

You have **1 free member-only story left** this month. [Sign up](#) for Medium and get an extra one.

◆ Member-only story

Data Vault Naming Standards



Patrick Cuba · [Follow](#)

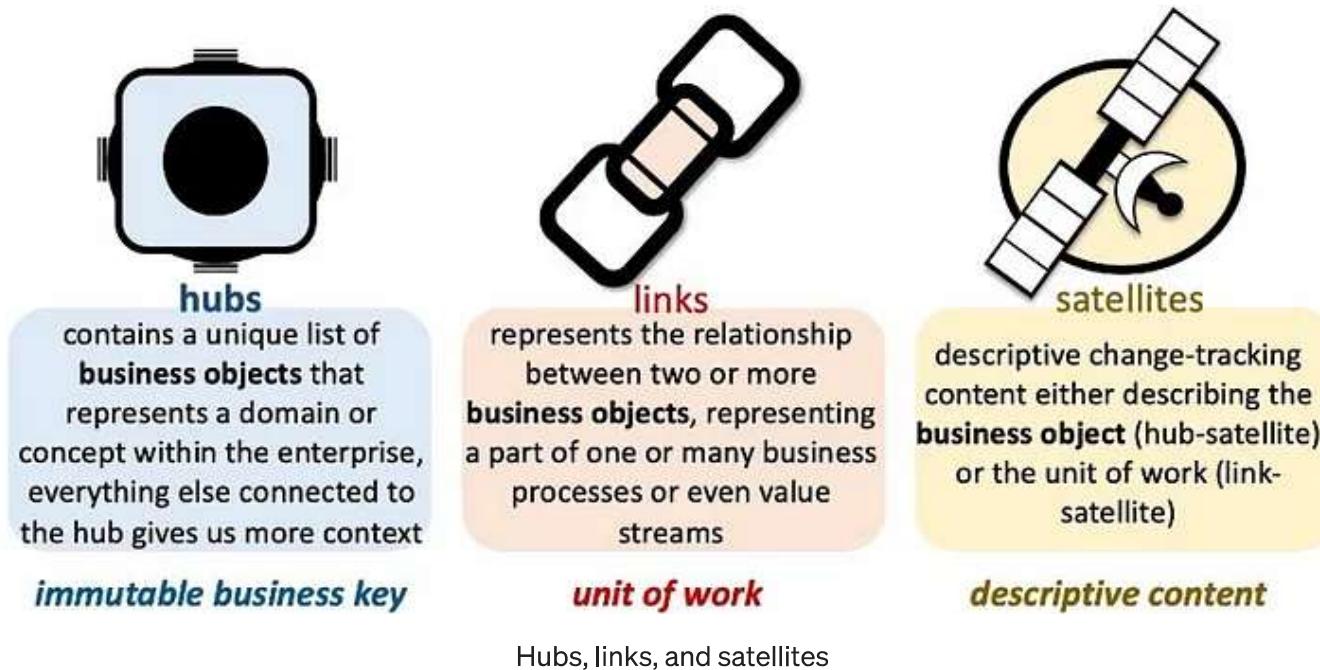
Published in Snowflake · 7 min read · Jan 17



Data Vault as a practice is happy for you as the practitioner to build whatever you need for your business case and ultimately your data platform. It does however require discipline and part of that discipline is establishing consistent naming standards.

Tables

Now, as a reminder your data vault table types,



The above are the **raw vault** table types, let's break that down to the naming standards you would expect:

Data Vault Table	Naming Convention
Hub table	$\${\text{hub-prefix}}_{_}\${\text{business-entity object concept}}$ Hub-prefix: hub_ / H_
Link table	$\${\text{link-prefix}}_{_}\${\text{unit-of-work transaction relation}}$ OR $\${\text{link-prefix}}_{_}\${\text{hub_1}}_{_}\${\text{hub_2}}_{_}\${\text{hub_n}}$ Link-prefix: lnk_ / L_
Satellite table	$\${\text{sat-prefix}}_{_}\${\text{source-badge}}_{_}\${\text{source-file}}$ Sat-prefix: sat_ / S_

If a link table **exclusively** represents a same-as link or a hierarchical relationship between business objects, then it should reflect as such in the naming standard

Data Vault Table	Naming Convention
Same-as link table	Link-prefix: lnk_sa_ / LSA_
Hierarchical link table	Link-prefix: lnk_hy_ / LHY_
Non-historized link table	Link-prefix: lnk_nh_ / LNH_
Exploration link table	Link-prefix: lnk_bv_ep_ / LEP_bv_

As for the extensions of satellite tables then the following applies

Data Vault Table	Naming Convention
Satellite with a dependent-child key / intra-day key	Sat-prefix: sat_dp_ / SDP_
Multi-active satellite table	Sat-prefix: sat_ma_ / SMA_
Status tracking satellite	Sat-prefix: sat_st_ / STS_
Record tracking satellite	Sat-prefix: sat_rt_ / RTS_
Effectivity satellite	Sat-prefix: sat_ef_ / EFS_
Extended record tracking satellite	Sat-prefix: sat_xt_ / XTS_
Non-historized satellite	Sat-prefix: sat_nh_ / NHS_

Should we designate the difference between a hub and link satellite? You can, if you want. The loading pattern is the same for both however the parent table of each could be reflected in either table. If you choose the

designate the difference, then the recommendation is to add the designation after the most generic of table designations like so:

```
sat_${hub|lnk}_ or s_${hub|lnk}_  
sat_${hub|lnk}_dp_ or sdp_${hub|lnk}_  
sat_${hub|lnk}_ma_ or sma_${hub|lnk}_  
sat_${hub|lnk}_st_ or sts_${hub|lnk}_  
sat_${hub|lnk}_rt_ or rts_${hub|lnk}_  
sat_${hub|lnk}_xt_ or xts_${hub|lnk}_  
sat_${hub|lnk}_nh_ or nhs_${hub|lnk}_
```

No EFS you say? Effectivity satellites are only ever about the relationship and therefore will only exist as a child table off a link table!

For all the tables (except for the hub table) business vault adds to the naming standard as well!

```
lnk_bv_ or L_bv_
lnk_bv_sa_ or LSA_bv_
lnk_bv_hy_ or LHY_bv_
lnk_bv_ep_ or LEP_bv_
sat_bv_${lnk}_ or s_bv_${hub|lnk}_
sat_bv_${lnk}_ma_ or sma_bv_${lnk}_
sat_bv_${lnk}_st_ or sts_bv_${lnk}_
```

[Sign up](#)[Sign In](#)

Search Medium



Write



itself) no distinction is necessary!

No NHS you say? Non-historized tables (old t-link and t-sat) are designed for immutable sources like streaming data, these are raw vault entities by definition!

Any link table variation could include a dependent-child key, you *could* include that in the name too (although I think it is unnecessary to do so).

```
Ink_${bv_dp_} or L_${bv_dp_}  
Ink_${bv_sa_dp_} or LSA_${bv_dp_}  
Ink_${bv_hy_dp_} or LHY_${bv_dp_}  
Ink_bv_ep_dp_ or LEP_bv_dp_  
sat_${bv}_${hub|ink}_dp_ or s_${bv}_${hub|ink}_dp_
```

No SMA you say? A multi-active satellite already has multiple active records by all attributes in that satellite table!

What about RTS, STS and EFS? Well, those are about the parent key!

Pro tip: once you have decided on a prefix it must be applied universally across your data vault model.

You may have noticed how the designations of these classification have been applied as **prefix** and **not suffix**. There's a reason for this, and it relates to taxonomic hierarchies (like you see in nature) and vocabularies as defined as:

"The relationship between and among terms in a controlled vocabulary that depicts broader (generic) to narrower (specific) or whole-part relationships; begins

with the broader term (BT) and leads to the narrower term (NT)" — ANSI/NISO Z39.19–2005 (R2010)

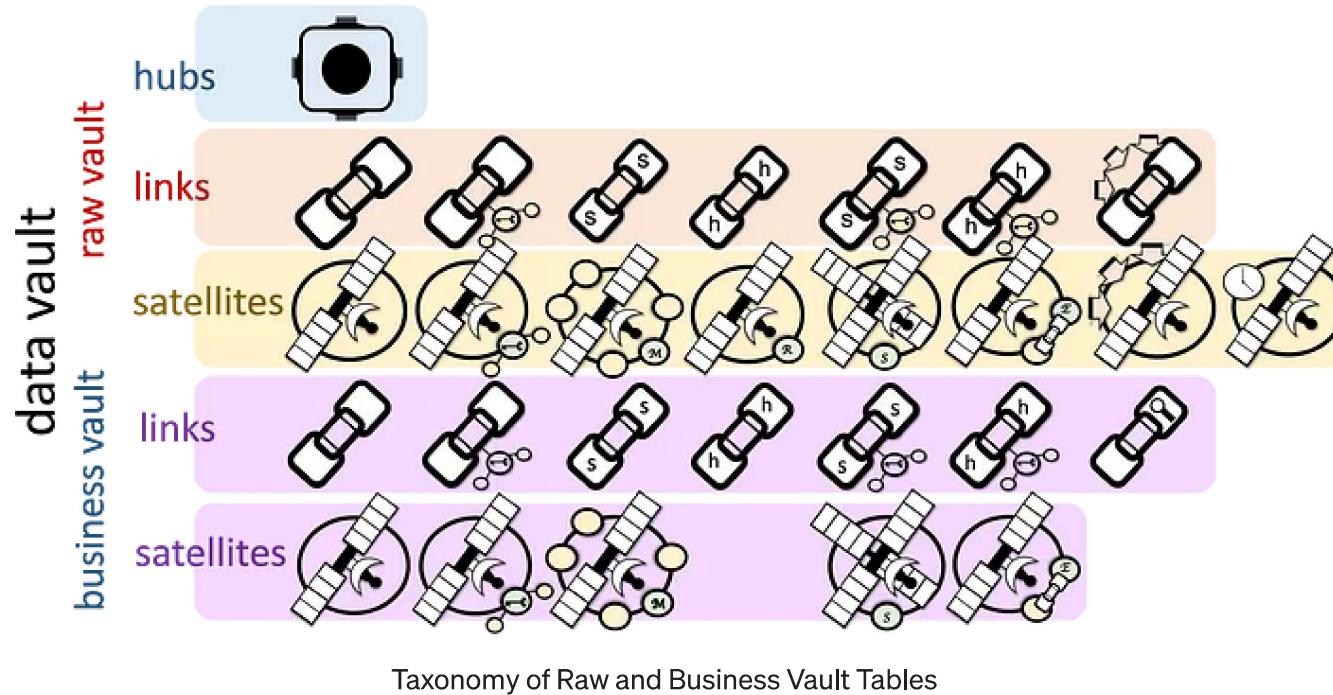
BT → NT

A broader term is a term to which another term or multiple terms are subordinate in a hierarchy. A narrower term that is subordinate to another or multiple terms in a hierarchy. The two together promote term navigation, from BT down to NT.

BT = superordinate term (parent) represents a class or a whole.

NT = subordinate term (child) represents a member or parts.

How does this navigation pan out in data vault?



From top to bottom:

- Hubs (*modelled* into raw vault, *referenced* in business vault)
- Links, (*modelled* into raw vault), from left to right, (simplest to more complex):
 - Link, link with dependent-child key(s), same-as link, hierarchy link, same-as link with dependent-child key(s), hierarchy link with dependent-child key(s),

- o Non-historized link

- Satellites (*modelled* into raw vault)), from left to right (simplest to more complex):

- o Satellite (hub or link), satellite with dependent child key(s) (hub or link), multi-active satellite (hub or link), record tracking satellite (hub or link), status tracking satellite (hub or link), effectivity satellite (link *only*),

- o Non-historized satellite (hub or link),

- o Extended record tracking satellite (hub or link)

- Links, (*modelled* into business vault), (simplest to more complex):

- o Link, link with dependent-child key(s), same-as link, hierarchy link, same-as link with dependent-child key(s), hierarchy link with dependent-child key(s),

- o Exploration link

- Satellites (*modelled* into business vault), (simplest to more complex):

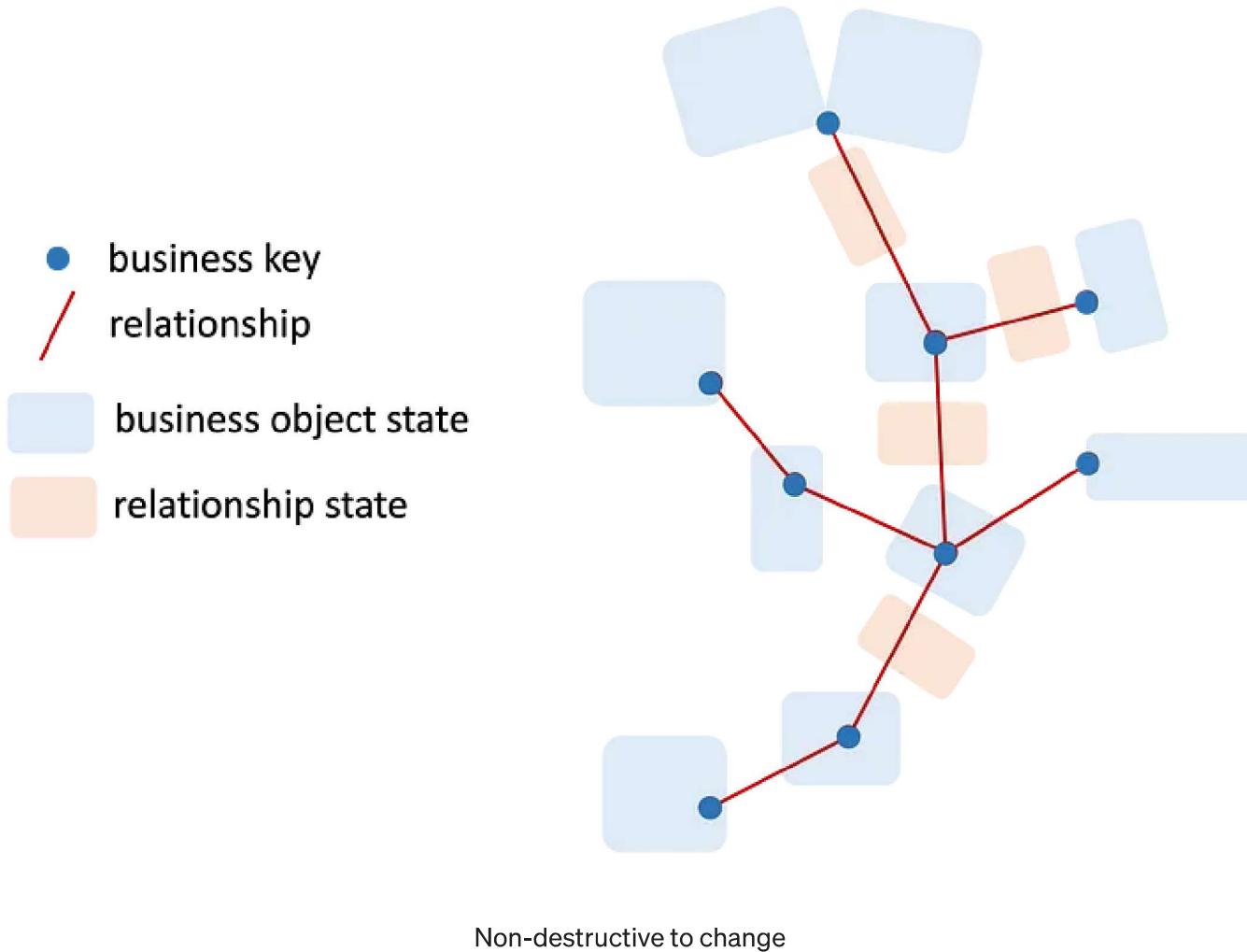
- o Satellite (hub or link), satellite with dependent child key(s) (hub or link), multi-active satellite (hub or link), status tracking satellite (hub or link), effectivity satellite (link *only*)

Columns

Every data vault table will include standard data vault columns we call dv-tags, *observe...*

Column	Hub	Link	Satellite
dv_hash_key_\${hub link-name}	✓	✓	✓
dv_load_timestamp	✓	✓	✓
dv_applied_timestamp	✓	✓	✓
dv_recordsource	✓	✓	✓
dv_hashdiff			✓
dv_object			✓ - for semi-structured satellite data
dv_sequence_id			✓ - used in SnoPITs
dv_subsequence_id			✓ - multi-active satellite
dv_collisioncode	✓		
dv_tenant_id	✓	✓	✓
dv_jira_id	✓	✓	✓
dv_task_id	✓	✓	✓
dv_status			✓ - status tracking satellite
dv_start_date			✓ - effectivity satellite
dv_end_date			✓ - effectivity satellite
dv_record_target			✓ - extended record tracking satellite

These columns are defined first, in the left most ordinance and non-dv-tag columns come after. This sets the expectation that every data vault artefact has its immutable tags on the left and variable columns thereafter on the right.



Depicting your business

As described under ANSI/NISO Z39.19–2005 (R2010) and illustrated above, naming standards under data vault reflect the labelling from BT to NT.

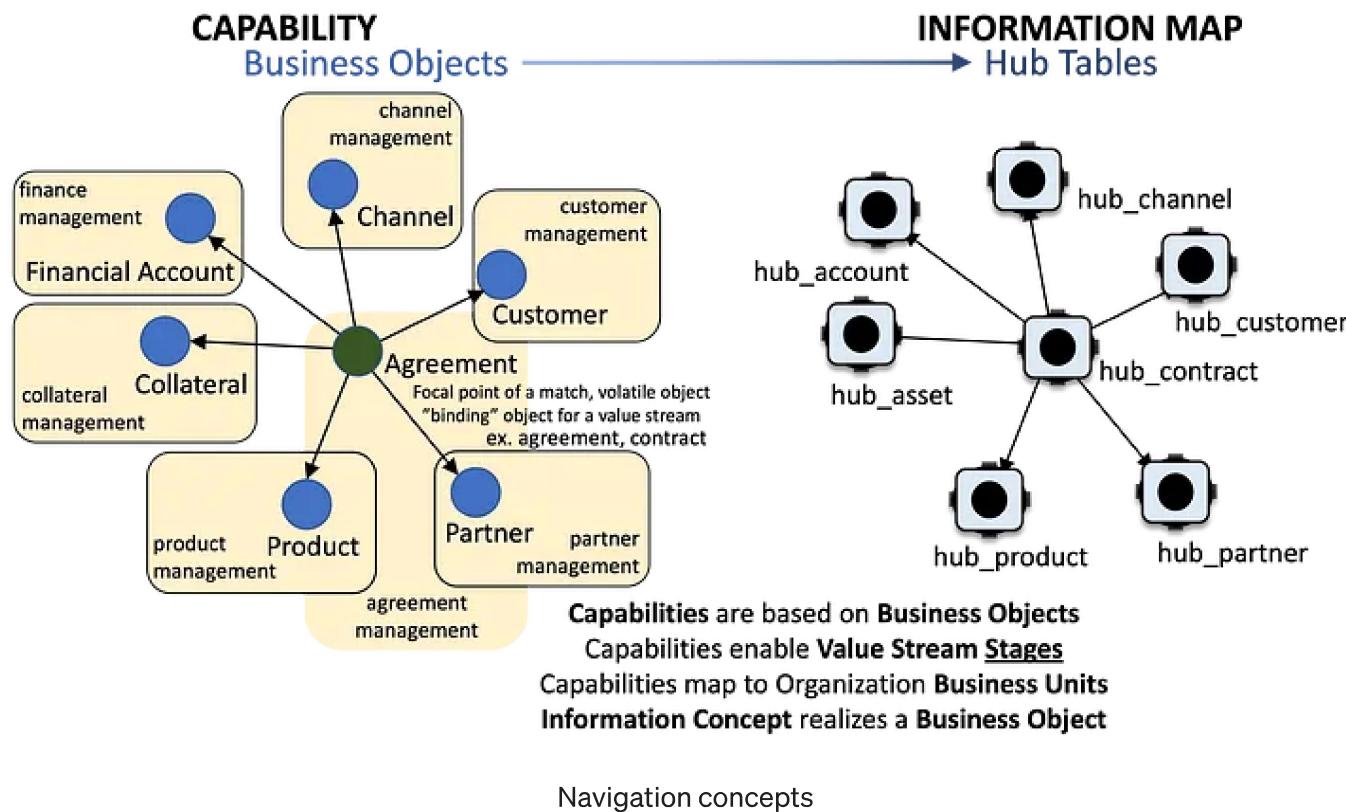
When modelling a data vault, one starts with

- The **business entities / objects / concepts**, followed by
- The **relationships, units of work and transactions** between the above entities, and to complete the model
- The **descriptive attributes** describing the above two points

From a navigation standpoint, exploring a data vault model could start by,

- *Binding/volatile* business object, ex. a contract or an agreement
- From there do we,
 - o require the *other* business entities/concepts the volatile object is related to, transacted with or the unit of work in that relationship? Or...
 - o require only the descriptive details of that entity/concept?

- Do we then also seek descriptive details of the other business objects in the relationship?



Now when one needs data for their business case it is easy to determine what *kind* of details the business is after. Raw and business vault are merely the data vault modelled output of business processes and rules, raw came

from a source system and business came from raw and/or other business vault artefacts.

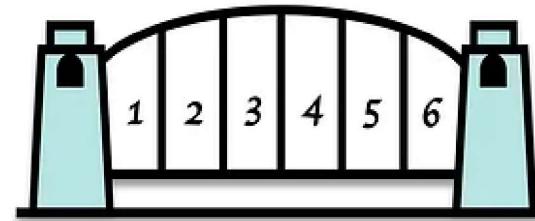
With the data vault naming standards and theory out of the way, let's turn our attention the so-called query-assistance tables designed to optimize querying out of the data vault, point-in-time (PIT) and bridge tables?

Getting the data out



disposable table designed
to simplify querying of
business object or
relationship state
attributes for information
marts

simplify with equi-join



Bridges
disposable table designed
to shorten the distance
between business objects
for information marts

shorten the distance

Query assistance tables

Yes, these should have naming standards too and they are tables, not views. PITs and Bridges defined as tables takes advantage of the native OLAP capabilities of the platform, as we have shown on [Snowflake](#).

Query Assistance Table	Naming Convention
PIT table	$\${\text{pit-prefix}}_{_}\${\text{business-concept unit-of-work}}_{_}\${\text{snapshot reporting-period}}$ PIT-prefix: pit_ / P_
Bridge table	$\${\text{bridge-prefix}}_{_}\${\text{business-concept unit-of-work}}_{_}\${\text{snapshot reporting-period}}$ Bridge-prefix: bdg_ / B_

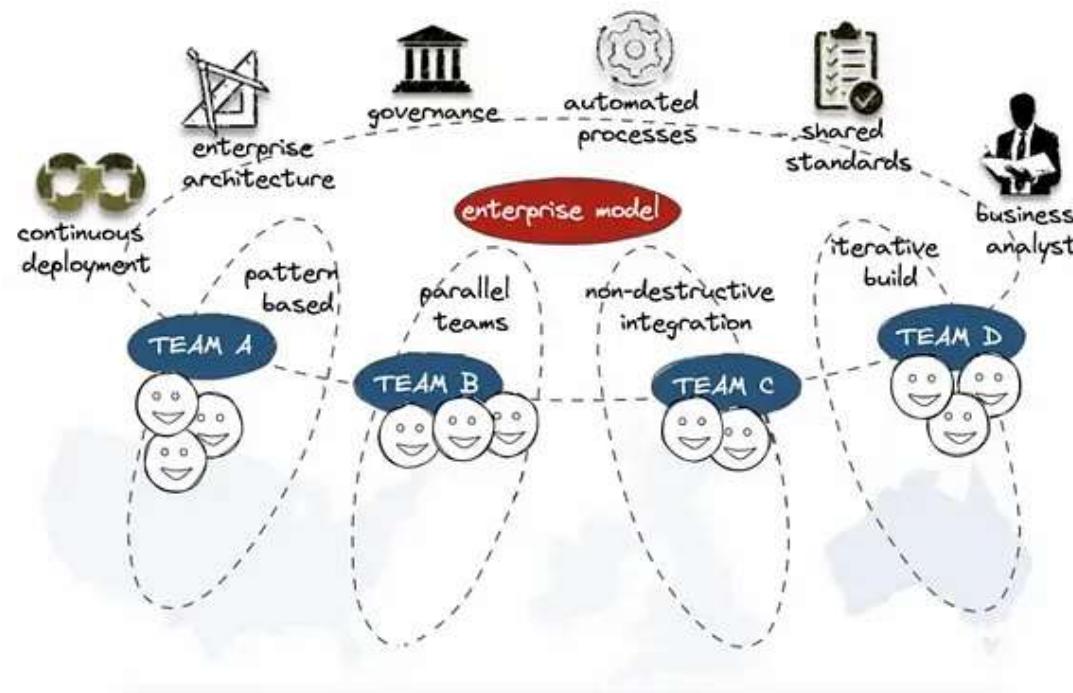
PITs and Bridges are snapshots of the applicable surrogate keys and applied date timestamps at that point in time. Therefore, the expected columns in these tables are the following.

Column	PIT	SnoPIT	Bridge
snapshotdate	✓	✓	✓
dv_hash_key_\${hub link-name}	✓		✓
dv_hash_key_\${hub link-name}_\${satellite-table-name}	✓		
dv_applied_timestamp_\${satellite-table-name}	✓		
dv_sequence_id_\${satellite-table-name}		✓	

A PIT should also contain the business key(s) of the business concept or unit of work, in addition you may also include lightweight calculations that can be made available for all information mart views. These are lightweight and reproducible and calculated at the aggregate partition the PIT is designated for, for example a weekly balance column to a weekly PIT.

Fin

Whatever standard you decide on, it must be centrally agreed upon to be universally applied throughout your enterprise data vault model. This makes it intrinsically easy to join tables and find the business object relationship and descriptive state you need.



The views expressed in this article are that of my own, you should test implementation performance before committing to this implementation. The author provides no guarantees in this regard.

Data Vault



Written by Patrick Cuba

1K Followers · Writer for Snowflake

Follow



A Data Vault 2.0 Expert, Snowflake Solution Architect

More from Patrick Cuba and Snowflake

The screenshot shows a Microsoft Excel spreadsheet titled "Credit Card Src to Tgt Mapping 0.2.xlsx". The spreadsheet contains a table with four columns: Date, Version, Author, and Change log. The data shows two entries: one for May 14, 2018, version 0.1, and another for May 23, 2018, version 0.2. The "Change log" column for version 0.2 includes the note: "Added join to target table to determine previous credit limit". The "Data Tools" ribbon tab is selected.

Date	Version	Author	Change log
14-May-18	0.1	Patrick Cuba	Initial Version (PC)
23-May-18	0.2	Patrick Cuba	Added join to target table to determine previous credit limit



Data Mapping

An oldie but goodie...



Snowflake launches Dynamic Tables

Better seamless Data Engineering in Snowflake

◆ · 9 min read · Aug 17, 2021

👏 32 💬 1



 Daniel Myers in Snowflake

How to Use OpenAI to Transform JSON in Snowflake

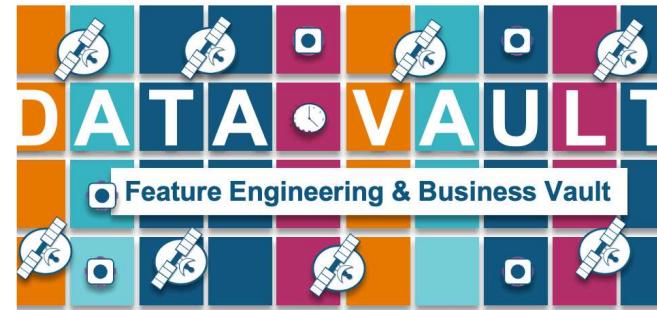
Transforming JSON at Scale using LLMs and Snowflake

6 min read · Jun 9

👏 106 💬

◆ · 2 min read · Jun 28

👏 192 💬 1



 Patrick Cuba in Snowflake

Data Vault on Snowflake: Feature Engineering & Business Vault

“The features you use influence more than everything else the result. No algorithm alon...

◆ · 16 min read · Mar 31

👏 30 💬 1

See all from Patrick Cuba

See all from Snowflake

Recommended from Medium



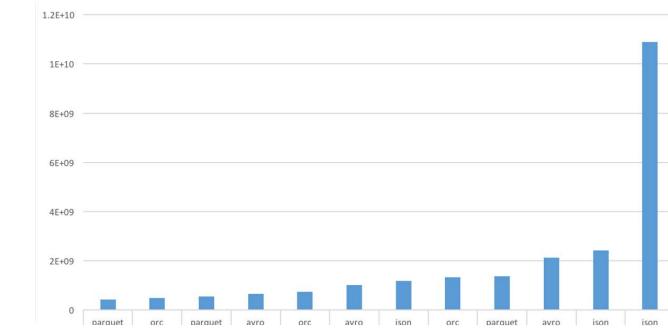
The Useful Tech in Mac O'Clock

9 New Must-Have macOS Productivity Apps For Daily Usage

How did you manage all these days without these apps?

★ · 11 min read · 6 days ago

1.6K 18



Park Sehun

Avro vs Parquet

Let's talk about the difference between Avro and Parquet.

★ · 5 min read · May 28

20



Lists

**Staff Picks**

382 stories · 140 saves

**Stories to Help You Level-Up at Work**

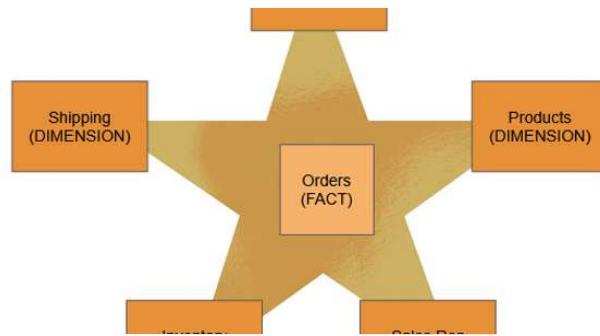
19 stories · 130 saves

**Self-Improvement 101**

20 stories · 221 saves

**Productivity 101**

20 stories · 243 saves



Manoj Kukreja in Towards Data Science

Handling Slowly Changing Dimensions (SCD) using Delta...

Handling the challenge of slowly changing dimensions using the Delta Framework

◆ · 10 min read · Jan 24

👏 118 💬 1

👏 6 💬 2


5 min read · Jun 6



6



2



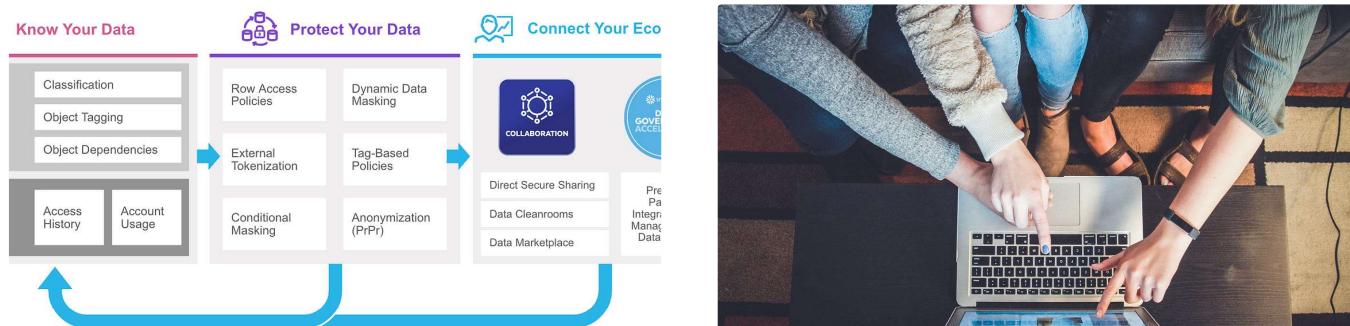
+



Keith (Datafluencer) Belanger

My Ah ha moment... A non-technical perspective on Data Vault

My mom has always said “Keith, Important things in life will happen when you least



Josh Crittenden in Snowflake

Exploring Snowflake Data Governance with Power BI

In this post, we'll delve into Snowflake's data governance features and explore how colum...

13 min read · May 26



29



See more recommendations



DataHub Community in DataHub

Starting a Data Governance Journey

Why do we need Data Governance at all?

8 min read · Feb 4



80



1

