A Rose by any other name… WAIT.. is it still the same Rose?

a Data Vault perspective

A Rose by any other name… wait.. is it still the same Rose?

The title and subject of this discussion is the contextual nature of ***business keys***, keys are the ***immutable*** value that represents a ***thing***, ***entity***, ***interest*** or ***business object*** uniquely and is used by a business to identify that thing forever. However, a key by itself is *meaningless…*

Key
32451
29469
89978

Table 0-1 Keys without Descriptions

What information can we derive from these key values? Nothing really *(smart-keys excepted)*.
Equally, ***descriptive data*** without an immutable key is of ***little value*** to the business and its **business processes***.* Being able to track a business entity through its life cycle in the business is a requirement of ***business systems***.

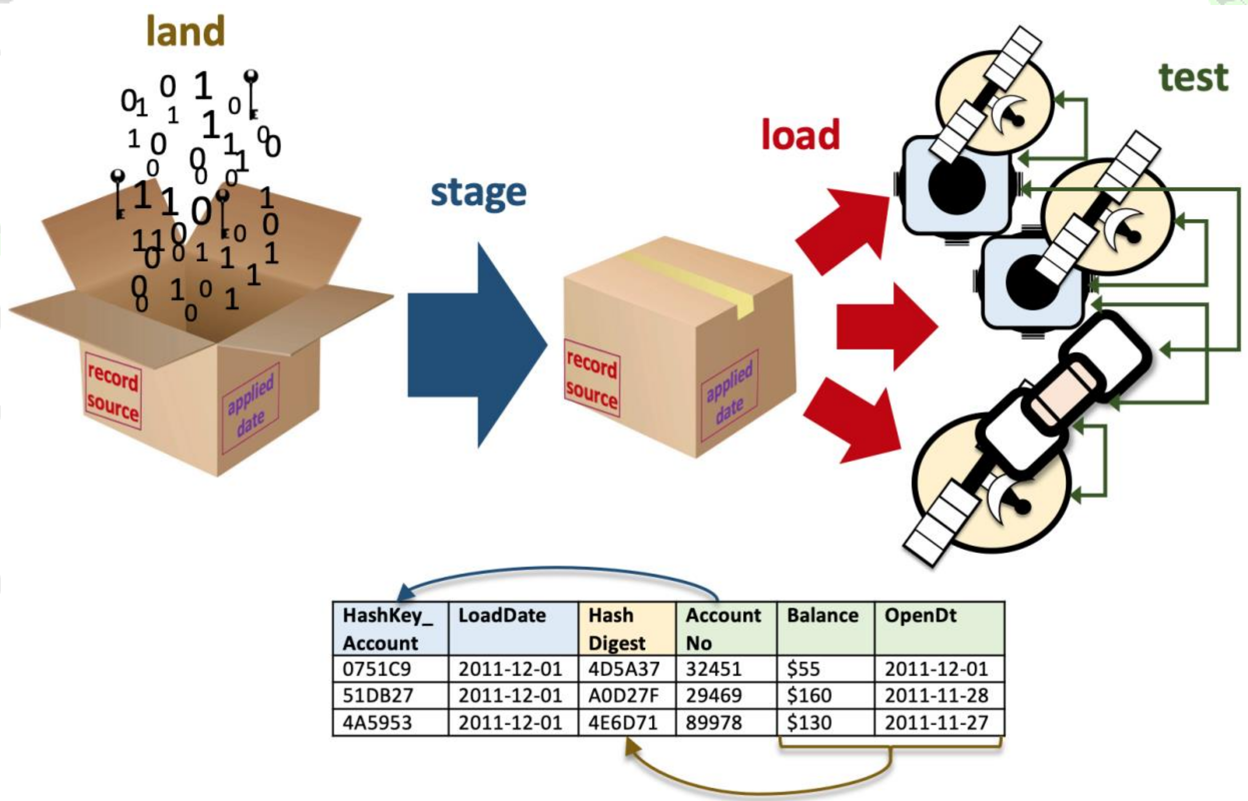
Balance	OpenDt
$55	2011-12-01
$160	2011-11-28
$130	2011-11-27

Table 0-2 Descriptors without a key

We can agree that in describing a customer we need to ***identify it***, If we were to load a data vault hub table with unique keys we see that the business key in fact is just a value with ***no context***. Only when queried with a ***satellite*** table do we learn the historical descriptive change data about that business key, and when queried with a ***link*** table we determine how the entity is ***related to other entities***.

AccountNo	Balance	OpenDt
32451	$55	2011-12-01
29469	$160	2011-11-28
89978	$130	2011-11-27

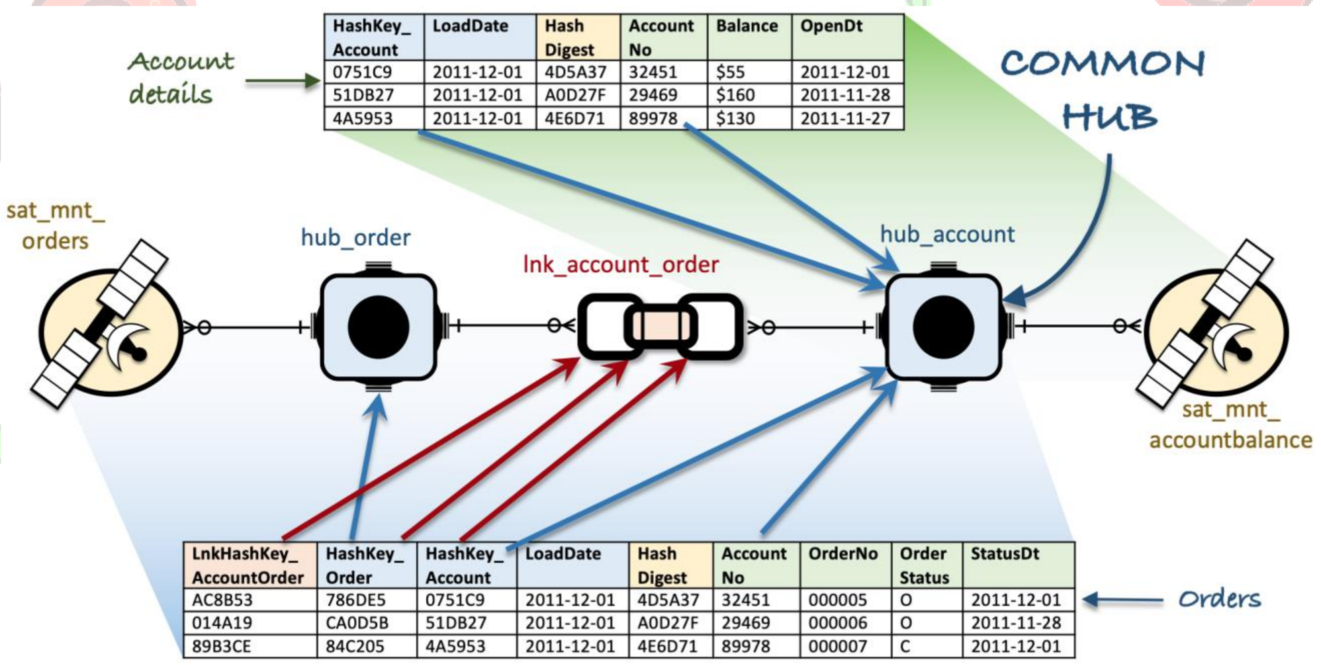
A source system (3rd party or built internally) exists to automate and manage the data that represents the ***business processes*** and ***events*** the organization is invested in. Simplified, these could be to track customer details (name, date of birth…), addresses, contracts, accounts, properties, contact details, orders, products, the things that are ***core*** to your business. Data ***packaged*** as either a snapshot or a delta (*only changes*) is ***landed*** (push or pull) and ***staged*** with hard rules applied (basic column clean-up & time zone alignment) and data vault metadata tags (record source, load date timestamp, applied date timestamp, surrogate hash keys and record digest) and loaded to their respective data vault artefacts (hubs, links and satellites).



HashKey_ Account	LoadDate	Hash Digest	Account No	Balance	OpenDt
0751C9	2011-12-01	4D5A37	32451	$55	2011-12-01
51DB27	2011-12-01	A0D27F	29469	$160	2011-11-28
4A5953	2011-12-01	4E6D71	89978	$130	2011-11-27

Figure 0-1 Data is landed, staged, loaded and tested (sample includes some DV-tags shortened hash digest)

A single source platform will have multiple source files to pull/push into a landing zone, each focussing on a ***unit of work*** that will load to hubs, links and satellites and in some cases data will be loaded to common hub tables. Let's expand our business model to include orders…



HashKey_ Account	LoadDate	Hash Digest	Account No	Balance	OpenDt
0751C9	2011-12-01	4D5A37	32451	$55	2011-12-01
51DB27	2011-12-01	A0D27F	29469	$160	2011-11-28
4A5953	2011-12-01	4E6D71	89978	$130	2011-11-27

LnkHashKey_ AccountOrder	HashKey_ Order	HashKey_ Account	LoadDate	Hash Digest	Account No	OrderNo	Order Status	StatusDt
AC8B53	786DE5	0751C9	2011-12-01	4D5A37	32451	000005	O	2011-12-01
014A19	CA0D5B	51DB27	2011-12-01	A0D27F	29469	000006	O	2011-11-28
89B3CE	84C205	4A5953	2011-12-01	4E6D71	89978	000007	C	2011-12-01

Figure 0-2 Two source files that map to a common hub table but separate units of work

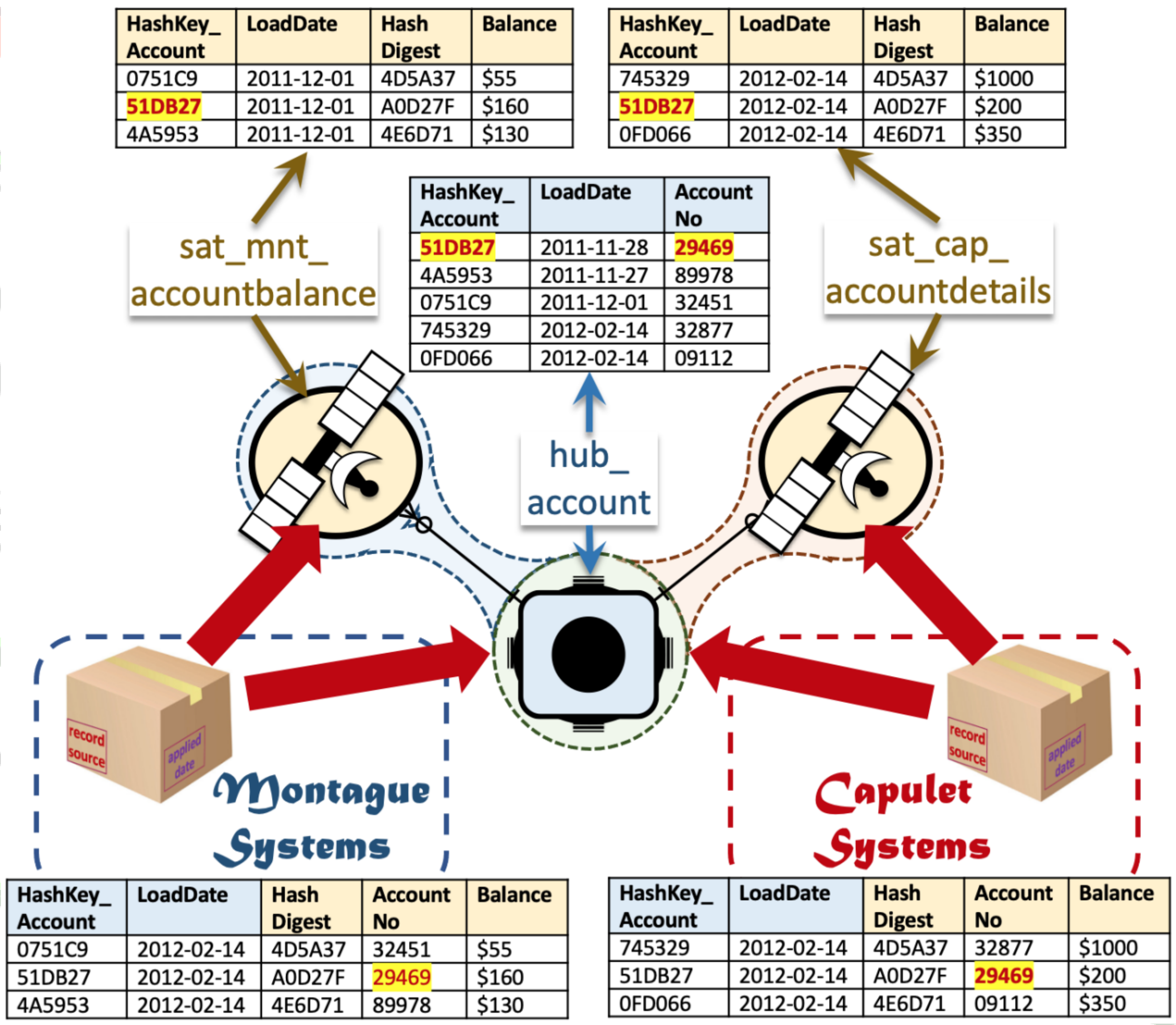
The business key is ***treated*** and loaded to the hub table (*unique list of business entities*) with a hash value representing those business entities uniquely and the descriptive content loaded to adjacent satellite tables and relationships loaded to adjacent link tables. Although the common hub has multiple sources the content is still ***a unique list of business keys…***

HashKey_ Account	LoadDate	Account No
51DB27	2011-11-28	29469
4A5953	2011-11-27	89978
0751C9	2011-12-01	32451

*Table 0-3 Same business key, hub loader will only load **new** business keys*

Business key collisions

Now let's consider another data source added to our business, for example in an acquisition a fictitious corporation called ***Montague Systems*** purchases ***Capulet Corp***. As it happens Capulet Corp tracks account numbers and orders too! And here is the situation, account numbers are in the format ***nnnnn*** and they have tracked their own business entities according to their own business! We need to ***integrate*** the new source into data vault but we know with certainty that the ***same business key*** that appears in each source system will ***not be the same business entity***!



HashKey_ Account	LoadDate	Hash Digest	Balance
0751C9	2011-12-01	4D5A37	$55
51DB27	2011-12-01	A0D27F	$160
4A5953	2011-12-01	4E6D71	$130

HashKey_ Account	LoadDate	Hash Digest	Balance
745329	2012-02-14	4D5A37	$1000
51DB27	2012-02-14	A0D27F	$200
0FD066	2012-02-14	4E6D71	$350

HashKey_ Account	LoadDate	Account No
51DB27	2011-11-28	29469
4A5953	2011-11-27	89978
0751C9	2011-12-01	32451
745329	2012-02-14	32877
0FD066	2012-02-14	09112

HashKey_ Account	LoadDate	Hash Digest	Account No	Balance
0751C9	2012-02-14	4D5A37	32451	$55
51DB27	2012-02-14	A0D27F	29469	$160
4A5953	2012-02-14	4E6D71	89978	$130

HashKey_ Account	LoadDate	Hash Digest	Account No	Balance
745329	2012-02-14	4D5A37	32877	$1000
51DB27	2012-02-14	A0D27F	29469	$200
0FD066	2012-02-14	4E6D71	09112	$350

Figure 0-3 AccountNo has a collision between Montague and Capulet!

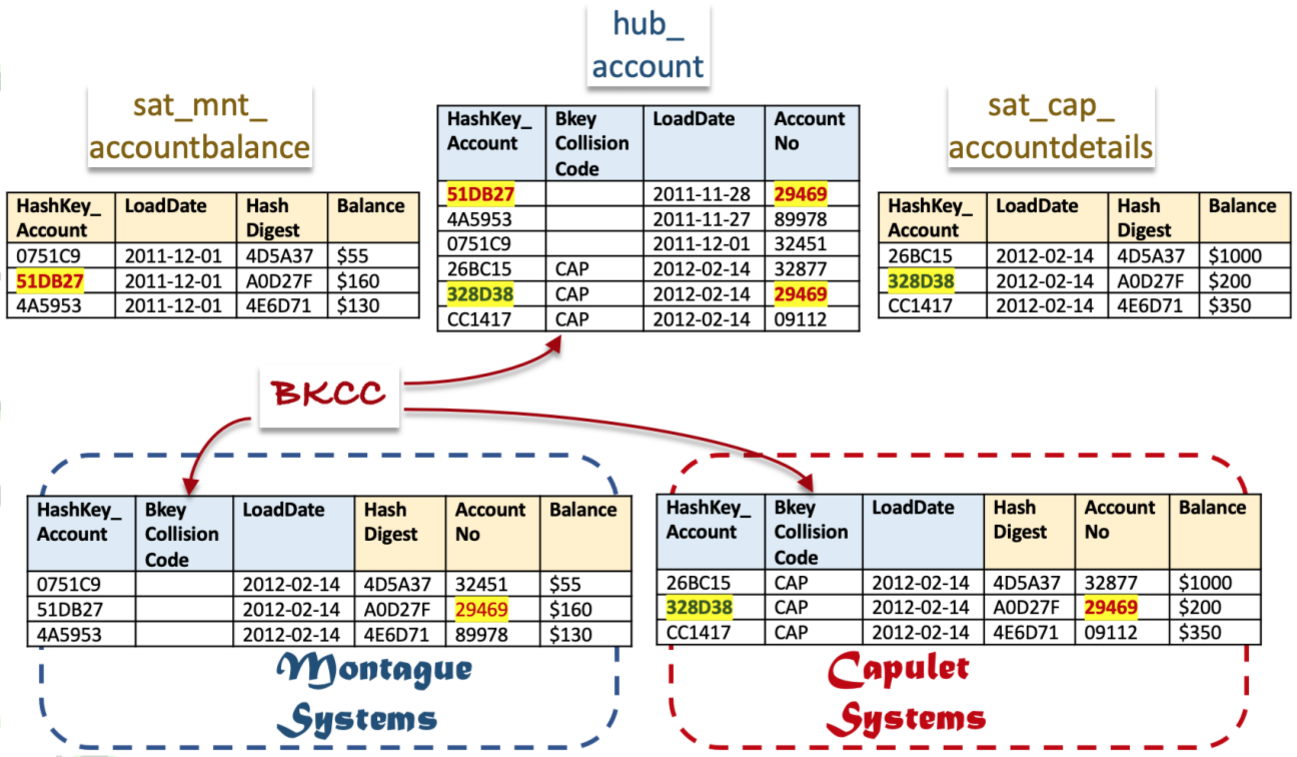
Account number '29469' has a balance of $160 in Montague Systems **and** a balance of $200 in Capulet Corp but they are not the same business entities!
How is this resolved? The very nature of hashing is that if you ***change*** just a ***single byte*** character in the value digest then the generated hash value ***changes completely*** but still maintains the ***same digest value length***.

Hash algorithm	Bit length	Word size	Availability	Collision
MD5	128	4x32	Widely used	Extensive vulnerabilities, useful for even distribution
SHA1	160	5x32	Widely used	Vulnerable, SHAttered.io
SHA256	256	8x32	Widely used	Stable, according to Wikipedia bitcoin runs a double SHA256 hash at 300 quadrillion hashes per second without collision.

Hash algorithm	Bit length	Word size	Availability	Collision
Murmur	32		Used in distributive computing	Vulnerable, not intended to be cryptographic

Table 0-4 *"Hash Collision Probabilities", bit.ly/37O3rT3*

Because we have various data sources and applications whose business keys are loaded to a ***common*** hub table, these natural keys if representing the ***same business entity*** naturally integrate in the hub. If a source loading to the common hub has the same key ***but*** it is in fact a ***different business entity*** we then simply ***add a prefix*** to that business key, in a separate column, and we call it a ***business key collision code (or BKCC)***, only applied if there is the ***possibility*** of a collision between having the same business keys creating the ***same hash key*** within the ***same*** hub table even though they are ***different business entities***.
Once applied the business key collision code is used along with the business key(s) to produce a ***consistent*** surrogate hash key value, that is one of the characteristics we rely on in data vault, the ***same values*** using the ***same hashing algorithm*** and **business key treatments** will consistently produce the ***same surrogate hash key value*** which leads to ***passive integration*** of the business process data into the data vault model.
Let's apply this to the data vault model, remember ***only** use the business key collision when there* **could** *be a business key collision*. In some environments natural keys that represent the same business entities are ***shared*** across systems and thus does not need a collision code.



hub_account

HashKey_Account	Bkey Collision Code	LoadDate	Account No
51DB27		2011-11-28	29469
4A5953		2011-11-27	89978
0751C9		2011-12-01	32451
26BC15	CAP	2012-02-14	32877
328D38	CAP	2012-02-14	29469
CC1417	CAP	2012-02-14	09112

sat_mnt_accountbalance

HashKey_Account	LoadDate	Hash Digest	Balance
0751C9	2011-12-01	4D5A37	$55
51DB27	2011-12-01	A0D27F	$160
4A5953	2011-12-01	4E6D71	$130

sat_cap_accountdetails

HashKey_Account	LoadDate	Hash Digest	Balance
26BC15	2012-02-14	4D5A37	$1000
328D38	2012-02-14	A0D27F	$200
CC1417	2012-02-14	4E6D71	$350

Montague Systems

HashKey_Account	Bkey Collision Code	LoadDate	Hash Digest	Account No	Balance
0751C9		2012-02-14	4D5A37	32451	$55
51DB27		2012-02-14	A0D27F	29469	$160
4A5953		2012-02-14	4E6D71	89978	$130

Capulet Systems

HashKey_Account	Bkey Collision Code	LoadDate	Hash Digest	Account No	Balance
26BC15	CAP	2012-02-14	4D5A37	32877	$1000
328D38	CAP	2012-02-14	A0D27F	29469	$200
CC1417	CAP	2012-02-14	4E6D71	09112	$350

Figure 0-4 Applying Business Key Collision Codes to model

How are the new hash keys generated? ***SHA1(BKCC, '||', <business-key(s)>)*** (*apply delimiter between columns that make up a composite business key*)

As you can see it is now ***not*** possible to join Capulet Corp-sourced account details to account details sourced from Montague Systems because the business key collision code is ***included*** in the hash key generation.
Now that the surrogate hash keys have been generated; when you query the data vault you will only return the records in the hub and its satellite table for the content you are after when utilizing the ***EQUI-JOIN/INNER-JOIN***. Equally when utilizing inner joins between a hub and its links, a link and its link-satellites the content returned in the query will only be applicable to the that business entity.

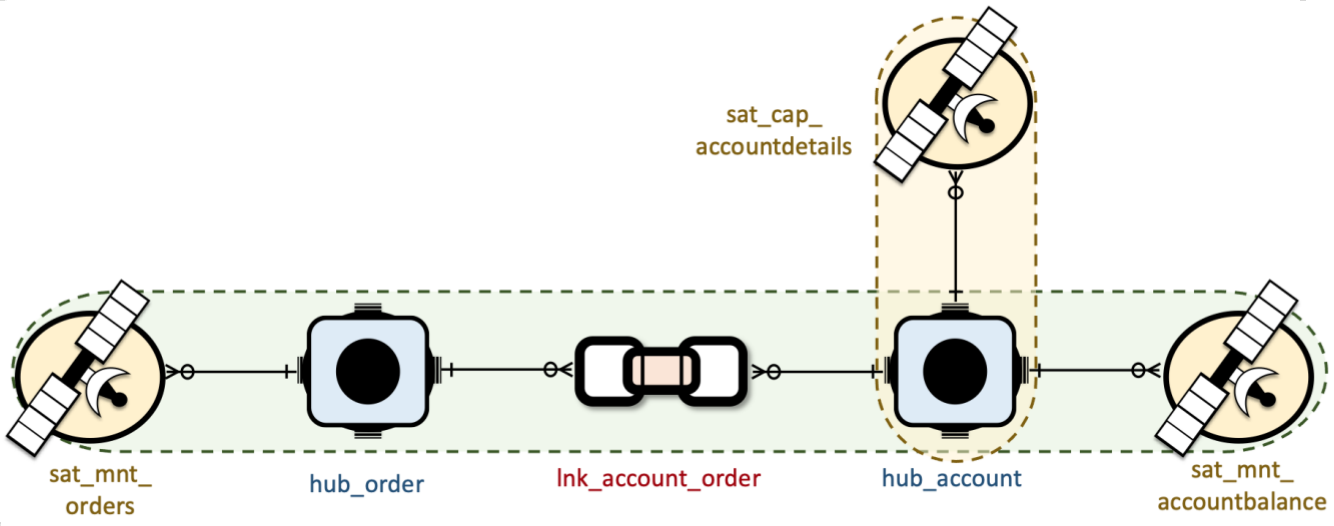


Figure 0-5 Integrated Data Vault

Therefore when we describe a business entity you will return only the descriptive content and actual relationships applicable to the business entities you are querying using EQUI-JOINs. Joining to non-applicable satellite tables to a business entity will ***not*** return any records, and that is ***ideal by design*** and why a single data vault model ***will integrate multiple business processes and data sources***…

… in Data Vault… "A Rose is a Rose is a Rose" - Gertrude Stein