Awel Eshetu Fentaw

Data Vault Modelling

An Introductory Guide

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

31 March 2014

Helsinki Metropolia University of Applied Sciences

Author(s) Title	Awel Eshetu Fentaw Data Vault Modelling: An Introductory Guide
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Olli Hämäläinen, Senior Lecturer

The theme of this thesis is to prepare an introductory guide to Data Vault Modelling. The Data Vault is a relatively new method of modelling enterprise data warehouses. The Data Vault uses three core components to model an entire data warehouse. Thus it provides an easy alternate solution for data warehousing.

The thesis project was conducted by researching books, scientific journal articles, professional blog posts and professional community discussions. By doing so scattered information about Data Vault Modelling was gathered and analysed and compiled in to a guidebook. This paper could be used as a quick guidebook for those interested in finding information about Data Vault Modelling.

Data Vault is a relatively new solution to enterprise data warehousing. Although it introduces a better way of modelling enterprise data warehouses, it still has limitations when it comes to providing strict guidelines and handling unstructured text, semi-structured Doc Style data, XML and other data types which are broadly known as Big Data. These data types cannot be handled using the current approaches, tools and techniques; however there is an ongoing research to make the Data Vault Model be able to handle such Big Data. Thus this thesis includes only officially published articles about Data Vault.

Keywords	Data Vault, Hub, Link, Satellite, Data Warehouse

Contents

Abbreviations

3NF	Third Normal Form
BDW	Business Data Warehouse
CIF	Corporate Information Factory
EDW	Enterprise Data Warehouse
ETT	Extraction Transformation-Transportation
ETL	Extract Transform and Load
EWBK	Enterprise Wide Business Key
MDDB	Multi-Dimensional Database
OLAP	On-line Analytical Processing
RDBMS	Relational Database Management System
ROLAP	Relational On-line Analytical Processing
SaaS	Software-as a Service
SID	Sequence Identifier
SQL	Structured Query Language

1 Introduction

Information plays a fundamental role in decision making in everyone's life. In the enterprise world every piece of information from different sources of operation are described as data. Data are the backbone of an enterprise and plays a major role in decision making in different aspects regarding the enterprise. Choosing the proper way of the data warehouse modelling method helps modelling a database layer where data are stored and become available for retrieval for the purpose of decision making, data mining and other activities.

There are several data warehouse modelling methods. The most common competing options for modelling are either modelling with confirmed dimensions and an enterprise data bus, or modelling with the database normalized. Although these methods have their own advantages, they have issues when it comes to changes in the systems feeding the data warehouse. Data cleansing for confirmed dimensions will lead to losing information. A new way of data warehouse modelling, Data Vault came into case in year 2000 and it was designed to minimize the impact of the issues mentioned above.

The purpose of this paper is to prepare an introductory guide to Data Vault modelling basics. Since Data Vault modelling is a new concept of data warehouse modelling, there are few published reference books. Most information regarding Data Vault modelling is scattered in the form of journals, newsletters, blogs and community discussions. The goal of this thesis is to provide information about what Data Vault modelling is, what core components it contains, how it can be modeled and used as an enterprise data warehouse model.

2 Theoretical Background

2.1 Data Warehouse in General

The idea of collecting and integrating all operational information of an organization in a centralized place for the purpose of conducting analysis and making decisions has been the aim of many information managers. However, there was no formalized architecture or thought for developing an information management tool until W.H. Inmon came up with the term data warehousing in 1990. [12, 23]

Since the introduction of the Third Normal Form (3NF) in the early 1960's for on-line transaction processing systems, the need for data warehouses was increasingly growing. Data warehousing, when used properly will enhance information managers of an organization with timely information necessary to effectively make crucial business decisions and solutions. [12, 23].Figure 1 shows the data warehousing timeline.

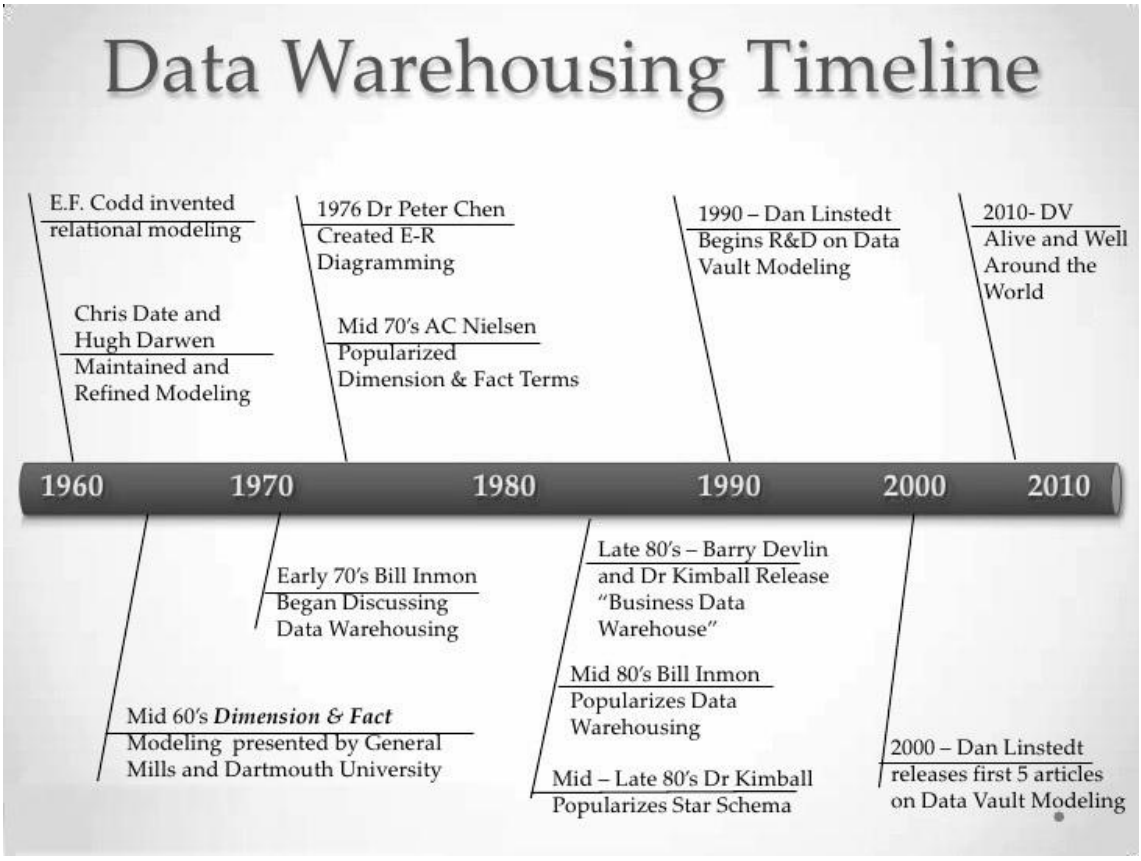

Figure 1.The Data Warehousing Timeline. Adapted from Kent Graziano (2011) [10, 8]

The term data warehouse is a generalized phrase that has taken different meanings according to different experts in the field. According to Michael Brackett, a data warehouse is “a repository of consistent historical data that can be easily accessed and manipulated for decision support.”[13, 268-269] W.H. Inmon who came up with the term data warehouse initially defines it as “a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management’s decision-making process.”[14, 2]

A data warehouse is a relational database designed for query and analysis. It usually includes historical data derived from transaction data or other means of sources. In addition to relational databases, a data warehouse environment may include an extraction, transportation, transformation, online analytical processing (OLAP) and data mining capabilities, different client analysis tools and other applications that manage the process of gathering and delivering data.[15,35-36].
There are different data warehouse characteristics set by William Inmon such as:

- Subject-Oriented
- Integrated
- Non-Volatile
- Time Variant. [15, 36]

Data warehouses are designed to help data analysis. The concept “subject-oriented” depends on the change of data from application-oriented to decision support. Application oriented data are detailed data based on functional requirements, whereas data for data warehouses only include data for operating decision analysis. [14, 3-4; 15, 36]

Integration in data warehousing focuses on maintaining consistency throughout different databases from which data are extracted. Consistency in data warehouses can be achieved by standardizing naming conventions, encoding structures, units of measurements and physical characteristics of data. Integration assures data are stored in a single globally accepted manner so as to resolve conflicts in naming and inconsistencies among measurement units. [14, 5-7; 15, 36]

One of the key elements for decision analysis to predict future matters about an organization is historical data. Historical data refers to changes over time and these changes can be tracked by introducing a time element in the data warehouse. Time variance in a data warehouse can be shown in several ways. Data warehouses contain data over

different periods of time encompassing years, year-to-day, months, month-to-day, weeks and days. Thus the time variance is clearly shown. The index key structure of a data warehouse maintains an explicit time dimension which shows a specific time element as part of the index key. Data in a data warehouse are a series of snapshots from the operational database that cannot be updated and this is useful to track historical data. The term time variant therefore focuses on changes over time in a data warehouse. [14, 8-9; 15, 36] The time variant is a useful characteristic to defining data warehouse to discover trends in a business.

Another important characteristic defining a data warehouse is non-volatility. The concept of non-volatility comes from the idea that a data warehouse holds a snapshot of operational data which will not be updated in the traditional sense. Activities that could happen in a data warehouse are loading the data into the warehouse and accessing that data for the purpose of decision making analysis. Non-volatility strengthens the purpose of a data warehouse by enhancing decision makers to analyze what has occurred. [14, 10-12; 15, 36]

Traditionally data warehouses fall in to two categories, either Relational Database Management systems (RDBMS) or Multi-Dimensional Database (MDDB). Data warehousing gives an organization an important insight about the information embedded in their operational data sets. It is necessary to reorganize and structure operational data in the data warehouse architecture to maintain quality performance and integrity in the operational and evaluational data sets.

An RDBMS is a database system that organizes and accesses data as two dimensional rows and columns. In this case data are organized in such a way that related information can be accessed using Structured Query Language (SQL). Although it is one way of organizing data warehouse, RDBMS faces difficulty when it comes to supporting complex analytical and decision making processes. The reason is that each time a query is executed it must collect the data the query is seeking and that sometimes involve going through millions of records. Due to this reason, RDBMS was not the best choice for data warehousing projects that involved numerous complex queries. Thus it has led new technologies to emerge to handle the case. [12, 26]

An MDDB is a database technology that represents multi-dimensional data as aggregations of data in cells that are the intersection of multiple dimensions. A dimension is a

table with a single-part key that relates directly to the fact table. The fact table contains the measures of interest and relates all the dimensions in a star-like structure by using a multi-part key. Figure 2 shows an example of a fact table in MDDB.

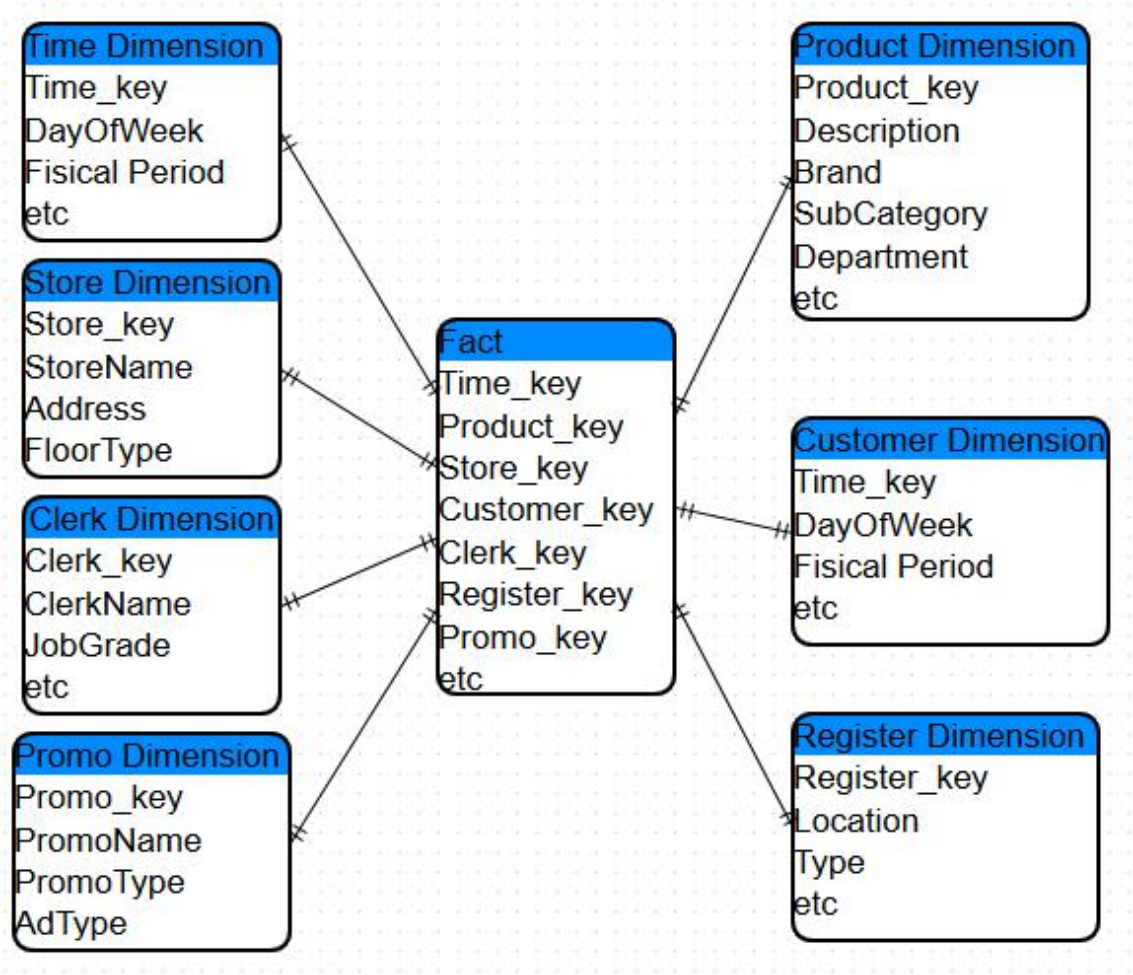

Figure 2. Fact table in MDDB. Adapted from Robert W.Dill (1985) [12, 27]

In MDDB the fact table is used to traverse data across multiple attributes quickly. The dimension tables contain the actual descriptive data. Data in MDDB are stored according to common usage patterns of users in such a way that frequently accessed data are summarized, preprocessed and made available for a user to query when needed. Contrary to RDBMS that has to process the query dynamically each time a request is made for data, MDDB is preferable for quick retrieval of predefined calculations and results. [12, 27]

2.2 Data Warehouse Architecture

In Information technology data architecture is composed of models, policies, standards or rules which govern data collection, storage, arrangement, integration and usage in organizations. Architecture is different from methodology. Often times these two terms are used interchangeably, which leads to confusion. A data warehouse architecture identifies component parts, their characteristics and the relationships among the parts. On the other hand a methodology identifies the activities that have to be performed and their sequencing. Furthermore it should be noted that the architecture is the end product and the methodology is the process for developing an end product. [6, 14]

Regarding modelling and data logistics, data warehouse architectures can generally be categorized as traditional, hybrid and modern. A traditional architecture includes information factory and enterprise buses. On the other hand hub-and-spoke is an example of a hybrid architecture. Modern architectures include Data Vault and Anchor Modelling. There are several other types of data warehouse architectures. Independent data marts, data mart bus architecture with linked dimensional data marts, a centralized data warehouse and federated data warehouse are among other architectures which data warehousing literature suggests. Despite their variety, data warehouse architectures contain common core components such as source systems (external or operational), data staging area, data presentation area and data access tools. Figure 3 explains the data warehouse architecture with common components.

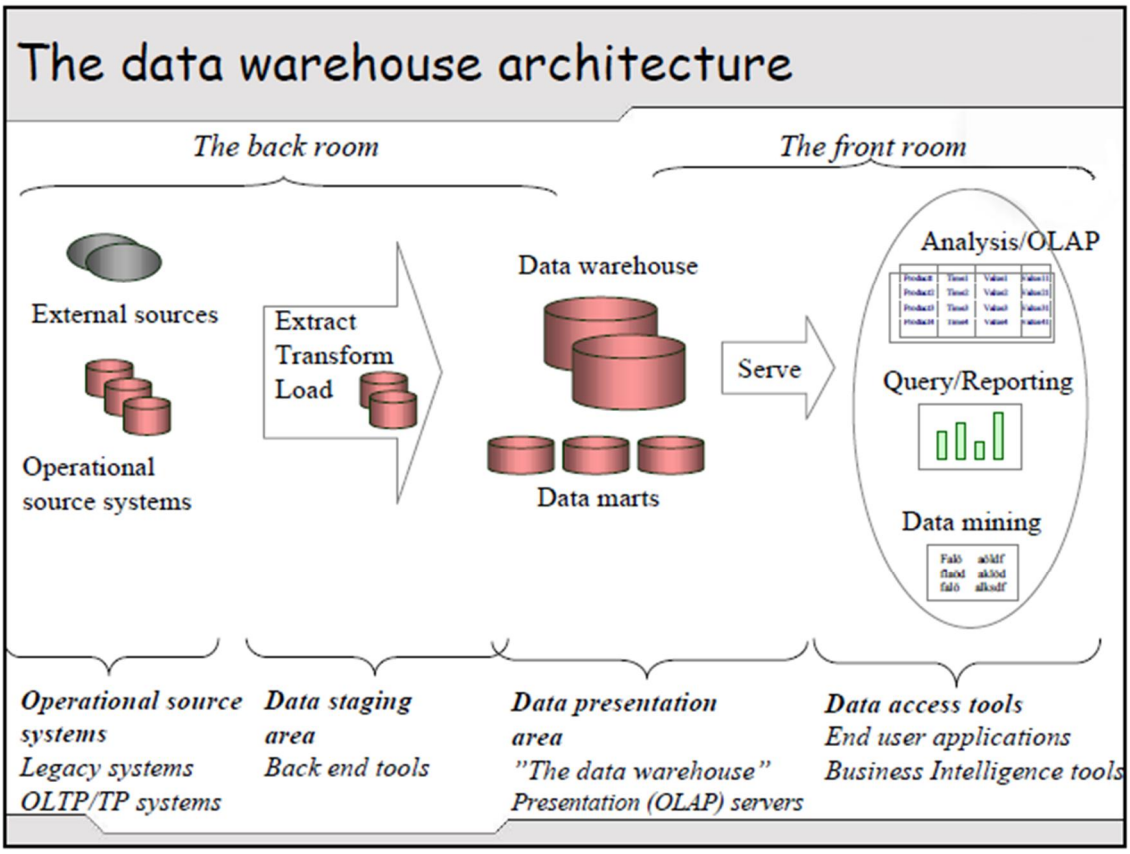

Figure 3. Data warehouse architecture with common components. Adapted from Erike Perjons [7, 1]

Source systems refer to different data sources that feed data into the data warehouse. Data can be available in any format such as relational database, excel files or plain text files. Data sources can generally be classified as operational, which is usually available in the data warehouse operating systems and external, which includes data out of the operating system. Operational data such as sales data, HR data, product data, marketing data, system data, and web server logs with a user browsing data combined with external third-party data can all act as source systems. [8]

The data staging area is where data stands before it is transformed into a data warehouse or data mart. It is often the complex part in the architecture and involves extracting, transforming, loading and indexing activities. Extraction refers to reading and understanding the source data and copying the data needed for the data warehouse for further manipulation and other activities such as transformation [7, 4-5; 8].Transformation involves the following activities:

- Data conversion: specifies transformation rules to convert information in to common data formats and semantics.
- Data cleansing: data scrubbing by using domain specific knowledge to check the data and data auditing by finding suspicious patterns and violation of stated rules.
- Combining data from multiple sources
- Assigning. [7, 9]

The data presentation area usually refers to a series of integrated data marts. A data mart is a flexible set of data based on granularity. In its simplistic form a data mart represents data from a single business process. The data presentation area is where information that reaches the users is available [7, 9; 8].Data accessing tools refer to activities made by the end user to get data from the presentation area. These activities include query/reporting, data mining and analysis.

2.3 Data Marts

A data mart is a decision-support database application which enhances decision-making solutions for a specified subject area such as sales, finance or marketing. Being built and controlled by a single department within an organization, data marts usually draw data from a few sources. These sources could be external data, central data warehouse or internal operational systems. [12, 30]

Based on the data source that feeds the data mart, data marts could be categorized as dependent, independent and hybrid. Dependent data marts get data from a central data warehouse that already exists. As the name implies, independent data marts are standalone systems which draw data directly from operational or external sources. Hybrid data marts can get data from operational systems or a data warehouse.

The key difference between dependent and independent data marts is the way data is populated from the sources into the data mart. This process is called Extraction-Transformation-Transportation (ETT). It involves moving of data from operational systems, filtering and loading it to the data mart. The ETT process in dependent data marts is simplified due to the existence of formatted and cleaned data in the central data warehouse. Therefore the ETT process in dependent data marts is a matter of identifying the right set of data related to the specified subject area in the chosen data mart. Independent data marts however should be dealt with all aspects of the ETT process. The motivation behind the choice of different types of data marts depends on improved performance and availability, better control and costs of time for solutions. [16]

Like data warehouses, data marts can be categorized into two design architectures as relational and multi-dimensional data marts. The choice between these architectures depends on the type of analysis to be performed and the type of data to be analyzed. [12, 32]

Multi-dimensional data marts (MDDM) which maintain a large amount of numeric data (such as sales data) are used to suggest different analytical ways on the same data. Once the data are loaded from data warehouse or external sources, it should be available in a structured framework to facilitate analysis. MDDM are efficient for analyzing

numeric data. The method of analytical processing for decision support is known as On-line Analytical Processing (OLAP). [12, 32]
The relational data marts allow more general-purpose decision-making tools than MDDM. The relational data mart uses a type of analysis process called Relational On-line Analytical Processing (ROLAP).It supports a range of purposes for numeric and textual data. [12, 32]

The main difference between a data warehouse and a data mart is in the size of data maintained. The range of subject areas to deal with and the number of sources the data are drawn from create size differences that a data warehouse and data mart could handle. Although the difference in size is visible in most cases, there exist more differences in the application and implementation. Table 1 shows the main differences between data warehouses and data marts.

Table 1. Differences between data warehouses and data marts. Adapted from Robert W.Dill (1985) [12, 30]

Property	Data Warehouse	Data Mart
addresses	many subject areas, maybe the entire enterprise	single subject area
sized at	gigabyte(GB) to terabyte(TB)	megabyte(MB) to low gigabyte(GB)
accessed by	business analysts and front-line users	business analysts and front-line users
costs	$millions	$tens and hundreds of thousands

2.4 Future Data Warehousing

Every day organizations generate and record massive volumes of data. There exist many business intelligence tools which could extract these data for the purpose of better business analysis and decision making. However business environments and consumer behaviors are altering rapidly and thus the market is so unpredictable that requires the use of all forms of data that are available. [22, 2]

With the huge growth in Internet-based businesses and availability of many social networks, many enterprises now record terabytes and petabytes of data in their data warehouse. Even though there exist enough business intelligence tools for analytical reporting purposes, these abundantly available data are not considered on analytical purpose. Enterprises now need better business intelligence tools to utilize the full potential of ‘Big Data’ that are available. This in turn will help them take analytical decisions with innovative ideas and firm business decisions in real time. [22, 2]

Researchers define Big Data as "high-volume, high-velocity and high-variety information assets that demand cost-effective and innovative forms of information processing for enhanced business insight and decision making”. Most of these data are unused and that is why some researchers call it dark data. Like dark matter in physics, dark data cannot be seen directly, yet it is the bulk of the organizational universe.[22, 2]

High-velocity in Big Data is attached to real-time analytics. It is about the rate of changes that linking data sets arrived with different speeds and other activities. The volume expresses the amount of Big Data available (such as in social media). [22, 2; 3]

On a daily basis businesses today create 2.5 quintillion bytes of data which is so much that 90% of the data in the world today has been created in the last few years alone. These data generates from everywhere: sensors used to collect different information, posts and conversations in social media sites, digital pictures and videos, transaction records and cellphone signals among others. These data are Big Data. Thus Big Data are beyond the matter of size. It is an opportunity to find new insights about emerging types of data and content to make businesses more agile and to solve potential problems that are considered beyond the reach of business intelligence. [22, 3]

New and emerging analytical processing technologies such as operational business intelligences that enables automated real-time action and intraday decision-making, faster hardware ranging from multi-core processors and large memory spaces to solid state drives, cloud data storage, on-demand software-as a service(SaaS) analytical solutions in public clouds and data platforms, make possible to exploit such big data. Data warehouse systems today are shaping up to fit Big Data architecture so that unstructured data can be extracted from multiple sources such as Internet-based businesses and social networking sites. The Big Data, which is taken from within and outside the organization, is then used for analytical reporting purposes. [22, 3-4]

There are a number of educational institutions, government and non-government organizations and investment ventures that provide millions of dollars for emerging new technology researches. One of these emerging technologies is cloud computing. With the help of this technology, researchers are now challenging themselves on using big data over cloud data warehouses and the usage of these data for data warehousing needs. [22, 5]

Many companies have traditional data warehouses that were not designed to handle big data. However nowadays there are ways to evolve these traditional data warehouses into 'analytics warehouses' so that processing structured and unstructured data could be possible. Traditionally data warehouses were designed to handle structured data, not the type of unstructured data from social media, mobile devices, web traffic and other sources that are feeding data into enterprises now. However vendors are building new generation data warehouses with statistical capabilities for performing analytics and forecasting. New generation data warehouses are designed to support structured and unstructured data. Thus enterprises will be capable of having all-rounded views of their operations in order to make better decisions about the future. [23]

Basically, the analytics warehouse servers as a central repository for an enterprise's structured and unstructured data. In traditional data warehousing architectures, structured data from different sources, file shares and line of business applications are processed into the data warehouse by using ETL (extract, transform and load) database processes. The architecture for the analytics warehouse can be built on the traditional data warehouse architecture [23]:

- A distributed file system, which sits between source data systems and the data warehouse so that it collects, aggregates and processes massive volumes of unstructured data and stages it into the data warehouse.
- Structured and unstructured data from backend systems which can be brought into the data warehouse in real and near-real time.
- Engines that use predictive modeling techniques to identify patterns in big data and support real-time decision making. [23.]

Many companies are now considering their traditional data warehouse infrastructures to integrate big data technologies, analytics technologies and other emerging business intelligence technologies to exploit the full potential of data in their data warehouse for better analysis and decision support.

3 Data Vault Modelling

The research and development on the Data Vault modelling approach began in the early 1990s by Dan Linstedt and was completed around 1999.The following three years the Data Vault design was tested, refined and deployed to selected adopter sites. In addition to that a series of articles on Data Vault modelling were released by Dan Linstedt. In 2002 the Data Vault architecture was reviewed to meet the needs of the data warehouse in industries. In 2003 teaching the Data Vault modelling techniques began to the public. According to its creator Dan Linstedt, the Data Vault is “a detail-oriented, historical tracking and uniquely linked set of normalized tables that support one or more functional areas of business. It is a hybrid approach composed of the best breed between 3NF and star schema. The Data Vault is designed to be flexible, scalable, consistent and adaptable to the needs of the enterprise.”[9, 9]

The Data Vault focuses on data integration, traceability, auditability and resilience to change. It is particularly strong in tracing of data warehouse data back to their source system ensuring a complete audit trail. The Data Vault Model can easily be adapted to varying requirements and it is highly scalable. It is based on principles that are related to normalized data models in such a way that the Data Vault model consists of structures which resemble the traditional definitions of star schema and 3NF including dimensions, many- to- many relations and standard table structures. However there is a difference in relationship representations, field structuring and granular time-based data storage. With the volume of information warehouses have to store exponentially increasing every year, 3NF and star schema have issues when it comes to scalability, flexibility and granularity of data. The Data Vault is modeled to solve these bottlenecks of data warehouses with its core components, the hubs, links and satellites. [19, 8]

The Data Vault can be applicable in Dynamic Data Warehousing where dynamic automated changes are made to both process and structure within the warehouse. It allows users to explore the structures of the data warehouse without losing the content. In addition, the Data Vault could be applied in In-Database Data Mining by allowing data mining tools to use the historical data and to fit with the functions of artificial intelligence. [19, 9]

3.1 Data Vault EDW Architecture

The foundation the Data Vault architecture is rooted in reduction of redundancy. The Data Vault architecture fits with the traditional Bill Inmon's Corporate Information Factory (CIF) approach in such a way that the Data Vault plays the role of centralized enterprise data warehouse (EDW).In this case the Data Vault provides data to star schema data marts, report tables and exploration marts.[2,2] Figure 4 explains the Data Vault EDW architecture.

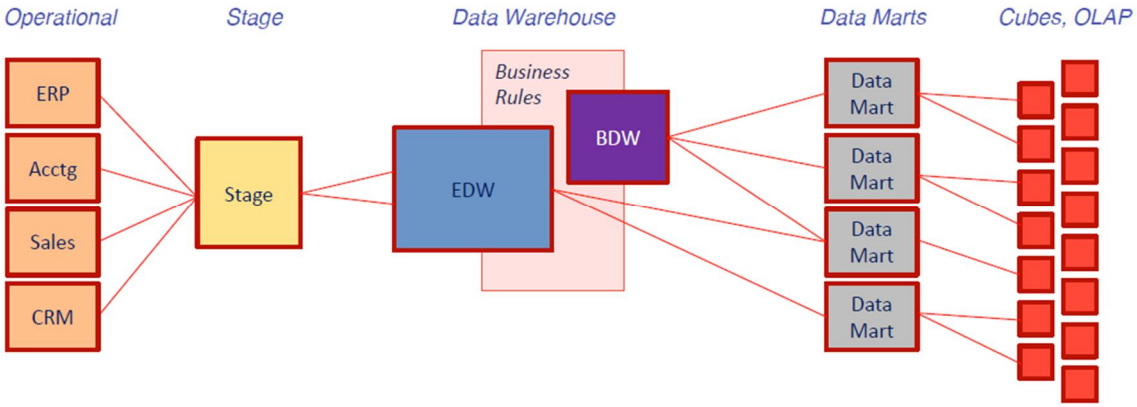

Figure 4. Data Vault EDW Architecture. Reprinted from Enterprise Wide Business Key [18, 17]

Figure 4 represents high level Data Vault architecture in enterprise data warehouse. Business rules in BDW (Business Data Warehouse) layer could be applied in the same or separate area of the EDW (Enterprise Data Warehouse) layer. Data marts can draw data from the BDW or EDW layers as appropriate. Data in the Stage and Data Mart area are not kept but rather overwritten, so the Stage and Data Mart areas are not persisted. Components in the EDW and BDW layers are commonly persisted. The EDW, which is the core historical data repository, consists of Data Vault modeled tables. The tables in the EDW are composed of hubs, links and satellites which are considered to be the fundamental elements of the Data Vault model. The EDW layer is then comprised of the Data Vault Model where raw granular data sets are stored. The OLAP (On-line Analytical Processing) cubes draw data from data marts which could be used for decision-making, data mining or artificial intelligence tools. [3, 14]

3.2 Data Vault Core Components

There are three core entity components in the Data Vault model. The hub, link and satellite entities. The Data Vault design is focused on the functional areas of businesses with hubs representing the primary business keys. The link entities associate two or more business keys and they are commonly referred as transactions. The satellite entities provide a descriptive context of the primary keys or their associations. [1; 2, 3]

3.2.1 Hubs

Hubs are defined by a unique list of business keys which represent core business concepts such as customer, vendor, sale or product. These business keys are vital to businesses to track, locate, and identify their information. The hub table being a business key recording device is established the first time a new instance of the business key is introduced to the EDW. They also provide additional technical information such as load time stamp, record source and sequence number also knows as the surrogate key. [2, 5]

The purpose of a hub is to trace the first time the Data Vault experiences the business key arrival in the warehousing load and where it came from. It provides a soft-integration point of raw data that have the same semantic meaning to the source system. The hub key allows a business to track their information across their business line, which in turn provides a consistent view of the current state of their systems. [2, 6; 3, 4]

The hub entity table structure consists of a minimum of the required elements such as a surrogate sequence id, a business key, a load date / time stamp and a record source. Hubs are parents to all other tables. Figure 5 shows the basic hub entity table structure for an example business concept customer.

Figure 5. Hub Entity Structure. Adapted from Dan E.Linstedt [3, 4]

The above H_Customer table consists of:

- H_Customer_SID - Hub Sequence Identifiers or a Surrogate Key generally generated from a data warehouse system for proper management in the data warehouse.
- Business Key - a core business entity and an enterprise wide key generally a string to handle any data type used by business in search of context.
- Date / Time Stamp -generally date and time recording when the key itself first arrived in the data warehouse.
- Record source -generally a string which is the recording of the source system utilized for data traceability. [1]

The Hub is logically an independent key and none of its part is dependent on another business key. In some cases there might exist two business keys. In these cases one of the keys is the surrogate key of the other business key. The surrogate key cannot become a Hub when there is another independent key. Thus we can choose to ignore surrogate keys altogether. The surrogate key can be considered as a technical key that can serve as a Hub when there is no other candidate independent key. [2, 6]

3.2.2 Links

Link entities are the glue that pulls together any relationships between two or more business keys and sometimes other Links. They are established the first time a new unique association is presented to the EDW. Links act as the flexible component of the Data Vault model and are created when business keys interact. The Links contain the surrogate sequence ids from the Hubs and Links that it associates to, warehouse sequence id, a load date / time stamp and a record source. Like the Hubs, Links contain no descriptive information. [2, 8; 3, 4]

The job of the Link is to capture and record the past, present, and future associations of data elements at the lowest possible grain. It provides flexibility and scalability to the Data Vault model by allowing changes, modification and adaption to the structure over time without losing auditability and compliance. [2, 8]

Data Vault modelling constructs form a link any time when there is a one-to-one, one-to-many, many-to-one, or many-to-many relationship between business keys or data elements. Although these relationships come with their own benefits and drawbacks, allowing many-to-many relationships within business keys provide the following benefits: [2, 9; 5, 7]

- Flexibility: ability to enable rapid business changes on a large scale without downstream impacts.
- Granularity: ability to detail or summarize the units of data in a data warehouse.
- Dynamic adaptability: ability to be suitable to faster changes and situations in data warehouse.
- Scalability: ability to increase size and scope of the business. [5, 7]

A many-to-many relationship enables rapid data changes on a massive scale with little to no impact on both existing data sets (referred as history) and existing processes (commonly known as load and query).It makes businesses change at the speed of a business and become more agile and responsive to handle those changes. In general by modelling the Links as a many-to-many relationship we can accomplish the above goals. [2, 9]

The Link entity table structure consists of the basic required elements such as surrogate sequence id, multiple business sequence keys from the Hubs and other related Links, load date stamp, and record source. Depending on business requirements for query purposes, items such as last seen date, encryption key and other metadata elements may be added. Figure 6 shows the Link table structure based on an example business concept Customer and Hub table in Figure 5.

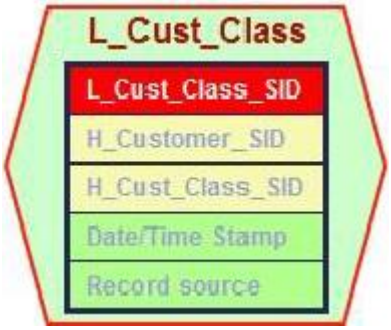

Figure 6. The Link table structure. Reprinted from Dan E.Linstedt [3, 4]

The above structure contains Hub surrogate keys from H_Customer in Figure 5 and H_Cust_Class which is another business concept to which relationship would be extended. Generally the links contain the following attributes: [2, 9]

- Hub surrogate keys: business sequence keys which establish the linkage back to the Hub unique entity.it is considered a mandatory key for the links.
- Link surrogate key: an optional key but it could be considered as a mandatory primary unique key in case of Link associated to other Links. It is highly important in allowing Satellite data to relate to Links in flexible and dynamic manner.
- Loading timestamp: a mandatory key for the links and it audits data recording when an association was first introduced.
- Record source: a mandatory key used for traceability and integrations purposes. [1]

3.2.3 Satellites

A Satellite is a time-dimensional table containing detailed descriptive information which provides context to the Hub's or Link's business keys. A Satellite can have only one parent table about which it can describe. However several Satellites may be used to describe a single business key or association. There are different approaches of designing and building Satellites hence the most common ones include arranging by subject area, type or classification of data, data source system and rate of change of data. Satellites do not have their own Sequence ids and cannot contain foreign keys except the Hub or Link to which they are attached to. Mandatory attributes for a Satellite includes Hub or Link Sequence Identifier, Load Date/time and Record Source. [2, 11; 4]

Descriptive information in a data warehouse is often exposed to changes. The purpose of the Satellite is therefore to capture all changes to any descriptive data which occur over time. Furthermore Satellites provide detailed descriptive data about the business key or the relationship of the business keys. Their job is to track data by delta (changes), to record data as it is loaded from the source system and to describe relationship changes over time. Record life-cycles of the Satellites are indicated by load dates and load-end-dates.

The Satellite entity table structure consists of basic required attributes such as surrogate sequence id from the Hub's or Link's table, load date stamp and record source. The primary key of the Satellite is a two part key combining the Sequence key of the parent (Hub or Link) with the Load Date time stamp. Real-time data can easily be added in to Satellites by adding millisecond timer and without creating duplicate primary keys. Figure 7 shows the Satellite entity table structure based on a business concept Customer which is also used for Hub's and Link's structure tables above.

Figure 7. The Satellite entity table structure. Reprinted from Dan E.Linstedt [3, 5]

Components of the Satellite table such as Data/Time Stamp and Record source are similar to the Hub and Link table's entities. These entities are very important in tracking history of data by exposing the date and time the data was entered to the system and where the data came from. Some of the entities figure 7 contains are explained below.

- H_Customer_SID: is the Hub's Sequence id migrated to the Satellite. It is used as one of the primary keys for the Satellite table.
- Date/Time Stamp: is a load Date /Time stamps recorded when the context information is introduced in the warehouse and it is also used as one of the primary keys for the Satellite table. Thus both H_Customer_SID and Date/Time Stamps are mandatory keys for the Satellite table.
- Descriptive information about the Customer Satellite (S_Customer) is indicated as Context A, Context B, Context C and Context D.
- Record source: is a recording of the source system which is important for data traceability. [1]

3.3 Constructing the Data Vault Model

Hubs, Links and Satellites are the building blocks for the Data Vault in an EDW. The Hubs and Links together create the backbone or skeletal structure of the model and the Satellites provide all the context and all the history (changes over time).The skeletal structure represents a one-to-one relationship with the core business concepts (Hubs) and their business associations (Links).Adding Satellites to the backbone structure will form a complete data vault model and all the core constructs together represent all integrated data from the organization.

The process of modelling with the Data Vault involves several steps. The first step though is to identify business concepts for the given subject area. Identifying business concepts help to establish EWBK (enterprise wide business key) for Hubs, which in turn will lead to defining or modeling the Hubs. Once the Hubs are defined, the next step is identifying the natural business relationships between the Hubs. Identifying relationships between Hubs help to model Links. Now the skeletal structure could be formed using the modeled Hubs and Links. The next step is to gather context attributes to define keys and establish criteria to model satellites, which will provide descriptive information to the Hub and Link constructs. [3, 6]

There are several rules which should be considered when modeling a Data Vault. Some of the reference rules for Data Vaults are the following and the detailed rules are provided in appendix 1:

- A Hub table always migrates its primary key outwards: Hub keys cannot migrate into other Hubs, which means there should not be parent or childlike Hubs.
- Hub-to-Hub relationships are allowed only through a link structure: Hubs must be connected through links and more than two Hubs can be associated through links.
- Recursive relationships are resolved through a link table.
- A Link structure must have at least two foreign key relationships: Links must have at least two Hubs connected to them in order to be instantiated.
- A Link structure can have a surrogate key representation: Surrogate keys may be utilized for Hubs and Links but not for Satellites.
- A Link structure has no limit to the number of Hubs it associates.
- A Link-to-Link relationship is allowed.

- A Satellite can be connected to Hubs or Links.
- A Satellite can only have one parent table.
- A Satellite cannot have any foreign key relationships except the primary key to its parent: Satellites often contain a load date-time stamp, or a numeric reference to a standalone load date-time stamp sequence table.
- Satellites are often change-driven and duplicate rows should not appear.
- Data is distributed into Satellite structures based on the type of information and rate of change.[1]

The backbone structure (Hub and Link) and Satellite combine to form a Data Vault. A Data Vault can be as small as a single Hub with one Satellite, or it could be as large as the business scope permits. The Data Vault allows scope modification. Neither scalability nor granularity of the information is not an issue for scope modification. A Data Vault model with an example Customer business concept is provided in Figure 8.

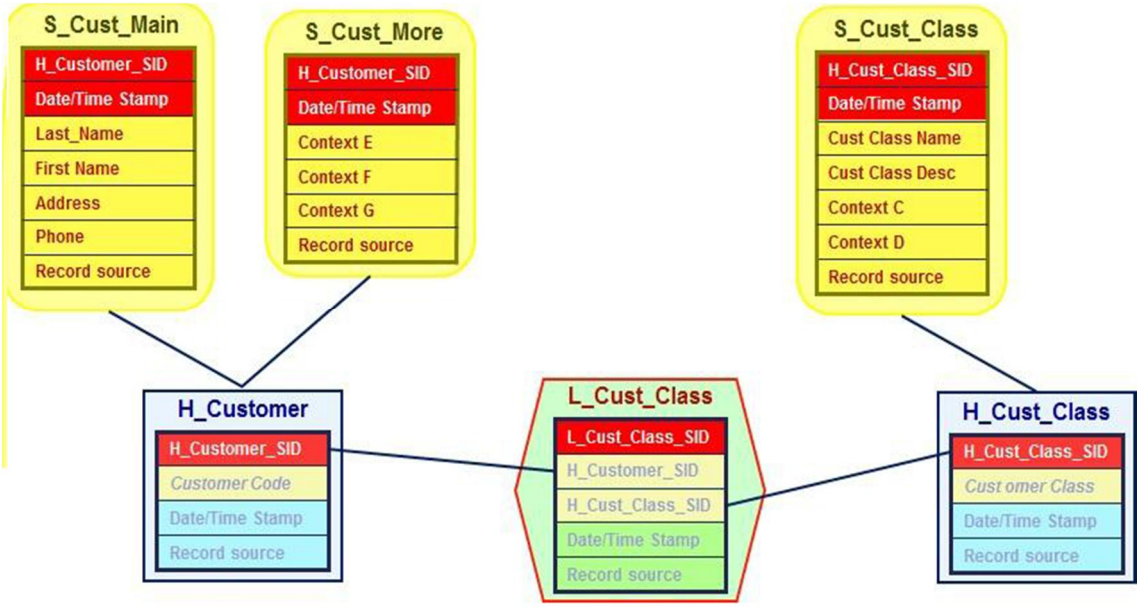

Figure 8. Complete Data Vault Model. Reprinted from Dan E.Linstedt [3, 5]

Figure 8 shows how simple a Data Vault model could be constructed, the connection between the skeletal structure and their descriptive information. A full example of a sample Data Vault model is provided in appendix 2.

4 Loading Practices of the Data Vault Model

Loading a data warehouse is the key activity of data processing. It is performed in different processing modes. A processing mode is the way in which data are grouped and handled to the warehouse for the purpose of generating reports, computation, analysis and other outputs. Traditionally data warehouses are loaded with data from operational systems in a batch mode. The Batch mode (batch processing) is a method in which a number of similar transactions to be processed are batched (grouped together) as a single unit at predetermined times. These accumulated transactions are then presented to the data warehouse at regular intervals of time. It means that data in the warehouse become available as extracted file on nightly, weekly or monthly schedule after the system has transformed and cleansed it. As a result data warehouses do not contain today's data for decision support and this caused problems in a fast changing business decisions. [19, 3]

Another way of data processing is on-line processing. It involves an On-line-direct-access system, as data enters the system from the point of origin and it is transmitted directly to the end user as specific output format. This system enables direct communication between a data processing unit and an output point. It could be used to process transactions, update files and other inquires. Even though it is costly, on-line processing ensures that data are in the updated status at any time, which is not the case in batch processing where data get updates at regular intervals of time. [19, 4]

Today businesses demand information as fresh, timely and accurate as possible to make up-to-date decisions. Thus real time data warehousing was introduced to capture source changes and immediately apply to the data warehouse. Real time refers to the method of updating files with transaction data immediately after the event to which it relates occurs. Real-time systems are on-line systems with tighter constraints on response time in such a way that data is processed and results are generated quick enough to influence ongoing activity. In a real-time loading system, only a small volume of data is processed at any one time. The system responds to and often controls its environments in order to keep pace with the loading process and other external events. Real-time loading systems have a fast response time (in a fraction of a second); thus business decisions could be made on the basis of the latest information available from files which get continuously updated. The Data Vault as the data warehousing solution supports both real time and batch loading mechanism. [19, 3]

There are several objectives to be achieved when a business chooses among different data warehouse loading mechanisms. The following are some of the goals and objectives for loading the Data Vault. [20]

- Consistent Process Architecture: Loading Hub, Link, and Satellites has to be consistent and exactly the same process.
- Restartable Process: Processes built should recognize what has been loaded already and be restartable without requiring change to the process itself in the case of process breaking.
- Maintaining full Traceability of Data: Providing information when data is changed, what kinds of changes are applied, and probably who changed it.
- Data Driven Design: During loading process all data should make it to the data warehouse (Data Vault) and thus there should not be any data drop out of the load as the Data Vault supports “all of the data all the time”, no matter how bad it is.
- Real-Time Provisions: All Data Vault loading templates should support real-time loading.
- All of the data Loaded to the Data Vault all of the time: All data entering the data warehouse must be loaded to the Data Vault all of the time regardless of how bad the data formats or values are.
- Large Data Scalability: The loading process should be linearly scalable both in volume and timing.
- Tracking and Traceability: The loading process should contain load-dates, load-end-dates, last seen dates and other means for tracking data and tracing it back to the sources system. [20]

4.1 Hub Loading

Hubs record the first time the business key reached the Data Vault. They do not record additional loads or other changes in data. Before loading a hub to the warehouse it is always advisable to check if it already exists in order to avoid information duplication and broken load process. Loading a hub could be simple in consideration that all the hub business keys must be unique and one-to-one with surrogate keys, hub keys are not time-based, hub keys do not support duplicates and hub keys are defined at the same semantic layers. Identifying the master system where most business keys are generated is the first step of loading a hub. These keys in the staging table that are targeted to the hub are loaded as distinct or unique list and provided a surrogate key at load time. [21]

The process of loading is repeated for each source system with a hub key field. The field used to populate the hub key may not have any relation with foreign key in the source system design. If a key exists, it must then be placed in the Hub. Information with no key can be attached to a hub row called zero row key or UNKNOWN, so that the data is not lost during the load of the Satellites. [21]

Hub duplication can cause issues ranging from query problems to complicated modeling problems. Thus it is recommended to avoid loading Hub keys with duplicates. However in the case of bad data, including the record source as part of the business key could offer uniqueness to the Hub keys. In actual batch loading processes, all the hubs can and should be loaded in parallel so as to produce a highly scalable architecture with more data and more Hubs. [21]

In a real-time loading system the business keys almost always arrive with transactions. In real-time business keys should be provided by the system to help the source system attach the data to its own business processes. Real-time hub loading follows the same cycle as the batch hub loading in such a way that it inserts the key if it does not already exist. In addition to this architectures that handle millisecond feeds use a millisecond time-stamp hub as their key. [20]

4.2 Link Loading

Link tables without surrogate keys follow the same loading process as Hubs. However, since most enterprise data warehouses today use surrogate key structures, the load process in the Link tables is extended to locate Hub rows containing surrogate keys. Loading the Link tables consist of, formulating a unique list of all the business keys that build up the relationship for the Link, locating each of the Hub's relating surrogate keys and inserting the key with a new Link surrogate if it does not already exist in the target. If the key already exists, it will be filtered out from the feed. [21]

The objective of loading the Links is to avoid duplicates across the business key structures in such a way that each relationship or intersection must be unique. Duplicates of business keys in Link tables usually cause a poor definition of grain (meaning of single row in the table) or mis-modeled Link entity. If there is a grain problem in the Link table model in the Data Vault, the process cannot find an old satellite record to end date even though it exists. To avoid this Links must be designed to fit the grain. [21]

Link table loads are considered successful when all relationships between business keys have been made. In case of missing or "null" business keys in links, load "zero" keys and tie the relationships to "unknown" Hub records. Thus data loading will be continued into the model without breakages. All the relationships have to be captured to provide a full audit trial capability and trace the information back to its source system. Like Hub loading, all Link loads can be run in parallel. [21]

Link loading in real-time could be easy as the Hubs match keys to business keys and this matches are built into a new staging surrogate table. Thus Link load time is reduced. Millisecond feed transactions are kept with millisecond hub stamp and are joined across the links via a process that run once a second or few seconds to build the associations. Since the information in real-time arrives so quickly, introducing parallelism to the process can avoid complexity out of the direct loading path. [20]

4.3 Satellite Loading

The main objectives of Satellite loading are to load delta changes and to split the work by type of data or rate of change. The process of batch loading on a Satellite involves gathering the link or hub surrogate key, selecting rows that have changed, joining to a current load-date table and comparing the most recent data in the staging area to the Satellite. Sometimes Satellites require Link PK's (Link sequence number) so loading process also involves putting the new load-date into the PK of the Satellite. Finally track the rows and insert the new Satellite attributes as simple as possible to end-date old rows in the next pass and to update bridge tables. [20]

Establishing a unique list of Satellite attributes avoids loading duplicate attribute rows into the Satellite target.it is suggested to have default values as part of the process to utilize default data sets for some columns. There must be either the Hub key or the Link parent key to attach the Satellite row properly. If parent key does not exist, loading process must be stopped and design should be fixed for reuse. Furthermore comparison of the columns of the data coming in with the most recent Satellite data should be made to ensure not to load duplicate data. Using end-dating process can make the loading process fully restartable and scalable. The most current Satellite record that has not been end-dated will be end-dated during the run. End-dating Satellite helps to identify which data is active. [21]

5 Merits and Demerits of the Data Vault

As a hybrid data modelling approach Data Vault arguably consists of the best modelling solutions from the Third Normal Form (3NF) and Dimensional Modelling. However when it comes to applicability Data Vault has its downsides. Table 2 explains a selected list of merits and demerits of the Data Vault as an Enterprise Data Warehouse.

Table 2 Merits and Demerits of the Data Vault as an EDW. Data gathered from Data Vault Overview [17, 16]

Merits	Demerits
completely auditable architecture	requires highly structured architect role
highly adaptable to (business) changes	performance issue due to scalability
designed and optimized for EDW	additional storage of data
scalability(provides multi-terabyte storage)	lacks strict or formal implementation guidelines
supports real-time processing	requires business analysis to be firm
can be incrementally built and easily extended	not conductive to OLAP processing
supports data mining and artificial intelligence	introduces many joins and tables in queries
supports flexible / extensible development	not intended for end user access(such as BI tools and OLAP)

The Data Vault as a hybrid data warehouse solution allows scaling to a large extent which leads to more joins and tables in queries. In other words tables need to be joined to achieve the result when making analysis or building a report. Although the concept of the Data Vault is well documented there exists no strict implementation guideline, which leads different people to apply different views on implementing it. Generally the data warehouse layer contains a copy of data which could be considered as redundant. Due to this we need additional storage area to handle data.

6 Conclusion

Data Vault modeling is very important when modeling a data warehouse. It contains a wide range of modelling features for enterprise data warehouse projects. One of the key features of Data Vault modeling is its ability to adapt to changes in requirements both in sources and data marts. Using this modeling technique an enterprise could be able to get the benefits of building dynamic data warehousing depending on dynamic automated changes to the process and structure within the warehouse. Moreover it helps to explore warehousing on which users could play with the data warehouse structures without losing any content.

Another benefit of using Data Vault modelling in an enterprise is data mining. The Data Vault model allows data mining tools to make use of the historical data and fit the structure with the function of artificial intelligence. In addition to that it has an ability to rapidly link and adapt to the structures of an external information source and create useful data in the data warehouse without losing an existing content.

In contrast to a wide range of benefits, using Data Vault modeling has its own business and technical issues. Data in the Data Vault are non-user accessible. Data Vault modeling requires more up-front work even though it is for long-term benefits. Data in Data Vault are not cleansed or its quality is not checked. Data Vault model introduces many database join operations and it is not conductive to an OLAP (online analytical processing) processing. Thus it requires a firm business analysis.

With a good range of benefits and limited disadvantages the usage of the Data Vault approach is growing globally. Currently there are more than 500 companies which exist in different parts of the world and which are using the Data Vault. IBM, Intel, Microsoft Corp., Teradata Corporation are among the technology giants which use the Data Vault. In general with the Data Vault team actively working on bringing the best modeling approaches, rules and practices to meet the industrial needs for data warehousing, using the Data Vault could be seen as an up-to-date alternative approach for data warehousing.

References

1. Dan E. Linstedt. Data Vault Series 1 - Data Vault Overview [Online]. [Accessed 2013 Dec 15];
   URL: http://www.tdan.com/view-articles/5054/
2. Kent Graziano, Dan Linstedt. Introduction to Data Vault Modeling [e-Book]. [Accessed 2014 Jan 3];
   URL:http://kentgraziano.files.wordpress.com/2012/02/introduction-to-data-vault-modeling.pdf
3. Dan E. Linstedt. Data Vault Modeling Guide [e-Book]. [Accessed 2014 Mar 31].
   URL:http://hanshultgren.files.wordpress.com/2012/09/data-vault-modeling-guide.pdf
4. Jonathan Shirey. Data Vault: Hubs, Links, and Satellites with Associated Loading Patterns [Online]. [Accessed 2014 Jan 19].
   URL:http://makingdatameaningful.com/2012/09/18/data_vault-hubs_links_and_satellites_with_associated_loading_patterns/
5. Martino Adriano. Further in Agility with Data Vault [e-Book]. 2013 05 [Accessed 2014 Feb 10];
   URL:http://www.trivadis.com/uploads/tx_cabagdownloadarea/Further_in_agility_with_data_vault_01.pdf
6. Ariyachandra HJW, Thilini. Data Warehouse Architectures : Factors in the S e lection Decision and the Success of the Architectures. Terry College of Business University of Georgia [e-Book]. 2005 Jul [Accessed 2014 Feb 19];
   URL: http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf
7. Erik Perjons. DW Architecture and Lifecycle [e-Book]. [Accessed 2014 Jan 19].
   URL: http://people.dsv.su.se/~petia/is5/Lectures/F4.pdf
8. Data Warehouse Architecture [Online]. [Accessed 2014 Jan 25];
   URL:http://www.1keydata.com/datawarehousing/data-warehouse-architecture.html
9. Kent Graziano. (OTW13) Agile Data Warehousing: Introduction to Data Vault Modeling [Online]. 2013 [Accessed 2014 Jan 30].
   URL: http://www.slideshare.net/kgraziano/agile-data
10. Kent Graziano. Why Data Vault? [Online]. 2011 [Accessed 2014 Jan 25].
    URL: http://www.slideshare.net/kgraziano/why-data-vault

11. Oracle Corporation. Oracle9i Data Warehousing Guide [Online]. [Accessed 2014 Jan 30]; Release 2 (9.2).
URL: http://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm
12. Robert W. Dill LC Unites States Navy. Data Warehousing and Data Quality for a Spatial Decision Support System [e-thesis]. 1985 [Accessed 2014 Feb 20].
URL: https://archive.org/stream/datawarehousingd00dill#page/n7/mode/2up
13. Brackett, Michael H. The Data Warehouse Challenge. New York, NY: John Wiley & Sons, Inc.; 1996.
14. Inmon, William H. and Hackathorn, Richard D. Using the Data Warehouse. New York, NY: John Wiley & Sons, Inc.; 1994.
15. Paul Lane. Oracle Database Data Warehousing Guide [e-Book]. Release 2 (10.2). 500 Oracle Parkway, Redwood City, CA 94065: Oracle Corporation; 2005 [Accessed 2014 Mar 5].
URL: http://docs.oracle.com/cd/B19306_01/server.102/b14223.pdf
16. Oracle Corporation. Oracle8i Data Warehousing Guide[Online]. [Accessed 2014 Mar 10]; Release 2 (8.1.6).
URL: http://docs.oracle.com/cd/A87860_01/doc/server.817/a76994/marts.htm
17. Empowered Holdings, LLC. Data Vault Overview [Online]. 2011 [Accessed 2014 Mar 12].
URL: http://www.slideshare.net/dlinstedt/data-vault-overview
18. Genesee Academy, LLC. Enterprise Wide Business Key: Aligning the Business w/ EDW & MDM. 2010 [e-Book] [Accessed 2014 Mar 20];
URL:http://www.enterpriseiq.com.au/documents/pres/DW_BI_2.0_Summit_D1_S6_Hans_Hultgren_2010_Genesee.pdf
19. Oracle Corporation. Best Practices for Real-time Data Warehousing [e-Book]. 2012 Aug [Accessed 2014 Mar 21];
URL:http://www.oracle.com/technetwork/middleware/data-integrator/overview/best-practices-for-realtime-data-wa-132882.pdf
20. Dan Linstedt. Data Vault Loading Specification v1.2 [Online]. 2010 May 13 [Accessed 2014 Mar 24];
URL: http://danlinstedt.com/datavaultcat/standards/data-vault-loading-specification-v1-2/
21. Dan E. Linstedt. Data Vault Series 5 - Loading Practices [Online]. 2005 Jan 1 [Accessed 2014 Mar 25];
URL: http://www.tdan.com/view-articles/5285

22. Lt. Dr. Santhosh Baboo & P. Renjith Kumar. Next Generation Data Warehouse Design with Big Data for Big Analytics and Better Insights. Global Journal of Computer Science and Technology Software & Data Engineering [online].2013 [accessed 2014 April 9]
URL: https://globaljournals.org/GJCST_Volume13/3-Next-Generation-Data.pdf
23. The Future of Data Warehouses in the Age of Big Data, Deloitte CIO Journal Editor [online]. July 17, 2013.[accessed 2014 April 9]
URL: http://deloitte.wsj.com/cio/2013/07/17/the-future-of-data-warehouses-in-the-age-of-big-data/

Data Vault Rules v1.0.8 Cheat Sheet [3]

Data Vault Rules v1.0.8 Cheat Sheet

Hub

A **Hub** is a list of unique business keys with low propensity to change

Required fields

- PK: (Hub) Sequence ID
- DataVault Load Date/Time
- Record source
- At least one Business key

Optional fields

- Manual update user
- Manual update timestamp
- Sourcesystem extract date
- Last seen in DataVault date

Rules

- No composite set of business keys (*)
- Have at least 1 satellite
- Business keys SHOULD be true ID's of business entities *(license plate, SSN, ordernumber, etc.)*
- Business keys CAN be synthetic source-system created keys
- Business keys MUST stand alone

(*) This is not the same as "no composite keys"

Link

A **Link** is a list of (n to n) relationships between business keys

Required fields

- PK: (Link) Sequence ID
- DataVault Load Date/Time
- Record source
- At least two "imported" Hub or Link Sequence ID's

Optional fields

- Manual update user
- Manual update timestamp
- Sourcesystem extract date
- Last seen in DataVault date

Rules

- A link indicates (timeless) existence of a relationship
- The imported keys composite must be unique
- Satellites are optional
- We can end-date links using an 'exists'-attribute in a satellite
- Links exist at the lowest level of detail for the imported entities

Additional rules for hierarchies

- Hierarchical relationships are implemented using a link between exactly two hubs: the Child and Parent Hubs
- End-dating links is mandatory
- The Child-key determines the end-of-life of the relationship

Satellite

A **Satellite** contains the historical data associated with the Hubs and Links

Required fields

- PK: Link or Hub Sequence ID
- PK: DataVault Load Date/Time
- Record source
- DataVault Load End Date/Time

Optional fields

- Manual update user
- Manual update timestamp
- Sourcesystem extract date
- Last seen in DataVault date
- PK: Sequence ID for uniqueness

Rules

- Contains all non-key data
- Attached to exactly one Hub or Link
- Has at least one descriptive field
- Data is grouped into separate satellites according to type, rate of change and source
- References (not foreign keys) to stand-alone Reference tables or single Hub/Satellite combinations are allowed
- Insert 'default' satellite rows for new hub/link keys without data to avoid outer joins

Reference

A **Reference table** is an a-historical lookup table

Required fields

- PK: Reference(d) code
- One or more attributes

Rules

- Ref-tables are stand-alone tables
- Ref-tables do not store historical data
- Ref-tables are used for performance improvement or lookups on the way to the reporting layer/data mart

Naming

Recommended naming conventions

Entities

- Hubs: HUB or H prefix or suffix
- Links: LINK or L prefix or suffix
- Hierarchical links: HLNK or HIER prefix or suffix
- Satellites: SAT or S prefix or suffix
- Reference tables: REF or R prefix or suffix

(Prefix to sort on table type. Suffix to sort on business concept)

Fields

- Record source: REC_SRC or RECORD_SOURCE or prefix/suffix with RCSRC or RSRC
- Sequence ID's: SEQ_ID or SEQUENCE_ID or prefix/suffix with SQN
- Date/time stamps: prefix or suffix with DTS
- Date stamps: prefix or suffix with DT
- Time stamps: prefix or suffix with TM
- Load Date/Time stamps: prefix or suffix with LDDTS
- Load End Date/Time stamps: prefix or suffix with LEDTS
- User watch fields: prefix or suffix with USR

Complete Data Vault Model Example [3]