

## Data Vault Business objectives for next generation data warehousing.

*Tom Breur, Principal XLNT Consulting*  
*Ronald Damhof, Principal at Prudenza B.V.*

### Introduction

Business intelligence (BI) is a relatively new profession. Although there have been some (very) early attempts (e.g.: Luhn, 1958), it wasn't until data warehousing was 'invented' in the 90's that large numbers of businesses could reap sufficient benefits to make these endeavors economically viable. Kimball and Inmon have fought a ferocious debate on how "best" to store and organize data in a central repository.

There are considerable technical, organizational, and software engineering challenges to overcome when engaging in a data warehouse project. This has given BI and data warehousing a dubious reputation. Expectations were not always met. There's pressure to deliver more business value in less time. Regulatory needs, data quality, scalability, in short: business aligned solutions that can withstand the test of time.

It has taken a while to see the illusion of a "single version of the truth" for what it is: an attempt to solve business problems with a technological solution. Of course that can never work. You do need central storage of facts (including history), though, yet at the same time make room for multiple interpretations of reality. By performing integration (and major Transformations) as late as possible, you create economic solutions. You also help the business surface requirements when they're ready. And no sooner.

By design, Data Vault enables businesses to use data quality issues to move the business forward, and create alignment instead of friction between BI functions and operational processes. It is perfectly suited for Agile development methodologies, and hence efficient delivery of sustainable BI solutions.

### Kimball - Inmon debate

The data warehousing profession grew up under the assumption one had to choose between an Inmon (Corporate Information Factory, 1997) and a Kimball architecture (The Data Warehouse Lifecycle Toolkit, 1998).

This now turns out to be a false premise. Devised in the 90's, and popularized over the last decade, a new and superior approach to enterprise data warehousing has emerged. Data Vault combines the best of both worlds: a hub and spoke model (Inmon) where data are centrally stored and subsets of the data (data marts) are derived.

Furthermore, Data Vault enables business area specific, incremental development of functionality (Kimball). On top of this, Data Vault is the architecture of choice for agile data warehousing. It enables both a future proof architecture (minimizing rework), as well as allowing for rapid and very early delivery of functionality. The Data Vault helps guard the long-term value of the data warehouse by providing a guaranteed linear path to development. This avoids the progressively incomprehensible and unwieldy solutions characteristic of "classic" Kimball EDW's.

### Business objectives that have led to the Data Vault approach

The first priority in setting up an architecture and infrastructure for any Enterprise Data Warehouse (EDW) is to clearly define business objectives. One should avoid getting bogged down in a quagmire of technology.

Instead, the objectives should be worded in such a way that they are understood by non-technical business people, and backed by senior management. Obviously, these ambitions must be supported by a detailed business case. That's how you determine the purported returns for the organization and a realistic time window for break even.

The bar has been raised with regards to business objectives, and this calls for a new generation of data warehouses with superior architecture and infrastructure:

Complete traceability and auditability of data. Data stored in the EDW must be traceable to the original source, and be auditable in terms of reliability, completeness, accurateness and correctness (Olson, 2003). Additionally, the data must meet contemporary compliance requirements. Basel II (and III) in banking, and Solvency II in insurance, for example, pose stringent data quality requirements.

Uncoupling of warehouse data and operational systems. Changes in source systems should have minimal impact on the EDW and no consequences for the end user. When properly componentized, data structures in the EDW can be significantly more stable, compared to constantly changing organizational processes and their impact on the operational system landscape. This decoupling is necessary to architect robust BI systems.

Data quality and business rules are the responsibility of the business, not the domain of the EDW team nor ICT department. Your EDW must provide a clear separation of these responsibilities.

Scalability in its many guises. An EDW is, by definition, an incrementally growing architecture. More data, more users, more applications. A future proof architecture must allow the infrastructure to grow linearly with requirements.

The EDW must enable business users to integrate data any way they see fit. Maximum flexibility is required. This should be done 'just-in-time', no sooner. 'Just-in-time' means we only integrate when the business requires this. Storage needs may have preceded this, sometimes considerably so.

EDW must constantly drive down delivery times of newly requested information products. Contrary to the often observed progressive increase in delivery costs (and time) for "classic" data warehouses, a sustainable solution must show a steady decline in delivery times. Data Vault supports this notion, in particular when (almost) all source data have been loaded in the Data Vault already.

Data warehouses are increasingly applied for operational purposes. Data from the originating sources must be delivered ever faster to end users (downward trend in data latency). The next generation EDW must be prepared for a high mixed workload (Linstedt et al, 2009).

It should be able to handle both query and aggregation jobs, as well as high throughput processing that is traditionally associated with an operational data store (Inmon, 1999). When properly architected, a Data Vault can obviate the need for an ODS, at considerably lower cost.

The data warehouse must also comply with contemporary requirements of software engineering. We've grown beyond the age of 'fooling around with data.' An important and useful indicator is provided by the five maturity levels of the Capability Maturity Model Integration (CMMI®) as drawn up by the Carnegie Mellon Institute (Carnegie Mellon, 2010). A data warehouse should have a minimum ambition of maturity level 3 for established businesses.

Primary business processes usually live in symbiosis with supporting IT and applications. When these (fundamentally OLTP) systems are subsequently charged with reporting requirements and data delivery, this tends to compromise their architecture.

The operational system must support the primary process, but often these same systems are also burdened with ad hoc questions and numerous datasets for various (external) users (typical "ODS-style" functions). Horses for courses. By decoupling BI functionality from the primary process, both systems can gravitate towards an optimal (simple!) design, and BI output can actually strengthen the primary process by providing accurate and timely information.

The new generation EDW enables organizations to effectively access sources and distribute them in a manner that satisfies all the aforementioned ambitions. On top of that, an EDW must of course provide economic performance and user-friendliness, while BI services (reporting, analysis, mining, etc.) must facilitate deployment of information products.

Below, we explain how these objectives can be met by taking a different approach to data and data logistics. The principles as laid out by Dan Linstedt with his Data Vault architecture play a very prominent role in this.

## The false prophesy of a 'Single version of the Truth'

Up until now, data warehouses had the objective, implicitly or explicitly, to create a single version of the truth for their users. Companies were struggling with inconsistent information drawn from disparate data silos. They embraced the data warehouse as the ideal solution for the underlying disarray.

Gradually, however, they discovered that this was little more than an attempt to create an IT solution for what was, in essence, a business problem. For if the organization does not have a company-wide uniform data vision of its own, the data warehouse is unable to impose this. After all, there will always be users who (voluntarily or out of necessity) adopt a deviating view of the available data.

By enforcing "one version of the truth" in the data warehouse, those 'dissident' users are left out in the cold. Let's take for example a relatively 'simple' yet commonly recurring theme like "how many customers do we have?" Marketing may need to take a "lifecycle view" thus embracing both prospective customers at initial stages of acquisition, and lapsed customers they would like to win back. Finance will base their view on all parties who have made some kind of "payment", and operations will have dialogues with any party regardless of lifecycle stage or payment status. How can their numbers ever match? Usually they can't, at least not without hampering one of these parties' primary process. So there is no "single version of the truth", even when these business units may agree on a single version of the underlying facts.

More pernicious is a (very common) situation where insight into what actually constitutes "the truth" evolves over time. Only after delivery of information products can business users dig into the data, and clarify their specifications for how data should be presented. This is, by nature, an iterative process. It's never "done." How often have you sat through ever-lasting requirements workshops, only to see them change almost instantly after delivery of solutions-as-specified? Our BI solutions need to acknowledge this reality.

It is simply an illusion to ever expect a “common truth” to be known. Or to attempt to specify once-and-for-all-right user requirements. Yet as time 5 by, the EDW loses its opportunity to go back to the source systems and recapture reality (“the facts”) as they were known back then.

Other considerations also started to play a role. Roughly since the 21st century, companies have been confronted with the need for compliance with regulations such as Sarbanes-Oxley, Solvency, HIPAA or Basel II, to name just a few. Solvency II, specifically, is being held up in large part by data management issues (FSA UK, 2011).

These requirements are squarely at odds with storage of interpreted, processed data. If the original source data is no longer explicitly tied to contents of the DWH (and the data warehouse is typically the only place where history is stored), we lose sight of the ‘data trail’; data is no longer traceable and the data warehouse is not compliant.

New generation data warehouses abandon the idea of a single version of the truth, adopting instead a “truth is in the eye of the beholder” perspective. The goal is therefore to create a system, i.e., a “single version of the facts” (which are and remain fully auditable and traceable), which allows for idiosyncratic interpretation.

This single version of the facts now becomes a time-invariant system of record (S-o-R) where the facts as they were known to the business at *any* point in history are stored. This offers scope for multiple versions of the truth, depending on usage and business line. Whatever interpretation users apply, they can always be attributed to the bare facts (at least the way they appeared in source systems) – which provides the additional benefit that your EDW now does continue to meet compliance requirements.

## Data quality

The desire to maintain a system of record (S-o-R, an EDW as we define it here) also leads to a different perspective on data quality. A fundamental principle is that all data should always be loaded, irrespective of its (low) quality. We load 100% of the data, 100% of the time. Justification for this principle is found in the following three arguments:

- “Low quality” is a matter of interpretation; what may be unusable (unacceptable) for some users, might be adequate for others. Moreover, this concerns data that are stored on an operational system that may form the basis for day to day decision making. It is therefore up to individual user groups to determine whether the data is good enough to use - and to specify how this quality should be improved in their version of the truth.

- What is considered “the truth” can sometimes evolve over time. After your EDW goes “live”, end users will begin to challenge the facts. Sometimes rightfully so. It is then up to the business to decide how to deal with irregularities, and to decide if and when they should be corrected in the source systems. As these changes are recorded, your EDW becomes a persistent S-o-R.
- There is tremendous value in storing bad data; it is the place where breakdowns in your corporate value chain will surface. Where they become tangible and measurable. “Bad data” is rarely a technical anomaly (Breur, 2010). In the overwhelming majority of cases it points to cracks in your value chain. The data warehouse is perfectly placed to mine and expose the frequency of occurrence and context of these data faults. Armed with a comprehensive view on data quality errors, senior management can decide when and how to fix these broken business processes. And reap rewards commensurately.

## Data integration

The T in ETL stands for Transformation and the question is where the bulk of the transformations (“big T”) should be positioned in the EDW. In traditional data warehouses, it was positioned between staging and the EDW (see Figure 1). It is said that the data warehouse should contain clean data; the concept of ‘a single version of the truth’ often relates to this notion of the data warehouse..

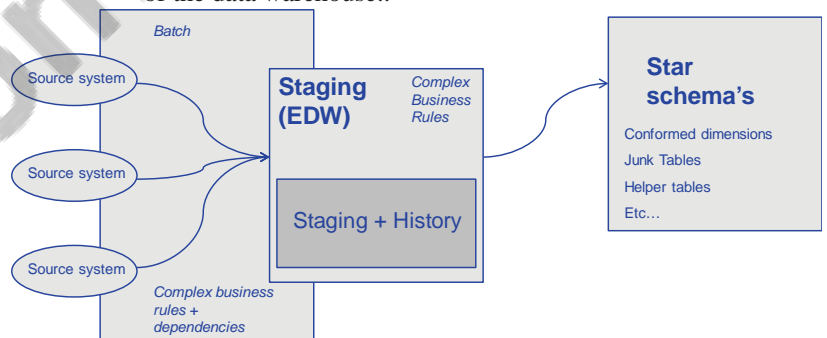


Figure 1: Traditional architecture, business rules before the EDW. Source: Dan Linstedt, Kent Graziano & Hans Hultgren (2009)

However, in time this will give rise to problems:

- increasing difficulty/impossibility to trace the data;
- increasing difficulty to scale the ETL or succeed to load within batch windows;
- projects take ever longer to complete, becoming more complex and expensive – a painful manifestation of the size-complexity dynamic (Weinberg, 1991);
- diminishing flexibility of the data warehouse architecture, i.e., its ability to adjust to continuously changing requirements of the business in a timely and economic manner;

# Data Vault, Business objectives for next generation data warehousing

- Premature decisions on how to handle history. In the past we needed to make a choice on how to handle storage of historical facts upstream from the DWH. Yet many questions haven't been asked! This is why you want to commit to these choices as late as possible. Late, so as to postpone necessary investments. And also late to allow the business more time to surface their requirements.
- Testing, especially regression tests become increasingly difficult. The result is a data warehouse that becomes outdated relatively quickly, thereby losing its ability to support the business with high-quality and timely data.

It is plain wrong to position Transformations upstream from the EDW. In the next generation EDW, the “big T” is positioned downstream from the EDW (see Figure 2). Preferably as close as possible to end users and their requirements. Ideally, the Transformations should take place the very instance that the user requests it (“just-in-time”). That is often not possible, however, primarily due to (technical) performance reasons. The rise of solid state disk, in-memory tooling and Analytical

When properly designed, a Data Vault behaves just like a fractal with infinitely repeating (recurring) patterns that are essentially identical in shape and structure. The Data Vault has so few of these self-similar structures, or model component patterns, that there is ample opportunity to automate code generation for ETL as well as query access.

The EDW is modeled according to a standard method (Data Vault) with a limited number of entity types. Data Vault has only three components: HUB's LINK's and SAT's. An expanding EDW exhibits behavior similar to a fractal, it's simply “more of the same.” Every entity type has uniform loading characteristics. This level of standardization opens the door to ETL generation rather than traditional coding.

It is important that ETL suppliers recognize this development and support it with either standard ETL templates/mappings or some form of metadata-driven data warehousing. And this area is booming already. As a result, loading data from the source to the EDW becomes a kind of ‘conveyor belt process’. Virtually any request made by the business can be met quickly by entering the required source data into this highly automated process.

This allows for more rapid and economic delivery of BI solutions. What's maybe just as important, these development efforts can be much more accurately estimated than “traditional” data warehouse approaches ever allowed. And changes as well as maintenance become standardized (efficient) and predictable.

## Conclusion

As our profession is maturing, rules about “best practices” are evolving. Over the past two decades we have learnt how EDW's should be designed and built to embrace changing requirements. And to gracefully absorb these changes.

On top of that, we've seen new needs arise, like the desire for auditable and compliant data warehousing. As organizations rely on BI for regulatory requirements, there information products need to withstand the scrutiny of auditors. This is not something we've been used to doing, so we need to adapt our storage accordingly. The EDW needs to contain traceable data, over time taking on the role of System-of-Record.

Standing on the architectural shoulders of giants like Devlin (1988), Inmon et al (1997) and Kimball et al (1998), Dan Linstedt's (2011) Data Vault methodology is a next evolutionary step in enterprise data warehousing.

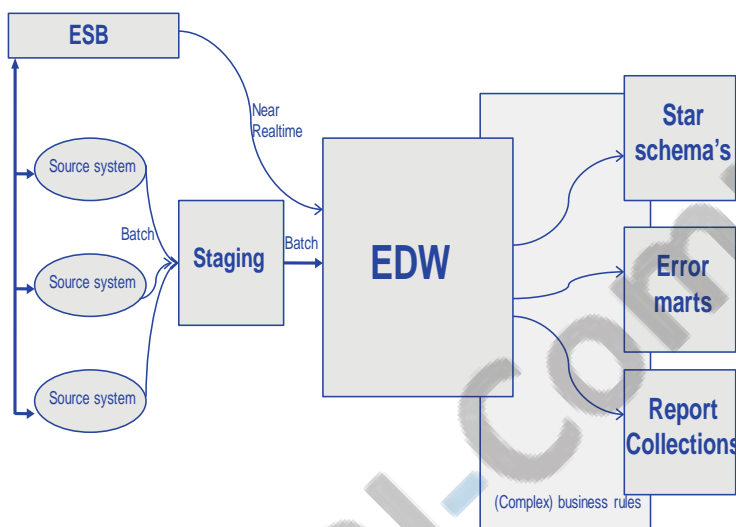


Figure 2: Fundamental architecture; Business rules after the EDW  
Source: Dan Linstedt, Kent Graziano & Hans Hultgren (2009)

DBMS's may change this in the foreseeable future. An important architectural concept of the next generation EDW is that data isn't integrated until the business requires this. Upon demand, data gets integrated between the EDW and data marts. The EDW itself is filled with one-on-one data from the source systems that can be loaded in a parallel, asynchronous, and highly standardized manner. This is due to the “fractal” nature of the Data Vault (Scholten, 2010, private conversations).

The business benefit of a Data Vault's similarity to fractals (Mandelbrot, 1977) implies that when properly designed (i.e.: adhering to proven standards and guidelines of good Data Vault modeling), there is enormous opportunity for efficiency gains.

Data Vault methodology has two fundamental architectural characteristics: a distinction is made between facts and truths at conceptual, logical and technical level; the level of data integration is determined by the end-user and, if technically possible, is carried out at the time of request.

What does that mean for the architecture and the data warehouse process? A radical change. The central data warehouse is no longer a clean, integrated data collection; it serves solely as a storage space for facts. Integration and cleaning only take place after the data warehouse, when the data marts are made. Tailored to meet user needs, it reflects the user's view of the facts and thus the user's truth.

Because a Data Vault model is simple, with abundant self-similarity in its structures, a high degree of standardization is possible. This enables automation of ETL and loading, which allows for fast and therefore economic delivery of BI products. The inherent support for Agile delivery makes it possible to postpone many engineering decisions until a point in time where business users have been able to interact with the data. In 2008 Bill Inmon deemed "The Data Vault is the model of choice for EDW 2.0"

## References

Luhn, P. (1958). A business intelligence system. IBM Journal of Research and Development, Vol. 2, issue 4.

Bill Inmon, Claudia Imhoff & Ryan Sousa (1997) Corporate Information Factory. ISBN# 0471197335

Ralph Kimball, Margy Ross, Warren Thornthwaite & Joy Mundy (1998) The Data Warehouse Lifecycle Toolkit 2nd Edition. ISBN# 0470149779

Jack Olson (2003) Data Quality – the Accuracy Dimension. ISBN# 1558608915

Dan Linstedt, Kent Graziano & Hans Hultgren (2009) The Business of Data Vault Modeling. ISBN# 9781435719149

Bill Inmon (1999) Building the Operational Data Store, 2nd Edition. ISBN# 047132888X

Carnegie Mellon (2010), CMMI for development, version 1.3

Financial Services Authority (2011)  
[http://www.fsa.gov.uk/pubs/international/imap\\_final.pdf](http://www.fsa.gov.uk/pubs/international/imap_final.pdf)

Tom Breur (2010) Data Quality: Cost or Profit Center?  
<http://www.beyenetwork.be/channels/5089/view/12957>

Jerry Weinberg (1991) Quality Software Management, Vol. 1: Systems Thinking. ISBN# 0932633226

Benoit Mandelbrot (1977) Fractals: Form, Chance, and Dimension. ISBN# 0716704730

Barry Devlin (1988), An Architecture for a Business and Information System, IBM systems journal, vol. 27, no.1

Dan Linstedt (2011), Supercharge your Data Warehouse. ISBN# 97809866757-1-3

