IJCSI International Journal of Computer Science Issues, Volume 13, Issue 4, July 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org http://dx.doi.org/10.20943/01201604.1631

Effectiveness of Data Vault compared to Dimensional Data Marts on Overall Performance of a Data Warehouse System

Zaineb Naamane[1], Vladan Jovanovic[2]

[1] Computer Science Department, Georgia Southern University
Statesboro, 30458, USA

[2] Computer Science Department, Georgia Southern University
Statesboro, 30458, USA

Abstract

The purpose of this pilot study was to compare the traditional Kimball's approach (using star schemas Dimensional Model, directly from the data sources source) to define Data Marts vs. newest Linstedt's approach (using Data Vault 2.0 modeling of EDW with virtual/materialized Data Marts as star schemas). The comparison is performed experimentally using a specifically designed, realistic and broadly applicable case study. The study includes implementing a Customer Experience Data Mart in the telecommunication domain extending the work done by the TM Forum's standards/guidelines demonstrating a feasibility of a complete implementation. A generic application implementation is completed with experimental test data and our comparative analysis included both qualitative and quantitative elements.
The paper also presents a novel systematic technique for data vault modeling, in the presence of highly generalized source entities, emerged from this experiment. This satellite per subtype technique is devised to fit any highly generalized industrial source data model.

Keywords*: Data Warehouse (DW), Data Vault (DV), Extract Load Transform (ETL), Data Mart (DM), Raw DV, Dimensional Model, Hub, Satellite, Link*

1. Introduction

Strategic decisions increasingly rely on data warehouses which provide a multidimensional, clean, and well-organized view of the data coming from several operational sources. To discover trends and critical factors in business, the analytical results generated by the reporting tools need to be accurate and reliable, which means that the data warehouse needs to be carefully designed.

The common assumption about DW is that once created it remains static. This is incorrect because business needs change and increase with time, and business processes are subject to frequent change. As a consequence of this change, a new type of information requirement that requires different data becomes necessary. These new requirements lead to changes in the DW schema and need to be incorporated in the data warehouse system while ensuring accurate insight of the business data. The data warehouse should be designed in a way that allows flexibility, evolution, and scalability. Moreover, as data sizes are steadily increasing, a data warehouse should not only be able to scale but also support velocity and variety of incoming data [1]. When data volume grows and the amount of querying and reporting increases, performance and complexity issues also rise. To encounter these problems, an EDW /BI system should adapt to a changing business environment, support very large data, simplify design complexities, and allow addition and removal of data sources without impacting the existing design.
Designing such a robust system motivated focus of this paper on exploring the Data Vault methodology (DV 2.0), which is an agile data modeling solution for a system of record. It solves primary business requirements and addresses flexibility and scalability. The study conducted in this paper aims to measure the effectiveness of a Data Vault based system compared to a traditional DW system (i.e dimensionally modeled Kimball style one).

2. Problem Statement and research objectives:

To meet technical expectations, data warehouse engineers can use various architectures to build data warehouses. Common data warehouse architecture is based on layered

approaches, which is often the case in information systems. One of these typical architectures is two-layer architecture, which refers to Kimball Data Warehouse style. It contains only two layers that are part of the data warehouse system itself: a temporary staging area that contains an exact copy of all data that should be loaded into the data warehouse, and a data warehouse layer modeled after the dimensional model made up of data marts representing subject areas. This type of architecture doesn't require any additional layer, as a dimensional model is directly queried to present the information to the user. This architecture is easy to implement but is more complex when the source changes and is not able to provide information about source and extraction time of data stored within the system [1].
As today's business environment is characterized by rapidly changing conditions, it is common that business requirements change frequently. For that reason, data warehouse developers should carefully select adequate architecture. Thus, the goals of our experimental research here are:

- To design and implement a Customer Experience data mart, as a realistic case study, using two design approaches (Kimball DW and Data Vault 2.0 [1])
- To measure and compare the impact of each implementation on an EDW in terms of load performance, traceability, auditability, scalability and flexibility.

These goals are designed to present the importance of the architecture choice when implementing a data warehouse and the impact of this choice on the overall performance of a data warehouse system.

3. Research Environment:

3.1 Software Specifications:

All tests and experiments were carried out on the same test machine. All non-essential programs and services were shutdown to reduce the number of uncontrolled variables in the system and allow maximal resource utilization. The following Microsoft applications are used in this study: Microsoft SQL Server 2014, Microsoft Visual Studio 2012 and SQL Server Data Tools for Visual Studio 2013. CA Erwin Data Modeler r9.64 is used for Data requirements analysis and Data design.

3.2 Study Inputs:

a) TM Forum Standards

The main goal of this study is to measure the impact of an alternative DW implementation (DV 2.0) on the overall performance of a data warehouse system. In order to achieve this goal the study data set needs to reflect real-world situations. The idea behind this study is to offer an alternative standard decision-making solution focused on the measuring and monitoring of the customer experience in the telecommunication domain using DV2.0 and compare it to a traditional BI solution.
TM Forum [2] has been one of the organizations who addressed general problem of customer experience, through the publication of guide books and reinforced by various initiatives of its members in expanding the understanding of this complex subject. In order to set a decision-making system based on the measurement of customer experience key performance indicators, one needs to have a full view of the enterprise in terms of the business that directly impacts the relationship with the customer, as well as the information used by these processes. The study input will be the results of a scrupulous study previously done to establish links [3], through the use of TM Forum standards related to the three aspects:

- Processes: using eTOM (business process framework that represents a hierarchical catalog of the key business processes required to run a service focused business[2])
- Information: using the information framework (SID) that provides a reference model and common vocabulary for all the information required to implement Business Process Framework (eTOM) processes [2]. This will be used to design our project data sources.
- Key performance indicators: using standardized business metrics that capture performance indicators within the domain of Customer Experience.

These standards and results will be used to implement an alternative Business Intelligence solution supporting decision-making in the customer experience domain using the data Vault methodology, this solution will be compared to the Kimball BI solution [3] in terms of requirements analysis, performance, and flexibility to business changes.

b) Case Study from Previous Work:

Based on the TM forum definition of the customer experience there are four cornerstones of the customer experience: brand image, pre-service marketing, in-service quality management, and in-service customer facing processes which are concerned with modeling the customer's interaction with the service provider. They start with the customer initiated contact and end with the fulfillment of the request.
eTOM is the framework that identifies the business processes of a service provider. When modeling the business process, we have to involve some business entities. In this paper, we will focus on the order-to-payment process. In this process, a customer selects from the product catalog a product offering. During this

interaction, the customer also chooses some specifications of the product he is ordering (for example the type of the physical resource that will be delivered) along with the resource, and as part of the product, he also buys a service (for example a communication service as DSL). The process goes on until the order is fulfilled and successfully delivered to the customer.

An in-depth analysis of the order to payment process has been done previously using the three TM Forum Standards: eTOM, SID and business metrics [3], [4], [5], [6]. This analysis explicitly details the business entities that play an important role in our specific scenario such as customer, product catalog, product, service, resource, offering, order and interaction. This paper does not focus on that analysis, but its findings [3] will be used as requirements for our DW experiment.

4. Data Source:

All data model examples are shown as diagrams (from the case tool ERwin 9.64) using appropriate style-related variations based on the standard Idef1X data modeling notation.

The object of this section is to build a database model for generating the necessary data sources. Our database model is based on business entities from the SID framework that are related to the order to payment end-to-end process [3]. However, it was necessary to explore the Information Framework (SID) [6] in order to extract the existing associations and aggregations between these class concepts along with the relationships to the Common Business Entities [6].

However, the SID model is designed in UML; we recommend to use Idef1X standard notation (instead of the UML) as a data modeling language specialized in DB for its strict standardization modeling. One other reason for selecting Idef1X is its minimal set of visual data modeling elements and very convenient closeness to relational model i.e. a direct visual representation of tables and FK.

4.1 Data Source Model:

Only the business entities participating in the order to payment process have been included in the data source model.

We have decoupled the data sources into two subject areas, with major redundant entities “BusinessInteraction” and “BusinessInteractionItem”.

- **Customer Order and Business Interaction Data Model:**

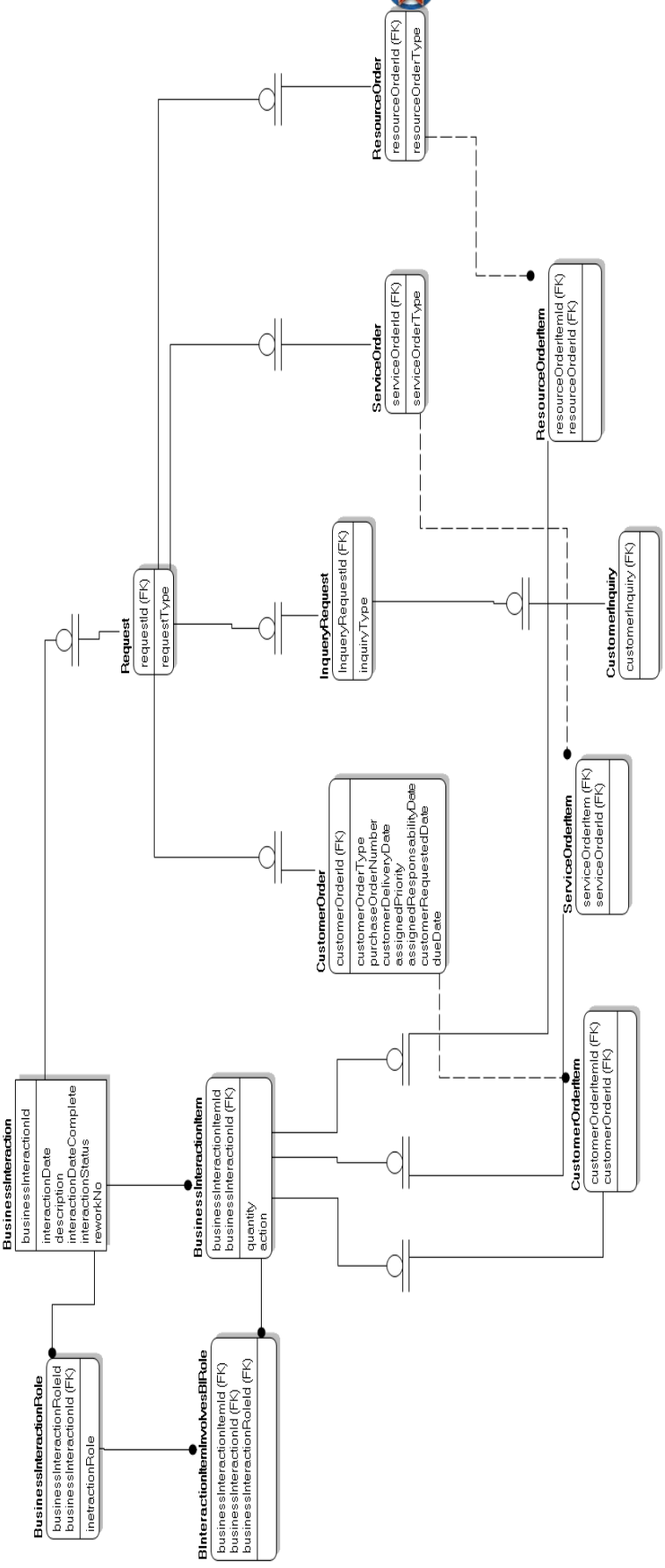


Figure 1: Customer Order and Business Interaction Data model

- **Party Data Model:**

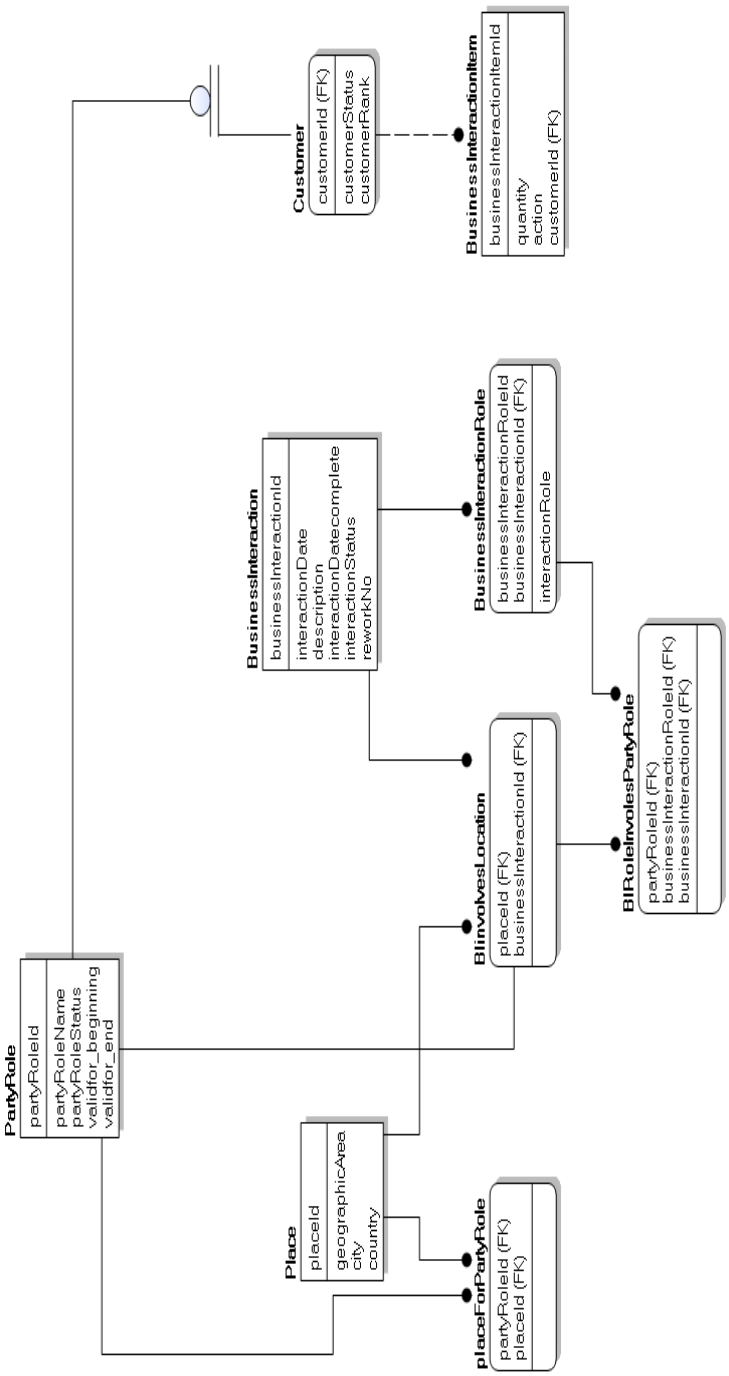


Figure 2: Party Data model

4.2 Business Requirements:

The Work Information Package (WIP) is constructed for three business metrics [3], [5] within the Customer Experience Data Mart:

F-CE-2a: Mean duration to fulfill Customer order;
F-CE-2b: Mean time difference between customers requested delivery date & Planned date;
These Business metrics are represented by one Fact table. The tables below contain the user requirements glossary, dimensions, and hierarchies regarding the fact measurements for each table as well as a preliminary workload.

a) Glossary Based-Requirements Analysis:

Table 1: Glossary Based-Requirements Analysis- Customer Order Data Mart

Fact	Dimensions	Measures	History
Order fulfillment Fact(F-CE-2a-2b-2c)	CustomerOrder; BusinessInteraction;Customer; Time; Place	Average time order fulfilment. Percentage of on time fulfilled orders. Average time order delay	2 year

b) Preliminary Workload:

Fact	Query
Order fulfillment Fact(F-CE-2a-2b-2c)	• What is the mean time to fulfill customer order by customer segment by time? • What is the mean time to fulfill customer order by customer segment by location? • What is the mean time difference between customers requested delivery date & planned date by customer segment by time? • What is the mean time difference between customers requested delivery date & planned date by customer segment by location? • What is the percentage of orders delivered on committed date by customer Segment by time? • What is the percentage of orders delivered on committed date by customer segment by location?

Table 2: Preliminary workload- Customer Order Data Mart

5. Experiment: Data Warehouse Modeling

This experiment does not explicitly compare the two direct implementations as their purpose is different. DV is an approach to a sustainable EDW design and represents a style of modeling for DW that can be characterized as dependencies minimization aiming at flexibility and performance [9]. It is designated to keep track of data and preserve its history but is "non-query-able" by end-users.

The objective of this experiment is to show how different modeling styles for a DW can impact its performance and flexibility.

5.1 Kimball Style DW

Dimensional modeling is an approach proposed by Kimball Ralph [7], oriented toward a specific business process and intended to optimize understanding and querying information. Most reporting tools require a dimensional model; It is intuitively understandable by business users.

At the center of the dimensional model are the numeric measures (Facts) that we are interested in measuring. Related measures are collected into fact tables that contain columns for each of the numeric measures. The most used dimensional model organization is the star schema where we find one or many fact tables in the center, joined to dimension tables.

To model the Order to Payment data mart, we need to specify the fact tables and the dimension tables. Using our requirement analysis [3], a star schema has been designed to analyze customer orders.

Data Mart model:

The following star schema fulfills all report requirements defined in the preliminary workload.

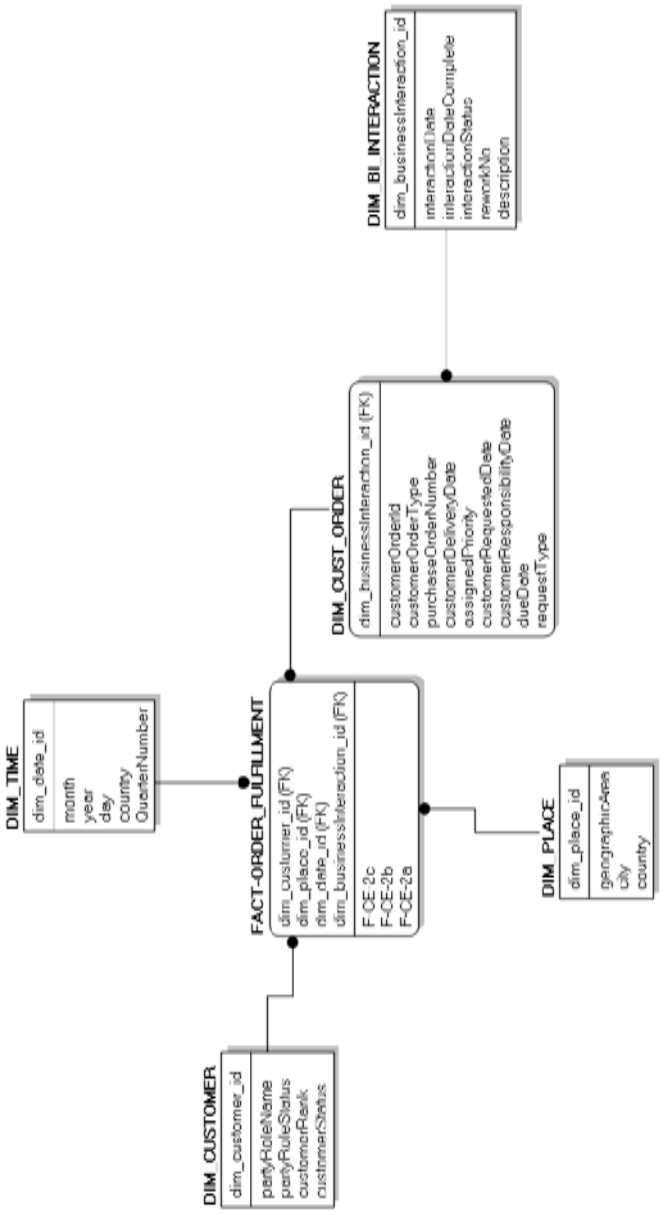


Figure 3: Star Schema for Customer Order Data Mart

The star schema has less information than the original source data model. It has only information related to the customer order, and that is required for reporting.
When designing a star schema it is a valid approach to simplify, reduce information and pre-calculate certain measures(e.g. counting time duration for a business interaction) to derive a model, that is easy to understand and answers the given analytical questions and which is optimized regarding performance. However, any additional analytical question cannot be answered if not considered during the phase of requirements analysis.

- The model does not contain any information about other business interaction type, such us service order or resource order. Analyzing order duration per business interaction type would require sourcing additional tables and a modification of the model.
- The model is not able to answer questions like "what is the mean time to fulfills customer orders by customer order item", since the star schema does not contain information about order items. The same is true for business interaction item.

Since user's requirement often change the need to add new analytical questions is frequent. The star schema designed for the customer order, even if fulfills the given user's reports, might turn out to be restrictive in the long term. For this purpose, the source system should be reconsidered as well as the ETL packages, when any additional requirements will have to be implemented.
The dimensional model has raised a lot of questions about its simplicity versus flexibility. The more complex and different system sources are the more issues and questions may rise and the more difficult it would become to anticipate them.

a) Load Performance (ETL):

The complexity of the ETL processes is determined by several factors, such as the number of source systems and their complexity, as well as the complexity of the integration and business transformation steps. Loading dimension tables is usually less complex than loading the fact tables, since less source tables have to be considered. Loading Fact tables is very often more complex since it usually requires transformation and integration of numerous source tables, as well as calculating measures.
The Extract, Transform, Load (ETL) process involves fetching data from transactional systems, cleaning the data, transforming it into appropriate formats and loading the result to the data mart. In the ETL process, data from data sources are extracted by extraction routines. Then, data are propagated to the data staging area before they are transformed and cleaned to be loaded into the data warehouse [8].

To experiment with load performance, we have implemented an ETL for our Order Fulfillment Data Mart. Loading dimension is pretty easy since one single source table is used for each dimension.
Figure 4 shows the different transformations realized in order to calculate and load, into the Fact table, the following measures: Customer Order Delay, Customer Order Duration and on time Customer Orders.

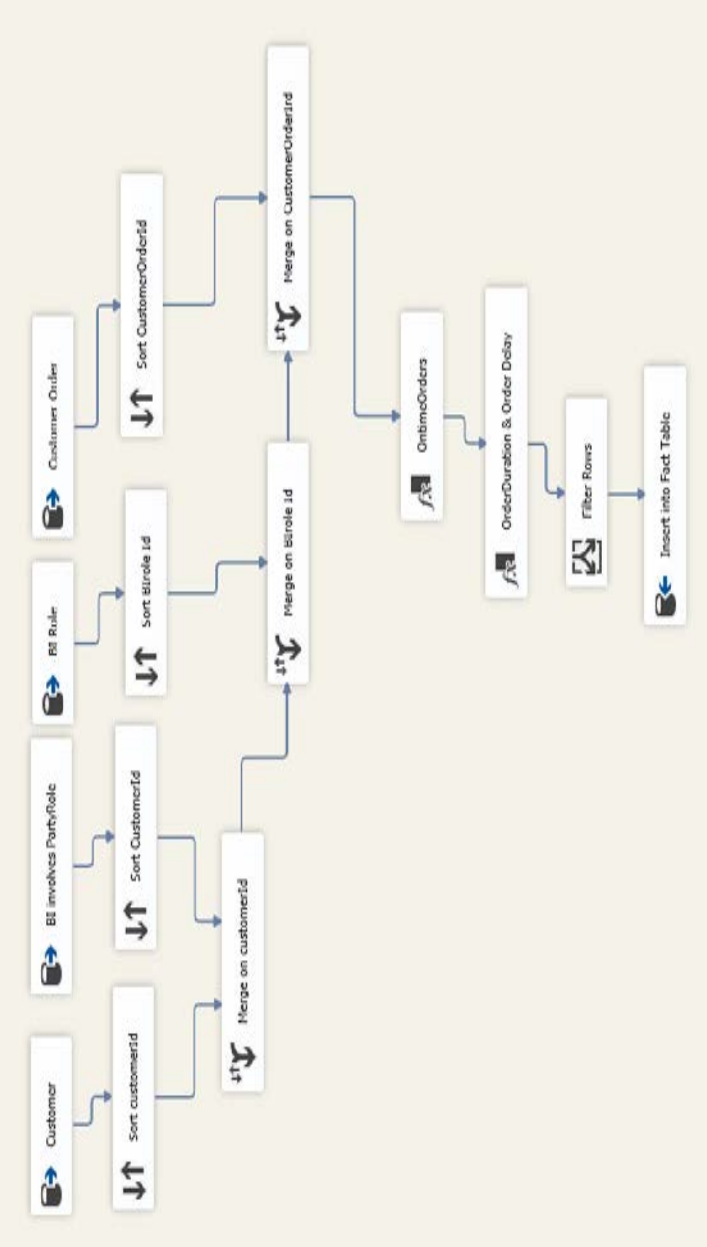


Figure 4: An Example of an ETL pattern to load Fact table

For the metrics F-CE-2a, F-CE-2b and F-CE-2c, customer orders will be analyzed by 'Customer Segment', this information exists in the customer table. To recreate the

link between Customer Order and Customer, it was necessary to go through the table 'BusinessInteractionRole' and the association table 'BusinessInteractionRoleInvolvesPartyRole'

This is why we used a 'Join on CustomerOrderId field', 'Join on BusinessInteractionRoleId' field, and a'Join on CustomerOrderId'. At the end of each join step, a selection step is added to filter the fields.

Concerning 'OntimeOrder calculation', it implied the difference between 'CustomerDeliveryDate' and 'DueDate'. 'OrderDuration' is the difference between 'InteractionDateComplete' and 'InteractionDate.' This calculation corresponds to F-CE-2a. It follows that for F-CE-2b, 'OrderDelay' is the difference between 'DueDate' and 'CustomerRequiredDate'. Finally, we insert the rows into the Fact target table.

Since a star schema is completely different than the typical OLTP data model, data model restructuring is extremely complex and often requires multiple steps and intermediate tables to cope with that complexity and to achieve the necessary load performance.

The following outcomes have been noticed when loading data to the Order Fulfillment Data Mart:

- A large number of complex transformations to consolidate data from numerous systems.
- Star schema is a non-normalized structure, so the data has some redundancy, creating several anomalies in the data which need advanced data cleansing transformations to reduce data duplication and dirty data.
- When the data integrity is low, and redundancy is high, loading time of dimension tables increases, and updating data becomes more complex.

b) Traceability:

The data warehouse team faces difficulties when an attribute change, they need to decide whether the data warehouse keeps track of both the old and new value of the attribute or not, what to use for the key? And where to put the two values of the changed attribute?

The data warehouse team usually decides to overwrite data when there is no need to track the old value of the changed dimension attribute. For example, if you find incorrect values in the service name or resource name attributes in a customer order, then overwriting would certainly be chosen. However, how to deal with this issue when the change is significant and a copy of the old attribute needs to be preserved.

In a dimensional database, the issues of describing the past mostly involve slowly changing dimensions (SCD) [11], which is a dimension that stores and manages both current and historical data over time. A typical slowly changing dimension is a service dimension in which the detailed description of a given service is occasionally adjusted. For example, a minor detail about the service change may be so small that production does not assign the service a new service Id (which the data warehouse has been using as the primary key in the service dimension), but nevertheless gives the data warehouse team a revised description of the service.

Although implementing slowly changing dimensions has solved the issue of tracking change of record over time, but this solution raises many issues related to performance and maintenance:

- When a large number of rows needs to be updated, this can result in substantial locking and logging, which severely affects the performance.
- A necessity of complicated ETL processes to implement that need frequent maintenance and configuration.

c) Auditability:

One other typical requirement for a data warehouse is the ability to provide information about source and extraction time of data stored in the system [1]. One reason for that is to trace down potential errors and try to understand the flow of the data into the system.

To support auditability in the dimensional model, we add meta-information to the data to track the data source and load time. However, it is more complicated to answer the question of where the data were used because data marts often aggregate data to create information that is used for analysis purposes.

d) Scalability and Flexibility:

Data warehouses must be designed to accommodate current and future business needs of the enterprise. It must be scalable enough to accommodate additional demands with a minimum of change to the fundamental design of the warehouse.

The problem of scalability and flexibility brings up some interesting data modeling issues in the Kimball methodology. For example, when an additional parent table is added, the change is forced to cascade down through the low level tables. Also, when a new row get inserted with an existing parent, all child rows must be reassigned to the new parent key. This cascading effect has an important impact on the processes and the data model which means that the larger the model is, the greater the impact. This makes it difficult (if not impossible) to extend and maintain an enterprise data warehouse model. The architecture and design are affected as a result because it was not built with those changes in mind.

We will illustrate a scenario where a business interaction is subject to change and needs to be revised. Since the underlying process has been changed, one solution is available by adding a table called

"BusinessInteractionVersion" that stores all the business Interaction versions and keep track of all the revisions that have been made to a business interaction. This table will contain the following attributes: "businessInteractionRevisionType", "businessInteractionRevisionNumber", "businessInteractionRevisionDate", and "businessInteractionRevisionDescription".

The "BusinessInteractionVersion" table stores weak entities precisely because a Business Interaction Version has no meaning independent of a Business Interaction.

"BusinessInteractionVersion" would be identified by a compound key consisting of both the Business Interaction id (foreign key) and the "BusinessInteractionVersion" id.

An impact of this process change is that business interaction is now related to a business interaction version. Now let speak about this impact on the order fulfillment data mart, suppose that with this new change in the process, users reporting requirements change as well and that they want to be able to analyze the number of revised orders by customer segment. This new requirement will bring up a lot of changes first to the data mart model and then to the ETL packages.

To conclude, the gap between the analytical capabilities of your model and the future analytical needs will become larger with increasing complexity of your business and your source systems. Although, many solutions have been proposed to encounter these problems in the dimensional model and maybe solved the scalability, the flexibility, and the traceability at a certain point but decreased the overall performance of the EDW.

5.2 Data Vault Style DW

DV was developed as an approach to a sustainable EDW design, and represents a style of modeling for DW that can be characterized as dependencies minimization aiming at flexibility and performance. The main advantages of DV over traditional 3NF EDW designs, in a style advocated by [9] are:

a) Inserts, deletes, or updates of rows are implemented only as additions (nothing ever get lost/overwritten) [9].

b) Structural changes of and in data sources results in model expansion, principally by new links and without structural reconstruction of existing DW elements (a holy grail of architectural stability) [9].

c) Enable rapid parallel data loads [9].

The DV modeling style is recognizable by its major concepts i.e. Hubs, Links, and Satellite entities. The Data Vault design is focused on the functional areas of business with the Hub representing business keys. The Link Entities representing transactions between business keys. The Satellite Entities providing the context of the business keys. Entities are designed to allow maximum flexibility and scalability while preserving most of the traditional skill sets of data modeling expertise [14].

DW Models:

5.2.1 Data Vault Model (backend)

a) Customer Order and Business Interaction data model

Hubs:

Since Hubs are a list of business keys, it is important to keep them together with surrogate keys. Upon evaluation of the model we find the following business key groupings:

- BusinessInteraction: BusinessInteractionId is the business key since, no surrogate key is explicitly provided, SK_B_interaction will be the surrogate key. This will constitute a HUB_B_INTERACTION.
- BusinessInteractionItem: BusinessIteractionItemId is the business key, since no surrogate key is explicitly provided, SK_B_InteractionItem will be the surrogate key. This will constitute a HUB_B_INTERACTION_ITEM
- BusinessInteractionRole: BusinessIteractionRoleId is the business key, since no surrogate key is explicitly provided, SK_B_InteractionRole will be the surrogate key. This will constitute a HUB_B_INTERACTION_ROLE
- CustomerOrderItem: CustomerOrderItemID is the business key, since no surrogate key is explicitly provided, SK_CustomerOrderItem will be the surrogate key. This will constitute a HUB_CUSTOMER_ORDER_ITEM
- ServiceOrderItem: ServiceOrderItemID is the business key, since no surrogate key is explicitly provided, SK_ServiceOrderItem will be the surrogate key. This will constitute a HUB_SERVICE_ORDER_ITEM
- ResourceOrderItem: ResourceOrderItemID is the business key, since no surrogate key is explicitly provided, SK_ResourceOrderItem will be the surrogate key. This will constitute a HUB_RESOURCE_ORDER_ITEM

When the source is already highly integrated using abstracted generalized entities most of the subtypes can be ignored using satellites per subtypes (all attached to the root where the identity is)

Below are the entities that have been omitted from the Data Vault design:

- Request, InquiryRequest, CustomerInquiry, CustomerOrder, ResourceOrder, ServiceOrder

Figure 5 below shows an example of this strategy

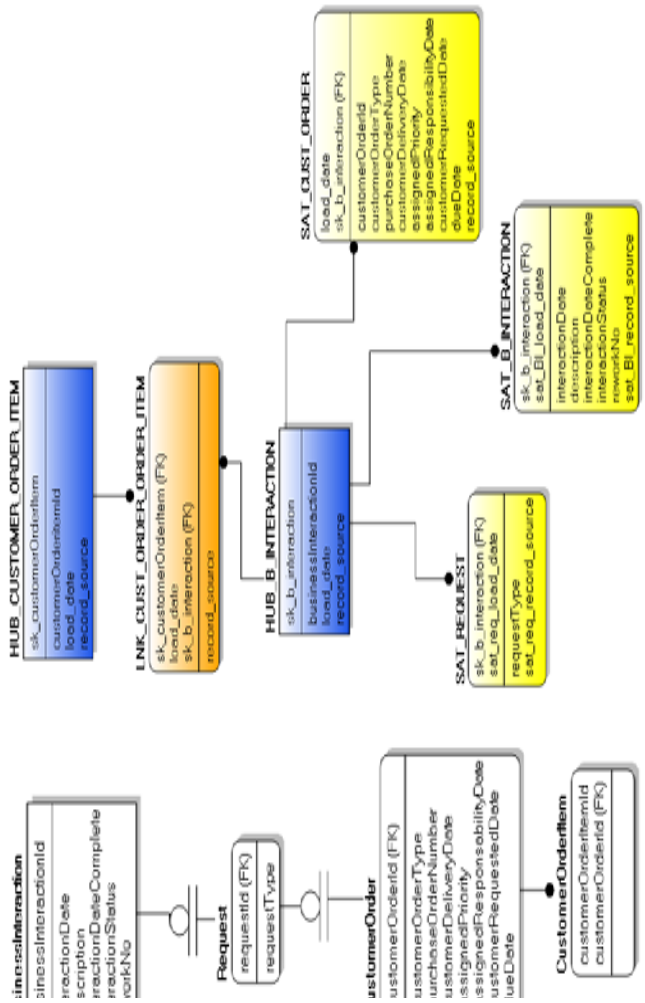


Figure 5: An Example of mapping abstracted generalized entities to Data Vault model

Links:

The Links represent the business processes; they are considered as the glue that connects the business keys together. They describe interactions and relationships between the keys.

The link tables of this model are as follows:

- BInteractionItemInvolvesBIRole: Many to Many, excellent Link Table. LNK_B_INT_INVOLES_BI_ROLE will be constituted.
- BusinessInteraction table is parent table of BusinessInteractionRole and BusinessInteractionItem, this will constitute two link Tables LNK_BITEM_BI_ROLE and LNK_BI_BI_ITEM, including the surrogate key from BusinessInteraction.
- CustomerOrder is a parent table of CustomerOrderItem. Thus, LNK_CUST_ORDER_ORDER_ITEM will be constituted.
- ServiceOrder is a parent table of ServiceOrderItem. Thus, LNK_SERV_ORDER_ORDER_ITEM will be constituted
- ResourceOrder is a parent table of ResourceOrderItem. Thus, LNK_RES_ORDER_ORDER_ITEM will be constituted

Also, we have used same-as links between the master entity BusinessInteractionItem and its subtypes (ServiceOrderItem, ResourceOrderItem, and CustomerOrderItem) those links represent hierarchical (parent-child) between the master entities.

SAL_BI_ITEM_SERV_ITEM;
SAL_BI_ITEM_CUST_ITEM;
SAL_BI_ITEM_RES_ITEM.

Satellites:

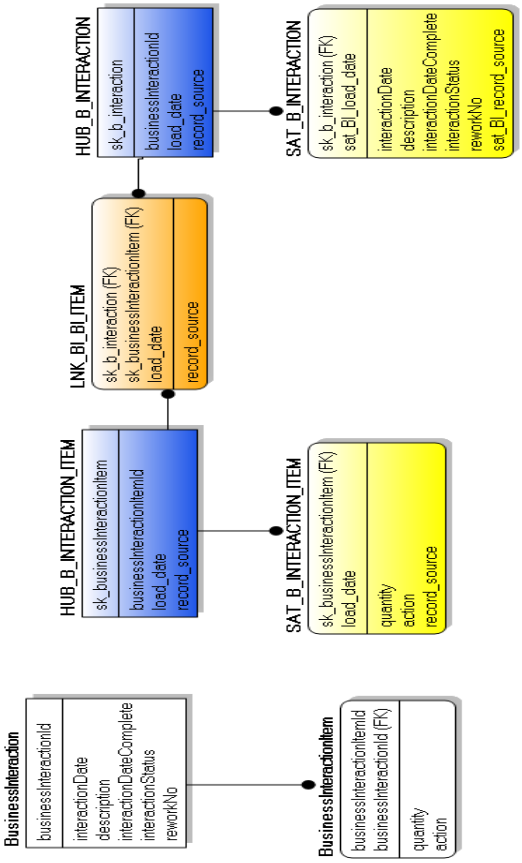
The rest of the fields can change over time. Hence, they will be put into Satellites.

The tables below will be created as Satellite structures:
SAT_B_INTERACTION,
SAT_B_INTERACTION_ROLE,
SAT_B_INTERACTION_ITEM, SAT_CUST_ORDER,
SAT_INQUERY_REQUEST.

Satellites include only non-foreign key attributes. Satellite primary key is the primary key of the Hub with a LOAD_DATE incorporated.

The same analysis has been done for the remaining data sources. Figure 6 shows a portion of the diagram.

Figure 6: An example of mapping a relational to the data vault physical model



Customer Order and Business Interaction Data Vault Model:

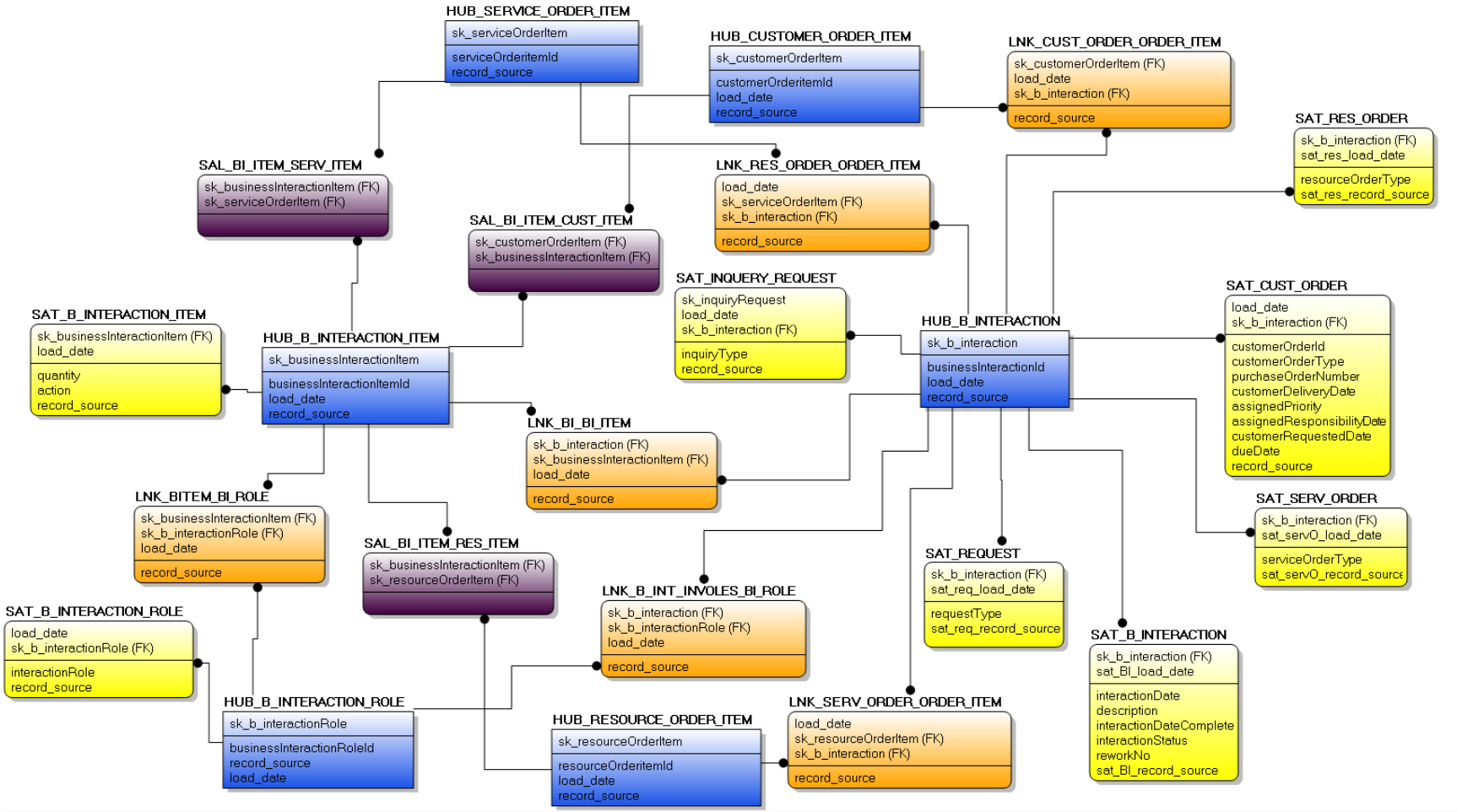


Figure 7: Data Vault Model for Business Interaction Data source

Figure 7 shows a working version of the order and business interaction data source.
Hubs (shown in blue) have a corresponding satellite attached to them (shown in yellow).

b) Party Role Data Vault Model:

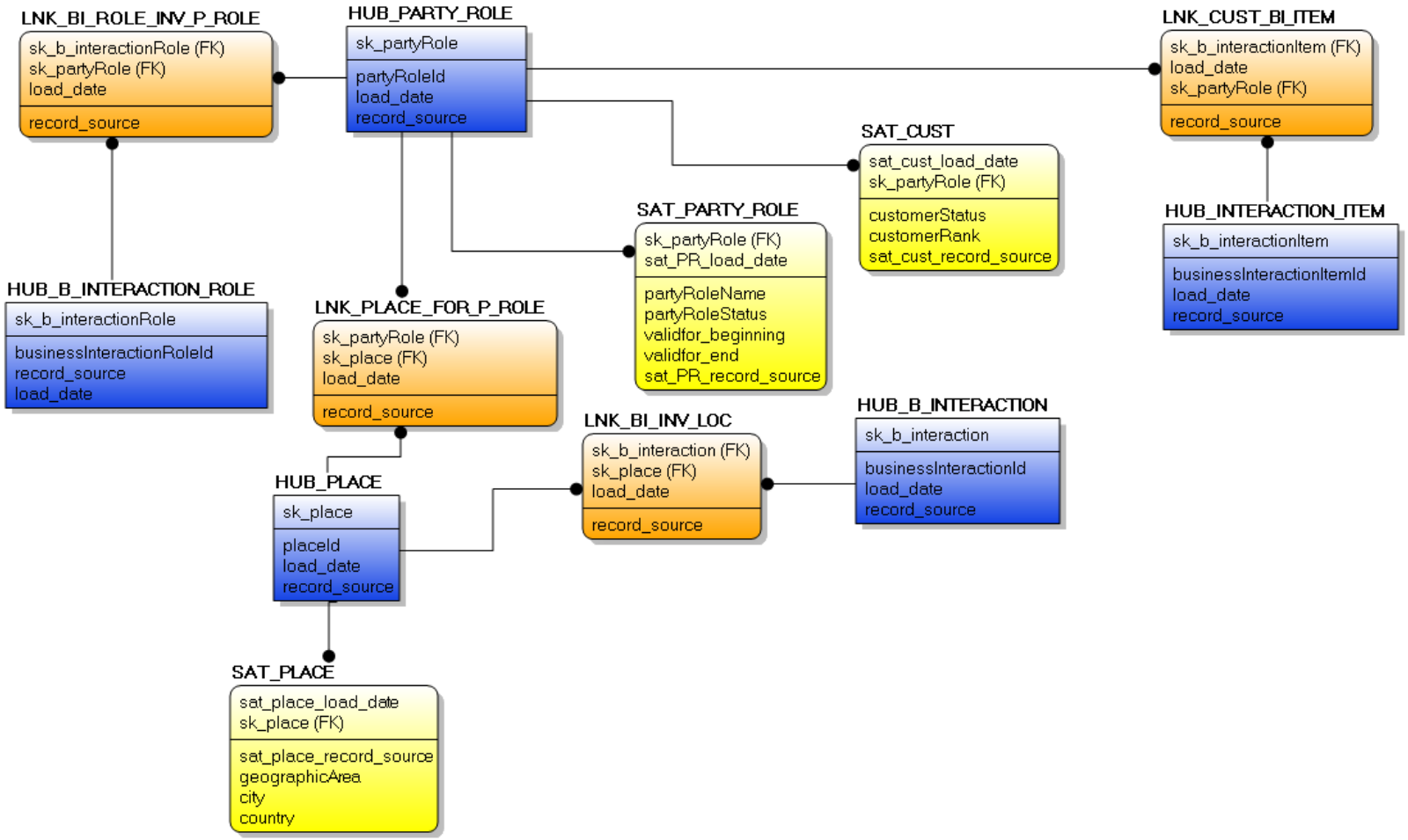


Figure 8: Data Vault Model for Party Role Data Source

The Data Vault model keeps intact the source system context. Data coming from different sources are integrated into a Data Vault type warehouse without undergoing transformations. Data are then rapidly loaded in their raw format, including the date and the modification source. It is, therefore, possible to rebuild a source's image at any moment in time.

The Data Vault approach solves many of the problems faced with the Kimball modeling DW style:

- It is flexible and is modification resistant.
- It is extendable.
- Modifications in the sources are rapidly shown in the warehouse.
- It easily allows reconstituting data source image at any moment in time.

However, the data vault model is not as an end-user data mart accessible model because requires many joins in a query which will have a direct impact on query performance, star schemas remains the best for delivering data directly to end-users. Data Vault is the best for the Data Warehouse environment. For this reason, a Data Vault style DW needs an additional layer to deliver analysis reports to the users. We will discuss how to implement this additional layer later in this paper.

5.2.2 Data Delivery for a DV based DW (frontend)

Dimensions are much easier to understand and query that's why we don't query data vault directly. Data Vault is a system of record. However, data mart are for analysis and reporting. The objective of this section is to build a star schema to be able to perform analysis and reporting on data. The star schema will be directly derived from DV model, all the dimensions used are type 1 dimension which means, show the most current values of all the attributes. Using the results of the requirement analysis, the data mart has four dimensions which are: BusinessInteraction, CustomerOrder, Customer, Time, and Place.

a) Derive Dimension from DV:

All dimensions are built by joining a Hub and a Satellite.

We will show as an example how to derive a CustomerOrder type 1 dimension from a DV model.

The hub HUB_B_INTERACTION surrogate key can be used as primary key for the dimension, all the other columns will come from the contributing Hub and Satellite which are in this case (SAT_CUST_ORDER, SAT_REQUEST, HUB_B_INTERACTION), we will need to get the MAX (load date) from the Satellite tables in order to have the recent view of data. To optimize performance and simplify the query, a materialized view have been used to perform all the different joins between Hubs and Satellites.

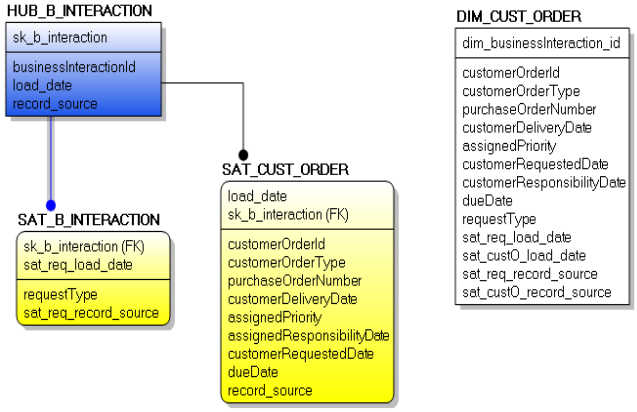


Figure 9: An example of transforming raw Data Vault to Dimension in a Data Mart

b) Derive Fact from DV:

The following links and satellites will be used to derive our fact table LNK_B_INT_INVOLES_BI_ROLE; LNK_BI_ROLE_INV_P_ROLE; LNK_BI_INV_LOC; SAT_CUST_ORDER; SAT_B_INTERACTION.

Measures:

F-CE-2c ('OntimeOrders') is the difference between 'CustomerDeliveryDate' and 'DueDate.'

F-CE-2a ('OrderDuration') is the difference between 'InteractionDateComplete' and 'InteractionDate'.

F-CE-2b ('OrderDelay') is the difference between 'DueDate' and 'CustomerRequiredDate'.

First, we will create a materialized view to get all the surrogate keys from the dimensions using links; then we will create the SQL code to project our FACT table based on the materialized view created.

Figure 10 shows the side-by-side model of the Data Vault tables and the deriving Fact.

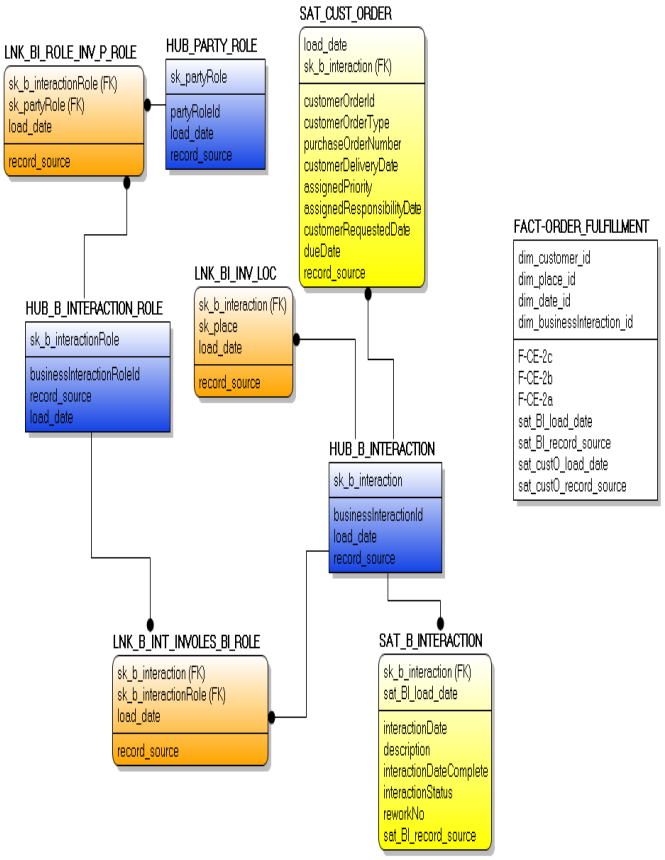


Figure 10: An example of transforming raw Data Vault to Fact in a Data Mart

Data vault base W still needs Kimball approach to present information to the end users. The star schema is built on the top of the raw data vault by simply joining some DV tables together. As such, the goal is to turn data into information that is useful for making business decisions.

5.2.3 DV system Performance:

a) Load Performance (ETL):

Data integration of multiple source systems into one DWH is always hard work. For Data Vault, the strategy will be to find the core business concepts and the natural business keys in the source systems and then verify that they conform to the business point of view. The same approach is used for the relations.

The flexibility of the Data Vault model is because it allows adding objects without a complete redesign of the existing structure and the possibility to have different Satellites for the source systems. This strategy allows tracking of the differences and quality job into the core DWH but postpones the sourcing choice of the attributes to the Data Mart layer. For example, if a new source system, also containing customer entities, has to be loaded into the Data Warehouse, a new Satellite with different attributes can be added to the model. A simple report comparing the two Satellites will show the quality issues to the business.

The ETL jobs to load a Data Vault typically run in two steps: In a first step, all Hubs are loaded in parallel. In a second step, all Links and Satellites are loaded in parallel. The individual ETL operations for each type of objects are:

- Hubs: The business key must appear only once in the Hub; insert with a lookup on the business key
- Links: The association must appear only one time in the Link; insert with a lookup on the association of surrogate key
- Satellites: The loading of the object is the most complex, it depends on the historization mode. Either the changed attributes are just updated, or a new version must be inserted.

Every object (Hub, Link, and Satellite) in the Data Vault methodology stores two auditing attributes:

The first load date (i.e. the first date when the specific line appeared) and the first load source (i.e. the first source where the specific line appeared). They are providing audit information at a very detailed level.

Figure 11 illustrates the load pattern used to load Hubs in our Data Vault model

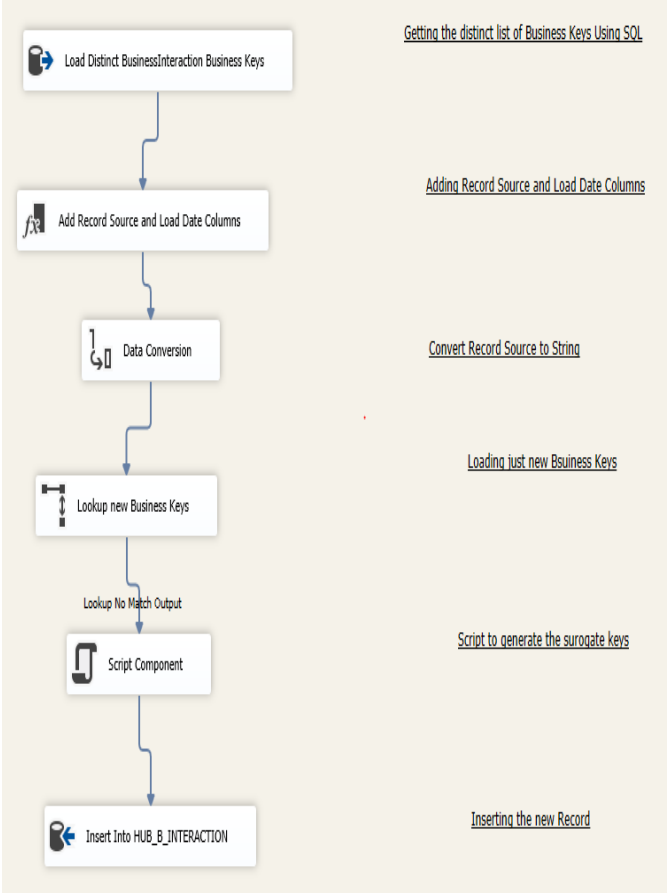


Figure 11: An example of an ETL pattern to load Hub

In a data vault model, we use business keys to integrating data. Since the hub table contains a distinct list of business keys for an entity, it makes the hub the master object of a data vault DW. To load data to HUB_BI_INTERACTION, we first select the distinct list of business keys from BuisnessInteraction data source table; we set the metadata values for load date and record source per row, we perform then a lookup function that looks for only new business keys, then generate the hash key using a.Net MD5CryptoServiceProvider class. Finally, rows are inserted into the target table HUB_BI_INTERACTION.
One of the big advantages of DV 2.0 was the replacement of the standard integer surrogate keys with hash-based primary keys. Hashing allows a child "key" to be computed in parallel to the parent "key" and be loaded independently of each other, which highly increases load performance of an Enterprise Data Warehouse especially when working with large volumes of data.
To conclude, Data Vault dependencies minimizations and rapid loads opportunities enable greatly simplified ETL transformations in a way not possible with traditional data warehouse designs.

b) Traceability:

Data Vault represents a modeling architecture for the enterprise data warehouse; it is considered as a system of record for the enterprise. It ensures the preservation of history and provenance which enable tracking the lineage of data back to the source. To ensure data traceability, Data Vault methodology does not transform data coming from different sources before they are inserted into the warehouse, thus enabling permanent system of records:

- Historical changes are captured by inserting new links and satellites.
- Change tracking is fairly complex due to a highly normalized structure.
- Provides the most detailed and auditable capture of changes.

c) Auditability:

Since Data Vault keeps a comprehensive history including the ability to record where data came from, it makes the perfect choice for any system where keeping an audit trail is important. Each row in a Data Vault is accompanied by record source and load date information.
Data warehouse developers are constantly in needs to trace down any potential errors in order to improve the system performance, Data Vault help them answer all the questions related to auditability at any time they can know where is a particular data asset extracted, when has been extracted, what was the process that extracted it and where was it used.

d) Scalability and Flexibility:

One of the biggest advantages of Data Vault is its scalability and adaptability to business change through the separation of business keys and the associations between them from their descriptive attributes. Data are organized around these business keys. The Hubs (business keys), Links (associations), and SAT (attributes) allow a highly adaptable data structure while enabling a high degree of data integrity. Dan Linstedt often compares Data Vault structure to a simplistic view of the brain where neurons are associated with Hubs and Hubs Satellites and dendrites are Links (i.e., vectors of information), other Links are like synapses (i.e., vectors in the opposite direction). They can be created or dropped on the fly as business relationships change automatically morphing the data model as needed without impacting the existing data structures.
Furthermore, Data Vault architecture supports parallelism while loading multiple data sources (hubs loaded first, then links, then satellites) which allows ETL scalability.
Speaking flexibility Data Vault methodology combines SEI/CMMI Level 5 best practices with best practices from Six Sigma, TQM, and agile methodologies. Data Vault projects have short controlled release sprints that can result in a production release every 2 or 3 weeks adopting the consistent, repeatable, and measurable projects expected at CMMI Level 5. When a new change in data sources need to be added, new Hubs, Satellites or Links can be added and then linked to the existing Data Vault structures without requiring alterations of the existing data model elements.

6. Results and Summary

Kimball approach is typically least complex, fastest and easiest to implement, provides the best combination of loading and querying performance. A data warehouse style Kimball contains consistent, quality cleansed and business aligned data. However, lack Scalability, auditability and flexibility.
Data Vault is an innovative concept, and it has merits when compliance demands are very high, and auditing and traceability requirements frequently change.
The EDW model is built to be a back-end powerhouse for enterprises. It is not built for end-user reporting which is why we still need a vital part of "data warehousing", the star schema, to deliver data to the business user.
Table 3 summarizes the characteristic of each implementation, and its impact on the overall performance of an EDW.

Enterprise Data Warehouse (EDW) characteristics	Kimball Style DW	Data Vault based DW
Load	Complex: It	DW style requires

Performance (ETL)	requires single ETL process containing multiple steps and transformations that load final data model used for reporting.	2 ETL processes: 1) loading from source systems, scalable and less complex (backend) 2) building reporting data marts(frontend) Separation of backend and frontend, allow high performance on the EDW
Traceability	Uses concept of slowly changing dimensions (SCD) to track historic changes. Requires business to identify attributes, requiring tracking before load. Necessity of complicated ETL processes to implement SCD that needs frequent maintenance and configuration. Updates of large data decrease system performance.	Historical changes are captured by inserting new links and satellites. Provides the most detailed and auditable capture of changes. Change tracking is fairly complex due to the highly normalized structure. System performance is high.
Auditability	Can add meta-information to the data to track the data source and load time. Cannot know at any time where the data was used because of aggregated data in data marts.	Answers all the questions related to auditability at any time. Knows where a particular data asset is extracted from when it has been extracted, what was the process that extracted it, and where was used.
Scalability and Flexibility	New requirements bring up a lot of changes to the data mart model and the ETL packages. Cost time and resource.	Scalability and adaptability to business change through the separation of business keys and the associations between them from their descriptive attributes. Highly scalable.

Table 3: Effect on DW performance on the Kimball Style DW vs. Data Vault based DW

7. Conclusion

The key contribution of this paper was an experimental comparison of DW performance of traditional Kimball's data warehouse design with a design using the Data Vault approach. Main findings can be summarized as follow:
a) Dimensional modeling is still the best practice for analysis and reporting and as a visual star schema model best understandable by business users. However, it does not meet the performance expectation for an EDW system.
b) Data Vault is more suitable for large Enterprise Data Warehouse, but not suitable for direct analysis and reporting. For that, we still need dimensional modeling for creating our "virtual" data marts to respond to business users requirements.
c) The problem of schema evolution after the changes in the data sources or user requirements is present in both of these approaches.
d) Data Vault is easier, more flexible, to add new sources, more auditable and keeps all the data all the time so you will be able to always recreate your DM's.
e) The conclusion is to use Data Vault for Enterprise Data Warehouse and Dimensional Modeling for derived Data Marts.
Furthermore, a novel systematic technique emerged for designing data vault models (from highly generalized source data models i.e. models with elaborate subtypes) using satellites per subtype as illustrated in Figure 5. Fully understanding long-term limitations/advantages of this technique is now the immediate focus of our applied research.

References:

[1] Linstedt, D., Olschimke, M. Building a Scalable Data Warehouse with Data Vault 2.0, 1st Edition, 2015.
[2] https://www.tmforum.org/about-tm-forum/
[3] Benhima, M., Reilly, J., Naamane, Z., Kharbat, M., Kabbaj, M., & Esqalli,O. "Design and Implementation of the customer Experience Data Mart in the telecommunication Industry: Application Order-To-Payment end to end process,". International Journal of Computer Science Issues, Vol. 10, Issue 3, No 1, May 2013.
[4] https://www.tmforum.org/resources/standard/gb921e-end-to-end-business-flows-r15-0-1/.GB921E End-to-End Business Flows R15.0.1
[5] https://www.tmforum.org/resources/standard/gb935-b-business-metrics-development-guidegb935

b_business_metric_development_guide-v7-1-1/. GB935-B Business Metrics Development Guide V7.1.1.
[6] http://inform.tmforum.org/wp-content/uploads/2014/05/Implementing-the-SID-v1dot0b-Chapters-1-through-3.pdf.
[7] Kimball, R., Ross, M. The Data Warehouse Toolkit, Second Edition, Wiley & Sons, Inc., 2002
[8] Krneta, D., Jovanovic, V., Marjanovic, Z. "A Direct Approach to Physical Data Vault Design," Computer Science and Information Systems, 2014.
[9] Jovanovic, V., Subotic, D., Mrdalj, S. "Data Modeling Styles in Data Warehousing," Information and Communication Technology, Electronics and Microelectronics, 2014.
[10] Jovanovic, V., Subotic, D., Poscic, P. "Data Warehouse and Master Data Management Evolution – A Meta-Data-Vault Approach," Issues in Information Systems, Vol. 15, Issue II, pp. 14-23, 2014.
[11] http://www.kimballgroup.com/2008/09/slowly-changing-dimensions-part-2/comparison DWH core modeling.
[12] Linstedt, D., Grazianno, K., Hultgreen, H. The business of data vault modeling, 2nd edition, 2009.
[13] http://tdan.com/data-vault-series-1-data-vault-overview/5054
[14] Jovanovic, V., Bojicic, I. "Conceptual Data Vault Model," Proceedings of the Southern Association for Information Systems Conference, 2012.
[15] Kimball, R. "The evolving role of the enterprise data warehouse in the era of big data analytics. Kimball," Group White Paper, 2011.
[16] Golfarelli M. "Data warehouse life-cycle and design," White Paper, DEIS – University of Bologna, 2008.
[17] Jovanovic, V., Bojicic, I., Knowles, C., & Pavlic, M. "Persistent Staging Area Models for Data warehouses," Issues in Information Systems, Vol. 13, Issue 1, pp. 121-132, 2012.
[18] Hogan, M., Jovanovic, V. "ETL Workflow Generation for Offloading Dormant Data from the Data Warehouse to Hadoop," Issues in Information Systems, Vol.16, Issue I, pp. 91-101, 2015.
[19] Krneta, D., Jovanovic, V., & Marjanovic, Z. "An Approach to Data Mart Design from a Data Vault," INFOTEH-Jahorina BiH, Vol.15, March 2016.
[20] H. W. Inmon. Building the Data Warehouse. Wiley Computer Publishing, 1992
[21] Golfarelli, M., Rizzi, S. "A Survey on Temporal Data Warehousing," International Journal of Data Warehousing and Mining (IJDWM), Vol. 5, n. 1, pp. 1-17, 2009.
[22] Jovanovic, V., Benson, S. "Aggregated Data Modeling Style," Proceedings of the Southern Association for Information Systems Conference, Savannah, GA, USA March 8th–9th, 2013.
[23] Naamane, Z. "Design and Implementation of a Customer Experience Data Mart in Telecommunication Domain," term project report for MS CS Data Warehousing course, Georgia Southern University, May 2016.