



On today's episode of Data Vault Mysteries we discuss Driving Keys and Effectivity Satellites!

One of the most complex and often misunderstood Data Vault 2.0 artefacts is the beloved **Effectivity Satellite**. Some argue it doesn't fit the **Insert-Only** architecture of Data Vault 2.0 but I can assure you, through a detailed example that an Effectivity Satellite does indeed fit the Data Vault 2.0 architecture and I will show you where and why you would even consider using Effectivity Satellite and why you would not.

## Relationship Effectivity

Although Data Vault 2.0 is INSERT-ONLY why then do we have a start **and end date** in an Effectivity Satellite?

To understand this let's first discuss relationship effectivity. Hypothetically if two business entities we are tracking have a relationship between each other, an account and a product (for instance) can be related in a one-to-one or one-to-many cardinality it stands to reason that if an account begins a relation with a product the relationship itself will have a start date with no end-date value. To draw a parallel with slowly changing dimension (type 2) this means the end-date is assigned a **high-date** like '9999-12-31' meaning that until the relationship changes, when we query this relationship it is the effective relationship until the **end of time**.

ACCOUNT_ID	PRODUCT_ID	INTEREST	START_DT	END_DT
123	ABC	4.50%	2018-10-01	9999-12-31
456	DEF	4.50%	2018-10-01	9999-12-31
789	GHI	4.50%	2018-10-01	9999-12-31

Table 1 Source Table, 1 October 2018

Note: we are using natural keys for simpler representation of the concepts!

When the relationship changes however, the account is now related to a **different** product (or products) we then have to end what we previously knew about this relationship, in a slowly changing dimension (Kimball style modelling) we would have updated that record and inserted the new effective (or active) record.

ACCOUNT_ID	PRODUCT_ID	INTEREST	START_DT	END_DT
123	ABC	4.50%	2018-10-01	9999-12-31
456	DEF	4.50%	2018-10-01	2018-10-09
456	ZZZ	4.50%	2018-10-10	9999-12-31
789	GHI	4.50%	2018-10-01	2018-10-09
789	GHI	5.00%	2018-10-10	9999-12-31

Table 2 Source Table, 10 October 2018

This can occur in an operation source as well and not just a data warehouse, imagine if this effectivity is being tracked in a system like master data management (MDM). MDM uses an **enterprise-type data model** with relationships between entities tracked with start and end dates and when sourced and loaded to Data Vault the model will look something like this, a bi-temporal link-satellite.

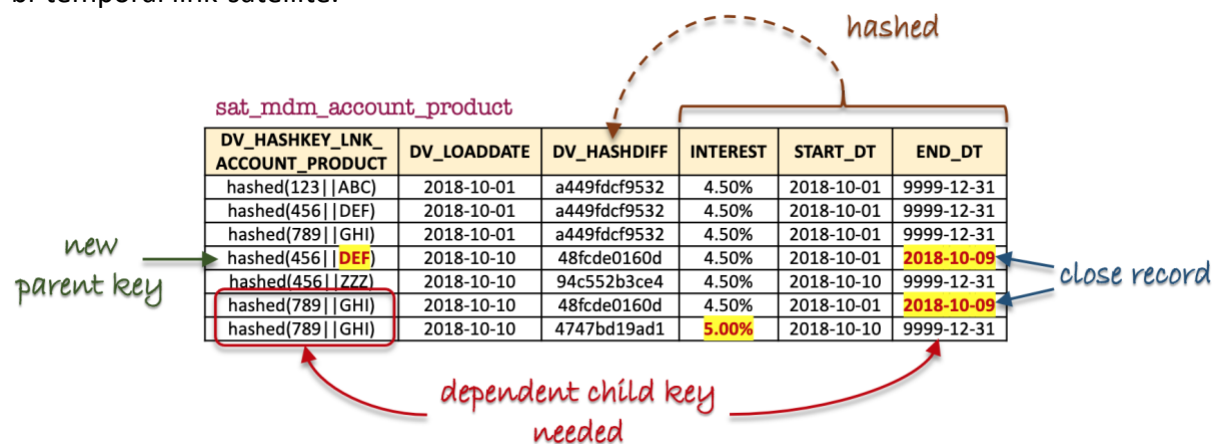


Figure 1 Link-Satellite

Although we have done nothing out of the ordinary in our link-satellite load, the content is loaded and hashed for the record hash and persisted into the link-satellite, notice however, the value of the surrogate link-hash-key...

Now remember, if the source system is doing the effectivity tracking then this is the best scenario, Data Vault scales naturally and simply ingests what the source tells us is happening to the account with its relation to the product. You could say the account number is the **master** entity MDM tracking the changes against.

Let's see what the effectivity satellite does for us shall we...

## Effectivity Satellite

Column	Description
DV_HASHKEY_LNK_ACCOUNT_PRODUCT	The parent link table's surrogate hash key

Column	Description
DV_LOADDATE	Load date timestamp of when the record was loaded
DV_START_DATE	Effective start date timestamp of the relationship
DV_END_DATE	Effective end date timestamp of the relationship
DV_HASHDIFF	Record Hash, this is made up of: DV_START_DATE    DV_END_DATE

Table 3 Effectivity Satellite structure, name: sat\_ef\_mdm\_account\_product

If the source does not provide a business date tracking the change in a relationship, or at least the business date to the cardinality of change we want then we request that they do. If they can't then we may have to derive this relationship movement ourselves. Now that we have defined account as the **master** entity, or rather, the entity **driving** the change tracking in the link. Yes that's right, account id is the **driving key** of the relationship. Could we simply rely on the link structure to show the effectivity change in the relationship? Let's see...

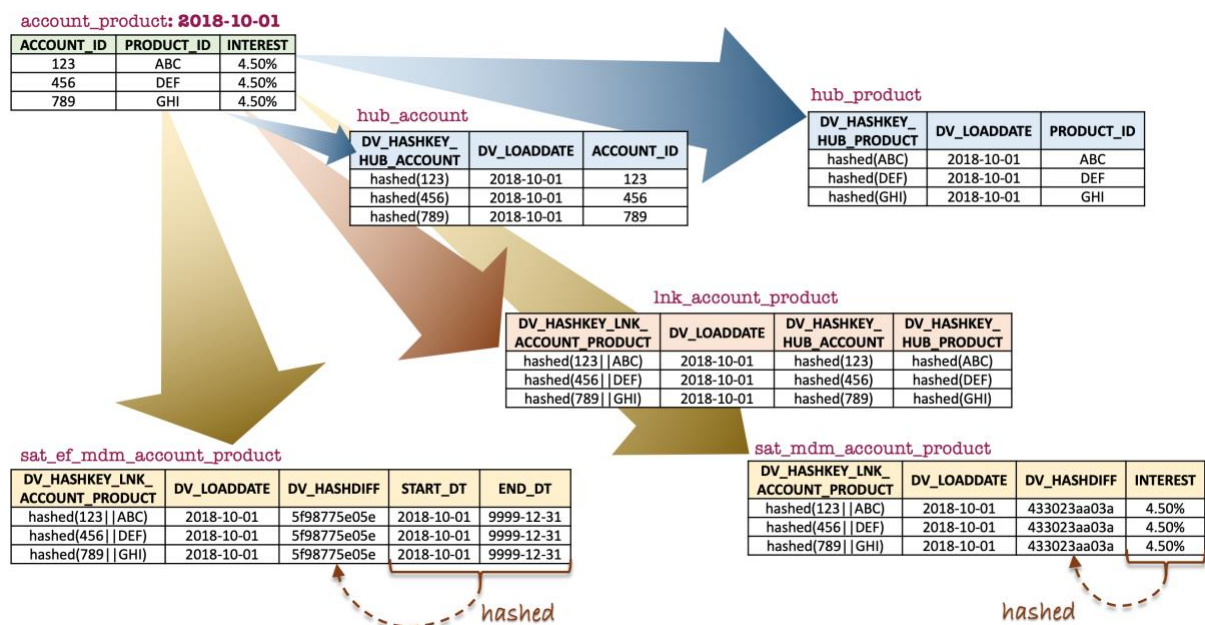


Figure 2 Day 1, 2018-10-01

Remember the effectivity satellite is only about the **relationship** tracking and nothing else, on day two we show some changes, notice how the link-satellite and effectivity satellite have the same change iterations....

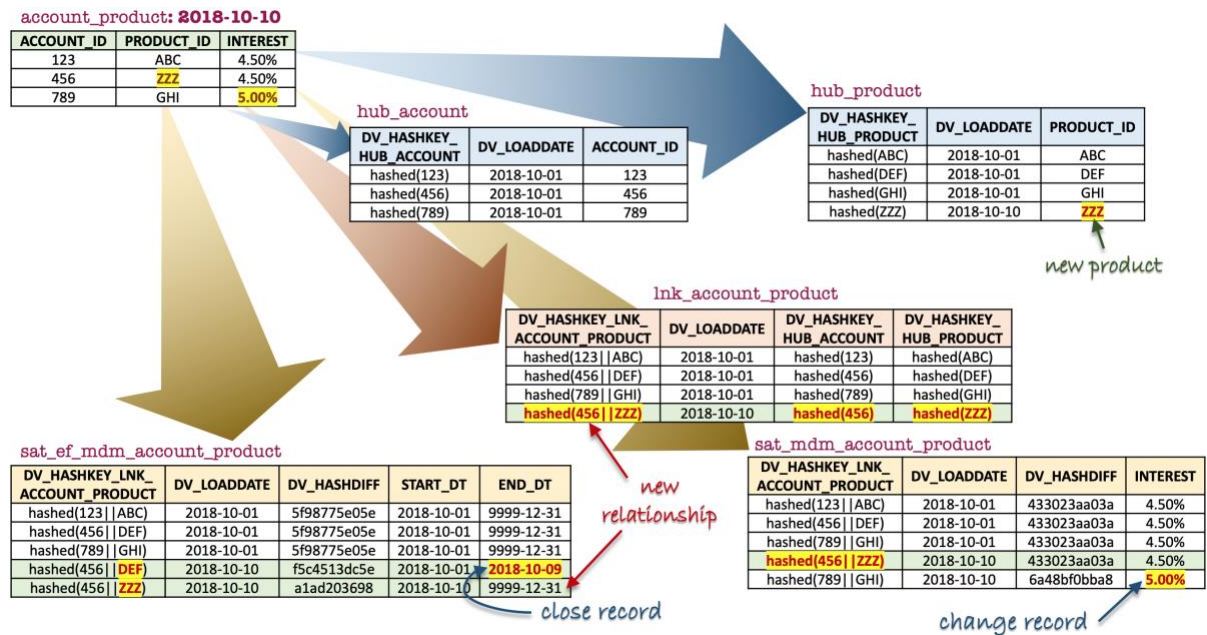


Figure 3 Day 2, 2018-10-10

Now, this is the tricky bit. What happens when the relationship goes back....?

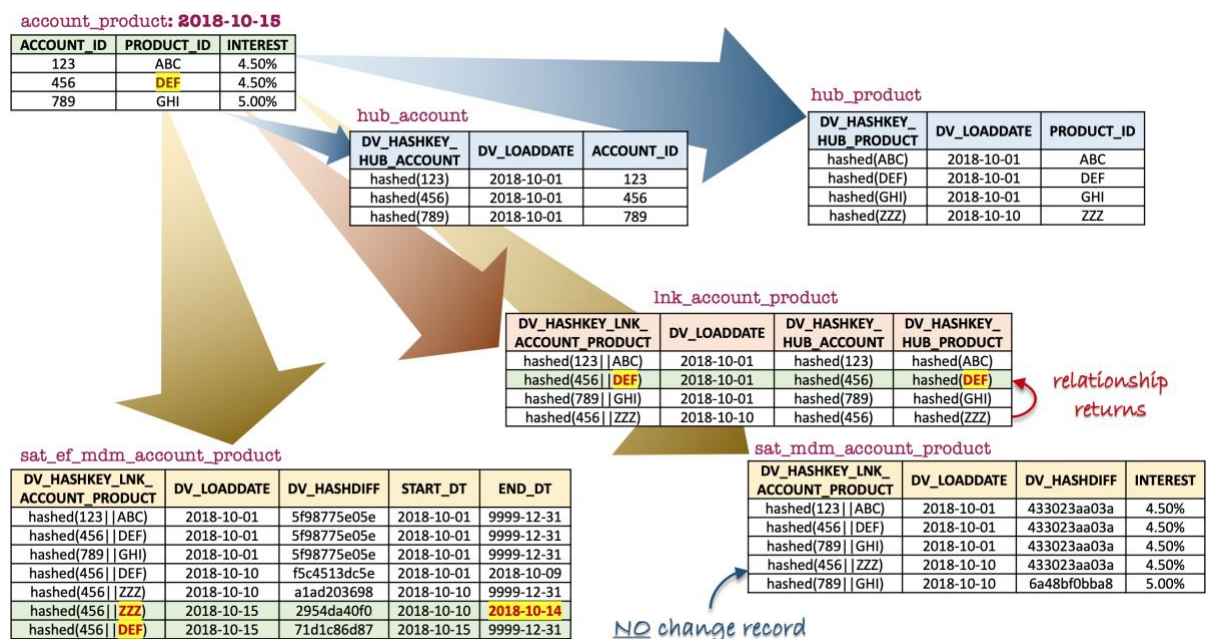


Figure 4 Day 3, 2018-10-15

Ah now you can see that the effectivity satellite is only about the relationship and nothing else, the end-date is persisted but in essence never updated. With no change to the "interest" column the only way to detect a relationship change is in the effectivity satellite, you cannot rely on the link table because the link table is a unique list of relationships and therefore by returning to the original relationship the load date of the previous relationship is not the **active relationship**.

To see this data model click here for IBM MDM's [locationgroup](#) and [addressgroup](#), the effectivity in these tables are tracked in START\_DT and END\_DT columns.

**Could we instead put effectivity in the link table structure?**



To answer that let's explore a similar relationship tracking satellite, the status tracking satellite. Remember, just like the effectivity satellite, we only really use these structures if we do not get this intelligence from the source data already. For effectivity satellite it's the business date of relationship changes to a driving key, for the status tracking satellite it's because we do not have **change data capture (CDC)** statuses for either business keys or relationships from the source.

For a two-part link effectivity the outcome of effectivity satellite and status tracking satellite will be the same, if it were a three-part link (three link participants / three hubs) the outcome between the effectivity and status tracking satellites will be very different...

### Effectivity in Status Tracking Satellite

Column	Description
DV_HASHKEY_LNK_CUSTOMER_ACCOUNT_PRODUCT	The parent link table's surrogate hash key
DV_LOADDATE	Load date timestamp of when the record was loaded
DV_STATUS	Status of parent key: 'I' for INSERT, new business key or relationship 'U' for UPDATE, business key or relationship is updated 'D' for DELETE, the business key or relationship is deleted
DV_HASHDIFF	Record Hash, this is made up of: DV_STATUS

*Table 4 Status Satellite structure*

Status tracking satellite can be applied to a business key (based on a hub) or unit-of-work (based on a link) from a source. Can we use the status tracking satellite to infer effectivity? It does not have a start and end date and no driving key defined. Let's compare the two, this time we are not showing the whole target satellites, just the records that get inserted and two keys are driver keys.

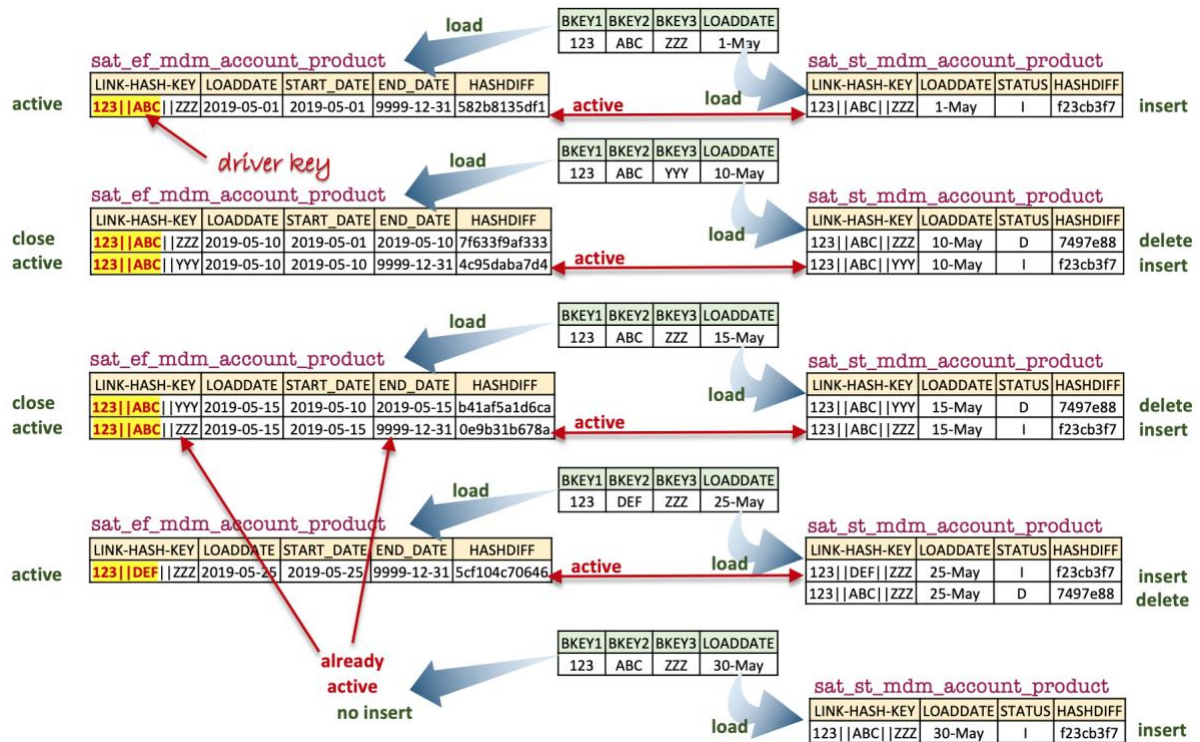


Figure 5 EFS vs STS, 2x Driver Keys

Now what happens when we change the active driving key columns from two to one, yes we can track different effectivity on the link based on a different set of driving keys. Of course they would against different effectivity satellites. In the following example we use the same source file (green header) but track different driver keys.

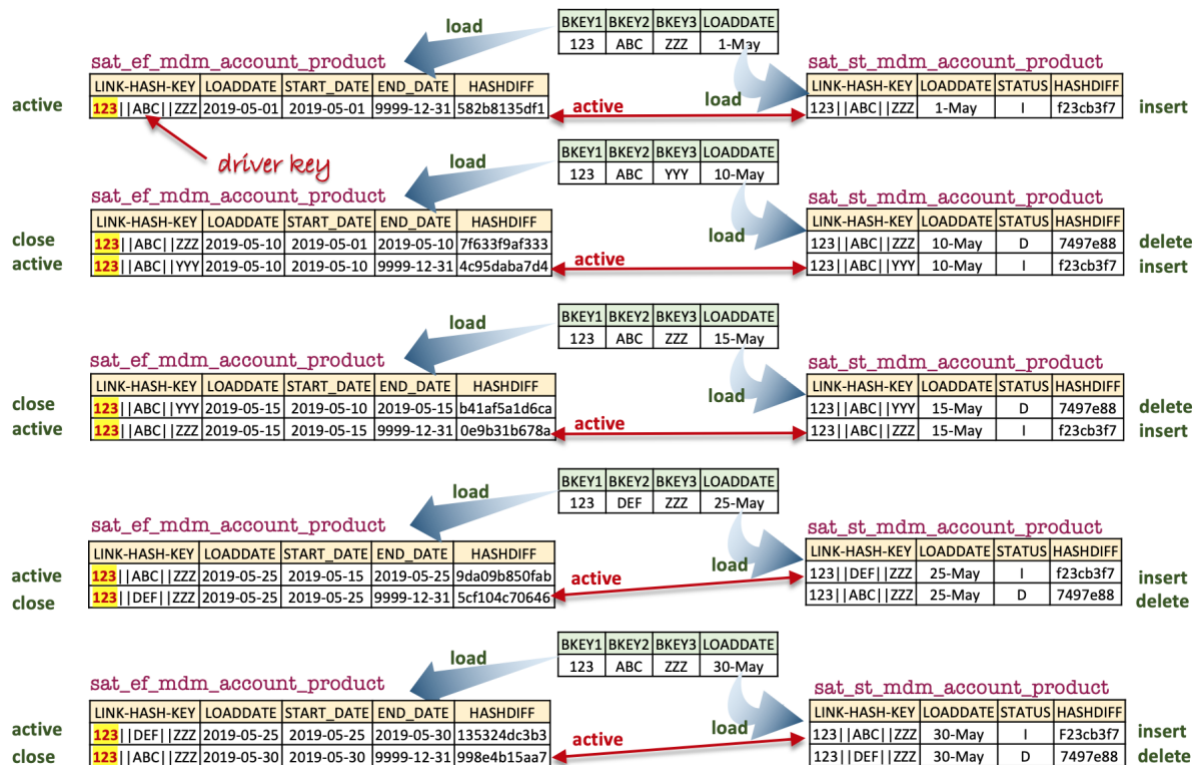


Figure 6 EFS vs STS, 1x Driver Key

A single driver key would cause the same output between the two, the caveat is that the source file is a snapshot of the source. Status Tracking Satellite in the traditional implementation does not work with delta loads.

### 1x driver keys

#### sat\_ef\_mdm\_account\_product

LINK-HASH-KEY	LOADDATE	START_DATE	END_DATE	HASHDIFF
123 ABC ZZZ	2019-05-01	2019-05-01	9999-12-31	582b8135df1
123 ABC ZZZ	2019-05-10	2019-05-01	2019-05-10	7f633f9af333
123 ABC YYY	2019-05-10	2019-05-10	9999-12-31	4c95daba7d4
123 ABC YYY	2019-05-15	2019-05-10	2019-05-15	b41af5a1d6ca
123 ABC ZZZ	2019-05-15	2019-05-15	9999-12-31	0e9b31b678a
123 ABC ZZZ	2019-05-25	2019-05-15	2019-05-25	9da09b850fab
123 DEF ZZZ	2019-05-25	2019-05-25	9999-12-31	5cf104c70646
123 DEF ZZZ	2019-05-25	2019-05-25	2019-05-30	998e4b15aa7
123 ABC ZZZ	2019-05-30	2019-05-30	9999-12-31	998e4b15aa7

#### sat\_st\_mdm\_account\_product

LINK-HASH-KEY	LOADDATE	STATUS	HASHDIFF
123 ABC ZZZ	1-May	I	f23cb3f7
123 ABC ZZZ	10-May	D	7497e88
123 ABC YYY	10-May	I	f23cb3f7
123 ABC YYY	15-May	D	7497e88
123 ABC ZZZ	15-May	I	f23cb3f7
123 DEF ZZZ	25-May	I	f23cb3f7
123 ABC ZZZ	25-May	D	7497e88
123 ABC ZZZ	30-May	I	f23cb3f7

### 2x driver key

#### sat\_ef\_mdm\_account\_product

LINK-HASH-KEY	LOADDATE	START_DATE	END_DATE	HASHDIFF
123 ABC ZZZ	2019-05-01	2019-05-01	9999-12-31	582b8135df1
123 ABC ZZZ	2019-05-10	2019-05-01	2019-05-10	7f633f9af333
123 ABC YYY	2019-05-10	2019-05-10	9999-12-31	4c95daba7d4
123 ABC YYY	2019-05-15	2019-05-10	2019-05-15	b41af5a1d6ca
123 ABC ZZZ	2019-05-15	2019-05-15	9999-12-31	0e9b31b678a
123 DEF ZZZ	2019-05-25	2019-05-25	9999-12-31	5cf104c70646

Figure 7 EFS vs STS, final tables

Of course I don't have to tell you that setting the driver key to **all** columns in a link does not make sense, I mean what are you comparing the driver key(s) against?! However the flexibility increases logarithmically as the number of participants in the link table increases. What I mean is, if

- there is no business date tracking relationship changes or;
- you want to track different cardinality to what is supplied by the source;

you could in fact create **as many** effectivity satellites from a single link table.

- 2 hub link: maximum 2 effectivity satellites,
- 3 hub link: maximum 4 effectivity satellite
- 4 hub link: maximum 8 effectivity satellites, etc.

The flexibility is there, if you were to introduce effectivity into the link table itself it means that every link table is a specialised configuration in the data vault model; i.e. the solution **will not scale**. It is also an anti-pattern if you build the data warehouse a switch-architecture, as we demonstrated above, you could have as many effectivity satellites as you want! In essence you would need as many links as there are effectivity requirements which could be disastrous to the number of tables you end up managing and the number of tables to join!

And in the governance itself the model is no longer self-describing! How do you pick which link table to solve your query? To those that suggest it is possible, go back to start and study Data Vault 2.0 all over again!

No, effectivity is a decoupled component for a reason... **it will scale**.

Note that the effectivity satellite not only solves

- Relationships that return to a previous relationship state,
- Business tracking of relationship effectivity,
- but also it does not matter what cardinality the relationship is either.



*Figure 8 Buzz finally realised that "upgrading" link tables with effectivity columns does not scale*

We have discussed relationships that are 1:1, but yes the same structure is usable in a 1:M cardinality as well, after all the link table itself is M:M and will always remain that way! Other peripheral satellite tables like record tracking and status tracking can live off a hub and link table but an effectivity table cannot, it is purely a link table offspring. Unless how you choose to populate an effectivity satellite is not in the standard driver to non-driver key way.

The last point to make about effectivity satellites is this, identify the need for a driver key early. As you saw in the illustrations above, the effectivity satellite captures **when a previous relationship returns**. What this means is, there is no other way in data vault to capture that effectivity by relying on a link table itself, an effectivity satellite is the **only** way to solve this source system gap, in the absence of the source providing a business date of that movement.

Use the effectivity satellite sparingly, if you need to implement it, and want to learn how to use it, all SQL is provided in the Data Vault Guru, see the link below.

[amzn.to/3d7LsJV](https://amzn.to/3d7LsJV)

**#datavault #datawarehouse #analytics #datamodelling #thedatamustflow**





Launch that satellite!