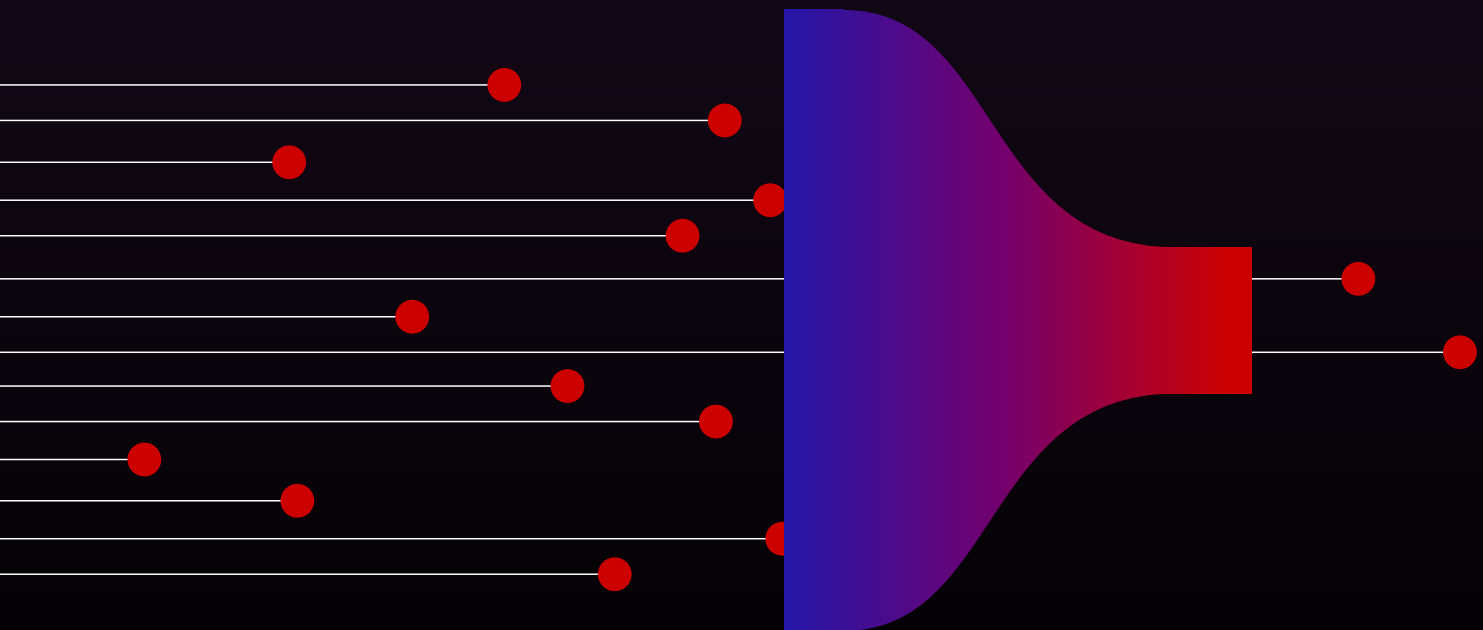
VAULTSPEED

Data lake vs data warehouse. Is there a middle ground?

Michael Olschimke
CEO Scalefree



Introduction

Business is being transformed by data: today's decision-makers have access to vast amounts of data on which to base their strategic actions. But how can you choose the best solution? Data warehouse or data lakes? Do you really have to sacrifice speed and agility for information reliability? This article will guide you through your options.

The traditional **data warehouse** used by businesses to turn operational raw data into useful information is often limited in its agility: it just takes too much time to process data and deliver it as actionable information.

The alternative, self-service business intelligence (BI) for **ad hoc analytics**, using data lakes for example, promises much faster information delivery, and potentially more insightful information. However, this agility comes at a price: data processing is not standardized so the resulting information is often inconsistent, leading to conflicting answers and insights.

Data Vault 2.0 offers a happy medium; a middle ground between traditional data warehousing and ad hoc data analytics by providing aspects of data warehousing – notably structured data processing – and aspects of ad hoc analytics and self-service.

About the author

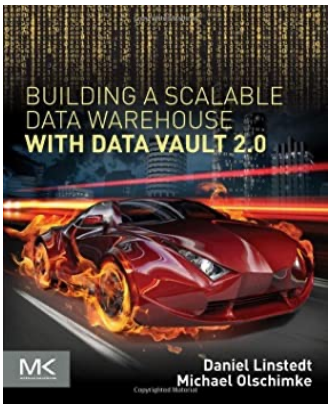
SCALEFREE

Michael Olschimke
CEO Scalefree International GmbH

Michael has more than 20 years' experience in Information Technology. For the past eight years he has specialized in Business Intelligence topics such as OLAP, Dimensional Modelling and Data Mining. He has consulted for a number of clients in the automotive, insurance, banking and non-profit fields.

His work includes research on massively parallel processing (MPP) systems for building artificial intelligence (AI) systems for the analysis of unstructured data. He co-authored the book "Building a scalable data warehouse with Data Vault 2.0," which explains the concepts of Data Vault 2.0, a methodology to deliver high-performance, next-generation data warehouses.

Michael holds a Master of Science in Information Systems from Santa Clara University in Silicon Valley, California. Michael is co-founder and one of the Chief Executive Officers (CEO) of Scalefree, where he is responsible for the business direction of the company.



Turning data into information

The key purpose of a data warehouse and many other analytical platforms is to transform raw data – such as detailed customer records, purchase orders, legal agreement documents, etc. – into useful information. Raw data originates from operational source systems and is often scattered across multiple source systems.

Data structures are defined by the source system and could have any form (structured, semi-structured or unstructured).

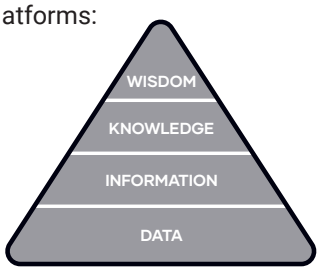
- **Structured data** is often stored in relational databases and consists of clearly defined attributes that describe the relations (e.g. customers and their attributes, such as first name and last name, birth date, etc.). These data structures are related to each other: for example when customer purchase patterns are also stored in the database, they refer back to the customer record.
- **Semi-structured data** also has a structure, but it is not a fixed one. Instead, each data record could be described by a different structure. A web page is a good example: every web page has certain metadata (e.g. title and keywords) and uses (X)HTML markup language to describe the formatting of the page. These (X)HTML tags define the structure of the data.
- **Unstructured data** includes plain text documents, images, and video files. They also have a structure, but it is hidden until uncovered by additional data processing, for example, data mining or feature extraction.

Information is derived from the raw data and used in the decision-making process. For example, a report on customer lifetime values could be used to provide an insight into the importance of a customer. Information should be, primarily:

- useful to the business
- actionable
- trustworthy.

Information is generated from the raw data by transforming and preprocessing the data into the final information. For example, the customer lifetime value might be derived from the customer purchase patterns in the raw data example above. The structure of the information is typically defined by the information user in accordance with the use case.

The following pyramid, known as the DIKW model, shows the relationship between raw data and information, both relevant for analytical platforms:



DIKW pyramid by Maslow

The next layers, knowledge and wisdom, are achieved by the user and human capabilities.

The problem is that there is a gap between the raw data from the source systems and the information resulting from this data: business users often need information for monitoring and improving business processes and understanding how the business operates. The data should describe business objects and transactions processed in these business processes. But in some cases, data is not available at all, or not available in the desired quality.

To bridge the gap, transformation rules – such as business rules and data cleansing operations – are applied to turn the raw data into the desired information. For example, a business rule could be applied to transform foreign currency amounts into the user's preferred currency before aggregating the customer purchase patterns.

Data cleansing includes the de-duplication of customer records and product records to be able to merge the purchase patterns for the de-duplicated customer and product records. The resulting information (and sometimes the raw data) is then provided to the organizational decision-makers to support their decision-making process. They rely on the information being good, i.e., relevant, timely, accurate, formatted, complete and accessible.

In short, data transformation supports organization transformation, by helping businesses make decisions that are data driven, based on facts, not gut feeling.

In the past, several vendors have made 'silver-bullet' claims, promising a solution that would solve all potential problems at once and forever. Such solutions include the traditional data warehouse or self-service BI, for example in conjunction with a data lake.

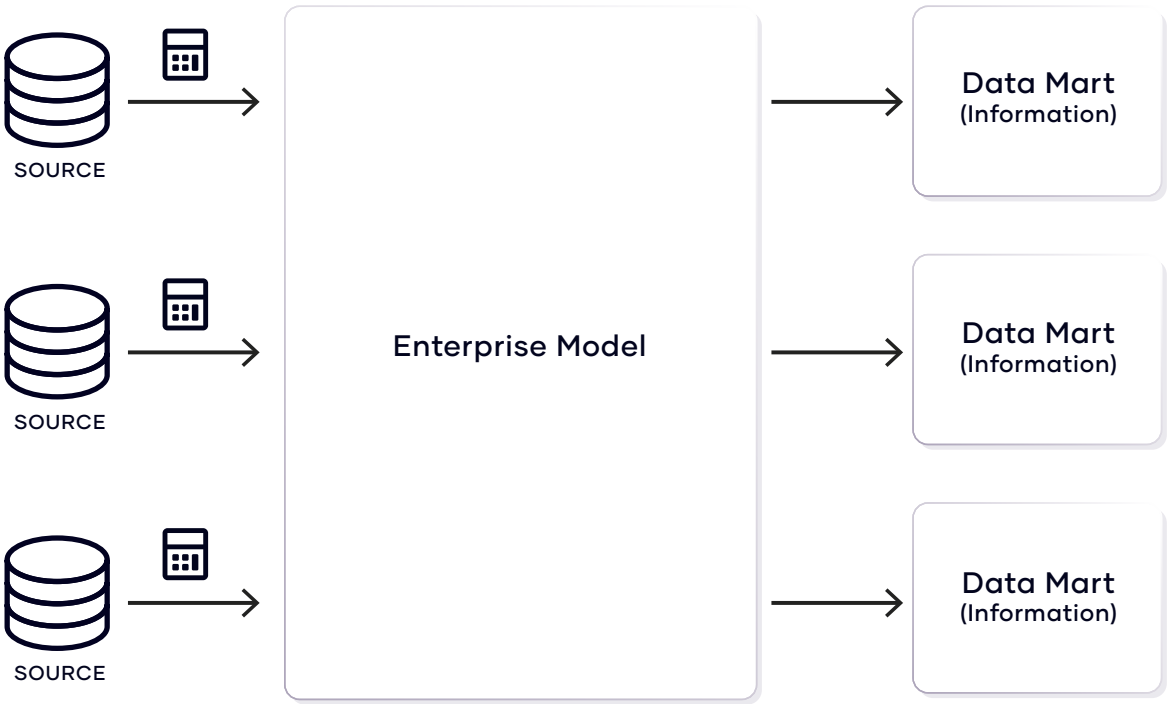
The traditional data warehouse

A traditional data warehouse is a popular design choice in enterprise data modeling. In a traditional data warehouse solution, the 'enterprise model' or 'information model' describes the business objects and their attributes and the relationships between business objects as they are understood by business.

However, there are several issues with this approach. First, there is a gap between the business expectation and the actual raw data. To bridge the gap, the raw data must first be transformed into information. This will work initially but will go wrong when the enterprise model changes, for example because the definition of a customer changes and therefore affects the lifetime value. On top of that, raw data from additional source systems is required, adding to the overall complexity.

The problem with the enterprise model is that it only models the current state of the enterprise, but due to the business changes, that's like modeling a moving target. The information designer is unable to keep up with the business changes in the model. No matter how fast these changes are implemented in the model, it's always behind reality.

In addition to only modeling the current state, the enterprise model only models a single state. The data warehouse modeler tailors (i.e., limits) the solution to one business perspective, e.g., that of headquarters. But there are many more business perspectives, for example those of subsidiaries, historical perspectives, and external ones from regulators. In traditional data warehousing, each of these specific data marts is merely a subset of the enterprise model in a different shape:



While the data mart follows a different modeling style than the enterprise model, it is based on the pre-processed data in the enterprise model.

Loading the data into the model, while a simple task initially, becomes a complicated task: how to deal with historical data and historical information structures? How to deal with cut-over dates where an old transformation rule should be applied until the cut-over date and a new transformation rule should be applied after it? How to deal with subsidiaries and the need for external reporting?

The resulting activity to fine-tune the model is often called ‘overmodeling’ as the value of the enterprise model (the promised silver bullet) is overstated and doesn’t reflect its real value to the business.

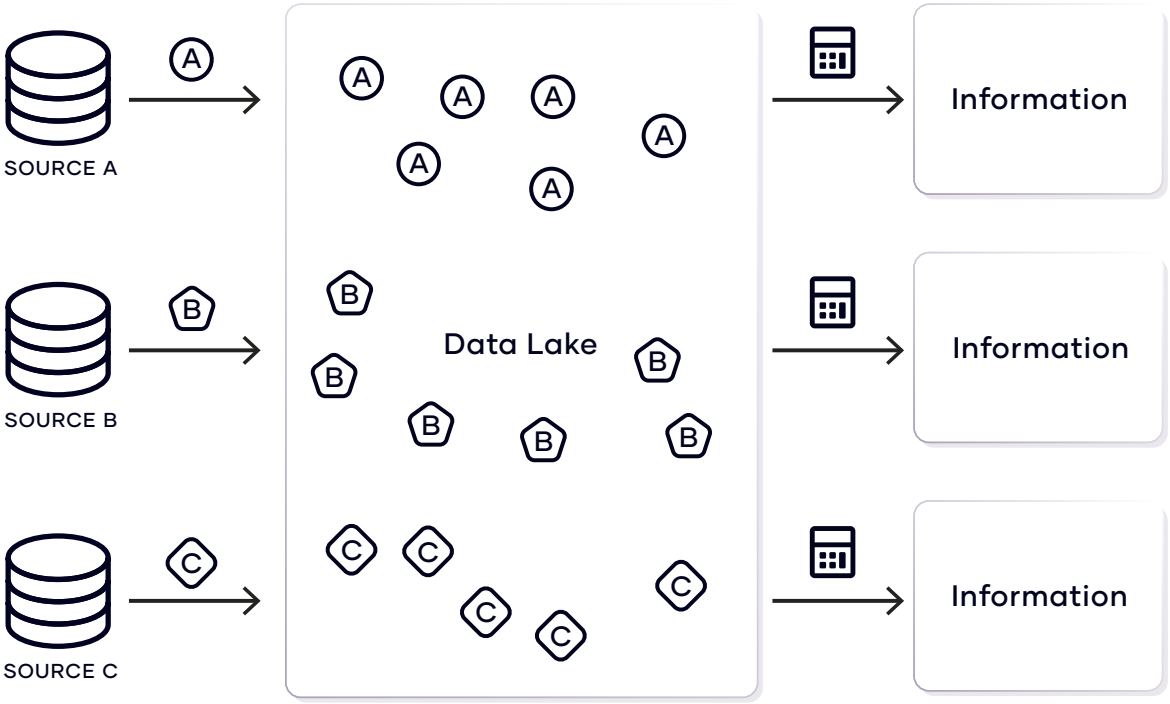
Self-service business intelligence

To overcome the issues of the traditional data warehouse, some organizations use a more laissez-faire, self-service approach to data analytics. Instead of modeling the data, the data is just stored (for example in a data lake), and the information model is only applied when information is read from the system during ad hoc analyses for business intelligence purposes. The data lake is a popular choice in such organizations, but other options are available. In many cases, they share the 'schema on read' concept. In this concept, the information model is applied during query time, typically with graphical tools or NoSQL engines that apply SQL-like queries against the data.

A typical use case involves direct access to the source system or a data lake and the processing of the data in a dashboarding solution with data preparation capabilities, or the data preparation in some Python scripts or hybrid SQL engines such as Apache Drill, Impala, or Polybase. The preferred choice depends on the technical capabilities of the users.

A typical requirement is driven by business school-trained managers: their desire is to make the right decisions based on the information. They also know what information they need based on well-established concepts. But then the chaos starts: every business is different, the data collected doesn't meet the expectations regarding completeness and quality, the transformation rules must be interpreted and implemented by developers and the information must be presented in an effective format.

The pendulum switches from an overmodeled solution to an undermodeled solution: because every user is left to apply the information schema, this often results in inconsistencies in the reports and dashboards, non-standardized data processing capabilities, and privacy and security issues.



In this approach, the data lake (used as the ad hoc analytical solution) contains the unmodified raw data, and transformation rules are performed when the target schema for the information is applied.

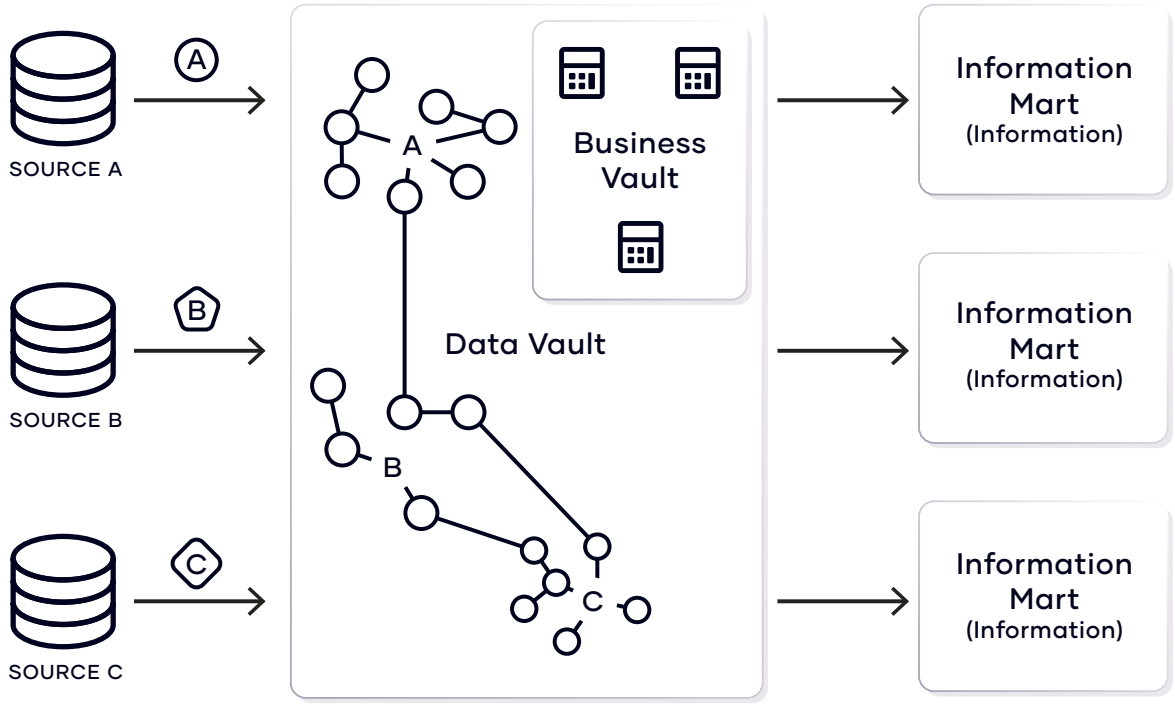
Looking at the two options, in a black and white world with no grey areas, what would you choose? In this world, only extreme solutions exist, and organizations must select one of them to take its advantages but also accept all the disadvantages. The first option – the traditional data warehouse enterprise model – sounds great but takes a lot of time and often fails to meet business expectations. The second option – self-service, data-lake BI – also sounds great and agile, but it often lacks true data management and leads to inconsistent reporting.

We believe that, in truth, there is no silver bullet. However, in the grey area between those extreme solutions, there is a balanced solution that offers the best of both worlds. A sophisticated solution for building powerful enterprise data warehouse solutions that are both agile and reliable.

Is Data Vault 2.0 the happy middle ground?

Data Vault 2.0 is a method of data warehousing but it follows agile, data-driven concepts such as the 'schema on read' concept.

For example, there are two layers in the Data Vault model: the Raw Data Vault and the Business Vault. The first layer captures the unmodified raw data while the second layer bridges the gap between the raw data and the information, which is delivered by the next layer, called the information mart. The information mart layer is equivalent to the data mart layer in traditional data warehousing. The changed name is due to a better distinction between raw data storage (in the Raw Data Vault) and information delivery (in the information mart).



In the Raw Data Vault layer, the incoming dataset, which is modeled after the source system, is broken into the foundational components (or elements) of all data: business keys, relationships between business keys and descriptive data. In the Data Vault model, this translates into hubs, links and satellites.

- **Hubs** store a distinct list of all business keys used to identify business objects across the enterprise.
- **Links** store a distinct list of all relationships between business keys.
- **Satellites** store (and version) the descriptive data about business keys and their relationships.

This model and the other accompanying pillars of Data Vault 2.0 (the architecture, the agile project methodology, and the implementation patterns) have had great success in the industry as a valuable alternative between the inflexible, traditional data warehouse and the self-service BI approach.

Is Data Vault 2.0 the silver bullet? Certainly not. Data Vault has been designed for enterprise data warehouse systems. While it works for data warehouse systems of any scale, from small installations to highly distributed petabyte-scale installations, its primary focus is on building those analytical systems. Occasionally, it has also been tried for more operational applications but it could not fully play out its advantages in such scenarios, for example regarding data integration across data from multiple source systems.

But in the analytical world, Data Vault has many advantages.

- Data can be integrated by business keys, which make it easy for business users to understand.
- The data is versioned and keeps track of all changes to the source data set.
- The model is easy to extend due to its simplicity and can be loaded using repeating patterns.
- The breakdown of the data structures into hubs, links, and satellites.

This breakdown into hubs, links and satellites, leads to many entities. While this sounds like a disadvantage, it is typically not: only a subset of the entities from the Raw Data Vault will be used by the information mart derived from the Data Vault model.

VaultSpeed advantages

The many entities and their loading procedures must be created and maintained. Changes to the source system should be integrated and the changes should often be deployed into the production EDW. This is where VaultSpeed comes into play: it helps to break down the incoming raw data into the target entities in the Raw Data Vault, manages the resulting entities, and generates both the SQL statements for the model and the loading procedures. If the structure of the incoming dataset changes (for example due to the business model change), the organization can speed up the resulting changes to the data warehouse model by leveraging VaultSpeed to generate the resulting code changes for the model and the loading procedures.

VaultSpeed also provides another advantage: standardized (and reviewed) patterns for the automation are included in the package. They make sure that all team members, both on-site or in remote locations, apply the Data Vault patterns in the same, standardized manner, thus decreasing any potential deviations from practice and, in turn, increasing the quality of the overall solution.

Besides these adjustments to the model and loading procedures of the data warehouse, VaultSpeed also integrates with deployment and orchestration tools and manages the required processes.

By doing so, VaultSpeed simplifies and speeds up the development and maintenance process of the data warehouse based on Data Vault 2.0 patterns. The included automation templates put Data Vault into practice and ensure that the generated artifacts meet the quality standards even with distributed teams where communication might be limited.

Conclusion

Data Vault 2.0 is a pragmatic, middle-ground solution between traditional, less-agile data warehousing on the one hand and self-service BI, with its laissez-faire approach and the resulting inconsistencies. It mitigates many of the disadvantages of these solutions while making the most of their advantages. It ensures that the agility of the project is maintained, and the data science team has enough freedom to build custom solutions but in a managed environment. VaultSpeed helps to find the right amount of modeling to build the enterprise data warehouse while avoiding overmodeling.

Visit our site
vaultspeed.com

Contact sales
sales@vaultspeed.com

Book a demo
vaultspeed.com/book-a-demo

Join our community
community.vaultspeed.com

Sluisstraat 79 03-01
3000 Leuven
Belgium

VAULTSPEED