An Overview of Data Vault Methodology and Its Benefits

Andreea VINEȘ, Radu-Eleonor SAMOILĂ
Bucharest University of Economic Studies
andreea.vines@csie.ase.ro, radusamoila2@gmail.com

Business Intelligence plays a vital role in helping organizations to extract meaningful insights from their data and make informed decisions. However, choosing the right data modeling approach can be challenging, as different methodologies have unique advantages and limitations. The focus of the current paper is to provide an insight into the Data Vault architecture, which is an emerging approach to data modeling, compared to the established methods of Kimball and Inmon. The paper conducted a comparative analysis of these methodologies, with a particular emphasis on the benefits of Data Vault. This study highlights the advantages of the Data Vault model in managing data from multiple sources and its compatibility with agile implementation practices. Overall, this paper sheds light on the relevance of Data Vault in contemporary data management practices.
Keywords: *Data Vault, Data Warehouse, Architecture, Data modelling, ETL*
DOI: 10.24818/issn14531305/27.2.2023.02

1 Introduction

In the era of digitalization, most functions of business and society are depending on the data that comes in many forms. To gain a large competitive advantage and some insights from the data, there is required to apply data analytics on top of it. [1] Hence that, the volume, velocity and variety of data are increasing rapidly over the last decades. An advanced method to overcome all the shortcomings of traditional data warehouse solutions that might appear became required.

For example, considering the volume that keeps increasing, traditional data modelling approaches can become difficult to maintain to store all the information. Another example would be the data sources that are constantly changing; hence the classical data warehouse model is quite robust, and it can't handle all business requirements. Moreover, there are multiple issues that affect the classical systems, such as missing source keys, data issues, data quality, performance gaps and lack of database tuning or partitioning. [2] [3]

As traditional data warehouses were designed to handle structured data, not semi-structured or unstructured data that comes from social media, mobile devices or other sources, vendors started building a new generation data warehouse with capabilities of performing analytics and forecasting, to support structured and unstructured data. Thus, enterprises will be capable of making better decisions for their organizations. [4]

To address the challenges posed by big data, new data warehousing techniques have emerged that are better suited to handling large and variated data sources. One such technique is the Data Vault methodology, which was developed by Dan Linstedt in the early 2000s. Its main aim is to addresses all the challenges of building and maintaining a data warehouse in the context of big data, providing a flexible, scalable, and auditable approach to data modelling and integration.

Data Vault is designed to accommodate a wide variety of data sources, including both structured and semi-structured data, making it well-suited to the challenges of big data. In this context, it is important to explore the principles and practices of Data Vault, and to understand how it can be used to create effective data warehousing solutions in the era of big data.

The purpose of this paper is to conduct a comprehensive analysis of the evolution of data warehouse models, with a specific focus on the Data Vault methodology proposed by Dan Linstedt, and its relevance in the context of Big Data. The paper includes an overview of this area and also a use case that

demonstrates how semi-structured data can be integrated using Data Vault architecture.

To achive this objective, the current paper is structured as follows: The first section provides an introduction of the paper. The second section describes the three available methodologies of Data warehouses developed by Kimball, Inmon and Linstedt. The third section specifically focuses on the Data Vault, presenting its architecture and core components. The fourth section presents a use case that illustrates how data can be modelled using Data Vault 2.0. Finally, the last section of the paper presents the conclusions of the study.

2 Data Warehouse models – from Kimball to Linstedt

Data Vault represents a conceptual and logical data model used to build data warehouses for enterprise-scale analytics. It comes as an alternative of the architectures proposed by Bill Inmon and Ralph Kimball. The concept of Data Vault appeared in 1990 when Dan Linstedt published some papers regarding this method, being able to propose a final version of the architecture in 2020. In the following 10 years, the author adjusted the method, and adapted it to the current necessities and design patterns, providing an adaptive method called Data Vault 2.0.

Kimball's architecture [5] that was proposed in 1996 consist in a two-layer model. Raw data is copied into a staging area and all the transformation happen before it is ingested into the second layer, data warehouse. Using this approach, end users will connect to the same data warehouse model and all the dimensions will be common. The main advantage of it is that it can be easy to implement, but it can also be quite challenging if there is required to build a second model using the same landing data, as everything needs to be loaded again. [5]

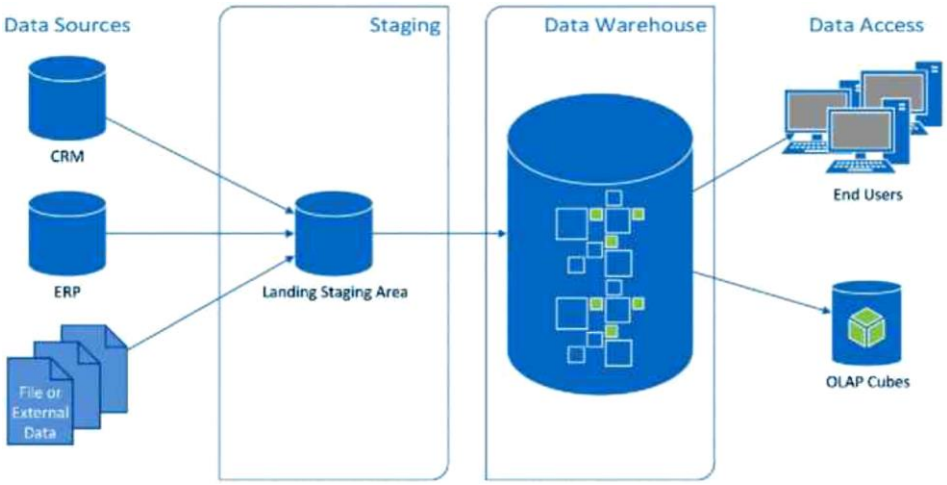

Fig. 1. Kimball approach architecture
Source - [7]

Inmon's architecture [6] has three layers to overcome the challenges presented earlier, where the new layer consists of data marts which are some mini-data warehouses from where users can extract the information that is required. This also requires more data processing to create all the data marts. The advantage of separating the data warehouse from the data mart is beneficial in the context of models with multiple use cases. It also offers scalability in the context of integrating new requirements or defining new data marts. [6]

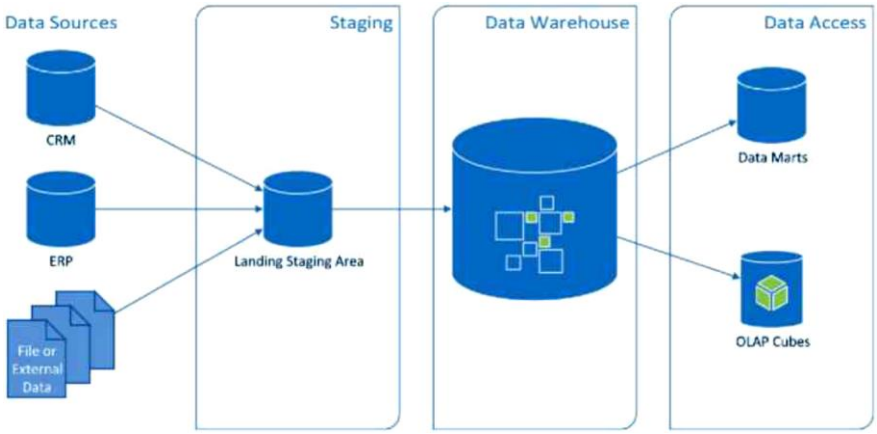

Fig. 2. Inmon approach architecture
Source - [7]

Data Vault 2.0 was designed in 2010 by Linstedt as a necessity to come with a new proposed model scalable in the context of big data, to be able to handle unstructured data and to be integrated with more systems. One of the main differences is the change in using hash keys, which are primary keys instead of sequential numbers. This change allows all the hubs and satellites (which will be presented in detail in Section 3 – Data Vault Architecture). to be processed in parallel and removes all the dependencies on the data processing side. The hash key is calculated based on the combination of the columns that constitute the business key of the table, using different methods such as SHA-1 (Secure Hash Algorithm-1) or MD5 (Message-Digest Algorithm). Several of the differences between Data Vault 1.0 and Data Vault 2.0 are synthesized in the table 1. [7]

Table 1. Data Vault 1.0 & 2.0 Comparison

Data Vault 1.0	**Data Vault 2.0**
Sequence numbers for Surrogate keys.	Use Hash keys for surrogate keys.
DV 1.0 also used MPP but only to a certain extent.	DV 2.0 takes advantage of MPP style platforms and is designed with MPP in mind.
Limited support to real time load.	It's real-time ready, cloud ready, NoSQL ready and big data friendly.
Not very flexible to automation and virtualization.	DV 2.0 has a very strong focus on both automation and virtualization.
DV 1.0 had a major focus on modelling and many of the modelling concepts are similar.	DV 2.0 is a complete system of Business Intelligence. It talks about everything, from concept to delivery.

The model proposed by Dan Linstedt represents a third approach of data modelling and it comes with a series of benefits, such as: [8]

- It contributes to the organization ability and improves the speed at which the business can learn and exploit more opportunities.
- It delivers a modernized data service.
- It enables new business capabilities such as data-driven decision-making, data science and it can be considered the key to new business models.

It represents an agile, structured solution with flexibility for refactoring.

According to the paper of Cernjeka et al. that developed a metamodel, Data Vault can be used to integrate semi-structured data as

well.[9] The authors translated some NoSQL document stores that use JSON files in a data Vault model using the key-value concept of the files. The following translation rules were applied:

- TR0: MongoDB collection is translated into a set of hubs, links and satellites.
- TR1: each document id is translated into a business key and a hash key is calculated and added to the hub.
- TR2: the non-id fields of the document are translated as attributes of the satellite entity where the value of the satellite attribute is the value part of the document.
- TR3: a document reference to another document is translated into a Data Vault link entity connecting the current hub and a different hub (referenced document).
- TR4: an embedded document is translated into a DV entity connected to the current hub. In this case TR1-TR3 are applied accordingly.

However, this metamodel doesn't provide details regarding nested attributes as part of JSON files, or how the JSON files with different structure or attributes can be integrated in the Data vault model, which will be the subject of future work.

On the other side, authors of paper [10] provided a methodology for mapping a Data Vault schema to a standard XML schema, that offers flexibility and also helps if the data from the source systems come in various formats, and they all need to be stored in a centralized data warehouse. The authors also mentioned that using XML schema might have some limitations, such as identifying the hub key or the data type variations.

As Data Vault is a quite new approach, most of the studies found in the literature focus on comparing the methodologies of Inmon and Kimball. However, there were also some papers that consider Data Vault in the comparison [7][8]. The conclusion of these works is that all three methods have advantages and disadvantages, depending on the aim of the implementation. In the two mentioned researches, it was highlighted that Data Vault implementation requires simple and standard ETL rules to load data, compared to Kimball's approach where the transformations to create the dimensions and fact tables can be complex. Another aspect worth mentioning is the **historization**, where all three approaches can handle data versioning. Regarding the lifecycle of the model, it was already mentioned that Data Vault provides an **easy to change model**, where the existing tables are not affected by any changes. To conclude, for all modelling approaches presented, developers and data architects are required to understand the requirements and choose the options that best fit.

3 Data Vault architecture

The main objective of Data Vault is to ensure quick adaption to changes in case a new business object is defined or some new sources are ingested. The architecture is defined on three layers: [9]

- **Staging area** – which ensures ingesting data from different sources and keep them in their raw format; this layer doesn't keep the history; hence it gets truncated every time a new batch is being processed. It allows only applying different ***hard business rules*** which doesn't impact the meaning of the data (for example, applying different data time transformations, define hash keys, defining Unicode format etc.)
- **Enterprise data warehouse area** – which represents the second layer of data which is modelled using the data vault 2.0 architecture, also keeping the history of the data. This component is designed using hub, satellites, and link tables. This layer allows applying ***soft business rules***, calculating different KPIs. This area can be also dividing into a **Raw Vault** where raw data is transposed into a data vault, or directly into a **business vault**, where all the business rules are applied, and data is modelled using Data vault.
- **Information delivery area** - which allows end users to access the data using data marts and use it to get different insights. Data is aggregated and prepared in order to be used for different use-cases,

such as decision-making, prediction, artificial intelligence or machine learning.

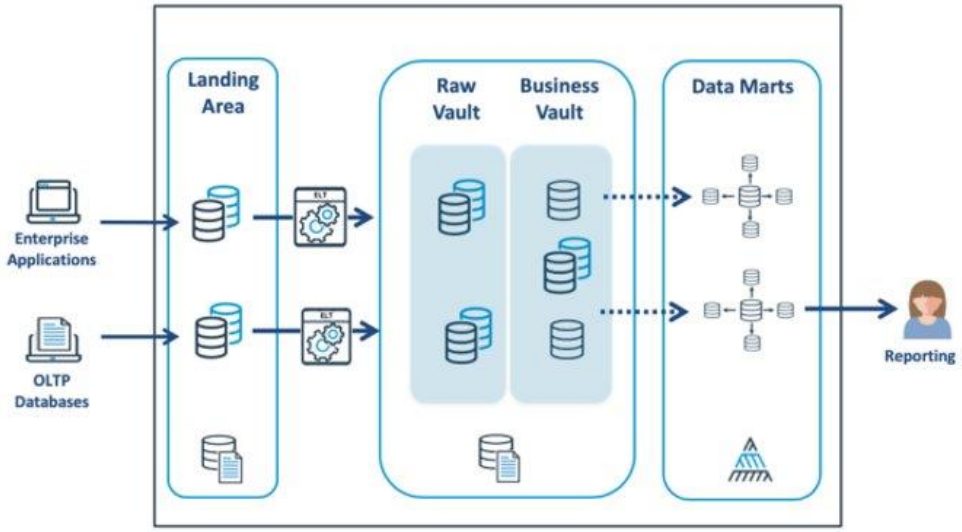

Fig. 3. Data Vault architecture
Source - [7]

Comparing with the dimensional approach, where the main components of a data warehouse are facts and tables, in Data Vault there are defined three core components: hub, link and satellite. [9]

A hub represents the core business concept (customer, product, sales, vendor); it is created as table that contains the business key and some metadata about when the key was first loaded and the source of it. With the Data Vault 2.0 model, the primary key of the hub table was replaced with a hash key of the business column (in the Data Vault 1.0 version, the primary key was created using a sequence number).

A link represents relationships between multiple business objects. It contains a primary key defined also as a hash key and foreign keys for the hubs that are being interconnected, like a bridge table. Links make the model more flexible as for every new functionality added, it is just required to create a new hub table and connect it to a link table. A satellite stores information about the hubs and the relationship between them, contains all attributes and information about when the data was loaded and its source, keeps also all the history to track the changes. The structure of it consists in a primary key created using the parent hash key and the foreign key of the load data and different other columns for source, load date and attributes. Each hub can have multiple satellites with correspond to different data sources that are integrated into the system and they have the same business key. Another alternative is to define a new satellite if some new attributes are being ingested in the system, instead of updating the current tables.

The popularity of Data Vault was defined based on the following characteristics [13]:

i. Flexibility which allows new sources to be easily integrated in the correct model, with no or only some small updates to the existing tables. Once a new entity is being defined, there is required only to create a hub table and update one of the link tables to connect with the new hub that was created.
ii. Loading efficiency is another characteristic, as all the entities can be loaded in parallel without requiring a lot of data dependencies. The only dependency required is that all the entities need to be processed in the following order: hubs, links and satellites, but all components can be loaded in parallel (for example, all hub tables can be loaded in parallel, followed by linked tables and the satellites).
iii. Third characteristic is auditability that keeps track of all the changes made to a source system as all the changes are

tracked using the satellites, meaning that each change of the current data is being stored as new records with the timestamp that indicates its expiration date, similar with slowly changing dimensions type 2.

4 A simplified implementation of Data Vault

This section presents an alternative of implementing a Data Vault model and its benefits of integrating new sources to the current model.

The initial data model contains two objects – Products that initially gets data from two sources (S1 and S2) and another object for Categories that only ingests data from one source (S1). All data comes in a structured format from the client's systems, and it needs to be ingested in the final data warehouse in order to be used in the presentation layer, where users can consume the data through the reports and dashboards. The table structure of the input data that has the column names, description, and data types is presented in the Tables 2 & 3:

Table 2. Products table structure (S1 & S2 sources)

Column name	Description	Data type
id	Unique identifier	String
Name	Name of the product	String
Description	Description of the product	String
Price	Current price	Float
Category_id	Category of the product	String

Table 3. Categories table structure (S1 source)

Column name	Description	Data type
Category_id	Unique identifier	String
Name	Name of the category	String
Description	Description of the category	string

Considering the two data source presented above, the proposed Data Vault model is represented in the figure below. It contains two hubs (one for Categories and one for Products) and the two hubs are being connected using a link table.

The primary key of the hub is created as a hash key of the business key of the Products source (which is the id) and the record source. In this case, if the data that comes from different data sources will have the same id, they will be distinguish in the hub as the record source is difference and hence, that hash key generated will be different.

The link table created will contain the hash key of each hub as foreign key and a new hash key (created as a combination of the two mentioned) as a primary key.

The Product hub is connected with two satellites, representing one satellite for each source. The Categories hub has only one satellite with its attributes to display the category name and description.

One of the main principles of the Data Vault is that data won't be deleted. [14] Considering that, the information regarding when the data was ingested are kept in every source using 'Load Date' columns. As the satellites keep the history of the data, the column 'Load End Date' will be used to track when the record become inactive – for example, a new record for that key was added in the table (it was modified in the source); hence that, the old record will be marked as inactive by populated 'Load End Date' column.

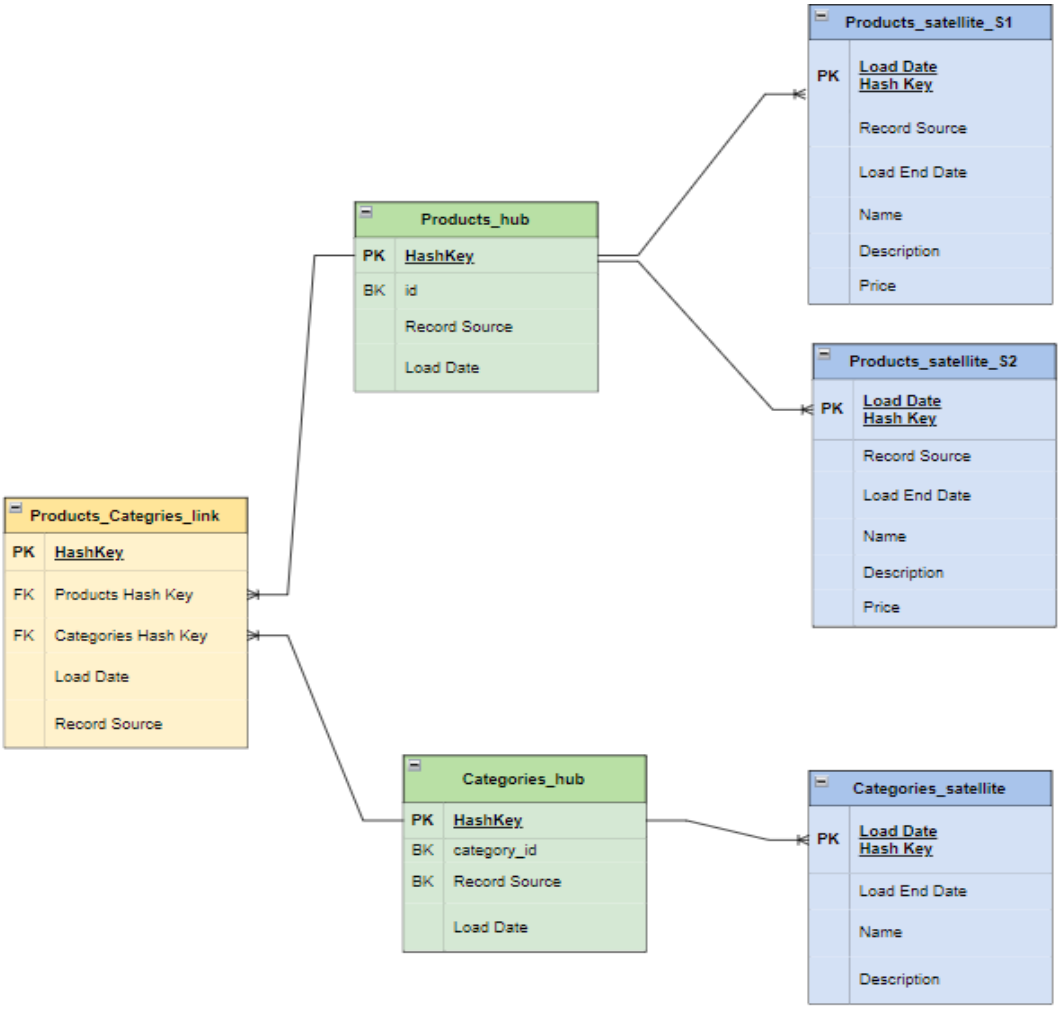

Fig. 3. Initial Data Vault model

If a new source of products is ingested, the approach is to create a new satellite for it. Considering that the data source will have a new attribute, size, it won't affect the other sources that don't have this attribute and the current ETL process of ingesting data and uploading it into the satellites won't be affected. To distinguish between data source, the hash key of the hub is creating using the business key (which is the id) and the record source. In this case, the hash key will be unique, and it won't be affected by having the same id value in multiple sources. The new data model is displayed below.

Table 4. Products table structure (S3 source)

Column name	**Description**	**Data type**
id	Unique identifier	String
Name	Name of the product	String
Description	Description of the product	string
Price	Current price	Float
Size	Current size of the product	String
Category_id	Category of the product	String

From a data integration standpoint, when comparing all three methodologies, it becomes apparent that Data Vault simplifies the process of adding new sources by keeping

the satellites separate. In contrast, with the traditional Inmon and Kimball approaches, if a new data source is added to the system, the ETL process must be adjusted to incorporate it.

The new data model can be observed in the Fig. 5. below, where the only change that happen compared to the previous version, was to create a new satellite table (Products_satellite_S3).

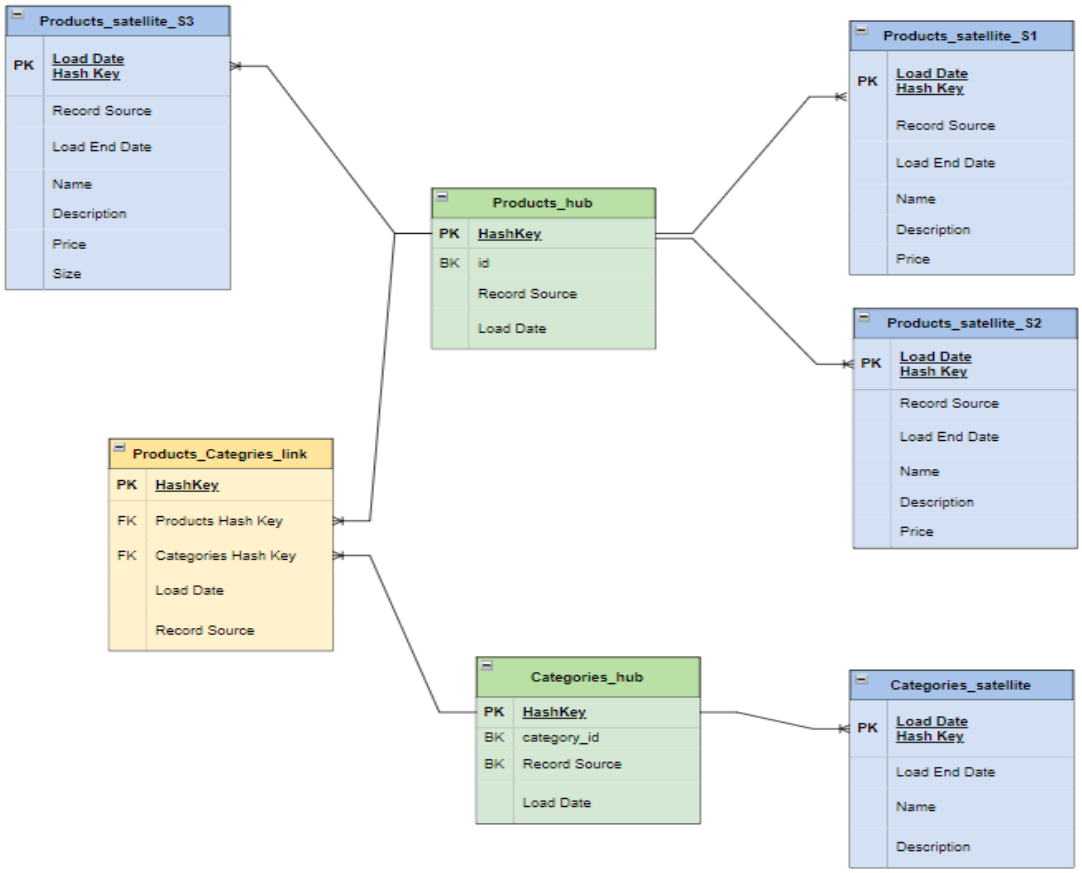

Fig. 5. Final Data Vault model

5 Conclusions

In this paper it was presented a comprehensive overview of the integration of Data Vault architecture within data warehouse solutions. The study starts by comparing this approach with the methodologies introduced by Kimball and Inman. The traditional approaches, Kimball and Inman, are prone to challenges generated by changes in the current data sources, and they require considerable data engineering work to adjust the current ETL flow.

Data Vault architecture, on the other hand, provides flexibility and scalability to overcome these issues. Although this modelling architecture is relatively new, its agile methodology is increasingly being used by many more companies to develop data solutions.

It should be also highlighted that all three approaches have their respective advantages and disadvantages, depending on the defined purpose. The traditional approaches are an excellent fit if the system is stable, and the requirements are well-known. In contrast, Data Vault architecture is a powerful approach if the data warehouse that needs to be built has to be flexible and scalable due to its architecture that allows easy integration of new sources.

Therefore, the selection of an appropriate approach should be based on the specific needs and objectives of the data warehouse solution that needs to be implemented. While Kimball and Inmon are conventional methodologies that have been in use for a long

time and have proved their efficiency in certain contexts, Data Vault provides a modern and flexible solution that is better suited for rapidly evolving environments.

Moreover, Data Vault provides support for semi-structured data which seems to be more and more used to develop modern data warehouse solutions. Data Vault architecture enables the construction of incremental models at a low cost, making also easy to modify the business rules. One of its main benefits is that it helps in the context of Big data, when it comes to data variety, being possible to integrate unstructured data or to perform near-real-time data loading.[15] Another advantage for Big data processing refers to the volume of data that can be ingested in the system, allowing petabyte-scale management.[9]

Some papers have explored the overview development of metamodels for semi-structured data, but there is a need for a more in-depth analysis on this area.

As a future work, the paper can be extended to include the implementation of semi-structured data, considering nested JSON or XML files and how they can be modelled using Data Vault. Another aspect that could be considered when developing solutions for semi-structured data is how to handle cases where the attributes differ from one file to another, and how they can be modelled effectively within the Data Vault architecture. As semi-structured data becomes more prevalent in the modern data landscape, exploring the integration of this data type within the Data Vault architecture is being critical to ensure that data warehousing solutions remain flexible and scalable. By addressing these issues, the Data Vault architecture can continue to provide a comprehensive and robust solution for data warehousing in the future.

References

[1] J. C. Margulies, "Data as competitive advantage," Winterberry Group, Oct. 2015, Tech. Rep. 2015-02, pp. 1-28

[2] D. Linstedt, K. Graziano, and H. Hultgreen, "The business of data vault modeling, 2nd edition," in Proceedings of the 2009 International Conference on Information and Knowledge Engineering (IKE), Las Vegas, NV, USA, July 2009, pp. 232-238.

[3] D. Linstedt and K. Graziano, "Super Charge your Data Warehouse," Createspace Independent Pub, Scotts Valley, CA, USA, 1st ed., 2011

[4] Deloitte CIO Journal Editor. "The Future of Data Warehouses in the Age of Big Data" [Online]. Available: http://deloitte.wsj.com/cio/2013/07/17/the-future-of-data-warehouses-in-the-age-of-big-data/. [Accessed: Apr. 14, 2023]

[5] R. Kimball and M. Ross, "The data warehouse toolkit: The definitive guide to dimensional modeling," John Wiley & Sons, 3rd ed., Indianapolis, IN, USA, 2013

[6] W. H. Inmon and B. D. Eggleston, "Building the data warehouse," John Wiley & Sons, 4th ed., Indianapolis, IN, USA, 2005

[7] L. Yessad and A. Labiod, "Comparative Study of Data Warehouses Modeling Approaches: Inmon, Kimball and Data Vault," in 2016 International Conference on System Reliability and Science (ICSRS), Algiers, Algeria, 2016, pp. 77-82

[8] D. Schneider, A. Martino and M. Eschermann, "Comparison of Data Modeling Methods for a Core Data Warehouse," in Trivadis, pp. 20-21, 2014

[9] D. Linstedt, "Building a Scalable Data Warehouse with Data Vault 2.0," Morgan Kaufmann, 1st ed., Boston, MA, USA, 2016

[10] "What is Data Vault? - Data Vault," Data Vault, 2019. [Online]. Available: https://www.data-vault.co.uk/what-is-data-vault/. [Accessed: Apr. 14, 2023]

[11] K. Cernjeka, D. Jaksic, and V. Jovanovic, "NoSQL Document Store Translation to Data Vault Based EDW,"

in Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, May 2018, pp. 1105-1110

[12] C. Knowles and V. Jovanovic, "Extensible Markup Language (XML) Schemas for Data Vault Models," Journal of Computer Information Systems, vol. 53, no. 4, pp. 12-21, 2013

[13] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, "Modeling Data Lakes with Data Vault: Practical Experiences, Assessment, and Lessons Learned," in Proceedings of the 38th Conference on Conceptual Modeling (ER 2019), 2019, pp. 90-99

[14] D. Linstedt, "Super Charge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault," 1st ed. CreateSpace Independent Publishing Platform, 2011

[15] I. Nogueira, M. Romdhane and J. Darmont, "Modeling Data Lake Metadata with a Data Vault" in Proceedings of the 22nd International Database Engineering & Applications Symposium (IDEAS '18), New York, USA, 2017, pp. 253-261

Andreea VINEȘ has graduated the Faculty of Economic Statistics, Cybernetics and Informatics in 2019. Following the bachelor's degree, she also pursued a Master program at the Bucharest University of Economic Studies in Information Systems for the Management of Processes and Economic Resources. She is currently working as a Data Engineer, and she comes with a wealth of experience in designing, developing and maintain data solutions, her expertise including building and managing data pipelines, data warehousing and ETL processes, focusing more on the cloud technologies. Her main fields of interest are cloud technologies, big data, data warehousing and BI.

Radu SAMOILA has graduated the Faculty of Accounting, Audit and Information Technology in 2009. He also holds a master's degree in Economy, Audit, and Information Technology since 2011 at the Bucharest University of Economy. He had different managerial roles for multinational companies operating in the energy industry, business, and financial consultancy or FMCG business sectors. His main roles were covering internal and external audit, business advisor and corporate governance activities. He is a PhD Student at Bucharest University of Economy, since 2019. Main fields of interests are business process optimization and automation, as well as the continuous improvements concept.