DATA VAULT MODELING GUIDE

Introductory Guide to Data Vault Modeling

GENESEE ACADEMY
World-Class Training

GENESEE ACADEMY, LLC

2019

Authored by: Hans Hultgren

DATA VAULT MODELING GUIDE

Introductory Guide to Data Vault Modeling

Forward

Data Vault modeling is most compelling when applied to enterprise integration initiatives, such as a data warehouse program (EDW). Data Vault, as a form of Ensemble Modeling, is optimized for programs that are based on an enterprise business view, including all organizational data, integrated from multiple divisions, departments and functions. In short, the organization contemplating this type of initiative is committing to an integrated, non-volatile, time variant and core business concept driven data warehouse program.

Ensemble Modeling principles are specifically well suited for such programs and – when applied consistently – can provide the organization with some very compelling benefits. These include agility, auditability, adaptability, alignment with the business, and support for both strategic and operational data warehousing initiatives.

To gain these benefits, the organization will need to commit to both EDW program factors and specific data vault modeling patterns. This guide presents data vault modeling in the context of the EDW.

Index

The EDW Program

The Enterprise Data Warehousing (EDW) Program represents the ongoing data warehousing activities of the organization. These activities will include the maintenance functions of the data warehouse in addition to the continuous flow of incremental projects related to the enterprise data warehouse. These incremental projects are comprised of

a) Adapting to new data sources from new internal systems, external integrations, and new sources related to acquisitions/mergers, and
b) Absorbing changes to existing sources including new tables, new attributes, new domain values, new formats and new rules, and
c) Adapting to new business rules concerning the alignment, grain, cardinality and domain values as well as changes to the relationships between them, and
d) Accommodating new downstream delivery requirements including new subject areas, new business rules, additional regulatory and other compliance reporting initiatives and changes to operational latency requirements.

For this reason, the EDW itself is not designated as a "project" (there is no discernable beginning and end, and no pre-determined set of specific goals) but rather as a "program" driving an "integration machine" based on continually adapting to change.

In a broader sense, this program can be defined as the BI Function or BI Program within an organization. To be clear, this is not simply the group that owns the BI tools. This is the higher level view of all data warehousing and business intelligence (DWBI) within the organization which includes the business intelligence competency center or BICC, the EDW or CDW team, the related governance components and the environment both technical and organizational. The success of a DWBI program depends on an organizational commitment and a corporate BI culture.

It is precisely in this context where the data vault approach is the most valuable. So the Data Vault EDW is defined first and foremost by the enterprise wide, long term DWBI program – not from a technical architecture perspective but from an organizational cultural alignment perspective.

Data Modeling for the EDW Program

The requirements of the EDW Program imply also requirements for the data model of the data warehouse and the data modeling approach that is used. We need to apply a data modeling approach that can accommodate the goals of the EDW program from a modeling perspective. So for example the EDW program objective of adapting to the continuous flow of incremental projects means that the data model must be able to adapt to these changes. At the same time, the EDW needs to effectively integrate data, it needs to track history and it needs to remain auditable.

This is a tall order for modelers and architects in this space. And a tall order for the database that needs to accommodate these requirements. Over the past decade we have discovered some interesting design pattern elements that help us to address these needs. One example is Unified Decomposition – the idea that we separate the things that are changing from the things that don't change. This sounds simple (and really it is) but we just have not been thinking in these ways before. We typically look to *encapsulate* everything we know about a concept, entity or dimension into a single table construct. *See Diagram*.

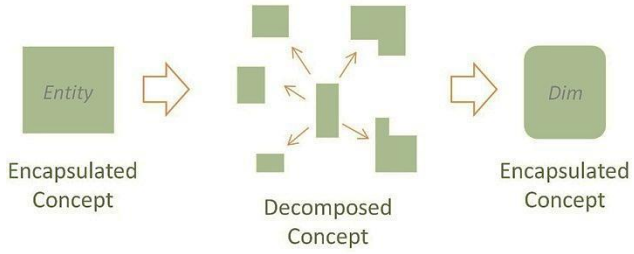


Fig. 1 ***Decomposition***

With Unified Decomposition we move away from encapsulating and *decompose* (deconstruct) the concept into component parts. Once we embrace these paradigms we start to see our data warehouse behave with improved agility.

To avoid random decomposition and table explosion we next look to *Unify* the resulting parts into a form we call an Ensemble. This **Ensemble** can be categorized in this way:

> **All the parts of a thing taken together, so that each part is considered only in relation to the whole.**

Data Vault Ensemble: parts are unified using a hub (single instance of the concept).

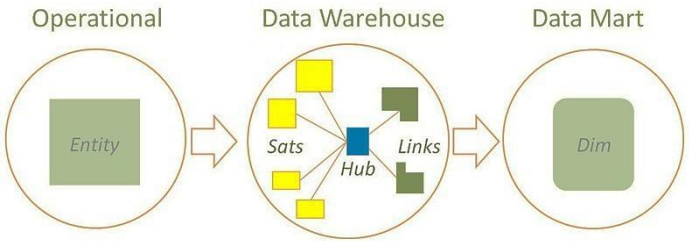


Fig. 2 ***Data Vault Ensemble***

The Data Vault Fundamentals

The data vault ensemble consists of three core components, the **Hub, Link** and **Satellite**. Above all other DV Program rules and factors, the commitment to the consistency and integrity of these constructs is paramount to a successful DV Program.

The Hub represents the heart of the Core Business Concept (CBC) such as Customer, Vendor, Sale or Product. The Hub table is formed around the Enterprise Unique Key (Ensemble Identifier) of this concept and is established the first time a new instance of that identifier is introduced to the EDW. Note the Ensemble Identifier is the EDW equivalent of the Operational DB Business Key. Because the EDW deals with multi-source integration, it most commonly requires a multiple part key to assure an enterprise wide unique key however the cardinality of the Hub must be 1:1 with a single instance of the core business concept (CBC). The Hub contains no descriptive information and contains no FKs. The Hub consists of the key only, with a warehouse generated sequence id or hash key, load date/time stamp and a record source.

Fig. 3 ***Hub***

A Link represents a unique, specific, natural business relationship between business keys and is established the first time this new unique association is presented to the EDW. It can represent an association between several Hubs. It does maintain a 1:1 relationship with the unique and specific business defined association between that set of keys. Just like the Hub, it contains no descriptive information. The Link consists of the sequence ids from the Hubs that it is relating only, with a warehouse generated sequence id, a load date/time stamp and a record source.

Fig. 4 ***Link***

The **Satellite** contains the descriptive information (context) and all history for a CBC. There can be several Satellites used to describe a single key however a Satellite can only describe one key (Hub). There is a good amount of flexibility afforded the modelers in how they design and build Satellites. Common approaches include using the subject area, rate of change, source system, or type of data to split out context and design the Satellites. The Satellite is keyed by the sequence id from the Hub to which it is attached plus the load date/time stamp to form a two-part key.

Note that the Satellite then is the only construct that manages time slice data (data warehouse historical tracking of values over time). It is considered a best practice to also include business dates (effective dates) since the load date is driven by the data warehouse.

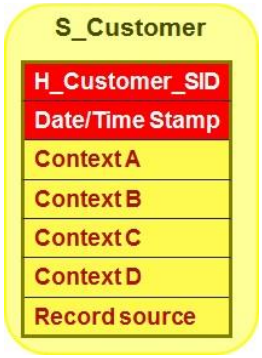


Fig. 5 ***Satellite***

Satellite does not have a Sequence ID of its own and in fact cannot have a different key than the Hub sequence (or hash) to which it is attached. Further, a Satellite does not have any foreign key constraints (no snow-flaking, branching or bridging).

These three constructs together form an Ensemble which includes the complete picture of a given CBC. In turn, these Ensembles are the building blocks for the DV EDW. Together they can be used to represent all integrated data from the organization. Ensembles represent Events, Persons, Organizations, Things, Places and other concepts. For each Ensemble, the Hubs represent the instances (unique identifier/key), the Links represent unique, specific relationships and the Satellites provide all the context and changes over time (history).

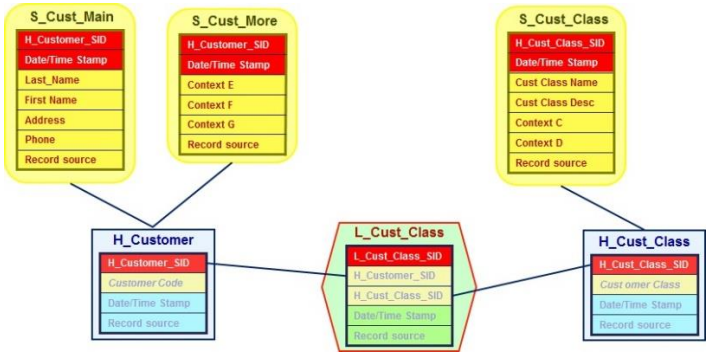


Fig. 6 ***Data Vault Model***

When we look at the Hub and Link together, they form the ***backbone*** or "Skeletal Structure" of the model. This backbone model represents a 1:1 relationship with core Business Concepts and their natural business relationships.

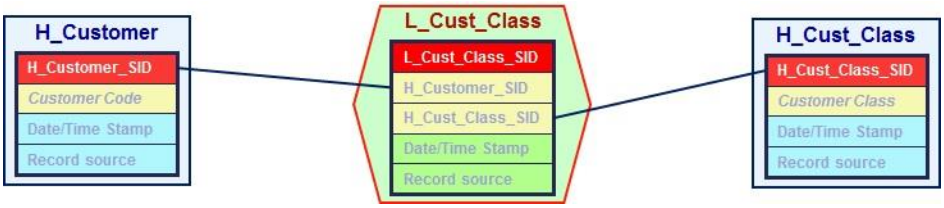


Fig.7 ***Backbone or Skeletal Structure***

Note that ***all context*** (descriptive information) and ***all history*** are found in the Satellites.

Thinking Differently

Modeling with Data Vault requires us to *think differently*. Most of us first learned 3NF modeling for operational databases. To manage third normal form, all attributes in an Entity must depend directly on the key of that Entity. So the context attributes that describe a customer (last name, first name, address, city, state, postal code, home_phone, mobile_phone, etc.) must be placed in the Customer Entity where the key uniquely identifies an instance of a customer. If we included attributes that do not depend on the key of that entity then we would not be in 3rd normal form. Likewise if we placed some of the attributes that depend on that key into another entity then again we would no longer be in 3rd normal form.

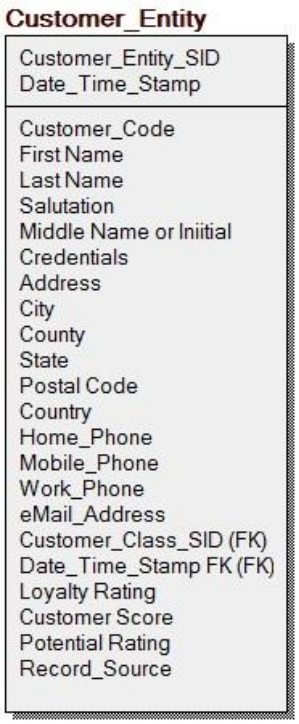


Fig. 8 ***3NF Customer***

At some point we may have also learned how to model using dimensional modeling techniques. Though different modeling constructs and other rules for modeling, the concept of including context attributes inside a table with a key for those attributes remains the same. A Conformed Dimension requires that context attributes depend on the key of that Dimension. Again if we move out attributes depending on a dimension key to some other construct then we no longer have a conformed dimension.

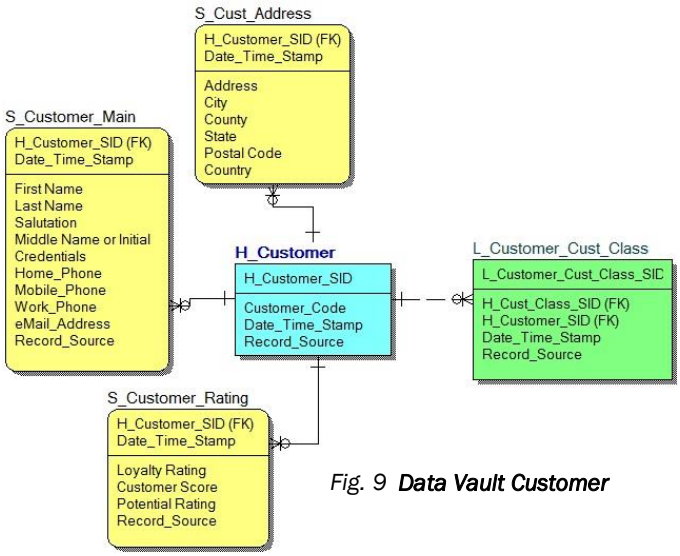


Fig. 9 ***Data Vault Customer***

Shown on the prior page is a Customer_Entity in 3NF where we can see the Business Key (Customer_Code), the relationship (Customer_Class_SID) and all context in the form of all remaining attributes in the table. Notice that this is one table including all of this these components.

With Data Vault modeling we separate the "business keys" (Ensemble Identifiers) from the relationships from the context. All of the instances/keys are modeled as Hubs, all relationships and associations are modeled as Links, and all context and history is provided for through the Satellites. Shown here we can see that the Identifier/Key (Customer_Code) is in a Hub (H_Customer), the relationship (Customer_Class_ID) is in a Link (L_Customer_Cust_Class), and the context is modeled in several Satellites.

Look back to the 3NF model and now consider that all of the same information (the same components of data) about "Customer" are represented fully in both models. Interestingly both models represent a ***dependency*** on a single instance/key. Actually, if we draw circles around each of these models we can see that what is inside each circle is a representation of the same single key, the same set of attributes and the same relationship.

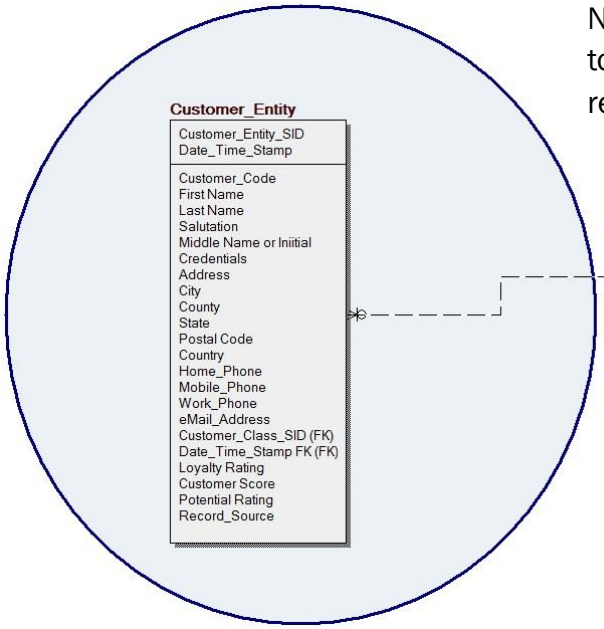


Fig. 10 ***3NF Model***

Notice here that the reaching out from the "Customer" to the Customer Class is modeled through a relationship with a FK inside the 3NF circle.

The same is true for the DV circle in that reaching out from "Customer" to Customer Class is modeled though a relationship (Link) with a FK in that Link and on the perimeter of the circle.

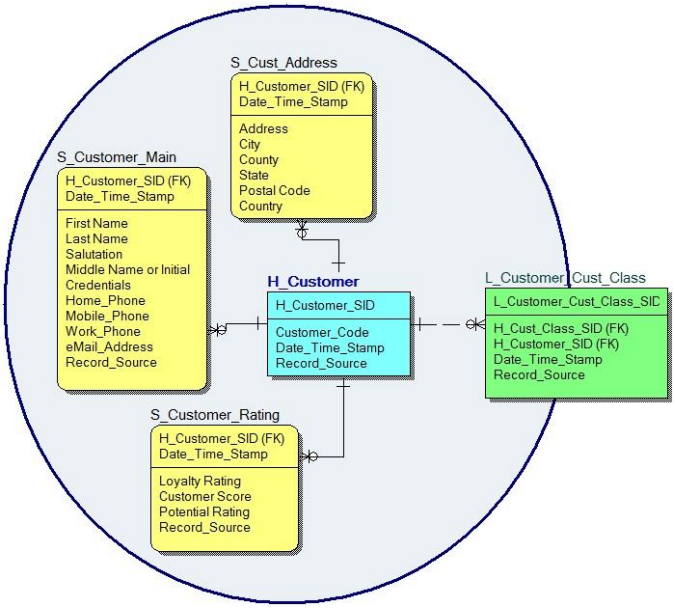


Fig. 11 ***Data Vault Model***

It is important to think of the DV circle in the same way as the 3NF circle.

This means that a) all things in either circle have the same idea of a key dependency, b) relationships pass through the circle directly from the

table with the BK, c) the only grain shift in either circle is based on Date/Time stamp for the purpose of tracking history.

> ***HINT:*** *As you progress with Data Vault Modeling, this view of thinking differently will become more and more important. We tend to see tables the same way we have always see them. For this reason, we tend to re-combine keys with relationships with context. But as soon as you do, you actually stop vaulting and return to other forms of modeling. So before you change the grain of a satellite, include a relationship FK in a Satellite or Hub, please consider the above circles analysis and reconsider.*

Modeling with the Data Vault

The process of modeling with the Data Vault is closely aligned with business analysis. The first step is to identify the CBCs for the given subject area. Once the CBCs are defined we next model the natural business relationships between the Hubs. Then we design and add the Satellites to provide context to these constructs.

STEP	TASK
1.1	Identify Core Business Concepts
1.2	Find/Establish/Create key for Hubs
1.3	**Model Hubs**
2.1	Identify Natural Business Relationships
2.2	Analyze Relationships Unit of Work
2.3	**Model Links**
3.1	Gather Context Attributes that Define Keys
3.2	Establish Criteria & Design Satellites
3.3	**Model Satellites**

Fig. 12 ***Steps to modeling with Data Vault***

This process is not concerned with separating facts from dimensions, or from separating master entities from events or transactions. The focus is squarely on core business concepts – and their unique business keys. In that regard, all of the above are candidates for CBCs / Hubs. For example, note that all events including transactions are modeled as Hubs.

The Business Key

At the core of the Data Vault is the **Hub** which we refer to as the ensemble identifier (enterprise-wide unique key). Perhaps the most important initial step in modeling a DV EDW is to identify and thoughtfully design these keys. To begin with, in operational systems, a Business Key is representative of the core business entity like "customer" or "product" for example. In addition, the BK also represents event based keys such as "sale" or "transfer". In this way, the design process for the Data Vault does not concern itself with the differences between the person/place/thing type entities and the event type entities. To put this another way, we are not concerned with differentiating Dimensions from Facts but rather are focusing on identifying Business Keys which can represent either.

This approach is then different from traditional approaches for modeling operational systems or data marts. The closest comparison would be to consider our efforts in defining Master Data elements for an MDM initiative. In this case as well, the focus is on the core terms used in managing the business.

Since the DV Program is organizational in scope, the business keys should also strive to be meaningful across the enterprise. So our quest for these keys should result in Enterprise Wide Business Keys (EWBKs). Note also that the keys arriving from specific source systems are typically not fully aligned with these EWBKs. For this reason, we do not place too much emphasis on the keys represented in any particular source system.

> *NOTE: Since we are typically dealing with dozens or hundreds of sources, each commonly subject to updates and changes, we should not plan to model our EDW using keys driven by a subset of these source systems.*

The process of identifying and modeling these EWBKs is then closer to a business requirements gathering process than a source system analysis. Balancing the various input factors, with an emphasis on the business versus the technical, effectively summarizes the best practices for this process.

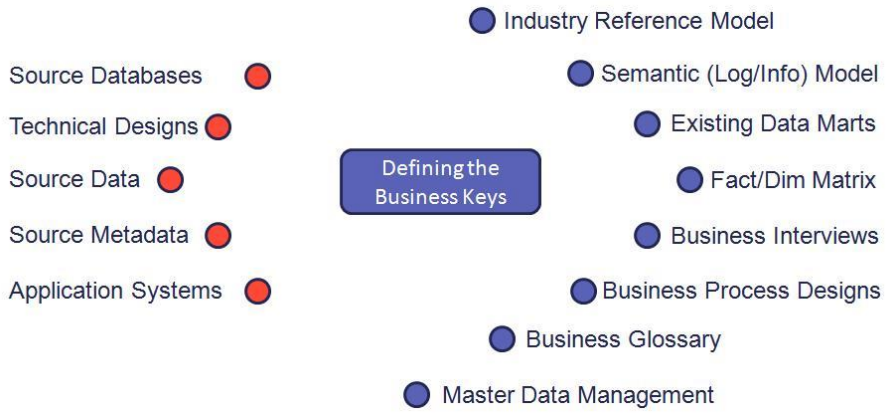


Fig. 13 ***Input factors for designing the DV EDW Business Keys***

The primary inputs for this process include:

First Ensemble Logical Modeling (ELM) workshops, business interviews, semantic models, logical models, and then business process designs, existing data marts, business metadata, process metadata, a business glossary if is exists, information models and the master data management artifacts (if existing) and the industry reference model (to the extent certain components are aligned) and then

Second the technical designs, source databases, source metadata, application system (guides, manuals, designs) and actual source system data.

Note that the Ensemble Identifier should be a key that transcends time and withstands the replacement of any specific source system. The source system keys will then require some form of alignment to match up with their related target Ensemble Identifier keys. This alignment will often be at odds with the fully raw and auditable characteristics of persisted source system loads. In the past we have either resolved this alignment on the way to the marts, or more commonly, created a cleansed "gold" record within the four walls of the data warehouse itself. The former solution leads to silos and anomalies while the latter can compromise auditability and user acceptance.

RAW and BDV Layers

Because the DV EDW absorbs all data all of the time and maintains full traceability back to source system feeds, the data warehouse must not lose resolution on these auditable systems of record. At the same time integration around the enterprise key – the Ensemble Identifier – is a core function of the DV EDW. So the EDW today has a built-in challenge related to data integration – the alignment of the Business Keys (enterprise-wide) with the Raw & Auditable components of the Data Vault.

Business Data Vault (BDV or BDW) is the end goal for the data warehouse. If all sources loaded into the data warehouse were perfectly aligned in terms of semantic meaning, consistent use of enterprise wide unique keys, and all at the same grain then we would not need to consider a raw layer. The RAW layer only exists because our source systems are not aligned, do not have consistent enterprise keys or a consistent grain. We need to land this data in the EDW as a true auditable path to the sources. See the "Raw Keys" in figure 14.

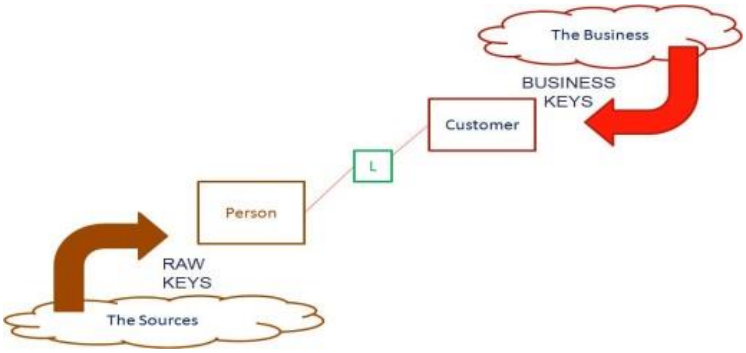


Fig. 14 ***DV EDW Key Alignment***

Ultimately the enterprise data warehouse (EDW) must be aligned with the organizational view of the business. For this reason the data is integrated, aligned and reconciled with the BDV layer. See the top right "Business Keys" in figure 14.

The integration of these Raw keys with the EWBKs represents a core function of the EDW today. In effect, we have been boxed in by the upstream requirements (build a DW that includes all data at the atomic level and with full traceability) and the downstream requirements (align the Ensemble Identifiers/Keys with the organization at the enterprise level using business terminology).

> *NOTE: We cannot rely on having these transformations happen in the Mart Staging or Data Mart layers as a) the Mart Staging is not intended to be persisted, b) the Dimensional modeling common to the Data Marts is not agile – does not respond well to changes, and c) Data Marts are typically departmental in scope (not Enterprise Wide).*

The naming conventions are not adequate in and of themselves to warrant the separation of Raw and Business key Hubs. If Person and Customer meant the exact same thing to the business and were in true business term synonyms, then the raw system load of Person records could populate the Customer Hub directly. However, in the example provided the Person concept means something different than the Customer concept. In this case there are business rules at play – for example a Person record is determined to be a Customer if they were involved in a Sale transaction, there was a non-zero purchase price, the transaction was successfully completed, and the Sale was not cancelled. As you can see here, the raw auditable load is to the Person and the business aligned load is to the Customer.

Those tables that are loaded using this type of business processing must be identified as "***sysgen***" record source records (generated by us through a business rule driven process).

> *NOTE: Business logic in the BDV can take many forms and can relate to many types of transformations. The logic specifically targeting the alignment of raw and business keys is a subset of this area and often referred to as the BAR component (Business key Alignment Rules). Since the primary objective of the DV EDW is to integrate and historize data from various sources, the BAR rules are the heart of this activity.*

The BDV will always have some level of BAR rules supporting the business aligned data warehouse.

Architecture

The high level DV EDW architecture includes an EDW that aligns the RAW and BDW layers. The high level architecture can be represented as seen here.

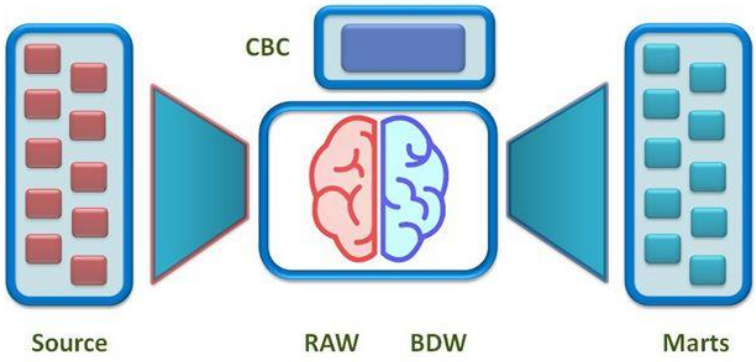


Fig. 15 ***DV EDW Architecture***

In a typical scenario, the stage area (between Source and RAW) is not persisted (data is not kept but rather overwritten). The EDW components are persisted, and the Data Marts are not persisted. This latter point represents an important distinction between dimensional (federated) data warehouses and the DV data warehouse. Unlike the DV EDW, Dimensional data warehouses are based on the idea of persisting the dimensions and associated facts.

The RAW and BDW layers represent a logical designation. While this separation is could also be physical the most common approach includes a logical separation between these layers.

Data Virtualization

As noted the data marts are not required to be persisted. Ideally the data marts exist only in-memory using some form of data virtualization.

> *The agility requirements of the modern data warehouse have spurred the development of agile modeling techniques (Ensemble methods like Data Vault), agile project management approaches (SCRUM, BEAM and others), automation and tooling (Varigence, Data Vault Builder, WhereScape, etc.) and other agility enabling features in the broader DWBI program (testing, production move schedules, etc.).*

Because dimensional models do not adapt easily to change, persisting data mart layers compromises this program agility that we are otherwise addressing. For this reason, the ultimate data warehouse architecture includes a virtualized data mart / data presentation layer.

Sample Data Vault Model

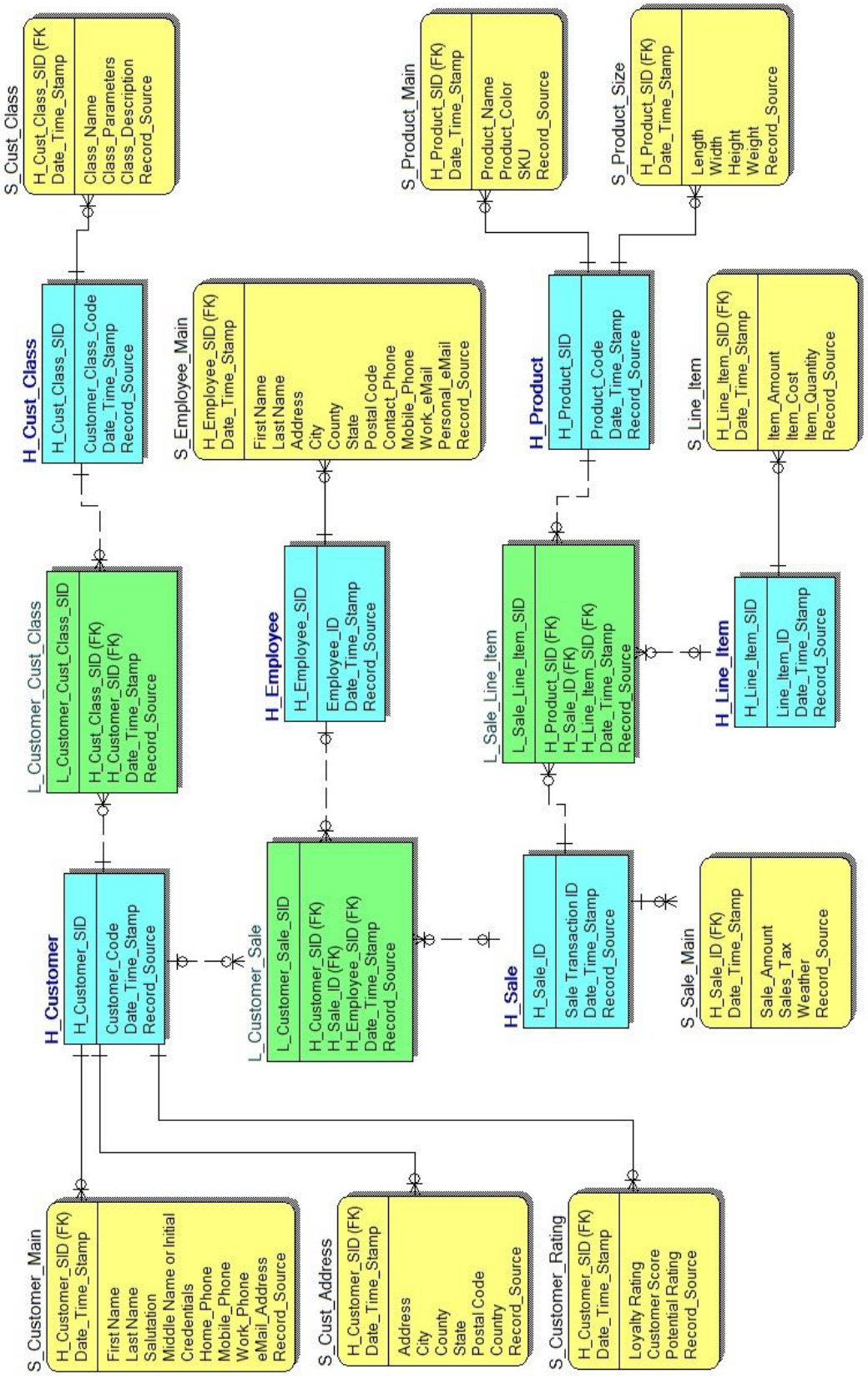


Fig. 16 ***Full Data Vault Model***

Useful Extra Tables

The data vault approach has defined a set of useful extra tables that can be used to make the overall deployment more efficient. These are applied on a case-by-case basis as appropriate for the specific circumstances.

The **Point In Time table (PIT)** is a modified Satellite table that tracks the valid time slices of the satellites surrounding a particular Hub. This is populated to make the process of associating relative context/descriptive data together for reporting purposes.

The **Bridge table** is a modified Link stable that flattens the relationship between Hubs including important related context/descriptive data (potentially also the business keys) into a single table for ease of access and performance.

In all cases, these and other constructs can coexist in the DV EDW provided however that they are always noted as "sysgen" tables and utilized only for performance reasons. The related historical and auditable data that is used to load these constructs must remain the sole source of the EDW data over time.

Applying the Data Vault

Data Vault modeling is uniquely useful when modeling a data warehouse. An Enterprise Data Warehouse (EDW) project is specifically well aligned with the features of data vault modeling. One primary benefit is the ability to adapt easily to changes in both upstream sources and downstream data mart requirements. This provides us the ability to build incrementally and to run a truly agile data warehouse program. The data vault data warehouse also easily integrates data and inherently manages history providing for a true enterprise data warehouse.

Data Vault modeling has also proven to be the preferred modeling pattern for special data warehouse situations including truly operational data warehousing, Big Data integration, Streaming, Virtual Data Warehouse (SuperNova), Information model based DW models, and meta-data driven data warehouse deployments.

Understanding the full benefits of the data vault & ensemble modeling patterns starts with getting your certification. This process is facilitated by Genesee Academy and includes materials, online lectures, exercises, three days in a classroom with lectures, labs and group modeling exercises. On the last day there is an exam which results in the certified data vault data modeler (CDVDM) designation.

Please visit GeneseeAcademy.com for more information on course schedules and registration.

Final Note

The Data Vault approach is growing and adapting from year to year. Incremental changes to the modeling approach, rules and best practices can be expected with some frequency. Please note that this guide should be applied in concert with current updates found online on data vault forums, LinkedIn and from certified practitioners. See Data Vault Standards online for the latest guidance from the international Data Vault & Ensemble modeling Enthusiast (DVEE) consortium.

GENESEE ACADEMY
World-Class Training

www.**GeneseeAcademy**.com

Data Vault Standards & Guidance

www.**DVStandards.**com

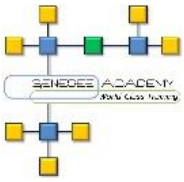
Hans@GeneseeAcademy.com

Gohansgo

HansData.WordPress.com

HansHultgren

Genesee Academy



Online video-lesson training

eLearning.GeneseeAcademy.com