Walker, C. (2018). SemLinker: Automating big data integration for casual users. *Journal of Big Data*, *5*(1). Scopus. https://doi.org/10.1186/s40537-018-0123-x

An Efficient Data Lake Structure. (2019, December 19). *Scalefree*. https://www.scalefree.com/scalefree-newsletter/efficient-data-lake-structure/

Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). *Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics*. 11th Annual Conference on Innovative Data Systems Research (CIDR ’21).

Arnold, B. (2021, May 24). *How to Build a Modern Data Platform Utilizing Data Vault*. PhData. https://www.phdata.io/blog/building-modern-data-platform-with-data-vault/

Autarrom, S., Chantaranimi, K., Chompupoung, A., Jinapook, P., Mahanan, W., Lumpoon, P. N., Natwichai, J., Sugunsil, P., Sangamuang, S., Sukhvibul, T., & Thiengburanathum, P. (2022). *Data Service Platform for Social and Community to Drive the Royal Project Foundation* (Vol. 118, p. 10). Springer Science and Business Media Deutschland GmbH; Scopus. https://doi.org/10.1007/978-3-030-95903-6_1

Azevedo, V., Carvalho, M., Fernandes-Costa, F., Mesquita, S., Soares, J., Teixeira, F., & Maia, Â. (2017). Interview transcription: Conceptual issues, practical guidelines, and challenges. *Revista de Enfermagem Referência*, *4*(14), 159–167. Academic Search Complete. https://doi.org/10.12707/RIV17018

Baumann, P. (2022, July 5). *Data Architecture with SAP – Data Lake*. SAP Blogs. https://blogs.sap.com/2022/07/05/data-architecture-with-sap-data-lake/

Blueprint: Cloud Data Platform Architecture – Part 2: Data Lake. (2021, June 21). B.Telligent. https://www.btelligent.com/en/blog/blueprint-cloud-data-platform-architecture-data-lake-1/

Chu, L. (2020, July 6). *Implementing a Data Lake or Data Warehouse Architecture for Business Intelligence?* Towards Data Science. https://towardsdatascience.com/implementing-a-data-lake-architecture-for-business-intelligence-f2c99551db1a

Coates, M. (2017, March 29). *Designing a Modern Data Warehouse + Data Lake* [PowerPoint]. https://static1.squarespace.com/static/52d1b75de4b0ed895b7e7de9/t/59e3bd8464b05fe9e6bbe969/1508097416856/DesigningAModernDWandDataLake_MelissaCoates.pdf

Data Lakehouse: Concept, Key Features, and Architecture Layers. (2021, November 10). AltexSoft. https://www.altexsoft.com/blog/data-lakehouse/

Eichler, R., Giebler, C., Gröger, C., Schwarz, H., & Mitschang, B. (2021). Modeling metadata in data lakes—A generic model. *Data & Knowledge Engineering*, *136*, 101931. https://doi.org/10.1016/j.datak.2021.101931

Etse, G. (2022, May 17). *Comparison of database architectures: Data warehouse, data lake and data lakehouse*. Adaltas. https://www.adaltas.com/en/2022/05/17/data-warehouse-lake-lakehouse-comparison/

European Parliament & Council. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. *59*(L 119), 43–44.

Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). Making data platforms smarter with MOSES. *Future Generation Computer Systems*, *125*, 299–313. https://doi.org/10.1016/j.future.2021.06.031

Galletta, A. (2013). *Mastering the semi-structured interview and beyond: From research design to analysis and publication*. New York University Press.

Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, *106*, 101–121. https://doi.org/10.1016/j.infsof.2018.09.006

Giebler, C., Gröger, C., Hoos, E., Schwarz, H., & Mitschang, B. (2019). *Modeling data lakes with data vault: Practical experiences, assessment, and lessons learned: Vol. 11788 LNCS* (p. 77). Springer Science and Business Media Deutschland GmbH; Scopus. https://doi.org/10.1007/978-3-030-33223-5_7

Golec, D. (2019). Data Lake Architecture for a Banking Data Model. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3490476

Hai, R., Geisler, S., & Quix, C. (2016). Constance: An intelligent data lake system. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, *26-June-2016*, 2097–2100. Scopus. https://doi.org/10.1145/2882903.2899389

Hlupic, T., Orescanin, D., Ruzak, D., & Baranovic, M. (2022). An Overview of Current Data Lake Architecture Models. *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology, MIPRO 2022 - Proceedings*, 1082–1087. Scopus. https://doi.org/10.23919/MIPRO55190.2022.9803717

Holom, R.-M., Rafetseder, K., Kritzinger, S., & Sehrschön, H. (2020). Metadata management in a big data infrastructure. *Procedia Manufacturing*, *42*, 375–382. Scopus. https://doi.org/10.1016/j.promfg.2020.02.060

Hultgren, H. (2012). *Modeling the agile data warehouse with data vault*. New Hamilton.

Hultgren, H. (2015). *Data Vault Certification CDVDM Book: Genesee Academy*. Brighton Hamilton.

Inmon, B. (2016). *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump* (1st ed.). Technics Publications, LLC.

Inmon, W. H., Linstedt, D., & Levins, M. (2019). Chapter 2.1—The End-State Architecture—The "World Map." In W. H. Inmon, D. Linstedt, & M. Levins (Eds.), *Data Architecture (Second Edition)* (pp. 47–57). Academic Press. https://doi.org/10.1016/B978-0-12-816916-2.00008-5

Kitchenham, B. (2004). Procedures for Performing Systematic Reviews. *Keele, UK, Keele Univ.*, *33*.

Kukreja, M. (2021, October 22). *Combining the Power of Data Lake and Data Warehouse—Lakehouse Architecture*. Medium. https://aws.plainenglish.io/combining-the-power-of-data-lake-and-data-warehouse-lakehouse-architecture-70262424050e

Kyslyi, A. (2021, May 21). *Data Warehouse and Data Lake: Why Go for a Hybrid Scenario*. Infopulse. https://www.infopulse.com/blog/hybrid-data-lake-data-warehouse

Linstedt, D., & Olschimke, M. (2016). *Building a scalable data warehouse with Data Vault 2.0*. Morgan Kaufmann.

Llave, M. R. (2018). Data lakes in business intelligence: Reporting from the trenches. *CENTERIS 2018 - International Conference on ENTERprise Information Systems / ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018*, *138*, 516–524. https://doi.org/10.1016/j.procs.2018.10.071

Mazumdar, D. (2022, October 4). *What Is a Data Lakehouse?* Dremio. https://www.dremio.com/blog/what-is-a-data-lakehouse/

Megdiche, I., Ravat, F., & Zhao, Y. (2020). A use case of data lake metadata management. In *Data Lakes* (pp. 97–122). wiley; Scopus. https://doi.org/10.1002/9781119720430.ch5

Mike. (2022, March 10). Building the Lakehouse Architecture With Azure Synapse Analytics. *SQL Of The North*. https://sqlofthenorth.blog/2022/03/10/building-the-lakehouse-architecture-with-synapse-analytics/

Mikhailouskaya, I. (2018, May 21). *Alternative Approaches to Implementing Your Data Lake*. ScienceSoft. https://www.scnsoft.com/blog/data-lake-implementation-approaches

Mitruś, P. (2021, June 2). *Data Lake Architecture: How to create a well Designed Data Lake*. Lingaro Group. https://lingarogroup.com/blog/data-lake-architecture

Nambiar, A., & Mundra, D. (2022). An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management. *Big Data and Cognitive Computing*, *6*(4), Article 4. https://doi.org/10.3390/bdcc6040132

Nogueira, I. D., Romdhane, M., & Darmont, J. (2018). Modeling data lake metadata with a data vault. *ACM International Conference Proceeding Series*, 253–261. Scopus. https://doi.org/10.1145/3216122.3216130

Olschimke, M. (2022, August 30). Data Vault 2.0: Best of Breed from Data Warehousing and Data Lakes | Experts in Consulting and Training. *Scalefree*. https://www.scalefree.com/scalefree-newsletter/data-vault-2-0-best-of-breed-from-data-warehousing-and-data-lakes/

Orescanin, D., & Hlupic, T. (2021). Data Lakehouse—A Novel Step in Analytics Architecture. *2021 44th International Convention on Information, Communication and Electronic Technology, MIPRO 2021 - Proceedings*, 1242–1246. Scopus. https://doi.org/10.23919/MIPRO52101.2021.9597091

Oukhouya, L., El haddadi, A., Er-raha, B., Asri, H., & Laaz, N. (2023). A Proposed Big Data Architecture Using Data Lakes for Education Systems. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 147, pp. 53–62). Springer Science and Business Media Deutschland GmbH; Scopus. https://doi.org/10.1007/978-3-031-15191-0_6

Oukhouya Lamya, El haddadi Anass, Er-raha Brahim, & Asri Hiba. (2021). A generic metadata management model for heterogeneous sources in a data warehouse. *E3S Web of Conferences*, *297*, 01069–01069. https://doi.org/10.1051/e3sconf/202129701069

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst.*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pisoni, G., Molnár, B., & Tarcsi, Á. (2021). Data science for finance: Best-suited methods and enterprise architectures. *Applied System Innovation*, *4*(3). Scopus. https://doi.org/10.3390/asi4030069

Prat, N., Wattiau, I., & Akoka, J. (2014). Artifact Evaluation in Information Systems Design Science Research ? A Holistic View. *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2014*.

Priebe, T., Neumaier, S., & Markus, S. (2021). Finding Your Way Through the Jungle of Big Data Architectures. *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, 5994–5996. Scopus. https://doi.org/10.1109/BigData52589.2021.9671862

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for Design Science Research Evaluation. *European Conference on Information Systems*.

Ravat, F., & Zhao, Y. (2019a). *Data Lakes: Trends and Perspectives: Vol. 11706 LNCS* (p. 313). Springer; Scopus. https://doi.org/10.1007/978-3-030-27615-7_23

Ravat, F., & Zhao, Y. (2019b). *Metadata Management for Data Lakes* (Vol. 1064, p. 44). Springer Verlag; Scopus. https://doi.org/10.1007/978-3-030-30278-8_5

Ren, P., Li, S., Hou, W., Zheng, W., Li, Z., Cui, Q., Chang, W., Li, X., Zeng, C., Sheng, M., & Zhang, Y. (2021). *MHDP: An Efficient Data Lake Platform for Medical Multi-source Heterogeneous Data: Vol. 12999 LNCS* (p. 738). Springer Science and Business Media Deutschland GmbH; Scopus. https://doi.org/10.1007/978-3-030-87571-8_63

Saddad, E., El-Bastawissy, A., Mokhtar, H. M. O., & Hazman, M. (2020). Lake data warehouse architecture for big data solutions. *International Journal of Advanced Computer Science and Applications*, *11*(8), 417–424. Scopus. https://doi.org/10.14569/IJACSA.2020.0110854

Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, *56*(1), 97–120. Scopus. https://doi.org/10.1007/s10844-020-00608-7

Sawadogo, P. N. (2019). *Textual Data Analysis from Data Lakes* (Vol. 1064, p. 563). Springer Verlag; Scopus. https://doi.org/10.1007/978-3-030-30278-8_54

Sawadogo, P. N., Darmont, J., & Noûs, C. (2021). *Joint Management and Analysis of Textual Documents and Tabular Data Within the AUDAL Data Lake: Vol. 12843 LNCS* (p. 101). Springer Science and Business Media Deutschland GmbH; Scopus. https://doi.org/10.1007/978-3-030-82472-3_8

Sawadogo, P. N., Kibata, T., & Darmont, J. (2019). Metadata management for textual documents in data lakes. *ICEIS 2019 - Proceedings of the 21st International Conference on Enterprise Information Systems*, *1*, 61–72. Scopus. https://doi.org/10.5220/0007706300720083

Sawadogo, P. N., Scholly, É., Favre, C., Ferey, É., Loudcher, S., & Darmont, J. (2019). *Metadata Systems for Data Lakes: Models and Features* (Vol. 1064, p. 451). Springer Verlag; Scopus. https://doi.org/10.1007/978-3-030-30278-8_43

Sienkiewicz, M., & Wrembel, R. (2021). Managing data in a big financial institution: Conclusions from a R&D project industrial paper. *CEUR Workshop Proceedings*, *2841*. Scopus. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85103498556&partnerID=40&md5=aa0fd5c13a0206e1982e96f725e1946f

Solodovnikova, D., & Niedrite, L. (2020a). *Change discovery in heterogeneous data sources of a data warehouse*. *1243 CCIS*, 23–37. Scopus. https://doi.org/10.1007/978-3-030-57672-1_3

Solodovnikova, D., & Niedrite, L. (2020b). Handling evolution in big data architectures. *Baltic Journal of Modern Computing*, *8*(1), 21–47. Scopus. https://doi.org/10.22364/BJMC.2020.8.1.02

Späti, S. (2022, August 25). *Data Lake / Lakehouse Guide: Powered by Data Lake Table Formats (Delta Lake, Iceberg, Hudi)*. Airbyte. https://airbyte.com/blog/data-lake-lakehouse-guide-powered-by-table-formats-delta-lake-iceberg-hudi

Tondak, A. (2022, August 11). *Delta Lake Architecture & Azure Databricks Workspace*. K21Academy. https://k21academy.com/microsoft-azure/data-engineer/delta-lake/

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Design Science Research in Information Systems. Advances in Theory and Practice* (pp. 423–438). Springer Berlin Heidelberg.

Wang, X.-Y. (2020). *PowerDesigner* (17.1). SAP.

Wieder, P., & Nolte, H. (2022). Toward data lakes as central building blocks for data management and analysis. *Frontiers in Big Data*, *5*. Scopus. https://doi.org/10.3389/fdata.2022.945720

Zaloni. (2016). *Why Your Data Warehouse Needs a Data Lake and How to Make Them Work Together*. https://cdn2.hubspot.net/hubfs/443949/DW%20Augmentation%20White%20Paper%2011.1.16.pdf

Ziegler, J., Reimann, P., Keller, F., & Mitschang, B. (2020). A Graph-based Approach to Manage CAE Data in a Data Lake. *53rd CIRP Conference on Manufacturing Systems 2020*, *93*, 496–501. https://doi.org/10.1016/j.procir.2020.04.155

APPENDIX A: SATELLITES

Figure 26 - All Satellites of the Data Vault 2.0 model

APPENDIX B: INTERVIEW SCRIPT

1. Greet participant and ask for consent to record the interview.
2. **Explain the agenda for the interview:**
    a. *Powerpoint* presentation
        i. Metadata tables used
        ii. Business concepts that were modelled (hubs)
        iii. Explain some specificities of the model: multi-active satellites, dependent child keys, reference tables and keyed-instance hubs
        iv. Model with focus on clients and current accounts
        v. Model with focus on beneficiaries of credit card accounts
        vi. Full model with all the tables
        vii. Presentation of examples of new requirements
        viii. Demonstration on how the model can be altered to comply to the requirements
    b. Introductory personal questions.
    c. Interview questions about the artifact.
    d. Emphasize that questions and additional comments should all be made at the end of the meeting, in the interview section.
3. Share *Powerpoint* presentation and explain in some detail the concepts and tables modelled and why (answer some questions regarding readability, but otherwise save comments for the end of the meeting).
4. **Introductory questions:**
    a. *Qual é a sua idade?*
    b. *Qual é o seu sexo?*
    c. *Qual é a sua função na empresa?*
    d. *Em que área da empresa trabalha?*
    e. *Há quantos anos trabalha na banca?*
    f. *Há quantos anos trabalha na área de dados e analítica?*
5. **Interview questions:**
    a. *Em termos de beneficiários de contas-cartão e relações associadas, considera que o modelo reflete o negócio de forma precisa?*
    b. *Em termos de titularidade de contas, considera que o modelo reflete o negócio de forma precisa?*
    c. *Em termos da relação entre os clientes, considera que o modelo reflete o negócio de forma precisa?*
    d. *De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete o negócio de forma precisa, cumprindo simultaneamente com as práticas recomendadas de Data Vault 2.0?*
    e. *Considerando apenas os conceitos de negócio modelados, como classificaria o modelo em termos de completude?*
    f. *Na sua opinião, o modelo é simples de entender e utilizar?*
    g. *Como classificaria a robustez do modelo, na sua capacidade de se adaptar tanto aos requisitos de negócio apresentados quanto aos que possam surgir no futuro?*

h. *Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?*
i. *De que forma considera a existência do modelo proposto pertinente e/ou importante no contexto da organização?*
j. *Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros, e analistas de dados da organização?*
k. *Que recomendações/sugestões daria para melhorar o modelo?*
l. *Que outros comentários pode fornecer sobre o modelo proposto?*

6. Thank the participant for their time.

APPENDIX C: INTERVIEW 1

I: Em termos dos beneficiários das contas cartão e essa parte do modelo, considera que o modelo reflete o negócio de forma precisa?
P1: Essa pergunta… essa pergunta é uma pergunta com rasteira, porque teria de fazer eu uma análise para ver se de facto se isso, se isso era a melhor solução ou não. Agora, nos pressupostos daquilo que vocês me apresentaram sim. Esses pressupostos sim.

I: Em termos da titularidade das contas, e daquela questão dos satélites multi-ativos, o modelo reflete o negócio de forma precisa?
P1: Sim, sim. Lá está, sempre com as reservas dos pressupostos não é –
I: Claro, claro –
P1: Por mim, por exemplo aí (…) a introdução de, no caso da empresa, nesse caso não, mas e outras podia fazer outro tipo de coiso, mas nos pressupostos que vocês têm sim.
I: Sim, sim, sim. Não, mas esses comentários também são importantes para nós porque também, para nós sabermos que há sempre espaço para melhorar.

I: Em termos de relação entre clientes. Acha que o negócio é refletido de forma precisa?
P1: O que é que queres dizer com relação entre clientes?
I: Eu posso mostrar. Temos uma tabela link que vai relacionar dois clientes. Eu posso mostrar, é mais fácil, sem mostrar não, ok. Ok portanto, ali em baixo temos uma "customer", uma inter-relação que vai pegar em dois "customers", vai haver uma chave extra, que vai ser a tal titularidade, vai ter os atributos descritivos, da granularidade dum cliente com o outro e –
P1: Sim, sim, sim, sim (…) sim, está bem, sim, sim. A resposta é sim.

I: Agora a questão é no geral, se considera que o modelo reflete o negócio de forma precisa, considerando os conceitos de negócios apresentados, em cumprimento com os conceitos de Data Vault 2.0?
P1: Lá está, a minha questão é só com o código de empresa não estar aqui dentro do modelo, mas não vos tendo sido passado de alguma forma esse… esse… esse… esse… a importância desse dado, digamos que sim, eu considero que sim, dentro daquilo que vocês apresentaram considero que sim.

I: Considerando então, apenas os conceitos de negócio modelados e aquilo que nós apresentámos, como é que classificaria o modelo em termos de completude?
P1: Em termos de completude, isso é um bocado, isso é um bocado (…) tás a ver, eu dentro daquilo que vocês mostraram, fazia-me falta ter os subprodutos, mas vocês não vão a esse nível de detalhe, fazia-me falta ter a empresa, mas vocês não têm esse nível de detalhe. Se me perguntas, a completude do modelo apresentado, no abstrato eu diria que falta aí muita coisa base, que tá associada à, aos próprios modelos que vocês apresentaram, agora se não vos foi dada essa informação, se vocês não têm acesso, à totalidade da estrutura do banco, etc., penso que deve ter havido aí algumas limitações, eu teria que dizer que sim, que tá completo dentro daquilo que eventualmente vos foi facultado.

I: Na sua opinião, o modelo é simples de entender e utilizar?
P1: Tamos a falar de Data Vault, acho que dentro do Data Vault não há nada simples de utilizar, nem, nem, no caso vocês meteram aí aquela questão daquela bridge. A bridge que é tipo o "snapshot" não

é, vocês meteram isso. Gosto disso de alguma maneira, não é, principalmente eu que tou na qualidade de dados, não é. Tenho que fazer testes sobre os dados que estão a ser disponibilizados, e isso, esse modelo do Data Vault, embora o (eliminação de conteúdo sensível) goste muito dele, e eu acredito que ele em termos de futuro desde que seja bem documentado, desde que hajam ferramentas que simplificam de alguma forma a gestão desse modelo, não é. E a forma como nós vamos buscar os dados desse modelo. Que as ferramentas é importante, já nesse nível de complexidade, desde que isso exista, eu diria que o modelo que vocês apresentaram é simples qb (quanto baste) e (…) considerando que tamos a falar de Data Vault, não é. Considerando que estamos a falar de Data Vault acho que sim. O modelo que vocês apresentaram é simples.

I: Como classificaria a robustez do modelo, na sua capacidade de adaptar tanto aos requisitos de negócios apresentados, tanto aos que possam surgir no futuro?
P1: Sim, aqueles que vocês apresentaram(…) (atendeu uma chamada) (…) aquilo que vocês apresentaram, pareceu-me que sim, não é. Agora, a novos outros requisitos, isso já é entrarmos na bola de cristal, não é. Depende do requisito que surja agora no futuro. Parece-me haver uma certa margem de adaptabilidade e exponencialidade do modelo, agora, tar aqui a fazer previsões no futuro, e se vai-se adaptar ou não, não sei. Agora no presente sim, e aqueles casos que vocês apresentaram, pareceu-me, pareceu-me bem.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P1: Sim, prova de conceito sim, falta aí muita coisa como é evidente, como já vos disse, mas sim, aquilo que vocês reforçaram têm umas ideias giras. Até porque nós já tivemos aí, nós, quer dizer, a parte que tá aí com os modelos de arquitetura, etc. Já teve umas abordagens e a coisa não foi assim muito produtiva, portanto apresentaram aí. Portanto, o que vocês apresentaram gostei de ver, parece uma coisa simples, não fui minuciosamente ver as relações que vocês têm, etc., isso já levava muito mais tempo, mas daquilo que olhei pareceu-me tudo bem.

I: De que forma considera a existência do modelo proposto pertinente e/ou importante?
I: O modelo proposto por quem, por vocês ou –
P1: Sim, o que estivemos a apresentar, o que nós apresentamos de que forma é que ele pode ser pertinente ou importante, dentro do contexto do banco?
I: É assim, dentro dos meus conhecimentos, que são assim [partes?] relativamente a Data Vault. Partes, quero dizer que conheço alguns mas não conheço na profundidade, que vocês com certeza ficaram a conhecer (…) O modelo tem relevância, parece-me que há aí soluções que podem ser aproveitadas até para outras coisas e mesmo a nível de objetos. Vocês acho que exploraram bem os objetos do modelo Data Vault, portanto, há aí mais valias aí sem dúvida, para aplicar até a outras situações.

I: Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros, e analistas de dados da organização?
P1: Se nós adotarmos o modelo do Data Vault, sim com certeza.
I: Que recomendações/sugestões daria para melhorar o modelo?
I: Sendo que, já deu algumas, mas pronto.
P1: Sim, tirando aquilo que eu disse, é como vos digo, tirando aquilo que para mim foi mais impactante naquilo que vi, tudo o resto já carecia de uma análise muito mais profunda, e sinceramente eu não a

fiz. E portanto, eu não vos posso dar esse contributo, dessa forma não vou tar aqui a inventar coisas para vos tar a dizer.

I: Que outros comentários pode fornecer sobre o modelo proposto?
I: Se tiver algo a dizer –
P1: Não, não. Aquilo que tinha a dizer já fui dizendo. Vocês estudaram isso melhor que eu, portanto há coisas que já me foram apresentadas e eu concordo. Vocês de certeza tiveram a ver milhares de soluções (…) portanto, acho que aquilo que vocês apresentaram está bom, relativamente àquilo que vos foi proposto. Não tenho assim mais a acrescentar.

APPENDIX D: INTERVIEW 2

I: Em termos de beneficiários de contas-cartão e as relações que estão associadas a isso, considera que o modelo reflete o negócio de forma precisa?
P2: Sim, né? Eu precisava ver o modelo agora sim, se pudesse me mostrar, para dar uma olhada.
I: Sim, claro, claro (…).
P2: Parte dos beneficiários, né? Pronto, então temos aqui, vamos pegar pelo "hub" principal, que foi aquela parte complexa. Temos o cliente, o cliente pode ser um beneficiário, mas um beneficiário (…) nem todos os beneficiários são clientes, né? Aqui o meu colega (eliminação de conteúdo sensível) até comentou, eu não sei se é o caso, eu não conheço profundamente essa área de cartões, mas uma empresa (…) ela pode ter um cartão, né? Um cartão da empresa, mas que é utilizada por exemplo por vários sócios, então sim, são vários beneficiários, né, mas o cartão provavelmente é da empresa. O cliente aqui seria um cliente empresa e os beneficiários não, né? Seria um beneficiário, mas eu não sei exatamente como funciona em termos de negócio, mas sim, do meu ponto de vista, em termos de desenho, tá correto, que é, um cliente é- pode ser sempre um beneficiário, agora um beneficiário nem sempre será um cliente, não é?
I: Daí este número de beneficiá-
P2: Exatamente. Certo. Depois temos aqui em cima a conta corrente, que tá ligada ao cliente. Temos aqui a parte toda, conta corrente, o hub de conta corrente e temos o hub do beneficiário com a conta corrente, certo, que é esse "link", tá bem. E vai dar no de cima (…) Pode subir um pouquinho mais para eu ver? Certo. E é o… a conta com os beneficiários da conta corrente. Certo.
I: Mm, mm. Sim, é basicamente… é aquela tabela auxiliar dos beneficiários.
P2: A (eliminação de conteúdo sensível), sim.
I: Sim, essa aí já… esses atributos já têm todos a relação com uma certa conta corrente e portanto pusemos como atributos deste link, porque esta tabela já tem tudo. Tem beneficiários, tem a conta-cartão e tem a conta corrente.
P2: Tá-me aqui a fazer confusão é que um beneficiário, não sendo um cliente, não tem conta corrente, certo?
I: Certo.
P2: É o cliente quem tem conta corrente, né?
I: Sim, sim.
P2: Temos que ter esse cuidado
I: E o cliente tá ligado aqui por este link.
P2: Certo, então (…)
I: Aqui é só porque a nível da fonte de dados e dos metadados que nós vimos (…)
P2: Sim.
I: Onde há a tal chave estrangeira da conta DO-
P2: Então ele não é beneficiário da conta corrente, ele é beneficiário da conta-cartão.
I: Sim, sim, sempre da conta-cartão. É só porque a relação com a DO é feita nesta tabela que também tem informação dos beneficiários.
P2: Ah, certo! Tá bem.
I: Nós decidimos manter assim só porque tamos a representar o negócio como ele existe agora-
P2: Certo. Tá bem. Certo. Tem os beneficiários do hub beneficiários e tem o satélite dos beneficiários conta corrente, que se ligam com a conta corrente, né? Certo. Percebi. Tá bem, tá bem. É porque eu

tou ainda aqui com o desenho do Data Vault ainda me… familiarizando. Percebi. Temos o satélite só do beneficiário e temos o satélite do link beneficiários com (…)
I: Conta corrente.
P2: Conta corrente.
I: Sim, sim, sim. Exatamente.
P2: Com o hub de conta corrente. Que é esse que eu tenho um pouquinho de dúvida. Certo.
I: Mas sim, mas também, pode fazer comentários que não sejam tão positivos (riu-se).
P2: Sim, é que essa questão aqui ainda tá me fazendo espécie, certo? Porque, como disse, o beneficiário não me parece (…) que essa parte de cima exista. Que é (…) eu tenho (…) Esse link L beneficiários sim, é beneficiário com "customer", certo? E se esse… esse satélite que tá lá em cima, ele pra mim tinha que tar aqui nos beneficiários, no L beneficiários, e não no beneficiários "current account", percebe? Porque eu não tenho link dos beneficiários com conta corrente.
I: Ah (…) nós-
P2: Percebe?
I: Sim, sim, sim, nós-
P2: Eles têm é- O beneficiário é um cliente também, então ele tá nesse link customer conta corrente, current account, mas não há uma relação do beneficiário, da "business key" beneficiário, com a conta corrente.
I: Ok-
P2: Percebe? Temos que ter esse cuidado, eu não tive na análise da (eliminação de conteúdo sensível) e nem conheço bem essa área, mas tá me fazendo espécie, porque se eu posso ter um beneficiário (que) não é cliente, como é que existe esse hub L beneficiários current account, percebe?
I: Nós também nos fez um bocado de espécie, nós também não percebemos muito bem, mas a verdade é que era essa tabela que tinha como chave o número de beneficiário-
P2: Certo.
I: O número da conta-cartão-
P2: E o- Pois é.
I: Essa tabela é que tinha a conta DO.
P2: Certo.
I: Era essa.
P2: Tou percebendo.
I: Então pronto-
P2: Que foi a bendita (eliminação de conteúdo sensível). Percebo.
I: Sim.
P2: Percebo.
I: Embora não faça muita… muito sentido, era como estava.
P2: Certo. Tá bem, é isso mesmo.
I: Mas pronto, eu percebo, eu percebo.
P2: Percebe né? Como é que (…) Então é porque eu não respondi (…) Na verdade a preocupação que eu tava quando comecei a entrevista com vocês era exatamente porque nós nunca chegámos a confirmar se todos os beneficiários eram clientes, percebe? Porque é como você- Tá certa. A (eliminação de conteúdo sensível) é onde tem o trio, e quem é o trio? O trio é a conta DO, a conta-cartão e o beneficiário.
I: Mm, mm. Exatamente.

P2: E é estranho, é isso que é estranho, ele podia estar aí (…) Nós temos que ver mas é (segmento de texto incompreensível) É se esse beneficiário é o cliente. O beneficiário cliente.
I: Sim, sim, sim. Até pode ser.
P2: Ou então, como eu não cheguei a responder a vocês, que é, todos os beneficiários são clientes? São obrigados a ser clientes? Percebe?
I: Pois, até é possível. Pois.
P2: Pronto, se depois quiserem ainda resolver essa dúvida a gente depois fala, mas sim, já percebi, tá ótimo. É isso mesmo!
I: Pronto, então em termos dest-
P2: Sim, sim. Indo de acordo com as nossas fontes de dados, sim.
I: Exato.
P2: Certo.

I: Ok, ok, pronto. Depois, em termos de titularidade de contas, no geral, considera que o modelo reflete o negócio de forma precisa?
P2: Sim, esse não há dúvidas.
I: Ok, ok (riu-se).
P2: Esse não há dúvidas né, porque esse não tem nenhum problema. Há um cliente que tem uma titularidade e se relaciona com a conta, com essa titularidade. Perfeito.
I: E neste caso pode haver vários tipos.
P2: Exatamente.

I: Ok, em termos da relação entre os clientes, considera que o modelo reflete o negócio de forma precisa?
P2: Também perfeito. Que é uma relação de um cliente com outro cliente. E aí temos tutores, temos responsáveis, temos vários (…) Ou mesmo numa conta empresa, um cliente é conta empresa, qual é a função dele? É administrador? É sócio-gerente? É gerente? (…) Tá ótimo. É (segmento de texto incompreensível)? Tá ótimo.
I: Agora, de uma forma geral, tendo em conta os conceitos de Data Vault, acha que os conceitos de negócio modelados foram representados de forma precisa?
P2: Pode repetir? Desculpa. Pode repetir?
I: É aquela pergunta, só de uma forma geral, se o modelo de forma geral representa o negócio de forma precisa.
P2: Sim, eu acho que-
I: Respeitando as práticas de Data Vault.
P2: Sim, com perfeição. Principalmente na distribuição dos satélites, porque um dos maiores problemas que nós temos no Data Warehouse é tratarmos as tabelas com a democracia que não se aplica, ou seja, eu trato tabelas sempre com a mesma importância quando elas não têm. Elas têm importâncias distintas. E trato também… trato todas as tabelas com o mesmo grau de mudança, ou seja, de guardar histórico, e trato todos os campos com essa mesma importância, quando na verdade o Data Vault tem aqui alguma ênfase em… o que é que eu utilizo mais e o que é que eu utilizo menos… pra eu separar, pra eu não tar precisando taxar tempo e fazendo atualização de tabelas que não são necessárias. Eu acho ótimo, tá muito bom.

I: Então, considerando apenas os conceitos de negócio que foram modelados, como é que classificaria o modelo em termos de completude?
P2: Sim, então vamos falar um pouquinho sobre isso. Nós colocámos clientes e nós colocámos a conta corrente, que nós chamamos costumeiramente a conta depósito à ordem, colocámos os cartões, principalmente cartão de crédito (segmento de texto incompreensível). Depois colocaram também a conta… os empréstimos, não é? E colocaram (…) Ou seja, dentro do cofre, nós colocamos a parte que entra, que é a parte de depósitos, depósitos à ordem e depósitos a prazo. Em termos de saída, nós colocamos empréstimos e os cartões. Eu acho que sim. Duas entradas e duas saídas e uma conta que administra essas entradas e saídas. Eu acho que tá ótimo.

I: Ok. Na sua opinião, o modelo é simples de entender e utilizar?
P2: Pra mim sim, eu (…) e até quando escolhi o meu curso era porque gostava muito de matemática e (…) mesmo quando fiz a faculdade, a parte que mexeu mais comigo foi a parte de base de dados, porque pra mim era quase teoria dos conjuntos e pra mim era muito fácil ver bases de dados. Quando veio agora o Data Vault separando esses conjuntos com essas características, pra mim ficou até mais simples do que os outros relacionais, porque essa característica de separar chaves para um lado, atributos para o outro e relações para o outro, essa visão de conjunto matemática pra mim fica muito simples.

I: Agora, como classificaria a robustez do modelo quanto à sua capacidade de se adaptar tanto aos requisitos de negócio apresentados, quanto aos que possam ser apresentados no futuro?
P2: Também eu acho, do que eu li, essa é a mais valia do Data Vault. Tem que lembrar sempre, e eu sou apologista disso, e defendo essa ideia, que é (…) parece que há duas correntes muito distintas no que diz respeito a guardar dados e utilizar dados que é (…) Para mim o Data Warehouse, ele tem uma linha que divide muito bem que é guardar os dados, e outra coisa é utilizar esses dados, e para mim essa linha, ela separa muito bem a forma de eu guardar e depois a forma de eu utilizar. Para mim, o Data Vault, do que eu li e do que eu conheço das outras modelagens, ele parece-me, e eu não tenho a experiência, ser a melhor forma de guardar. É mais difícil na hora de tirar? Pode ser. Mas hoje há tantas outras ferramentas para tirar dados e se desenvolveram muito mais ferramentas de extrair dados, de manipular dados, do que metodologias de guardar e metodologias de desenhar os dados para serem guardados. Então sim, para mim tá robusto e faz todo o sentido para guardar os dados de uma companhia, de uma empresa. Para extrair não, vamos para outros modelos. Mas para guardar, eu acho que é extremamente flexível e robusto.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P2: Sim, principalmente para a criação de um Data Warehouse de grande dimensão. E ainda para empresas com muitas empresas adquiridas, ou seja, com fontes de dados distintas muitas vezes para os mesmos dados.

I: Agora algumas perguntas mais gerais. De que forma é que considera a existência do modelo proposto pertinente e/ou importante?
P2: A minha dúvida maior sobre o Data Vault é que (…) normalmente nós já temos um Data Warehouse e custa as pessoas, principalmente pela multiplicidade das tabelas né, pela dimensão que as pessoas pensam em tabelas, mas eu acho que se fosse me dito hoje que eu tinha que criar um

Data Warehouse novo para uma empresa, eu não teria menor dúvida que escolheria o Data Vault nesse sentido, por ser o mais flexível e o mais simples matematicamente de desenhar. É o que eu acho. E eu acho que ele não é mais utilizado, porque quem já tem um Data Warehouse tem dentro de um modelo, ou de um modelo Inmon ou de um modelo Kimball, e aí você fazer migração de modelos eu nunca vi e aqui tá quase acontecendo. Mas você criar um novo para mim fazia todo o sentido que fosse no Data Vault. Não sei se respondi bem à pergunta.
I: Era mais no sentido de ser importante (…) Qual era a importância ou pertinência se isto fosse implementado ou uma versão mais completa disto.
P2: Ah, sim. Novamente para mim, eu acho que da experiência que eu tenho com outros modelos, principalmente com o Inmon, eu penso que esse modelo seja muito mais rápido de implementar.
I: Ok, pronto, é a vantagem maior.
P2: É.

I: Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros, analistas de dados da organização?
P2: Para arquitetos (…) Eu não sei qual é a nomenclatura correta hoje utilizada. É para ser um modelador de repositório de dados para o uso analítico. Porque hoje os nomes tão mudando, Data Lake, Data Warehouse, Lakehouse, etc. Eu acho é que a profissão que é o modelador do repositório de uma companhia, esse modelo é extremamente importante. Ou seja, qual é o nome que nós tamos dando se é engenheiro de dados, se é arquiteto de dados, se é o modelador de dados do Data Warehouse ou do repositório de Warehouse... Para os demais, eu acho que vai demorar um pouquinho, que é, para quem faz análises de dados, eles vão- Vai custar a perceber, vai custar a entender como é que nós distribuímos em tantas tabelas. Porque ele vai achar que ?(picou)? ainda mais né, que normalizou demais, mas eu acho que é tudo também uma questão de hábito, porque nos primórdios, quando eu comecei a trabalhar quase nem havia modelagem relacional, era quase tudo muito flat, né, era “flat files”. É uma questão de hábito. Agora, para um modelador, a profissão que se identifica como sendo a pessoa que vai desenhar o modelo para conter toda a informação de uma empresa, com histórico, ele tem que entender hoje disso, tem que conhecer o Data Vault.
I: Então acha que é mais útil para esses arquitetos de dados e pessoas de modela-
P2: Para os arquitetos de dados.
I: …do que propriamente para os analistas?
P2: É, não, um analista ele vai… Quer dizer, hoje nem tanto porque nós tamos também voltando tanto… O Data Lake agora é tudo tão distribuído que eu acho que eles não vão sentir tanto mas eu acho que o Data Vault é direcionado para modeladores de repositórios de dados, que são utilizados para tudo, para BI, para IA, para tudo o que for “analytics”. Porque aqui você consegue guardar rapidamente, se adaptar rapidamente, entrando ESGs, entrando o RGPD, entrando qualquer outro tipo de característica. Nos modelos atuais que nós temos, os Inmons e os Kimballs, o impacto dessas mudanças é muito sentido e acho que aqui no Data Vault ele entra normal… não tem impacto. Como ele é mais rápido, no meu entender, você rapidamente disponibiliza rápido para os analistas de dados, porque nos Inmons da vida e nos Kimballs você demora para ter dados prontos para entregar aos analistas de dados porque você vai meter no modelo, vai ter que entender toda a lógica.

I: Que recomendações/sugestões daria para melhorar o modelo?
P2: Pronto, ah, como falamos da profissão, para fazer um modelo, nós temos de conhecer o negócio, né, então, a sugestão é que seja sempre muito acompanhado, o modelador seja acompanhado dos

responsáveis técnicos das aplicações. Eu acho que, sendo para mim o Data Vault um modelo muito matemático, de fácil distribuição das tabelas até lendo para as fontes e distribuindo, mas eu acho que o ideal é sempre muito bom, muito bom, em qualquer modelagem que seja, indiferente de ser Data Vault ou não, é bom você conhecer o negócio. Porque essas dificuldades que eu penso que nós tivemos aqui veio muito da falta do meu próprio conhecimento do negócio, né, de ajudá-los. Então, é sempre bom que fique na cabeça de todos, eles e o modelador, ele tar muito próximo das fontes de dados e das pessoas responsáveis por essas fontes de dados. Se forem pessoas com um bom conhecimento de bases de dados, mesmo que seja relacional, ela pode identificar inclusive os problemas que ela teve enquanto base de dados relacional, como nós tivemos aqui essa questão do beneficiário, ou mesmo a questão do cartão. O cartão tá ligado a uma conta só ou tenho vários cartões para uma conta de cartão, uma conta DO? Então a recomendação é que, sendo um arquiteto de dados, ele tem que tar sempre muito aberto a ouvir os responsáveis pelas fontes. Não pode ser também essa matemática tão simples, olha, chave para um lado, atributo para o outro e relações para o outro. Não, porque, eu acho que a otimização do modelo, ela á muito ligada também ao conhecimento do negócio, então… arquitetos… sempre… Porque você modelar um banco é uma coisa, você modelar uma seguradora é outra coisa, você modelar uma empresa de telecomunicações é outra coisa… Então, eu acho que o Data Vault é muito democrático nessa parte matemática, isso é maravilhoso, é o que também traz muita rapidez, mas eu acho que, como… Porque uma coisa é você modelar uma aplicação. Quando você modela uma aplicação, por exemplo, se vocês, imagina, nós pegamos aqui cartão de crédito, você vai modelar cartão de crédito, só cartão de crédito. Liga com as outras, tudo bem, mas você tá modelando cartão de crédito, você se especializa em cartão de crédito. Quando você é um modelador de um repositório de dados, que antigamente se chamava Data Warehouse, você é um modelador de um tipo de indústria. Você não é um modelador de empréstimos, você não é um modelador de depósitos a prazo, você é um modelador de banca, você é um modelador de telecomunicações. Então, é ter consciência também de que um arquiteto de dados, um modelador de dados, um engenheiro de dados, de modelagem de repositório de insdústria, mesmo sem ter Data Vault, ele vai ter que escolher uma área para também se especializar, porque se você modelar uma indústria farmacêutica, né, você vai modelar em Data Vault o repositório de dados de uma indústria farmacêutica… completamente diferente, apesar do modelo ser matemático, é diferente de você modelar um outro tipo de indústria. Então a recomendação é, também procurem a área que você se identifique ou… não sei… e se especialize. Porque quando nós vamos ao mercado de repositório de dados, existem já modelos criados para determinadas indústrias. Eu vou buscar um modelo Data Warehouse para a banca, eu vou buscar um modelo Data Warehouse para a indústria farmacêutica, eu vou buscar um modelo Data Warehouse para aviação, percebe? O modelo tá ligado à indústria.

I: Certo. E há alguma recomendação assim mais específica para o nosso modelo, para o que nós propusemos?

P2: Não, eu acho que fizemos bem e foi como eu disse, eu acho que nós fizemos partes pequenas do banco, mais fizemos, né, fizemos algumas partes mais comuns, que é, a minha conta corrente, depois o meu cartão, que é o que hoje todo o mundo tem, depois um depósito a prazo, que é muito comum hoje as pessoas pensarem na reforma, depois empréstimo, que vocês provavelmente daqui a pouco vão tar a entrar numa conta dessa (riu-se), então assim, eu acho que ficou bem colocado. Começámos pela parte simples… claro que aí nas “table reference” é um mundo à parte, porque nós usamos duas aí que elas vão para além de table reference, que são os produtos e os balcões, eles

fazem parte de outras estruturas, que é a própria estrutura orgânica do banco, os produtos também que é uma dimensão muito maior (…)
I: Há alguma recomendação mais específica?
P2: Não, não, não, porque um banco vende muita, muita, muita, muita coisa que nós não fazemos ideia e eu acho que o vosso modelo está na modelagem do nosso dia-a-dia. Quem é que não tem esses negócios de banco que vocês puseram aí? Nós podíamos dizer que outros tantos, né, cofres… O banco vende um cofre para você guardar os seus tesouros e isso é uma forma e você vai ter que modelar isso. Os cofres, o que é que tem nos co- O que é que tem não, mas a administração que o banco aluga, arrenda para você guardar… Entre outras coisas de ações e muitas, muitas outras coisas, mas são coisas que, por exemplo, eu não tenho, eu nunca utilizei, então ainda seria mais difícil para eu falar sobre esse tipo de negócio. Essas eu acho que tão ótimas.

I: Que outros comentários pode fornecer sobre o modelo proposto?
P2: Não tenho, eu acho que vocês fizeram um ótimo trabalho, principalmente vendo essa parte que vocês colocaram aí de outras características que o Data Vault (…) Ou seja, deixando claro que o Data Vault não deixa nada de fora e inclusive que o criador tem sempre o cuidado de dizer “sim, temos solução para isso, tenham cuidado com a utilização, pode não ser bom para a performance, etc.”. Temos as bridges, que é um facilitador entre as duas áreas que eu falei anteriormente que é a área de guardar dados com a área que vai utilizar dados. Eu acho que tá tudo.

APPENDIX E: INTERVIEW 3

I: Em termos de beneficiários de contas-cartão, considera que o modelo reflete o negócio de forma precisa?
P3: É uma boa pergunta. Podemos voltar se calhar lá aos beneficiários só para (…)
I: Claro, tendo em conta que o modelo é bastante simples, mas de acordo com o que está apresentado, basicamente é essa a ideia.
P3: Sim.
I: (apresenta o diagrama do modelo) É mais aqui esta parte, diria.
P3: Sim, eu aqui só tenho aquele tema que tu até já tinhas referido, Inês, que seria… o ideal seria termos aqui umas entidades, não é, algum "hub" de entidades (…) se calhar faria aqui um bocadinho mais de sentido, até para agregarmos todas as entidades que temos aqui no banco, mas para aquilo que vocês têm aqui parece-me tar bem, não tenho aqui nada que acrescentar. Até porque também, a nível de beneficiários, confesso que não é bem aqui a… (riu-se).
I: Tudo bem (riu-se). É de acordo com o conhecimento que tem.
P3: Sim, sim, sim.

I: Em termos da titularidade de contas, aquela questão que nós modelámos com os satélites multi-ativos, considera que o modelo reflete o negócio de forma precisa?
P3: Sim. É assim, conforme estavas a referir, um cliente pode ter n tipos de intervenção numa conta e em diferentes contas, por isso, ao estar a contemplar, recorda-me lá aqui em cima…
I: Era a questão do "ownership type", é uma coluna que faz parte também da chave, para além da chave da relação de cliente com conta.
P3: Mm, mm.
I: Esta chave "customer credit card account" que passa para aqui já tem o cliente e a conta-cartão e depois, pronto, para além da "load date", que faz sempre parte da chave nos satélites, é mesmo o standard, ainda temos o "ownership type", que assim permite que para cada cliente-conta possa haver vários tipos de titularidade. Se não tivéssemos essa chave, isso não poderia acontecer.
P3: Sim. Mas tu não estavas a falar do tipo de titularidade que vem da… posso estar a fazer confusão… da (eliminação de conteúdo sensível)?
I: Sim, que é da relação cliente-conta, ou seja, neste caso é cliente com conta-cartão, mas também podemos ver da à ordem, mas a ideia é a mesma.
P3: Sim, ou seja, qualquer que seja o tipo de titularidade, tá ali associada a-
I: Sim, exatamente. E a ideia seria, qualquer outra conta que fosse, fosse à ordem ou o que fosse, haveria sempre este satélite.
P3: Esse cenário. No fundo, dá para adaptar a qualquer cenário.
I: Exato.
P3: Sim. É isso. Ok.

I: Em termos e relação entre clientes, considera que o modelo reflete o negócio de forma precisa?
P3: Qual é que é? Re-
I: Esta aqui, "customer" com customer.
P3: Mm, mm. Depois vocês no satélite têm o quê? A data início e fim de relação e-
I: Sim, para já só temos estes, que é a percentagem de participação, início de relação e fim de relação. Se não forem todos falta um, por aí.

P3: Ah, ok. Sim, a nível de relação, também é muito simples, não é? No fundo, é identificar o tipo de relação que existe entre os dois clientes, sim.
I: E o código de relação até acaba por ter já essa informação de qual é que é a relação.
P3: O que depois poderia ser interessante, mas isto lá está, como vocês têm aquelas tabelas de descodificação, que vocês acabaram por não usar muito, aquelas "reference tables", este tipo de código, por exemplo, é uma informação que pode ser descodificada.
I: Sim.
P3: Tudo o que é códigos depois pode ter essa abordagem também.
I: Sim, focámo-nos nos principais, os mais importantes.
P3: Sim. Sim, mas os mais importantes… Até porque quem vai ver depois não sabe o que é que significa aquele código… Se tiver o descritivo do código…

I: Claro. De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete o negócio de forma precisa, cumprindo as regras do Data Vault?
P3: Sim, eu acho que sim. Eu só deixava aqui aquelas notas que falámos aqui durante a apresentação. Aquela situação de multiempresa que acontece muito, mas que, na minha opinião, deveria existir ali um código empresa, uma descrição da fonte, por exemplo (eliminação de conteúdo sensível). Acho que talvez fizesse sentido existir esse código empresa. E aquilo que me salta mais aqui, que embora não esteja aqui no âmbito, e eu percebo isso, o que me salta aqui mais à vista são estas descodificações que acreditem que depois têm muito impacto em quem vai fazer as análises.
I: Pois.
P3: São as duas coisas que me saltam aqui mais logo assim (…) São esses dois pontos.
I: Considerando então apenas os conceitos de negócio modelados, mais uma vez, como classificaria o modelo em termos de completude?
P3: Mas… como assim? Que classificação (…)

I: Quão completo é que considera o modelo tendo em conta o que nos foi fornecido de informação?
P3: Sim. Eu acho que vocês tiveram aqui muito trabalho e tá bastante completo para a informação que vocês tiveram e para a informação que vos foi fornecida.

I: Na sua opinião, o modelo é simples de entender e utilizar?
P3: É simples de entender, é simples de utilizar e é simples de alterar e modificar alguma coisa que seja necessário de ajustar, também é simples de o fazer.

I: Como classificaria a robustez do modelo, na sua capacidade de se adaptar tanto aos requisitos de negócio apresentados, como a requisitos que possam ser apresentados no futuro?
P3: É assim, restringindo só mesmo aqui a este âmbito aqui, eu acho que, seguindo aqui um bocadinho aquilo que eu estava a dizer há pouco, acho que é muito simples de incluir e de alterar aqui qualquer ponto. Nesse sentido, acho que também tá bem conseguido.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P3: Isso é que seria o grande desafio (riu-se). Mas sim, acho que era um excelente ponto de partida para começar assim um projeto.

I: De que forma considera a existência do modelo proposto pertinente e/ou importante?
P3: Mas o que é que vocês pretendem saber exatamente aqui? Como assim?
I: Como é que poderia ser aplicado ou qual é que seria a importância se fosse implementado dentro da empresa ou de que forma é que poderia ajudar. E mesmo que não seja feita a tal implementação, mas conceitos extraídos que possam ser aplicados…
P3: É assim, considerando… Eu vou comparar um bocadinho também com o que temos hoje em dia implementado, ok? E, eu conhecendo mais ou menos o que existe hoje e olhando para o vosso modelo aqui, para mim a grande mais-valia e a grande importância que isto poderia ter seria mesmo a facilidade com que seria possível efetuar alterações. Vocês podem não ter bem noção mas hoje em dia, cada vez que queremos alterar ou incluir um atributo, aquilo é um processo que ainda é longo e é uma coisa que devia ser muito mais linear do que é hoje em dia e, sem ter visto nada implementado do Data Vault, mas olhando aqui para o modelo, se isto funcionasse como eu acho que funcionaria, de facto ia ser uma mais valia e… tanto que existia até o objetivo aqui inicialmente no (eliminação de conteúdo sensível) ou há uns tempos atrás mudarmos aqui um bocadinho também para o Data Vault exatamente por vermos essas mais valias aqui neste tipo de modelação.

I: Na sua opinião o modelo proposto pode ser útil para arquitetos, engenheiros, analistas de dados da organização?
P3: Sim, isso pode ser de certeza absoluta. Muito útil, a sério. Acho que vocês fizeram aqui um excelente trabalho.

I: Obrigada. Últimas duas perguntas. Que recomendações ou sugestões daria para melhorar o modelo?
P3: Eu acho que isso já respondi mais ou menos, não é? Eu continuo sem perceber, olhando para estes aqui, que a Inês me disse que estavam aqui estes dois separados aqui em baixo, mas continuo com alguma dúvida como é que isto depois se podia integrar no- É que não faz sentido estarem ali duas tabelas soltas ali. Aquilo de alguma forma tinha de ser ali interligado, para conseguirmos fazer a análise depois dessa informação. E no fundo as melhorias, foram aquelas que já referi, assim de repente não me recordo de mais nada.

I: Que outros comentários pode fornecer sobre o modelo proposto? Se existirem.
P3: Não tenho assim mais comentários.

APPENDIX F: INTERVIEW 4

I: Em termos da parte dos beneficiários de contas cartão e relações associadas, considera que o modelo representa o negócio de forma precisa.
P4**:** Era aquilo que eu dizia, o que vocês fizeram tenho de estudar, tenho de estudar isto realmente, porque aqui a parte em que vocês fazem a ligação entre conta, conta do, conta cartão e beneficiário, parece tar muito complicado, e parece que é mais simples pelo que eu vi nos dados. Não consigo responder a dizer que está, eventualmente estará, mas acho que pode ser simplificado.

I: Em termos de titularidade de contas, ou seja, aquela questão dos satélites multi-ativos que nós explicamos para guardar a titularidade do cliente com uma conta, considera que o modelo reflete o negócio de forma precisa?
P4: Pareceu-me que sim, digo pareceu-me porque não tou, não tamos a ver os dados.
I: Se quiser podemos ir mostrando o modelo.
P4: Vocês não fizeram, não fizeram, não implementaram pois não, é tudo muito pela teórica?
I: Não, infelizmente não tivemos acesso ao "data box" que era suposto (…)
P4: Mas pronto, pareceu-me que estava ok.

I: Em termos da relação entre clientes, considera que o modelo reflete o negócio de forma precisa? Portanto, é a questão do cliente com cliente.
P4: Em termos de relação com clientes (…)
I: - Entre dois clientes. Eu posso partilhar outra vez se for preciso.
P4: Relação dos clientes (…). Para o mesmo cliente (…)
I – Era este link aqui basicamente que faz tal a relação entre dois clientes que veem do "hub customer", e depois dá-lhe um código de relação para distinguir a relação entre eles –
P4: À sim, sim, essa descodificação dessa relação, pareceu-me porreiro, tava a pensar mais por terem o mesmo número de contribuinte e terem, e ser o mesmo cliente, aí a relação tava-me a fazer um bocado de confusão. Aqui é para clientes e clientes, com uma relação. Sim, acho que tá ok, acho que é isso mesmo.

I: De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete o negócio de forma precisa? Considerando o Data Vault?
P4: Sim, é nesta parte que tamos assim aqui a ver, relação de clientes com contas, relação de clientes com clientes, relação de (..) das contas com os cartões, parece-me que sim, salvaguardando aquela parte dos beneficiários, é aí que eu tou com dúvidas só nessa parte. O negócio de empréstimos é muito mais complicado do que isto, do que a parte que mostraram certo?
I: Sim, sim, claro
P4: Pronto, à bocado estavam a mostrar ali o parte dos empréstimos, se implementarem empréstimos, o empréstimos é uma aplicação, só por si é uma aplicação.
I: Pois –
P4: Certo, por isso é que eu estou a dizer, tá aqui assim bastante mais simplificado, é lógico que vocês aqui já estão (…) também estou a exagerar um bocado, também porque eu aqui tou dentro de tudo (…) mas vocês vão já aí buscar os dados depois do tratamento e carregar as tabelas, que é só isso que têm a fazer mais nada, por isso tá, é isso que tá bem.

I: Considerando apenas os conceitos de negócio modelados, ou seja tendo em conta o âmbito bastante reduzido do nosso modelo, como é que classificaria o modelo em termos de completude. Ou seja, quão completo é que é, tendo em conta as circunstâncias?
P4: Sim, eu acho que está bastante completo, para o que foi apresentado que está completo, é isso que têm de fazer, especialmente o principal acho que é ter os clientes bem definidos, as contas definidas, a relação desses dois e acho que isso aí tá muito completo.

I: Na sua opinião, o modelo é simples de entender e utilizar?
P4: Mm, mais ou menos (riu-se), pronto é assim, eu ainda tenho de estudar para perceber bem, mas é, tem muitas tabelas, e as ligações dos "links", muitos links, aquilo tem de se estudar bem para perceber, mas acho (…) acho que tenho de estudar, não posso dizer ou agora que sim ou que não.

I: Como classificaria a robustez do modelo, na sua capacidade de se adaptar tanto aos requisitos demonstrados, tanto a futuros requisitos que possam existir?
P4: Se for assim tão simples, tão direto como vocês apresentaram, acho que tá, tá bem, é so acrescentar (…) um "hub" e um "link", para acrescentar novos, novos dados, parece-me que está (…) está robusto, mas isto aqui tem que ser sempre testado não é?
I: Claro, claro, e idealmente seria isso, mas não conseguimos.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P4: Sim representa, acho que sim.

I: Mm, ok, esta aqui é um bocadinho mais, pronto, subjetiva, mas, de que forma considera a existência do modelo proposto pertinente e/ou importante?
P4: Eu acho que era muito importante e pertinente, mas isto aqui foi importante, inicialmente era suposto acontecer aqui, aqui no (eliminação de conteúdo sensível), e era isso que fazia sentido, vocês iam fazer um protótipo basicamente, iam fazer isso para nós implementarmos uma aplicação bancária inteira, por isso acho que é muito pertinente, e muito importante. E especialmente, porque acho que em termos de performance, isto ia melhorar muito aqui assim o banco, por isso é que eu perguntei essa performance, porque a ideia que eu tinha é que ia ser o principal (…) era a performance.
I: Sim, especialmente da parte de guardar os dados, não tanto talvez na extração, mas –
P4: Eu acho que o acesso é mais difícil –
I: Sim, sim
P4: É mais difícil, mas ao mesmo tempo vocês também demonstraram que têm assim, hipóteses, de que, de que esse, de criar tabelas auxiliares para facilitar esses acessos, por isso acho que haverá soluções para muita coisa.

I: Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros e analistas de dados da organização?
P4: Sim, já é uma pergunta assim um bocado, sim acho que sim, desde que se saiba, que consiga aceder aos dados (…) sim, não sei, não sei, não sei responder a esta aqui assim.
I: Sim, isto é perguntas um bocado de resposta aberta –
P4: Pois, é que esta aqui assim, para análise de dados, para análise de dados (…) não sei, não sei não vou responder, não sabe não responde (riu-se)
I: Ok, só faltam mais duas.

P4: Que recomendações ou sugestões daria para melhorar o modelo?
I: É aquela parte que já falei, vou analisar para ver se consigo fazer assim uma sugestão para melhorar em termos daquela relação de beneficiários e a cartões e contas, mas é, é uma incerteza que tenho neste momento, não sei se estou com razão em ver algum problema ou não.

P4: Ok, por último, é mais aberta. Que outros comentários pode fornecer sobre o modelo proposto?
I: Ui (…) Não tenho –
P4: Se houverem, sim
I: (riu-se) não sei (riu-se), acho que já não, já não posso acrescentar mais nada, (segmento de texto incompreensível) acho que falámos quase tudo.
P4: Sim, também acho que sim. Mas pronto é aquela pergunta final, só para ver se falta alguma coisa.
I: Não, mas acho que está porreiro, para pronto, acho que faltava mesmo era uma implementação com alguns de dados e para ver se, é muito mais fácil fazer demonstrações dados.
P4: Claro, sem dúvida.

APPENDIX G: INTERVIEW 5

I: Em termos de beneficiários de contas-cartão e relações associadas, considera que o modelo reflete o negócio de forma precisa?
P5: Sim, acho que sim (…)
I: Pode elaborar, se quiser.
P5: Sim, sim. Mas é assim, vamos lá ver uma coisa, eu não sou propriamente desta área.
I: Ok, ok.
P5: Eu estou na área de governo, portanto isto para mim é tudo tão novo, como para vocês, não é (riu-se).
I: Sim, acredito. Nós também com a Andreína, também tivemos um bocado de dificuldade porque ela às vezes certas coisas também tinha que ir procurar e tudo mais, portanto…
P5: Pronto, ah… Portanto, é assim, eu acho que isto tá muito bom. Eu em relação ao Data Vault tenho as minhas… as minhas… os meus pontos de interrogação, mas… mas só por um motivo, é que eu acho que são muitas tabelas e uma pessoa perde-se um bocado… ah… portanto isto para um utilizador… isto para um utilizador… normal acho que será um bocado… mas vocês já arranjaram ali uma solução… já não sei como é que lhe chamaram…
I: A "Bridge Table".
P5: Isso mesmo, pronto. Eu acho que resolve. Não resolve a cem porcento, mas pelo menos ajuda.

I: Mm, mm. Sim. Então e, em termos de titularidade de contas, considera que o modelo reflete o negócio de forma precisa?
P5: Reflete, reflete, reflete. Conseguem aí obter toda… toda a informação de titularidade, de contas DO. É assim, isto é só contas DO de empresas ou (…) Falaram do banco (eliminação de conteúdo sensível)… é particulares e empresas… é só empresas…?
I: É tudo (segmento de texto incompreensível)
P5: São todas as contas DO, ok.
I: Sim, sim. A ideia era modelar todo o tipo de clientes.
P5: Ok.
I: Ou seja, todo o tipo de contas também DO… desses clientes.
P5: Ok.

I: Em termos da relação entre os clientes, considera que o modelo reflete o negócio de forma precisa?
P5: Sim, eu acho que o modelo tá de forma precisa em tudo aquilo que vocês abordaram, portanto…

I: De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete o negócio de forma precisa, cumprindo as práticas recomendadas de Data Vault 2.0?
P5: Reflete. Reflete o negócio de forma precisa, em tudo aquilo que vocês abordaram. Claro que há aqui outras questões, mas… não foram abordadas, portanto não…
I: Sim, mas se houver alguma coisa dentro do âmb-
P5: Não, não. Não, dentro do… Acho que tá perfeito.

I: Considerando apenas os conceitos de negócio modelados, como classificaria o modelo em termos de completude?

P5: O que é que vocês querem dizer com classificaria… Bom, mau, médio…
I: Sim, sim, não tem de se-
P5: Excelente…
I: Sim, pode ser.
P5: Eu acho que o modelo tá correto. Tá… tá… tá bastante bom. Tá bastante bom.
I: Na sua opinião o modelo é simples de entender e utilizar?
P5: Ah… Lá está, simples de entender e utilizar (…) médio, vá.
I: Por causa da questão das tabelas?
P5: Sim.

I: Como classificaria a robustez do modelo na sua capacidade de se adaptar tanto aos requisitos de negócio apresentados, quanto aos que possam surgir no futuro?
P5: Eu acho que… eu acho que o modelo é cem porcento adaptável. Isto… por aquilo que vocês explicaram, qualquer alteração é fácil de resolver... qualquer… qualquer (…) alteração, qualquer… Não há aqui nada que não seja fácil de se fazer, não é, pelos vistos.
I: Sim, do que nós mostrámos pelo menos.
P5: Sim, do que vocês mostraram, logicamente.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P5: Sim, claro que sim. Sim.

I: De que forma considera a existência do modelo proposto pertinente e/ou importante, dentro do contexto da organização?
P5: É muito importante. Devia ser… Devia ser… Daquilo que eu estou a ver, devia ser uma das coisas a ser utilizada.

I: Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros e analistas de dados da organização?
P5: Sim, acho que sim.

I: Que recomendações ou sugestões daria para melhorar o modelo?
P5: É assim… de repente… não tou a ver nada (riu-se) que possa sugerir, porque como vos disse, não é a minha área, mas eu acho que o modelo tá bastante flexível e bastante… É entendível e acho que resolve várias questões da organização.

I: Que outros comentários pode fornecer sobre o modelo proposto?
P5: Nenhum (riu-se).

APPENDIX H: INTERVIEW 6

I: Em termos dos beneficiários de contas-cartão e das relações associadas, considera que o modelo reflete o negócio de forma precisa?
P6: É assim, a minha área não é bem esta. Eu tou mais ligada ao "data governance" mas, do que eu vi da apresentação, pareceu-me que sim.
I: Sim, pronto, pelo menos focando naquilo que nós falamos do que eram as chaves e tudo mais.
P6: Mm, mm. Certo.

I: Em termos da titularidade das contas, considera que o modelo reflete o negócio de forma precisa?
P6: Sim, até achei que era uma… uma… uma forma de ultrapassar as questões. Uma forma até inovadora.

I: Em termos da relação entre os clientes, considera que o modelo reflete o negócio de forma precisa, portanto, a relação entre dois clientes?
P6: Vocês trataram só clientes particulares, não é?
I: Não, na verdade, a ideia era que todos fossem carregados na mesma tabela, naquela do "hub customer", e depois diferenciamos com os satélites, onde tem os atributos específicos, aí é que temos para empresa, para individuais ou particulares, que é assim que acho que chamam.
P6: É. Nós temos os particulares e empresa. E temos depois outra questão que é o multiempresa. Nós temos várias empresas, não só o banco, como as seguradoras, como (segmento de texto incompreensível) empresas, pronto várias empresas à volta, e normalmente nos nossos esquemas de dados temos de ter sempre em consideração o multiempresa e saber exatamente qual é a empresa que tamos a trabalhar… Mas no vosso caso, vocês tão a focar só uma… uma pequena parte, não é.
I: Sim, mas por exemplo, aquela questão que estava agora a perguntar, que é da relação entre clientes…
P6: Mm, mm.
I: A nossa ideia, pelo menos, e lá está, isto está aberto a críticas, era modelar qualquer tipo de relação, ou seja, seja ela entre empresas ou entre uma pessoa e uma empresa, por exemplo… ou entre empregados de uma empresa, do género, ser um gerente do outro… A ideia é que o modelo seja flexível o suficiente para conseguir (…) Nós temos aquele "relationship type code" que faz com que qualquer que seja o tipo de relação, dá para fazer várias relações e até entre os clientes, pronto, a ideia era essa.
P6: Ok.

I: De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete o negócio de forma precisa, cumprindo as práticas recomendadas de Data Vault 2.0?
P6: Sim, sim. De uma maneira geral, sim.

I: Considerando apenas os conceitos de negócio modelados, como classificaria o modelo em termos de completude?
P6: Vocês têm uma escala ou…?
I: Não, é mesmo resposta aberta… Opinião geral.
P6: Então, bom.

I: Na sua opinião, o modelo é simples de entender e utilizar?
P6: Sim, pareceu-me que, da forma como vocês apresentaram e explicaram, que tá relativamente simples.

I: Como classificaria a robustez do modelo, na sua capacidade de se adaptar tanto aos requisitos de negócio apresentados como futuros, que possam surgir?
P6: Bom. Até porque vocês apresentaram naqueles requisitos, aqueles novos requisitos, várias situações e mostravam que se podia adaptar, portanto bom.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P6: Sim.

I: De que forma considera a existência do modelo proposto pertinente e/ou importante, no contexto da organização?
P6: Muito importante, porque nós tamos também a dar os primeiros passos (riu-se) em implementação, por isso é muito importante ter este tipo de situações que nos ajudam também a ver como é que se pode implementar, não é. É pedagógico.

I: Na sua opinião, o modelo proposto pode ser útil para arquitetos, engenheiros e analistas de dados da organização?
P6: Sim, sim.

I: Que recomendações ou sugestões daria para melhorar o modelo?
P6: É assim, vocês já… já tiveram contacto aqui com várias pessoas que estão até mais habilitadas a dar esse tipo de sugestões. Acho que, tendo em consideração isso, de certeza que já vos deram algumas sugestões nesse sentido. Eu de facto… não é exatamente a minha área, portanto não vos sei dizer assim nada concreto, por assim dizer.

I: Por último, que outros comentários pode fornecer sobre o modelo proposto?
P6: Que outros (…)
I: Se houverem.
P6: Eu não tenho assim nenhum… agora… (riu-se) presente.

APPENDIX I: INTERVIEW 7

I: Em termos de beneficiários de contas cartão e relações associadas, considero que o modelo reflete o negócio de forma precisa?
P7: Sim, a mim parece-me que reflete, e falámos aqui de alguns exemplos. E também reflete outra coisa que talvez no futuro pode vir a ser necessário refletir, que é a dados errados do operacional que se conseguem ainda assim integrar depois no próprio Data Warehouse e marcá-los como errados, isto é. Não impedir a integração desses dados, porque as vezes acontece até de outras fontes, não é. Se pensarmos em consolidar outras fontes de outros bancos do grupo ou de outras empresas do grupo, pode ser relevante. E por acaso estamos a pensar fazê-lo, agora com o (eliminação de conteúdo sensível).

I: Em termos de titularidade de contas, considera que o modelo reflete o negócio de forma precisa?
P7: Reflete.

I: Em termos da relação entre os clientes, considera que o modelo reflete o negócio de forma precisa?
P7: Em relação entre clientes, vocês tinham um satélite que tinha (…) tinha desculpam um link que ia e vinha –
I: Posso mostrar –
P7: Mas era isso, não era? Aqui em baixo. Isso, exato. Sim, claro.

I: De forma geral, tendo em conta os conceitos de negócio apresentados, considera que o modelo reflete os conceitos de negócio de forma precisa, cumprindo simultaneamente com as práticas de Data Vault 2.0?
P7: Sim, parece-me que sim.

I: Considerando apenas os conceitos de negócio modelados, como classificaria o modelo em termos de completude?
P7: Eu acho que ele tá muito completo, eu acho que ele tá muito completo, acho que abordou todos os temas relevantes e aliás ainda mais, os "tricky", aqueles que são menos óbvios acho que modelou bem, acho que tao bem transpostos.

I: Na sua opinião, o modelo é simples de entender e utilizar?
P7: Na minha opinião é. Lamento dizer-vos, mas na minha opinião é. Porque (…) o exemplo que costumo dar é o exemplo da numeração, não é. Os números existem tantos quantos aqueles que nós quisermos, e entre quaisquer dois números infinitamente próximos, existe uma infinidade de outros números. E o facto de existir muitos não o torna mais complexo, desde que tenhamos uma regra e uma lógica sempre subjacente na sua utilização. Portanto, para mim utilizar este modelo até para exploração de dados não acho que seja uma coisa extremamente complexa, antes pelo contrário. Ele basicamente o que tem de saber é onde estão os "hubs", ou quais são os hubs. E todo o meu caminho é feito a partir dos hubs. Vou dos hubs, tenho os "links", percebo como é que estas entidades se relacionam. E o meu modelo é fundamentalmente hubs e links. Os satélites são só o adicional, ou aquilo que necessito de ir buscar para responder a uma questão muito concreta. Mas o modelo é direto, se pensarmos noutro modelo relacional, ou num modelo de Data Warehouse mais

convencional, ou até, se quisermos pensar mesmo nos factos e dimensões aproxima-se um bocadinho do tema factos e dimensões, em que tenho as dimensões da análise e tenho os factos. Também é simples por isso, mas essa é simples também porque tem poucas tabelas. E depois tem outras limitações. Mas sim, por isso.

I: Como classificaria a robustez do modelo, na sua capacidade de se adaptar tanto aos requisitos adaptados, tanto aos que possam aparecer?
P7: O modelo, esse que vocês apresentaram, acho que ficou, acho que ficou evidente que ele é adaptável a novas situações de negócio. Também houve um tema que vocês não trouxeram, mas depois se calhar na vossa discussão podem levar. Eu não sei porque é que vocês me estão a fazer estas perguntas, nem sei como é que elas entram aqui, nem sei depois como é que vocês vão utilizá-las. Se vão meter isto no relatório ou se vão utilizar para se preparar também, para a discussão. Mas (…) há uma situação que pode ser interessante que é, todas as abordagens que vocês apresentaram não implicaram fazer alterações à estrutura do Data Vault, no entanto, o Data Vault também é muito (…) muito, como é que vou dizer, permissivo se quiserem, a alterações da própria estrutura, redesenhos de estrutura (…) sem perder todo o conteúdo que já tinham nas estruturas anteriores.

I: Na sua opinião, o modelo representa uma boa prova de conceito para uma futura implementação?
P7: Sim, eu acho que sim.

I: De que forma considera a existência do modelo proposto pertinente e ou importante, no contexto da organização?
P7: Bom, no nosso contexto acho que ele era super importante que nós tivéssemos adotado esta ideologia, por um motivo muito simples, que é nós estamos num processo de migração e queremos começar a entregar (…) e atualmente e estamos a viver esse problema hoje, ok. Optou-se fazer Data Warehouses Kimball. E o problema do Kimball é que ou se já tem completamente fechado o tema das dimensões todas muito bem definidas e os factos que estão, ou então vai estar constantemente a alterar o modelo, que significa alterações em tudo o que está à frente do modelo. E portanto, não prevê essa ideia evolutiva. Eu acho que uma das grandes vantagens do Data Vault é exatamente essa, eu posso começar por trabalhar clientes, e até expor modelos dimensionais ou de utilizadores finais, da lógica Kimball, sem stress. Se eu amanhã tiver uma evolução ao negócio, posso fazer o, descartar aquele modelo e recriar um modelo novo dimensional, com base nas minhas novas estruturas de Data Vault. Nós estamos a viver isso neste momento, e estamos a verificar precisamente que estamos com dificuldades a avançar, porque não existe a possibilidade de reaproveitar o quer que seja, enquanto o modelo não estiver fechado. E um exemplo simples, estamos a falar de (…) da recuperação, que ele projeta recuperação, em que temos clientes que estão para a recuperação de créditos, e é preciso ter a informação do cliente. E como é preciso, mas é preciso ser uma pequena parte da informação do cliente, que não tem a componente máxima, não tem nada disso. E, portanto, a dimensão cliente tinha que estar toda completa, segundo o nosso arquiteto de dados, para que possa implementar aquilo sem ter alterações no futuro, ou pelo menos alterações substanciais. E neste caso vai ter, portanto a vossa solução aqui quando disseram: "Ok, vou criar umas tabelas de "reference data" para aquilo que eu não vou modelar". Era isso que era expectável para um caso como este do projeto de (…) recuperação, que era, ok, vou criar um reference data temporário de clientes, com informação mínima necessária para trabalhar o tema da recuperação, e depois aprofundo o tema da recuperação em termos de modelo. Amanhã posso substituir esta reference data por um modelo adicional –

I: Certo –
P7: Portanto, eu acho que era, era para mim o modelo ideal, o modelo correto, como vocês imaginam.

I: Considera que o modelo proposto pode ser útil para arquitetos, engenheiros, e analistas de dados da organização?
P7: Sim, pode.

I: Que recomendações ou sugestões daria, para melhorar o modelo?
P7: Não, eu para melhorar o modelo não, acho que não, honestamente não tou a ver. Mesmo analisando o modelo de repente, e conhecendo até razoavelmente o negócio (…) Não sei se faria alguma sugestão de alteração. Acho que os temas mais pertinentes foram endereçados corretamente, por exemplo, um dos temas que foi sempre muito discutido, que é o tema do cliente, e o cliente particular, e o cliente empresa. Aqui é direto, com uma abordagem como vocês fizeram, não é, que é ok temos satélites de empresa, temos satélite de particulares. E, portanto, acho que a abordagem é essa. Outra coisa também que o Data Vault nos traz é exatamente esse conforto, que é, eu não preciso de já ter o modelo perfeito antes de começar a implementá-lo. E, portanto, honestamente eu também não pensei muito sobre isso, vou pensando á medida que as coisas vão nascendo. Se modelarmos "by the book", utilizando as regras de modelação que estão definidas, e quando não existe uma regra que seja aplicável, fazer aquilo que vocês fizeram, que foi pesquisar para ver que outras soluções existem, como o "multi-active satellite", e o "same-as", e os "PITs", que é os "point-in-times", resolvem esses problemas, portanto, é algo que, eu acho que isso é mais uma grande vantagem no fundo de um modelo deste género.

I: Que outros comentários pode fornecer sobre o modelo proposto?
P7: Já forneci, agora é vocês apanharem daí. Já forneci vários comentários.