Data Vault Dream Team

I have written a lot about the data vault vision, ideas, and patterns and how to deliver a data vault model *and* automation. When you're engaged at a customer whose data and analytics teams have decided to adopt Data Vault you will encounter those on the fringes that are incognizant of data vault, their sceptics, naysayers and sometimes even the obstinate!
But if you're in then you *must* have a **project champion**, a believer and only through sustained value delivery that you might just be able to turn those naysayers around! And of course, help deliver on the champion's promise!
Here is a collection of puzzle pieces that for us, helped drive the culture and understanding of what a data platform centred around Data Vault will deliver.

In no particular order

Capture the imagination….

Figure 0-1 Mona Lisa – Leonardo da Vinci

Humans are visual creatures and if you can tap into a person's imagination then adopting a new idea is less intimidating. I have used that concept to talk about cognitive load and extend it by comparing it how you would learn how to make beer. In the book I use a similar analogy to describe the three key elements needed to learn photography (ISO/CMOS chip

sensitivity, aperture and shutter speed). My former colleague captured the imagination by asking the audience, *"what are the three main elements needed to make music?"*
They were

- Rhythm – *"pattern of sound, silence, and emphasis in a song"*
- Melody – "*the aesthetic product of a given succession of pitches in musical time"*
- Harmony – "the sound of two or more notes heard simultaneously"

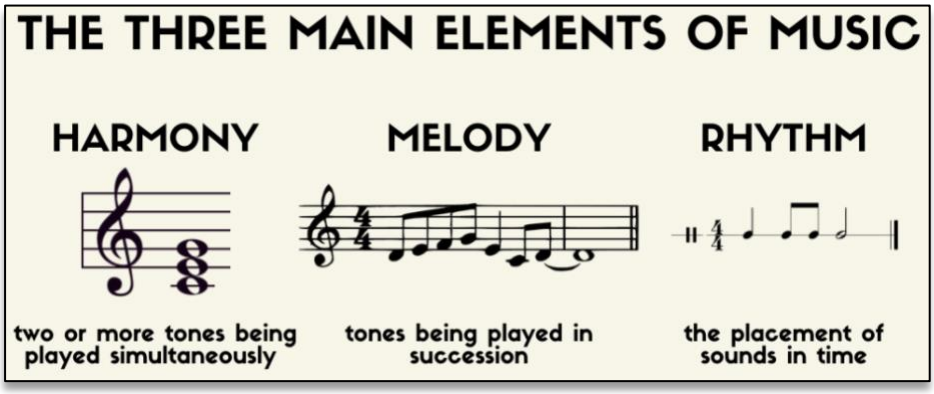

Figure 0-2 Cognitive load and three key elements

Of course, for a Data Vault *model* the same simplicity is described as hubs, links and satellites which holds true for the rest of the elements of data vault. *Like so…*

Variations of satellite tables:

- hub-satellite – parent key is a hub hash key
- link-satellite – parent key is a link hash key
- a satellite with a dependent child key – any of the above but with sub-typing
- a multi-active satellite – any of the above but of SETs
- a status tracking satellite – status of business objects and relationships
- a record tracking satellite – tracking business objects and relationships
- an effectivity satellite – active relationship in a link table
- an extended record tracking satellite – *fighting time crime*
- a non-historised satellite – immutable change records

Variations of link tables:

- a same-as link – two or more keys representing the same business object
- a hierarchy link – parent child relationship depicted within the same business object definition
- a link with a dependent child key – any of the above but with sub-typing
- a non-historised link – immutable change records

These are still simple variations of the three key data vault model elements! Behind modelling these patterns there are two **staging** patterns and four **loading** patterns,
Two staging patterns…

- Regular — supporting hubs, links, satellites, and multi-active satellites
- Inferred – supporting status tracking and effectivity satellites

Four loading patterns…

- Hubs & links follow the same loading pattern
- Satellites are very similar to hubs & links except it tracks changes to the **current** satellite record

- Multi-active satellites are like satellite loading patterns but tracks the SET of record changes
- Extended Record Tracking that if applied impacts all the above loading patterns

And finally, two query assistance patterns
- Point-in-Time (PIT) and
- Bridge tables

Define the initial Business Case

Figure 0-1 Liberty Leading the People - Eugène Delacroix

Businesses likely have a very different view of what they may define as a "Data Vault"; perhaps it's the equivalent of a *bank vault* room protected by cameras and other security technology to keep its contents (data) safe!
To adopt or not adopt a pattern as a part of a data strategy it is up to the data strategy team to secure funding for it, it needs a champion and a *champion* business case. It's not to say that business is *not involved* in a data vault, they must still have presence, be interviewed on how to define the **Business Ontology** if an ontology does not exist or is poorly documented. In a well-defined Data Management enterprise such business definitions are captured in Business Glossaries (Information Maps), and they can easily translate to Data Vault Hub Tables. This ensures that there is a common *business vocabulary*, for example, a facility in finance is quite different to a facility in construction! If you do not have a well-defined Business Ontology, then build one. You don't have to have all of it defined upfront before starting on your Data Vault project, but at least those Business Objects required for your Business Case, likely these will be your most active business objects, the ones involved in the most business processes *(ex. Hub_Account, Hub_Party…)*.
Identify
- business object definition, synonyms, and homonyms
- its grain and privacy requirements
- how to identify it uniquely, its business key (or keys) – *these must mean something to the business*
- its involvement in *value streams* and *business processes*
- products and services that influence or are used with this business object
- information *taxonomy* and *states*
- the inputs and outputs of a business object at a value stream stage
- enterprise and data policies affecting the data.

See: bit.ly/3fUL7fN

Build or Buy

Figure 0-1 Creation of a Man - Michelangelo

You've decided to build a Data Vault, and you want to automate the loading and modelling patterns as Data Vault defines them, *everything is a repeatable pattern!* What is the next step? *Build or Buy the automation…*

If you buy it…

- You will inherit *years of best practices* and a dedicated tool with *support levels* for bug fixes and features
- You will accelerate your deployment of data vault and reduce the time to delivery
- You will be beholden to the vendor's prioritization of fixes and bugs and therefore you may have to wait for a solution
- You may be subject to *vendor lock in* and their *interpretation* of Data Vault
- You may have to install thick clients or perform installation of other software that must pass your org's security clearances
- You are subject to a *license fee*
- You will need to ensure the software aligns to your architecture patterns and suites the data modeller's preferences, do they prefer scripting, or do they prefer working on a canvas?
- You will need to consider what you get with the whole package, does it come with an automated test framework? Dashboarding? Modelling tool (do you need to purchase a separate tool or licensing)?
- You will need to encourage your user base to adopt it and get vendor training on it.
- Does the tool cover all patterns for Data Vault? Does it have more than you need?

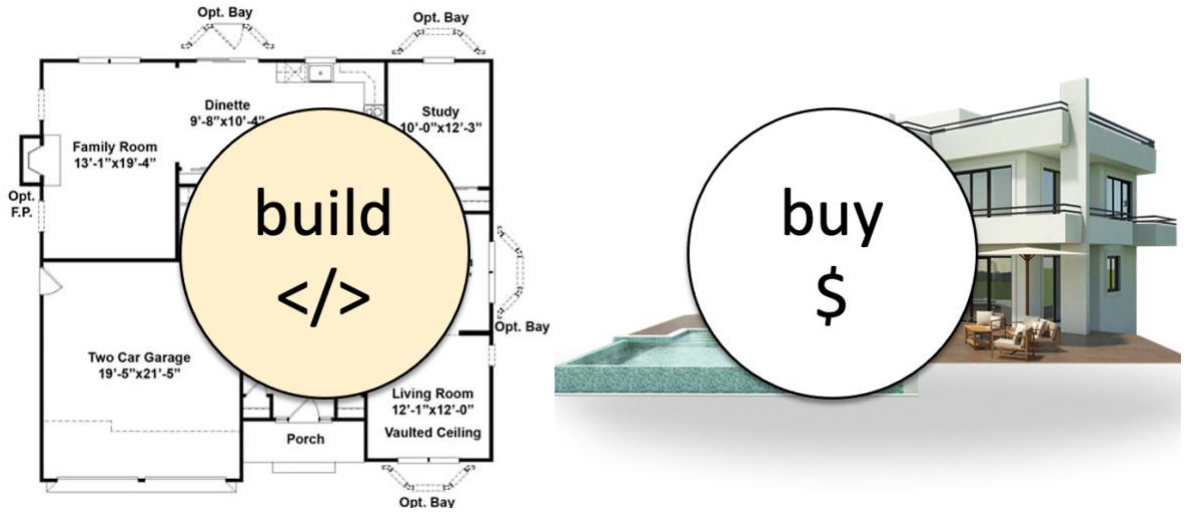

Figure 0-2 Build or Buy

If you build it...

- You must understand Data Vault *inside and out* to understand *all the variations* of all the *patterns* that you need, you might not need all the artefacts, but the core patterns you will
- You will have to wait before delivering a Data Vault because the time to build an automation tool *delays the delivery* of the Data Vault model
- You must secure funding for something that will only *eventually* show ROI
- Your *bugs and features* can be prioritised to *your own schedule and backlog*
- You can build the Data Vault model to your own *interpretation* of Data Vault
- You can architect the tool to your *architecture style*
- You will be building a tool that suits your analytics culture
- You may have to rely on open (but secure) source for libraries and modules needed for your overall delivery
- You don't pay a license fee and you can build out all your own test automation patterns
- You get to evolve the tool to your direction and imperatives
- You *could* even have a tool that can eventually be sold commercially

If you build, define what each target table must looks like, it's loading patterns and included metadata tag columns. Define *both* the positive and negative scenarios (where the tool is expected to pass and where it is expected to fail). Express it features, usage scenarios and scalability through demos and documentation. And *don't* get involved in the model outcomes, in fact data modellers should be your source of requirements for automation patterns to build and define the tool, make sure they're DV-certified and have some years of DV modelling under their belt!

What is the Steel Thread?

Figure 0-1 The Old Guitarist - Pablo Picasso

(Set expectations)

No one wants to be in a project with a *perpetually* moving end goal, nor does one *want* to be associated with a project that does not deliver business value! Accept that *scope creep* is a part of delivery but understand that expectations must be set up front, and deviations are owned by the business, the ones paying for the initiative, but managed by data delivery. Setting expectations upfront avoids (or at least limits) the dreaded *Blame Game*, besides, finger pointing is rude!
To the Data Vault, what does a *Steel Thread* look like?

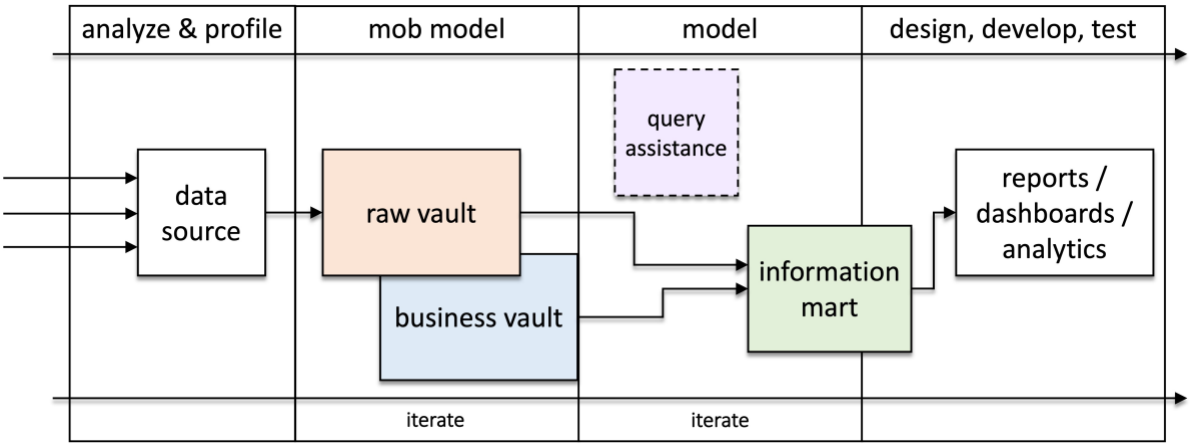

Figure 0-2 Steel Thread

Initialise – overall setup, most defined before first solution spring

Roles? Enterprise Data Architect, Data Steward, Data Modeller
What?

- Tenants on the overall Enterprise Data Model development
- Guidance on subject areas, domains
- Establish registers - hub, decision, source-badge, (possible) collision code
- Incorporate naming standards for tables, data. vault tag columns and business vault attributes. Information Mart can be more suited to information consumption requirements
- Business rule management through tooling, governance that includes raw and business vault
- Automation, tooling - build or buy *(see above notes)*
- Platform technology guidance *(see below notes)*

- Data quality – technical and business rule performance
- Approach to confidential and sensitive data, policies, agents
- Reference data management and management
- Outcome delivery options – views, marts, extracts, pre-canned reports, self-serve
- Model review board – model scoring, lessons learnt, decisions registered, alignment to enterprise data management principles
- Data retention policies, archiving
- High-level operating models of the platform, deal with failures, alerts, out-of-sequence data

Planning

Roles? Product owner, Technical Business Analyst, Agile coach, Solution Architect, Data Modeller, Source-System Subject Matter Expert (SSSME)
What?

- Scope requirements, sprint team(s)
- Identify stakeholder roles
- Identify source(s) / business process
- Pick 1-5 source files, ingestion patterns (snapshot, batch, increment, *change-data-capture (CDC)*, near-realtime, streaming
- Pre-mob homework – data profiling, identifying business objects, units of work, critical data elements, confidential data classification, grain
- Articulate outcome needed and in what form; data served or self-served, in what functional form for consumption, what works for the type of analyst or scientist
- Test patterns, business rule testing
- History / backfill requirements

Mob Modelling

Roles? SSME, Data Modeller, Solution Architect, Technical Business Analyst
What?

- Identify model (questionnaire guide)
- 1–2-hour Mob Modelling workshops for 1-5 source files
- Identify gaps – push to source or persist outcomes to business vault (temporarily or permanently)
- Enterprise or private business vault requirements
- Information mart outcome

Deployment

Roles? Data Engineer, Solution Architect, Data Modeller, Technical Business Analyst, Tester
What?

- Configure model
- Automate data flows
- Test automation & additional data quality testing

Post Deployment Review

Roles? Enterprise Data Architect, Solution Architect, Data Modeller, Data Engineer

Platform Engineers
What?

- Demo to product owner / business owner / champion
- Record Technical Debt, certify documentation, review scorecards and solution

The first few sprints are used to establish the thread, it becomes a Steel Thread when it becomes a repeatable pattern and speed bumps are ironed out; subsequent analytics requirements consume less sprints to deliver because of the established Steel Thread.
The Steel Thread should become unbreakable but malleable

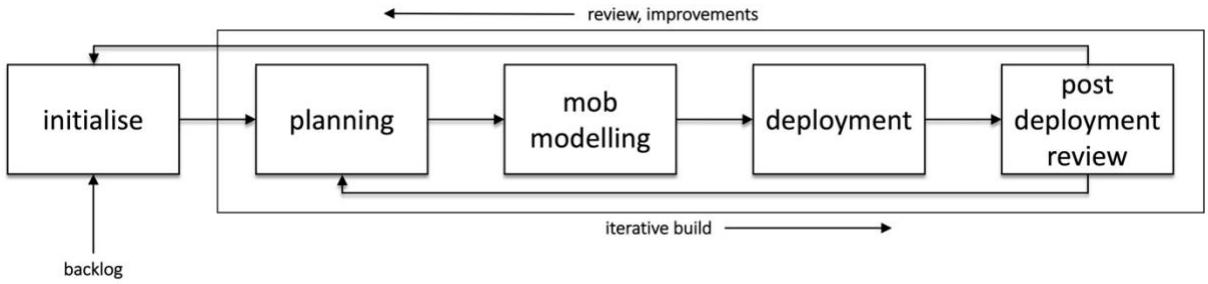

Figure 0-3 Work on the Steel Thread

See: bit.ly/2WEWCSw

Adopt the Standards

Figure 0-1 The Persistence of Memory - Salvador Dalí

Taking on something new takes time to learn, ideally when those new to Data Vault are most engaged are when their imaginations can be tapped within the building blocks of what makes a data vault. Take the example of Lego, once you have followed the in-box blueprint of what the outcome of the Lego set should look like do you not take that apart and see what else you could build from those same parts?

Training

To understand the building blocks of the standards training is recommended, ideally if you have already dabbled in Data Vault for a bit. Armed with the questions you need answered *then* attend a certification course through an authorised training partner. There you will

have a dedicated authorised trainer to teach you the material and answer your pertinent questions.
Where to get training? See, bit.ly/39o9qPX

Hackathon

As an extension to training why not include a Data Vault modelling *hackathon*? That is, select a business case as identified above, one with nuanced oddities that would best be served having an authorised trainer on hand and by the end of the day produce a draft model ready for implementation. *Try to include as many tricky scenarios as you can adopt!*

Data Vault Coach

Finally, the third suggestion is to onboard a *Data Vault Coach* for a period of two to six months initially. With a DV coach on hand there is no rush to have as many scenarios as possible for the DV coach to glean over. Instead, the DV coach can be engrossed in your data culture, drawing scenarios, helping you make the right decision and have tasks assigned to him/her. Eventually those being coached will have to assume all responsibility unassisted.

Remember, all businesses differ, although there are patterns based on experience be prepared to *fail fast, fail early, and fail cheaply*.

Understand that *Data Engineers are not necessarily Data Modellers*. Data Engineers should not be making modelling decisions, Data Modellers should not be making engineering decisions.
This is especially true around data vault, an engineer may question why link tables do not have effectivity columns… think about it, if you did that then you have made the link table tied to a single requirement and the link table itself no longer scales. Besides, what exactly are you deriving the start and end dates based on in a link table? Another potentially *horrendous* engineering decision is to *not* apply the standard business key treatments, this means *Passive Integration* is not achievable and no, business key treatments are not the same thing as mastering business keys! Another example of a potentially disastrous engineering decision is to (by default) use a source-system identifier as the business key collision code.
Please let the modellers make modelling decisions!

Build a Corporate DVBoK

Figure 0-1 The Anatomy Lesson of Dr. Nicolaes Tulp - Rembrandt

(Corporate Data Vault Body of Knowledge)
A referenceable body of knowledge written by the analytics teams building the enterprise data vault that suites the culture and style of the enterprise. The DVBoK should include, but not limited to.

- An attractive front page, include certified data modellers, a RACI of various roles and responsibilities and domain owners. From the front page I should be able to find everything I need to know about Data Vault and implementation. From the front page I should be able to access (represented as tiles):
- Reference the Data Vault modelling patterns, hubs, links and satellites and the variations listed above and how they are loaded
- Naming standards of tables and columns and standard data vault metadata tags expected to be visible in each data vault table
- Modelling flowchart or decision tree with exception patterns, and a flowchart for a migration path from onboarding source systems onto Data Vault
- Nuanced Business Vault implementations, business problem solved and sample scripts
- Example modelling techniques and why they were adopted
- Consumption patterns and the use of querying techniques to improve query performance. Such as SQL Window functions, query assistance tables and how to use them
- Templates for the automated test suite and other data quality checks
- Decision and hub registers and owners of these registers. The hub register includes what business key collision code was used for the format of business key
- Source system register and where the core of the data comes from, including what source-system badge to designate to each source that may or may not be used as a collision code depending on the business key profile
- Describing the Steel Thread to be used as a template for information delivery, the optional outputs
- How to build and promote derived business rules, promotion process, a checklist
- Model scorecards and agile story templates
- Derived business rule patterns and what tools are being used to deploy them
- Data profiling questionnaire to be used to interrogate new source data

Pick a theme that makes the various deployments unique and easily referenceable by a noun, the list of nouns used should be easily relatable and socialised so everyone in the analytics team can relate to it. Some ideas:

- Names of celestial bodies in the universe
- Character names from the Game of Thrones, the Marvel Universe or even Pokémon!
- Particles from the Standard Model
- Birds of paradise, native birds, birds of prey, especially those with interesting rituals
- *More…?*

Depends on how geeky you want to get!

"There are only two hard things in Computer Science: cache invalidation and naming things" -Phil Karlton

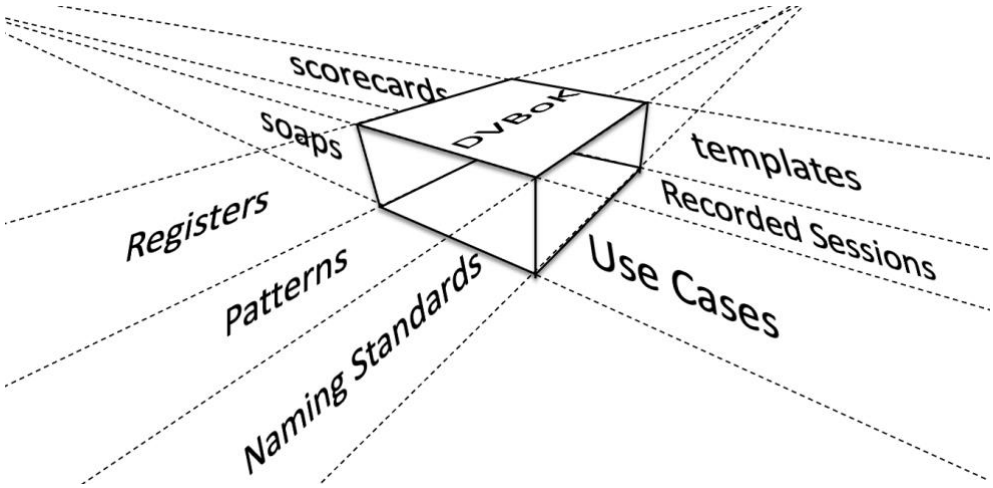

Figure 0-2 DVBoK, centre of corporate knowledge for Data Vault

When deploying these artefacts have them peer reviewed not only for spelling and grammar but for common understanding. A document is like a book, if the contents must be verbally explained by the author to the reader, then it is not a complete self-describing document. But don't repeat content that has been written elsewhere, set references to it and pick a universal documentation style.

The style of writing I like to use is as if I am taking the reader through a tutorial of the content.

Again, humans are such visual creatures so include diagrams and illustrations where you can describe the problem and solution. Also look to adopt iconography as tiles on the front page, these can be used to develop mental associations with the type of content and where they live. Adopt the corporate colour schema and ensure that the documentation standards remain consistent throughout the wiki.

Establish a DV-COP

Figure 0-1 Nighthawks - Edward Hopper

(Data Vault Community of Practice)
An open collaborative platform is encouraged! Setup a practice to:

- On a periodic basis meet up (weekly initially perhaps), onsite or offsite as you see fit
- Perform presentations of modelling problems, solution on a page (SOAP), modelling solutions
- Discuss model review scorecards, registries for decision made, pending data sources, business vault outcomes, information mart requirements, best practices, onboarding new tools (if needed)
- Discuss lessons learnt, what went well, what could be done better
- Invite guest speakers to the COP, it doesn't strictly have to just be about Data Vault! It could even be alternative methods that may be of interest!
- Set up a community page, link it to your DVBoK, and a workplace channel for internal discussions.
- Make use of appropriate visual tools for collaboration, whiteboards, or online collaboration tools like Miro
- Set up a newsletter or be a part of an existing corporate newsletter highlighting achievements, performance against a burn-down chart for outcomes and milestones the initiatives seek to achieve
- Share ideas / concepts encountered in the industry, key learnings, have viewing parties for live events from industry leaders, especially on what is trending!
- Tools (bought or built) that can be used to enhance the overall analytics experience, knowledge graphs, data governance tools
- Adopt agile games to change the pace at times, make DevOps and DataOps a part of discussions and delivery. Experiment with agile team structures and what works best with the size and pace of your organization. DV-CoP should not impede agile but rather augment it.
- Form a reading group for researching methods and techniques that *could* be adopted. These could be based on open papers or other published materials released by through leaders

Research the Technology Stack

Figure 0-1 The Starry Night - Vincent van Gogh

*You just might **not** know everything, and that's ok!*
From traditional RDBMS to NoSQL each have their nuances, and each have their intended purpose. Does it suite a Data Vault, can you deploy a Data Vault on it? What do you sacrifice by adopting a particular technology stack? Is it worth the effort?
Let's explore three examples:

Apache Cassandra

Apache Cassandra can deliver mind-blowing performance by supporting a dashboard with a 176 million record read under 5 seconds! Sounds great, *let's build a Data Vault on it!*
Cassandra is a distributed computing platform that conforms to the principles around eventual consistency (CAP theorem), in the event of a *network partition* you must choose between either *high availability* or *data consistency* between participating masterless nodes. A node contains a replica of data and the nearest node to a client serves the data to that client. Consistency between nodes is maintained using a consensus protocol and for Cassandra this is Paxos. The speed of the queries is further helped by storing the data in MEMTables that are eventually persisted into SSTables (following some in-memory compaction) which are immutable.

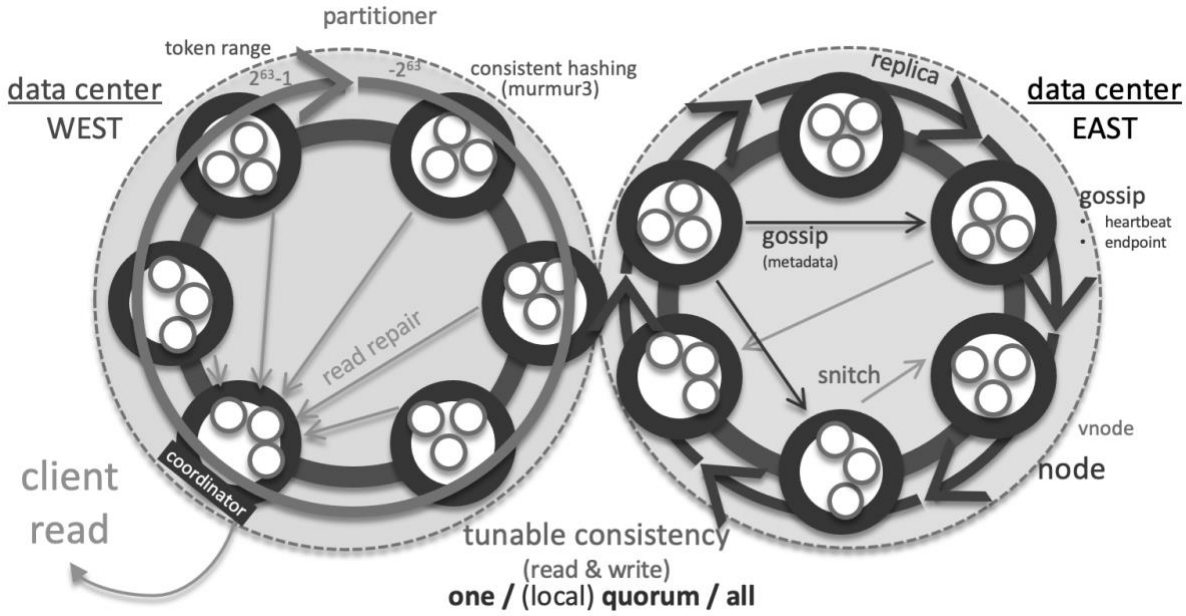

Figure 0-2 Cassandra architecture

The table structure itself is called a "column family" and querying the data is using Cassandra-Query-Language (CQL). The catch is however, CQL does not allow for joins

between tables and as we know Data Vault has many tables that requires joins. In fact, the paradigm for querying data you want from Cassandra is the opposite to traditional RDBMs, whereas on an RDBMs to get the analytics you want, joins are essentially "Joins-on-Read" on Cassandra the paradigm is "Join-on-Write".
Let's explain, another reason for blazingly fast analytics on Cassandra is that every query you could possibly think of must be designed up front and deployed as independent tables that must agree with each other's content. The approach to modelling is more aligned to *application workflow* rather than building a *relational model*, the process is captured nicely in a Chebotko diagram.

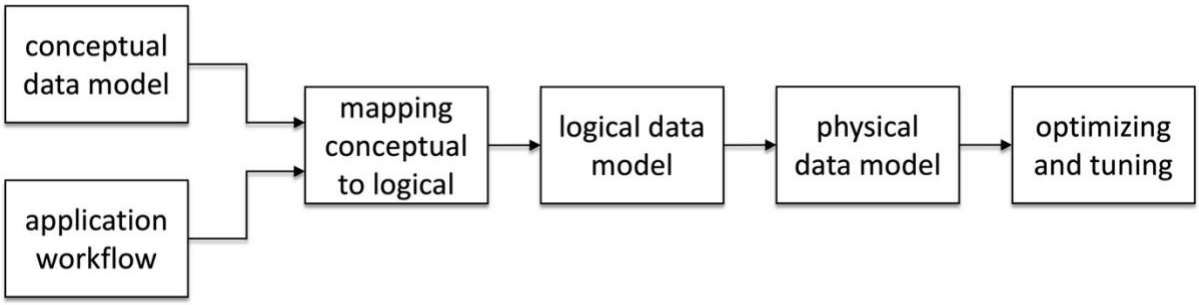

Figure 0-3 Cassandra modelling flow

Back to these independent tables, for each table you defined a primary (partition, composite, and clustering) key, any updates to that record by primary key marks the updated record with a "tombstone" meaning that the old version of the record is no longer accessible, the new version is. When MEMTables get full and flushed to SSTable (solid state tables on disk) compaction discards records with a Tombstone, *bye bye audit*.
There are workarounds to querying for sure, you could use secondary indexes, but these are for low cardinality columns that forces the query to do a full cluster scan (anti-pattern). You could deploy materialised views, but these depend on the rate of updates to the cluster nodes. You could also rely on Apache Solr for Lucene trie hierarchies that performs auto-completion of search text (the same search completion you see on the Google search bar). Of course, you could just throw Apache Spark at it to perform SQL joins, but as is documented by Apache Spark's global issue tracker, it has limitations (just like its slower cousin HiveSQL).
What was the outcome? No Data Vault on Apache Cassandra, the technology stack does not suite a Data Vault and not worth the effort to force it in.

Apache Hive and Spark SQL

Logically it can be done, but measure up the effort, limitations, and cost of relying on a NoSQL platform. Like the Cassandra option above the use of Spark SQL or Hive SQL has its limitations, and using parquet has several additional limitations, it's slow. Parquet organizes data into evolving partition snapshots, for the date you need to query for you design the parquet file to include the load date in the partition key. So, if you need to query the data for a particular date (like today) then you must include the date in all partitions you are including the join. For a Data Vault this could be a problem.
A Data Vault will load new records to hubs, links and satellites meaning that for each day the data loaded each day to parquet amounts to only a few records. Therefore, if a query is executed that needs to retrieve historical data at a point in time the query must scan ALL partitions of a parquet table to find the relevant record. Couple that with the need to join

(sometimes) multiple hub, link, and satellite tables in a single query all of which must perform full partition scans along the way. These are defined as external tables and therefore no indexing can be applied either. Periodically, you must perform parquet re-partitioning to improve query performance and table repairs to keep the table metadata up to date.
SparkSQL does provide some Window/Analytical functions, but the language is *not* intended for full scale analytics. For example, you will find a date_add function but no time_add function. Although the time increment operation is possible, it requires more effort (much like the rest of Spark).
And then you have to resort to this… bit.ly/3EQ9wO3
Phew! What was the outcome? Build it on Hive SQL, establish the framework to later deploy it on Snowflake

Snowflake

Much of my blogging revolves around Data Vault on Snowflake, see below

- Read "Data Vault 2.0 on Snowflake…To hash or not to hash… that is the question", - bit.ly/3dn83n8
- Read "Why EQUIJOINS Matter!", - bit.ly/3dBxOQK
- Read "Data Vault PIT Flow Manifold", - bit.ly/3iEkBJC
- Read "Data Vault's XTS pattern on Snowflake", - bit.ly/3aCCRhQ
- Read "Data Vault Agility on Snowflake", - bit.ly/337Jhp3

Think like a Consultant

Figure 0-1 The Son of Man - René Magritte

Steve Jobs famously quipped, "*...without owning something over an extended period of time, like a few years, where one has a chance to take responsibility for one's recommendations, where one has to see through all action stages and accumulate 'scar*

tissues' for the mistakes and pick oneself up off the ground and dust oneself off, one learns a fraction of what one can…"

You are on the outside looking in, ultimately your reputation as a consultant rest on your delivery, *one day you will have to handover the keys* so adopt the mindset that your customer's success is your success! The customer knows the organizational climate better than you do, and they will know their area of expertise better than you (likely). They are hiring you the consultant on how to adopt that knowledge into a Data Vault. Look to learn from the customer as the customer will look to you for your expertise.
A few bullet points on this.

- Praise the customer when things go well, work with the customer when things don't
- Advise but not direct; they're paying for your advice but not for another manager they would need to report to
- Your advisory is advice, don't take it personally if they do not adopt it, ultimately, they will have to live with the potential technical debt if they choose an anti-pattern but do your best to inform the customer as to why you think it is an anti-pattern
- Follow the guidance above, not only delve into their technology stack but research their business and industry as well as it relates to the business case, they have selected for Data Vault adoption. This may include a few sessions to understand some nuances of their business and of course some *homework* by you.
- Ensure you have a mutual success plan.
- *Prepare to (eventually) hand over the keys…*

For more sage advice see, bit.ly/34xAnkW

Now let's tap into your creative brain by looking at organization metaphors as described by Dr Gareth Morgan

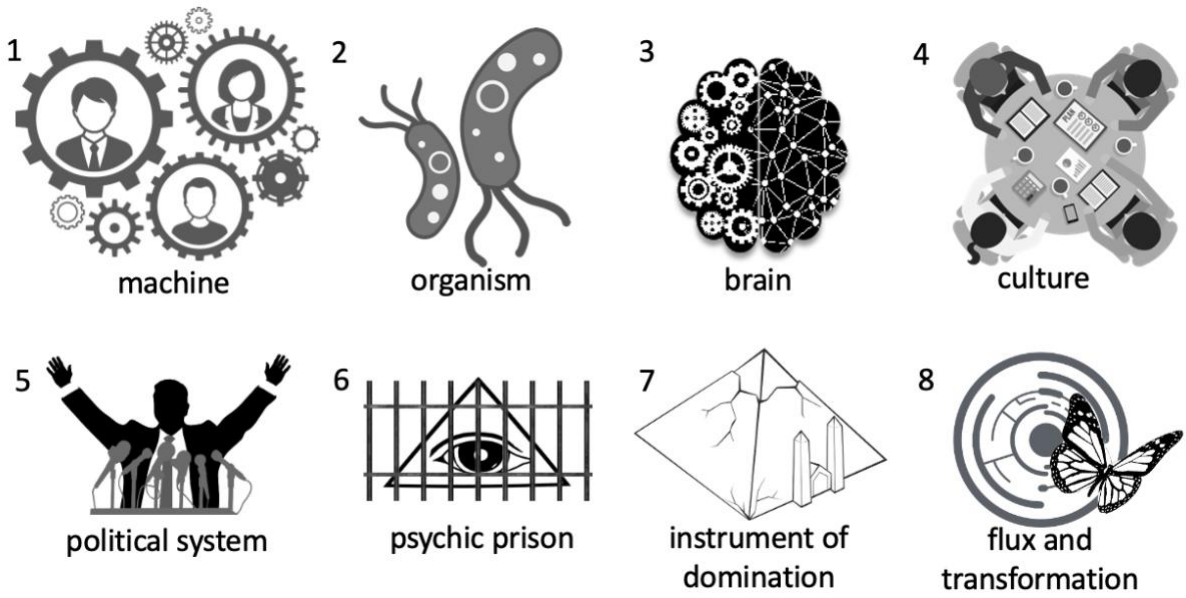

Figure 0-2 Images of Organization, Dr Gareth Morgan, for a deeper dive into this see: bit.ly/3gIrubr

1. **Machine** – mechanical organization, this organization is a series of connected parts arranged in a logical order to produce a repeatable output. This metaphor relates to the Bureaucratic organization

2. **Organism** – inspired by contingency theory, are open organizations that believe there is no best method to managing organizations and are malleable and must adopt as the environment changes
3. **Brain** - believes that an organization is a set of functions designed to process information and learn over time and encompasses learning theories and cybernetics, *ala double-loop learning*
4. **Culture** - emphasizes symbolic and informal aspects of organizations as well as the creation of shared meanings among actors, socially constructed realities, and *espoused values*
5. **Political system** - an organization is thought of as a game of gaining, influencing, and coordinating *power*, recognises the important role that power play, competing interests and conflict have in organisational life. Morgan describes this as the realm of *sociopaths*
6. **Psychic prison** - an organization is a collection of myths and stories that restrict people's thoughts, ideas, and actions and conflict is avoided – *ala groupthink*, see: bit.ly/3J3JLfi. The psychic prison relies on domination.
7. **Instrument of domination** – "the ugly face", the organization is a means to impose one's will on others and exploit resources for personal gains. The organization is class based, *ex. "Death of a Salesman".*
8. **Flux and transformation** - an organization is an ever-changing system indivisible from its environment, Greek philosopher Heraclitus noted that, *"you cannot step into the same river, for other waters are continually flowing on."*

Dr Morgan acknowledged that this is not a exhaustive list!

To perform as a dream team, bring an *enthusiasm for learning, empathy, humility, honesty, humour, creativity, and respect* wherever you go. You might find that what you do for a living cease to be an obligation but simply becomes an extension of your creativity and passion.

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.