

Data Vault Naming Standards

Naming Standards

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,

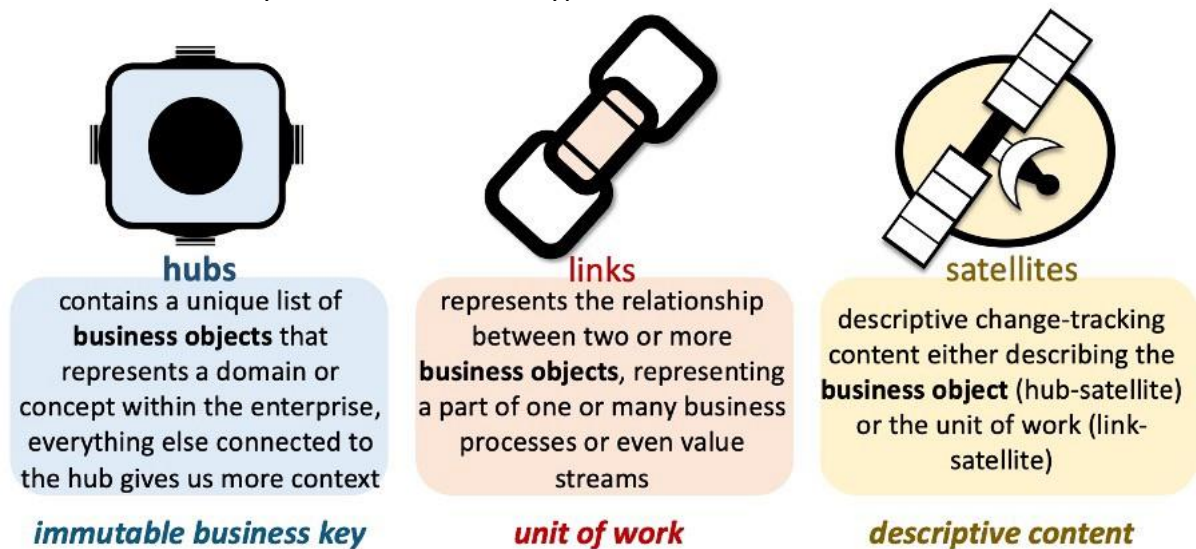


Figure 1 Hubs, links, and satellites

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	$\${\text{hub-prefix}}_{{\text{business-entity}} \mid {\text{object}} \mid {\text{concept}}}$ Hub-prefix: hub_ / H_
Link table	$\${\text{link-prefix}}_{{\text{unit-of-work}} \mid {\text{transaction}} \mid {\text{relation}}}$

Data Vault Table	Naming Convention
	OR $\${link-prefix}_{\${hub_1}}_{\${hub_2}}_{\${hub_n}}$ Link-prefix: lnk_ / L_
Satellite table	$\${sat-prefix}_{\${source-badge}}_{\${source-file}}$ 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}_
```

No XTS or RTS you say? An extended record tracking satellite hangs off a raw vault hub or link table to support all satellite tables surrounding that hub or link table, including business vault satellites. Therefore (like a hub table 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).

```
lnk_${bv}_dp_ or L_${bv}_dp_  
lnk_${bv}_sa_dp_ or LSA_${bv}_dp_  
lnk_${bv}_hy_dp_ or LHY_${bv}_dp_  
lnk_bv_ep_dp_ or LEP_bv_dp_  
sat_${bv}_${hub|lnk}_dp_ or s_${bv}_${hub|lnk}_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?

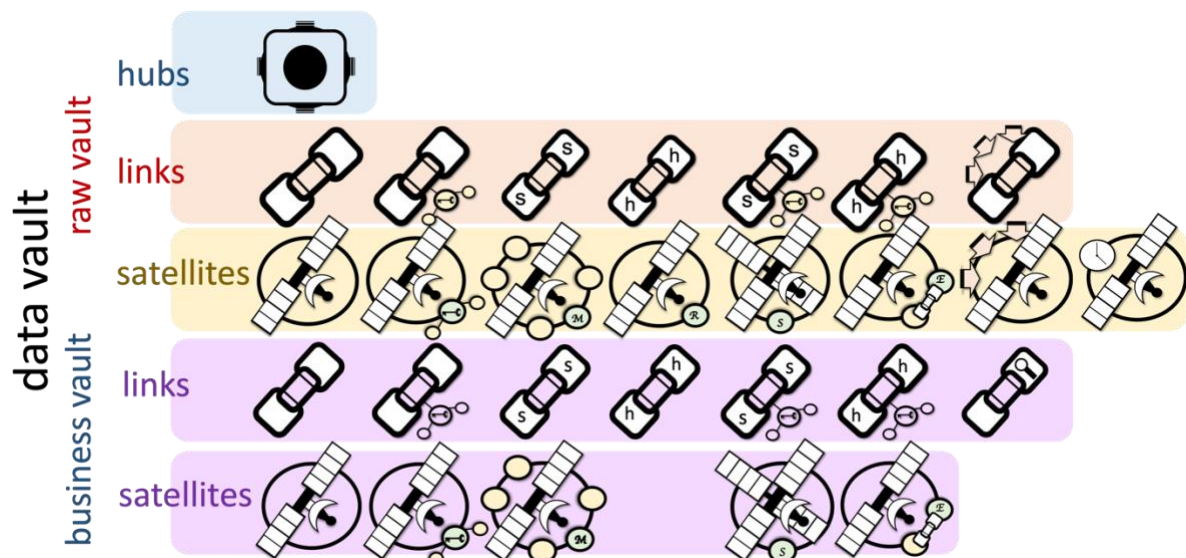


Figure 2 Taxonomy of Raw and Business Vault Tables

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),
- Non-historized link
- Satellites (*modelled* into raw vault)), from left to right (simplest to more complex):
 - 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*),
 - Non-historized satellite (hub or link),
 - Extended record tracking satellite (hub or link)
- Links, (*modelled* into business vault), (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),
 - Exploration link
- Satellites (*modelled* into business vault), (simplest to more complex):
 - 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

Column	Hub	Link	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.

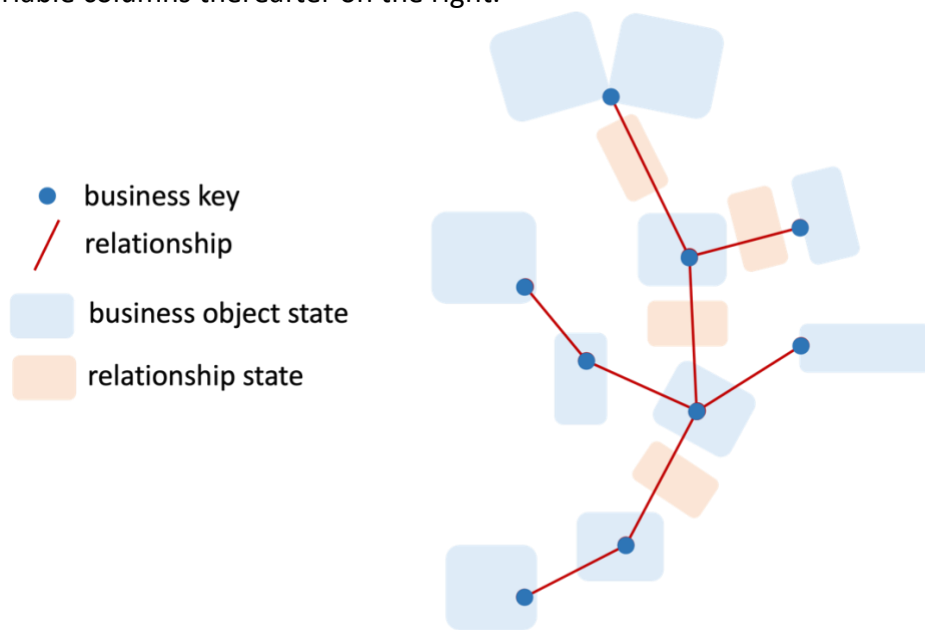


Figure 3 Non-destructive to change

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,
 - require the *other* business entities/concepts the volatile object is related to, transacted with or the unit of work in that relationship? Or...
 - require only the descriptive details of that entity/concept?
- Do we then also seek descriptive details of the other business objects in the relationship?

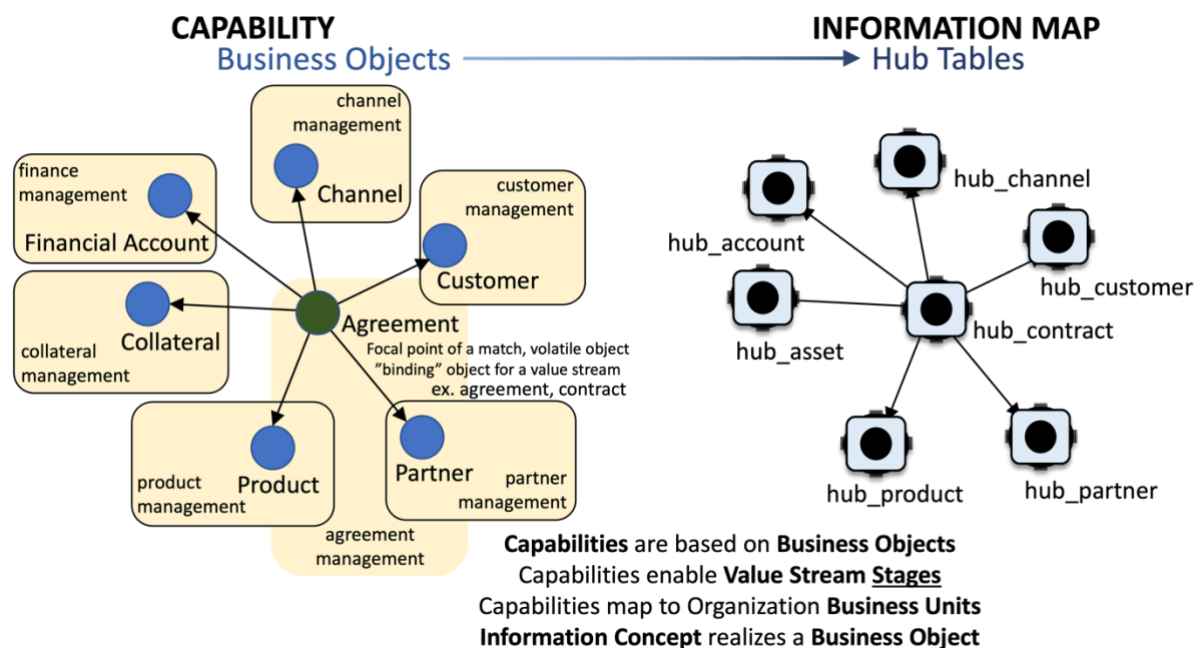


Figure 4 Navigation concepts

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

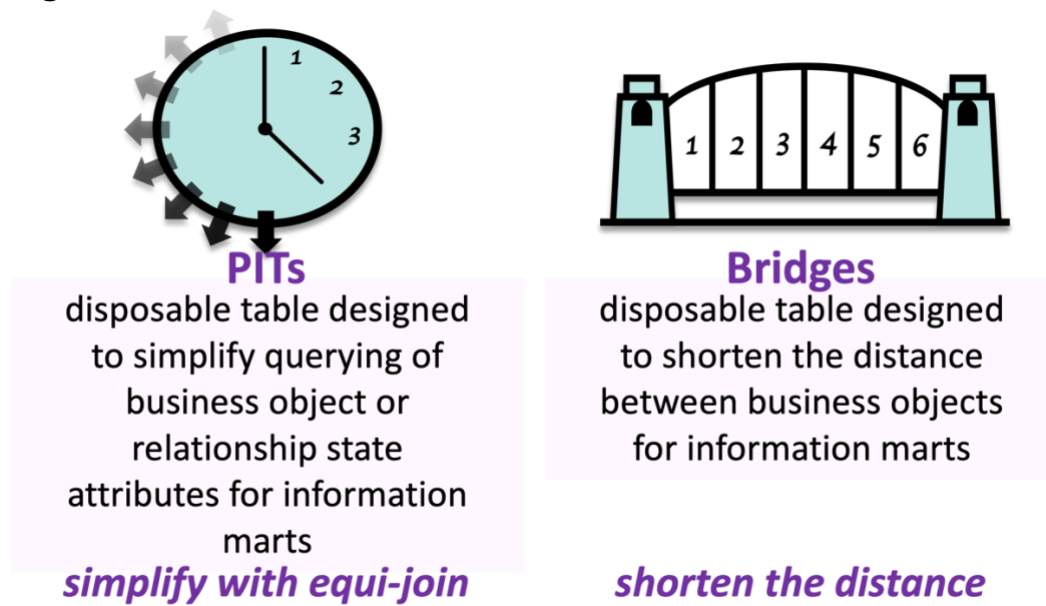


Figure 5 Query assistance tables

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

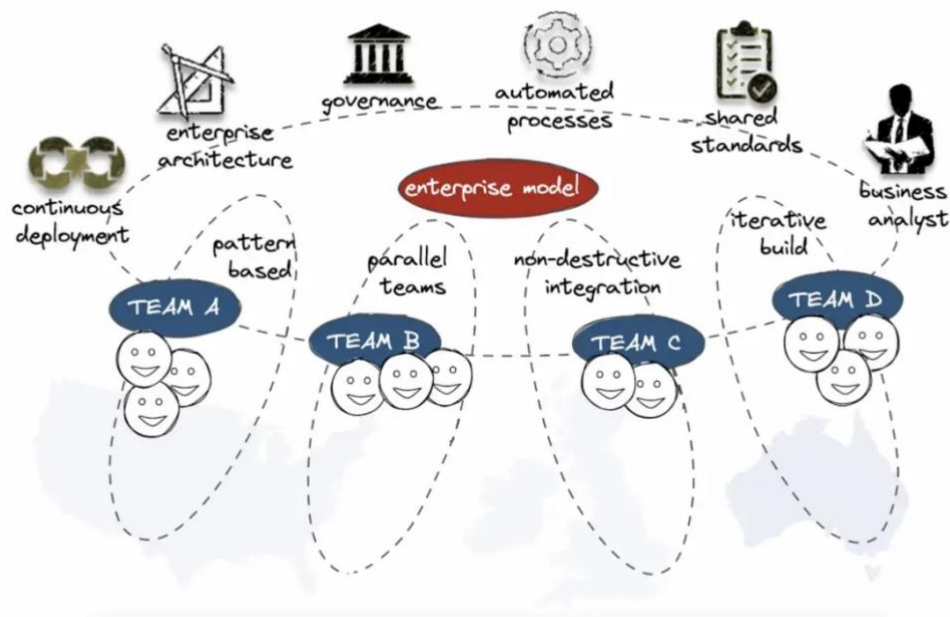
PITs and Bridges are snapshots of the applicable surrogate keys and applied date timestamps at that point in time. Therefore, the expected columns in these tables are the following.

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.