VAULTSPEED

Automating common understanding

Accelerating the integration of multiple source systems into one Data Vault

Dirk Vermeiren, Michael Olschimke

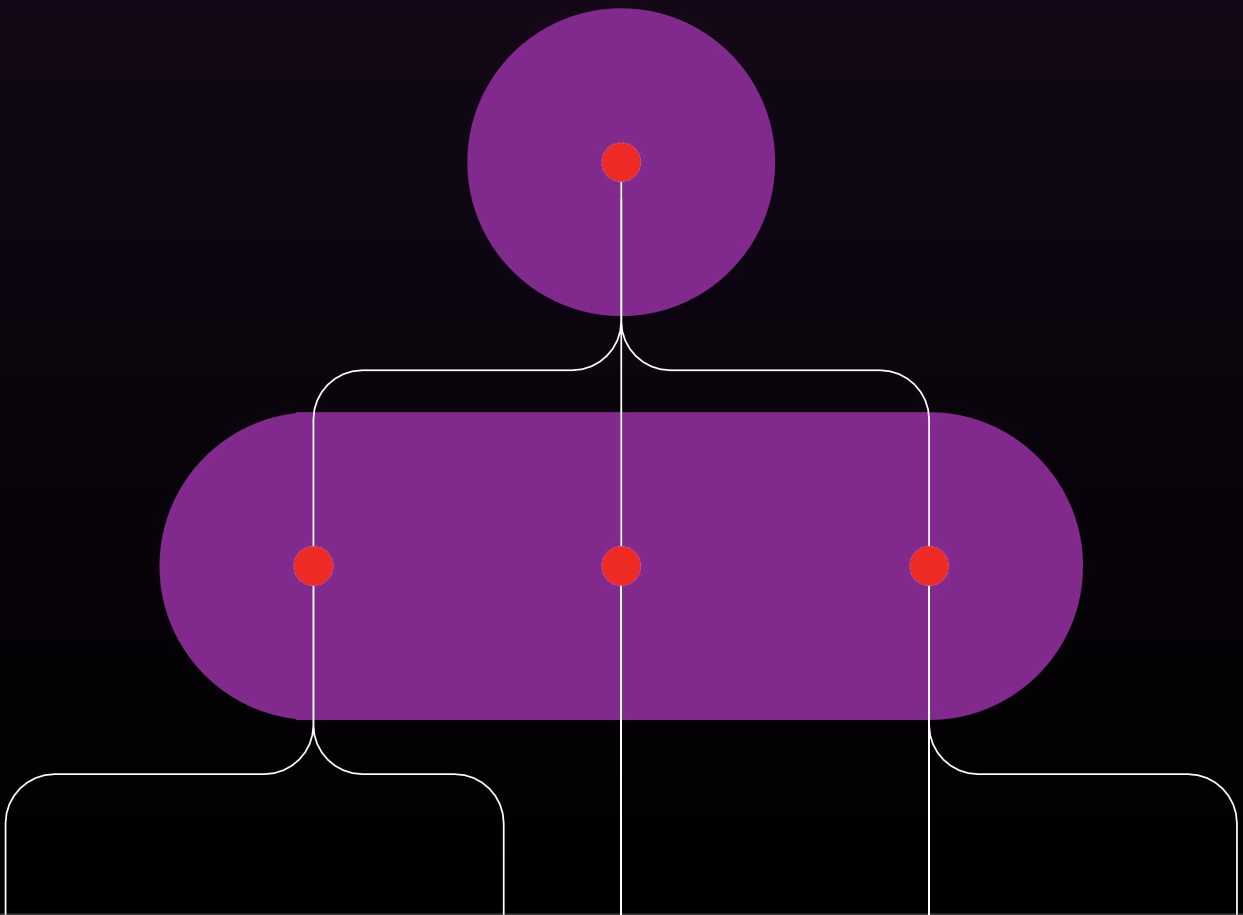


Table of Contents

This whitepaper was authored by Dirk Vermeiren, Co-founder and CTO at VaultSpeed,
and co-authored by Michael Olschimke, CEO at Scalefree International,
and Jonas De Keuster, VP Product Marketing at VaultSpeed.

Welcome to the real world

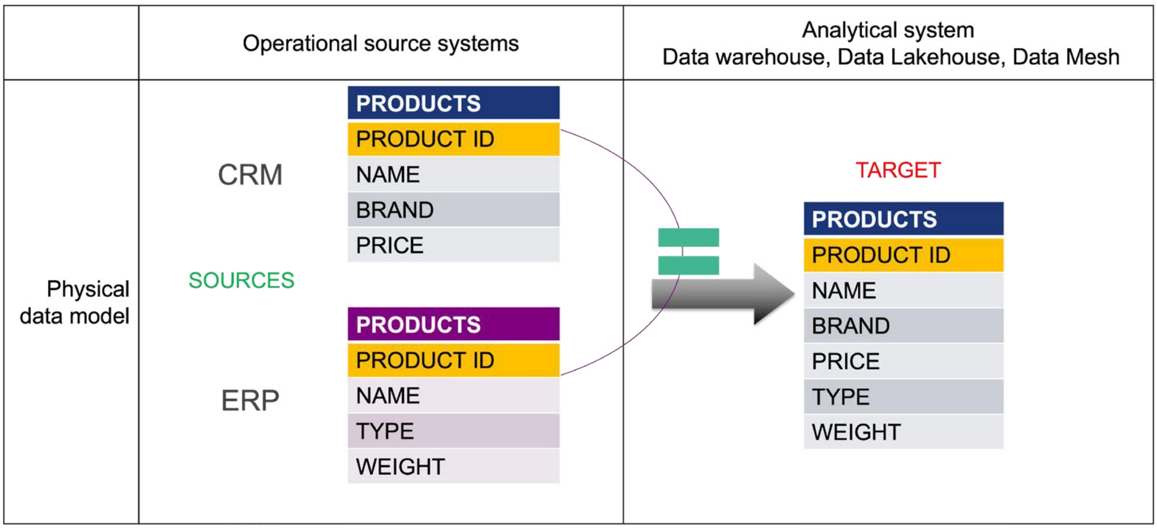
In corporate operations, a company's understanding of its business processes, products, and target audience often clashes with the varied terminologies used across different business lines. These differences in language—combined with the natural evolution of business processes over time—create significant challenges.

This lack of alignment can hinder data integration, delay decision-making, and ultimately lead to missed business opportunities.

A common misconception continues to persist: that a single entity owns both operational and analytical data models, and that these models remain fixed and unchanged.

To illustrate the real-world implications, imagine working for a vehicle dealership in a simplified, unchanging environment. You are solely responsible for developing all operational and analytical data systems. You've built two source systems—one for Customer Relationship Management (CRM) and another for Enterprise Resource Planning (ERP). Both systems include data related to the company's product offerings.

Now, you are asked to create an integrated solution to analyze the vehicle portfolio, including both bikes and cars. In this idealized world, where business definitions never shift, the product entity would be identically configured in both systems using the same unique identifier. This uniformity would allow for seamless data consolidation, making the integration process both simple and efficient.

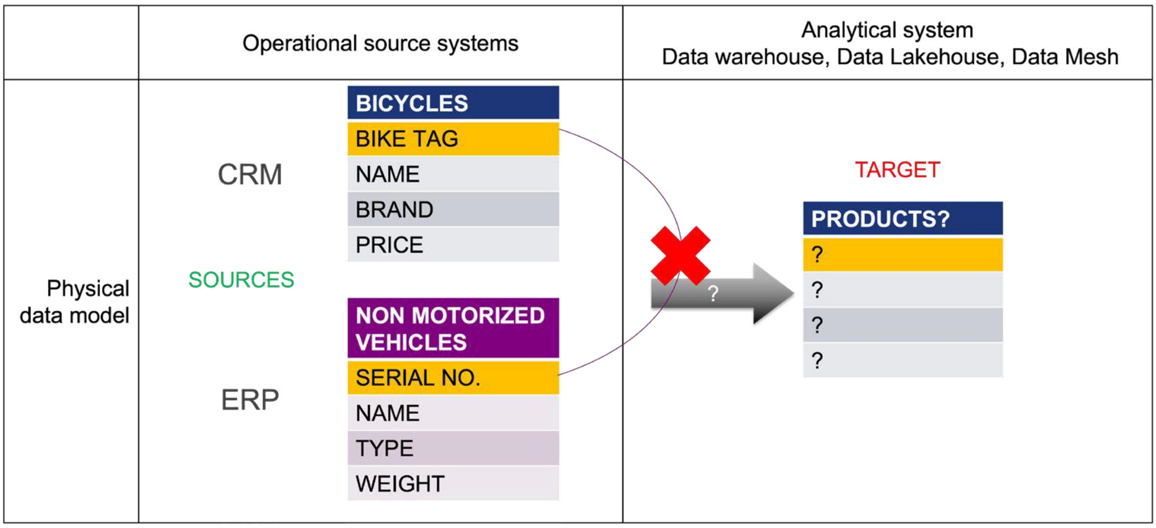


Picture 1, the ideal world

It’s time to acknowledge that perfect conditions don’t exist. Why? Because you're not alone in building and maintaining operational systems. These systems are often developed by different individuals or sourced from external vendors. As time passes, businesses evolve—what is valuable today may be irrelevant tomorrow. Growth brings added complexity, and frequent team turnover further complicates things by introducing shifting perspectives within the data team.

Each system inevitably develops its own interpretation of business concepts, processes, and development styles.

As a result, integrating datasets—especially product data—has become increasingly challenging. For instance, as shown in Picture 2, the CRM system focuses on bicycles, while the ERP system categorizes products as non-motorized vehicles. These differences lead to varying levels of detail and inconsistent use of business keys, making integration anything but straightforward.



Picture 2, the real world

Coping with data complexity

Industry data models are not silver bullets

In an effort to standardize analytics, vendors once promoted pre-defined industry data models as comprehensive solutions for organizing, storing, and processing data across specific verticals—all the way down to the physical model level. The promise was appealing: a ready-made framework to accelerate integration and reduce complexity.

In practice, however, this one-size-fits-all approach quickly showed its limits. Source systems rarely aligned with the imposed structures, and many organizations operated in ways that deviated—if only slightly—from the model. These mismatches led to increased integration complexity and substantial transformation overhead.

Rather than simplifying data integration, industry models often made it more difficult. What was intended as a shortcut became an extra layer of work. The silver bullet never hit its mark.

Addressing data complexity through proper data modeling

To address the complexity of data integration, data modelers use a layered approach that includes conceptual, logical, and physical data models. In the vehicle dealership example, the focus was primarily on integration at the physical level. But in real-world scenarios, effective data modeling should begin at the conceptual level.

Conceptual data model

The conceptual model—often called the business model—offers a perspective that is easy for business stakeholders to understand. Its primary purpose is to define key entities, such as objects, concepts, or events, and map the relationships between them. Operating at the highest level of abstraction, the conceptual model avoids technical details. Instead, it focuses on aligning data design with overarching business concepts and requirements, without addressing how or where the data is stored or accessed.

Logical data model

The logical data model transitions from business context to structure, focusing on defining data elements in a technology-agnostic way. It specifies entities, attributes, relationships, and constraints, often visualized using tools like entity-relationship diagrams (ERDs). This model offers a clear, detailed view of how data elements relate to one another and establishes the rules that govern those relationships.

Physical data model

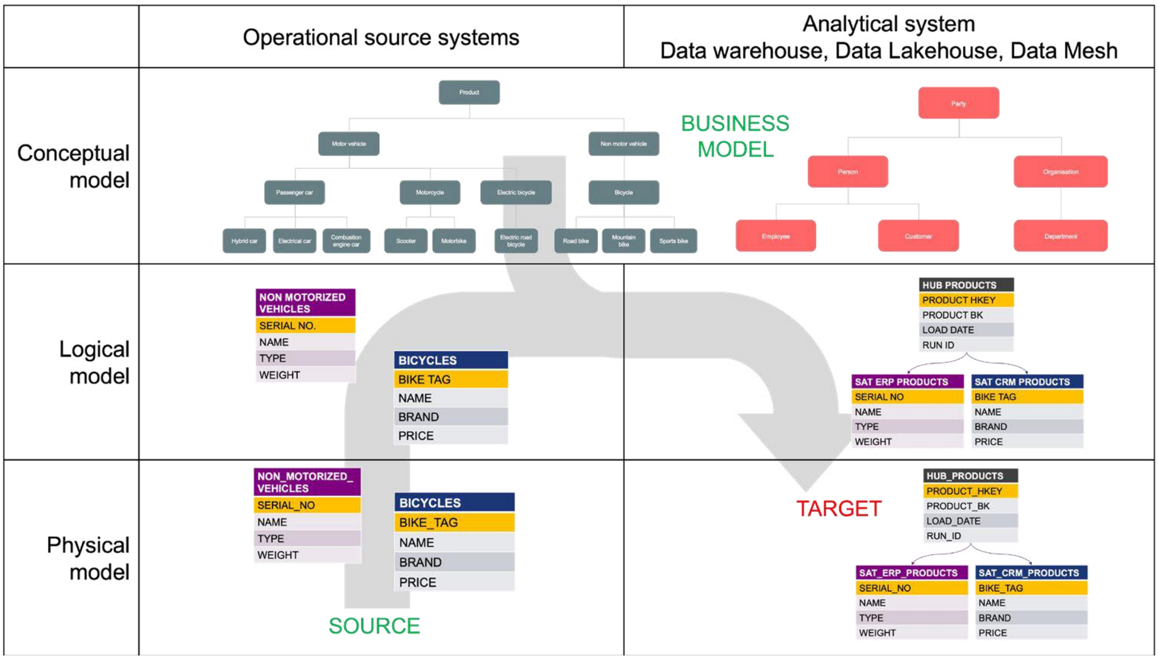
When designing a database, decisions about data storage, structure, and access become critical. This is where the physical data model plays its role. It addresses practical considerations like indexing, data types, and storage mechanisms to ensure optimal performance and efficiency. All these decisions must also comply with the limitations and features of the selected Database Management System (DBMS).

Together, the three levels of data modeling—conceptual, logical, and physical—help organizations manage the complexity of diverse datasets from an ever-expanding range of sources, each with its own taxonomy. Ideally, both operational and analytical systems are designed from a shared conceptual model. This ensures that while the logical and physical models may differ in granularity and use different business keys, they remain aligned to a common blueprint.

This alignment is essential when translating the physical data models of source systems into the physical model of the target analytical platform.

The diagram below illustrates the recommended approach for building data models in a data warehouse, data lakehouse, or data mesh architecture. It highlights two key inputs: the physical source model and the conceptual business model. The objective is to construct a well-aligned physical target model that supports accurate and efficient analytics

.



Picture 3

Another vital ingredient to deal with complexity and change is to automate the creation of data models and data runtime. This can significantly reduce the time it takes to accommodate new data requirements and mitigate the effects of change on the delivery of reliable, high-quality data. To achieve this automation, a clear definition of the relationship between physical source and target models is essential. We will delve into the topic of automation further ahead.

In conclusion, to create a data analysis system, such as a data warehouse, or data lakehouse, it is crucial to adhere to certain guidelines:

- Establish a clear and consistent conceptual data model that is easily understood by business stakeholders.
- Avoid creating or utilizing industry-specific data models at the physical level, as this may lead to issues when integrating different data sets.
- Avoid simply replicating the operational data models of your data sources, as this can lead to integration challenges.
- Embrace data automation whenever possible.

Balancing these seemingly contradictory constraints may appear challenging, but with the assistance of VaultSpeed, it becomes achievable. The VaultSpeed automation solution can translate any source data model into a comprehensive and integrated target data model, ensuring that your data analysis is accurate and efficient.

Tools to model common understanding

Create a shared language with taxonomies

In conceptual data modeling, taxonomies help organizations make sense of their data by providing structure and clarity. A taxonomy's main role is to identify, describe, categorize, and label objects based on shared characteristics. For example, bicycles, electric bicycles, and motorcycles all have two wheels, so they can be grouped under the category of "two-wheelers."

A taxonomy defines a formal structure of object classes within a specific domain. By organizing knowledge in a systematic way, it becomes easier to locate, relate, and manage data across systems.

A well-formed taxonomy follows several core principles:

- It uses a hierarchical structure, assigning names to objects based on their relationships to others.
- It defines clear membership properties between objects.
- It applies classification rules that are complete, consistent, and unambiguous.
- It ensures each object fits into one—and only one—category.
- It supports inheritance, allowing objects to carry properties from parent categories while adding their own specific attributes.

Taxonomy isn't just a theoretical concept—it plays a critical role in helping organizations understand their structure, manage data effectively, and support machine learning applications by revealing patterns in the data.

In practice, large organizations often operate with multiple overlapping taxonomies. For instance, a bicycle may be classified as a two-wheeler, a non-motorized vehicle, or a light vehicle. Each of these categorizations is valid, depending on the context. Different systems may apply classification based on factors like class, propulsion type, size, intended use, or design environment.

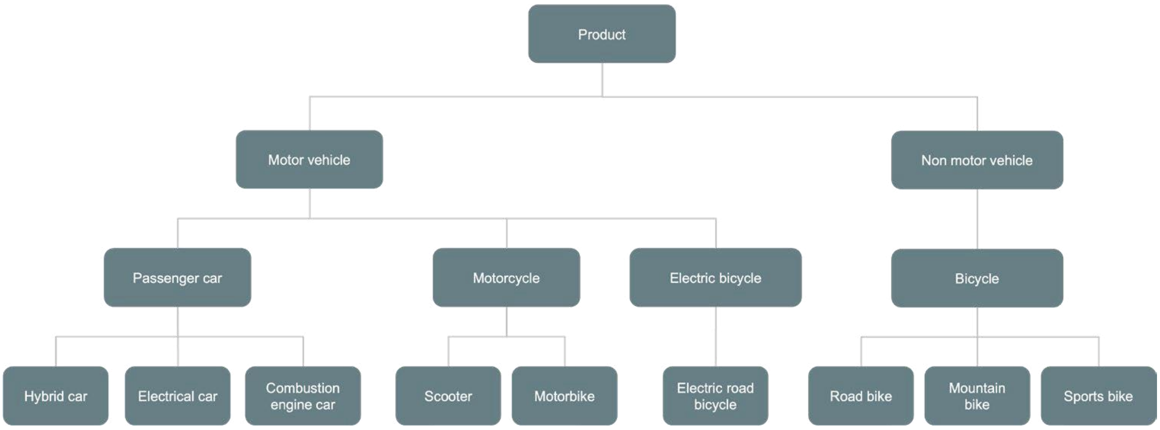
Determining the appropriate taxonomy for your business concepts is not a one-time task. It requires ongoing review and refinement to remain aligned with an evolving business landscape. This process should involve continuous input from those who work closest to the data—ensuring the taxonomy accurately reflects the organization's real-world structure and operations [1].

[1] Check out 'The Elephant in the Fridge' from John Giles on building business-centered models.

Once a shared language is established through a conceptual data model, the next step is to translate that model into a physical target model aligned with the appropriate level of the taxonomy. This physical model must accommodate different implementations across source systems, enabling integration through common business concepts defined in the taxonomy.

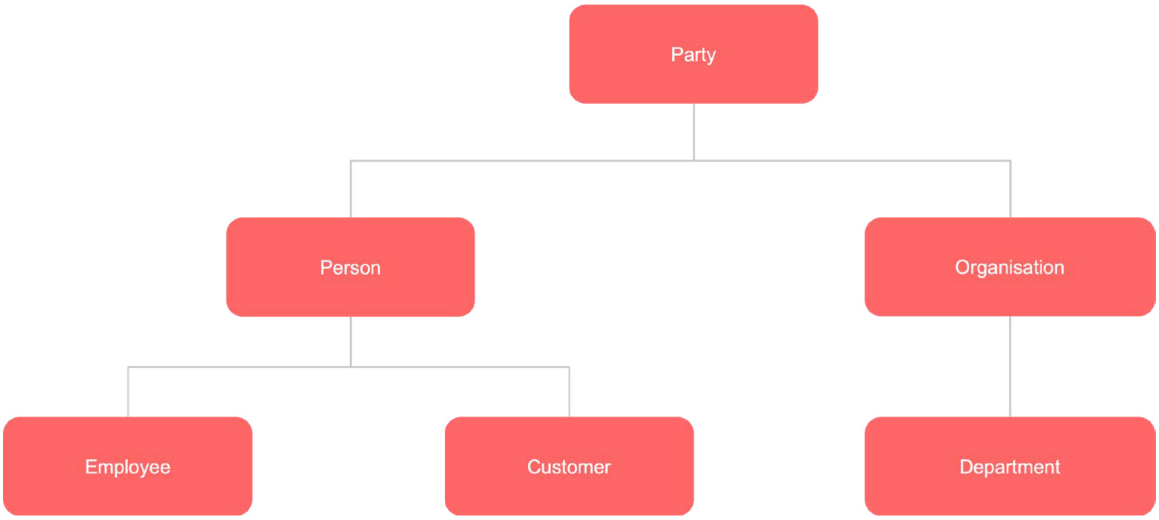
Consider the following example: A dealership that sells bicycles and other vehicles wants to build a data warehouse to better understand the purchasing behavior of its residential customers.

Here's how the company would represent its product range:



Picture 4

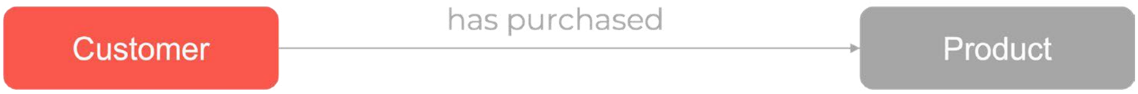
Which it sells to different parties:



Picture 5

The purchase relationship captures the interaction between customers and the products they buy. This conceptual model is deliberately aligned with business requirements, ensuring relevance for reporting and analysis. Business users have selected specific taxonomy levels to reflect this alignment: the product hierarchy centers on products, while the party hierarchy focuses on customers. These selected levels form what we refer to as the business model level.

Conceptual model at business model level



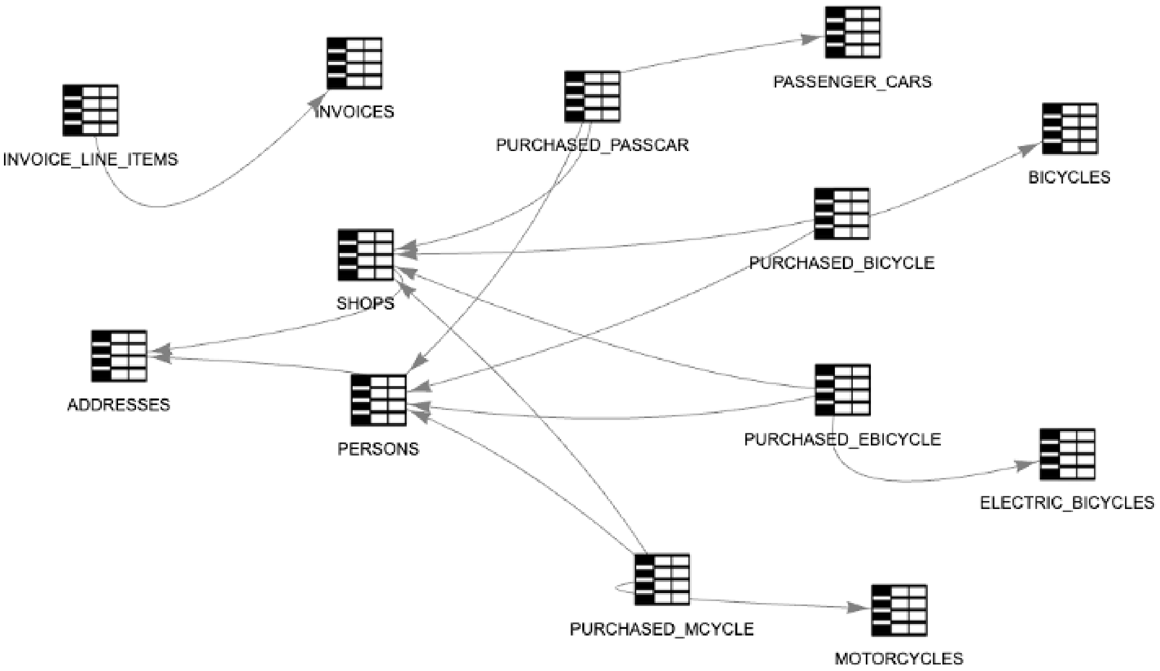
Picture 6

The master data supporting these taxonomies comes from multiple sources and is distributed across different taxonomy levels—this is known as the source model level. This master data may be maintained and replicated from a single authoritative source (a single-master system) or from several sources (a multi-master setup).

In our example, we're working with two distinct source systems from different subsidiaries. Each uses a product taxonomy at a different level of granularity. The dealership's goal is to analyze residential customer purchase behavior at the individual level.

Let's now examine the source data models and how each system implements product and party taxonomies.

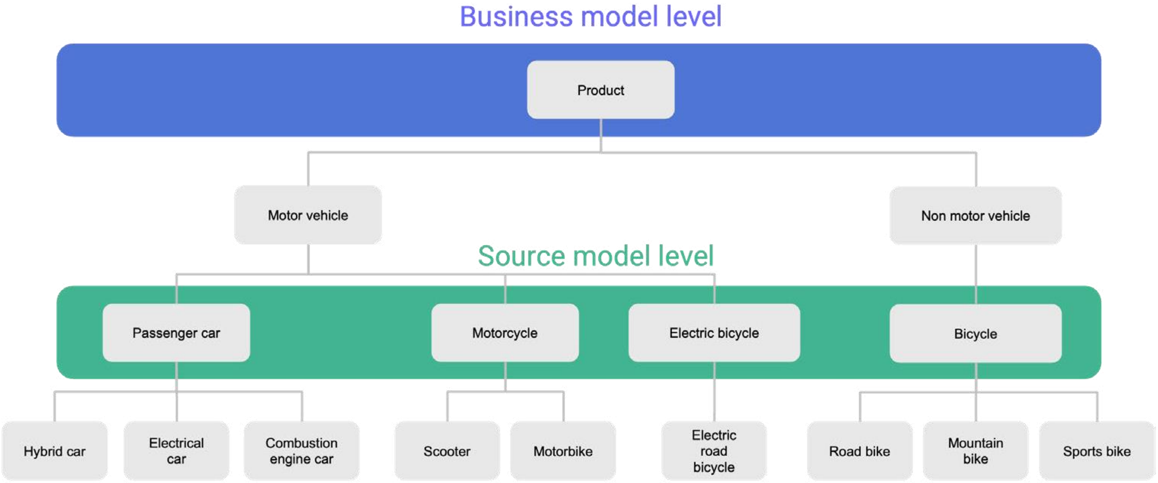
SRC1



Picture 7

The first source is a B2C (business-to-consumer) system that captures detailed information about customer purchases. It tracks who made the purchase, what was bought, and where the transaction took place. Additionally, it includes specific data related to invoicing processes.

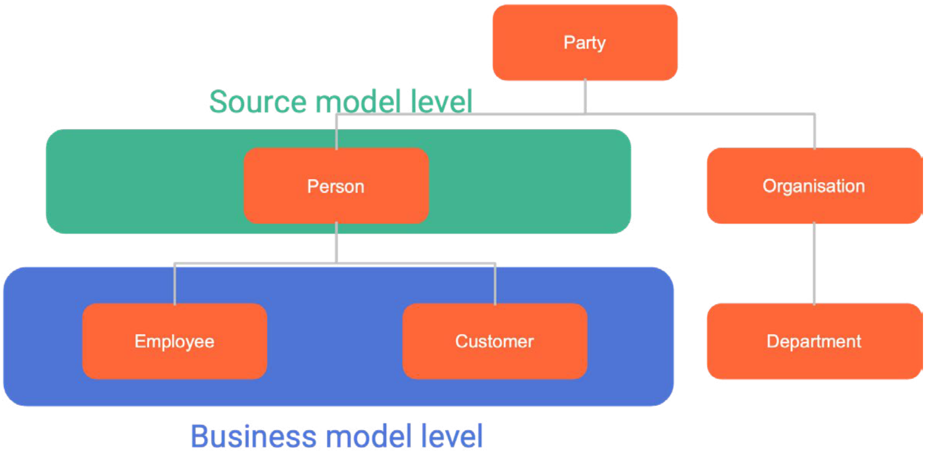
In this source model, the product taxonomy is implemented at the third level of granularity, as indicated in green:



Picture 8

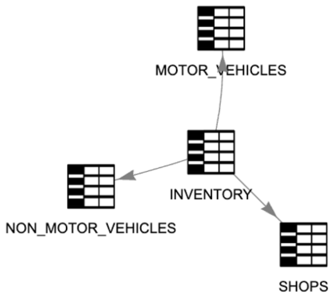
To support the business goal of reporting at level 1 of the product taxonomy (shown in blue), we need to *roll up* data from level 3 in the source. This requires moving upward through the taxonomy during the transformation from source to target.

Conversely, the party taxonomy requires a *drill-down*. The B2C source organizes party data at level 2, while the business model requires level 3 granularity. This mismatch must be carefully addressed when building the physical target data model to ensure accurate integration and alignment.



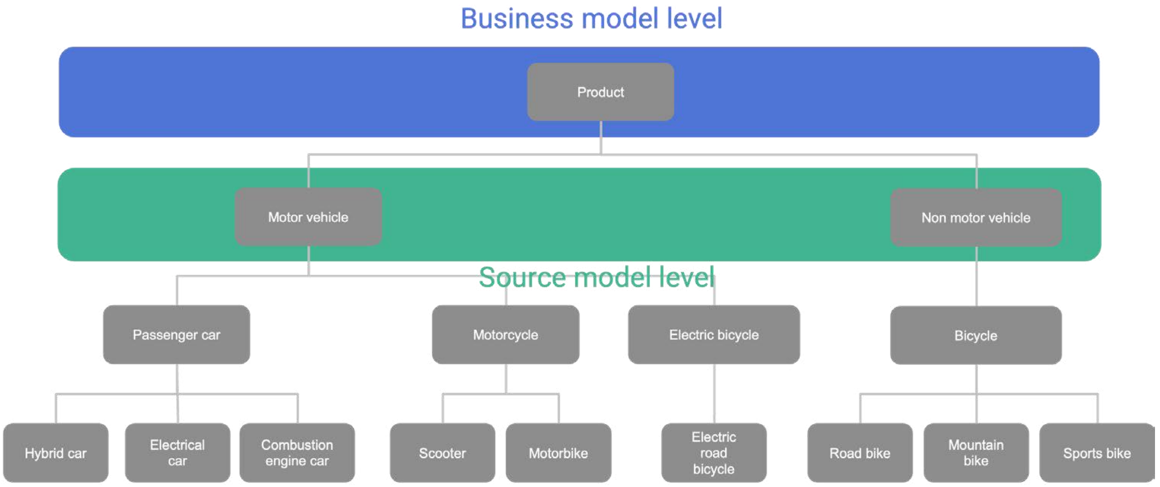
Picture 9

The second source is an ERP (Enterprise Resource Planning) system used for inventory tracking. It contains detailed information about products and stock levels across various retail locations—but it does not include customer data.



Picture 10

This source applies the product taxonomy at level 2, as shown in the following diagram. Once again, there's a mismatch between the source model level and the business model level that must be resolved during integration.



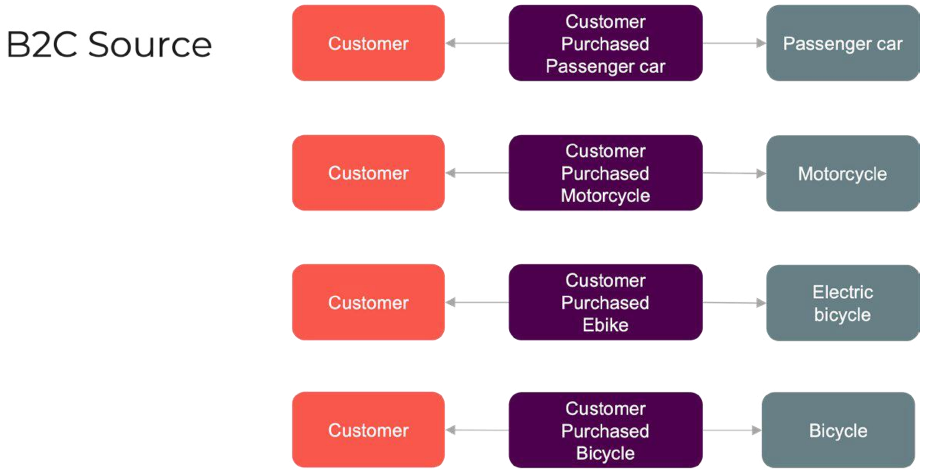
Picture 11

To enable meaningful reporting, it’s essential to establish a purchase relationship that links each customer to the products they’ve bought.

Within source 1, at level 3 of both the party and product taxonomies, four many-to-many relationships emerge (see Picture 12).

This dealership example has illustrated how a conceptual business model maps to the physical structures found in source systems. It also highlights the challenges of integrating these models into a unified data warehouse design.

In the next chapter, we'll explore how to accomplish that integration using Data Vault—a data modeling methodology purpose-built for this task. Data Vault stands out as the only approach that combines high levels of standardization with the flexibility needed to support evolving business needs.



Picture 12

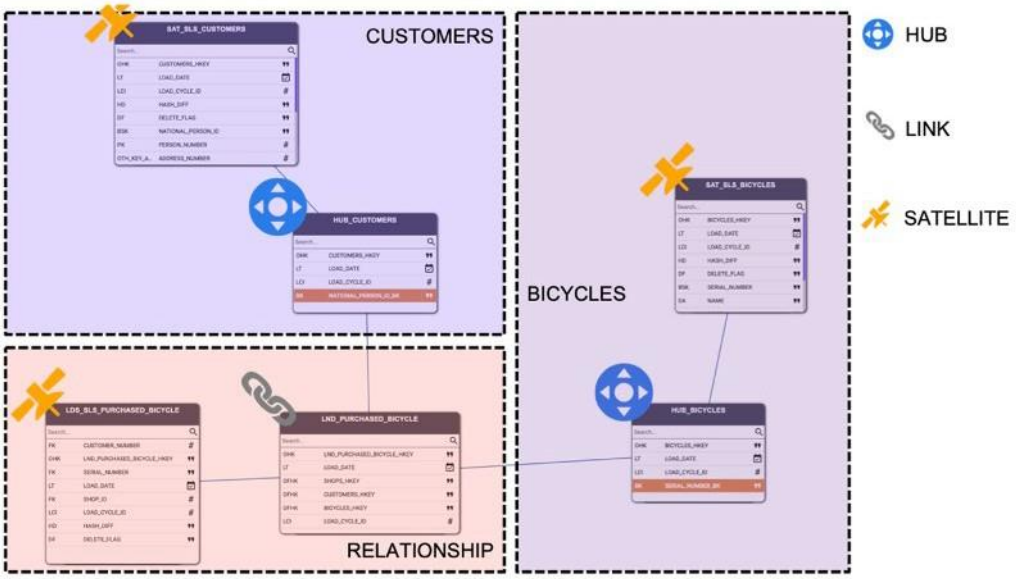
The optimal modeling language: Data Vault

Data Vault simplifies the physical integration of data from different taxonomy levels.

At the heart of Data Vault modeling is a powerful principle: multiple perspectives can be valid, even if they differ. The data itself remains objective. Rather than enforcing a single, uniform interpretation across the organization, Data Vault structures data into three universally understandable components:

- Hubs: represent business keys that identify core business objects, such as customers or products
- Links: define the relationships between those business objects
- Satellites: hold descriptive attributes related to hubs or links

Let's apply this model to our dealership example.



Picture 13

The diagram illustrates three core concepts: customers, bicycles, and the historical relationship between them. Each concept maps cleanly to Data Vault structures:

- Customers and bicycles are treated as business objects, captured by the HUB_CUSTOMER and HUB_BICYCLE, respectively.
- The purchase relationship between them is modeled as a link connecting the two hubs.
- Descriptive details, such as customer attributes, are stored in satellites, like SAT_SLS_CUSTOMERS.

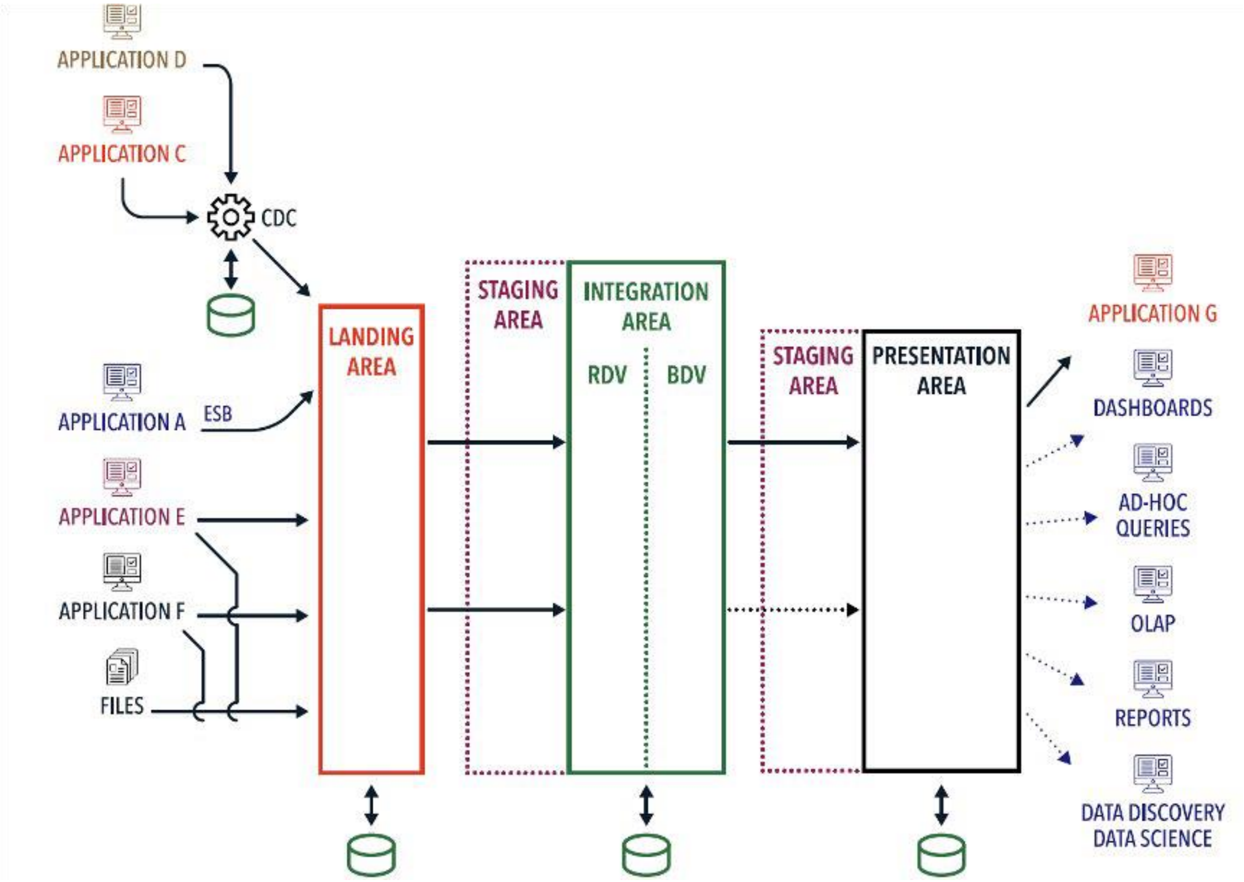
These three components—hubs, links, and satellites—enable the integration and harmonization of diverse data inputs, regardless of how they are classified in the source.

Importantly, Data Vault is also well equipped to handle multiple classifications of the same entity when they appear at different levels across various sources—a challenge that traditional models often struggle to address.

.

Data Vault architecture

Data Vault is more than just a modeling technique—it's also a reference architecture purpose-built to manage the integration, aggregation, and consolidation of diverse source data models, taxonomies, and naming conventions. It is specifically designed to address and simplify the inherent complexity of these tasks.



Picture 14

The first layer of the Data Vault architecture is the landing area, which captures data from source systems. Data can arrive through change data capture (CDC), real-time enterprise service buses (ESBs), direct database connections, or file-based ingestion.

Next is the integration area, which includes two internal layers:

- Raw Data Vault (RDV) – stores unmodified, historical data exactly as it comes from the source systems. This layer preserves source truth and serves as the factual record of what exists or has occurred.
- Business Data Vault (BDV) – contains pre-processed data and applies additional business logic. It aligns business keys and terms with the target model to ensure consistency, compliance, and semantic clarity across the integrated dataset.

The key distinction: RDV focuses on unfiltered, raw source data, while BDV refines and contextualizes that data for analytical use..

Both the Raw and Business Data Vault layers are modeled using standard Data Vault constructs: hubs, links, and satellites.

The third layer in the architecture is the presentation layer, which delivers information to downstream applications—such as dashboards, reports, and analytical tools—via information marts. These marts can reflect multiple business perspectives on the same data, all coexisting within the architecture. Each view is treated as a valid interpretation, tailored to specific reporting or analytical needs.

Importantly, the stability of artifacts varies across layers. While raw data in the RDV remains consistent over time, business perspectives in the BDV—complete with embedded rules—tend to evolve as the organization adapts to changing market conditions and requirements.

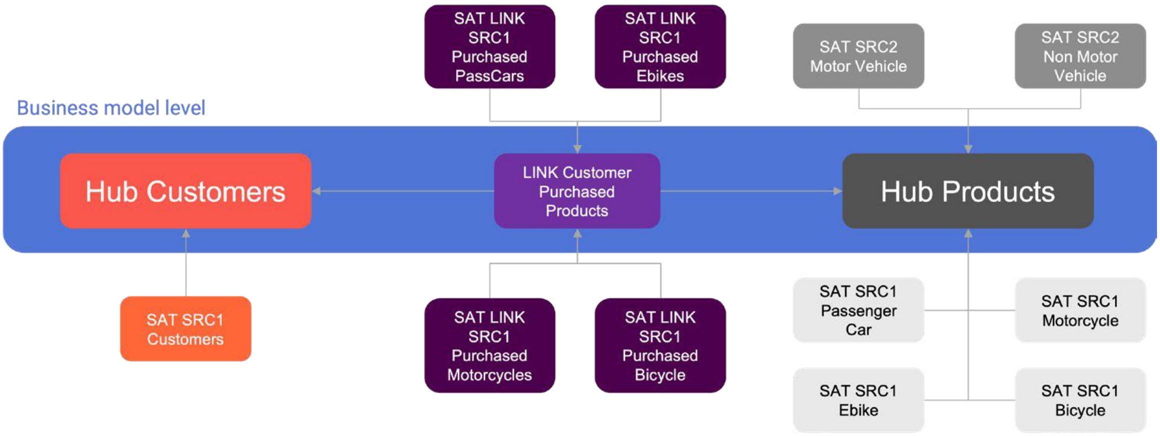
Data Vault also supports multi-temporal solutions. It not only provides standardized design patterns for managing multiple timelines, but also enables users to define and switch between them as needed—offering maximum temporal flexibility.

Beyond these core layers, Data Vault architecture provides several operational advantages:

- It enables distributed deployments, allowing an enterprise data warehouse, lakehouse, or mesh to span across clouds or geographic regions in multi-cloud setups.
- It supports hybrid environments, bridging on-premises and cloud systems seamlessly.
- It delivers full auditability, allowing every record to be traced back to its source and every report to be fully reconstructed.
- It accommodates granular security patterns, including cell-level access controls and GDPR-compliant data deletion or reduction strategies tailored to specific data needs.

Returning to our dealership example: the source models we explored can now be transformed into a Data Vault model. As intended, this target model should closely reflect the business conceptual model while ensuring that data from all relevant sources is accurately captured and integrated..

Target Data Vault model


SAT LINK SRC1 Purchased PassCars
SAT LINK SRC1 Purchased Ebikes
SAT SRC2 Motor Vehicle
SAT SRC2 Non Motor Vehicle
Business model level
Hub Customers
LINK Customer Purchased Products
Hub Products
SAT SRC1 Customers
SAT LINK SRC1 Purchased Motorcycles
SAT LINK SRC1 Purchased Bicycle
SAT SRC1 Passenger Car
SAT SRC1 Motorcycle
SAT SRC1 Ebike
SAT SRC1 Bicycle

Picture 15

The foundational structures of Data Vault make it possible to align source data with a unified business model. Hubs and links represent the business model, while satellites capture source-specific data. Let's take a closer look at how this works.

Hubs contain business keys, which are the identifiers of core business objects such as customers or products. These keys form the closest link between your source data and the business taxonomy. Identifying the correct business key for each concept is a critical step in the modeling process. These keys are not always the primary keys from the source system—they often come from different fields that better represent business meaning.

However, hubs and links don't represent the business objects directly—they represent the business keys. This distinction matters. For example, if a product appears multiple times in the source with different business keys, those duplicates all coexist within the same hub. The granularity of a hub is determined by its keys, not by a singular object definition. In some cases, the preferred business key might not even exist in a given source, forcing the use of an alternative identifier.

To align source models with the target Data Vault structure, the first step is identifying these business keys. In our dealership example:

- Cars might be identified by a vehicle number
- Bicycles by a serial number

Although these identifiers differ, both fall under the same business concept—products. Therefore, we group them in the same hub using a technique known as hub grouping:

- All business keys tied to the Product taxonomy go into HUB Products
- All business keys related to the Party taxonomy go into HUB Customers

Managing these groupings requires care. In cases where different source systems use the same key to refer to different entities, a Business Key Collision Code (BKCC) can help distinguish them. This prevents collisions and maintains data integrity across sources.

Continuing with our example, identifying the correct business key for customers introduces another layer of complexity. In the source, data is modeled at the second level of the party taxonomy—representing persons, a group that includes both customers and employees. While customers might be identified by a loyalty card or government ID, employees are typically tagged using an internal employee number.

This is a classic scenario where multiple business keys are entangled within a single source object. To align with the conceptual business model—which distinctly separates customers from employees—we need to split the source object. This typically involves a pre-staging area, where data is categorized and routed appropriately before it enters the Data Vault model..

Finally, links capture the unique relationships between business keys.

- In our business conceptual model, there's a defined purchase relationship between Products and Customers.
- This relationship is represented as a LINK_Customer_Purchased_Products in the Data Vault.

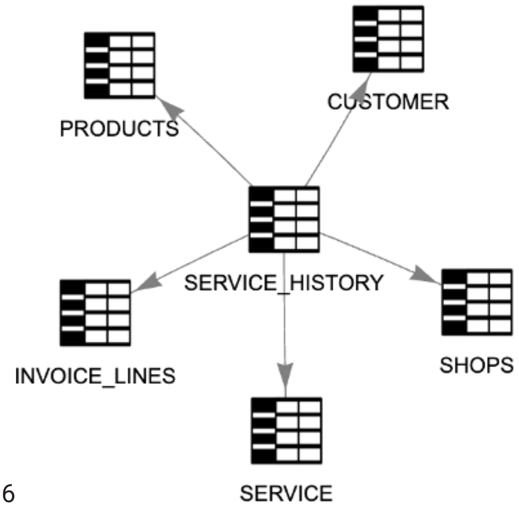
These definitions within the Data Vault standard ensure that the business model is faithfully represented in the physical target model using hubs and links.

Subtypes within the product taxonomy—level 3 in source 1 and level 2 in source 2—are handled as satellites of the product hub. This ensures that no relevant source data is lost during integration, regardless of its granularity.

Data Vault is also built to handle variations in naming conventions. For example, different systems might use "e-bike" or "public bike" to describe the same object. Not only can these names coexist, but they can also evolve over time without disrupting the integrity of the model. This makes the Data Vault highly resilient to change.

Now consider a new challenge: not everything sold by the dealership is a traditional product. Suppose a third source system tracks services—such as car repairs or bicycle rentals. This introduces new relationships between services, shops, products, and invoices (e.g., service history), and necessitates a separate service hub.

To integrate this new source into the existing Data Vault model, we need to carefully decide how to model the data—especially since some concepts like shops, customers, and invoices already exist in the previous sources.



Picture 16

.

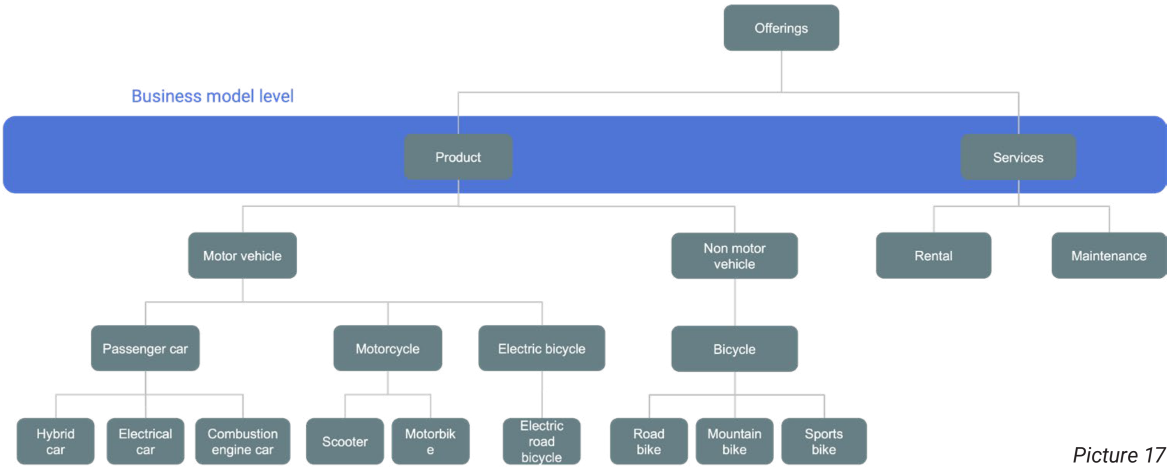
Source 3 introduces a new concept: services. To determine how best to integrate this concept into the existing business model, it’s essential to consult with key stakeholders across the organization.

There are two potential modeling approaches:

1. Treat services as another type of product and integrate them into the existing product hub

2. Recognize that services and products are fundamentally different and create a dedicated service hub

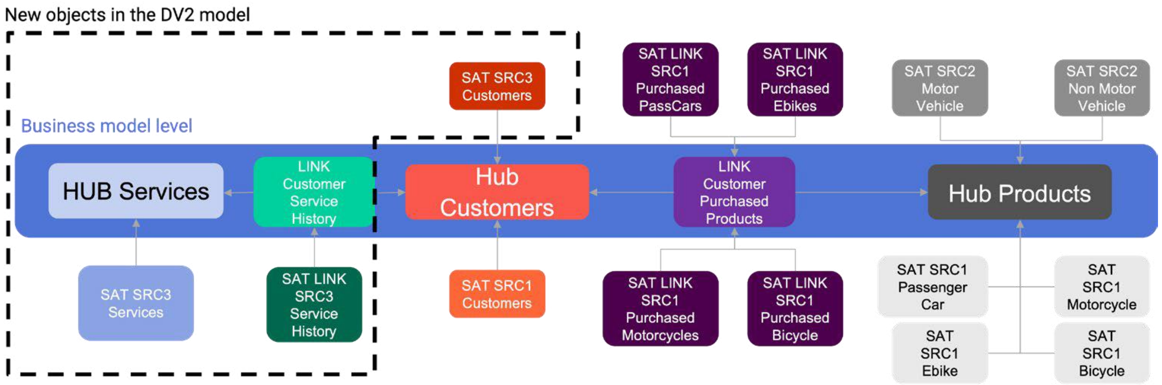
Given that services are managed by a separate sales team and follow a different customer journey timeline, the second option is more appropriate. As a result, the business model is updated to reflect both products and services as distinct concepts—each positioned at the second level of the revised taxonomy..



Picture 17

As a result, the Data Vault model incorporates a separate service hub, which links directly to the existing customer hub. Additionally, a new satellite from source 3 is attached to the customer hub to capture service-related attributes.

Adapted Target Data Vault model



New objects in the DV2 model
Business model level
SAT SRC3 Customers
SAT LINK SRC1 Purchased PassCars
SAT LINK SRC1 Purchased Ebikes
SAT SRC2 Motor Vehicle
SAT SRC2 Non Motor Vehicle
HUB Services
LINK Customer Service History
Hub Customers
LINK Customer Purchased Products
Hub Products
SAT SRC3 Services
SAT LINK SRC3 Service History
SAT SRC1 Customers
SAT LINK SRC1 Purchased Motorcycles
SAT LINK SRC1 Purchased Bicycle
SAT SRC1 Passenger Car
SAT SRC1 Motorcycle
SAT SRC1 Ebike
SAT SRC1 Bicycle

This update highlights the flexibility of the Data Vault model in accommodating new concepts. By identifying the correct placement within the business taxonomy and selecting the appropriate business keys, we can integrate new domains—like services—into the model without disrupting or reworking existing structures.

Picture 18

Automating multi-source data integration

We've established that Data Vault helps overcome the complexity of integrating multiple data sources and taxonomies into a unified target model. But when enterprises begin working with 20 or more sources, manual integration becomes a serious bottleneck.

Two major challenges emerge:

1. It becomes virtually impossible to manually construct a complete conceptual or target model.
2. Managing integration across multiple source systems, technologies, and data types becomes unmanageable at scale.

This is where data automation becomes essential.

Automation enables organizations to manage the vast volume of datasets by collecting large amounts of source metadata and enriching it—turning it into valuable, actionable input. The more metadata the system can process, the more scalable and intelligent the automation becomes. Think of it as upgrading from a copper UTP cable to fiber optics: same principles, exponentially greater speed and bandwidth.

In terms of output, automation covers three key areas:

1. The physical target data model – typically expressed as DDL (Data Definition Language) statements, also known as the relational model
2. The integration logic – the DML (Data Manipulation Language) code that transforms and loads data from source to target
3. The workflow orchestration – Python or equivalent scripts to schedule and manage the data pipelines

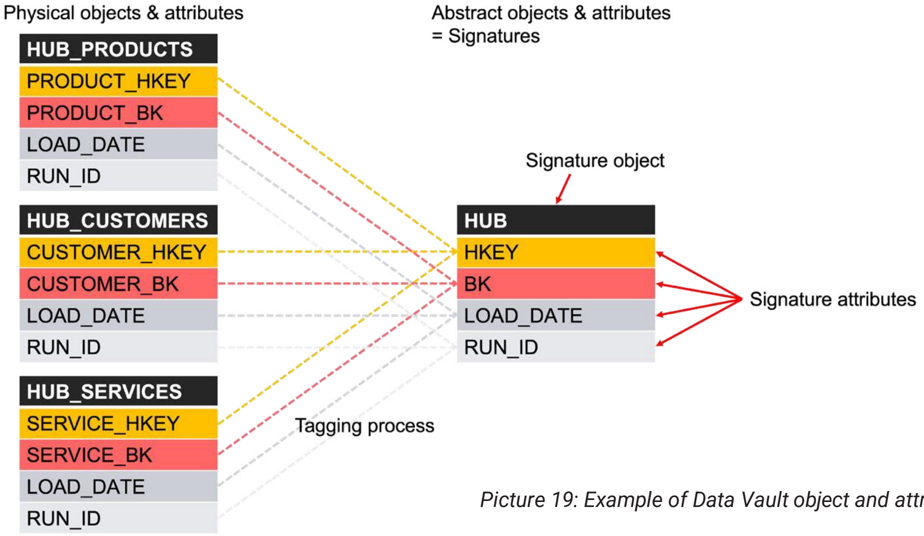
Without automation, productivity slows dramatically—but even more critically, consistency suffers. In data warehousing, quality is defined not by absolute precision, but by adherence to expected patterns. There are only a limited number of repeatable patterns for producing hubs, links, and satellites. Deviating from these patterns increases documentation effort, testing complexity, and ultimately, the risk of failure.

However, before automation can be effectively applied, a few **prerequisites** must be in place:

1. **Single-function object types**
   Each object type must serve a single, well-defined purpose. The more functions an object performs, the more complex and variable the loading patterns become—reducing repeatability and limiting the value of automation. Data Vault meets this requirement perfectly: hubs, links, and satellites are designed with only one or two functions, making them ideal for automation.
2. **Clear mapping between source and target**
   A reliable automation process depends on an unambiguous relationship between the source data

and its representation in the target model. In Data Vault, this is achieved through the *Raw Data Vault* layer, which stores the **Single Version of the Facts**—the raw, unaltered data as it exists in the source systems.

3. **Sufficient level of abstraction**
   Automation works best at an abstract level—not at the physical layer where implementation details vary widely. Data Vault supports this abstraction through its use of standardized object and attribute types (e.g., hubs, links, satellites, business keys, and historical attributes). These standardized components—referred to as **signature objects and attributes**—enable the development of reusable, pattern-based automation logic (as illustrated in Picture 20).



Picture 19: Example of Data Vault object and attribute signatures

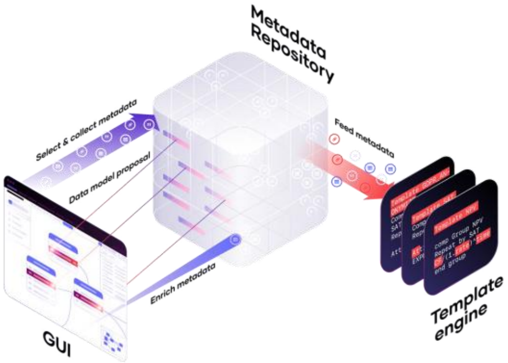
So, what is the data transformation equivalent of a **fiber optic cable**?

The answer lies in combining three essential components:

1. A **smart metadata repository**
2. **Built-in automation templates**
3. A robust **Graphical User Interface (GUI)** for intuitive data modeling

When properly configured, these components form a high-throughput, low-friction system for enterprise-scale data transformation.

In the next chapter, we'll show how these elements come together in practice—using our dealership example to illustrate how VaultSpeed brings this architecture to life.



Picture 20

Navigating the automation process with VaultSpeed

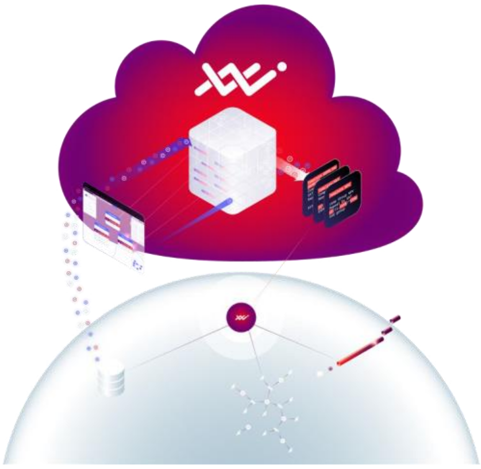
Getting your Data Vault model up and running is much like solving a puzzle. Without guidance, finding where to begin—and how the pieces fit together—can be time-consuming and frustrating. The hardest part is laying down the first pieces when there's no clear starting point.

That's where VaultSpeed changes the game. It pre-assembles the majority of the puzzle, giving you a solid foundation from the start. This allows your team to focus on completing the picture—filling in the remaining pieces with precision and confidence..

Step 1: harvest the metadata for the relevant data sources.

To help assemble the puzzle, VaultSpeed needs input—and that input is metadata. Automated data transformation starts with metadata collection, as it forms the foundation for building the target model.

VaultSpeed streamlines this step with a client-side agent that can harvest metadata from virtually any source technology, making the process both simple and efficient.



Picture 21

In this example:

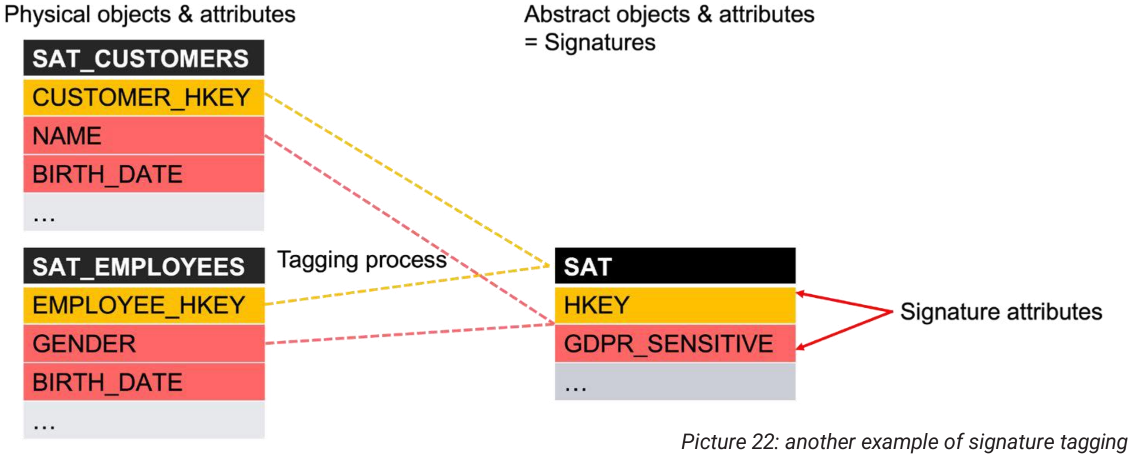
Data Source src1: the source that captured B2C sales data - as shown in Picture 7

Data Source src2: ERP source containing inventory data - as shown in Picture 10

Data Source src3: relating to the service operations – as shown in Picture 16

The harvested metadata is securely stored in a smart metadata repository, which serves as the foundation for automation. Within this repository, metadata can be organized into signature groups—collections that apply across multiple levels, including schema, object, and attribute.

These groups enable abstraction across different physical structures by classifying elements based on their role or business meaning. For example, the Data Vault standard defines common signature types such as hubs and business keys. You can also define custom, business-driven signature groups—such as one that flags all attributes containing GDPR-sensitive data (as shown in Picture 22)..



Picture 22: another example of signature tagging

Signature tagging acts as the crucial link between the physical structure of source systems and the abstract logic required for automation. It enables the use of **repeatable logic** across datasets, regardless of their origin.

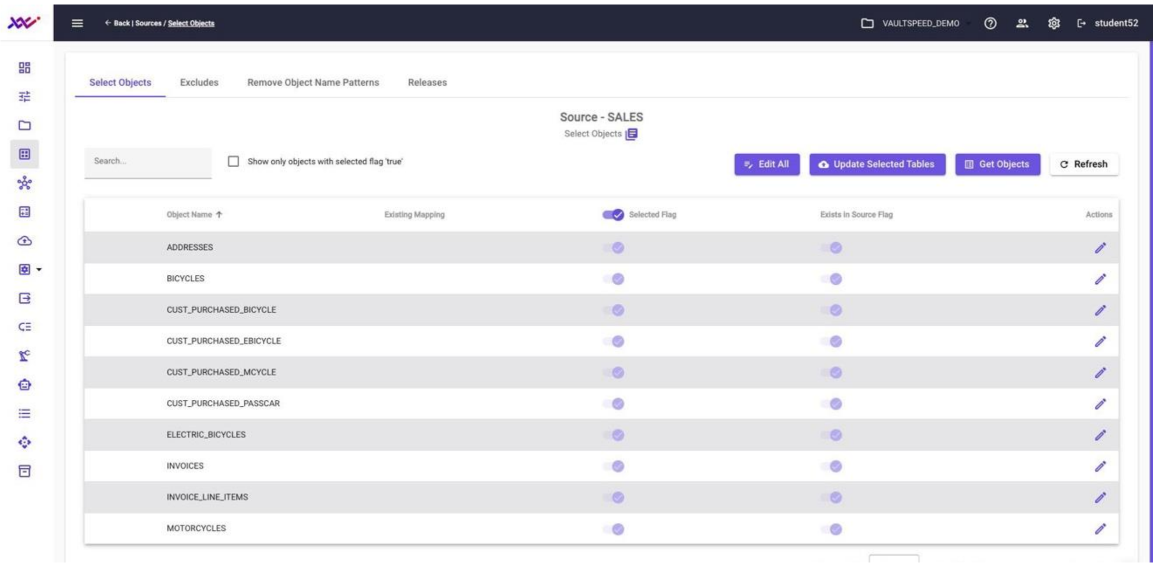
This logic is applied through **automation templates**, which drive consistency and efficiency. These templates support a broad range of use cases:

- **Data integration logic** (e.g., implementing Data Vault structures)
- **Business logic** (such as calculating total vehicle sales or service revenue)
- **Testing logic** (like verifying if values match across systems)

Instead of relying on physical components, these templates operate on **abstract signature elements**, allowing automation to scale across systems and scenarios.

VaultSpeed provides **pre-built Data Vault templates**, eliminating the need to build and test repetitive patterns from scratch—helping teams avoid costly errors caused by incorrect automation logic.

In the example below, you can see the **object selection screen for our B2C source**. Metadata has been automatically harvested, and users can now select the relevant source objects to be included in the integration process.



← Back | Sources / Select Objects
VAULTSPEED_DEMO
student52
Select Objects
Excludes
Remove Object Name Patterns
Releases

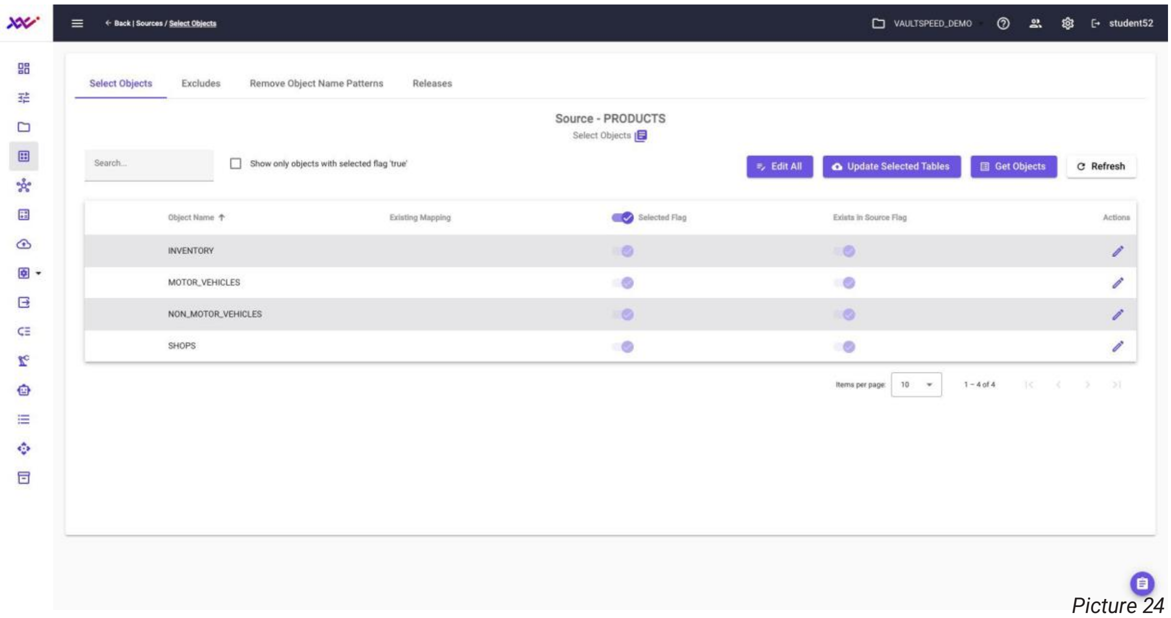
Source - SALES

Select Objects

Object Name ↑	Existing Mapping	Selected Flag	Exists in Source Flag	Actions
ADDRESSES		✓	✓	✎
BICYCLES		✓	✓	✎
CUST_PURCHASED_BICYCLE		✓	✓	✎
CUST_PURCHASED_EBICYCLE		✓	✓	✎
CUST_PURCHASED_MCYCLE		✓	✓	✎
CUST_PURCHASED_PASSCAR		✓	✓	✎
ELECTRIC_BICYCLES		✓	✓	✎
INVOICES		✓	✓	✎
INVOICE_LINE_ITEMS		✓	✓	✎
MOTORCYCLES		✓	✓	✎

Picture 23

Picture 24 shows the selection of the source objects for Data Source src2. The harvested metadata contains the objects relating to the product inventory.



Source - PRODUCTS

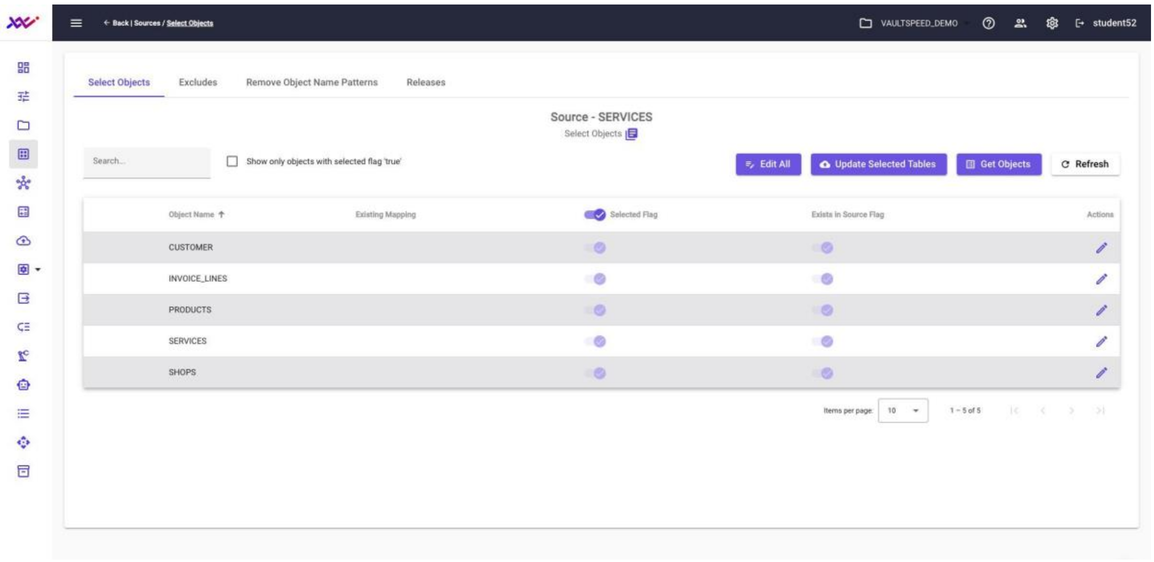
Select Objects

□ Show only objects with selected flag 'true'

Object Name ↑	Existing Mapping	Selected Flag	Exists in Source Flag	Actions
INVENTORY		✓	✓	✎
MOTOR_VEHICLES		✓	✓	✎
NON_MOTOR_VEHICLES		✓	✓	✎
SHOPS		✓	✓	✎

Picture 24

Picture 25 shows the selection of the source objects for Data Source src3. The harvested metadata contains the objects of the source that tracks maintenance and rental services.



Source - SERVICES

Select Objects

□ Show only objects with selected flag 'true'

Object Name ↑	Existing Mapping	Selected Flag	Exists in Source Flag	Actions
CUSTOMER		✓	✓	✎
INVOICE_LINES		✓	✓	✎
PRODUCTS		✓	✓	✎
SERVICES		✓	✓	✎
SHOPS		✓	✓	✎

Picture 25

Step 2: define the mapping of your source model toward a Data Vault model

The next challenge VaultSpeed solves is mapping large volumes of source metadata into pre-built automation templates.

To streamline this process, VaultSpeed relies on two key components:

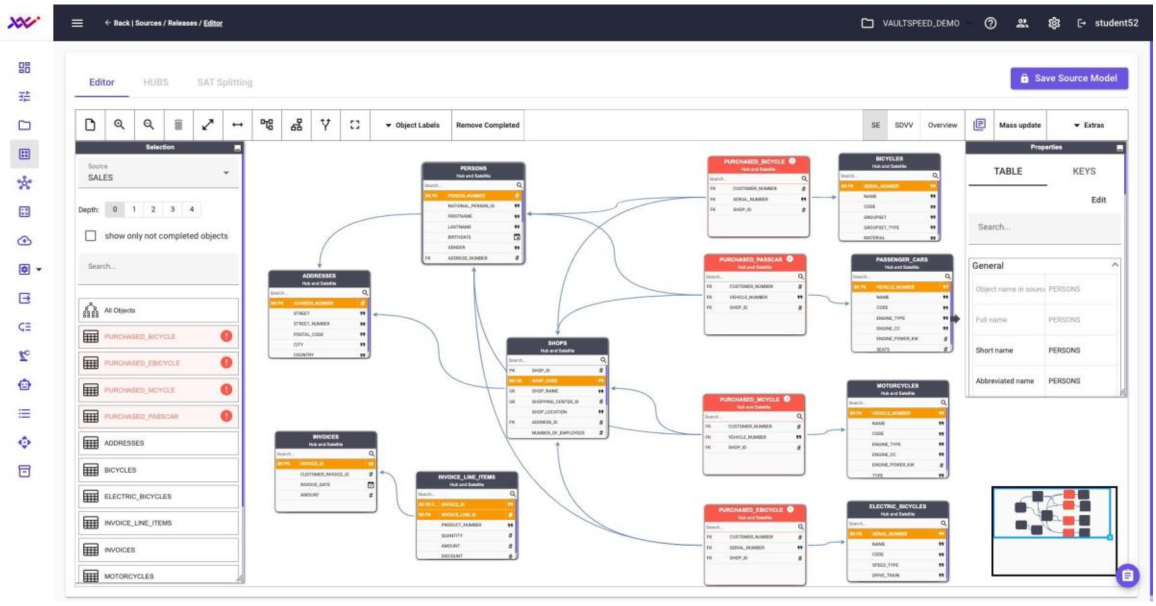
1. A smart rule engine within the metadata repository that analyzes harvested metadata and proposes a preliminary physical target model.
2. A graphical user interface (GUI) equipped with a powerful data modeler that lets users review, accept, correct, or enrich the proposed model—ensuring full control over the final outcome.

Together, this toolset enables teams to transform raw source metadata into a well-structured Data Vault target model, guided by the conceptual business model.

Modeling source 1

Based on the metadata captured in Step 1, VaultSpeed presents a proposed model—an initial solution to our Data Vault puzzle. This autogenerated model reflects VaultSpeed's interpretation of the source data and serves as a starting point for refinement.

In the remainder of this section, we'll walk through several examples to demonstrate how this modeling process unfolds in practice.



Source: SALES

Depth: 0 1 2 3 4

☐ show only not completed objects

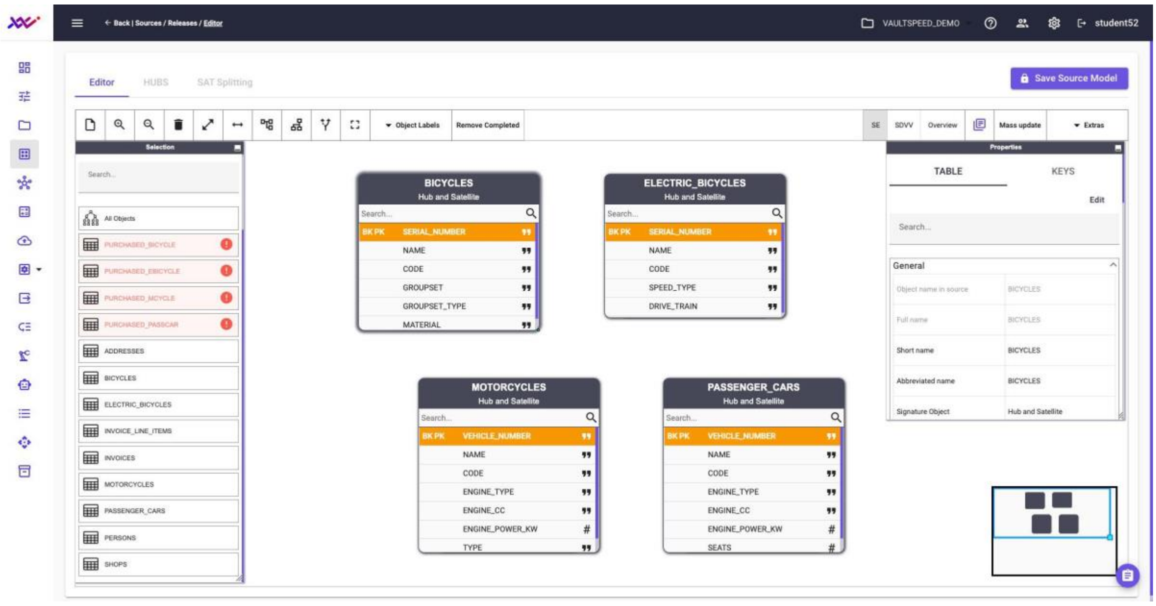
Object name in source	PERSONS
Full name	PERSONS
Short name	PERSONS
Abbreviated name	PERSONS

Picture 26

Accept

Within VaultSpeed's user community, over 70% of modeling proposals are approved without modification. This high success rate reflects the platform's ability to accurately interpret a wide range of settings—such as multi- vs. single-master configurations, default parameter values, naming conventions, object types, CDC (Change Data Capture) rules, and data quality controls.

A good example of this alignment is found in how VaultSpeed handles the modeling of bicycles, passenger cars, motorcycles, and e-bikes. The platform correctly proposes each as a hub with an associated satellite, using their respective serial numbers or vehicle IDs as business keys—just as a human modeler would.



BICYCLES
Hub and Satellite
BK PK SERIAL_NUMBER
NAME
CODE
GROUPSET
GROUPSET_TYPE
MATERIAL

ELECTRIC_BICYCLES
Hub and Satellite
BK PK SERIAL_NUMBER
NAME
CODE
SPEED_TYPE
DRIVE_TRAIN

MOTORCYCLES
Hub and Satellite
BK PK VEHICLE_NUMBER
NAME
CODE
ENGINE_TYPE
ENGINE_CC
ENGINE_POWER_KW #
TYPE

PASSENGER_CARS
Hub and Satellite
BK PK VEHICLE_NUMBER
NAME
CODE
ENGINE_TYPE
ENGINE_CC
ENGINE_POWER_KW #
SEATS #

General

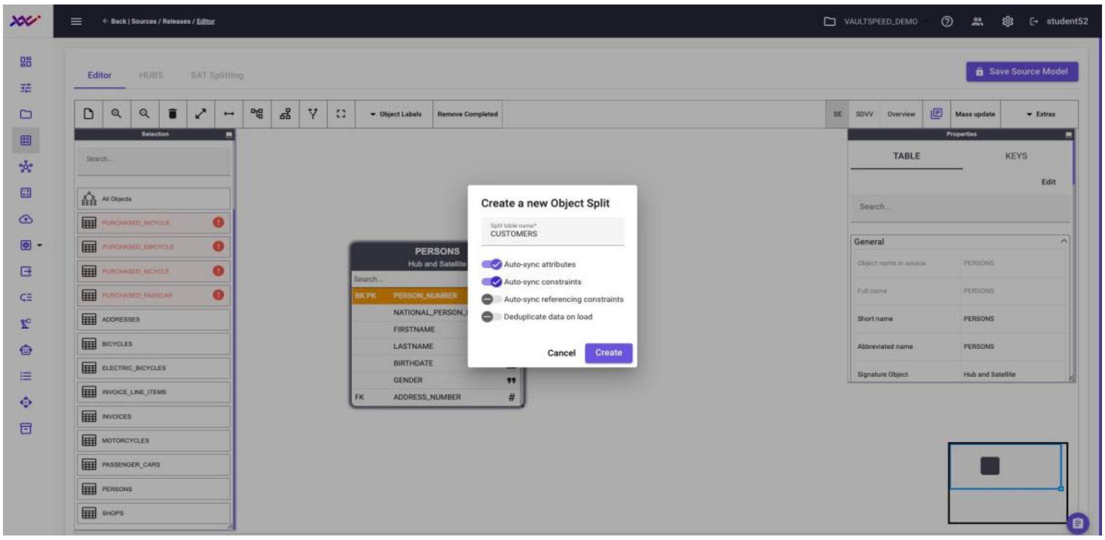
Object name in source	BICYCLES
Full name	BICYCLES
Short name	BICYCLES
Abbreviated name	BICYCLES
Signature Object	Hub and Satellite

Picture 27

Correct

In our source data, the persons object exists at level 2 of the party taxonomy and includes both employees and customers. To align with the business model—which treats these as distinct entities—we need to split this object and go one level deeper in the taxonomy.

VaultSpeed makes this easy. Simply right-click the source object and select the option to create a new object split. This allows you to separate customer data from employee data and map each to its appropriate hub in the Data Vault model..



PERSONS
Hub and Satellite
BK PK PERSON_NUMBER
NATIONAL_PERSON_
FIRSTNAME
LASTNAME
BIRTHDATE
GENDER
FK ADDRESS_NUMBER #

Create a new Object Split

Split table name*
CUSTOMERS

☑ Auto-sync attributes
☑ Auto-sync constraints
☐ Auto-sync referencing constraints
☐ Deduplicate data on load

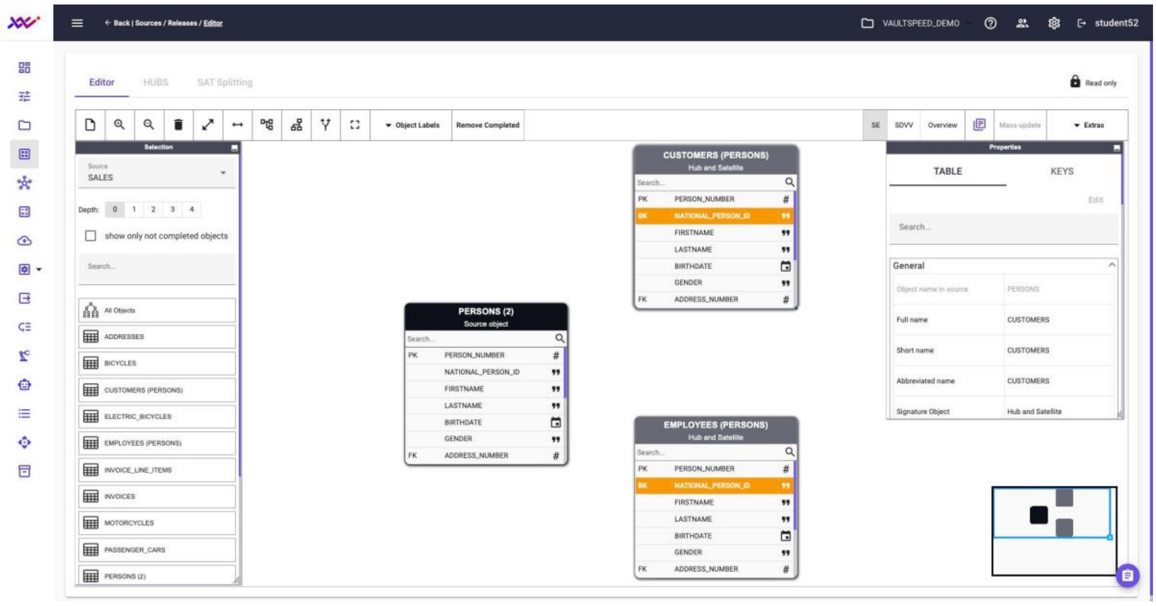
Cancel | Create

General

Object name in source	PERSONS
Full name	PERSONS
Short name	PERSONS
Abbreviated name	PERSONS
Signature Object	Hub and Satellite

Picture 28

Upon completion, the updated version of the model appears as follows:



PERSONS (2)
Source object

PK	PERSON_NUMBER	#
	NATIONAL_PERSON_ID	❞
	FIRSTNAME	❞
	LASTNAME	❞
	BIRTHDATE	📅
	GENDER	❞
FK	ADDRESS_NUMBER	#

CUSTOMERS (PERSONS)
Hub and Satellite

PK	PERSON_NUMBER	#
BK	NATIONAL_PERSON_ID	❞
	FIRSTNAME	❞
	LASTNAME	❞
	BIRTHDATE	📅
	GENDER	❞
FK	ADDRESS_NUMBER	#

EMPLOYEES (PERSONS)
Hub and Satellite

PK	PERSON_NUMBER	#
BK	NATIONAL_PERSON_ID	❞
	FIRSTNAME	❞
	LASTNAME	❞
	BIRTHDATE	📅
	GENDER	❞
FK	ADDRESS_NUMBER	#

General

Object name in source	PERSONS
Full name	CUSTOMERS
Short name	CUSTOMERS
Abbreviated name	CUSTOMERS
Signature Object	Hub and Satellite

Picture 29

As shown in the updated model, we've successfully separated customers and employees, reflecting their distinction within the party taxonomy as two different business concepts. Additionally, we've refined the business key selection—the natural_person_id field is now correctly assigned as the business key for both the customer and employee hubs.

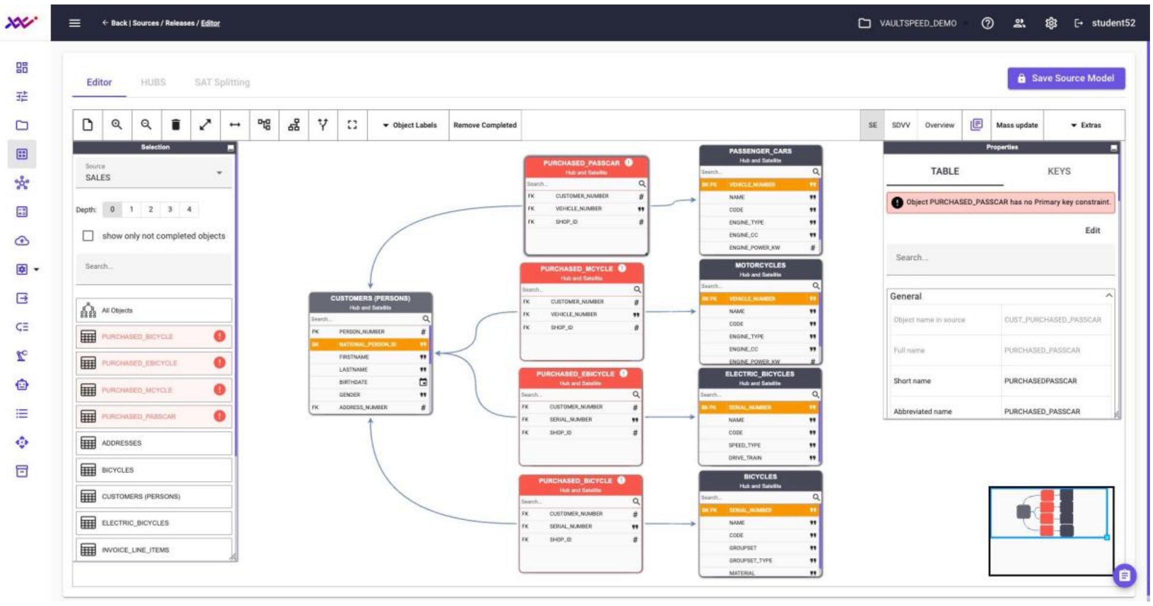
.

Enrich

Finally, let’s look at a case where metadata enrichment is required to enable VaultSpeed to generate a valid target model.

In Picture 30, the source editor displays several objects from src1 loaded onto the modeling canvas. VaultSpeed's built-in exception handling highlights four objects in red—indicating that they need attention before code generation can proceed.

When you drag an object into the canvas, VaultSpeed automatically displays the object along with its related elements, depending on the depth level set in the selection menu. This helps users quickly identify missing metadata or incomplete configurations that need to be addressed.



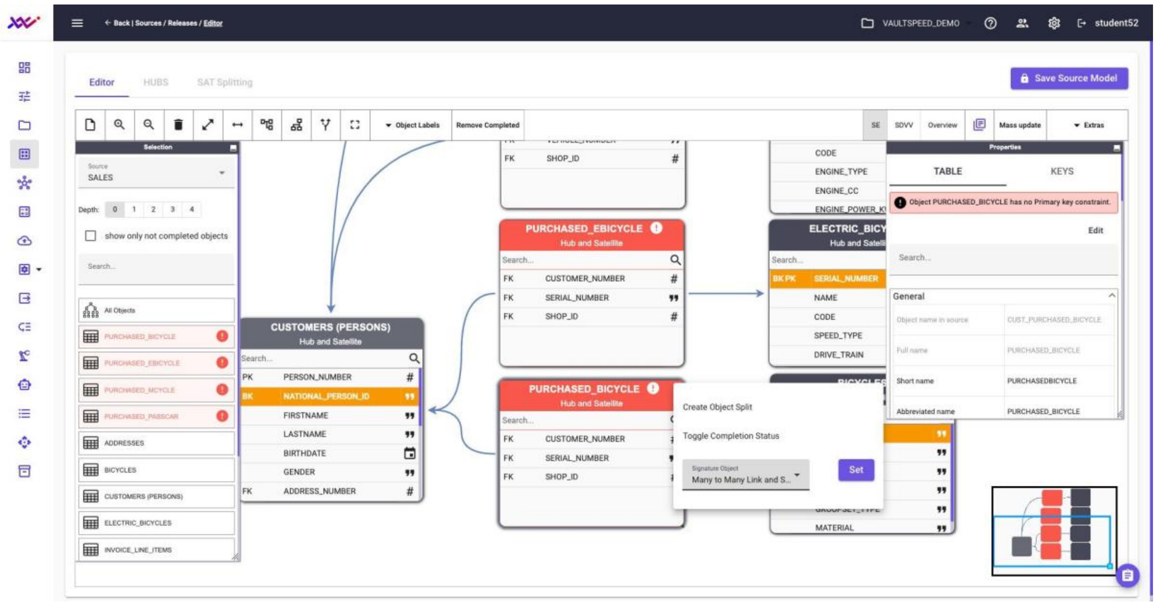
Object PURCHASED_PASSCAR has no Primary key constraint.

Object name in source	CUST_PURCHASED_PASSCAR
Full name	PURCHASED_PASSCAR
Short name	PURCHASEDPASSCAR
Abbreviated name	PURCHASED_PASSCAR

Picture 30

Let’s zoom in on the purchased_bicycle relationship and its related objects—bicycle and customer. This relationship is highlighted in red because it lacks a primary key, which is required for further processing.

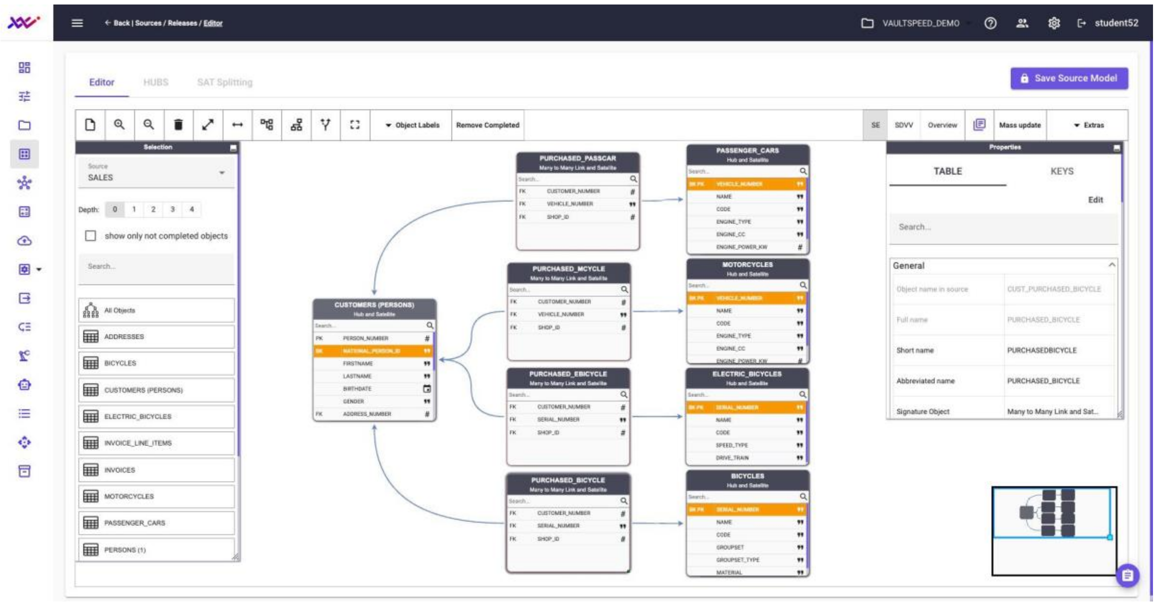
Since this is a many-to-many relationship, we need to explicitly define it in VaultSpeed. To do so, right-click the object and assign the appropriate signature object types: a many-to-many link and an accompanying satellite.



Picture 31

After defining all relevant purchased relationship objects as many-to-many links, the updated structure—shown in Picture 32—is now ready for integration into the target Data Vault model.

However, there's one final adjustment to make: this section must still be aligned with the taxonomy level defined for the product hub. We'll address that in one of the upcoming steps.

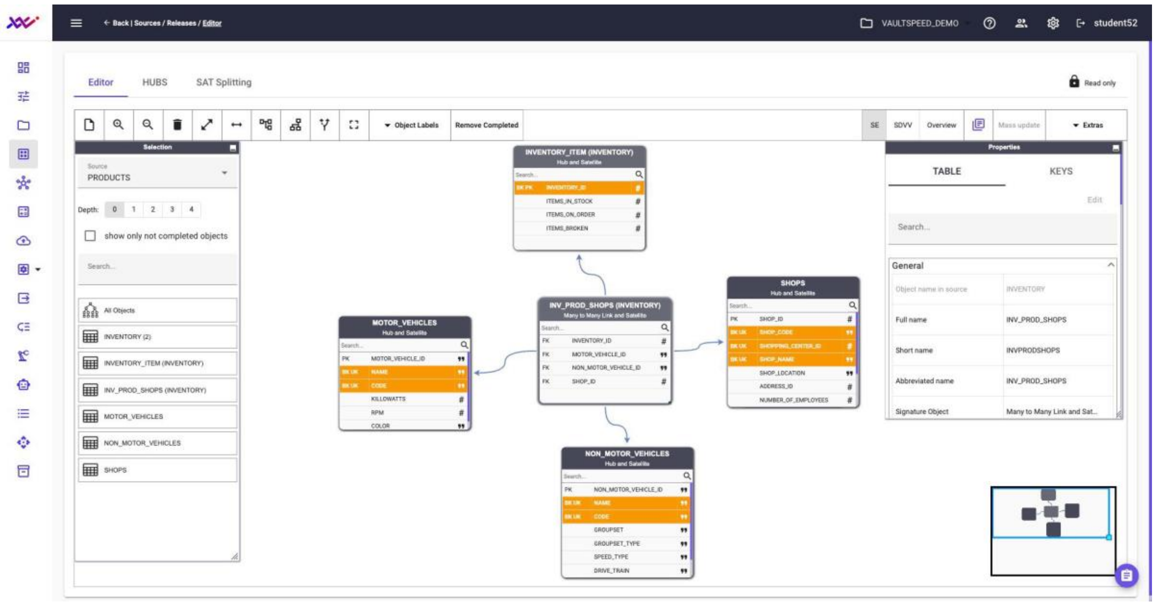


Picture 32

Modeling source 2

In the second source—the ERP-like system containing inventory data—we apply the same modeling approach. One of the key demonstrations of VaultSpeed's flexibility is its ability to customize proposal generation settings at the source level.

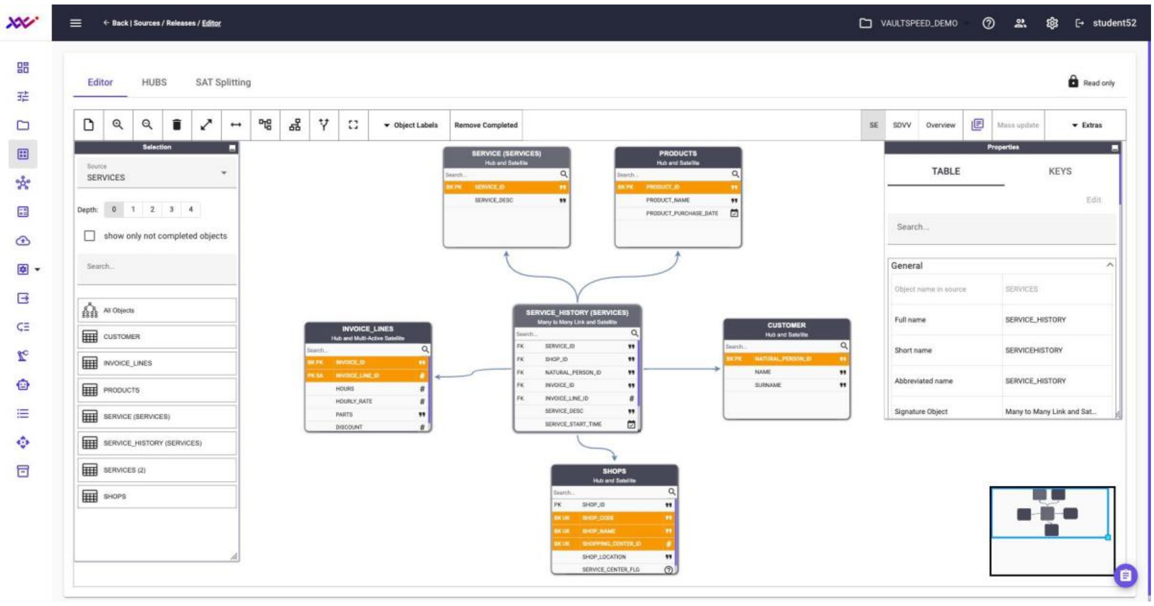
For instance, the USE_SOURCE_UK_AS_BK parameter allows VaultSpeed to automatically treat unique keys as business keys. This default behavior reduces the need for manual adjustments, streamlining the process of refining the source model.



Picture 33: The final source model for the ERP source.

Modeling source 3

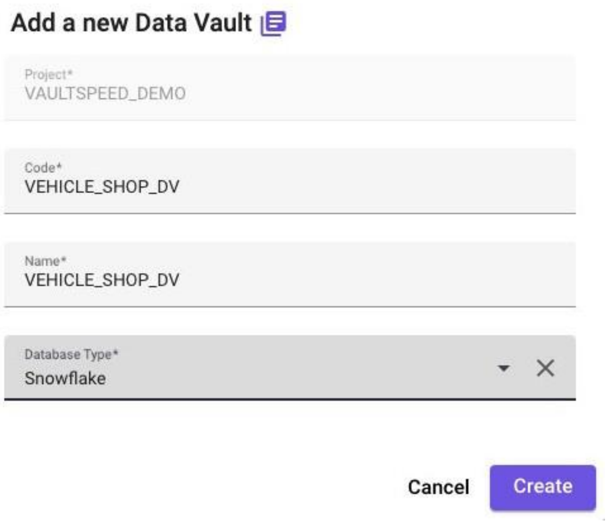
Finally, we apply the same modeling approach to src3, which contains the SERVICE source data. This results in the following configured source setup:



Picture 34

Step 3: Data Vault creation

The following step is to create a Data Vault.



Add a new Data Vault

Project*
VAULTSPEED_DEMO

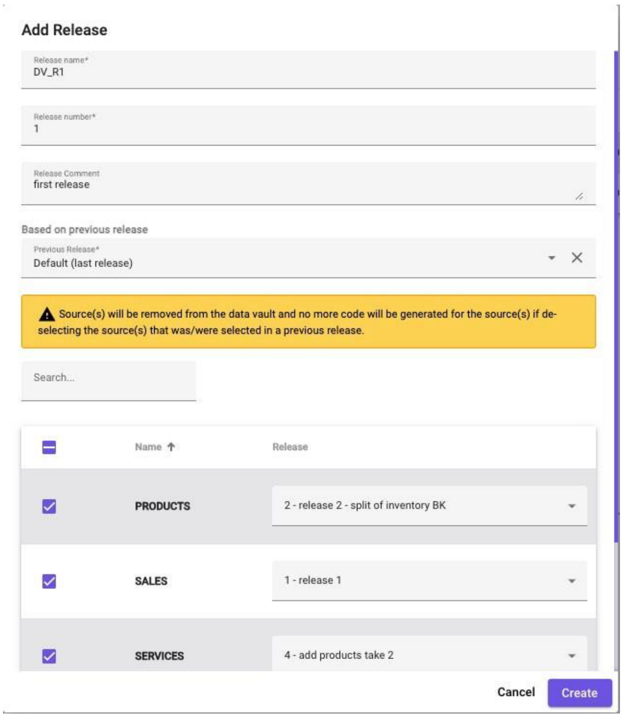
Code*
VEHICLE_SHOP_DV

Name*
VEHICLE_SHOP_DV

Database Type*
Snowflake

Picture 35

Create a new Data Vault release, and select all relevant sources and source releases:



Add Release

Release name*
DV_R1

Release number*
1

Release Comment
first release

Based on previous release

Previous Release*
Default (last release)

Source(s) will be removed from the data vault and no more code will be generated for the source(s) if de-selecting the source(s) that was/were selected in a previous release.

Search...

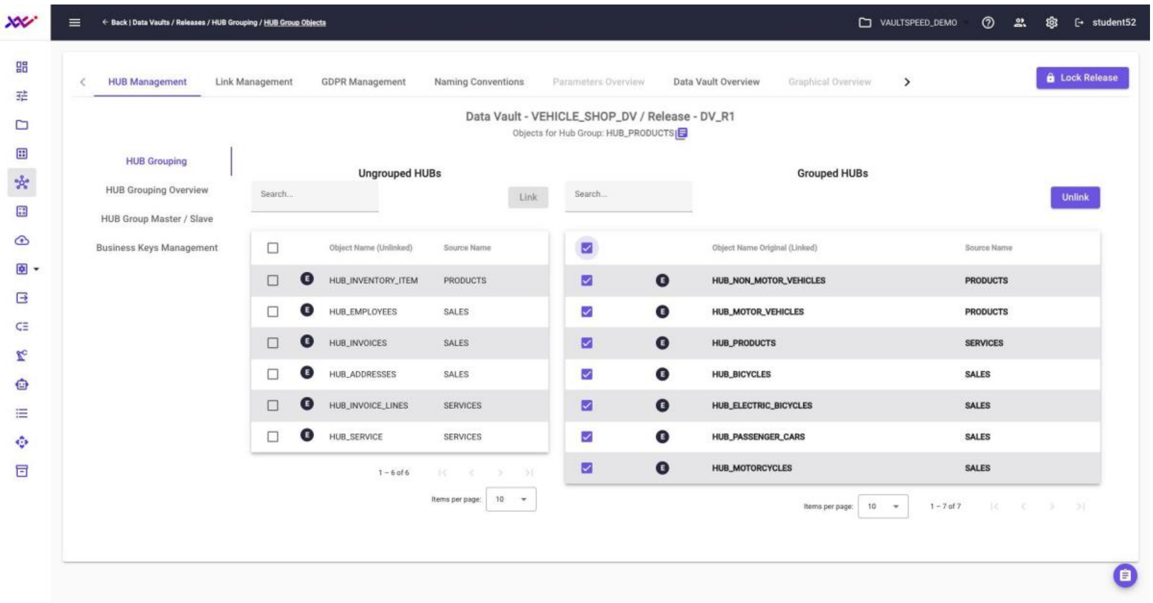
▬	Name	Release
☑	PRODUCTS	2 - release 2 - split of inventory BK
☑	SALES	1 - release 1
☑	SERVICES	4 - add products take 2

Picture 36

The next step is to align our target model with the business conceptual model, ensuring that all product-related data is grouped appropriately as satellites on the product hub.

VaultSpeed facilitates this through its hub group management screens. From here, you simply select the relevant source hubs on the left and assign them to the product hub group.

As outlined earlier, we intentionally avoid grouping SERVICE business keys with PRODUCT business keys. Despite surface similarities, their semantic meanings differ, and keeping them in separate hubs preserves clarity and structural integrity in the model.



Data Vault - VEHICLE_SHOP_DV / Release - DV_R1

Objects for Hub Group: HUB_PRODUCTS

Ungrouped HUBs

□		Object Name (Unlinked)	Source Name
□	E	HUB_INVENTORY_ITEM	PRODUCTS
□	E	HUB_EMPLOYEES	SALES
□	E	HUB_INVOICES	SALES
□	E	HUB_ADDRESSES	SALES
□	E	HUB_INVOICE_LINES	SERVICES
□	E	HUB_SERVICE	SERVICES

Grouped HUBs

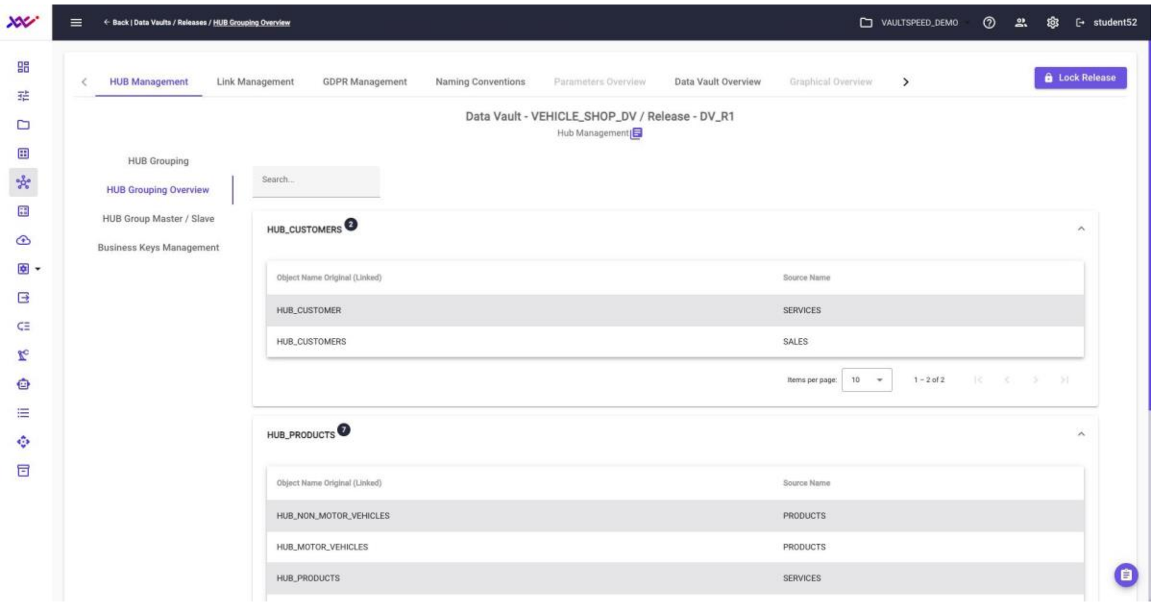
☑		Object Name Original (Linked)	Source Name
☑	E	HUB_NON_MOTOR_VEHICLES	PRODUCTS
☑	E	HUB_MOTOR_VEHICLES	PRODUCTS
☑	E	HUB_PRODUCTS	SERVICES
☑	E	HUB_BICYCLES	SALES
☑	E	HUB_ELECTRIC_BICYCLES	SALES
☑	E	HUB_PASSENGER_CARS	SALES
☑	E	HUB_MOTORCYCLES	SALES

Picture 37

We apply the same grouping logic to our customer data. In the hub group overview, two key hub groups emerge as central to our example:

- One for products, integrating data from src1 and src2
- One for customers, consolidating data from src1 and src3

This organization ensures that related business keys are grouped consistently, reflecting the business model while preserving source-level traceability.



Data Vault - VEHICLE_SHOP_DV / Release - DV_R1

Hub Management

HUB_CUSTOMERS 2

Object Name Original (Linked)	Source Name
HUB_CUSTOMER	SERVICES
HUB_CUSTOMERS	SALES

HUB_PRODUCTS 7

Object Name Original (Linked)	Source Name
HUB_NON_MOTOR_VEHICLES	PRODUCTS
HUB_MOTOR_VEHICLES	PRODUCTS
HUB_PRODUCTS	SERVICES

Picture 38

The proposed integration is designed to accommodate both technical and business considerations.

When integrating data from diverse datasets, a critical question arises: Which source is the master of the data?

There can be multiple answers to this question:

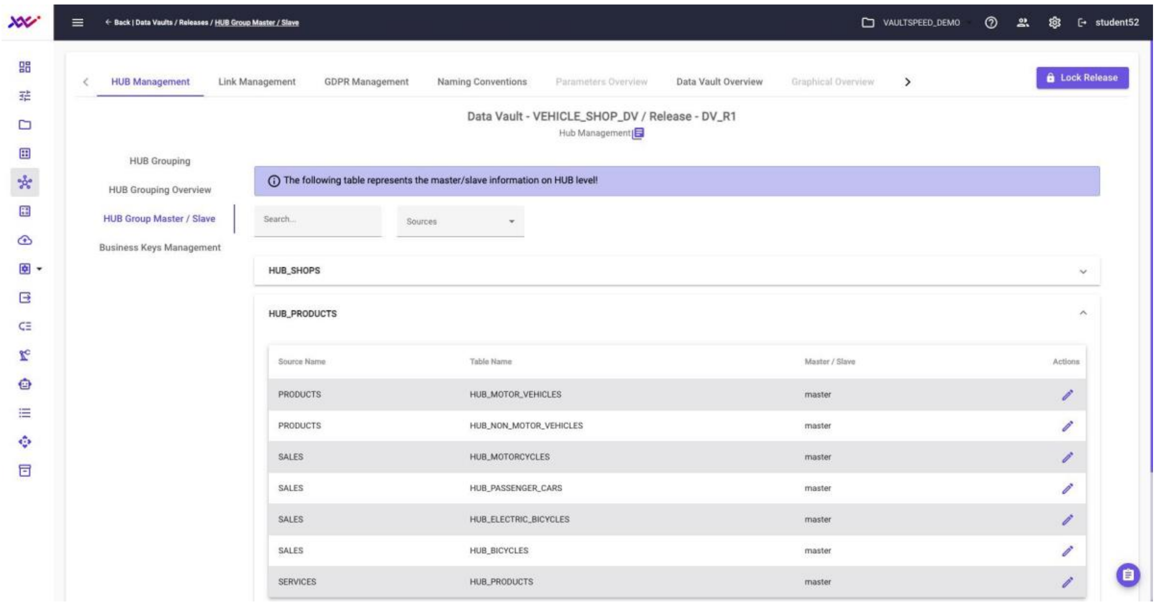
1. All sources contain master data.

2. One source is considered the primary/master, and others are secondary/slave.

3. A more intricate combination of the above, offering various options.

VaultSpeed empowers you to customize the setup to precisely match your requirements. Within the HUB management menu, you will discover an array of features designed for this purpose. In this instance, we can simply opt for the default setup for all hub groups, as all sources serve as masters of the data.

.



Data Vault - VEHICLE_SHOP_DV / Release - DV_R1

Hub Management

The following table represents the master/slave information on HUB level!

HUB_SHOPS

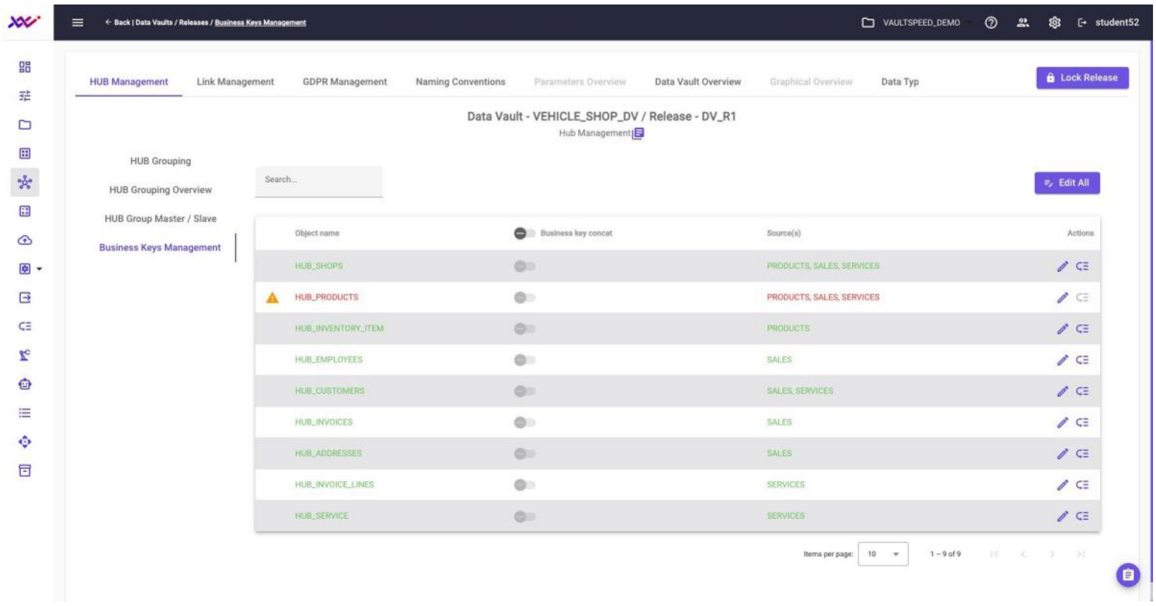
HUB_PRODUCTS

Source Name	Table Name	Master / Slave	Actions
PRODUCTS	HUB_MOTOR_VEHICLES	master	
PRODUCTS	HUB_NON_MOTOR_VEHICLES	master	
SALES	HUB_MOTORCYCLES	master	
SALES	HUB_PASSENGER_CARS	master	
SALES	HUB_ELECTRIC_BICYCLES	master	
SALES	HUB_BICYCLES	master	
SERVICES	HUB_PRODUCTS	master	

Picture 39

To streamline the Hub Business Key for the Product Business Element, we should create a concatenated key. This approach consolidates multiple key attributes into a single value, simplifying the hub structure—especially in multi-master scenarios where different source systems use different key names.

By concatenating the key components, we can generate a unified hash key that ensures consistency across diverse inputs. VaultSpeed intelligently flags where such concatenation may be needed, guiding you to apply it where appropriate and further strengthening the integrity of your target Data Vault model.



Data Vault - VEHICLE_SHOP_DV / Release - DV_R1

Hub Management

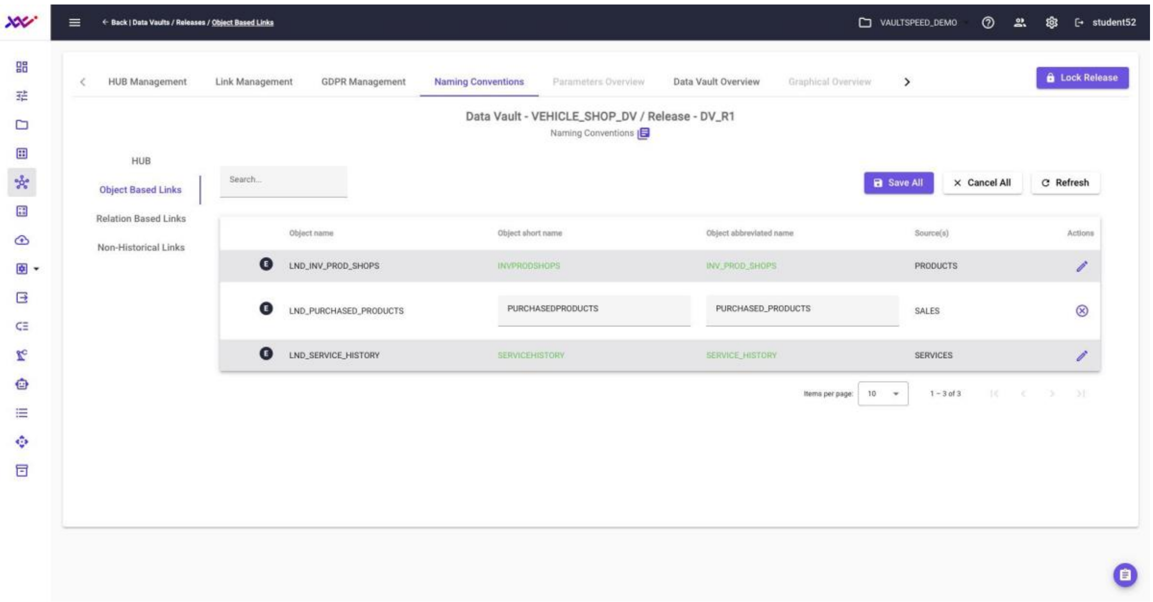
Object name	Business key concat	Source(s)	Actions
HUB_SHOPS		PRODUCTS, SALES, SERVICES	
⚠ HUB_PRODUCTS		PRODUCTS, SALES, SERVICES	
HUB_INVENTORY_ITEM		PRODUCTS	
HUB_EMPLOYEES		SALES	
HUB_CUSTOMERS		SALES, SERVICES	
HUB_INVOICES		SALES	
HUB_ADDRESSES		SALES	
HUB_INVOICE_LINES		SERVICES	
HUB_SERVICE		SERVICES	

Items per page: 10 1 – 9 of 9

Picture 40

Let’s take a moment to review naming conventions within VaultSpeed Data Vault models. By default, objects are assigned system-generated names, but these can be easily modified—either individually or through global parameter settings.

In our example, to better align with the business context and target model, we’ve renamed the link to “purchased_products”, providing clearer semantic meaning for end users and stakeholders.



Data Vault - VEHICLE_SHOP_DV / Release - DV_R1

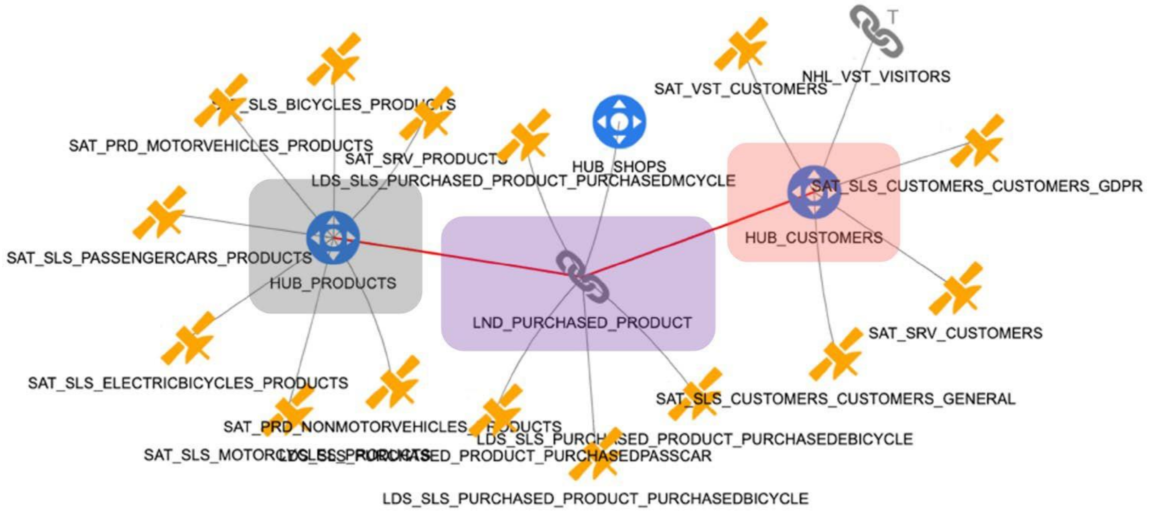
Naming Conventions

	Object name	Object short name	Object abbreviated name	Source(s)	Actions
E	LND_INV_PROD_SHOPS	INVPRODSHOPS	INV_PROD_SHOPS	PRODUCTS	
E	LND_PURCHASED_PRODUCTS	PURCHASEDPRODUCTS	PURCHASED_PRODUCTS	SALES	
E	LND_SERVICE_HISTORY	SERVICEHISTORY	SERVICE_HISTORY	SERVICES	

Items per page: 10 1 – 3 of 3

Picture 41: The result of the renaming to the Business level Taxonomy naming.

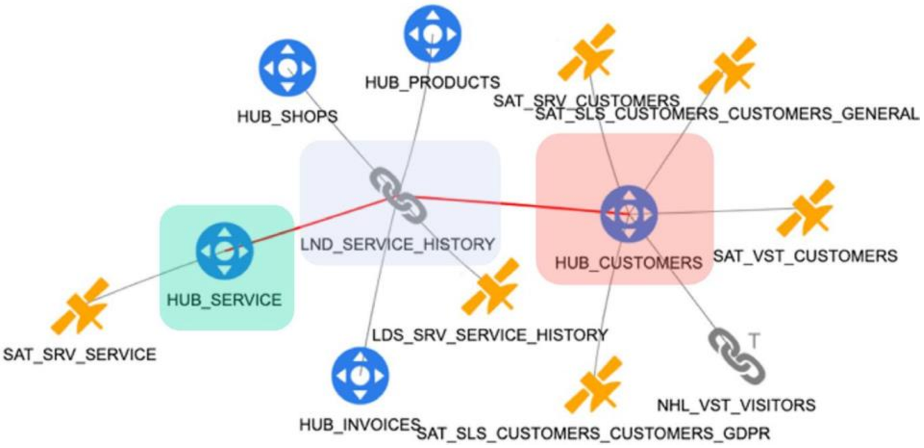
With all components now defined, we can present the complete Data Vault target model, as shown in Pictures 42 to 44. Picture 42 illustrates the integrated model for products from src1 and src2, along with the link connecting customers to purchased products, capturing the core relationship that drives the business use case.



T
SAT_VST_CUSTOMERS
NHL_VST_VISITORS
SAT_SLS_BICYCLES_PRODUCTS
SAT_PRD_MOTORVEHICLES_PRODUCTS
SAT_SRV_PRODUCTS
HUB_SHOPS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDMCYCLE
SAT_SLS_CUSTOMERS_CUSTOMERS_GDPR
HUB_CUSTOMERS
SAT_SLS_PASSENGERCARS_PRODUCTS
HUB_PRODUCTS
LND_PURCHASED_PRODUCT
SAT_SRV_CUSTOMERS
SAT_SLS_ELECTRICBICYCLES_PRODUCTS
SAT_SLS_CUSTOMERS_CUSTOMERS_GENERAL
SAT_PRD_NONMOTORVEHICLES_PRODUCTS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDEBICYCLE
SAT_SLS_MOTORCYCLES_PRODUCTS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDPASSCAR
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDBICYCLE

Picture 42

The diagram in picture 43 shows the detailed model of the service hub from src3:

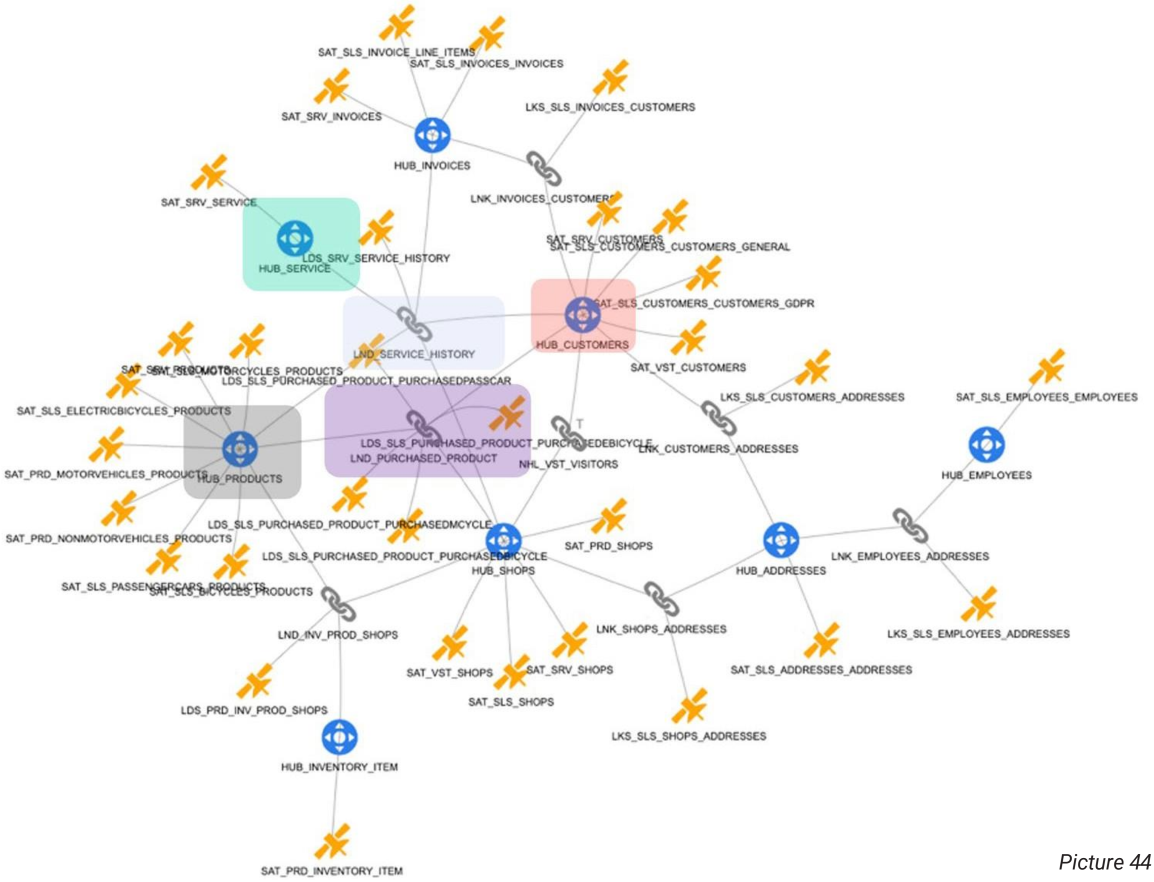


Picture 43

Picture 44 provides a comprehensive overview of the entire Data Vault model, integrating data from all three source systems. This unified structure enables the analysis of residential customer purchase behavior with full traceability and consistency.

To further simplify end-user querying, VaultSpeed also offers an additional automation layer that supports the creation of customized business logic and the presentation layer on top of the Data Vault. However, the details of that functionality are beyond the scope of this paper.

.


SAT_SLS_INVOICE_LINE_ITEMS
SAT_SLS_INVOICES_INVOICES
LKS_SLS_INVOICES_CUSTOMERS
SAT_SRV_INVOICES
HUB_INVOICES
SAT_SRV_SERVICE
LNK_INVOICES_CUSTOMERS
LDS_SRV_SERVICE_HISTORY
HUB_SERVICE
SAT_SRV_CUSTOMERS
SAT_SLS_CUSTOMERS_CUSTOMERS_GENERAL
SAT_SLS_CUSTOMERS_CUSTOMERS_GDPR
HUB_CUSTOMERS
LND_SERVICE_HISTORY
SAT_VST_CUSTOMERS
SAT_SRV_PRODUCTS
SAT_SLS_MOTORCYCLES_PRODUCTS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDPASSCAR
LKS_SLS_CUSTOMERS_ADDRESSES
SAT_SLS_EMPLOYEES_EMPLOYEES
SAT_SLS_ELECTRICBICYCLES_PRODUCTS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDEBICYCLE
T
LNK_CUSTOMERS_ADDRESSES
LND_PURCHASED_PRODUCT
NHL_VST_VISITORS
HUB_EMPLOYEES
SAT_PRD_MOTORVEHICLES_PRODUCTS
HUB_PRODUCTS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDMCYCLE
SAT_PRD_NONMOTORVEHICLES_PRODUCTS
SAT_PRD_SHOPS
LDS_SLS_PURCHASED_PRODUCT_PURCHASEDBICYCLE
HUB_SHOPS
HUB_ADDRESSES
LNK_EMPLOYEES_ADDRESSES
SAT_SLS_PASSENGERCARS_PRODUCTS
SAT_SLS_BICYCLES_PRODUCTS
LND_INV_PROD_SHOPS
LNK_SHOPS_ADDRESSES
LKS_SLS_EMPLOYEES_ADDRESSES
SAT_VST_SHOPS
SAT_SRV_SHOPS
SAT_SLS_ADDRESSES_ADDRESSES
SAT_SLS_SHOPS
LDS_PRD_INV_PROD_SHOPS
LKS_SLS_SHOPS_ADDRESSES
HUB_INVENTORY_ITEM
SAT_PRD_INVENTORY_ITEM

Picture 44

Does this scale?

Technical documents often rely on carefully crafted examples to explain complex ideas with clarity. These examples must strike the right balance—not so simple that they lack relevance, and not so complex that they obscure the core message. It's fair, then, to ask: does the Data Vault approach scale to real-world integration challenges?

The answer is yes—and here's the proof.

Picture 45 illustrates a real Data Vault model built for a large enterprise client. This project successfully integrated 58 diverse data sources and 592 distinct source objects. To protect confidentiality, the model is fully anonymized, yet several important observations can still be made.

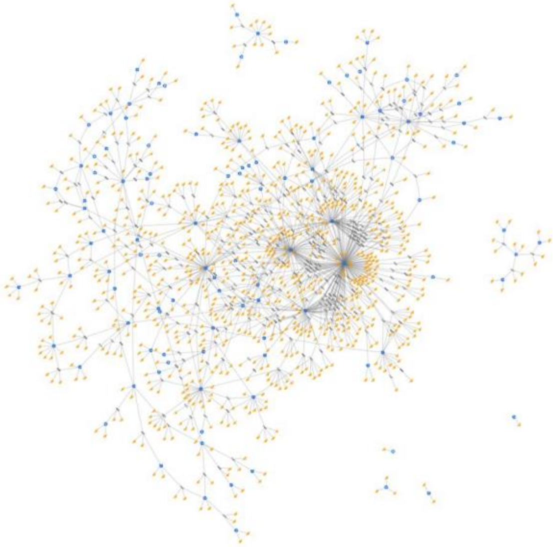
First, you'll notice that the number of hubs and links is significantly smaller compared to the volume of satellites. This reflects a deliberate effort to consolidate business keys—grouping data effectively around the most critical business concepts. Some hubs have an exceptionally high number of satellites, demonstrating the model's efficiency and scalability.

Second, only a few hubs and satellites appear disconnected from the rest of the model—further reinforcing the strength of its integration.

If you filter out the satellites, the underlying business concepts and their relationships become clear, making it easy for business users to recognize and navigate their own processes within the model.

Crucially, this model wasn't built all at once. The project followed an agile approach, gradually integrating new sources over time. Thanks to the flexibility of the Data Vault methodology, the team was able to add new components without reworking any previous structures—a powerful demonstration of adaptability at scale.

.



Picture 45

Conclusion

In conclusion, we successfully tackled the automation challenge and integrated data from three distinct sources into a cohesive, business-aligned Data Vault model. By using the business conceptual model as our foundation, we ensured that the resulting physical model accurately reflects business logic—making it both meaningful and accessible to business users.

VaultSpeed played a central role in this process. Its template engine leverages complete metadata access to convert repeatable logic into ready-to-deploy code—including DDL, DML, and workflow automation scripts. This code can be executed in your preferred data runtime environment, enabling fast, reliable deployment and ensuring your Data Vault model delivers value from day one.

.

Visit our site
vaultspeed.com

Contact sales
sales@vaultspeed.com

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

Join our community
community.vaultspeed.com

London Boston Leuven Vilnius
Vaartkom 41
3000 Leuven
Belgium

VAULTSPEED