Data Vault Basics

data vault basics

here, we offer the best of breed, hybrid data modeling solution for your enterprise integration needs. join the growing community today, and interact with the data vault community. learn how to meet the needs of the enterprise faster, cheaper, and more reliably.

the data vault architecture offers a unique solution to business problems and technical problems alike. it is focused squarely at the data integration efforts across the enterprise and is built from solid foundational concepts. a key to understanding the data vault is understanding the business. once the business is mapped out and the practitioner has a firm grasp on how the business operates, then the process of building the data vault can commence.

the data vault has many benefits which are produced as a by-product of the basic engineering. sticking to the data vault foundational rules and standards will help get any integration project off the ground quickly and easily. there are several areas of the data vault which we'd like to cover with you before diving into the community / forums. in case you are interested, you can also read about some of the ***issues*** faced by those who undertake the data vault modeling.

it is very easy to convert both 3rd normal form and star schema to data vault model architecture, here we show how to convert from 3rd normal form. inside the community we walk through the conversion steps to go from star schema to data vault model.

business benefits of data vault modeling

- manage and **enforce compliance** to sarbanes-oxley, hippa, and basil ii in your enterprise data warehouse
- **spot business problems** that were never visible previously
- rapidly **reduce business cycle time** for implementing changes
- merge new business units into the organization **rapidly**
- **rapid roi** and delivery of information to new star schemas
- consolidate disparate data stores., ie: ***master data management***
- ***implement and deploy soa, fast.***
- **scale** to hundreds of **terabytes or petabytes**
- **sei cmm level 5** compliant (repeatable, consistent, redundant architecture)
- trace ***all*** data back to the source systems

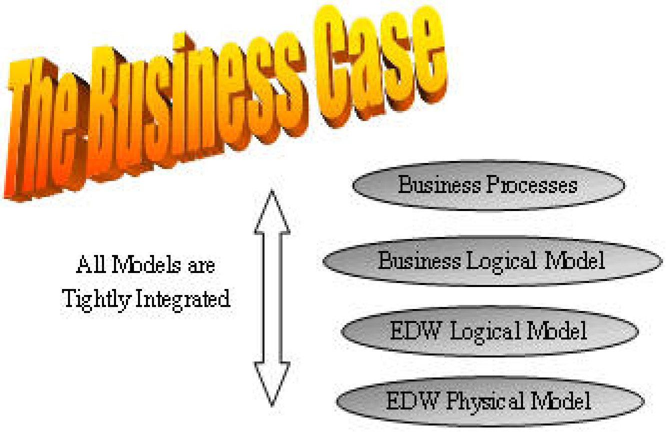
the following are the concepts which we cover on this site:

- definition of the data vault
- data vault benefits
- from business case to data vault
- data vault in 5 easy steps
- data vault through pictures
- sei / cmm / compliance

definition: 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 encompassing the best of breed between 3rd normal form (3nf) and star schema. the design is flexible, scalable, consistent and adaptable to the needs of the enterprise. it is a data model that is architected specifically to meet the needs of today's enterprise data warehouses.

- extensive possibilities for data attribution.
- all data relationships are key driven.
- relationships can be dropped and created on-the-fly.
- data mining can discover new relationships between elements
- artificial intelligence can be utilized to rank the relevancy of the relationships to the user configured outcome.

in business speak, it's the ability to adapt quickly, model the business accurately, and scale with the business needs – converging it and the business to meet the goals of the corporation. the data vault is a data integration architecture; a series of standards, and definitional elements or methods by way information is connected within an rdbms data store in order to make sense of it.



the following business benefits are available as a result of building a data vault within an organization:

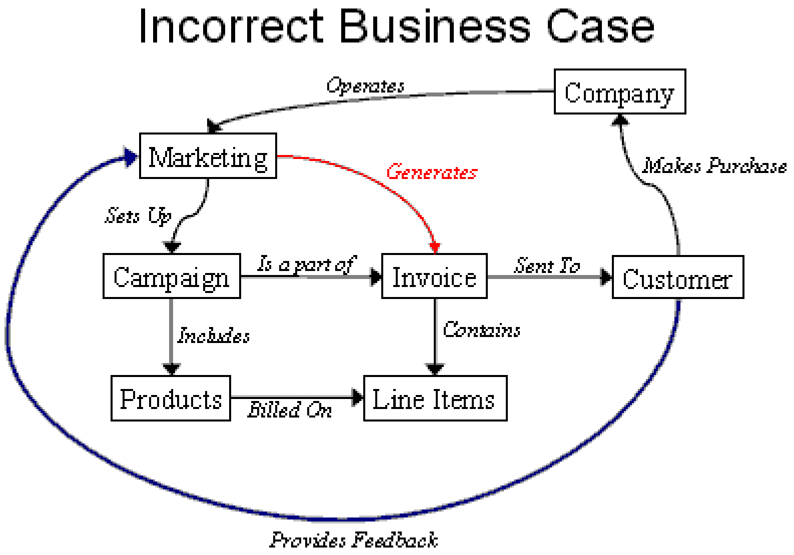
data vaults are extremely scalable, flexible architectures which allow the business to grow, and change without the agony and pain of "long it lists of change impacts, and large cost outlays." typically when businesses request a change to the data models (as a result of business changes), it comes back with high costs, long implementation and test cycles, and long lists of impacts across the "enterprise warehouse". with a data vault this is not the case, typically new functional areas of business are added quickly and easily, changes to existing architecture take less than 1/2 the traditional time, and usually have much less impact on the downstream systems.

technical benefits:

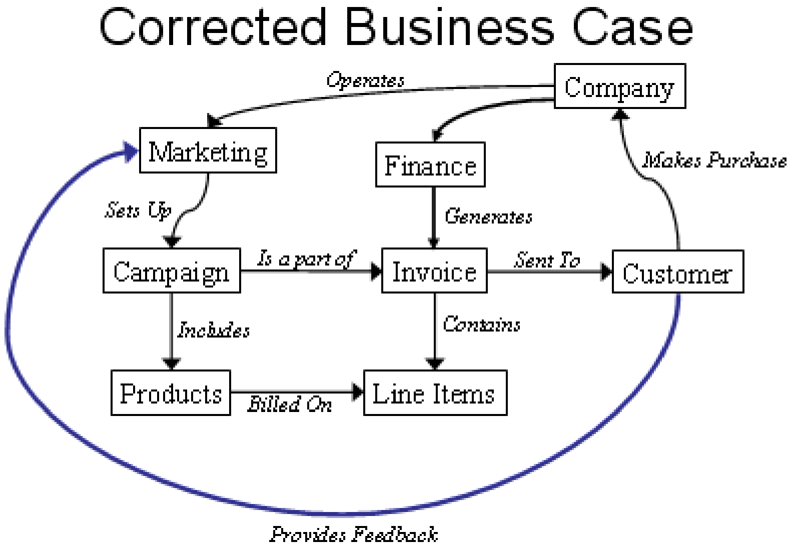
- near-real-time loads
- traditional batch loads
- in-database data mining
- terabytes to petabytes of information (big data)
- incremental build out
- seamless integration of unstructured data (nosql)
- dynamic model adaptation – self healing
- business rule changes (with ease)

since the data vault is based heavily on business process, it's important to see how the business model represents the data vault, and how to make the transition from one to the other. below is a series of descriptions and transitions that take you from one state of the business case model to the physical data vault data model that represents it. it also indicates how tightly tied the architecture is to the business itself. it also indicates how quickly the model can change when the business changes.

for this example, let's examine a company with a marketing department that wants to build a sales campaign to sell slow moving products. the hope is that the customer will see the campaign, like the new low price of the product, and make a purchase from the company. when the analyst re-iterates what they heard in the user interviews, the analyst says to the business users, that he heard them say: "marketing generates a sales invoice". the business users quickly correct him (and the case model) and state that finance generates the invoices. of course, for this example hopefully the customer then provides feedback to the company about the marketing campaign, and company friendliness.

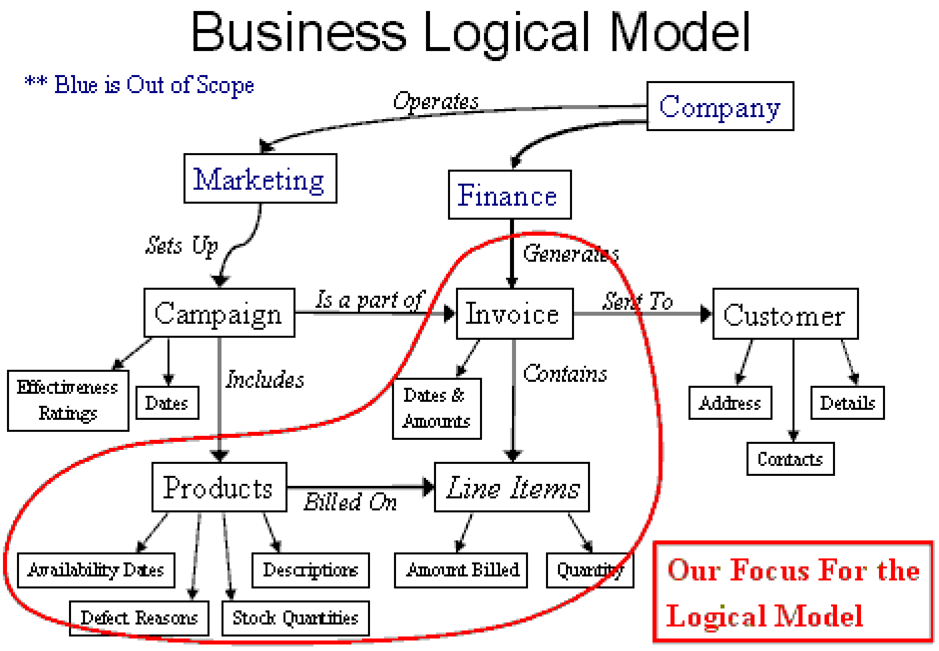


in the corrected business case, we change the model from the representation above, to the representation below. not a large change, just a difference in departments. hopefully there is communication between finance and marketing (which is not shown in this case example).

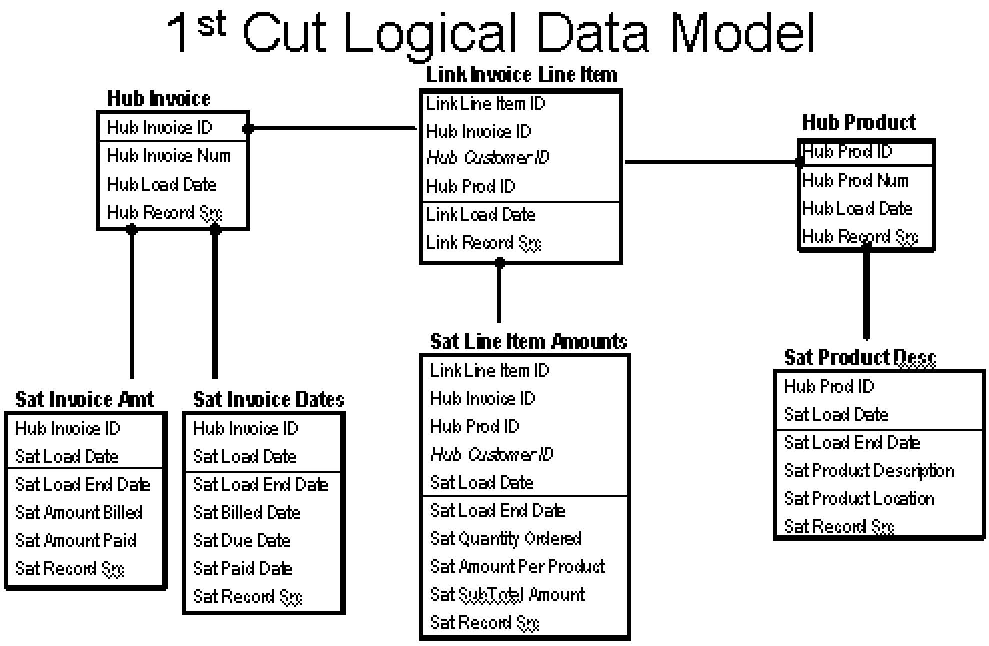


when we move on to discuss implementation of the business model, we begin to pair down the scope of the project – in order complete it in budget, on-time, and with limited resources. in this particular case, the company decides that we can add marketing, finance, and the enterprise (company) later – thus making it out of scope. they tell us that we should focus on the implementation of campaigns, invoices, products, and customers. all of which are free-standing elements in business and have their own “tracking numbers”. in other words, it’s campaign mkt-1, invoice numbers, customer accounts, and product numbers that tie all this data together.

upon further investigation, we discover that the business wants to track campaign effectiveness ratings, dates (length of time for the campaign), dates and amounts on invoices and line-items, products and their availability dates, descriptions, stock quantities and defect reasons; finally we discover they already have customer addresses, contacts, and other demographic details. the next model that we build is based on the concepts of the data vault and ties directly back to the business descriptions of granularity of data, and keyed information. for the purposes of scope and this example, we will limit the logical data model to the area outlined in red.



below is what our first cut logical data vault data model looks like. we take the business keys such as invoice number, and product number and build them into their own hubs. then, we take the interaction between invoices, and products and build a link table called link invoice line item. line items cannot "stand alone." they depend on other key information to locate, and describe what they belong to. there is however, an error in this model (see hub customer id embedded in the link invoice line item).



the error is a result of shifting grain into the wrong location. the correct grain is: invoices are tied to customers (usually 1 to 1 correspondence), with a customer capable of having many invoices. it is very rare (although does occur) that a business can actually have different customers per line-item, all build on the same invoice (this is the grain represented above). the proper line-item link table would not contain a customer id. the corrected model would have an additional link between hub invoice and hub customer, to represent the interaction between customers and invoices.

is the data vault simple or complex? is it easy to implement? yes, it's both simple, and easy to implement. it's based on a set of redundant structures, and auditable principles. by utilizing the data vault standards your project will automatically gain the benefits of auditability, scalability, and flexibility. the following set of web-pages will guide you through the process of building a data vault in 5 easy steps.

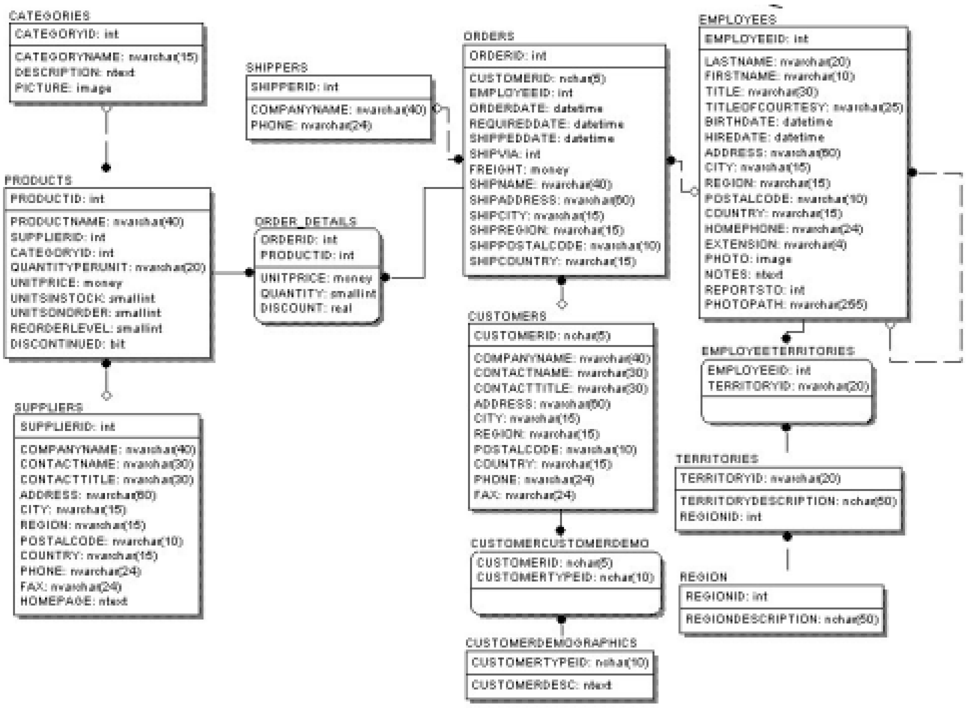
step 1: establish the business keys, hubs
step 2: establish the relationships between the business keys, links
step 3: establish description around the business keys, satellites
step 4: add standalone components like calendars and code/descriptions for decoding in data marts
step 5: tune for query optimization, add performance tables such as bridge tables and point-in-time structures

build your data marts, and your etl loading processes, and away you go. building a data vault gets easier as you go, eventually replacing the “band-aid” methods commonly used in enterprise integration architectures. the model is built in such a manner that it can easily be extended when desired. the worlds smallest data vault consists of 1 hub with 1 satellite, say customer. the flexibility is built into the link-table concepts.

for those of you wishing to view examples of data vaults, we’ve provided some samples of different models. these are for viewing purposes, and are generic models which can be customized to meet your needs. the ddl for the northwind data vault is available inside the forums. this is a standard data model, and full size images are available on request.

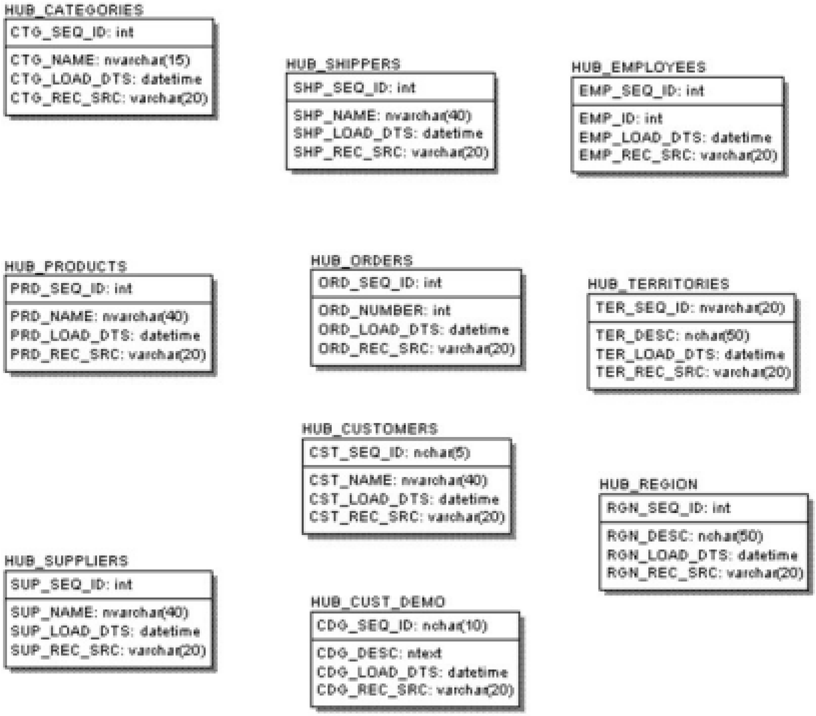
we start with a simplified 3nf data model, often times the source data model represents the business as it stands today – at least it represents the business processes which collect the information. of course, we wish to correct some of the errors occurring in the business process as well.

northwind 3nf data model:



the following is the first step in the series of 5 identified above. identifying the business keys, and placing them in the standard hub structures. this can be challenging if the business keys are “smart-keys”, or composite keys made up of several identifiable relationships. although, normalization to the nth degree is discouraged, it is good to identify and document the metadata for these composite keys.

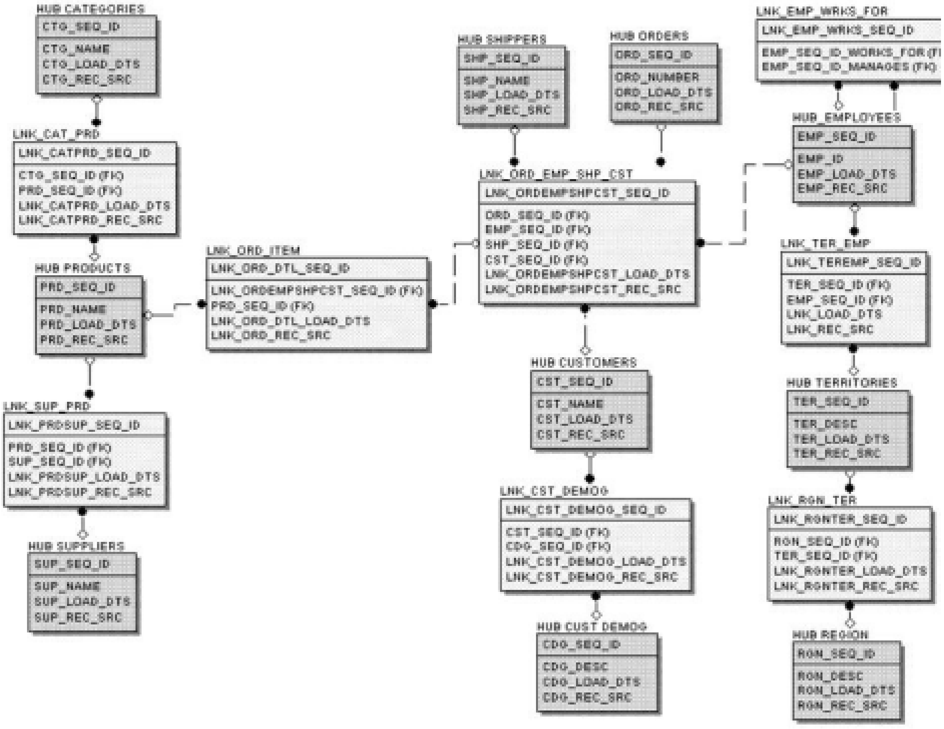
northwind data vault model, hubs identified



step 2, identifying the links or relationships between the business keys. this process can be a little tricky sometimes – particularly if the data set says the business keys are "weak", and can't stand on their own, or are non-identifying (non-unique). in these cases, we can end up with one-legged link tables, which is something we work through to correct during the modeling process. again, the link tables represent relationships, transactions, hierarchies, and define the grain of the data on the intersection.

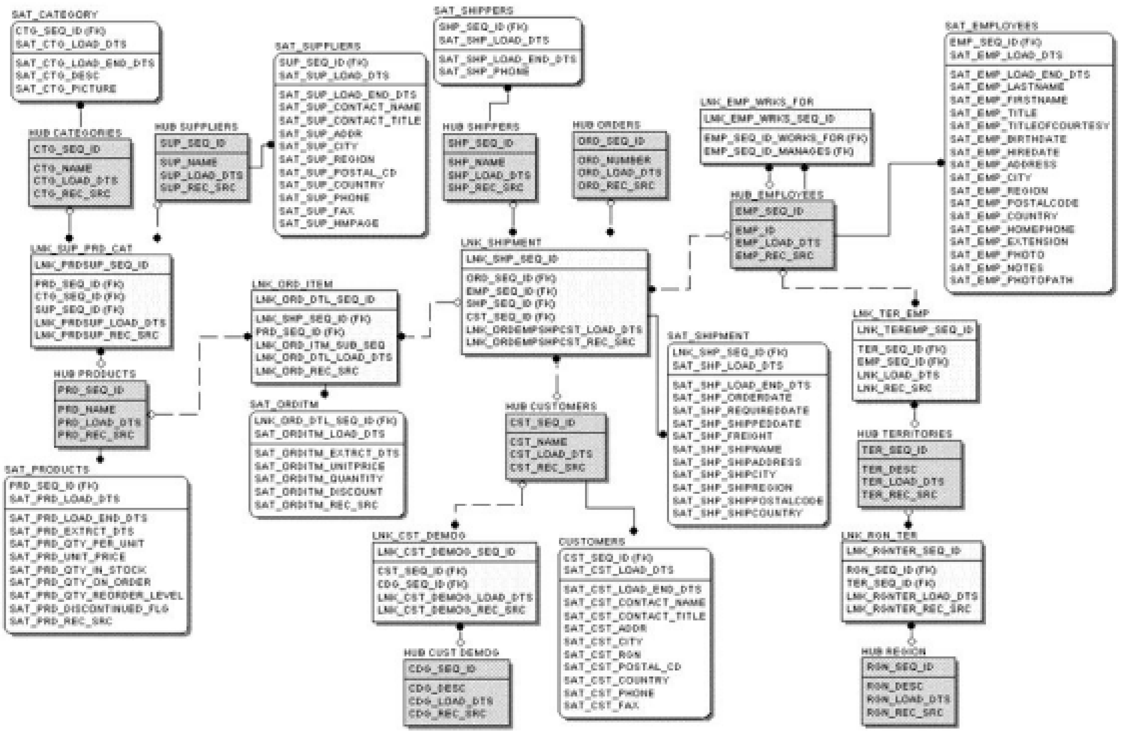
please note the importance of grain defined between order, employee, shipper and customer. the grain of this relationship (link table) is defined by the source system, however it is subtle and easy to miss.

northwind data vault model, hubs and links



the final image below shows a complete data vault, with all the hubs, links and satellites available. while the white-papers have defined point-in-time tables, and bridge tables, they are not required by the architecture. these alternate tables are utilized strictly for query performance reasons. there are certain mpp databases which can execute without “query assist tables” such as point-in-time and bridge tables. that said, many of the data vaults in place today are not implemented on mpp systems, and thus require these tables to be present in order to reduce the number of joins.

complete: northwind data vault model, hubs, links and satellites



the data vault comes with rules, or standards which make the design repeatable, and redundant. part of sei/cmm (software engineering institute, capability maturity model) is to have organizations reach a level of business processes which are: documented, repeatable, redundant, fault-tolerant, and eventually: automated. it also defines risk analysis, kpa’s (key process areas) and kpi’s (key process indicators – metrics) for the organizations by which they can measure the improvement and accuracy of these business processes. these are equivalents to iso900x, pmbok, six-sigma practices in other disciplines.

compliance has been around for years in the private sector, and when the government put a mandate out that all government contractors will become sei/cmm level 5 compliant, compliance took on a whole new meaning. data traceability came bubbling to the top, accountability of business users to find and fix their own business problems also rose to the top.

compliance itself has many meanings, but in the case of sarbanes-oxley and basil ii/iii accords, and other compliance initiatives – it strikes hard on data traceability, and business accountability. the it staff must be enabled through the use of their models (data vault or not) to track data as it stood, when it was created. in the case of the data vault, we’ve made it easier. the data vault provides a series of standard fields which track data changes by date, and where they came from – and the data vault is always modeled to hold the lowest possible grain of data from the source systems. that is to say: data is integrated, but not modified when it is stored in the data vault. the satellites are split by type of data and rate of change which allow reduced storage requirements, and increased traceability.

the business rules which are usually implemented “on-the-way-in” to the data warehouse, are moved, shifted, to be implemented “on-the-way from the warehouse to the data marts”. thus, allowing a single active data warehouse (data vault model) to be built as a statement of fact, and each of the data marts can represent their own “version of the truth” based on a single sourcing point.

when following the standards in loading information into the data vault, the it staff automatically inherits “data compliance and traceability”. the business can point at the data mart and claim that it’s “wrong” today, and “right” tomorrow, but it can always show where the data came from, when it came in, and what it looked like before alteration/transformation, aggregation and cleansing – thus meeting the compliance initiatives becomes easier.

sei/cmm processes are defined as a part of the standards and have been built into the architecture, making the loading process, querying process and discovery process repeatable, redundant, and fault-tolerant.