Data Vault und Ladeperformance

Markus Kollas
CGI Deutschland Ltd. & Co. KG
Sulzbach (Taunus)

Schlüsselworte
Data Vault, Beladen und Entladen, Data Warehouse, Core Warehouse, Data Marts, Sternschema, Hash-Keys

Einleitung

Data Vault entwickelt sich auch in Europa mit rasanten Schritten zur meist präferierten Datenmodellierungsmethode für ein Enterprise Datawarehouse. Hier wird kurz aufgezeigt, warum in Data Vault streng nach fachlichen Datentypen modelliert wird, um dann zu beleuchten, welche Vorteile sich durch diese Modellierungsmethode für die Beladung und die Beladezeiten ergeben. Auch die Bedeutung einer Trennung zwischen Raw Data Vault und Business Data Vault wird dargestellt. Erweiternd wird dann noch auf die Unterschiede bei der Beladung zwischen Data Vault Version 1 (Surrogate ID) und Data Vault Version 2 (Hash-Keys) eingegangen. Ein kurzer Blick in die einfache Erstellung von Sternschemata für Data Marts aus dem Data Vault heraus rundet das Thema ab.

Data Vault Basics


Dan Linstedt

Entwickelt von Dan Linstedt in den 90er Jahren hat Data Vault seit einiger Zeit auch Europa in seinen Griff genommen. Dabei setzt Data Vault prinzipiell auf eine Trennung nach fachlichen Datentypen und clustered diese in drei Gruppen: Business Keys, Details und Assoziationen. Eine konsequent und möglichst exakt durchgeführte Aufteilung seiner Daten in diese Gruppen führt erfahrungsgemäß auch zu einer deutlich vereinfachten Kommunikation über diese Daten zwischen Fachbereichen und IT. Dabei wird jeder Datentyp in einer oder mehrere eigens für ihn angepassten Tabelle gespeichert. Diese Anpassung wird für jedes Datum eines Typs meist in der gleichen Weise getätigt – Ausnahmen bestätigen die Regel.

Diese Tabellen werden wie folgt benannt:
Hubs beinhalten die Business Keys
Satelliten beinhalten die Details (von Hubs und Links) und
Links fügen die Assoziationen (Relationen) zwischen Hubs sprich Entitäten hinzu.

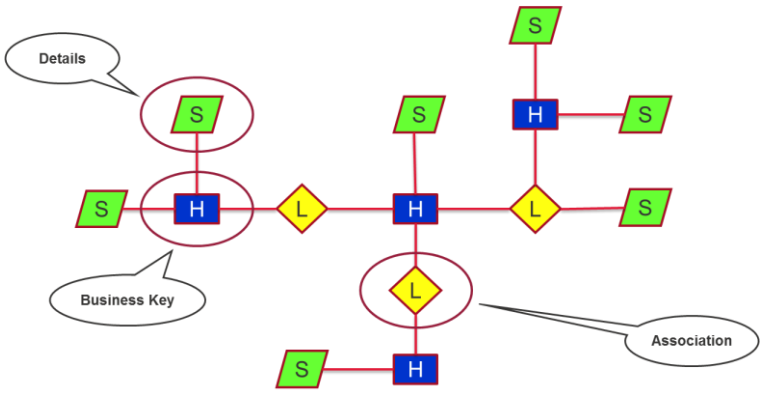


Abb. 1:Beispiel einer Data Vault Struktur

Diese strenge Trennung in Datentypen führt gleichsam zu folgenden vier gravierenden Vorteilen:

- Jede Entität aus Hub und Satellit(en) ist quasi isoliert und kann für sich betrachtet und verwaltet werden, ohne dass es zu einem Einfluss auf andere Entitäten kommt.
- Änderungen in den Quellen der Daten haben (oft) keinen und eben nur geringen Einfluss auf die Data Vault Struktur

- Alle Entitäten sind untereinander entkoppelt und über Links verbunden, was dazu führt, dass das Data Vault Modell sehr leicht und einfach erweitert werden kann.
- Ladeprozeduren können uniform gestaltet werden und weisen somit einen hohen Wert an Wiederverwendbarkeit auf und eignen sich dazu per Generator erzeugt zu werden (siehe hierzu auch unseren Artikel im BI Spektrum Ausgabe 2 2015).

Data Vault – Tabellenstrukturen

In Data Vault sieht nicht nur die gesamte Modellstruktur sehr gleichförmig auch, sondern auch die Tabellen eines Datentyps gleichen sich in ihrer Struktur. So beinhaltet zum Beispiel ein Hub neben dem eigentlichen Business nur drei weitere Felder: Eine SID (Data Vault 1.0) oder einen Hashkey (Data Vault 2.0), sowie eine Ladezeitstempel und eine Angabe über die Herkunft des Business Keys. Der Business Key selbst ist schließlich das bestimmende Datum (notfalls auch ein aus mehreren Datenelementen zusammengesetzter Schlüssel), der jeden Datensatz einer Entität eindeutig identifiziert.

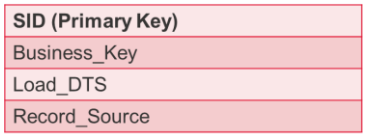


Abb. 2: Hub-Struktur DV 2.0

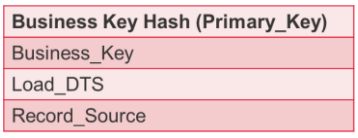


Abb. 3: Hub-Struktur DV 1.0

Auch bei den Link-Tabellen ist die Struktur recht überschaubar und derjenigen der Hubs sehr ähnlich. Neben den mindestens 2 Business-Keys enthalten auch Link-Tabellen entweder eine SID (Data Vault 1.0) oder einen Hashkey (Data Vault 2.0), sowie eine Ladezeitstempel und eine Angabe über die Herkunft der Relation. Der Hashkey in DV 2.0 wird dabei aus allen beteiligten Business-Keys gebildet, die erst zusammengesetzt werden, bevor der Hash aus einer MD5 oder SHA-1 Funktion gebildet wird.

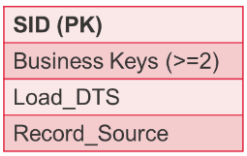


Abb. 4: Link-Struktur (DV 1.0)

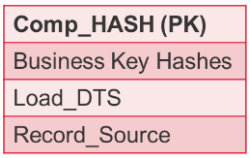


Abb. 5: Link-Struktur (DV 2.0)

Schließlich ist noch die Struktur innerhalb der Satelliten zu betrachten. Naturgemäß ergeben sich hier die größten Unterschiede zueinander aufgrund des Inhalts, aber auch hier finden sich wiederkehrende gleichförmige Elemente wie die SID (Data Vault 1.0) oder der Hashkey (Data Vault 2.0) aus dem jeweiligen Hub bzw. Link, sowie eine Ladezeitstempel und eine Angabe über die Herkunft der Relation. Optional ist ein Lade-End-Zeitstempel mit Hilfe dessen innerhalb einer Historisierung das Gültigkeitsende eines Datensatzes angezeigt werden kann. Für die Historisierung ist es ebenfalls notwendig den Primary Key einer Satellitentabelle um den Ladezeitstempel zu erweitern

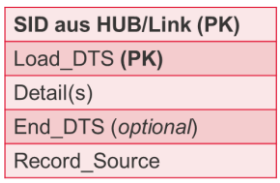


Abb. 6: Satellit in DV 1.0

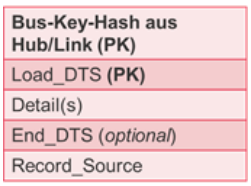


Abb. 7: Satellit in DV 2.0

Data Vault – Wo ist sein Platz?
Da Data Vault eine signifikante Erhöhung an Relationen aufweist, die über die Verbindungen Hub-Link, Hub-Satellit, Link-Satellit und eventuell sogar Link-Link abgedeckt werden, dürfte es auf den ersten Blick klar sein, dass hier keine große Performance für Reporting-Systeme zu erwarten ist. Ebenso scheiden alle transaktionalen Systeme aus, da hier die Lade- und Speichergeschwindigkeit meist noch relevanter ist, als bei den Reportingsystemen. Wo also findet Data Vault seine Existenzberechtigung?

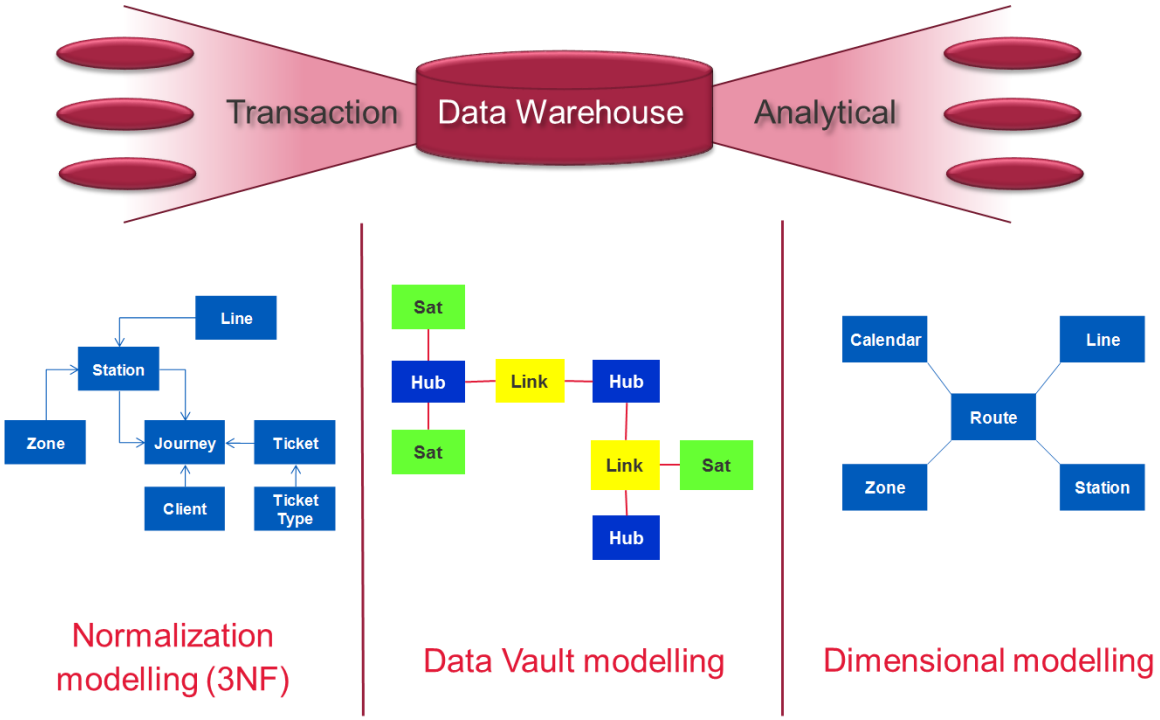


Abb. 8: Data Vault Verortung

Data Vault hat seine Heimat genau da, wo man bislang immer Kompromisse zwischen Performanz, Datenmenge (unter anderem auch durch Historisierung) und Verknüpfung von Datenentitäten eingehen musste: Das Core Data Warehouse. Während es bislang für transaktionale Systeme kaum eine alternative zu einer 3NF Modellierung gab und eben solches bei den analytischen Systeme von einem Sternschema bzw. Scheeflockenschema zu behaupten war, ist man bislang bei der Speicherung von großen Datenmengen in einem zentralen Warehouse meist einen Kompromiss eingegangen, der am besten mit dem Begriff „denormalisiertes 3NF“ zu beschreiben war. Eigentlich bereits ein Paradoxon in der Namensgebung. Hier bietet Data Vault nun deutlich Abhilfe.

Beladung eines Data Vault 1.0 Datenmodells
Bei Data Vault kann die Beladung der einzelnen Tabellen hochgradig parallelisiert werden. Wie in Abbildung 9 gezeigt, werden nach dem Beladen der Staging Area zuerst alle Hubs befüllt. Durch die Unabhängigkeit der Hubs untereinander können diese – nur begrenzt durch die Hard- und Software – alle parallel befüllt werden. Dabei werden die notwendigen SIDs erzeugt, die anschließend bei der ebenfalls parallel erfolgenden Beladung aller Hubsatelliten und Links als Lookups benötigt werden.

Nachdem dabei die SIDs der Links erzeugt wurden, werden nun deren Satelliten ebenfalls parallel beladen (mit Lookup der SID auf der Link-Tabelle).

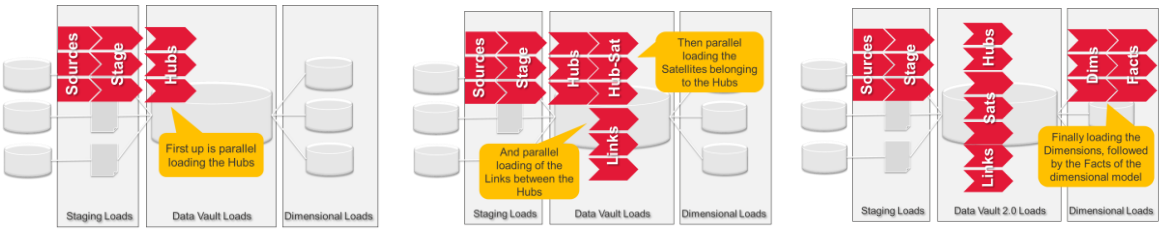


Abb. 9:Parallelität beim Beladen eines DV1.0 Modells

Beladung eines Data Vault 2.0 Datenmodells

Bei Data Vault 2.0 werden die allseits bekannten Surrogate IDs, die einen Datensatz eindeutig identifizieren durch Hash-Keys ersetzt. In diese Hash-Keys fließen die Business Keys einer Entität ein. Da die Berechnung über einen MD5 oder SHA-1 Algorithmus erfolgt, kann der Hashkey jederzeit erneut erzeugt werden und steht somit omnipräsent zur Datensatzsuche zur Verfügung. Dies erspart im Gegensatz zur Version 1.0 die Lookups in Hubs für Satelliten und Links. Aufgrund dessen ist die Parallelität beim Laden nochmals deutlich erhöht und im Vergleich zum Modell 1.0 fehlt ein kompletter Schritt, weil hier alle Hubs, Links und Satelliten parallel geladen können.

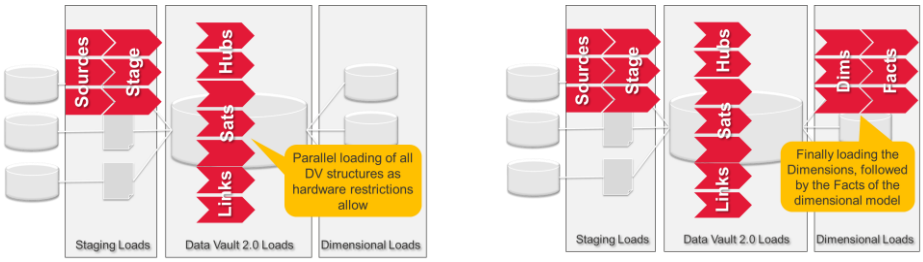


Abb. 10:Paralleles Laden bei Data Vault

Raw Data Vault vs. Business Data Vault

In einem Enterprise Data Warehouse mit einen zentralen Datenspeicher (Core-Warehouse) und mehreren themenbezogenen Data Marts mit unterschiedlichen Zielgruppen wird es immer wieder vorkommen, dass gewisse Indikatoren und Messgrößen mehrfach benötigt werden. Bereits an dem Punkt, an dem gleiche Berechnungen für unterschiedliche Abnehmer nur doppelt durchgeführt werden müssten, lohnt es sich, über eine Auslagerung dieser Berechnungen ins Core-Warehouse in eine sogenannte Business Schicht nachzudenken. Somit werden solche mehrfach verwendende Kenngrößen nur einmal berechnet und stehen allen Abnehmern durch ein einfaches Laden zur Verfügung. Data Vault bietet hier wieder einen deutlichen Vorteil, in dem man berechnete Kenngrößen als zusätzlich Satelliten an entsprechende Entitäten, sprich Hub, anhängen kann. Was nach außen wie eine eigene Schicht aussieht, ist technisch nur eine Erweiterung um eine oder mehrere Tabellen oder oft sogar nur Views. Somit kann die Berechnung dieser Kennzahlen bereits in den Ladevorgang

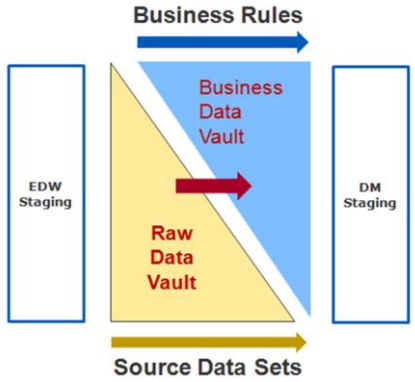


Abb. 11: Business Schicht im Core Warehouse unter Data Vault

integriert werden.

Das Erstellen eines Sternschemas aus Data Vault

Eine berechtigte Frage, die im Zusammenhang mit Data Vault immer wieder gestellt wird, ist die, wie man die Daten im Data Vault vernünftig und einfach in ein Sternschema überführen kann. Dies sie hier an einem Beispiel dargestellt. Gegeben ist folgendes Data Vault Datenmodell:

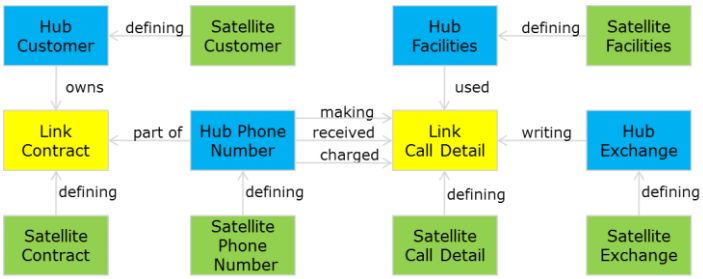


Abb. 12: Data Vault Beispiel

Das Beispiel ist aus der Telekombranche gewählt und beinhaltet folgende Vorgaben:

- Komplette Historie
- 100% versioniert und auditierbar
- Implizite Verwendung als MDM
- „Single point of Facts“

Um ein Sternschema daraus zu erzeugen, müssen als erstes die Dimensionen identifiziert werden. Sobald dies geschehen ist, können die Dimensionstabellen im Sternschema beladen werden. Dazu ist eventuell die zusätzliche Definition einer zeilenweisen (inhaltlichen) Einschränkung der Datensätze notwendig. Prinzipiell kann dies sogar automatisiert werden, da jeder Hub eine Dimension darstellen kann. Im Kombination mit angebundenen Links und eventuell deren Satelliten ergeben sich die Dimensionen (siehe Abb. 13).

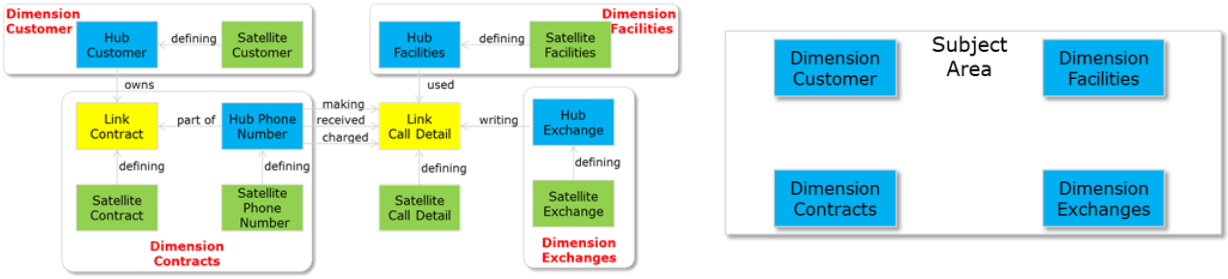
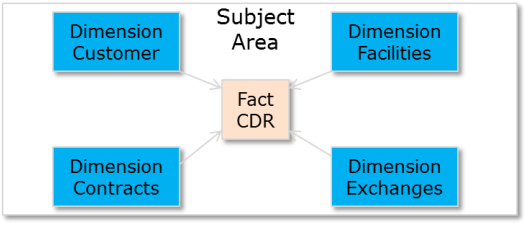


Abb. 13:Identifizierung der Dimensionen

Anschließend müssen schließlich die Fakten gefunden und geladen werden. Auch dies kann theoretisch automatisiert werden, da sich alle Fakten nur aus den Satelliten der Hubs und den verbindenden Links ergeben können, welche als Dimensionen identifiziert wurden (siehe Abb. 14). Auch hier muss gegebenenfalls dieselbe inhaltliche Einschränkung auf Zeilenebene vorgenommen werden wie bei den verwendeten Dimensionen.



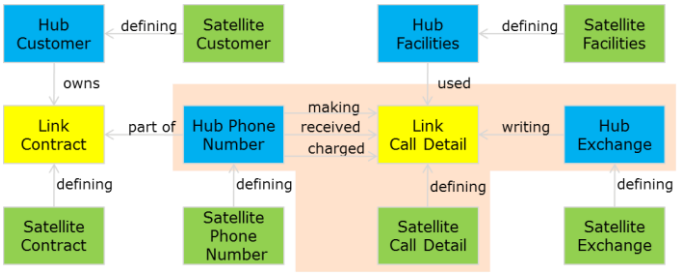


Abb. 14:Identifizierung der Fakten mit fertigem Sternschema

Kontaktadresse:
Markus Kollas
CGI Deutschland Ltd. & Co. KG
Am Limespark 2
D-65843 Sulzbach (Taunus)

Telefon: +49 (0) 175-579 4012
Fax: +49 (0) 6196 7742 555
E-Mail markus.kollas@cgi.com
Internet: de.cgi.com