VAULTSPEED

Why Data Vault is the best model for data warehouse automation

Michael Olschimke
CEO Scalefree

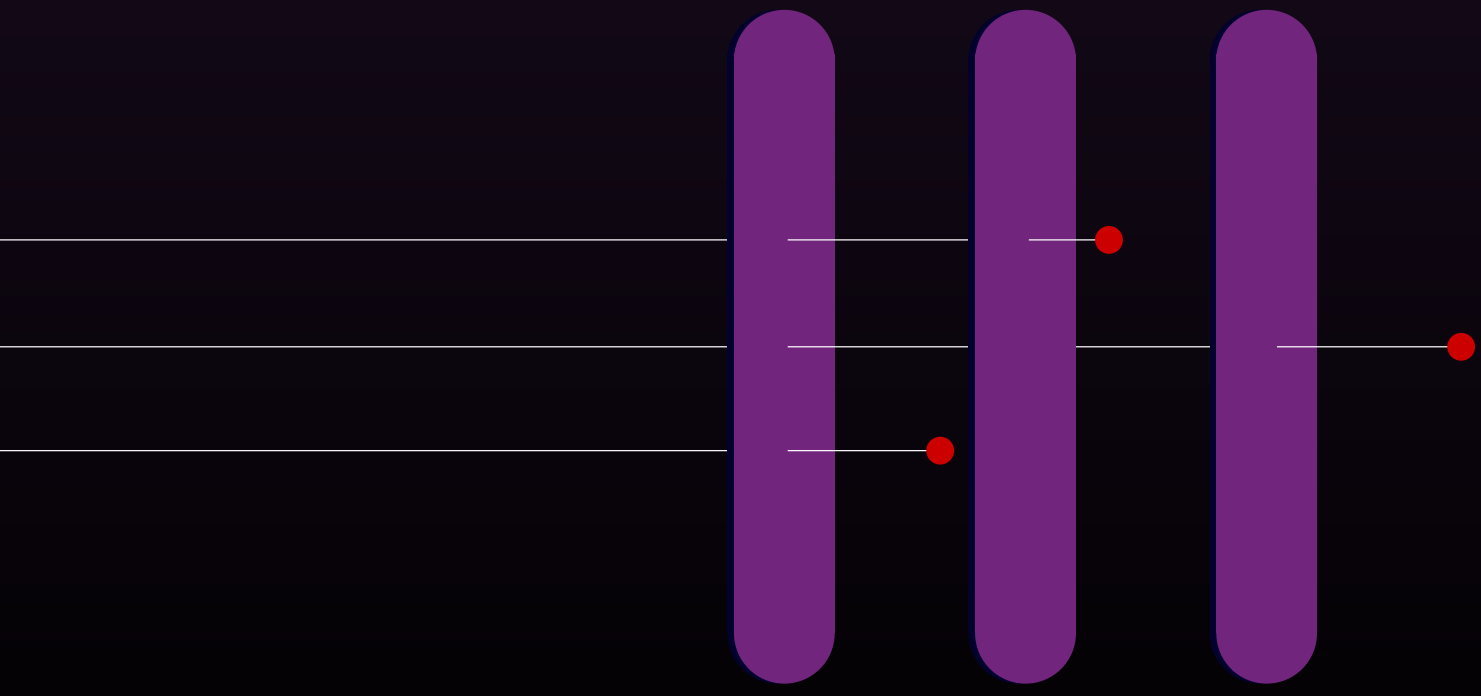


Table of Contents

Introduction

This paper describes and breaks down the pros and cons of different data modeling approaches. It explores the limitations of models such as Inmon and Kimball, and the comparative advantages of the Data Vault 2.0 model. It outlines how Data Vault 2.0 facilitates automation and speeds development and deployment through pattern-based structures and templates, explaining why it is the best approach for enterprise data warehouse (EDW) automation. It also describes how VaultSpeed tools support this model.

About the author

SCALEFREE

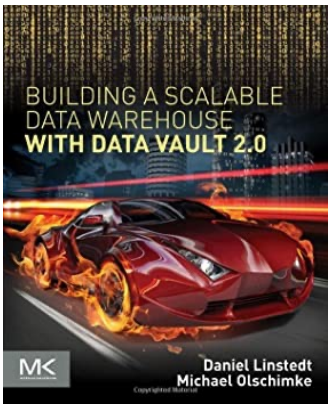
Michael Olschimke

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.



1. Overview of data warehouse approaches

Data warehouse basics

EDW systems are intended to process source data into useful information. The target model for the information is typically defined by the business user and is often a dimensional model, such as a star schema or snowflake schema, with facts and their dimensions. Business users select the model according to their information needs, for example when using a dashboard application.

The data model, however, is defined by the data warehouse architects and often selected based on traditional design decisions, known as bottom-up and top-down designs. These traditional models are limited in their automation capabilities.

Automation accelerates and standardizes the loading and modeling of the source data in the data warehouse data layer, which serves as a foundation for the next layer in the enterprise data warehouse, the information layer. Standardization makes it possible to deal with analysis over time and views from different data scientists.

More modern approaches have been designed for the automation of the enterprise data warehouse and have had great success in the industry.

Traditional data warehouse modeling

There are two popular traditional options for modeling the data warehouse:

- the Inmon data warehouse data model
- the Kimball federated star schema.

Both options use a data mart for information delivery. Often, these data marts (also known as information marts in other architectures) are modeled using dimensional models, such as star schemas or snowflake schemas.

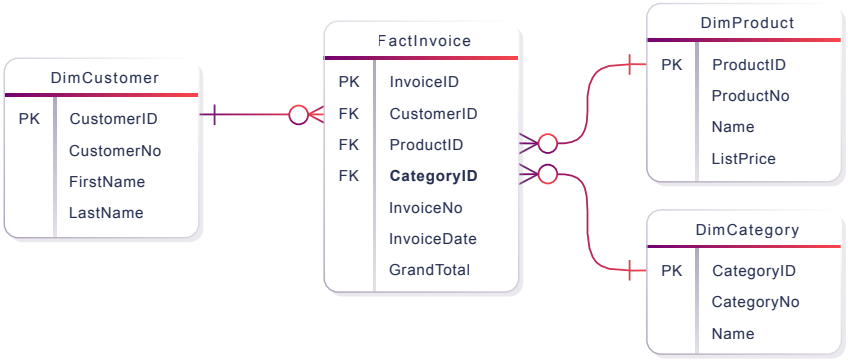
Star and snowflake schemas organize around fact entities that provide information that can be aggregated, for example about retail transactions, call records, flights and so on. These transactions or events provide measure values, such as, respectively, revenues, call durations and flight durations. They also provide dimensional attributes or dimension references that can be used to break down the aggregated measure values by dimensions such as products, customers or airport locations. The facts, their measures and the dimensions are defined by the business user in an information requirement.

The difference between star and snowflake schemas is quite simple. In a star schema, only fact entities can reference dimension entities, while in a snowflake schema, dimensions can also reference other dimensions.

For example, consider a fact entity that refers to products and their categories.

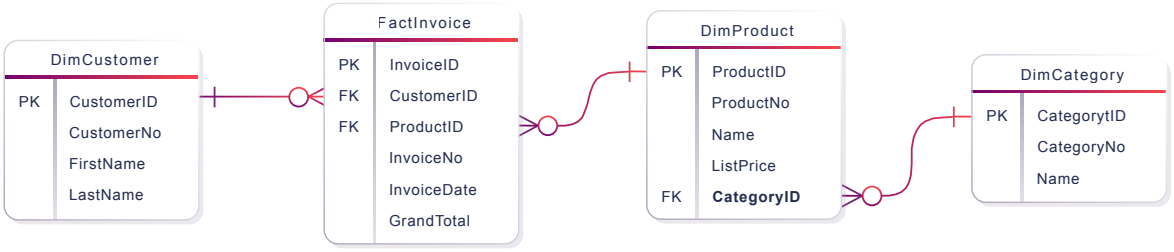
In a star schema, the fact entity references both the product dimension and the product category dimension.

Star Schema



In a snowflake schema, the fact entity references only the product dimension. The product dimension references the product category dimension.

Snowflake Schema

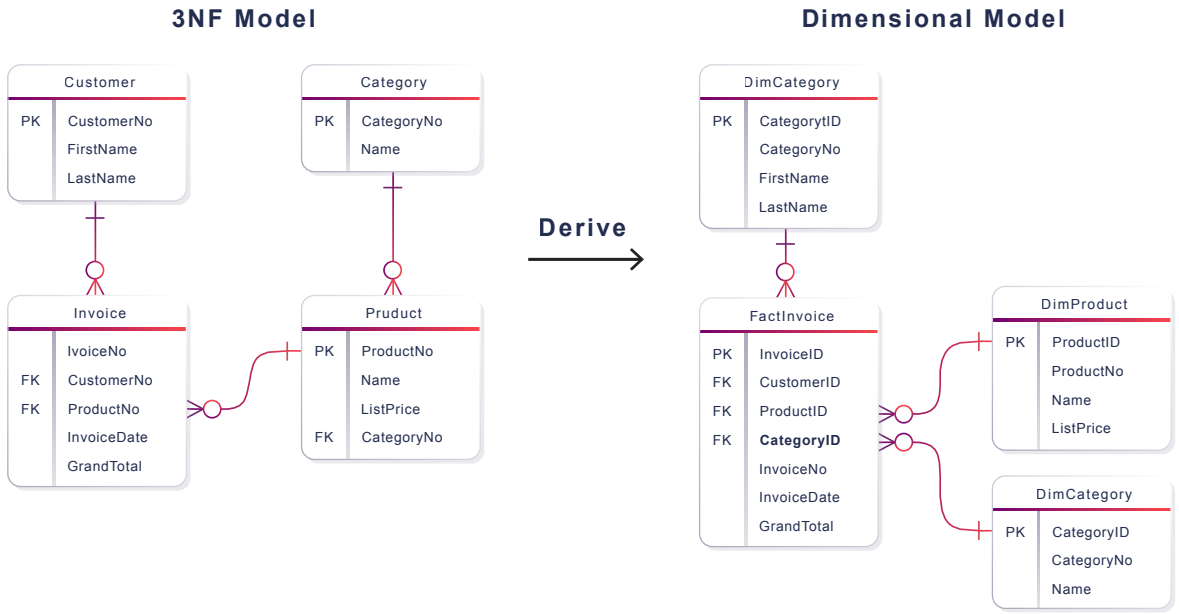


Other models for the data mart are possible. For example, a commonly-used model is a fully denormalized fact entity that contains all dimensional attributes; it's an easy-to-use model, especially with spreadsheet-like applications.

Inmon data warehouse data model

Inmon follows a top-down approach to data warehousing.

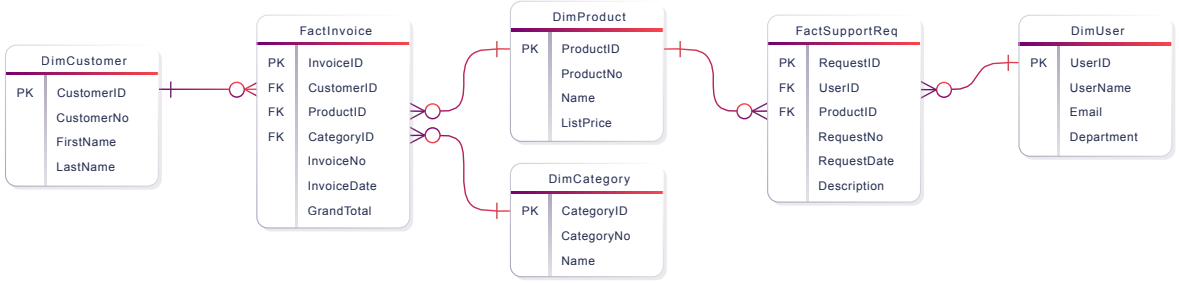
- The first component modeled is the core data warehouse model that describes the enterprise based on operational corporate data.
- The data warehouse model is modeled in a similar fashion, using a normalized data model in the third normal form (3NF), but adding effectivity timelines to the data model for data historization purposes.
- Once that data model has been established, individual data marts for reporting purposes are derived – and sourced – from the data warehouse model.



Kimball federated star schema

In the Kimball schema, the modeling starts with the individual data mart in a bottom-up approach.

- The individual data marts use conformed dimensions such as customer and product dimensions.
- These dimensions are used by multiple data marts and integrate the data marts into the so-called federated star schema via the information bus.
- The sum of all data marts is then considered the data warehouse.



In addition to these popular options for traditional data warehousing, organizations have adopted other approaches; in practice, they are often just an unmanaged, rampant mix of the above options with additional, free-style modeled entities.

Issues with third normal form

As already stated, Inmon's approach is based on an enterprise model that is modeled after the operational corporate data in 3NF with effectivity timelines. This model can handle a high volume of tightly integrated data in a highly structured data model with many-to-many linkages. It is relatively easy to extend and can process near real-time loads.

However, the 3NF model was originally not intended for use in data warehousing. It was modified for this new purpose through the addition of effectivity timelines. But the added timeline also adds additional complexities to the data model, especially the potential of having joins across timelines.

But there are bigger issues. Notably, when loading the model, the entities must be loaded in a certain order, driven by the business, creating dependencies that lead to cascading changes if any part needs to be modified. These dependencies can become quite a burden, particularly in larger enterprise models.

For example, the organization must be loaded before loading employees, because the organization is referenced by the employee to indicate the employer. If the organization is not known by the organizational entity, loading the employee will fail. If the organization entity needs to be modified, the employee entity must first be at least reviewed and tested, but potentially modified as well. But if the employee entity is touched, the salary payments entity must be reviewed, and so on.

As a result, most organizations try to prevent modifications to the model in the first place, but that often leads to a big bang approach to the enterprise model: first, build the whole enterprise model before implementing reports on top. This approach is not very conducive to agile development and it becomes difficult to model the enterprise as the enterprise is constantly changing.

Another issue with the 3NF model is that it is not well supported by business intelligence frontends or is hard to use for business analysts.

Issues with the federated star schema

This is where the federated star schema comes into play. It's based on a simple star schema with facts and dimensions and therefore easy to use in business intelligence solutions. It easily supports multidimensional analysis of the information and provides subject-oriented answers with aggregation points included.

Another advantage is that its subject orientation can facilitate development and deployment. Business users can easily define an information requirement; development can focus on this specific requirement without needing to analyze the wider enterprise model with data not relevant for this requirement.

But the federated star schema also has its shortcomings. While the individual star schemas are integrated using conformed dimensions, this doesn't replace an integrated enterprise information model.

Moreover, the model is driven by the information requirements of the business user. All new information is added by additional, subject-oriented star schemas. But in many cases, information requirements are not so clear, especially where business users want to broadly explore the data available from the source systems. In this case, business users would like to use a data model closer to the source schema, such as, operational data stores.

Another important shortcoming is the handling of real-time data. Because star schemas are often pre-aggregated, the ingestion and integration of real-time messages are limited due to the required aggregation on the fly. This issue is exaggerated when dimension records arrive late, for example when the web page visit is known before the details of the web page visitor arrive. Aggregations (for example by visitor's country, or page content) in this case cannot be performed (yet). This leads to delays in information delivery or worse.

It should be clear by now that the star schema is great for information delivery, as it was designed for this purpose. However, the data warehousing aspect of the EDW is where the approach fails because it was never meant for that.

This is where Data Vault modeling as the alternative comes into play.

2. Data Vault 2.0 for Enterprise Data Warehousing

The Data Vault 2.0 System of Business Intelligence provides concepts and techniques to build EDW solutions: an agile methodology, a reference architecture, best practices implementation and an extensible model.

The model is based on three basic entity types: hubs, links and satellites. They represent the three foundational components of any data, including enterprise data — all data consists of business keys, relationships between business keys, and descriptive data. For example, an employee has an employee number, which is the business key. It is related to an employer, which is also identified by a business key (e.g., an organization number) and there is a relationship between the two. Both business keys and their relationship can consist of descriptive data, for example, the employee's first and last name, the employer's organizational name, and the beginning and end date of the employment describing the relationship.

In the Data Vault 2.0 model, hubs capture a distinct list of business keys, links capture distinct lists of relationships and satellites capture any change to the descriptive data in delta records.

The Data Vault 2.0 model is sophisticated. It consists of roughly 18 entity types, depending on the definition. However, all these entity types are derived from the base entities (hubs, links, satellites) described above. They are called special entity types and are used to better capture specific enterprise data, such as transactional records, business effectivity timelines, or multiactive data (for example when employees have multiple phone numbers assigned). Because the special entity types are based on the three base types, there exists a clear learning path to learn the complete model.

The Data Vault model is used not for information delivery but for the EDW layer, which is the data layer. The information delivery model, typically a star or snowflake dimensional model, is derived from the Data Vault model.

Designed for EDW

The reason why the Data Vault model is a good fit for EDW is quite simple: unlike the 3NF approach (designed for operational systems) or dimensional models (designed for information delivery), the Data Vault model was designed for enterprise data warehousing. As such, it is not a very good fit for operational systems, even though it has occasionally successfully been used for operational purposes.

Advantages of Data Vault 2.0 for EDW

- The Data Vault 2.0 model is simple by design and it's easy to extend the model over time. It was designed for agile environments where a project team could start with a small scope and gradually extend the data model by additional entities — sprint by sprint. Even a zero-change-impact scenario can be achieved whereby existing entities are never touched when adding new entities. This reduces the need for testing and for required changes to downstream entities, such as in the dimensional model. It also simplifies the deployment of these additions.
- The model can spread across multiple environments. Some data could live in an on-premise installation, while other data might be placed in a cloud environment. Multi-Cloud scenarios? Yes; possible and done many times. Different database technologies? The integration of NoSQL databases with relational database technologies? All done before, very successfully.
- The model can also integrate structured data, often found in operational source systems based on relational database technologies, semistructured data, such as JSON and XML data from web services, and unstructured data, such as PDF documents.
- Besides ingesting data via traditional batch loads, e.g. once a night, other loading cycles are also used. Data can be loaded as fast as it becomes available, including Change Data Capture (CDC) data delivery, near-realtime or data streaming. Instead of processing the data in different speed and batch layers (as in the popular Lambda architecture), the Data Vault 2.0 model integrates all data sources regardless of the loading cycle. Realtime data is seamlessly integrated with batch data and can be used in a combined dimensional model during information delivery.
- Data Vault has been developed for classified environments and so supports sophisticated security requirements. For example, the application of cell-level security, including both row-level and column-level security, is possible by design and even if the underlying database technology supports neither. Organizations use these features to implement solutions that include the application of an access control list (ACL) for dynamic adjustments of usage rights on the data set.
- The model supports the removal or modification of records from the EDW to meet privacy requirements. Records to be deleted or reduced could be consumer records as required by the EU General Data Protection Regulation (GDPR), patient data as in the US Health Insurance Portability and Accountability Act (HIPAA), and similar regulations. In the best case, organizations want to preserve some non-personal data, thus reducing records instead of fully deleting them. This is easily supported by the Data Vault model. Data can be divided by privacy classes and their individual data retention periods respected.

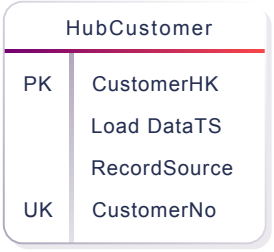
3. How Data Vault 2.0/VaultSpeed facilitate data warehouse automation

The most obvious advantage of automation is the increased productivity and the resulting higher agility of the project team. Instead of manually building Data Vault entities by hand, automation allows the teams to produce more entities in the same time span.

This section describes how VaultSpeed, based on Data Vault 2.0, offers extensive automation possibilities and has the potential for automating even more aspects of the EDW.

Standardized structures

An additional advantage of the Data Vault model is due to the standardized structures. All hubs look alike, as do all links and satellites.



A hub always consists of:

- a hash key for identification
- a load date to provide the technical historization
- a record source for debugging purposes, and
- the business key.

The business key is different per hub. For example, an employee would be identified by an employee number while a car would be identified by a vehicle identification number. It is also common to have multipart business keys, i.e., a business key that uses multiple columns.

The same is true for links and satellites. They also share a common structure but also have varying elements. For links, this is the number of hub references that implement the business key relationships. For satellites, it is the structure of the descriptive data. But in any case, there is a clear pattern behind these entities. Similar patterns are also seen with the loading procedures for these entities: all loading procedures for hubs are similar, etc.

Standardized structure is the basis for data warehouse automation. It's possible to use metadata-driven approaches to automating the EDW. This means that all the structures for the data layer can be automated, including the staging areas and the Data Vault entities. The automation capabilities are limited when it

comes to the automation of business logic, as in the so-called Business Vault, but there are also plenty of opportunities to automate some business rules.

The main limitation of the legacy approaches (Inmon and Kimball) regarding automation, described in the previous section of this white paper, lies in the fact that they apply business rules early in the architecture. Both models rely on business rules to load data from the source system into the data warehouse model. But automating business rules is a tough challenge and therefore moving business rules more downstream, as in Data Vault 2.0, enables the developers to automate.

Quality templates for quality, standardized output

The metadata describes the structure of the incoming data set, and identifies business keys, their relationships and descriptive data, to generate the CREATE TABLE and INSERT statements for all entities by using templates that provide the customizable target structure. With the use of templates, it is possible to generate the Data Vault model based on the same metadata to a variety of target databases and adjust the templates to the needs of the organization.

Another advantage is the standardization of the output. Manual development leads to inconsistencies in the model and the loading procedures. Using templates to standardize output removes some of the freedom of individual developers, preventing them from accidentally deviating from the standard development pattern. They can, however, still deviate intentionally, by adjusting the templates. Therefore, automation provides organizations with higher-quality solutions with less deviation from the standard procedures.

Good templates are therefore the foundation of a high-quality EDW. The better the quality of the templates and the metadata, the better the quality of the generated solution. The author of this white paper has reviewed the VaultSpeed patterns and the tool has passed the DVA vendor certification*.

** Certification program established by the Data Vault Alliance (DVA), the training and certification body of the Data Vault 2.0 inventor.*

Quality output through simplification

Another approach to increase the quality of the generated output is to reduce deviations from the standards by simplification. The simpler a process is in general, the fewer errors will be made. VaultSpeed is designed around this core principle and provides an easy-to-learn graphical user interface with a low learning curve. But because of the open templates in VaultSpeed Studio, the simple interface doesn't limit the solution.

The simplicity is achieved because of the tool's focus on Data Vault automation. Instead of providing a huge range of features, VaultSpeed focuses on core aspects of Data Vault automation. All other aspects that are required for building and maintaining an enterprise data warehouse are left to third-party tools, which VaultSpeed integrates with. This approach enables VaultSpeed to focus on their core tools and services, and is particularly useful for many customers who already have a scheduler in place, a CI/CD pipeline to be used, etc.

4. Conclusion

The pattern-based design of Data Vault 2.0, especially of the model, greatly supports the automation of the enterprise data warehouse and in turn, increases the productivity of development teams. The emergence of tools such as VaultSpeed has led to the increased adoption of Data Vault in organizations of all sizes, in all regions and across all industries.

With its unique low-code / no-code features, VaultSpeed has the potential of automating more aspects of the EDW, including parts of the business logic. This trend is already seen in new features such as VaultSpeed Studio where repetitive business logic, for example for currency conversions, can be automated.

With the automation capabilities of a modern tool, the extensibility of the Data Vault 2.0 model, and the ability to implement sophisticated and demanding user requirements, including security and privacy regulations, Data Vault 2.0 has a bright future for developing enterprise data warehouse solutions and we expect even more organizations to adopt the concept.

	Inmon	**Kimball**	**DV**
Model	3NF	Star schema	Data Vault
Original Purpose	Operational systems	Information delivery	Data warehousing
Simplicity	Medium	Simple	Medium
Scalability	Limited	Limited	Any volume, any speed
Multiple Environments	Limited	Limited	Span across different environments (cloud, on-premise)
Extensibility	Issues with dependencies	Easy	Easy
Agility	Initially low, but improves in the long run	Quick initial solutions, issues in the long run	Quick initial solutions, keeps agility in the long run
Support	Limited support by BI frontends	Preferred model by BI frontends	Not supported, but star schema can be derived easily
Real-time versus batch	Issues with real-time due to loading dependencies	No real time data; no aggregation on the fly	Supports any ingestion method (real-time, batch, etc.)
Automation compatibility	Limited	Limited	By design

Visit our site
vaultspeed.com

Contact sales
sales@vaultspeed.com

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

Join our community
community.vaultspeed.com

VAULTSPEED