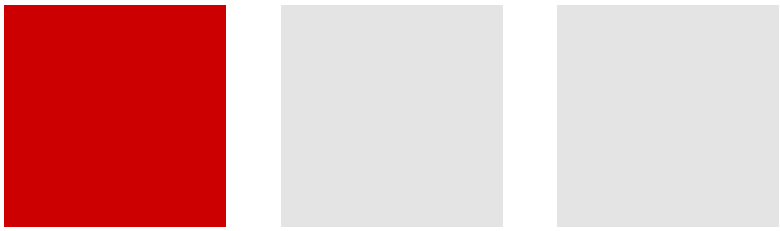

trivadis
makes IT easier.

Modellierung agiler Data Warehouses mit Data Vault

Dani Schnider
Principal Consultant
19. November 2015

Agile Softwareentwicklung hat in vielen Data Warehouses Einzug gehalten, und immer häufiger wird dabei Data Vault als geeignete Methode für die Datenmodellierung genannt. Die einen sehen darin die ideale Vorgehensweise für effizient erweiterbare Data Warehouses, andere sind eher skeptisch dazu eingestellt und befürchten eine unnötige Komplexität. Die nachfolgenden Erläuterungen sollen helfen, hier etwas Licht ins Dunkel zu bringen.

Was ist Data Vault Modellierung?

Data Vault Modellierung ist eine Datenmodellierungsmethode, die spezifisch für Data Warehouses mit häufigen Strukturerweiterungen geeignet ist. Sie wurde in den Neunziger Jahren von Dan Linstedt entwickelt und wird weltweit bereits in vielen agilen DWH-Projekten eingesetzt. Im deutschsprachigen Raum war die Methode bis vor wenigen Jahren kaum bekannt und beginnt sich nun langsam, aber sicher auch hier zu etablieren.

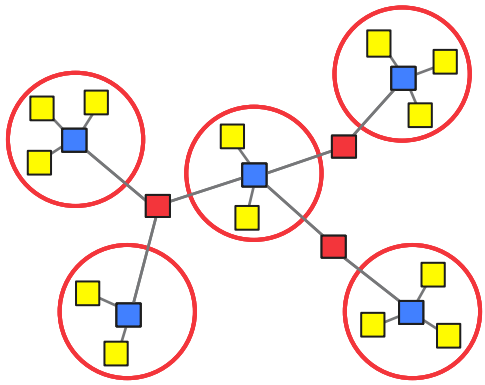

Abbildung 1: Grundprinzip von Data Vault mit Hubs (blau), Links (rot) und Satellites (gelb)

Die Grundidee von Data Vault besteht darin, die Informationen im Data Warehouse so aufzuteilen, dass eine einfache Integration und Historisierung der Daten möglich ist und das Modell ohne Migration der bestehenden Tabellen erweitert werden kann. Pro fachlicher Entität (in Abbildung 1 als rote Kreise dargestellt) wird eine Schlüssel-Tabelle (Hub) mit einem oder mehreren Satelliten-Tabellen erstellt. Eine fachliche Entität ist nicht zwingend identisch mit einer Quelltabelle oder einer Dimension, sondern kann zum Beispiel Daten aus mehreren Quellsystemen beziehen.

Was im ersten Moment sehr komplex aussieht, erlaubt die Modellierung von umfangreichen Datenmodellen mit Informationen aus unterschiedlichen Datenquellen. Für Abfragen mit einem BI Tool ist jedoch ein Data-Vault-Modell nicht geeignet. Data Vault kommt deshalb vor allem bei der Modellierung des Core-Layers in einem Data Warehouse bzw. in einem Enterprise Data Warehouse mit vielen Quellsystemen zum Einsatz. BI-Anwender greifen nicht direkt auf die Tabellen im Data Vault zu, sondern weiterhin auf Data Marts (typischerweise mit einem dimensionalen Datenmodell), welche aus dem Data Vault geladen werden.

Data Vault Elemente

Ein mit Data Vault modelliertes Core Data Warehouse besteht somit aus einer Vielzahl von Tabellen, die sich in drei Kategorien aufteilen lassen:

- **Hubs** dienen zur Identifikation einer fachlichen Entität und enthalten neben dem fachlichen Schlüssel (ein oder mehrere Attribute) einen künstlichen Primärschlüssel (Surrogate Key) sowie technische Attribute wie Ladedatum und Quellsystem.
- **Links** werden zur Verknüpfung der Entitäten verwendet. Sie enthalten Fremdschlüssel auf zwei oder mehr Hubs sowie ebenfalls technische Attribute wie Ladedatum und Quellsystem. Somit werden Beziehungen in einem Data Vault immer als n-zu-n-Beziehungen modelliert.
- **Satellites** enthalten die beschreibenden Attribute der Entitäten oder Beziehungen und sind über einen Fremdschlüssel mit genau einem Hub oder Link verbunden. Häufig werden pro Hub mehrere Satellites erstellt, beispielsweise einen pro Quellsystem. Auch in den Satellites werden Ladedatum und Quellsystem pro Datensatz festgehalten. Das Ladedatum ist zusammen mit der Hub-Referenz der Primärschlüssel der Tabelle. Dies erlaubt eine vollständige Historisierung der Datensätze.

Ein wichtiger Grundsatz in Data Vault ist, dass Beziehungen zwischen Entitäten immer über Links modelliert werden, die auf Hubs verweisen. Eine direkte Verbindung zwischen Hubs ist nicht erlaubt, ebenso verboten sind Fremdschlüssel in einem Satellite, die auf einen „fremden" Hub zeigen.

Die strikte Trennung zwischen Hubs, Links und Satellites ist zentral für die Integration von Daten aus mehreren Quellen und die Erweiterbarkeit des Datenmodells. Außerdem ermöglicht die „sture" Anwendung der vorgegebenen Regeln die automatisierte Generierung von Tabellen und ETL-Prozessen für ein Data Vault Modell. Für DWH-Generatoren ist es besonders wichtig, dass ein einheitlicher Ansatz mit möglichst wenig Spezialfällen und Ausnahmen verwendet wird.

Design eines Data Vault Modells

Die Erstellung eines Data Vault Modells erfolgt typischerweise in mehreren Schritten:

1. Zuerst werden die fachlichen Entitäten ermittelt, die für das Datenmodell relevant sind. Entitäten sind Objekte oder Ereignisse, die eine eigene Identität haben. Im nachfolgenden Beispiel (siehe Abbildung 2) sind dies Kunden, Produkte und Bestellungen. Für jeden dieser Entitätstypen wird ein **Hub** erstellt (H_CUSTOMER, H_PRODUCT, H_SALES_ORDER). Die wohl schwierigste Aufgabe dabei ist es, einen geeigneten Business Key zu finden, einen fachlichen Schlüssel, der zur Identifikation einer Entität verwendet werden kann. Ein naheliegender, aber ungeeigneter Ansatz wäre es, den Primary Key des Quellsystems zu verwenden, aber bei unterschiedlichen Datenquellen funktioniert dies nicht mehr. Wie kann sonst festgestellt werden, dass der Kunde „Hans Müller" aus der Kundendatenbank die gleiche Person ist wie der Online-User „hans@mueller-gmbh.com" aus dem Web-Shop?

2. Nun werden die Beziehungen zwischen den Hubs modelliert. Dazu wird zwischen zwei oder mehreren Hubs ein **Link** erstellt. In unserem Beispiel wird die Beziehung zwischen Kunden und Bestellung über den Link L_CUST_ORDER abgebildet. Die Produkte, die zu einer Bestellung gehören, werden über den Link L_SALES_ITEM verbunden. Weshalb ist eine Bestellposition kein eigener Hub, sondern ein Link? Im Gegensatz zur Bestellung, die z.B. durch eine Bestellnummer oder eine Transaktions-ID identifiziert werden kann, handelt es sich hier nur um eine „Bestellzeile" (Sales Item), die kein eigenständiges Ereignis oder Objekt darstellt.
3. Schließlich werden die beschreibenden Attribute zu den Hubs und Links ermittelt und in Satellites modelliert. Ein **Satellite** ist immer einem einzelnen Hub oder Link zugeordnet. Wie unser Beispiel zeigt, können aber pro Hub oder Link mehrere Satelliten existieren. Für den Hub H_CUSTOMER sind drei Satelliten definiert. Was die Gründe dafür sind, werden wir gleich näher betrachten. Auch für Links können Satellites modelliert werden, falls es beschreibende Attribute zu einer Beziehung gibt. Bei den Bestellpositionen sind dies zum Beispiel die Anzahl und der Preis pro Stück.

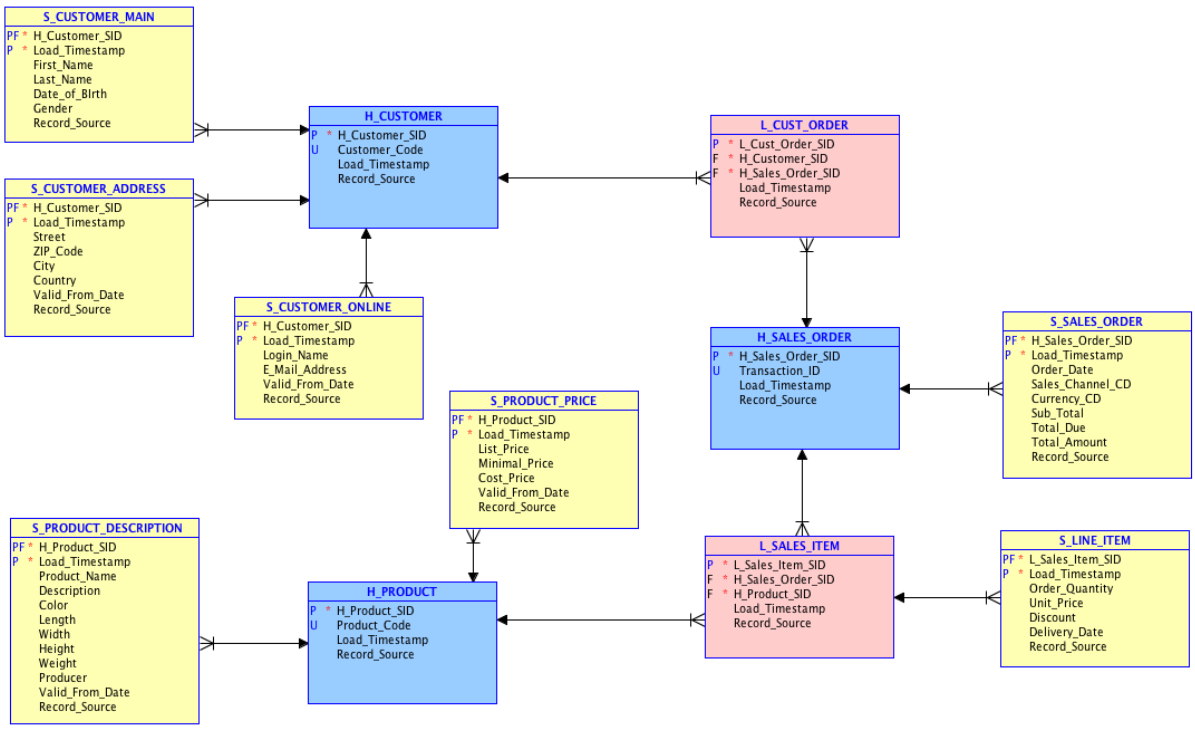

Abbildung 2: Beispiel für ein Data Vault Modell

Für den Hub H_CUSTOMER sind drei Satelliten definiert:

- S_CUSTOMER_MAIN enthält die allgemeinen Informationen wie Name des Kunden
- S_CUSTOMER_ADDRESS wird zur Speicherung der Adressdaten des Kunden verwendet
- S_CUSTOMER_ONLINE enthält zusätzliche Daten aus einer Web-Shop-Applikation

Weshalb diese Aufteilung? Für die Verteilung der Attribute in mehrere Satelliten gibt es verschiedene Strategien. Beispielsweise können Daten, die selten ändern (Name, Geburtsdatum, Geschlecht) in einem anderen Satelliten gespeichert werden als häufiger ändernde Daten (Adresse). Ein anderer Ansatz besteht darin, pro Quellsystem eine Satelliten-Tabelle zu erstellen, um die Informationen zu unterschiedlichen Zeitpunkten zu laden. Oder es werden nachträglich Satelliten hinzugefügt, um Erweiterungen des Datenmodells implementieren zu können.

Hier kommt eine wichtige Eigenschaft von Data Vault ins Spiel: Die einfache Erweiterbarkeit des Datenmodells. Durch die strikte Aufteilung in Hubs, Links und Satellites können neue Anforderungen (zusätzliche Attribute und Beziehungen für bestehende Entitäten sowie neue fachliche Entitäten) in einem Data Vault Modell ergänzt werden, indem neue Tabellen hinzugefügt werden. Bereits vorhandene Tabellen müssen durch solche Erweiterungen weder erweitert noch migriert werden. Dies ist ein wichtiger Vorteil in agilen DWH-Projekten mit häufigen Datenmodellanpassungen.

Historisierung

Ein weiterer wichtiger Aspekt in einem Data Warehouse ist die Historisierung der Daten. In einem Data Vault Modell erfolgt diese ausschließlich in den Satelliten. Da der Ladezeitpunkt Teil des Primärschlüssels jeder Satelliten-Tabelle ist, können alle Änderungen von fachlichen Attributen lückenlos festgehalten werden. Ein Data Vault Modell bietet somit die Möglichkeit, jeden Stand der Daten zu einem beliebigen Zeitpunkt in der Vergangenheit zu ermitteln. Das folgende Beispiel zeigt verschiedene Datenänderungen einer Kundin zu unterschiedlichen Zeitpunkten (t_1 bis t_8).

Jede Änderung führt zu einer neuen Version in einem oder mehreren Satelliten-Tabellen des Hubs H_CUSTOMER (siehe Abbildung 3).

	Quelldaten	Art der Änderung
t_1	ANNA BIERI, ZUERICH	Initiale Version
t_2	Anna Bieri, Zuerich	Korrektur Groß-/Kleinschreibung
t_3	Anna Bieri, Zuerich, abieri@greenmail.ch	Kundin hat neu eine Mailadresse
t_4	Anna Bieri, Zürich, abieri@greenmail.ch	Datenkorrektur (Umlaut in Wohnort)
t_5	Anna Bieri, Zürich, anna.bieri@yellowmail.ch	Neue Mailadresse
t_6	Anna Hartmann-Bieri, Hamburg, a_l_hartmann@web.de	Kundin heiratet und zieht nach Hamburg
t_7	Anna Bieri Hartmann, Hamburg, anna.bieri@web.de	Namensänderung, neue Mailadresse
t_8	Anna Bieri Hartmann, Basel, anna@hartmann-bieri.ch	Umzug nach Basel, neue Mailadresse

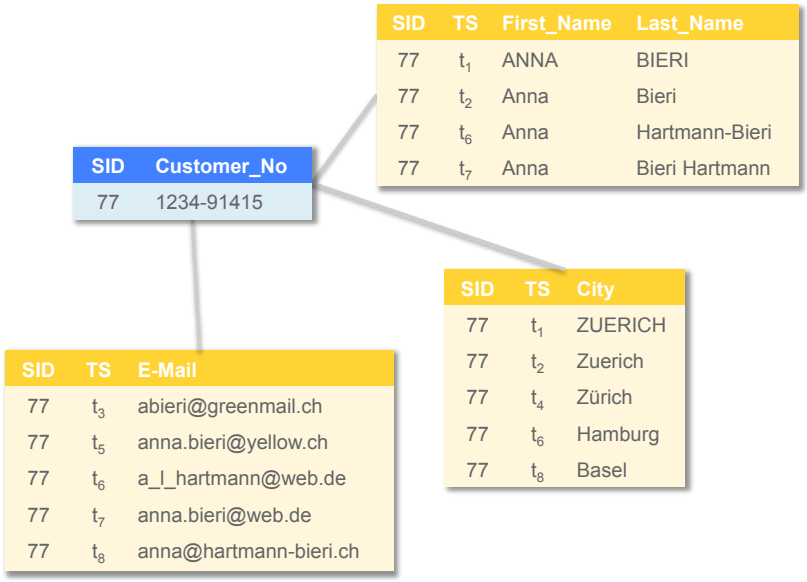

Abbildung 3: Historisierung von Kundendaten in den Satelliten

Die Historisierung erfolgt in jeder Satelliten-Tabelle separat, indem die Attribute der neuesten Datenlieferung mit der letzten Version im Satelliten verglichen wird (Deltaermittlung). Wenn sich die Daten in mindestens einem Attribut unterscheiden, wird ein neuer Datensatz mit dem Schlüssel auf den Hub und dem aktuellen Ladezeitpunkt geschrieben. Die Deltaermittlung und das Schreiben von neuen Versionen ist sehr einfach und autonom für jede Satelliten-Tabelle. Schwieriger wird es, die Zeitintervalle aus mehreren Satelliten zusammenzuführen, wie wir anschließend sehen werden.

ETL-Prozesse für Data Vault

Das Laden von Daten in ein Data Vault Modell erfolgt nach einfachen und einheitlichen Mustern. Die entsprechenden ETL-Prozesse bestehen aus Key Lookups, Deltaermittlungen und INSERT-Statements auf Hubs, Links und Satellites. Da die verschiedenen Tabellen eines Typs unabhängig voneinander geladen werden können, ist eine parallelisierte Ausführung der ETL-Prozesse möglich. Ein gesamter Ladelauf besteht aus folgenden Schritten:

1. Paralleles Laden aller Hubs
2. Paralleles Laden aller Links sowie Hub Satellites
3. Paralleles Laden aller Link Satellites

Dass die Links erst nach den Hubs und die Satellites erst nach den referenzierten Hubs bzw. Links geladen werden können, liegt an den Key Lookups auf die zugehörigen Surrogate Keys. Diese können vermieden werden, wenn als Schlüssel statt Sequenznummern Hashwerte verwendet werden Dieses Verfahren wird typischerweise bei Data Vault 2.0 eingesetzt.

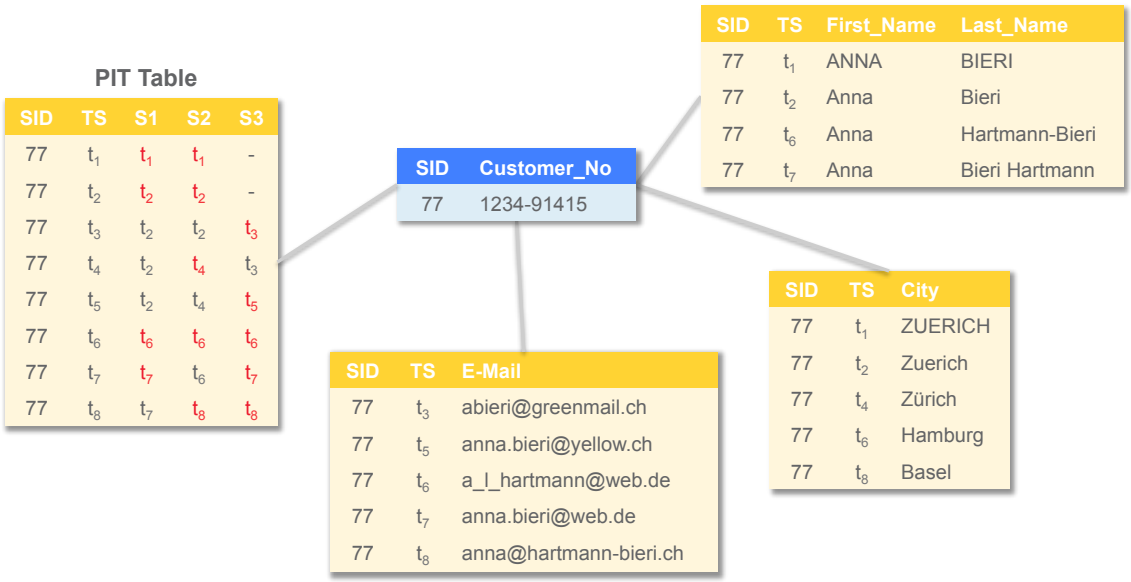

Abbildung 4: Point in Time (PIT) Tabelle zur Ermittlung von Gültigkeitsintervallen

Komplexer als das Laden eines Data Vaults ist das Extrahieren von Daten aus dem Data Vault in dimensionale Data Marts. Um eine Dimensionstabelle oder eine Faktentabelle zu laden, müssen teilweise mehrere Hubs, Links und Satellites zusammengeführt werden. Solange wir nur am aktuellen Stand der Daten interessiert sind, d.h. Dimensionstabellen mittels Slowly Changing Dimension Typ 1 (SCD1) laden, ist dies noch relativ einfach. Aber beim Laden von SCD2-Dimensionen stoßen wir auf eine weitere Herausforderung. Für Hubs mit mehreren Satelliten müssen allen Gültigkeitsintervalle ermittelt werden, die durch Kombination der Gültigkeiten der einzelnen Satelliten entstehen.

Dazu wird eine zusätzliche Satelliten-Tabelle erstellt, die mit der jeweils aktuellen Gültigkeit aller Satelliten zum Ladezeitpunkt gefüllt wird. Mit Hilfe dieser sogenannten „Point in Time"- oder PIT-Tabelle können bei der Extraktion die jeweils gültigen Versionen der einzelnen Satelliten ermittelt und in die Dimensionstabelle geschrieben werden (siehe Abbildung 4).

Die ETL-Prozesse zum Laden der Hubs, Links und Satellites, aber auch jede zum Extrahieren der Daten in Dimensionen und Faktentabellen sind nach immer gleichartigen Mustern aufgebaut. Das hat einen weiteren Vorteil: Wiederkehrende Ladelogik muss nicht manuell implementiert werden, sondern kann relativ einfach mittels geeigneten Generatoren implementiert werden. Vor allem in großen Data Warehouses mit vielen Tabellen und häufigen Strukturänderungen ist dies ein wesentlicher Aspekt.

Zusammenfassung

Data Vault Modellierung bietet Vorteile bei der Integration von verschiedenen Datenquellen, bei der Historisierung von Datenänderungen und bei Erweiterungen der Datenstrukturen. Durch die einheitlichen Regeln für Modellierung und Ladestrecken lassen sich die Tabellen und ETL-Prozesse einfach generieren. Diese Vorteile kommen hauptsächlich in agilen Entwicklungsprojekten von großen Data Warehouses mit unterschiedlichen Quellsystemen zum Tragen. Für kleinere oder relativ statische DWH-Systeme ist der Nutzen geringer. Hier werden wohl weiterhin eher klassische DWH-Architekturen mit relationalen oder dimensionalen Datenmodellen zur Anwendung kommen.

Entscheidend für einen erfolgreichen Einsatz der Data Vault Modellierung ist, dass die Konzepte und Elemente von Data Vault dem ganzen Entwicklungsteam bekannt sind und dass die vorgegebenen Regeln konsequent eingesetzt werden. Nur durch eine einheitliche Anwendung des Regelwerks von Data Vault kann sichergestellt werden, dass das Data Vault Modell verständlich und erweiterbar bleibt.

Dani Schnider
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg
Internet: www.trivadis.com

Tel: +41(0)44-808 70 20
Fax: +41(0)44-808 70 21
Mail: info@trivadis.com