



Automating common understanding

Accelerating the integration of different
data source views into one Data Vault

Dirk Vermeiren, Michael Olschimke

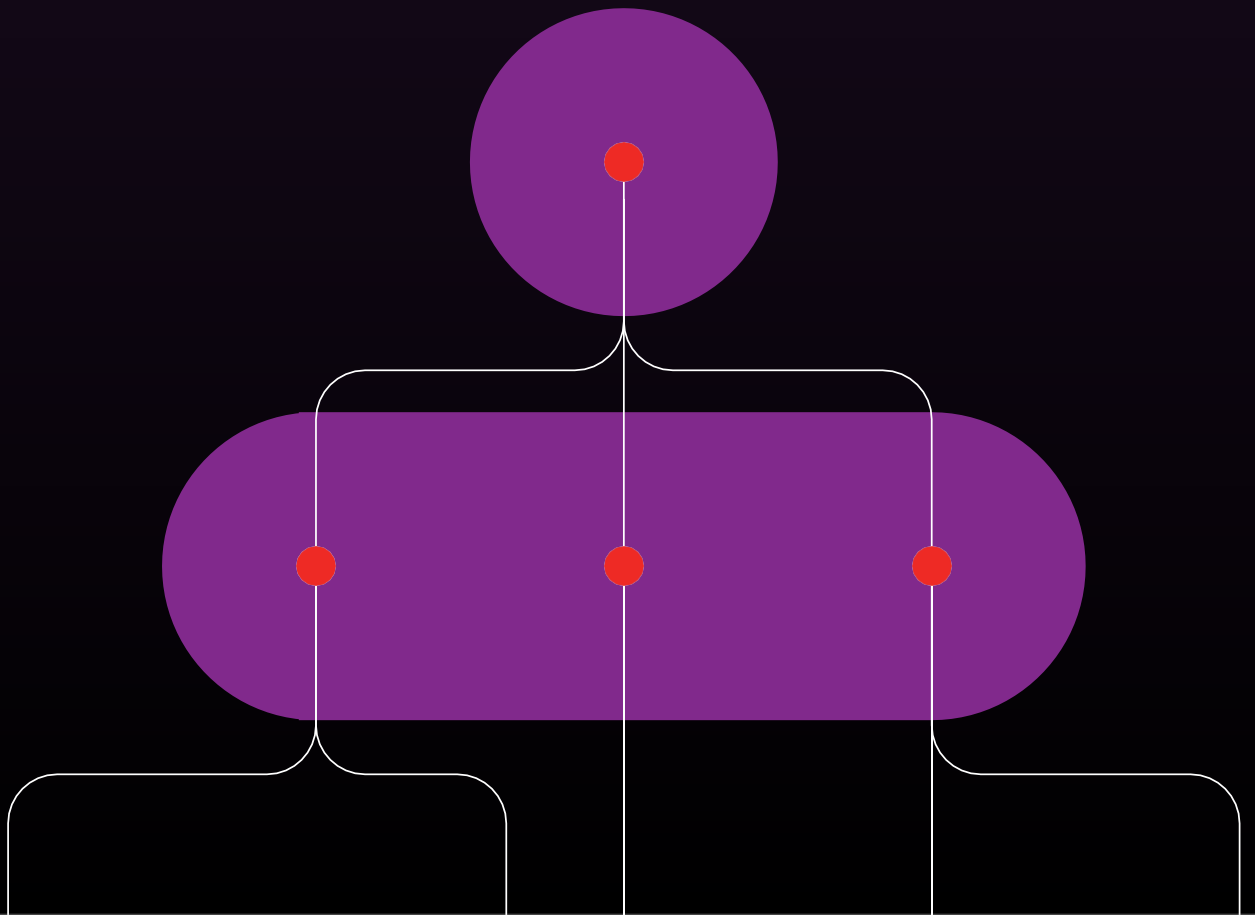


Table of Contents

| | |
|--|----|
| Welcome to the real world..... | 3 |
| Coping with data complexity | 5 |
| Industry data models are not silver bullets..... | 5 |
| Addressing data complexity through proper data modeling | 5 |
| Tools to model common understanding..... | 8 |
| Create a shared language with taxonomies | 8 |
| The optimal modeling language: Data Vault | 14 |
| Data Vault architecture | 15 |
| Automating multi-source data integration..... | 20 |
| Navigating the automation process with VaultSpeed..... | 22 |
| Step 1: harvest the metadata for the relevant data sources | 22 |
| Step 2: define the mapping of your source model toward a Data Vault model..... | 25 |
| Modeling source 1..... | 25 |
| Modeling source 2..... | 30 |
| Modeling source 3..... | 30 |
| Step 3: Data Vault creation | 31 |
| Does this scale? | 37 |
| Conclusion..... | 38 |

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.

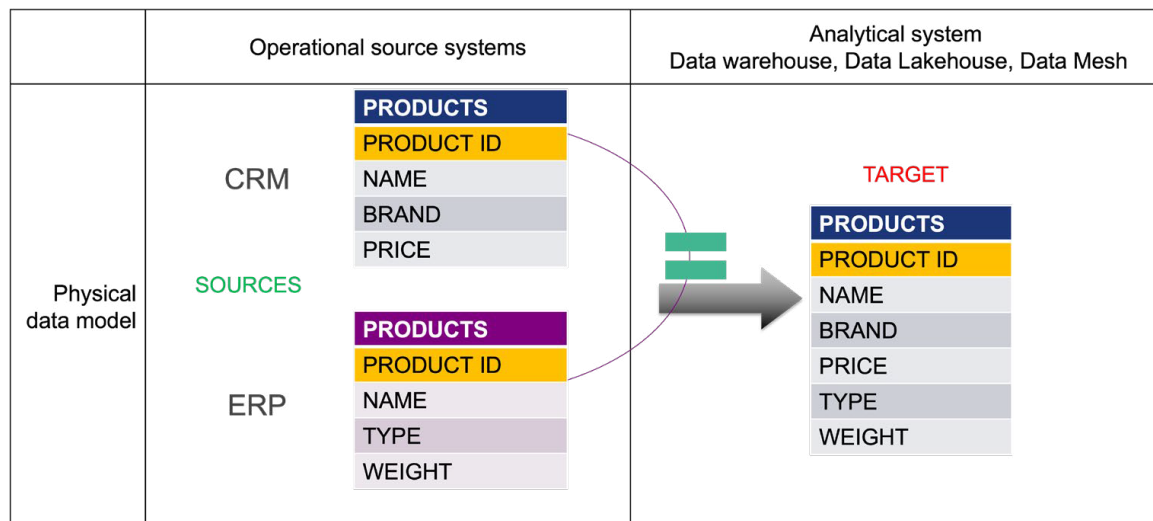
Welcome to the real world

In the realm of corporate operations, a company’s understanding of its business processes, its products, and its target audience frequently clashes with the varied terminologies used across different business lines. Additionally, business processes and terminologies are subject to change over time. This incongruity can give rise to substantial hurdles, including complications in data integration that not only impede timely decision-making but also have the potential to cause missed business opportunities.

Unfortunately, a prevalent misconception persists, wherein many individuals continue to believe that a sole entity holds exclusive ownership of both operational and analytical data models, while also assuming that these models remain perpetually unchanging.

Let’s illustrate this with an example: Imagine you work for a vehicle dealership in a timeless world where you single-handedly develop all operational and analytical data systems. You’ve created two operational source systems - a Customer Relationship Management (CRM) and an Enterprise Resource Planning (ERP) system. Both of these systems contain data related to the products.

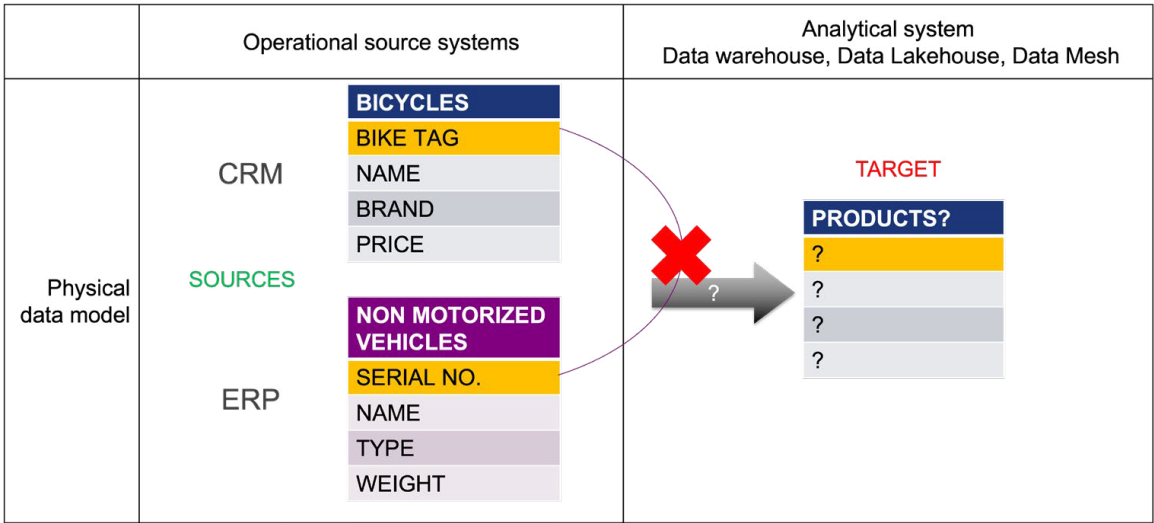
Now, you’re tasked with creating an integrated solution for analyzing the vehicle product portfolio including bikes and cars. In this hypothetical, timeless scenario, the understanding of business processes remains constant. Consequently, you would have configured the product entity in both operational systems using the same unique identifier. This would allow for seamless consolidation of the product information from both sources, making the data integration process straightforward and effortless.



Picture 1

It's time to recognize that perfect conditions don't exist. Why? Firstly, because you're not alone in this. Different operational systems are developed by different individuals or often procured from external providers. Furthermore, as time goes on, the nature of our business is bound to change. What's deemed valuable today might not hold the same importance tomorrow, and as businesses expand, they naturally become more complex. Additionally, job turnover is common, leading to a constantly changing data team. Therefore, each system has its unique take on business concepts, processes, and even development styles.

In the real world, the challenges of integrating our product datasets have grown significantly due to these factors. For example, in picture 2, the CRM and ERP systems now view products from distinct perspectives: the CRM focuses on bicycles, while the ERP deals with non-motorized vehicles. Different levels of detail and, consequently, different keys are employed.



Picture 2

Coping with data complexity

Industry data models are not silver bullets

In the past, vendors attempted to establish dominance by creating pre-defined industry data models for analytical workloads. These models aimed to provide a standardized framework for organizing, storing, and processing data within specific industries, all the way down to the physical data model level. However, this one-size-fits-all approach quickly encountered limitations. Source systems did not conform to the newly imposed models and many companies had activities that deviated slightly from the model, resulting in complex integration and substantial data transformation efforts. Instead of expediting data integration, these companies found themselves doing more work. Industry data models did not turn out to be the silver bullet everyone hoped for.

Addressing data complexity through proper data modeling

Data modelers have introduced various levels of data modeling to tackle the complexity of data integration: conceptual, logical, and physical. In the vehicle dealership example, we primarily discussed data integration based on the physical data model only. However, in practice, data modelers should initiate their work at the conceptual level.

Conceptual data model

The conceptual model, often referred to as the business model, provides a perspective that the business community can readily understand. It's primary purpose is to define entities, representing objects, concepts, or things, and the relationships between them. This conceptual model operates at the highest level of abstraction, without any technical implementation involved, and focuses on aligning with general business concepts and requirements. It refrains from delving into the intricacies of data storage or access within the physical database.

Logical data model

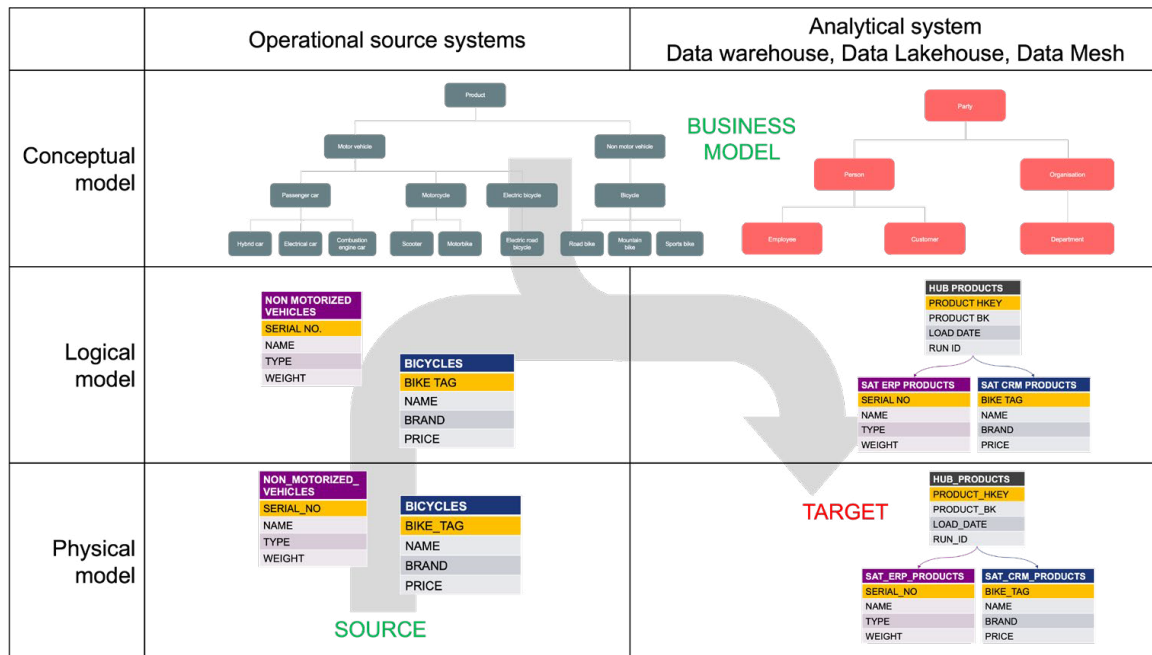
The logical data model shifts attention from business perspectives to establishing the structure for data elements that are technology-independent. This process encompasses the definition of entities, attributes, relationships, and constraints, often employing tools like entity-relationship diagrams (ERDs) or similar visual aids. The logical data model provides a precise understanding of how data elements interconnect and the rules that govern these connections.

Physical data model

When designing a database, it's important to make decisions about data storage, organization, and access. This is where the physical data model comes in. It considers factors like indexing, data types, and storage mechanisms to optimize performance and storage efficiency. Of course, all of these decisions must also adhere to the constraints of the chosen Database Management System (DBMS).

Using these three data modeling levels, organizations can easily navigate the complexity that comes with having to deal with data sets that are very diverse in nature from a growing and changing range of sources each with their own taxonomy implementation. Ideally, your organization uses the conceptual data model to design both operational and analytical data systems. In other words: the logical and physical data models of operational and analytical systems can be implemented on different levels of detail, using different business keys, but at least, they are based on the same conceptual blueprint. This will help us to translate the physical data models of the source into the physical data model for the target.

The diagram displayed below outlines the recommended approach for developing data models for a data warehouse, data lakehouse or data mesh. This process takes into account two key inputs: the physical source model and the conceptual business model. The goal is to construct a physical data model that can be used in the analytical system.



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, organizations use taxonomies to gain a better understanding of their data. The main purpose of a taxonomy is to identify, describe, categorize, and label objects based on their characteristics. For instance, bicycles, electric bicycles, and motorcycles all have two wheels, so they can be classified as two-wheelers.

Taxonomy represents the formal structure of classes or types of objects within a specific domain. It organizes knowledge, making it easier to find related information.

A taxonomy adheres to certain rules:

- It follows a hierarchical format and provides names for each object in relation to others.
- It captures the membership properties of each object in relation to others.
- It applies specific rules to classify or categorize any object in a domain. These rules must be complete, consistent, and unambiguous.
- It ensures any newly discovered object fits into one and only one category or object.
- It inherits all the properties from the class above it while also allowing for additional properties.

Taxonomy isn't just a theoretical concept. It plays a vital role in helping organizations gain a clear understanding of organizational structure, enabling effective data management governance, and facilitating the application of machine learning to detect patterns.

In practice, large organizations often must deal with different taxonomies. For example, a bicycle can be classified not only as a two-wheeler but also as a non-motorized vehicle or a light vehicle. All these categorizations are valid, and various instances may utilize different properties for classification, including class, propulsion, size, intended use, and design environment.

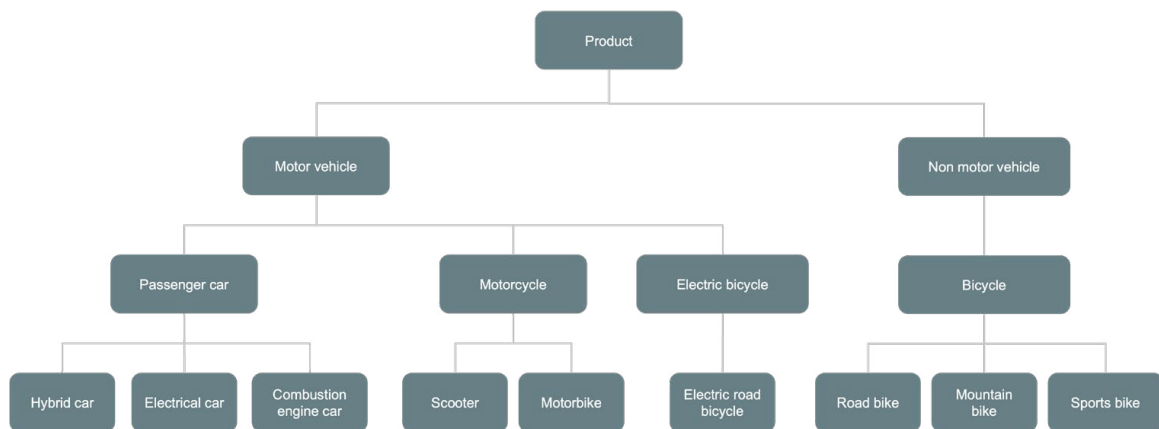
Determining the correct taxonomy for your business concepts is an ongoing process. Regularly reviewing and refining the taxonomies to ensure their relevance and effectiveness as the business environment evolves. It should incorporate input from those who interact with the business on a daily basis. This helps ensure that the taxonomy reflects the actual structure and dynamics of the organization¹.

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

Once a shared language has been agreed upon to describe the business through a conceptual data model, the subsequent step is to translate the conceptual model into a physical target data model that fits within the correct level of the taxonomy. This model should have the ability to handle various physical implementations across different sources. In essence, this involves integrating those sources through common business concepts as defined in the taxonomy.

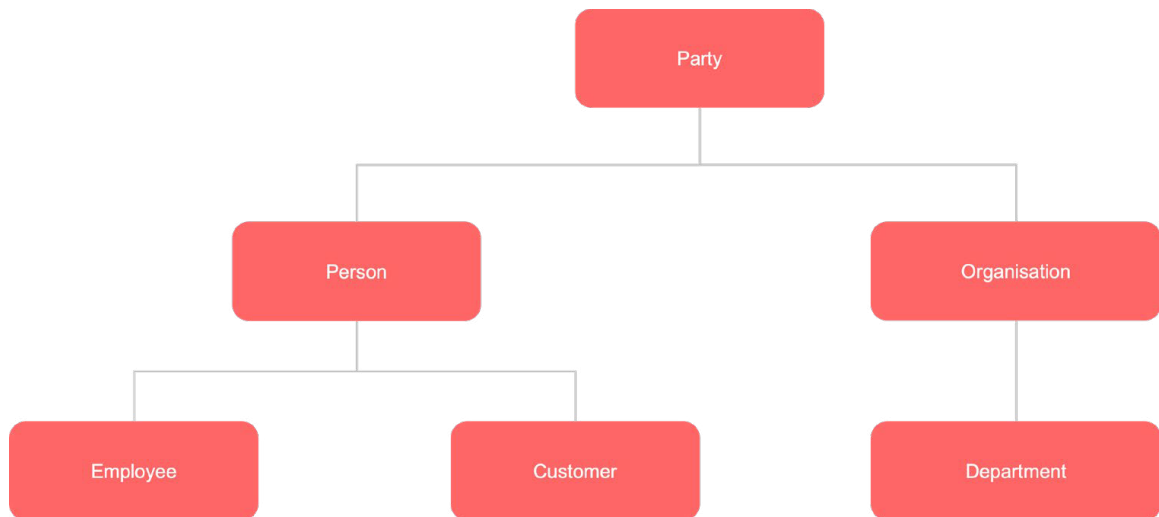
Let's illustrate this with an example: Our dealership which sells bicycles and other vehicles, wants to set up a data warehouse to gain a better understanding of the purchasing behavior of their residential customers.

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



Picture 4

Which it sells to different parties:



Picture 5

The purchase relationship represents the interaction of customers purchasing products. This conceptual model has been strategically chosen to align business requirements. As a result, business users have selected the taxonomy levels they intend to incorporate in their reports. The product hierarchy centers around products, while the party hierarchy revolves around customers. These levels will be referred to as the 'business model level'.

Conceptual model at business model level



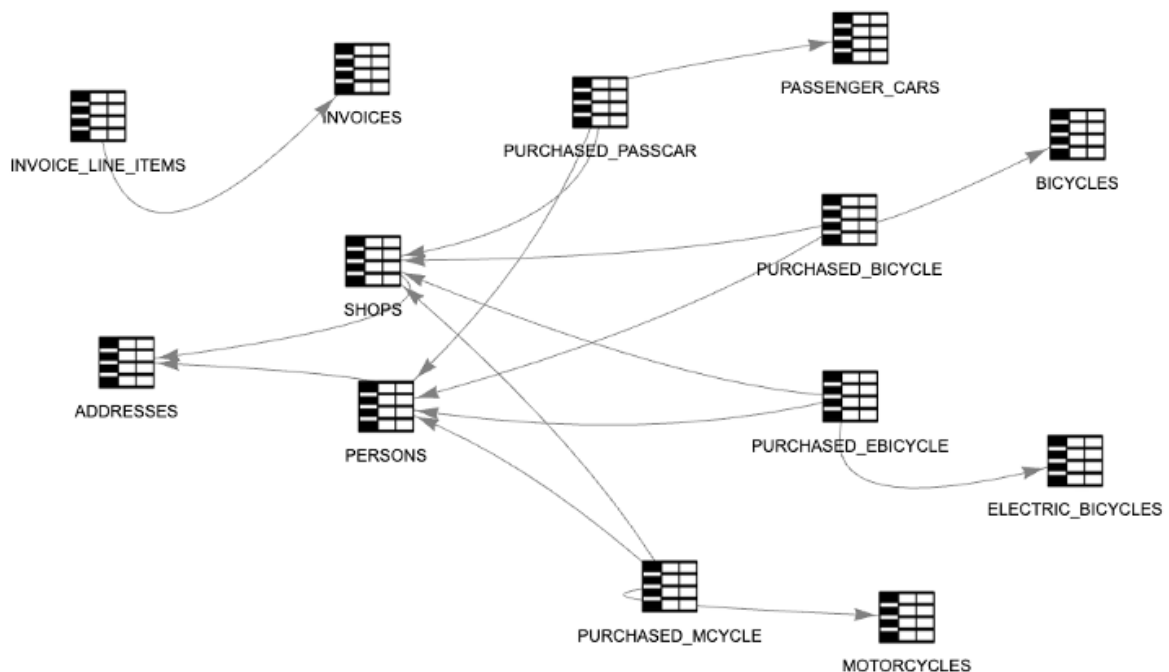
Picture 6

The master data feeding these taxonomies resides in multiple sources and is distributed across various taxonomy levels within these sources, referred to as the “source model level.” Additionally, this master data can undergo updates and replication, either from a single source (single-master system) or multiple sources (multi-master).

In our example, we are dealing with two distinct sources from different subsidiaries that employ product taxonomies at varying levels. The dealership’s objective is to analyse the purchase behaviors of residential customers on an individual basis.

Now, let’s examine the source data models and explore how they have implemented the product and party taxonomies within each source:

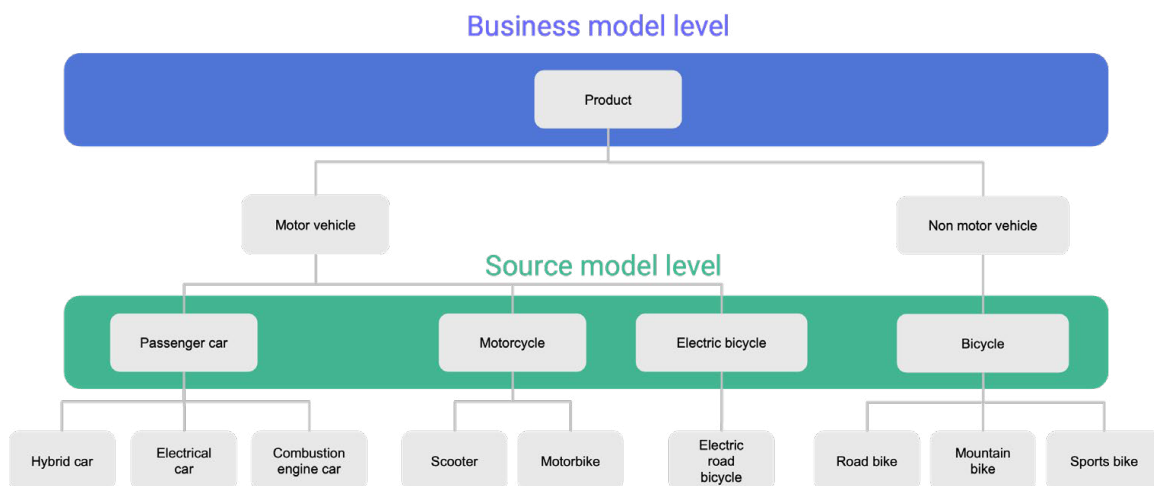
SRC1



Picture 7

The source is a B2C (business-to-consumer) source, responsible for retaining information regarding customer purchases. It tracks details such as who made the purchase, what was bought, and where the transactions occurred. Additionally, this source contains certain particulars related to the invoicing procedure.

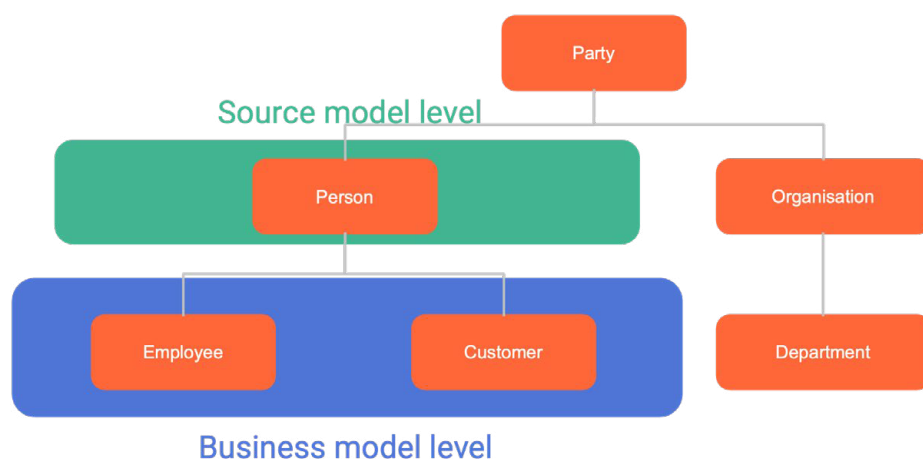
In this source model the comprehensive product taxonomy is incorporated at the third level, as indicated in green:



Picture 8

To achieve the business’s goal of generating reports at level 1 of the product taxonomy, as indicated in blue, we will need to move up multiple levels in this taxonomy when transferring data from the source to the target.

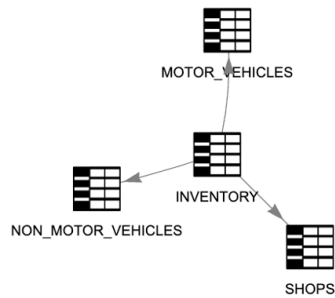
Conversely, regarding the party taxonomy, we must move in the opposite direction. The source model hierarchy from the B2C subsidiary is at level 2 of our taxonomy, whereas the business model level resides at level 3. It’s essential to address this misalignment while constructing the physical target data model.



Picture 9

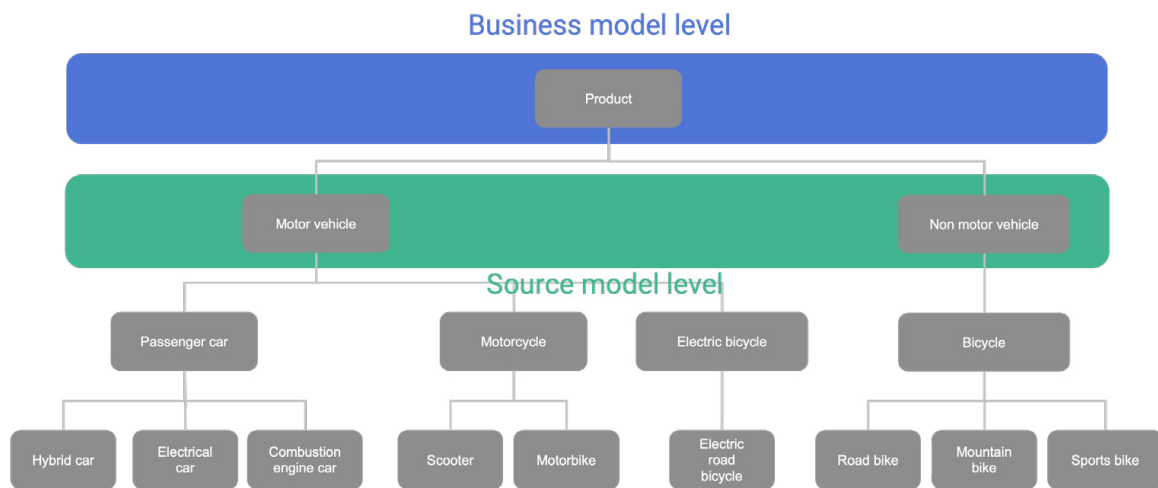
Source 2 is an ERP (Enterprise Resource Planning) inventory tracking system that contains details about the products and the inventory that is held in various shops. It does not store data about customers.

The second source system is implemented at the second level in the comprehensive product taxonomy,



Picture 10

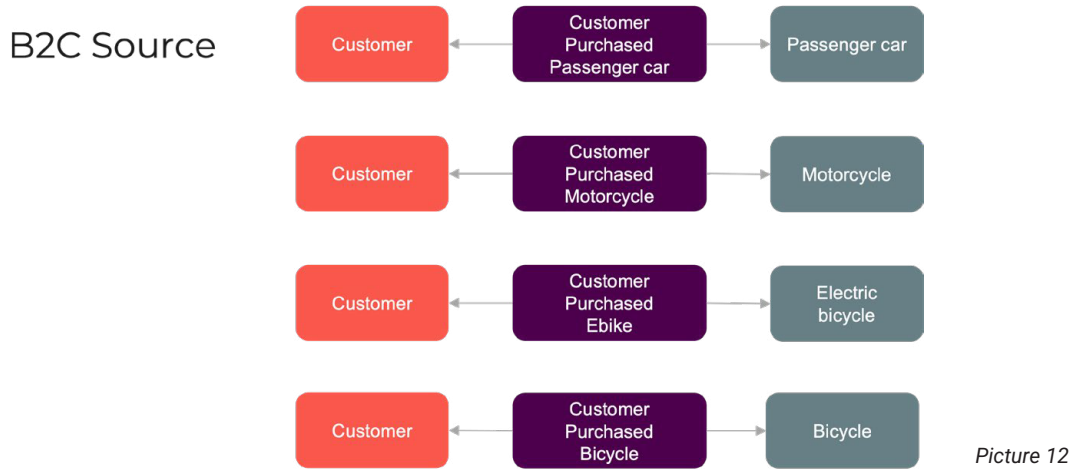
as shown in the subsequent diagram. Once more, a mismatch exists between the source model level and the business model level.



Picture 11

To generate reports, it is essential to establish a purchase relationship connecting the customer and the product.

Within source 1, at level 3 within the party and product taxonomies, four many-to-many relationships emerge:



Our dealership example has helped you gain an understanding into the conceptual model of our organization and the specifics of the physical source data models that we need to integrate into the physical data model for the data warehouse.

In the next chapter, we will talk about how to achieve this integration. We will introduce Data Vault, which is the data modeling methodology tailored for this purpose. It stands out as the sole model that offers exceptional levels of standardization and flexibility at the same time.

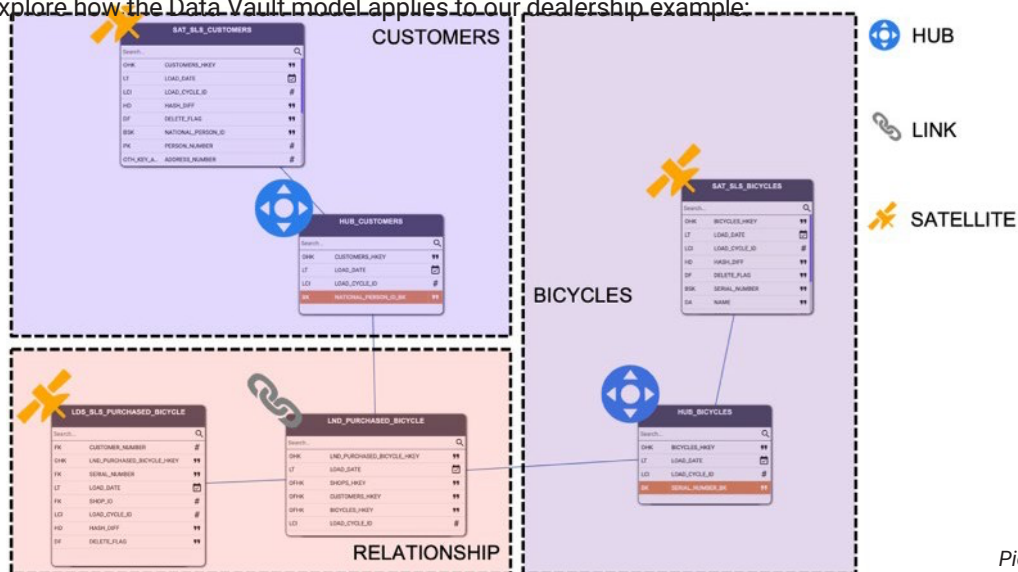
The optimal modeling language: Data Vault

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

Data Vault modeling starts with the principle that different perspectives can all be correct, while the data itself remains objective. Instead of imposing a uniform view throughout the organization, it chooses to break the data into three standard entities that everyone can understand in the same way:

- Hubs: business keys identifying core business objects such as product or customer
- Links: the relationships between Hubs.
- Satellites: storing descriptive information about the Hub or Link.

Let's explore how the Data Vault model applies to our dealership example:



Picture 13

The diagram effectively illustrates three key concepts of customers, bicycles, and the historical relationship between customers who purchased bicycles. Each concept comprises several components of Data Vault. For instance, bicycles and customers are considered business objects, and their respective business keys are captured by the HUB_CUSTOMER and HUB_BICYCLE, respectively.

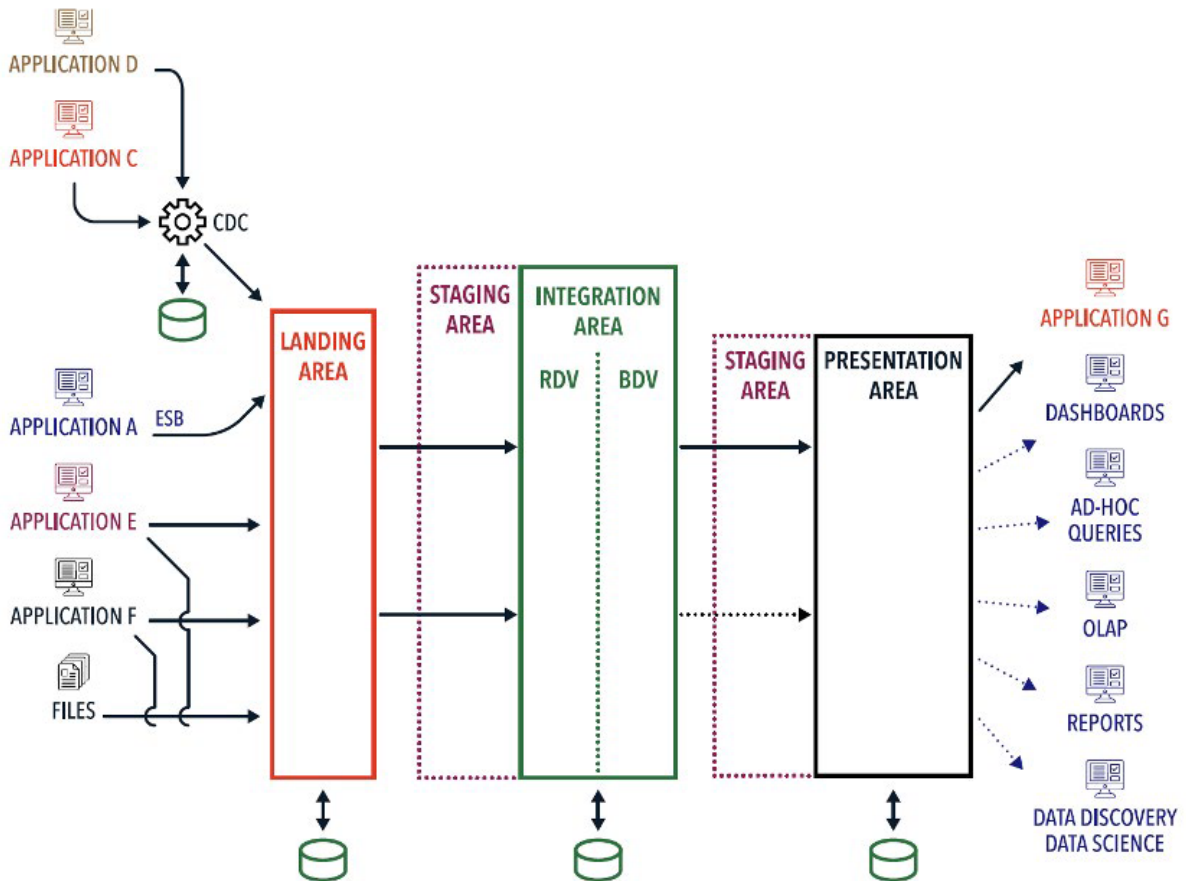
The relationship history is defined by the link connecting these hubs. Furthermore, satellites describe the hubs and links within these concepts. The customer hub, for instance, is described by satellite: SAT_SLS_CUSTOMERS.

These three fundamental entities capture the essential components of the incoming data set. This transformation enables the harmonization of diverse data types and ensures effective data integration.

However, as mentioned, the Data Vault model is also designed to handle different classifications of the same entity appearing at different levels in various source models.

Data Vault architecture

Data Vault is not just a model, it is also a reference architecture that is designed to deal with integrating/aggregating/uploading different source data models, taxonomies, and naming conventions and solving the complexity.



Picture 14

The first layer, the landing area, is used to capture the data from the source systems. The data might be delivered via change data capture (CDC), real-time enterprise service bus (ESB), direct database access, or files.

The integration area consists of two internal layers, namely:

- A Raw Data Vault (RDV) - capturing the unmodified raw data
- A Business Data Vault (BDV) – pre-processed data in a sparsely modeled layer

The difference between the Raw Data Vault layer and the Business Data Vault layer is that the first is focusing on the raw, historical, unfiltered data from the sources. The raw data describes the facts of the source system. They prove that something exists or has occurred.

The Business Data Vault harmonizes business keys/terms from the source system with the anticipated model, ensuring alignment and compliance. It is also the layer where additional business logic is implemented.

Both are modeled using the Data Vault model, which included hubs, links, and satellites.

Subsequently, the third layer, the presentation layer, offers information marts that deliver information to applications, encompassing dashboards, reports, and other formats.

With the inclusion of multiple information marts, diverse business perspectives on the same data are generated. These varying viewpoints coexist harmoniously within the architecture, each recognized as valid versions of the truth.

There is a difference in the stability of the “artifacts” in these individual layers: the business perspectives with the included business rules tend to evolve over time as the business adapts to changing markets and other factors, while raw data is more stable.

Data Vault supports multi-temporal solutions. Data Vault does not only provide standard patterns for implementing them but also permits the definition of many timelines in parallel, allowing users to switch between them as needed.

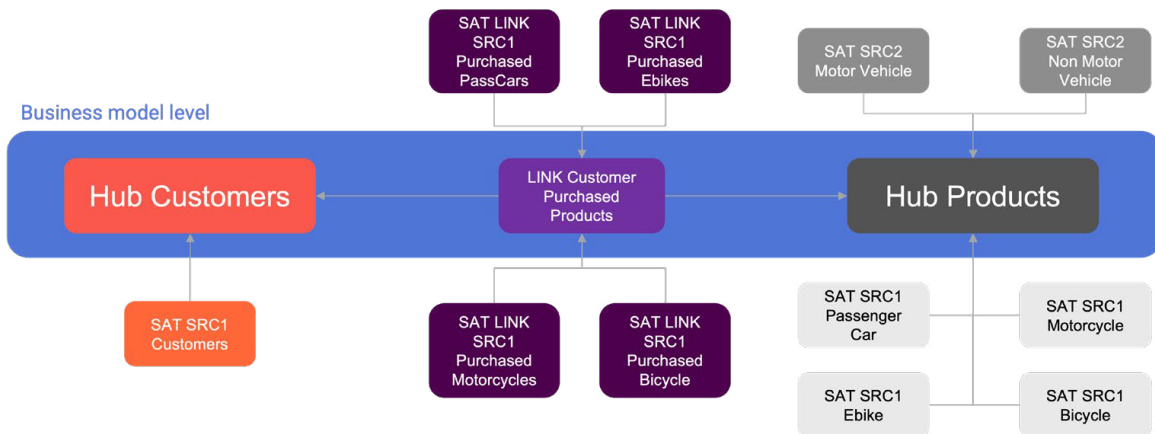
Furthermore, Data Vault boasts several other advantages. It facilitates distributed solutions, enabling the enterprise data warehouse, lakehouse, or mesh, to span different clouds or regions in a multi-cloud setup. Additionally, it seamlessly bridges on-premises and cloud environments.

Data Vault has been used to develop fully auditable solutions where every record can be traced back to an individual delivery, and every report provided to an information user can be reconstructed.

It also accommodates the implementation of cell-level security patterns and the deletion and reduction of records, for example, consumer records, as regulated by the General Data Protection Regulation (GDPR). It is not only possible to delete consumer records, but also to reduce them according to the data needs.

Returning to our example: The source models presented can be transformed into a Data Vault model. Remember that we wanted the Data Vault model to resemble as much as possible the business conceptual model we defined above, whilst also ensuring the data coming from various source systems has a place in the resulting Data Vault model.

Target Data Vault model



Picture 15

The basic structures of Data Vault help us to achieve this: The hubs and links represent the business model, while Satellites capture the data from the source systems. Let's delve deeper into how this works.

Hubs contain business keys. These business keys identify the underlying business objects in the business taxonomy (customer, product, etc.). Therefore, the hubs and links are as close as you can get to the business taxonomies. Identifying the correct business key to describe a business concept is a vital part of Data Vault modeling, as these business keys are not necessarily the primary keys of the source system.

However, there are important deviations: the hubs and links reflect the business keys and not the business objects. For instance, if a product has duplicates in the source dataset, resulting in multiple business keys, all these business keys would coexist within the hub. The granularity of the hub follows the business keys, not the business objects. Additionally, it is possible that the preferred business key may not exist in a particular source system, and a less desired one might be used to identify and integrate the source data.

To map our source data models to the target model in alignment with the business model, the first step is identifying the correct business keys. For instance, cars are identified by their vehicle number, while bicycles are tracked using serial numbers.

In the business model, both bicycles and cars fall under the category of "products." Consequently, we opt to load their respective business keys into the same hub, a technique we refer to as "hub grouping."

- We group all business keys related to the Product taxonomy into the "HUB Products."
- Similarly, we group all business keys associated with the Party taxonomy into the "HUB Customers."

Proper business key management is important when you group business keys into a HUB. Depending on the case, you may need to use a Business Key Collision Code (BKCC). This code will help ensure that any unexpected collisions for business keys (i.e., duplicate keys for dissimilar records) are handled correctly.

In the data modelling process, having dealt with the product hub, we still need to identify the correct business key for the customer object. In the source, the data is modeled at the second level of the party taxonomy, encompassing persons. Persons consist of both employees and customers whereas the business model distinctly emphasizes customers. Customers can be identified by various means, such as their customer loyalty card ID or government ID, while employees are typically distinguished by employee numbers. This is a typical example of a case where multiple business keys are caught up in one single source object. The solution to align the resulting Data Vault model with the conceptual business model is to split the source object. This typically involves some sort of pre-staging area to direct the source data correctly.

Finally, links contain the unique relationship between Business Keys.

- The business conceptual model defines a purchased relationship between Products and Customers.
- This results in the LINK Customer Purchased Products

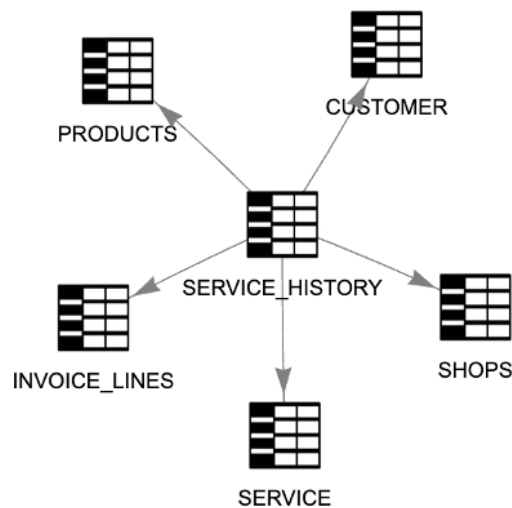
The very definitions inside the Data Vault standard ensure that the business model is represented inside the resulting physical target model utilizing hubs and links.

The subtypes of the business elements in the product taxonomy at level 3 for source 1 and at level 2 for source 2 become satellites of the product hubs, ensuring that no source data gets lost in translation.

Data Vault can deal with different naming conventions too. Different names are not only applied simultaneously but can also change over time (e-bike, public bike). The Data Vault model is resilient to change.

Speaking of change, everything that is sold will not automatically be considered a product.

Imagine that our dealership also needs to integrate a third source system, primarily focused on services offered, such as car repairs or bicycle rentals. Notice the link between service, shops, products, and invoices (service history) and the separate service hub.



Picture 16

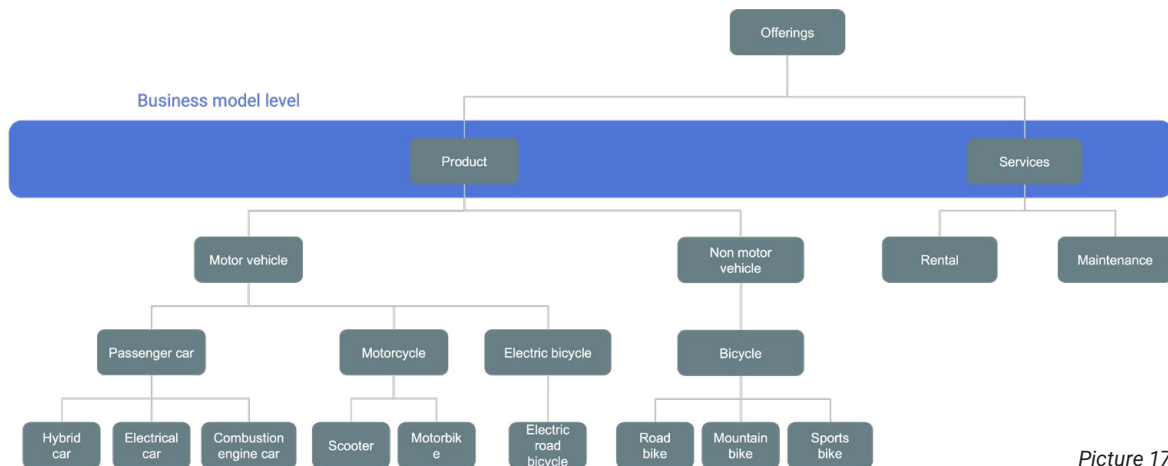
We need to decide as to how this new source model will be transformed into the Data Vault target model. Especially because some common concepts like shops, customers, and invoices are already present in the previous sources.

Source 3 introduces a new concept: services. To determine how to integrate this concept and fit it into our daily processes, we must engage in conversations with key stakeholders in the company.

It appears that there are 2 options to integrate the services into our existing business model:

1. We consider them another instance of a product and integrate them into the product hub
2. Services and products are too different, so we add a separate service hub.

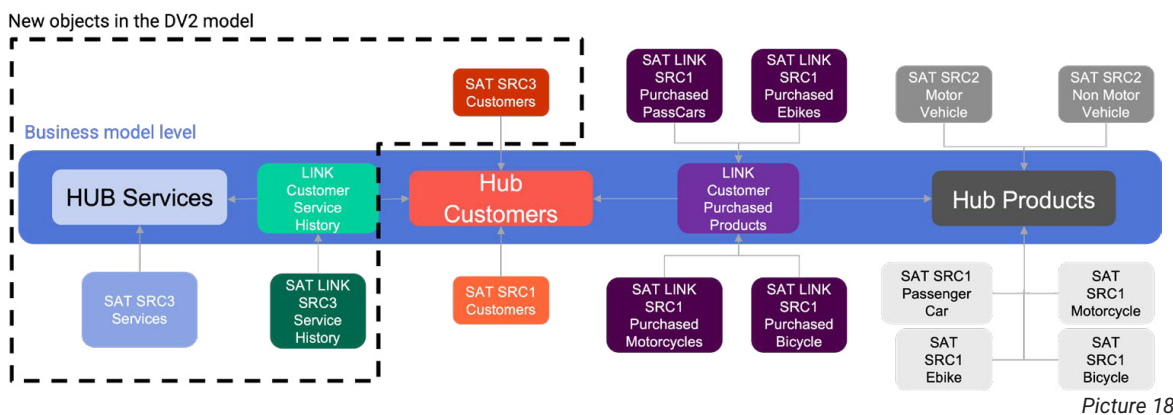
Suppose that both are handled by separate sales teams and have different timing in the customer journey, the second approach seems more apt. Consequently, the business model now reflects both products and services, which are positioned at the second level in our updated taxonomy.



Picture 17

The Data Vault model, as a result, has the service hub added separately, with its own link to the customer hub. Additionally, a new satellite from source 3 has been added to the customer hub. This demonstrates the flexibility of the Data Vault model in accommodating changes. We can incorporate an entirely new concept seamlessly by determining its appropriate placement in the business taxonomy, identifying the relevant business keys, and integrating it into the existing Data Vault model without altering any previous work.

Adapted Target Data Vault model



Picture 18

Automating multi-source data integration

We explained that Data Vault can help to overcome the challenges of integrating various sources, and various taxonomies, and mapping them to a desired target model. But when enterprises start integrating more than 20 data sources, they are dealing with capacity problems:

1. It is impossible to get the total picture or build the conceptual business model or target model manually
2. Managing the integration for each source, technology, and data type becomes an insurmountable task.

This is why organizations turn to data automation to help them manage the vast volume of datasets they need to integrate.

Data automation involves the ability to collect vast amounts of source metadata and enrich it, transforming this metadata into valuable business outcomes. The more metadata you can process, the more automated the process becomes. It's like comparing a copper UTP cable to a fiber optic cable, with the latter offering greater bandwidth and speed.

In terms of code output, data automation comprises the automation of:

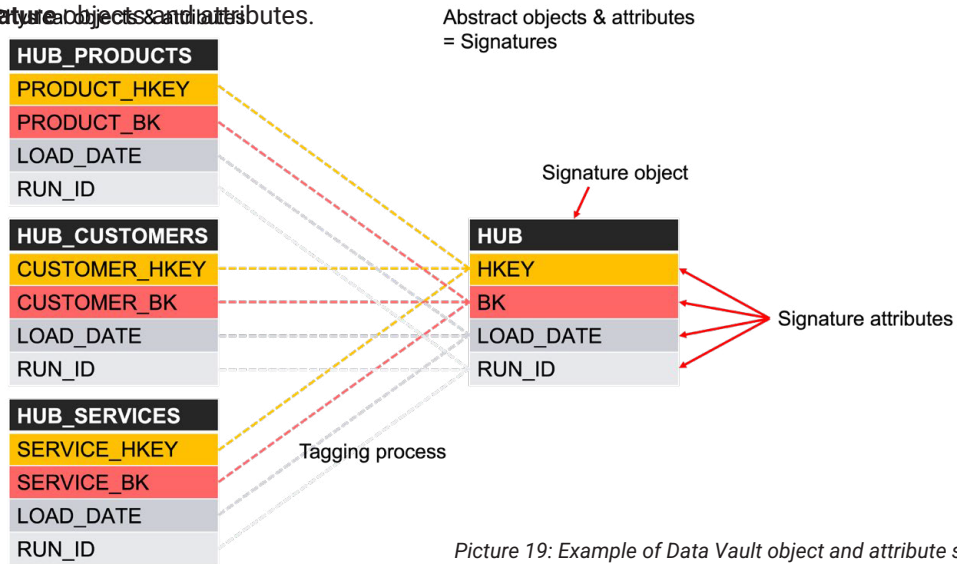
1. The physical target data model (usually in DDL statements), also known as the relational model,
2. the integration code (DML statements) to load the data from source to target, also referred to as transformational model,
3. and the automation of the workflow code (typically in Python) to orchestrate the data loads.

Without automation, not only will productivity be limited, but equally important, there will be differences in the delivery of loading procedures. When quality in mass production is defined to be the deviation from the expected quality, quality in data warehousing is defined as the deviation from the pattern. There are only a few patterns in how hubs, links, and satellites are produced. Deviations from these patterns complicate the solution, add complexity to documentation and test cases, and in return, complexity increases the risk of failure.

However, there are a few prerequisites before you can automate:

1. Object types must have a single function, the more functions, the more loading pattern combinations, the less repeatable the patterns are, and the smaller the use case for automation. This prerequisite is met by using Data Vault as the data integration model. Each object type (Hub, Link, Satellite) has only 1 or 2 functions.

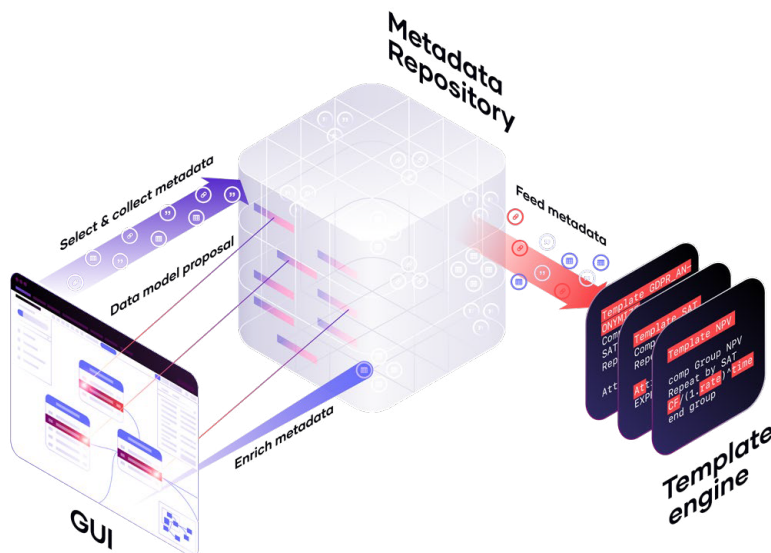
- You need an unambiguous relation between the source and the target: The Single Version of the Facts (as the data exists in the source) is stored in the Raw Data Vault Layer.
- The automation of loading logic is only possible at a certain level of abstraction rather than at the detail (physical) level. Data Vault provides this abstraction through object and attribute types (Hubs, Links, Satellites, business keys, historical attributes, etc. see picture 20) which we call **signature objects and attributes**.



Picture 19: Example of Data Vault object and attribute signatures

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

Essentially, the solution requires three key components: a smart metadata repository, built-in automation templates, and a proper GUI (Graphical User Interface) for data modeling. Additionally, these components need to be set up in the correct configuration. In the following chapter, we will demonstrate the integration of these components by using our dealership example with VaultSpeed.



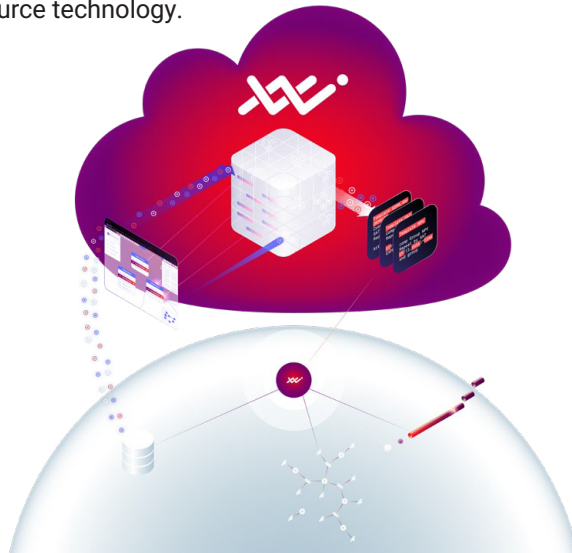
Picture 20

Navigating the automation process with VaultSpeed

Getting your Data Vault model up and running is like solving a puzzle. Without assistance, it will take you a lot of time. Laying down the first pieces of the puzzle is often the most challenging because you lack a starting point. VaultSpeed will already have assembled the biggest portion of the puzzle, leaving you to concentrate on filling in the remaining pieces.

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

To aid in assembling this puzzle together, VaultSpeed requires input. Automated data transformation relies on metadata. Therefore, to automate the target model, we must initially collect the source metadata. VaultSpeed simplifies this process by delivering a client-side agent capable of harvesting metadata from any source technology.



Picture 21

In this example:

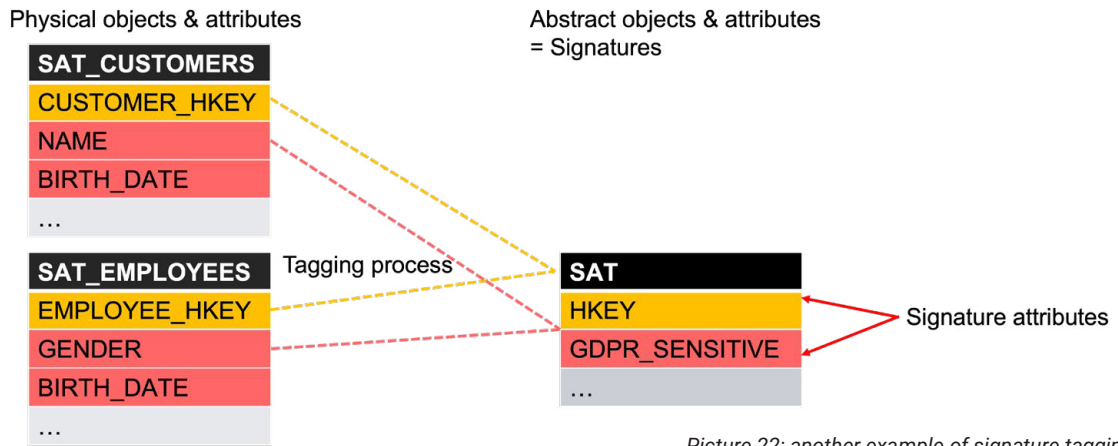
Data Source src1: the source that captured B2C sales data - as shown in Picture 7

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

This harvested metadata is safely stored in a smart metadata repository. It lets you group the metadata into 'signature groups'. They can be applied across all metadata levels, including schema, object, and

attribute levels, facilitating abstraction across various physical objects. The Data Vault standard uses standard signatures like hubs or business keys, another example is that we create a business-driven signature group for all objects or attributes that contain GDPR-sensitive data (Picture 22).

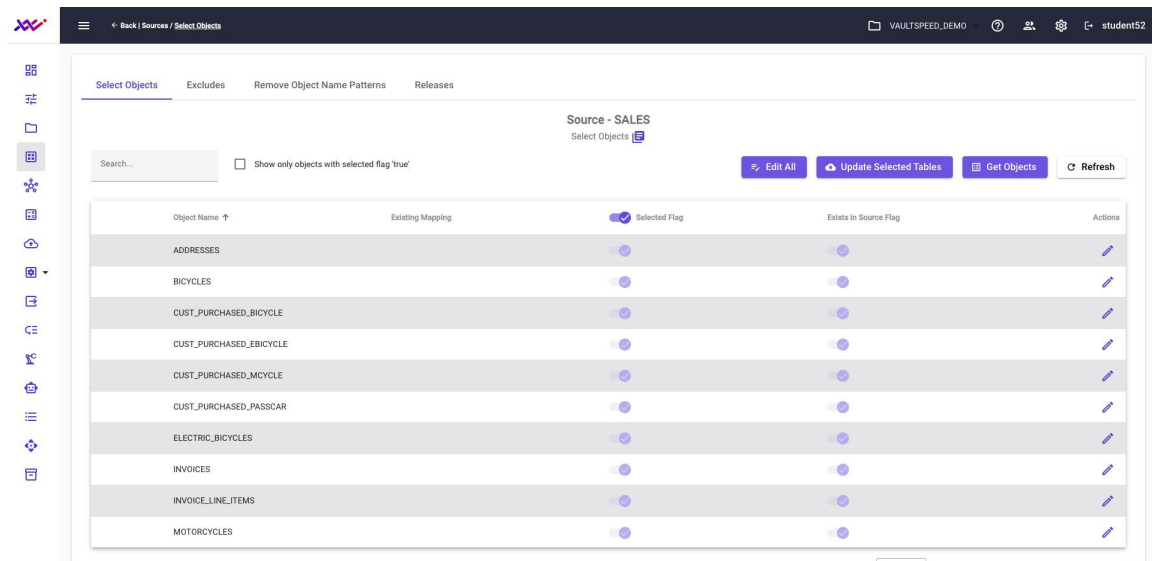


Picture 22: another example of signature tagging

Signature tagging serves as the connecting thread between the physical and abstract world. It enables the application of repeatable logic.

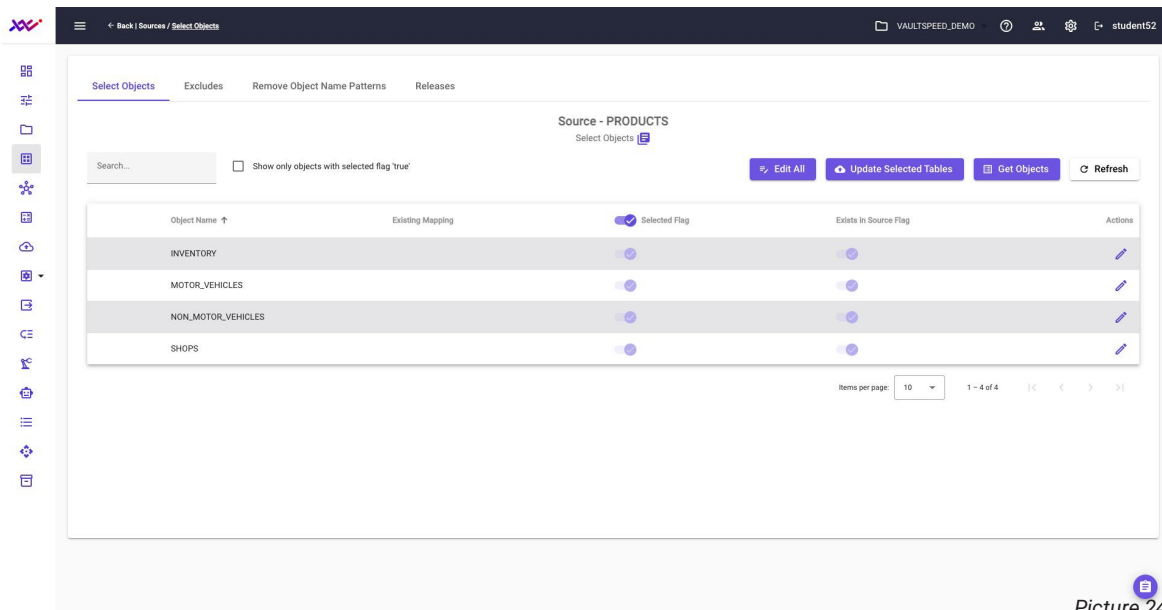
This repeatable logic is implemented through what we refer to as automation templates. The logic they contain can cover a wide range of functions, including data integration logic (such as Data Vault), business logic (like calculating total vehicle sales and service revenue), and testing logic (to verify if A equals B). These templates use abstract signature components instead of physical ones to achieve a higher level of abstraction. VaultSpeed offers pre-built Data Vault templates to save on costly repetitive template building and testing. The last thing you want is data error automation.

Below you can see the object selection screen for our B2C source. Metadata is automatically harvested from the source, and it is up to the user to select relevant source objects for integration.



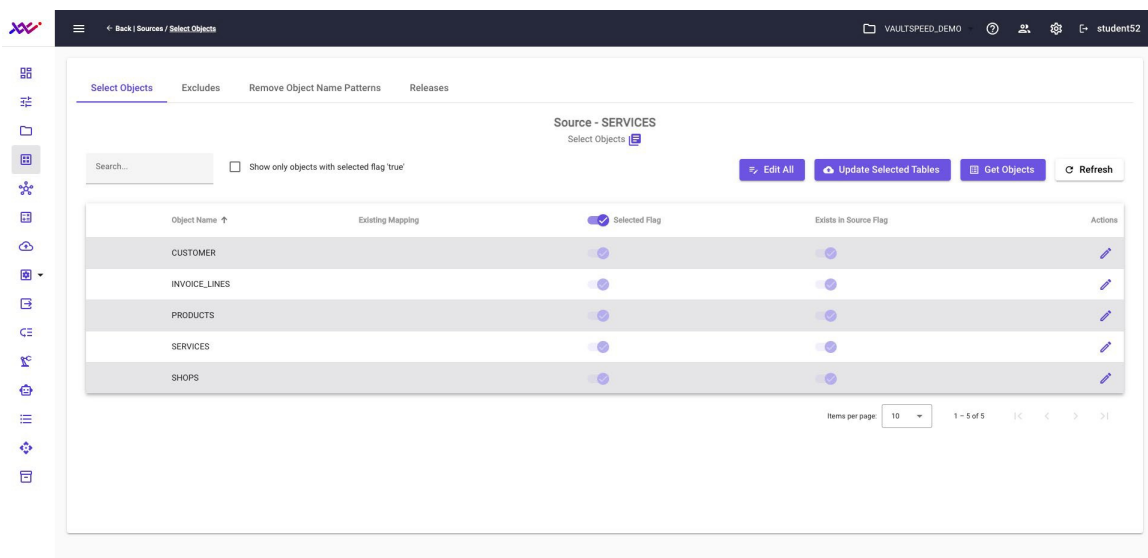
Picture 23

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



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.



Picture 25

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

The next issue VaultSpeed addresses is how to map a large amount of source metadata into the pre-built automation templates.

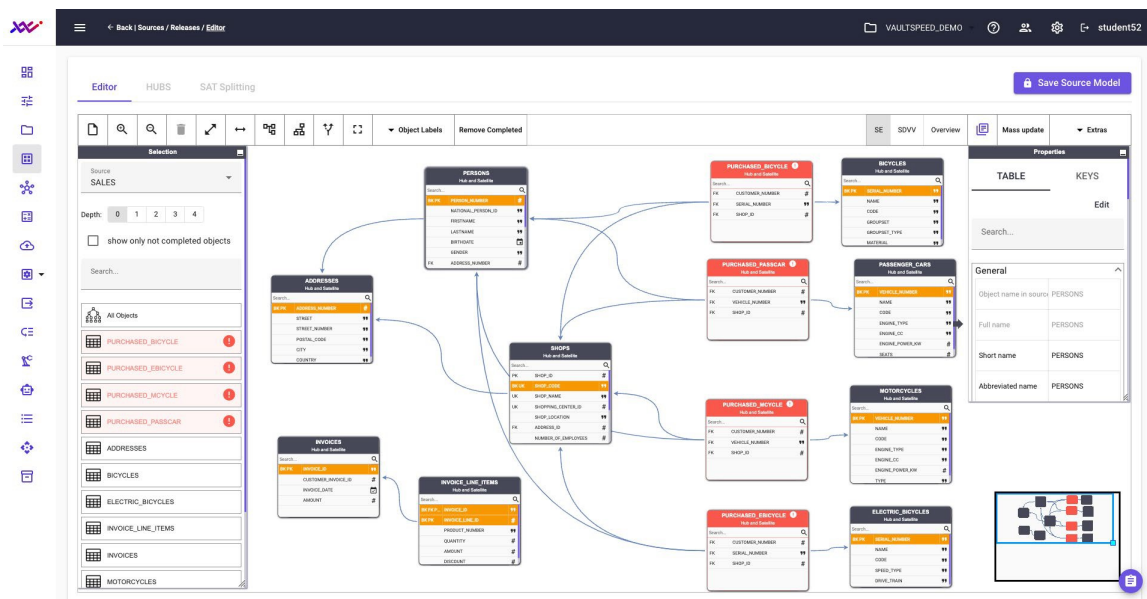
First, our metadata repository is backed by a smart rule engine, which draws assumptions from harvested metadata to propose a solution for the physical target data model.

Second, VaultSpeed's graphical user interface (GUI) comprises a comprehensive data modeler to accept, correct, or enrich the proposed solution, the final decision rests with the user.

This toolset allows us to model the raw source metadata into the target Data Vault model using the business model as our guide.

Modeling source 1

VaultSpeed shows a model based on the metadata we captured in step 1. This model is a proposed solution to our Data Vault puzzle. We'll highlight some modeling examples in the remainder of this section.

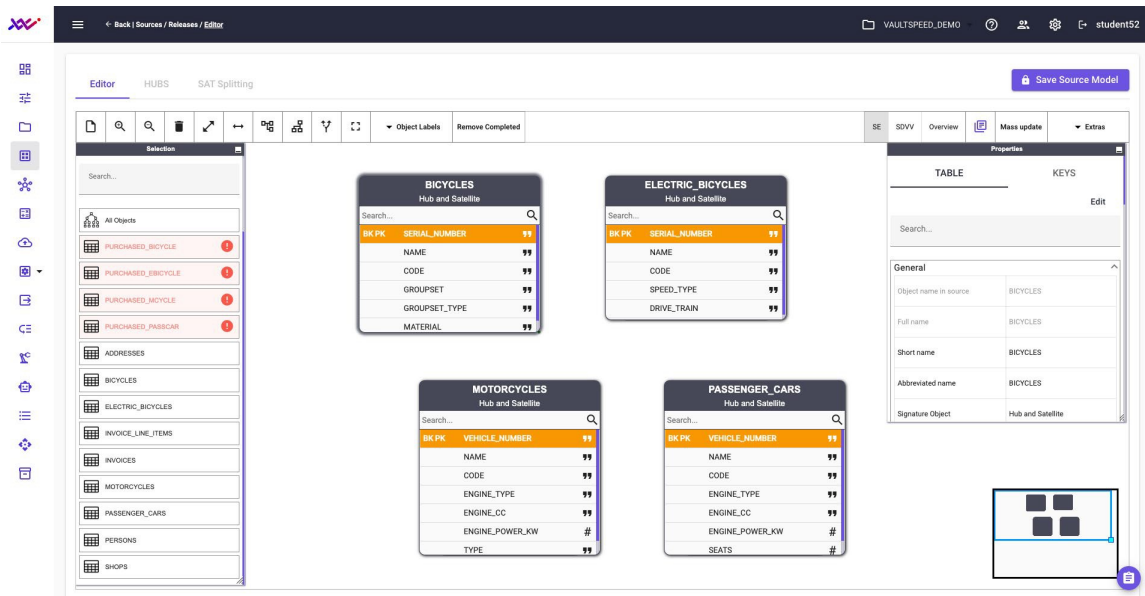


Picture 26

Accept

Within VaultSpeed's user community, it is common for more than 70% of the modeling proposals to gain approval. Consider aspects like multi- or single-master configurations, default parameter values, naming conventions, object types, CDC (Change Data Capture) settings, data quality configurations, and more.

A clear instance where VaultSpeed's proposal aligns with reality is in the object and BK settings for bicycles, passenger cars, motorcycles, and e-bikes. Indeed, all four objects should be modeled as a hub and satellite, and the business keys are indeed the serial numbers or vehicle numbers.

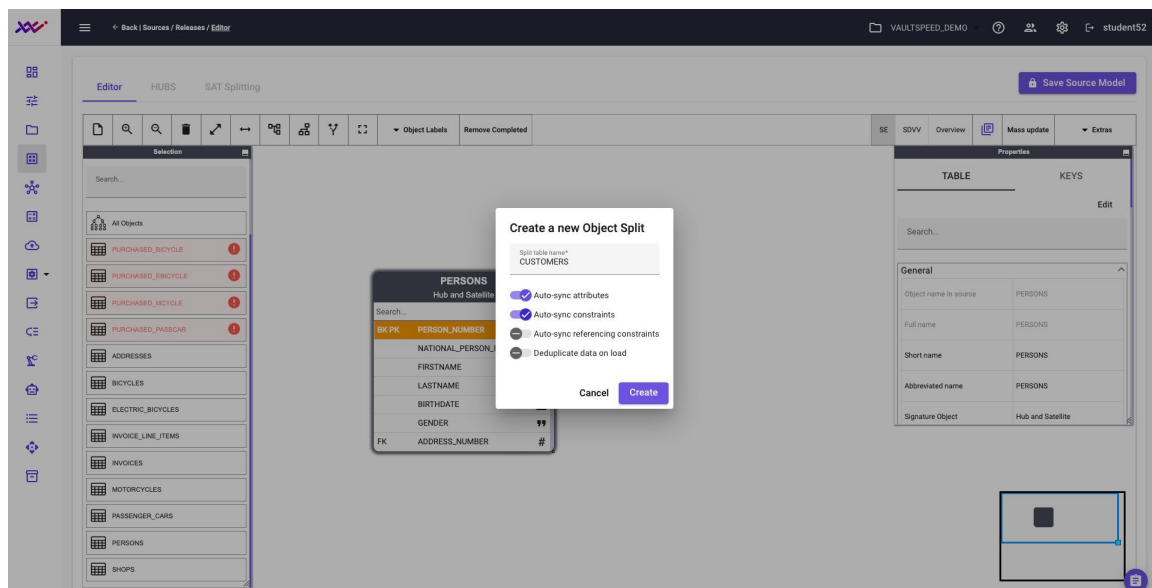


Picture 27

Correct

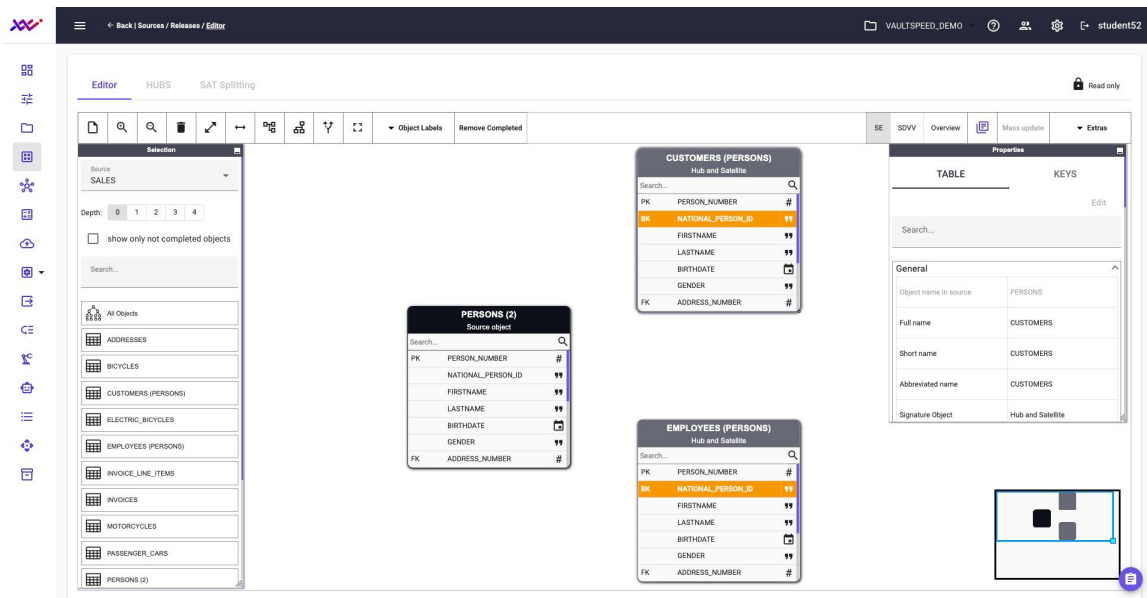
The persons source object was on level 2 of the party taxonomy, it combines employee and customer data. We need to split this object to drill 1 level deeper in the party taxonomy.

In VaultSpeed, you can execute a source split by right clicking the source object and creating a new object split.



Picture 28

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



Picture 29

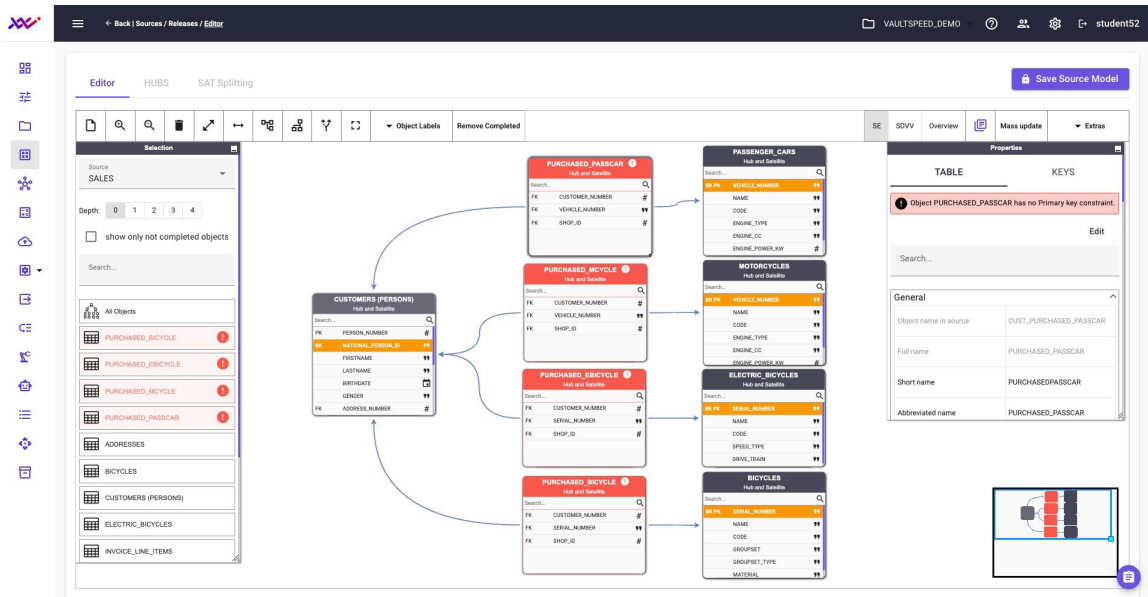
You can see that we separated both customers and employees as they are different business concepts in our party taxonomy. Also, you will notice that we corrected the business key, the `natural_person_id` fields are now selected to serve as the business key for the customer and employee hubs.

Enrich

Finally, we show an example of where we need to enrich the harvested metadata. This is needed in order to enable VaultSpeed to derive a working target model.

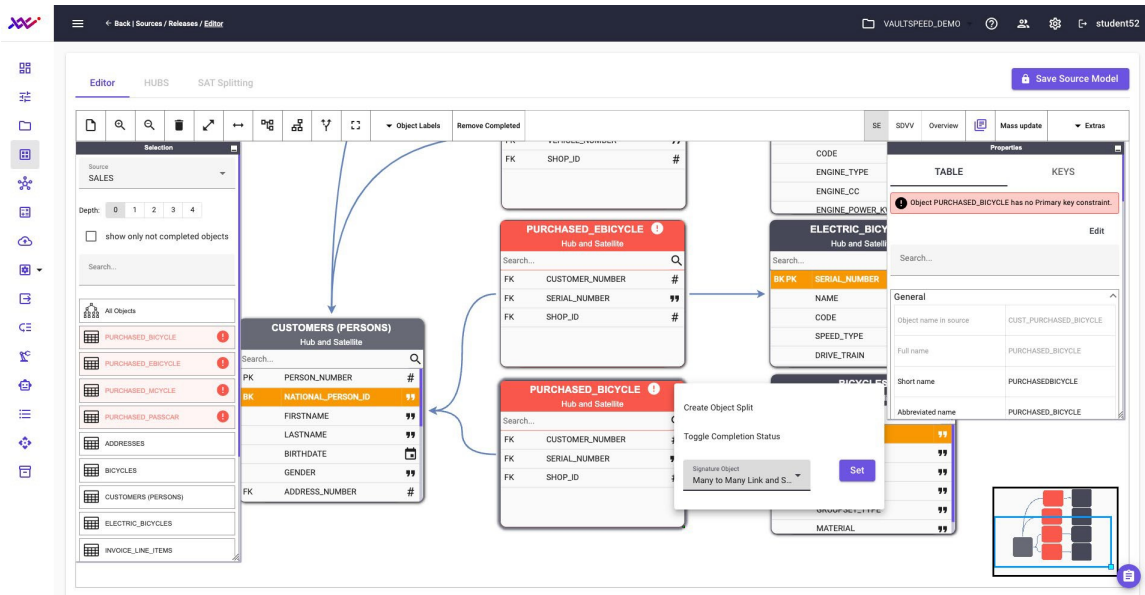
Picture 30 shows the source editor and some of the objects from src1 loaded on the canvas. The exception handling in the tool is apparent. Four objects are highlighted in red and require attention before the code can be generated.

When you drag an object into the canvas, VaultSpeed will show that object and its related objects, depending on the depth level you set in the selection menu.



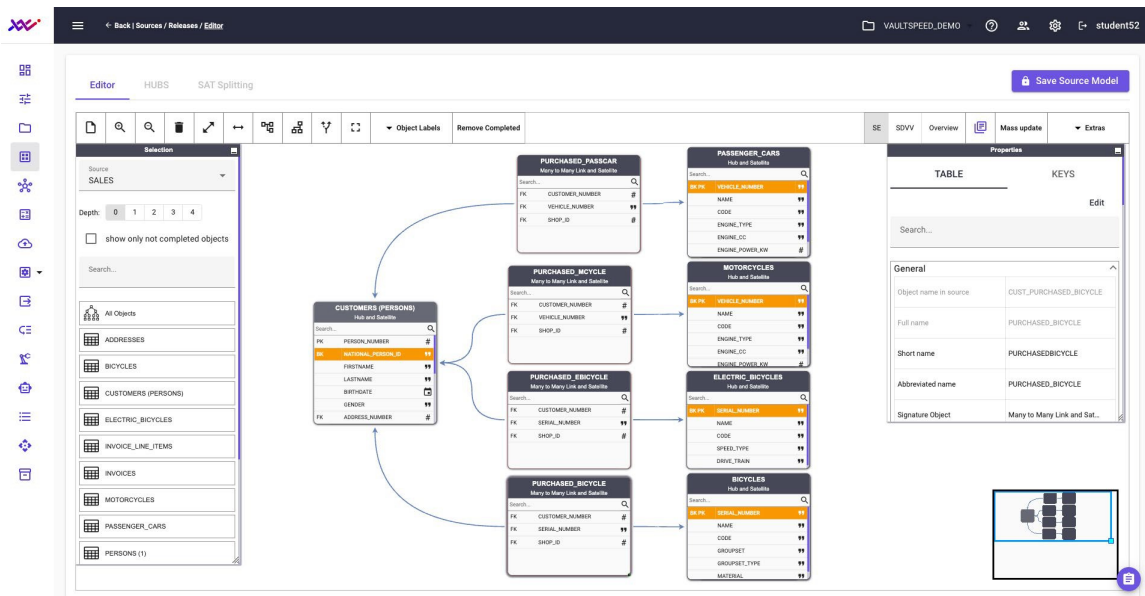
Picture 30

We zoom in on the purchased_bicycle relationship with the related objects bicycle and customer. It is highlighted in red because it does not have a primary key. As this is a many-to-many relationship, we must define it by right-clicking on the object and selecting the correct signature object type being a many-to-many link and satellite.



Picture 31

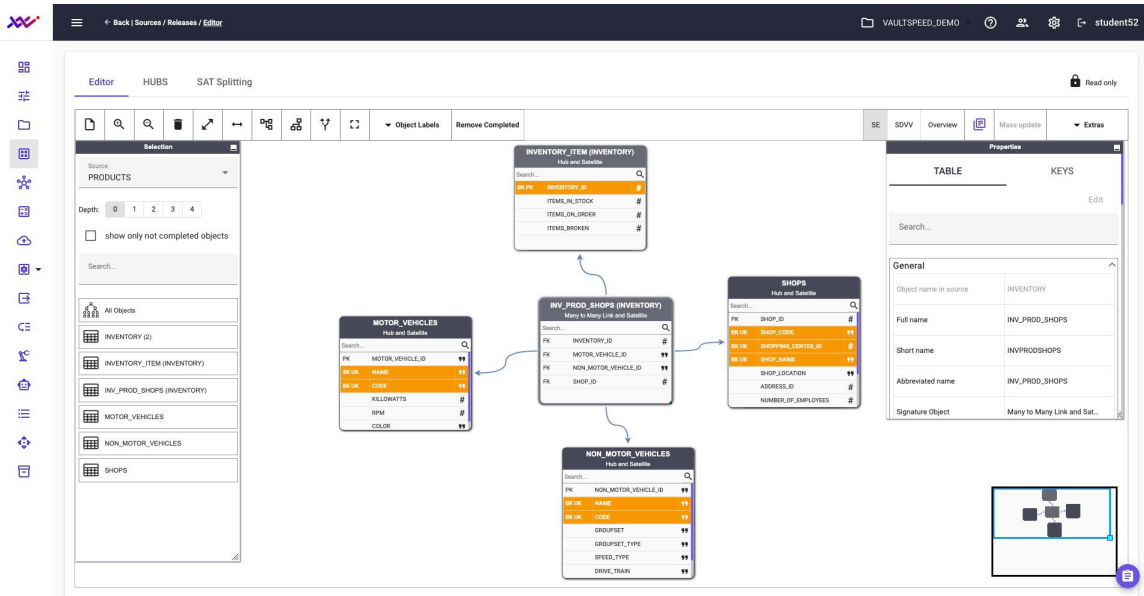
After defining all the source many-to-many purchased relationships objects as many-to-many links, you get the result shown in picture 32. This section of our model is now ready to be integrated into the target model. However, it still needs to adhere to the taxonomy level we had set for the product hub, we will do that in one of the next steps.



Picture 32

Modeling source 2

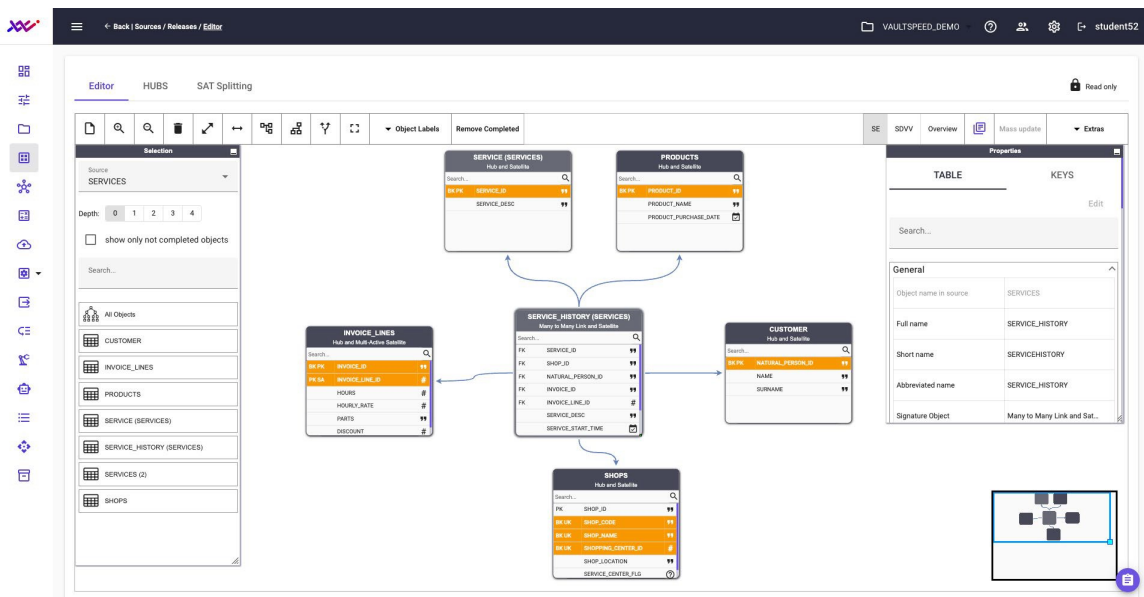
In the second source, the ERP-like source that contains inventory data, we follow the same approach. A great example of VaultSpeed's flexibility is the ability to change proposal generation settings at source level. For example, the USE_SOURCE_UK_AS_BK parameter influences the proposal by using unique keys as business keys by default, resulting in less manual effort to correct the source model.



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

Modeling source 3

Finally, the same approach is used for src3 with the SERVICE source data, resulting in the following 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 ▼ ✕

Cancel Create

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...

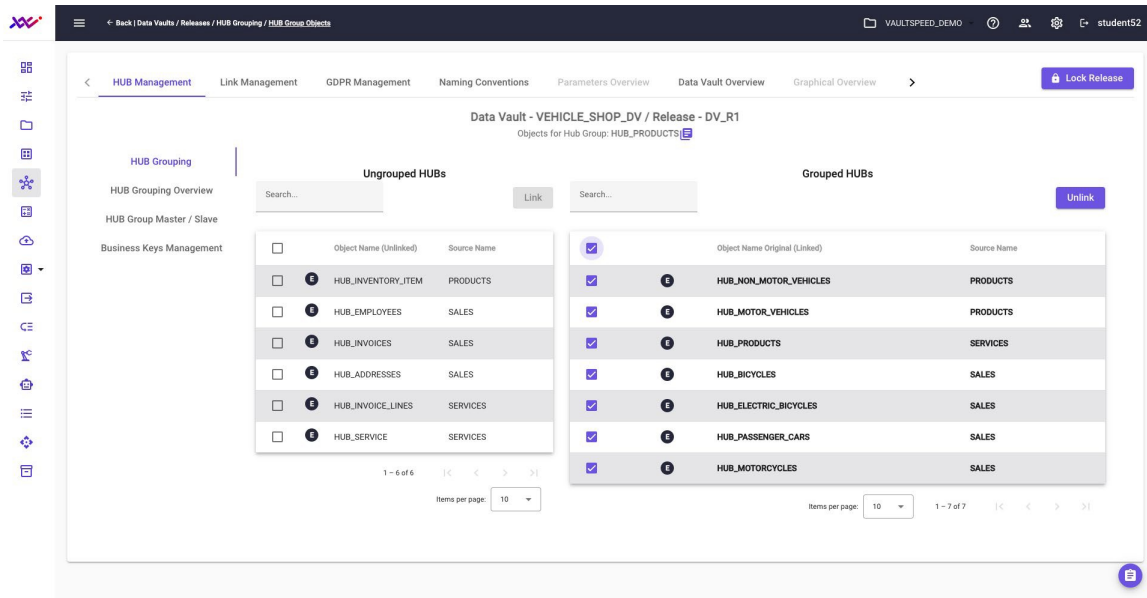
|  | Name ↑ | Release |
|---|----------|--|
| <input checked="" type="checkbox"/> | PRODUCTS | 2 - release 2 - split of inventory BK ▼ |
| <input checked="" type="checkbox"/> | SALES | 1 - release 1 ▼ |
| <input checked="" type="checkbox"/> | SERVICES | 4 - add products take 2 ▼ |

Cancel Create

Picture 36

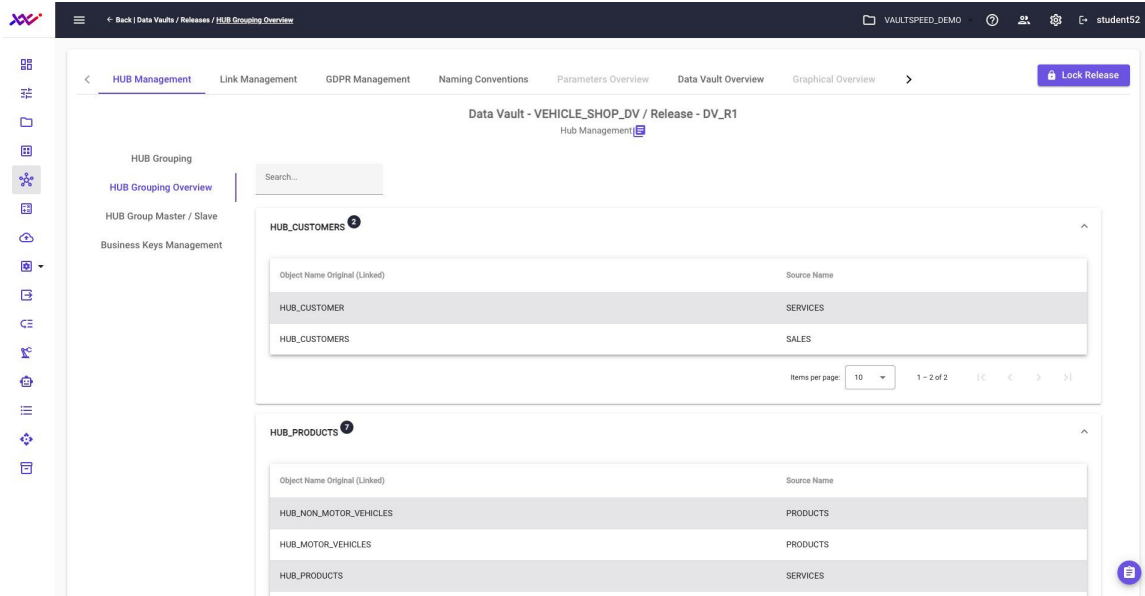
The next step is to map our target model in alignment with the business conceptual model to ensure we can group all the product-related data as satellites on the product hub. This is where we can use the VaultSpeed hub group management screens. Simply select the 'source hubs' from the left side and link them to the product hub group.

As stated in the previous chapters, integrating the SERVICE business keys with the PRODUCT business keys into the same hub is avoided due to the different semantic meanings of products and services provided.



Picture 37

We repeat the same exercise for our customer data. The hub group overview shows the two hub groups that are central to our example: one for products from src1 and src2 and one for customers from src1 and src3:



Picture 38

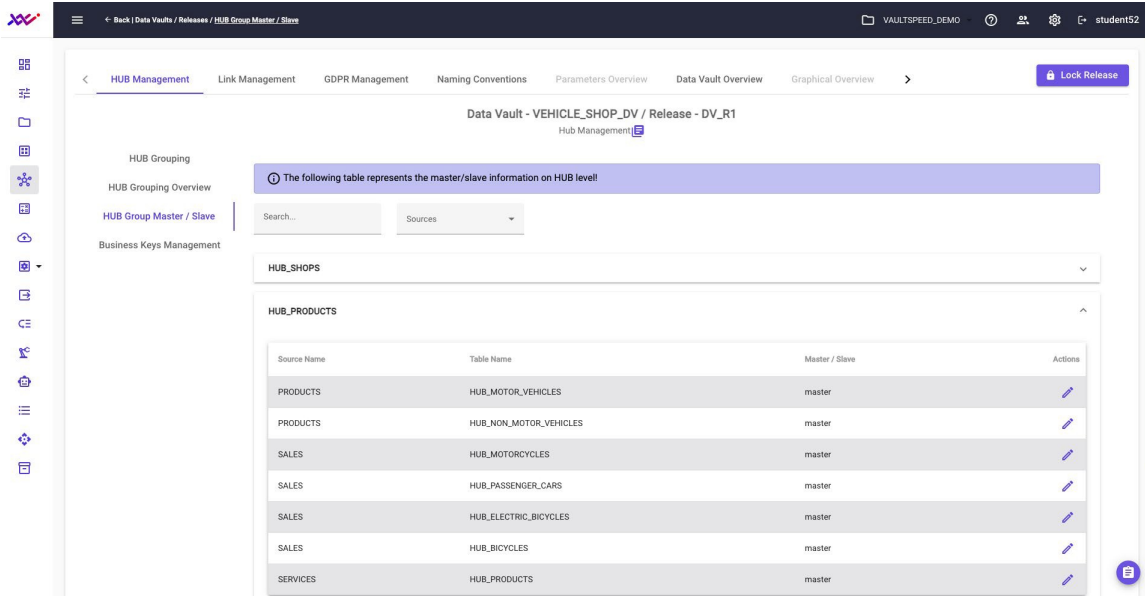
The proposed integration is designed to accommodate both technical and business considerations.

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:

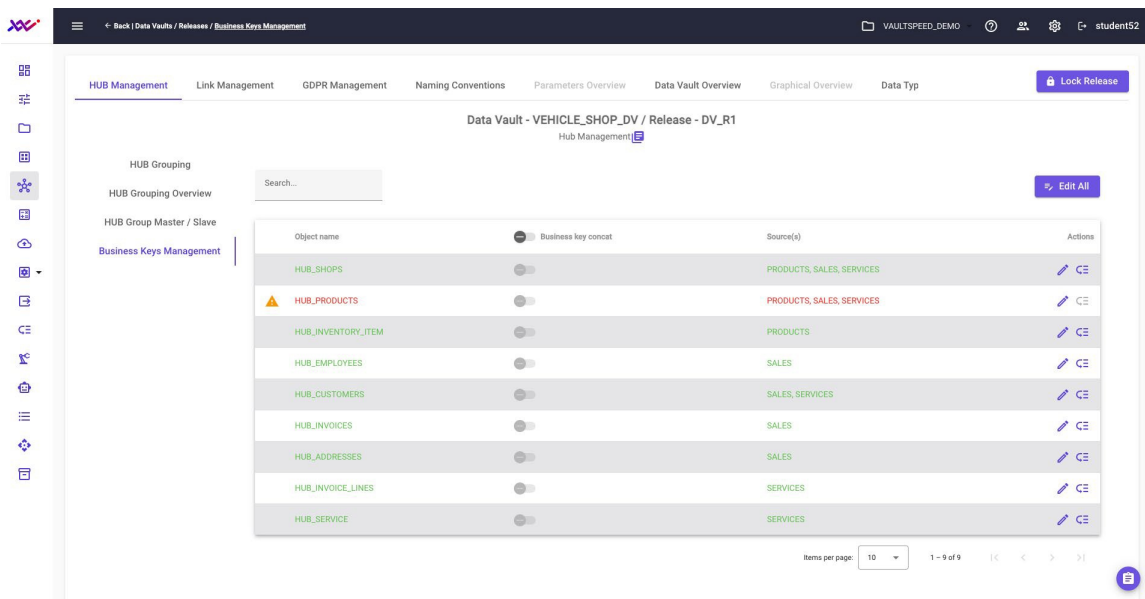
1. All sources contain master data.
2. One source is considered the primary/master, and others are secondary/slave.
3. A more intricate combination of the above, offering various options.

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.



Picture 39

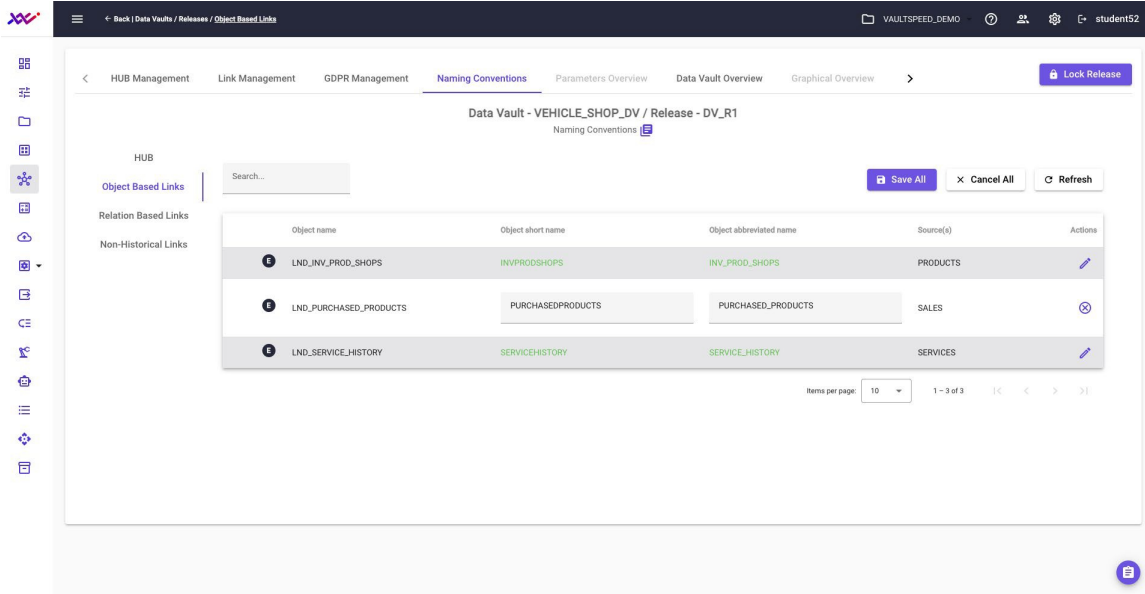
We should create a concatenated key for the Product Business Element's Business Key. This will help to simplify the Hub Business Key by consolidating the key attributes into a single attribute. Rather than using different key names for different multi-master product objects, we can concatenate the content to create a unified hash key. VaultSpeed will indicate where you might need to apply this, enabling you to enhance the Data Vault target model.



Picture 40

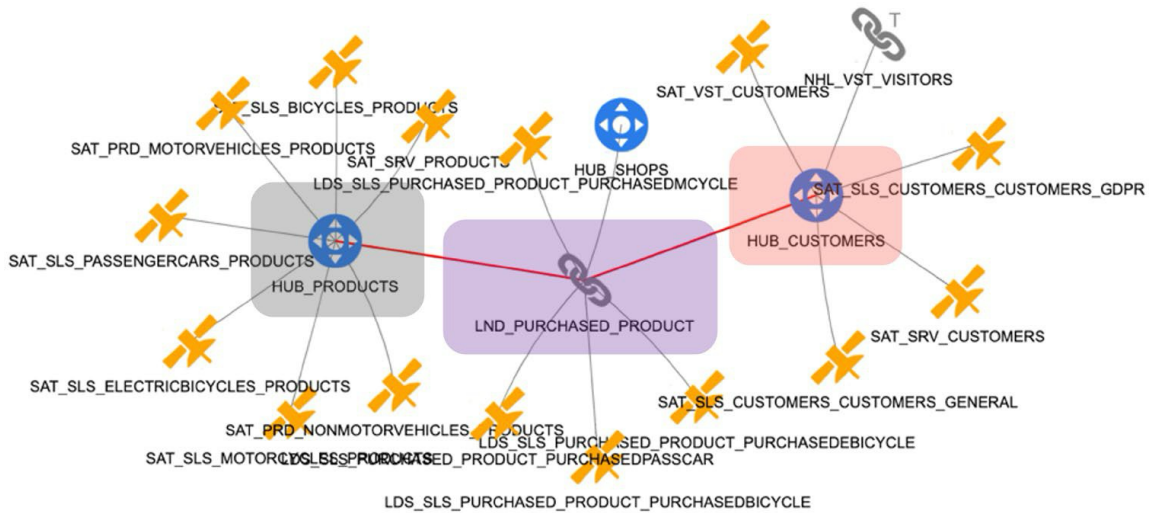
Let's review naming conventions for VaultSpeed Data Vault Models. Objects are initially assigned default names, but these can be changed, either individually or through parameter settings.

To better align with the target model in our example, we've modified the link name to "purchased_products"



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

Now that everything is defined, we can present the Data Vault target model in pictures 42 to 44. Picture 42 illustrates the model for products in src1 and src2 along with the integration through the link between the customer and the purchased product.



Picture 42

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



Picture 43

Picture 44 provides an overview of the entire Data Vault Model across all three source systems. We can analyze the purchase behavior among our residential customers. To simplify end-user querying, VaultSpeed offers an additional automation layer for creating customized business logic and construting the presentation layer on top of the data vault. However this is beyond the scope of this paper.



Picture 44

Does this scale?

It's common for technical documents to use examples that are specifically designed to explain concepts with the utmost clarity. It's important to ensure that these examples are not too complex, as they can cause readers to lose sight of the essence of what's being explained. Therefore, it's valid to question the complexity of the examples used in technical documents.

So, the question is, does the approach scale to real-life data integration challenges?

The answer is yes, it does scale. And here is the proof.

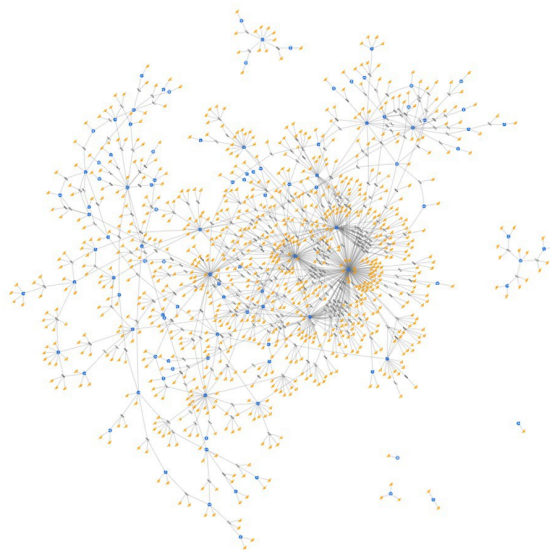
The Data Vault model depicted in picture 45 is the outcome of a complex integration task for a large enterprise client. The example successfully brings together 58 diverse data sources, which encompass 592 distinct source objects in total. To ensure privacy and confidentiality, the model is presented in a fully anonymized form, but a few key observations can still be made.

Upon closer inspection, it becomes clear that the number of hubs and links is significantly lower when compared to the abundance of satellites. This observation indicates that considerable efforts have been made to identify and group together the most prevalent business keys in the hubs. It is worth noting that certain hubs have a staggering number of connected satellites, which speaks to the effectiveness of this approach.

Secondly, only a few hubs and satellites are disconnected from the bulk of the model, providing more proof of its strong integration.

If you filter out the satellites, you can quickly see how different business concepts are interconnected. Business users will recognize their business process in a model laid out like this.

It is crucial to highlight that the model was not developed in a single iteration. The team responsible for its construction employed an agile approach, gradually incorporating additional data sources. By utilizing the Data Vault methodology, they accomplished this task without the need to modify any of their prior work.



Picture 45

Conclusion



In conclusion, we successfully addressed the automation challenge, and integrated data from three sources into a comprehensible Data Vault model. We used the business conceptual model as our guide, and as a result, this business model is reflected in the physical data model of the Data Vault, making it easily understandable for business users.

VaultSpeed's template engine has access to all metadata and can convert repeatable template logic into data definition and data transformation code. This code, including DDL, DML, and workflow code, can be used to install and load the Data Vault model in your preferred data runtime environment.

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 Seattle Leuven Vilnius

Sluisstraat 79 03-01
3000 Leuven
Belgium

© 2023 VaultSpeed – all rights reserved

