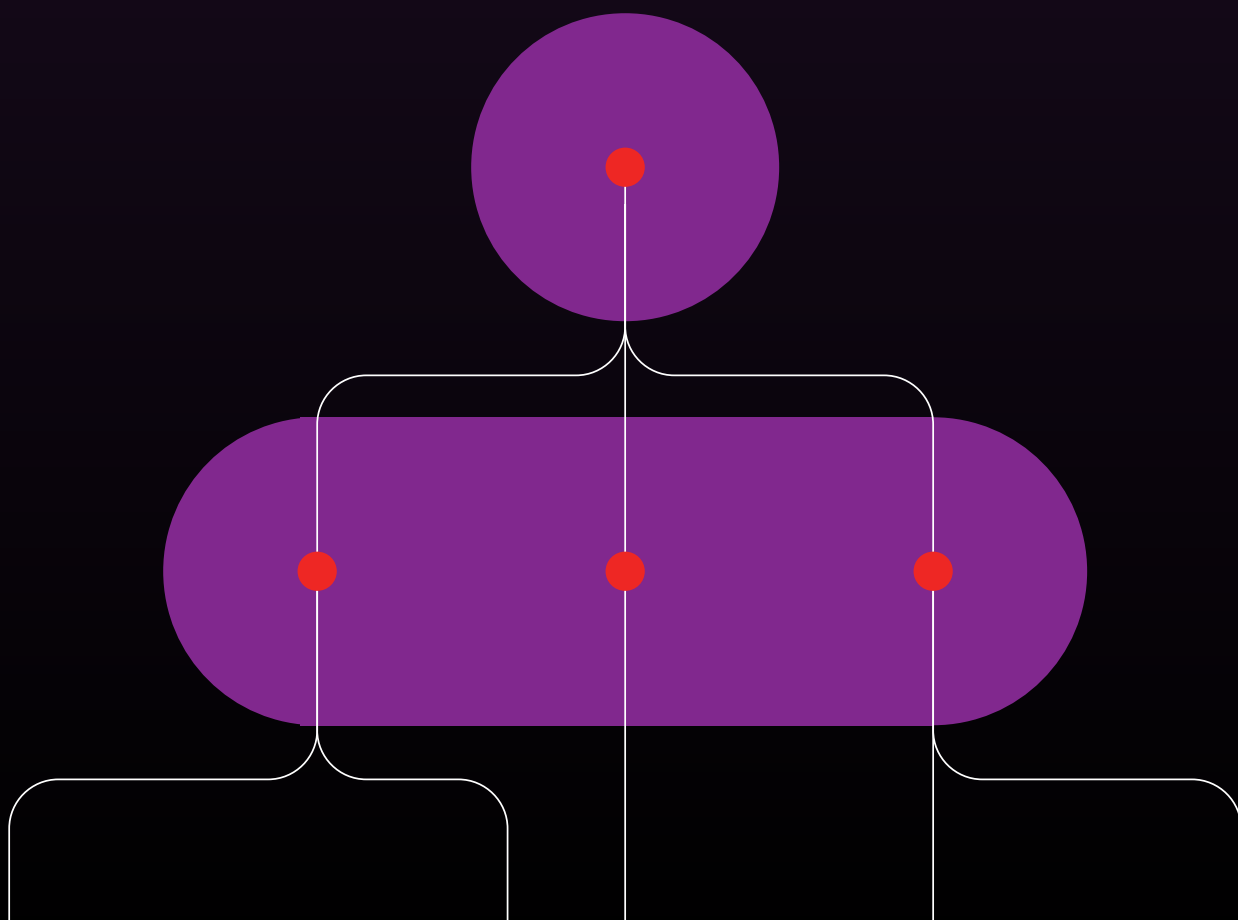




# Accelerate the mapping of your business taxonomy with VaultSpeed



# Table of Contents

Setting the scene.....	3
Data Vault 2.0 and automation.....	4
The Single Version of the Facts.....	5
The Data Vault 2.0 model.....	6
The Single Version of the Facts versus business taxonomy.....	8
What is taxonomy?.....	9
Use case: multi-source & relationship .....	10
The resulting Data Vault 2.0 business model .....	13
Business taxonomy mapping in VaultSpeed .....	15
Data Vault creation .....	20
Conclusions.....	23

# Setting the scene

---

Data Vault 2.0 is an architecture and design methodology suitable for systems required to provide long-term integrated storage of data relating to multiple operational systems.

In Data Vault 2.0 data integration solutions, some axioms need to be upheld:

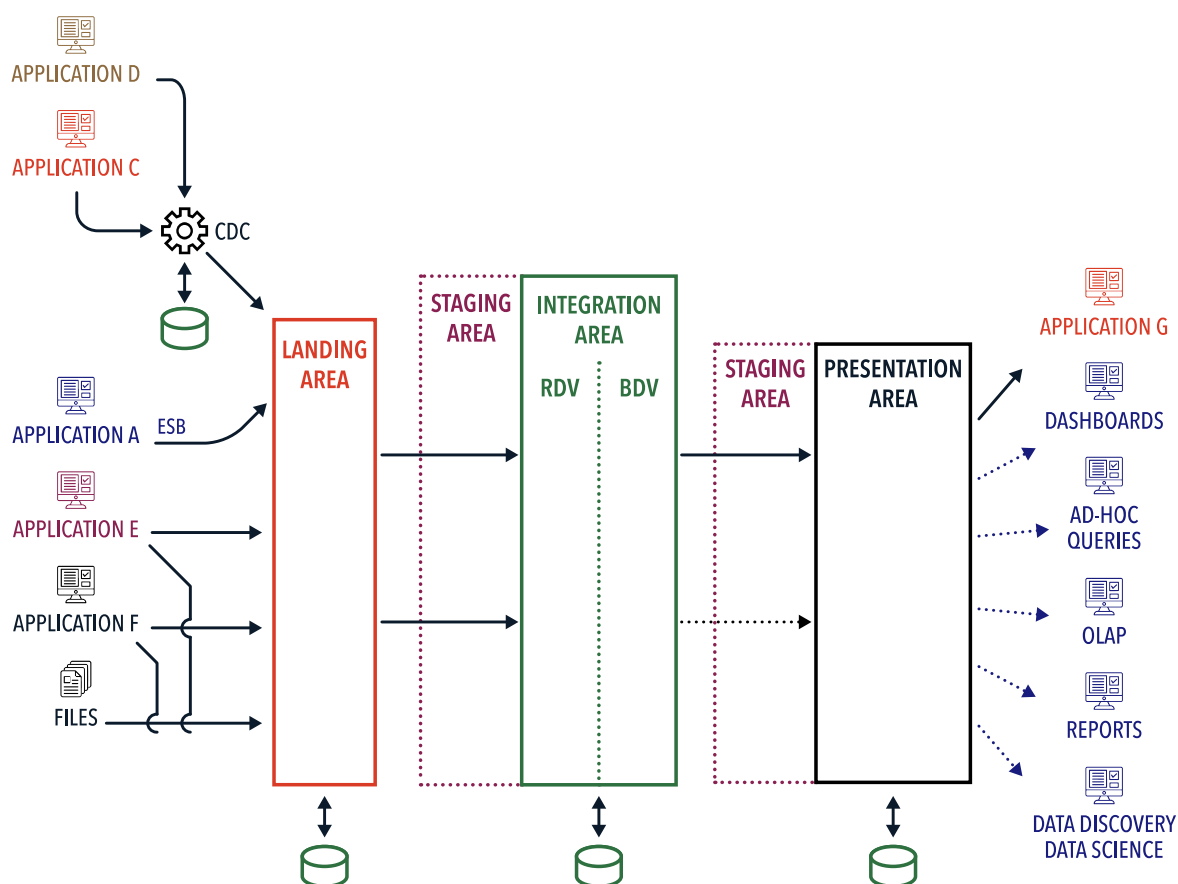
- Always use a data warehouse automation tool.
- Employ consistent business logic in the Data Vault 2.0 model (to facilitate business reporting and understanding).
- Do not build your own enterprise data model.
- Do not build a Raw Data Vault layer that is source-based (that tries to follow/copy the source model).

As you might have noticed, we're dealing here with axioms that seem contradictory, like oil and water. Automation requires a repeatable loading pattern from source to target.

Data Vault modeling involves converting the source model into a Data Vault model. But the Raw Data Vault that loads the raw, unfiltered data from the source into Hubs, Links, and Satellites, based on business keys, should not just copy the source model.

This paper explains why these contradictory axioms do not defy logic and why the VaultSpeed automation tool is designed to transfer any business taxonomy you might think of into the raw Data Vault layer.

# Data Vault 2.0 and automation



Picture 1

The Data Vault 2.0 architecture, as shown in Picture 1, is designed to be flexible and extensible.

The integration area - the Data Vault - consists of two layers, namely:

- A Raw Data Vault - a single version of the facts
- A Business Data Vault - single or multiple versions of the truth

The problem with truths is that they tend to vary over time and that most companies are not aligned and therefore struggle with competing versions of the truth. Facts on the other hand, bring a layer of stability.

# The Single Version of the Facts



## multiple versions of the truth

The Raw Data Vault (RDV) is designed to contain what is known in Data Vault 2.0 terminology as the 'Single Version of the Facts.' Facts are nothing more than raw, historical, unfiltered data from the sources.

These raw data are factual. They prove that something exists or has occurred. The data are unfiltered.

The Business Data Vault (BDV) aligns business keys/terms from the source system with these business views to ensure compliance.

Different viewpoints coexist and are all regarded by Data Vault 2.0 to be valid versions of the truth.

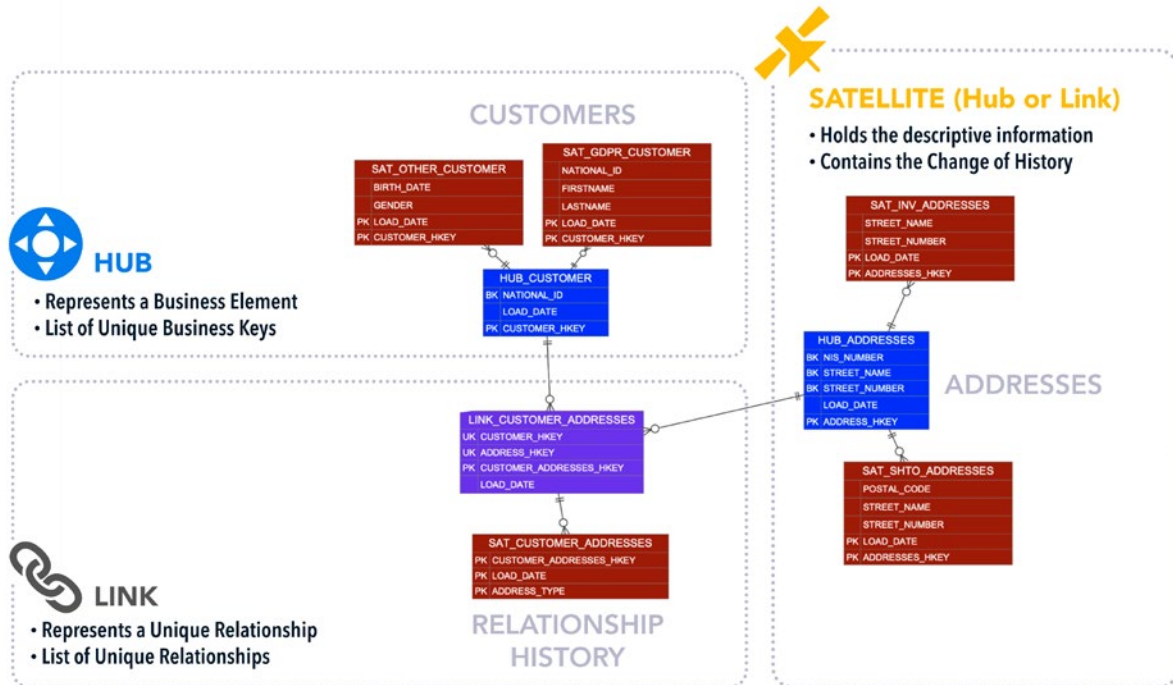
The RDV layer stores the original, unmodified data and complements the BDV in making sure that multiple versions of the truth can be delivered in a fast and flexible way.

## automation

Automation – generation of the loading logic - is only feasible when:

- A relationship can be identified between the incoming data and the target Data Model.
  - This is why the Single Version of the Facts is vital.
- A sufficient amount of objects can be loaded using the same loading pattern ( $\geq 10$  objects).
  - Otherwise, the effort to build the automation for the pattern will be more costly than building the pattern for every single object.

# The Data Vault 2.0 model



Picture 2

## key to supporting multiple versions of the truth

The Data Vault model is resilient to change. It can deal with multiple versions of the truth over time and support the model changes that emerge in agile organizations, as they react quickly to inevitable changes in requirements.

The advantage of using Data Vault is that it eliminates re-engineering ELT logic for all the dependent layers.

## key to automation

The Data Vault 2.0 model, as shown in picture 2 splits entities into their most minor components (Hubs = keys, Links = relationships, Satellites = descriptions).

Each component has a single function which results in a consistent loading logic for each particular component type.

The primary requirement to support automation is to construct the logic at the attribute level. You need to define the business key (a decipherable key to identify one record), not the physical key (ex. passport number).

This provides a repeatable loading pattern that is identical across all objects with the same object/component type (Hub or Satellite or Link).

As a result, the more objects you transfer to the Raw Data Vault, the faster the automation.

Automation – generation of the loading logic - is only feasible when:

- A relationship can be identified between the incoming data and the result to be achieved.
- A sufficient amount of objects can be loaded using the same loading pattern ( $\geq 10$  objects).
  - That is why the Data Vault 2.0 model is critical, as the same repeatable pattern can load every object type (Hub, Satellite, Link).

# The Single Version of the Facts versus business taxonomy



Modeling business taxonomy in the Raw Data Vault layer - The Single Version of the Facts – can appear daunting.

Data Vault provides some relatively straightforward ways of achieving it.

The physical data model represents a certain level in the taxonomy, and it is up to the business to decide what level that is.

The 'Hubs' represent the business elements, and the 'Links' the relationships between these business elements. Both reflect the business taxonomy as closely as possible.

The detailed mapping on object-level should best be stored in a business metadata layer which in itself contains a mapping to the technical data model layer of the Data Vault.

Let's see what this means in practice. We'll walk you through the example of how to convert two related business taxonomies to the Data Vault 2.0 model.



# What is taxonomy?



Taxonomy represents the formal structure of classes or types of objects within a domain. It organizes knowledge by using a controlled vocabulary to make it easier to find related information.

## A taxonomy:

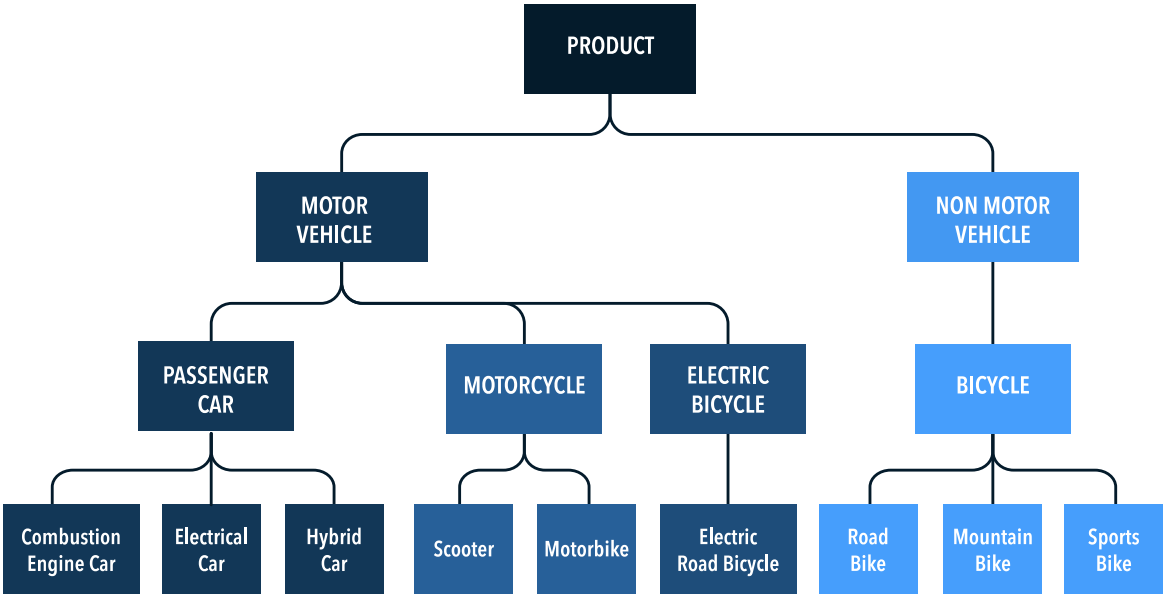
- Must be a hierarchical categorization and provide names for each object in relation to other objects.
- Has the ability to capture the membership properties of each object in relation to other objects.
- It follows specific rules to classify or categorize any object in a domain. These rules are exhaustive, consistent, and unambiguous.
- Applies rigor in the specification, ensuring that any newly discovered object fits into only one category or object.
- Inherits all the properties of the higher level but has the ability to add additional properties.

## Organizations use taxonomies to:

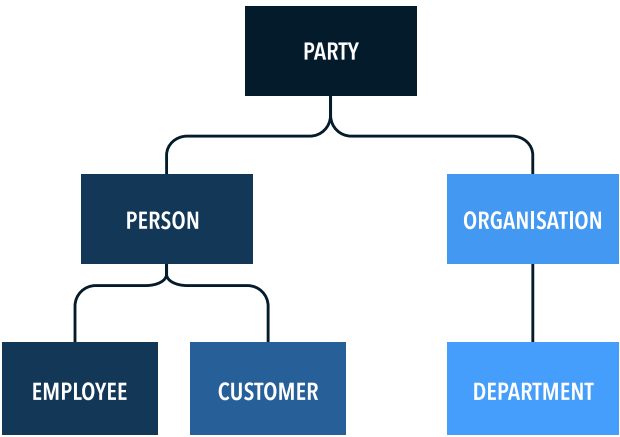
- Achieve better data quality.
- Organize metadata in an easy-to-grasp format (e.g., a website map).
- Manage data assets through data governance.
- Guide machine learning and data experiences towards identifying trends and Patterns.

# Use case: multi-source & relationship

Let's take the hypothetical case of a company selling a range of transport products as shown in picture 3.



Picture 3



Picture 4

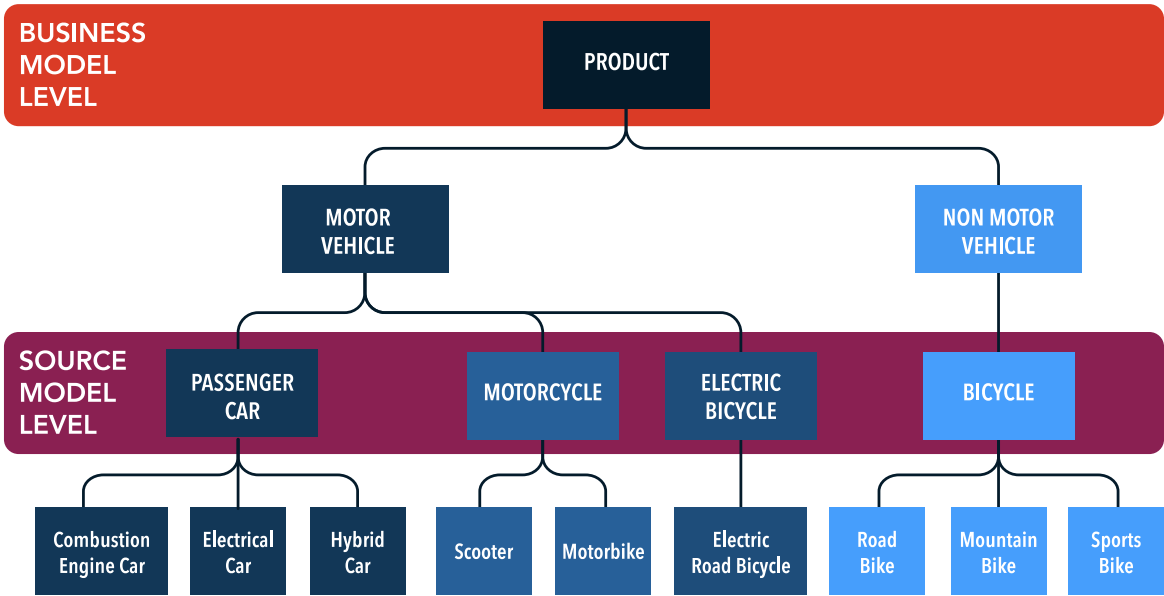
This company sells these products to customers. The relationship between products and customers is referred to as the purchased relationship.

This use case considers that there are multiple sources and that a relationship exists between business elements.

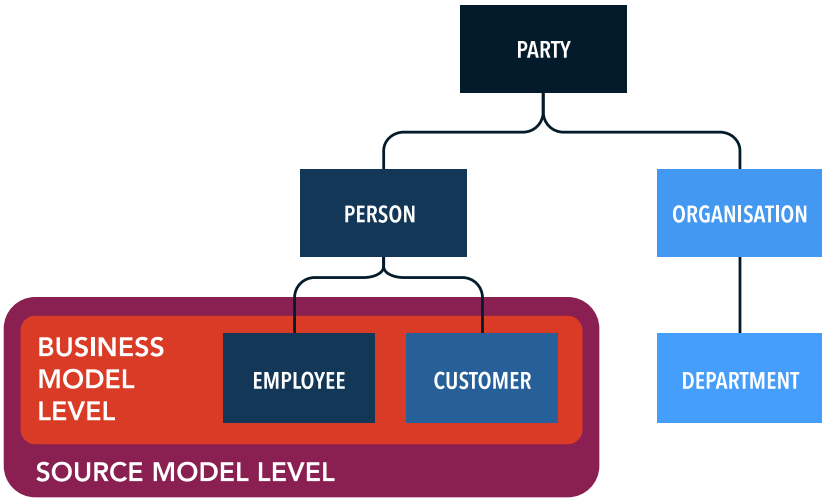
It is a typical master data case. Master data is frequently held by multiple sources, appears at different levels within these sources, and can be updated and replicated from one source (single-master system) or multiple sources (multi-master).

This scenario deals with two sources with different levels of data. But the organization requires them to be implemented at the same taxonomy level in the Business Data Model.

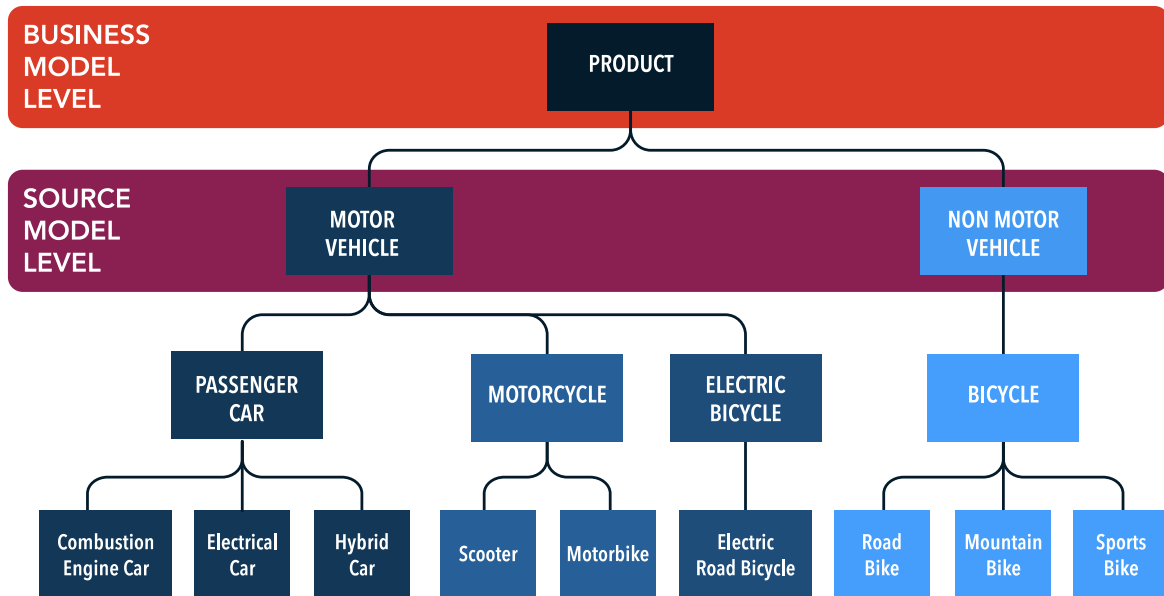
The Data Source src1 contains a PARTY taxonomy at the third level. Only the person part of the taxonomy is implemented in the source as this is a B2C company that only sells to individual persons.



Picture 5: The Data Source src1 is implemented at Level 3 in the taxonomy as indicated in purple. However, the business model is at Level 1 of the product taxonomy, as shown in red.

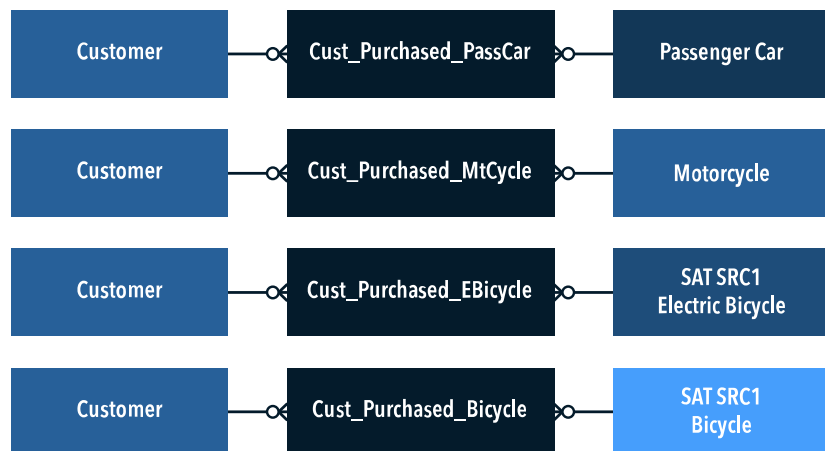


Picture 6: Both the Source Model and Business model are at Level 3.



Picture 7

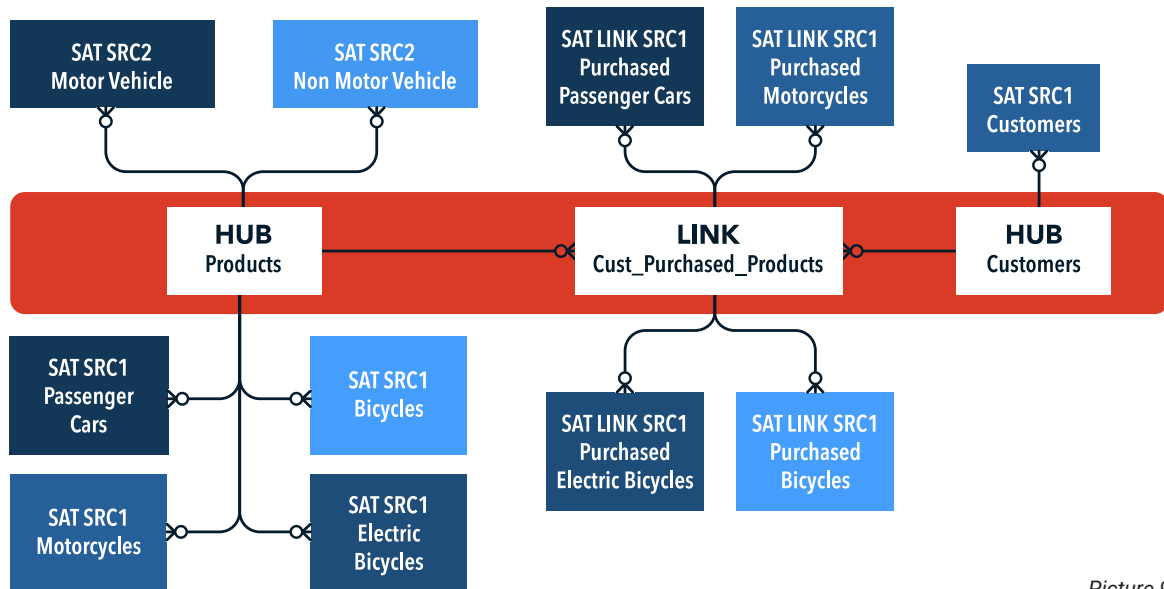
The Data Source src2, is implemented at Level 2 and Level 3 in the taxonomy, as indicated in purple in picture 7. When the need arises to produce a report on who bought what product, we'll define the purchased relationship between the customer and the product.



Picture 8

In the source there are four many-to-many relationships that appear at level 3 in the taxonomy for PARTY and PRODUCT.

# The resulting Data Vault 2.0 business model



Picture 9

## Definition 1

Hubs contain unique keys and represent the Business Elements.

- The business elements to be at level 1 in the Product taxonomy are defined as the HUB Products.
- The business elements to be at level 3 in the Party taxonomy are defined as the HUB Customers.

## Definition 2

Links contain the unique relationship between Business Elements.

- The business model defines a purchased relationship between Products and Customers.
- This results in the LINK Cust\_Purchased\_Products

As shown in picture 9 and indicated in red, the very definitions inside the Data Vault 2.0 standard ensure that the business model is represented inside the resulting model in the Raw Data Vault utilizing Hubs and Links.

Adding a PARTY element to Data Source src2 with the purchased relationship does not impact the Hub and Link canvas representing the business model.

The subtypes of the business elements in the product taxonomy at level 2 for src2 and at level 3 for src1 become Satellites of the Products Hubs.

The subtypes of the business elements in the Party Taxonomy at level 2 for src1 become Satellites of the product Hubs.

Even if the Raw Data Vault is built on the Single Version of the Facts, you still have a business taxonomy view implemented in the data model at the level of your Hubs and Links.

The Satellites contain descriptive information that changes over time. The mapping towards attributes in your business model needs to be described in the metadata to keep track. Typically, these are designed and managed in an enterprise metadata tool.

# Business taxonomy mapping in VaultSpeed

VaultSpeed can be used to create a model that reflects your business taxonomy.

## SRC1

To implement the use case outlined above several steps need to be followed.

### Step 1: Harvest the metadata for the relevant Data Sources.

In this example:

Data Source src1 relating to the taxonomy PRODUCTS - as shown in Picture 10

Data Source src2 relating to the taxonomy PARTY - as shown in Picture 11

Data Source for the many-to-many purchased relationships as shown in Picture 8

Table Name ↓	Selected Flag	Exists in Source Flag
bicycles	Yes	Yes
cust_purchased_bicycle	Yes	Yes
cust_purchased_ebicycle	Yes	Yes
cust_purchased_mcycle	Yes	Yes
cust_purchased_passcar	Yes	Yes
customers	Yes	Yes
electric_bicycles	Yes	Yes
motorcycles	Yes	Yes
passenger_cars	Yes	Yes

Picture 10. Shows the selection of the source objects for Data Source src1. The harvested metadata contain the objects of the source model level of the taxonomy for PRODUCT.

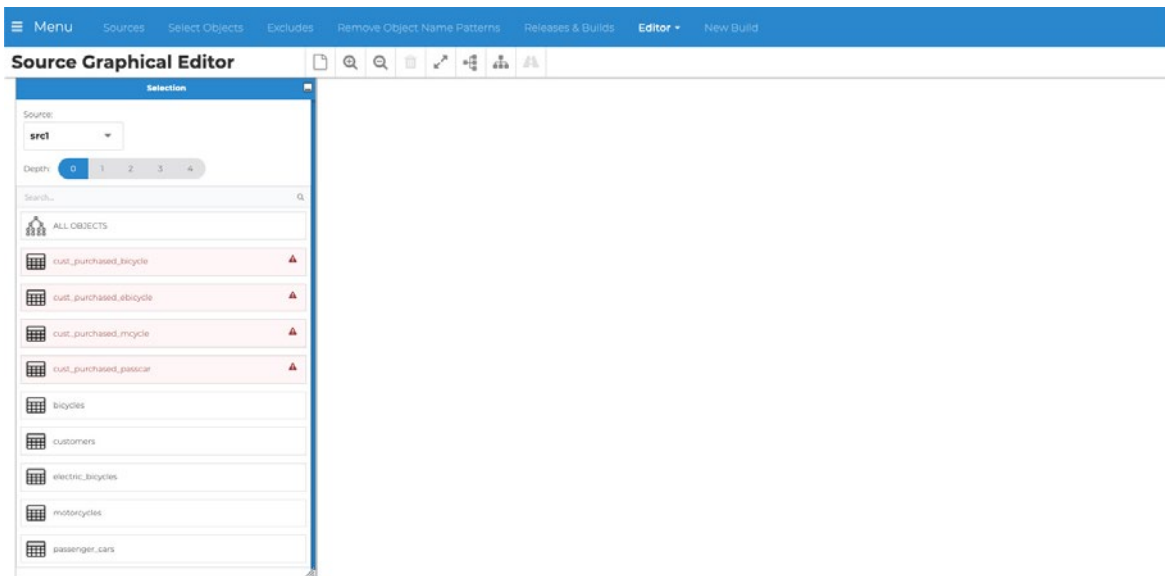
Table Name ↓	Selected Flag	Exists in Source Flag
motor_vehicles	Yes	Yes
non_motor_vehicles	Yes	Yes

Picture 11 shows the selection of the source objects for Data Source src1. The harvested metadata contains the objects of the source model level of the taxonomy for PARTY

## Step 2: Define the mapping of your source model towards a Data Vault model

### SRC1 graphical modeling:

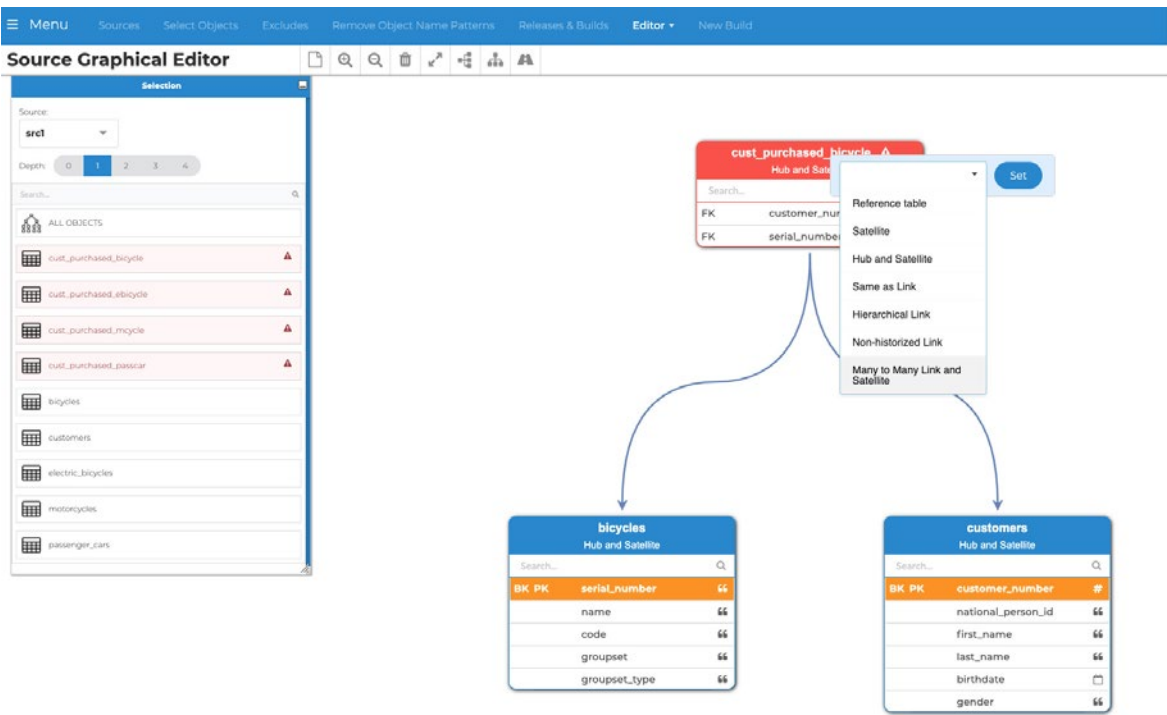
Picture 12 shows the selection pane with all the objects of src1 Data Source src1. The exception handling in the tool is apparent. Four many-to-many objects are highlighted in RED and require attention before code can be generated.



Picture 12



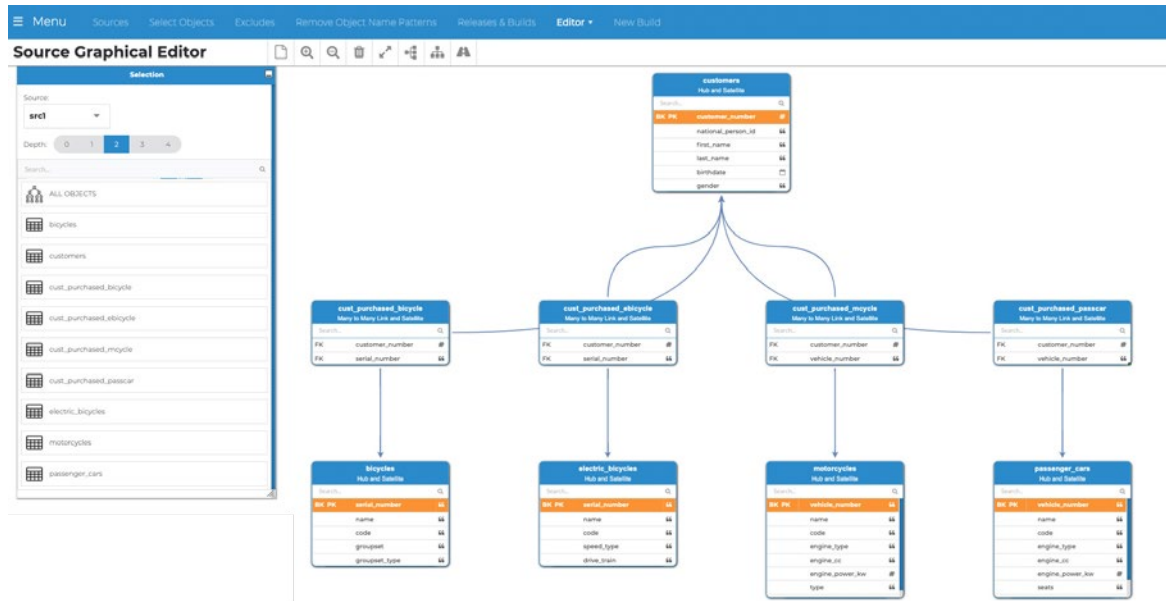
When you drag an object into the canvas, you'll get it and its related objects, depending on the depth level you set in the selection menu.



Picture 13

Picture 13 shows the cost\_purchased\_bicycle relationship with the related objects bicycle and customer after dragging it onto the pane. The error with the cost\_purchased\_bicycle is that it does not have a primary key.

As this is a many-to-many relationship, we need to define it by right-clicking on the object and selecting the correct object type = many-to-many relationship with Satellite.



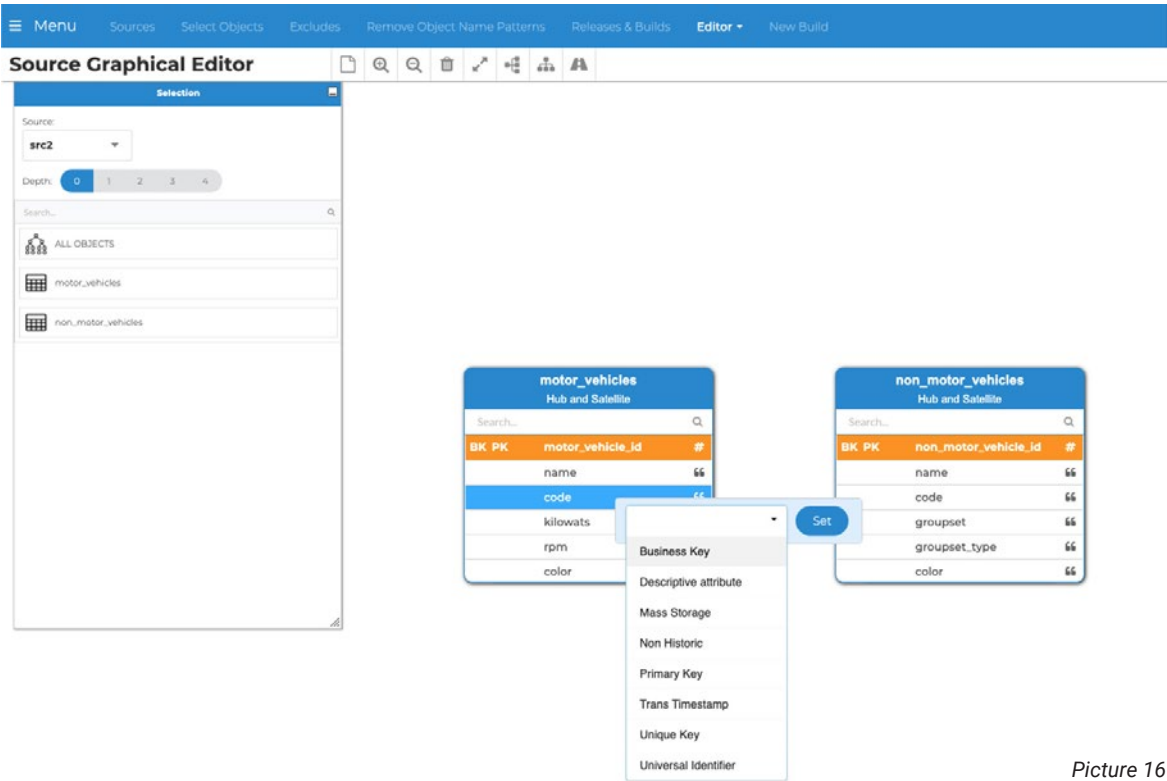
Picture 14

After defining all the source many-to-many purchased relationships objects as many-to-many links, you get the result shown in picture 14.

We now define that these objects are all multi-master objects, which is again standard behavior.

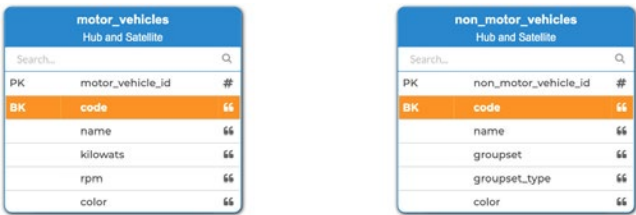
# SRC2 graphical modeling:

The second source contains Business Keys that differ from the Primary Keys. We need to set the Business Keys, which is the code attribute for both objects.



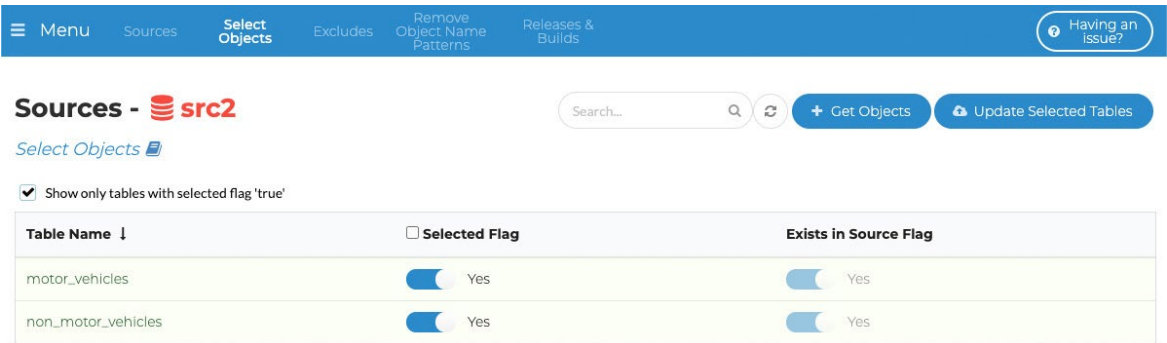
Picture 16

Setting the Business Keys for both the objects leads to the result shown in picture 17



Picture 17

We define these objects as Multi-Master Objects which is again standard. Nothing else needs to be done. See picture 18.

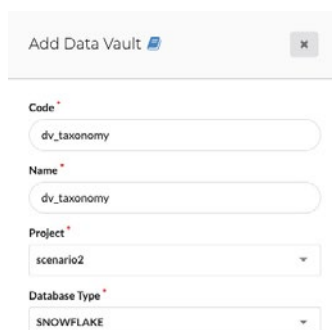


Picture 18

# Data Vault creation

The next step is to create a Data Vault.

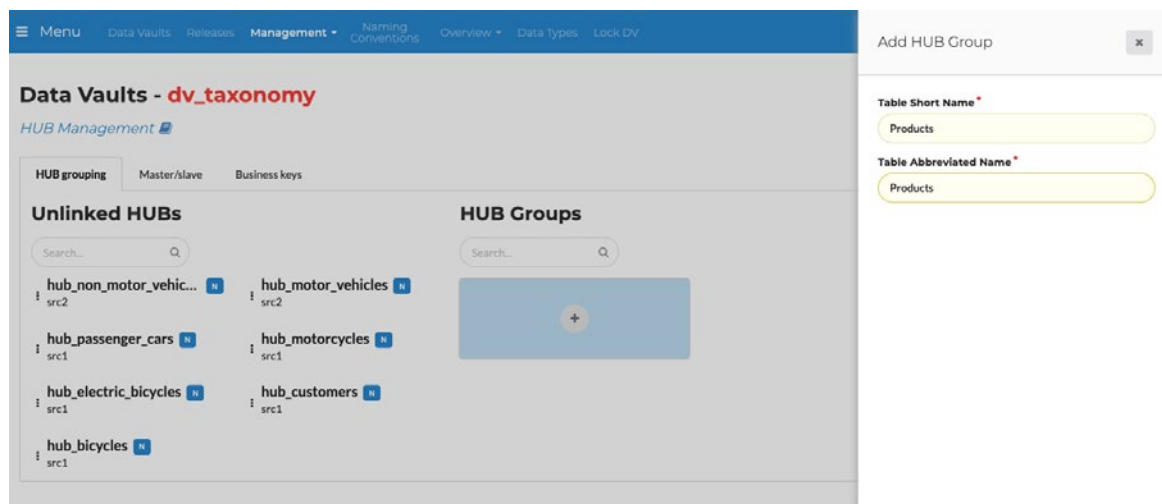
Select the double source project (scenario 2) that links the sources to the Data Vault Layer.

A screenshot of the 'Add Data Vault' form. It has a title bar 'Add Data Vault' with a close button. The form contains four fields: 'Code' with value 'dv\_taxonomy', 'Name' with value 'dv\_taxonomy', 'Project' with a dropdown menu showing 'scenario2', and 'Database Type' with a dropdown menu showing 'SNOWFLAKE'.

Picture 19

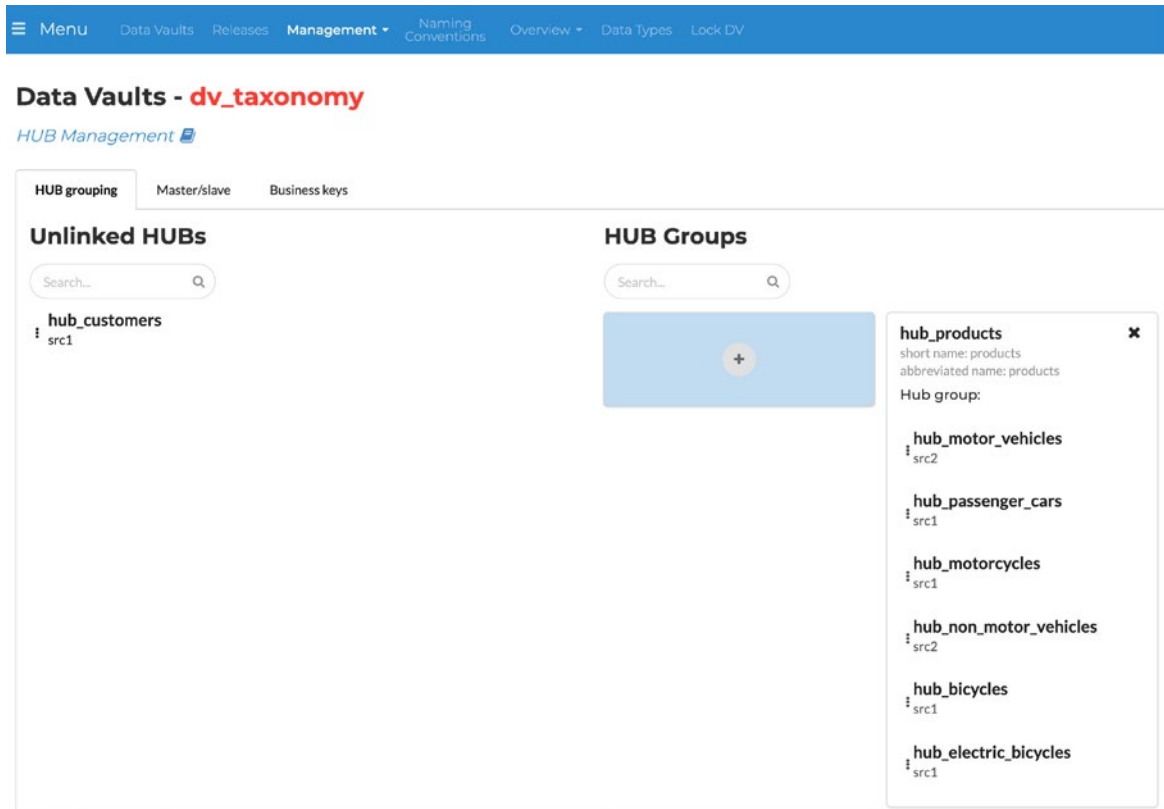
Push a Data Vault Release, and the Hub management screen will enable:

- Creation of the Product Taxonomy level by defining a Hub group with the name "Products", see picture 20.
- Link all the Hubs from the Different Source to this Hub Group

A screenshot of the 'Data Vaults - dv\_taxonomy' Hub Management screen. The top navigation bar includes 'Menu', 'Data Vaults', 'Releases', 'Management', 'Naming Conventions', 'Overview', 'Data Types', and 'Lock DV'. The main header is 'Data Vaults - dv\_taxonomy' with a sub-header 'HUB Management'. There are three tabs: 'HUB grouping', 'Master/slave', and 'Business keys'. The 'HUB grouping' tab is active, showing two columns: 'Unlinked HUBs' and 'HUB Groups'. The 'Unlinked HUBs' column lists several hubs with their source IDs (src1, src2) and a status icon. The 'HUB Groups' column has a search bar and a large blue button with a plus sign. On the right side, there is a 'Add HUB Group' form with fields for 'Table Short Name' (value: 'Products') and 'Table Abbreviated Name' (value: 'Products').

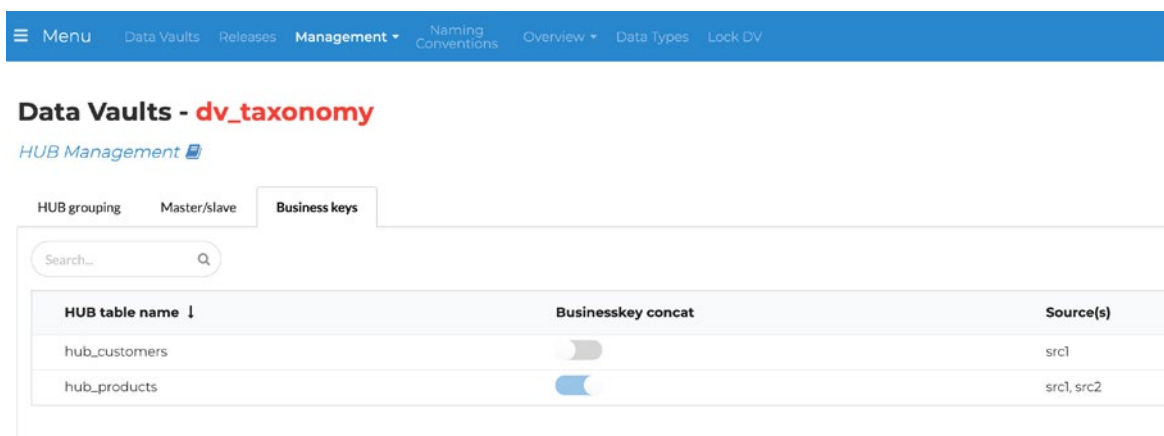
Picture 20

- This will result in a one-source multi-object, multi-source Hub group, see picture 21.



Picture 21

- Define the relationship between the sources, which in our case is a master-master relationship. This is standard, so nothing needs to be adapted in this tab.
- Define the Business Key for the Product Business Element to become a concatenated key to reduce the Hub business key to a single attribute where the content is concatenated instead of all the different key names in the different multi-master product objects, see picture 22.



Picture 22

The name of the link still needs to be adapted because the name will be based on the first source-level name object, defined as a many-to-many link.

Picture 23 shows the result of the renaming to the Business level Taxonomy naming.

Menu

Data Vaults

Releases

Management

**Naming Conventions**

Overview

Data Types

Lock DV

**Data Vaults - dv\_taxonomy**

Naming Conventions

HUB

**Object Based Links**

Relation Based Links

Non-Historical Links

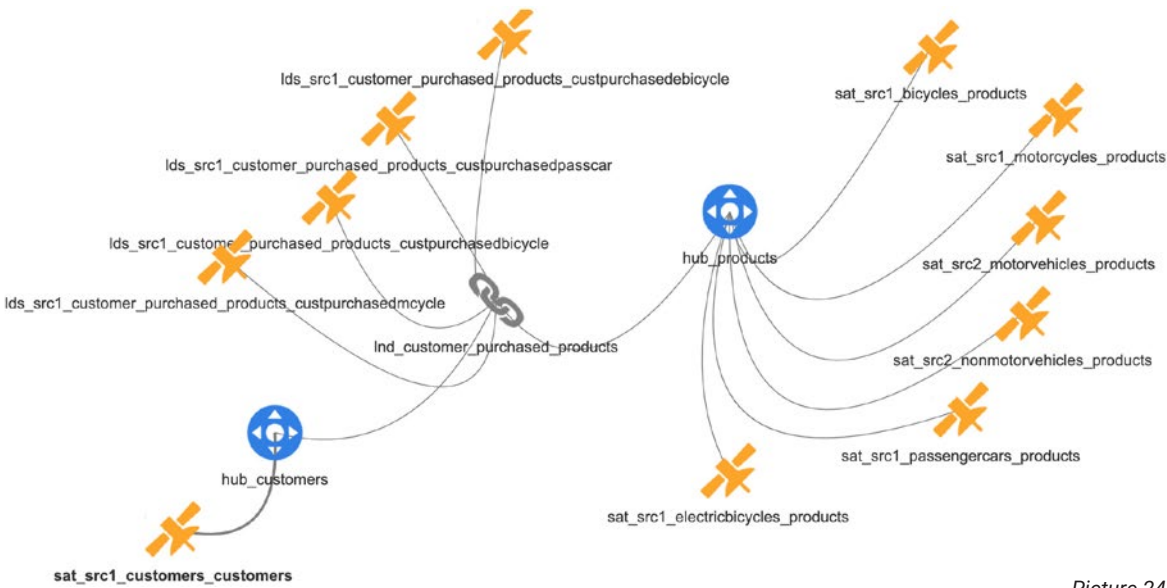
Search...

Q

Table name	Table short name	Table abbreviated name	Source(s)
lnd_customer_purchased_products	cust_purch_prod	customer_purchased_products	src1

Picture 23

Everything is now defined and the model can be shown, see picture 24.



Picture 24

# Conclusions



- 01 VaultSpeed adheres to the Data Vault 2.0 standard and therefore supports mapping any business taxonomy to the physical Data Vault Model of the Raw Data Vault.
- 02 We can handle sources with data models that are at a different or at the same level in the Taxonomy as the Business Model. We deal with Single Source objects or Multi-Source objects in a Single or Multi-Master setup.
- 03 We've shown that business taxonomy and the Single version of Facts can genuinely blend.
- 04 The VaultSpeed interface allows you to accelerate this integration and apply business logic with minimal effort.

---

**Visit our site**

[vaultspeed.com](https://vaultspeed.com)

**Contact sales**

[sales@vaultspeed.com](mailto:sales@vaultspeed.com)

**Book a demo**

[vaultspeed.com/book-a-demo](https://vaultspeed.com/book-a-demo)

**Join our community**

[community.vaultspeed.com](https://community.vaultspeed.com)



Sluisstraat 79 03-01  
3000 Leuven  
Belgium