Data Vault Modeling Patterns

Snowflake Data Vault User Group
2022-05-11

Short Introduction

Christian Kaul

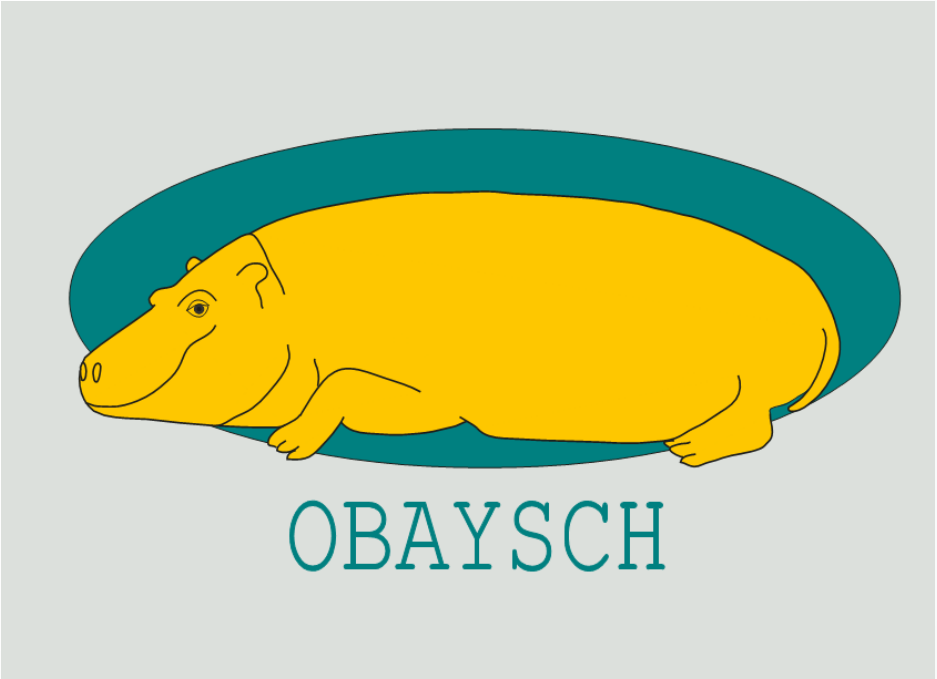
Data Modeling Aficionado & Data Structure Designer at Obaysch

München, Bayern, Deutschland

Ask me about the 2022 Knowledge Gap conference!

Email: christian.kaul@obaysch.net

https://www.linkedin.com/in/christian-kaul/



Why Data Vault Modeling Patterns?

- When building a data vault model, you encounter some of the same situations again and again.
- Therefore it is good to have some modeling conventions so that you deal with these situations in a consistent manner.
- The data vault modeling patterns shown in this presentation will help you to achieve this consistency by recognizing standard situations and sticking to standard solutions for resolving them.

Data Vault Modeling Patterns

- Keyed Instance
- Hierarchy
- Identity
- Header and Line Items
- Lookups

Data Vault Modeling Patterns

Keyed Instance

Keyed Instance

When representing connections between instances of concepts as a **link**, in many cases you have to include a so-called **keyed-instance hub**:

1. The connection is **transferable.**
2. You have to store some **details** about the connection.
3. There can be multiple **distinguishable connections** involving the same instances of concepts.
4. You want to be able to **connect** other concepts **to a specific connection.**

Keyed Instance: Case 1

Case:

The connection is **transferable**.

Example:

An Employee can work for different departments over time and you have to record the currently active Department Assigment.

Keyed Instance: Case 2

Case:

You have to store some **details** about the connection.

Example:

It’s necessary to store quantity and actual price (as opposed to list price) for a Product on a Sale.

Keyed Instance: Case 3

Case:

There can be multiple **distinguishable connections** involving the same instances of concepts.

Example:

A Visitor can go to the same Website multiple times, each time being a different Visit with different detail values.

Keyed Instance: Case 4

Case:

You want to be able to **connect** other concepts **to a specific connection** (without repeating certain combinations of concepts over and over again or building any link-on-link constructs).

Example:

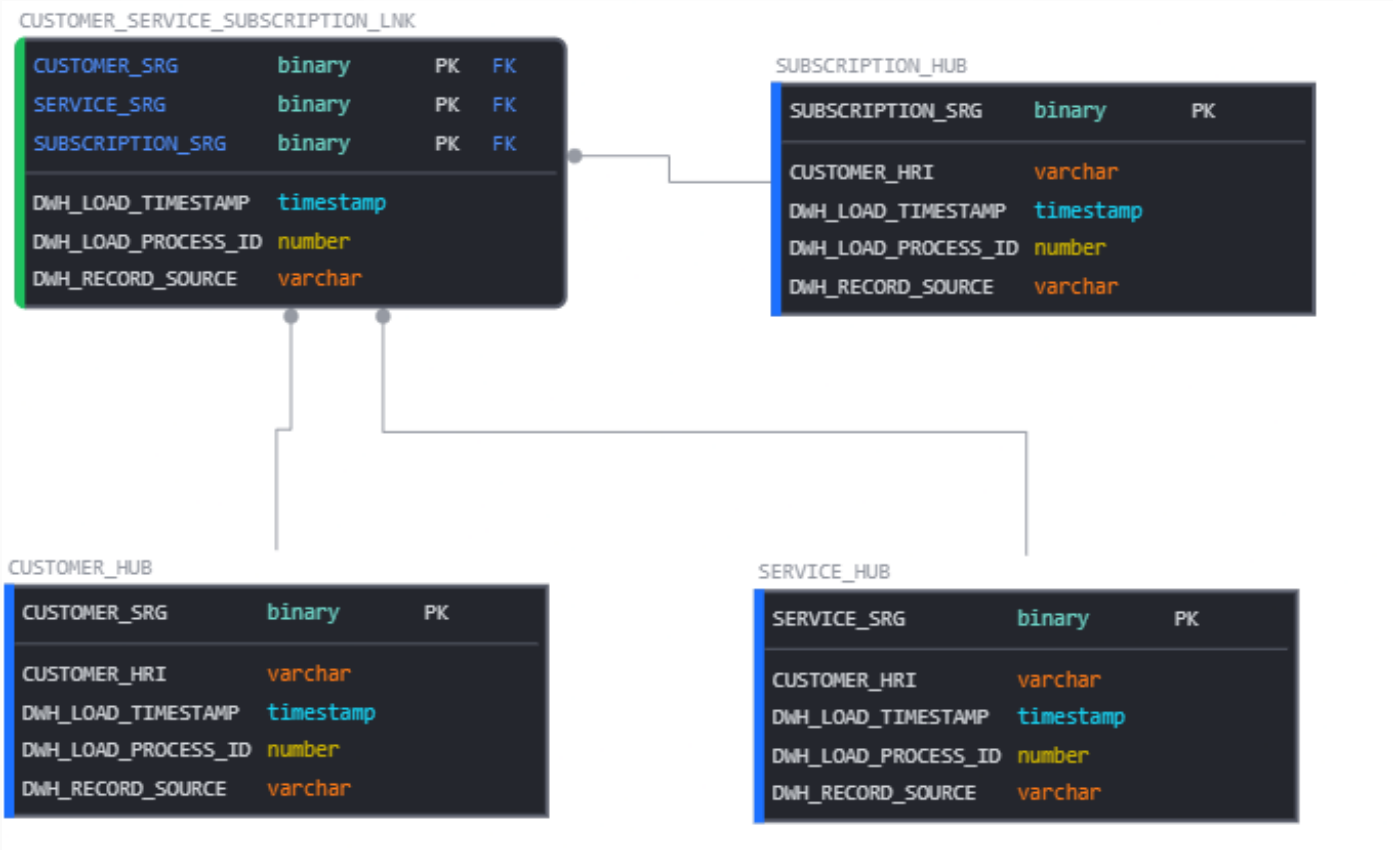
A Delivery can include some Products on a Sale but not others that are currently out of stock and will be part of a later Delivery.

Keyed Instance: Alternatives

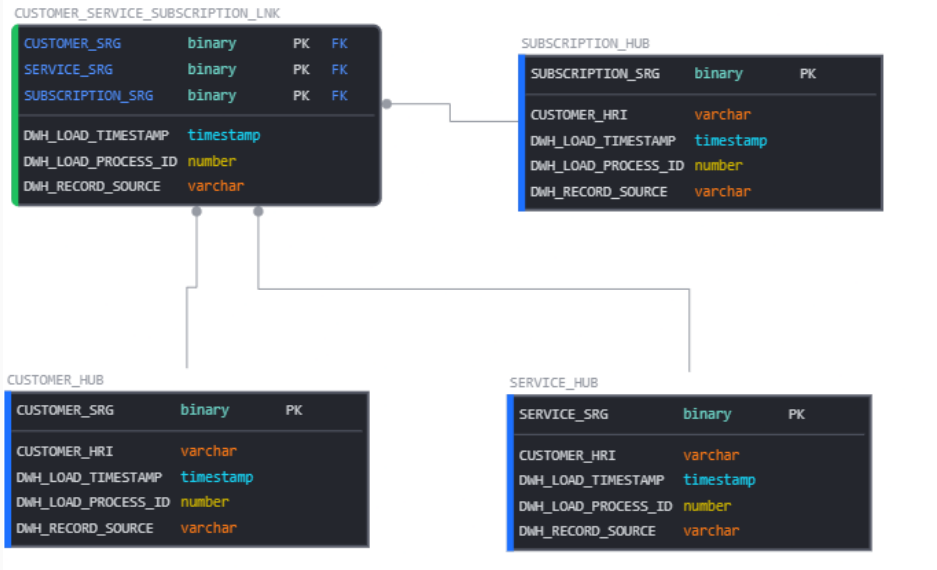
Can’t I just use link satellites instead?

1. Connection is transferable: Usually, yes.
2. Connection has details: Usually, yes.
3. Multiple distinguishable connections: No.
4. Connect to specific connection: No.

Keyed Instance: Example



Keyed Instance: Example & Cases



1. A Customer can subscribe to different Services over time.
2. There are relevant details about a Customer–Service combination (how the Customer subscribed to the Service, why the Customer subscribed to the Service , …).
3. A Customer can stop using a Service and then start subscribing to the same Service again.
4. There are possible connections to a Customer-Service combination (you send an Invoice to the Customer for using the Service for a period of time, the Customer can file a Complaint about some problems with the Service, …).

Data Vault Modeling Patterns

Hierarchy

Hierarchy

Hierarchical connections are another common use case:

- An Employee reports to another Employee.
- A Car Part is made of multiple other Car Parts.
- An Organizational Unit consists of multiple smaller Organizational Units.

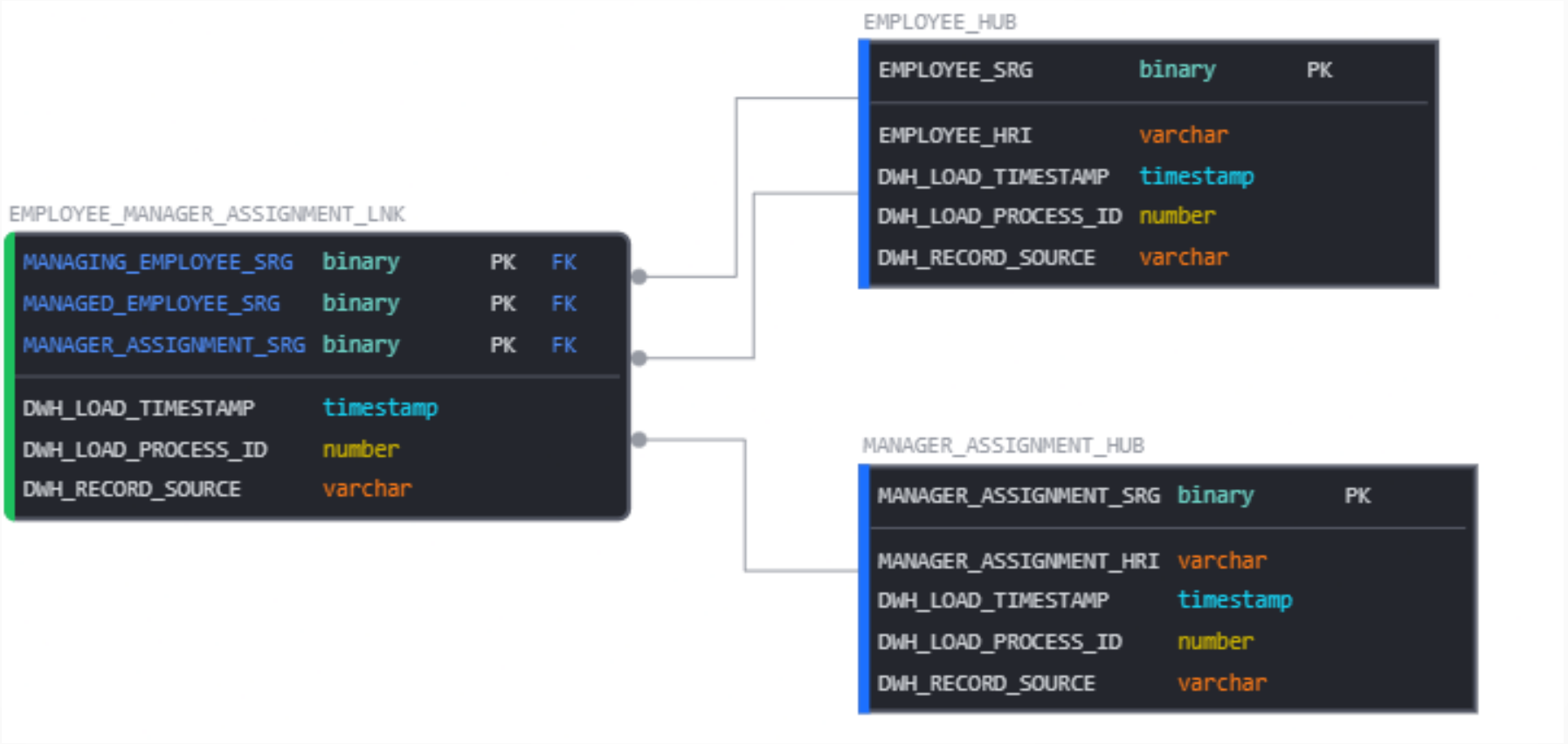
There are two ways of dealing with this use case.

Hierarchy: Same Concept

If the two instances involved in the hierarchical connection are indeed **instances of the same concept**, you can create a **hierarchical link** that includes the same hub twice (plus maybe an additional keyed-instance hub).

The Employee-reports-to-Employee connection most likely would be implemented as a hierarchical link. To avoid confusion later on, it is important to clearly mark the role played by each of the two identifiers referring to the same hub (by calling them MANAGING_EMPLOYEE_SRG and MANAGED_EMPLOYEE_SRG, for example).

Hierarchy: Example



Hierarchy: Different Concepts

If the two instances involved in the hierarchical connection represent **different concepts**, you can create a **regular link** between two different hubs (plus maybe an additional keyed-instance hub).

The Organizational-Unit-consists-of-multiple-Organizational-Units connection would be implemented as a regular link if in your organization, the lower-level Organizational Units are consistently called Teams and the higher-level Organizational Units are consistently called Departments.

Data Vault Modeling Patterns

Identity

Identity

In some cases, you find out that what seemed like two different instances at first actually are one and the same:

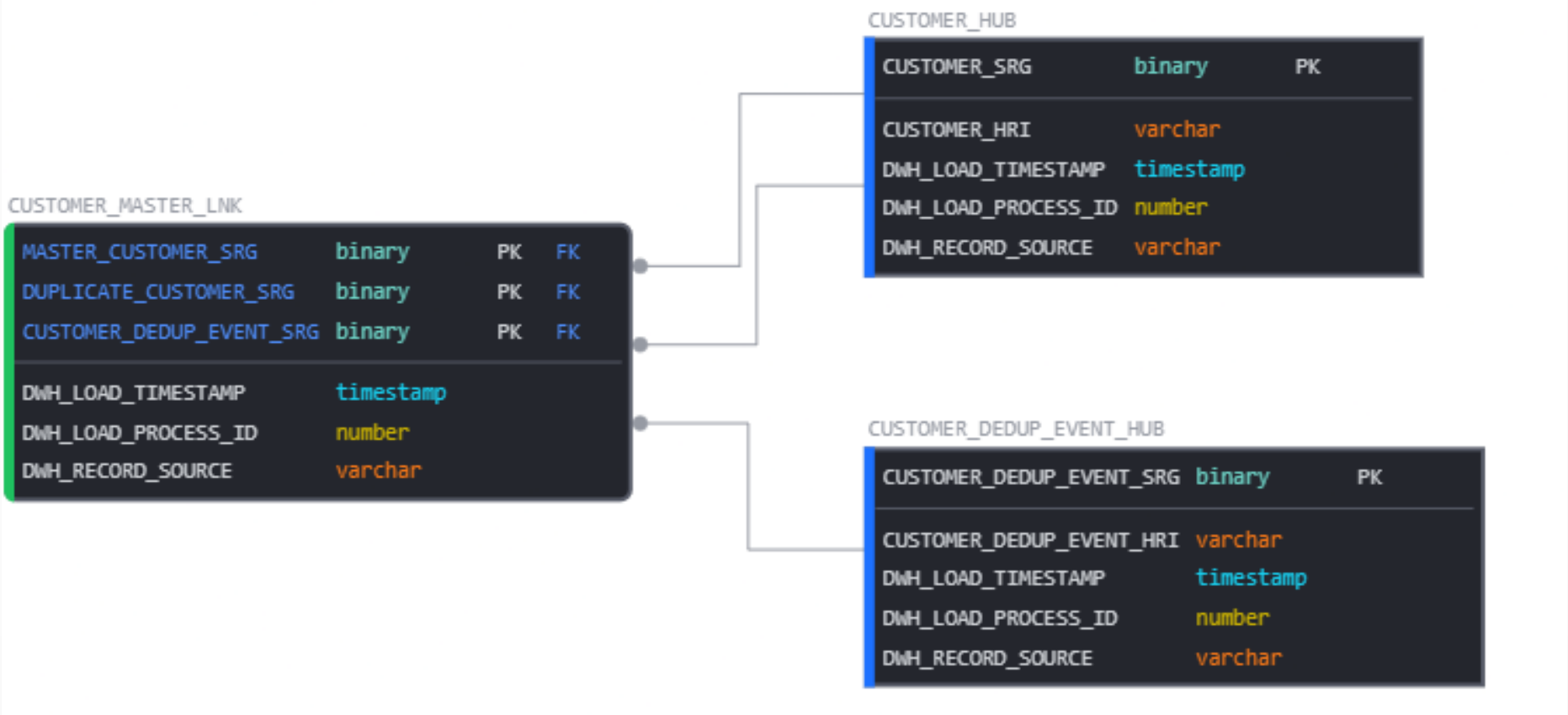
- Two customer codes from different source systems actually refer to the same Customer.
- A new customer code actually refers to an existing Customer that has lost access to our system for some reason.
- When migrating from one source system to another, new customer codes had to be created for existing Customers for technical reasons.

Identity: Same Concept

Many of these cases can be handled using a **same-as link** that includes the same hub twice (plus maybe a keyed-instance hub). One of the hub identifiers in the link refers to the master record and the other one the duplicate record.

Once again, the role played by each of the two identifiers referring to the same hub should be marked clearly (by calling them MASTER_CUSTOMER_SRG and DUPLICATE_CUSTOMER_SRG, for example).

Identity: Example



Identity: Different Concept

Sometimes, identity has to be documented using a **regular link**.

Source system A might treat both Customers and Suppliers as Partners while source system B clearly distinguishes between the two.

When it's not always completely obvious which Partners from source system A are actually Customers, it is better to load all of them to a Partner hub and then create a link between this Partner hub and the Customer hub for those Partners that you have found to be Customers.

Data Vault Modeling Patterns

Header and Line Items

Header and Line Items

Headers and line items are a very common use case that should be dealt with consistently.

- There can be one type of line items (dogbone pattern).
- There can be two types on line items (wishbone pattern).
- And there can be even more types of line items!

Headers and Line Items: Dogbone

The classic header–one type of line items situation appears when there is an event, often but not always a transaction, that involves multiple instances of some other concept:

- a Sale event involving multiple Products,
- a Delivery event involving multiple Products,
- a DWH Load Process event involving multiple DWH Tables or
- a Surgery event involving multiple Surgical Steps.

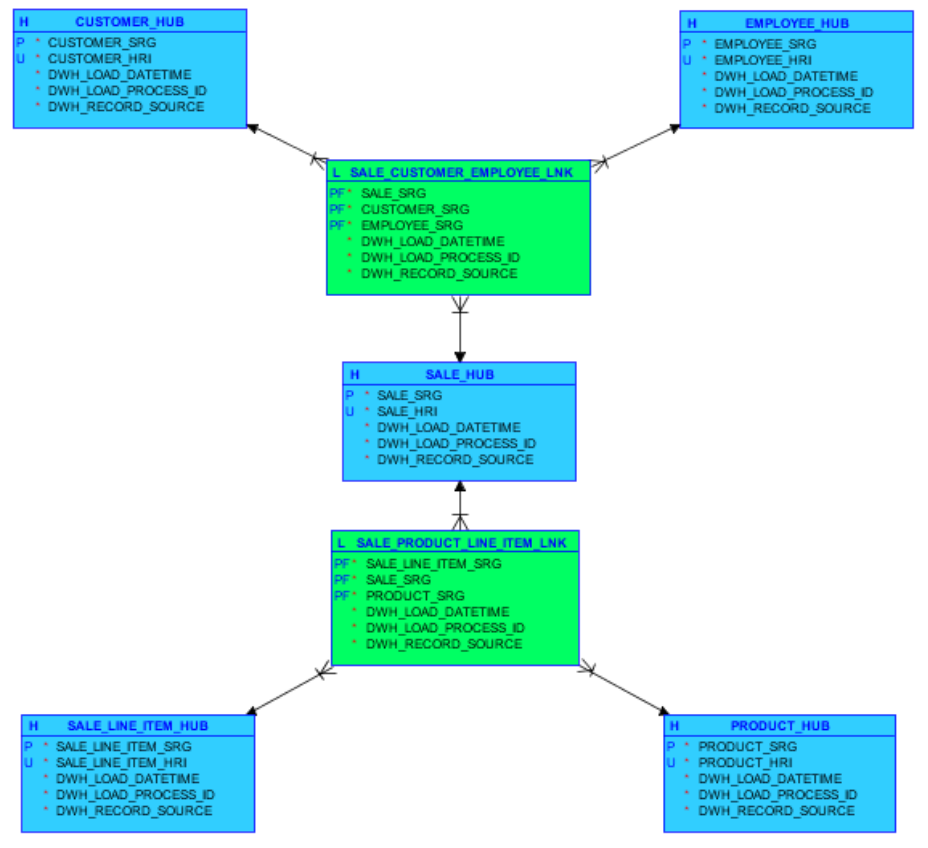
Header and Line Items: Dogbone

In this situation, there usually are two connections involving the respective event:

- one at the **header level** (e. g., connecting the Sale to a Customer, an Employee and a Store) and
- the other at the **line item level** (e. g., connecting the Sale to the Products that have been sold).

Because the resulting model tends to look like one, the header–one type of line items pattern is sometimes referred to as the **dogbone** pattern.

Header and Line Items: Example



Headers and Line Items: Wishbone

Some events are more complex and involve instances of two other concepts:

- a Car Repair event involving multiple Parts and multiple Service Actions,
- a Surgery event involving multiple Drugs and multiple Surgical Steps or
- a Car Accident event involving multiple damaged Cars and multiple injured Road Users.

Headers and Line Items: Wishbone

In this situation, there usually is

- **one connection** at the **header level** and
- **two different connections** at the **line item level**.

In the Car Repair example, one line item connection would connect the Car Repair event to the Parts used and the other line item connection would connect the Car Repair event to the Service Actions executed.

Because the resulting model tends to look like one, the header–two types of line items pattern is sometimes referred to as the **wishbone** pattern.

Header and Line Item: More LI Types

While events involving instances of three or more other concepts might seem unusual at first glance, they are surprisingly common:

- A Surgery can involve multiple Drugs, multiple Surgical Steps, multiple Surgical Instruments, multiple Nurses and multiple Surgeons.
- A Car Accident can involve damaged Cars, multiple injured Road Users, multiple Road Users that have witnessed the event, multiple Police Officers and multiple damaged Roadside Objects (streetlights, guard rails and so on).

Header and Line Item: More LI Types

The reasoning from the previous header–line items patterns applies accordingly:

- There usually is **one connection** at the **header level** (which translates into a link) and
- **three or more different connections** at the **line item level** (each of which translates into its own link, usually with a keyed-instance hub attached to it).

Data Vault Modeling Patterns

Lookups

Lookups

A **lookup** works like this:

- There is a code or surrogate identifier in one table (e. g., a status code).
- You can look up the description for that code in another table (that contains, in our example, all possible status codes with the respective descriptions).

In a data vault model, there are several different options for handling lookups.

Lookups: Implementation Options

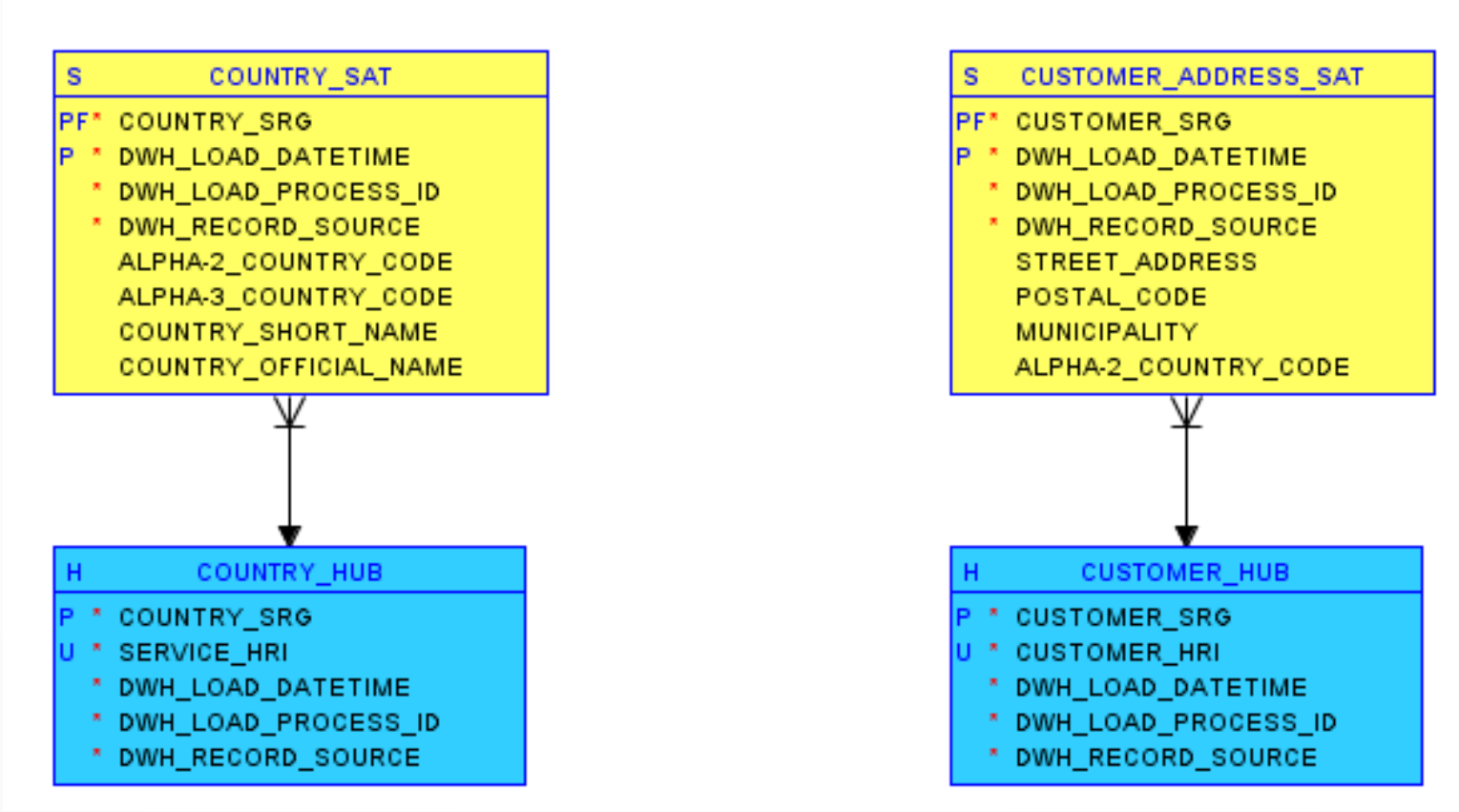
- Often, it is possible to simply put code and description in the **same satellite**.
- Sometimes, code and description are actually details about some other concept. Then, it makes sense to put them in a **satellite attached to another hub** standing for that new concept.
- And finally, you might create a **reference table** that contains code and description.

Lookups: Disconnected Pattern

When creating a new hub with a satellite for lookup data, you can either create a link to the new hub or use the so-called **disconnected pattern**:

- Just put the code in the existing satellite and then join directly to the code in the new hub to find the corresponding description.
- Save yourself the effort of building and loading a link.
- It’s not completely aligned with the usual data vault principles but you might want to make an exception for this limited use case.

Lookups: Example



Data Vault Modeling Patterns

Any Questions?