Argumentatie tegen 'end-dating links'

Asymmetrische links in Data Vault (1)

Harm van der Lek

In een vorig artikel (zie DB/M 2, 2010) hebben we Data Vault (DV) besproken en een methode laten zien hoe men een DV-model op natuurlijke wijze kan laten ontstaan door te vertrekken vanuit een Entiteit-Relatie-model, een informatiemodel waar het tijdsaspect (historie van attributen bijvoorbeeld) niet is gemodelleerd. Dit laatste is dus een model zoals de wereld (de 'universe of discourse') er op één moment in tijd uitziet.

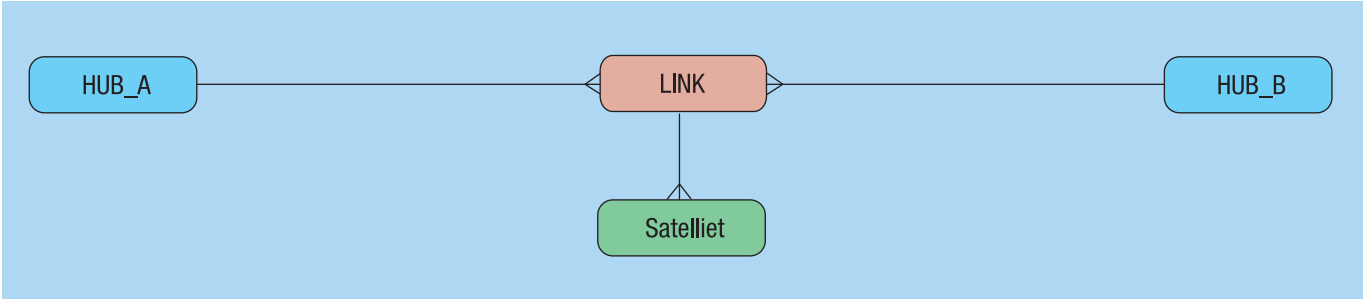
We hebben toen gezien dat we, onder wat extra aannamen, zoals de HUB isolatieregel, redelijk goed in de buurt komen van de Data Vault standaard. Er was echter nog één discussiepunt over: we bleven zitten met twee soorten linktabellen, hetgeen op dat moment nog 'verboden' leek te zijn. Inmiddels, na een uitvoerige discussie op de LinkedIn DV discussiegroep, lijken de gedachten dezelfde kant op te gaan. Zelfs de 'godfather' van DV, Dan Linstedt, heeft voorzichtig geopperd dat we maar eens moesten gaan experimenteren met wat ik asymmetrische links noem. We nemen deze eens uitvoerig onder de loep, want het blijkt tamelijk subtiele kost. In dit eerste deel introduceren we de discussie en de bezwaren tegen een einddatum in de link. Persoonlijk ben ik daar een voorstander van en dus ga ik in deel 2 die bezwaren ontkrachten.

In Data Vault hebben we drie soorten tabellen: HUB-, satelliet- en linktabellen. We gaan die laatste bekijken en beperken ons eerst even tot binaire linktabellen. Dit zijn tabellen die twee HUB's (zeg A en B) verbinden. Zoals we zien in afbeelding 1 verzorgt zo'n linktabel een veel-op-veel relatie tussen de twee HUB-tabellen. De reden dat de relatie tussen de rijen van A en B veel-op-veel is kan twee oorzaken hebben: op één moment in tijd is de relatie al veel-op-veel; op één moment in tijd is de relatie één-op-veel, maar aangezien er in de loop van de tijd veranderingen kunnen optreden wordt het een veel-op-veel.
In het tweede geval is er dus sprake van een 'business rule' die uitgedrukt kan worden in afbeelding 2. We noemen de link dan *asymmetrisch*, omdat immers de rollen van A en B verschillen.

Het probleemloze geval

Laten we eerst eens een voorbeeld bekijken van een symmetrische link (geval 1). Stel we registreren de belangstelling die klanten voor onze producten hebben en op één moment in tijd kan een klant voor meerdere producten tegelijk interesse hebben. Uiteraard kunnen meerdere klanten tegelijk hun oog laten vallen op hetzelfde product. Met andere woorden: de 'interesse' relatie tussen klanten en producten is een typische veel-op-veel relatie. We hebben dus A=Klant en B=Product.
In afbeelding 3 zien we een voorbeeld van zo'n linktabel.

Afbeelding 4 geeft voor de volledigheid de twee bijbehorende HUB-tabellen (4a en 4b). We zien dat de relatie tussen de kolommen A_SQN (SQN staat voor Sequence nummer, een in DV veel gebruikte naamconventie voor betekenisloze nummers) en



Afbeelding 1: Binaire linktabel met HUB's en satelliet.

Afbeelding 2: Business Rule.

Key 1
Key 2
Key 3

A_B_SQN	A_SQN	B_SQN	LOAD_DTS
1	5	8	2011-04-23
2	7	8	2011-04-23
3	2	4	2011-04-23
4	5	4	2011-05-16

Afbeelding 3: Linktabel met (alternatieve) sleutels.

B_SQN inderdaad veel-op-veel is. De kolom LOAD_DTS (DTS: Date Time Stamp) bevat de datum(-tijd) waarop de informatie het datawarehouse bereikte. Verder geven we elke rij ook nog een uniek volgnummertje (A_B_SQN), hetgeen zo dadelijk nog van nut zal blijken te zijn. Dit veld is de primaire sleutel (Key 1) van deze tabel en de combinatie van A_SQN en B_SQN is een secundaire sleutel (Key 2). Tot zover representeert onze linktabel de veel-op-veel relatie tussen klanten en producten, de 'interesse-link'. Dit soort zuivere veel-op-veel relaties komt naar mijn gevoel in de natuur niet zo veel voor. Met 'zuiver' bedoel ik, dat het hierbij dan ook blijft. Meestal is er namelijk wel sprake van een interessant attribuut dat men kan associëren met de elementen van zo'n relatie. In dit geval zou dat bijvoorbeeld de mate van interesse kunnen zijn. Gelukkig kunnen we in DV een satelliet hangen onder een link. Op deze manier kunnen we dit attribuut en zijn historie vastleggen. In afbeelding 5 is dit gebeurd. We kunnen constateren dat de interesse van Truus (5) voor een Auto (8) in eerste instantie 'Groot' was, maar later (op 2011-06-15) veranderde in 'Matig'. We zien dus dat in geval 1 er ook sprake kan zijn van extra attributen die men aan de relatie wil koppelen en waarvan men de historie vervolgens wil vasthouden. Dit kan met een satelliet onder de link. So far so good.

Het discussiepunt

Laten we nu eens het tweede geval onder de loep nemen. We veranderen het voorbeeld enigszins. De A_HUB-tabel blijft hetzelfde, zij het dat we deze personen nu beschouwen als werknemers. De B_HUB-tabel wijzigen we. De rijen hierin representeren nu afdelingen zoals in afbeelding 4c is weergegeven. Laat dit nu eens een asymmetrische link zijn. We nemen namelijk aan dat een werknemer op één moment in tijd, maar voor één afdeling werkt. Dit klinkt als een 'business rule' en dat is het ook! De vraag die we nu willen bespreken is of je het verschil tussen de twee typen links mag zien in de structuur van de link. Laten we ons eerst eens afvragen of je het verschil al helemaal wel kunt zien. We hebben al opgemerkt dat A_SQN en B_SQN, de verwijzingen naar de HUB's, een alternatieve sleutel vormen (Key 2). Zijn er nog meer? Van het voorbeeld (afbeelding 3) is het duidelijk dat de combinatie van B_SQN en LOAD_DTS geen alternatieve sleutel is. Wat we kunnen beweren is nu het volgende: *ALS de link asymmetrisch is DAN zal de combinatie van A_SQN en LOAD_DTS uniek moeten zijn (Key 3).*

Voorbeeld: stel dat we in rij 4 ook LOAD_DTS=2011-04-23 zouden hebben. Dan geeft dat aan dat op hetzelfde (laad)moment de relaties (5, 8) en (5, 4) geldig zouden zijn. Met andere woorden, dat Truus tegelijkertijd voor Marketing en voor Financiën zou werken:

A_SQN	Naam
2	Piet
5	Truus
7	Klaas

a.

B_SQN	Product
4	TV
8	Auto

b.

B_SQN	Afdeling
4	Marketing
8	Financiën

c.

Afbeelding 4: De HUB-tabellen.

A_B_SQN	LOAD_DTS	LOAD_EDTS	Mate van interesse
1	2011-04-23	2011-06-15	Groot
1	2011-06-15	9999-12-31	Matig
2	2011-04-23	9999-12-31	Klein
3	2011-04-23	9999-12-31	Groot
4	2011-05-16	9999-12-31	Groot

Afbeelding 5: Satelliet onder interesse-link.

a.

A_B_SQN	A_SQN	B_SQN	LOAD_DTS	LOAD_EDTS
1	5	8	2011-04-23	2011-05-16
2	7	8	2011-04-23	9999-12-31
3	2	4	2011-04-23	9999-12-31
4	5	4	2011-05-16	9999-12-31

b.

A_B_SQN	A_SQN	B_SQN	LOAD_DTS	LOAD_EDTS
1	5	8	2011-04-23	2011-05-16
4	5	4	2011-05-16	9999-12-31

Afbeelding 6: Link met einddatum.

in tegenspraak met onze aanname dat ze maar voor één afdeling tegelijk werkt. Het omgekeerde van bewering 1 hoeft niet waar te zijn. We hebben immers gezien dat hetzelfde voorbeeld (afbeelding 3) ook een symmetrische link kan representeren.
Stel, we kijken nu alleen naar afbeelding 3 en veronderstellen dat de link asymmetrisch is. Dit betekent dan dat we rij 4 als volgt moeten interpreteren: vanaf datum 2011-05-16 wordt de relatie (5, 4) van kracht. Dan kan vanaf dat moment de relatie (5, 8) niet meer geldig kan zijn, want deze beide relaties mogen niet tegelijkertijd waar zijn. In de praktijk wordt dit soort asymmetrische links gevoed door een combinatie van primaire en verwijzende sleutel in het bronsysteem, dus in dat geval zal (5, 8) inderdaad automatisch zijn verdwenen uit de bron, aangezien er een update van de verwijzende sleutel heeft plaatsgevonden. In ons voorbeeld betekent rij 4 dus dat Truus vanaf 2011-05-16 voor Marketing (4) is gaan werken en haar activiteiten voor Financiën (8) heeft gestaakt.

> Er blijkt in het asymmetrische geval een extra alternatieve sleutel mogelijk

De conclusie tot nu toe is dat je het verschil nauwelijks kunt zien. Er blijkt in het asymmetrische geval een extra (of zoals we zullen zien *een andere*) alternatieve sleutel mogelijk (Key 3). Dus 'no big deal so far'. Er ontstaat echter ook een natuurlijke behoefte om een LOAD_EDTS toe te voegen aan de tabel en wel zodanig dat de history van de objecten A en hun relatie tot B te volgen is.

In ons voorbeeld: Truus en haar carrièrepad langs afdelingen (*remember*: ze is zakelijk monogaam. Op één moment in tijd werkt ze maar voor één afdeling). In afbeelding 6a zien we dit gebeuren. Voor alle duidelijkheid hebben we in afbeelding 6b alleen de rijen getoond met A_SQN= 5. We zien dat de link nu trekjes begin te krijgen van een satelliet: we houden historie bij. De inhoud van deze kolom LOAD_ETDS is puur afleidbaar. Hij is alleen maar opgenomen om de query's te vergemakkelijken (inclusief de query's om datamarts te voeden). Wat dat betreft is de functie precies gelijk aan de LOAD_EDTS zoals we die ook in satelliettabellen opnemen. Langzamerhand hebben we de structuur gewijzigd. Raphael Klebanov noemt dit een satlink (slink).

De vraag is of dit is toegestaan. De bedenker van DV heeft zich hier lange tijd tegen verzet. Quote: "There is only one type of Link table and that is a Link table" (Dan Linstedt 2009). Op dit front nu heeft zich een fikse discussie afgespeeld op de LinkedIn DV-groep gestart door onze eigen DV-goeroe Ronald Damhof. De titel is 'End-dating Links'. De titel is eigenlijk wat misleidend want als het alleen om het toevoegen van een end-date aan de tabel zou gaan dan was het inderdaad 'no big deal'. De inhoud van zo'n kolom is (opnieuw) afleidbaar, dus geen probleem als iemand deze dropt. Er zijn echter wat dieper liggende verschillen zoals we hierboven hebben gezien, met name in de sfeer van de (alternatieve) sleutels. Verder is het zo, we moeten het toegeven, dat de load routines wat verschillen. Deze lijken wat meer op die van de satellieten.

Bezwaren?

We sommen de argumenten tegen 'End-dating Links' of beter asymmetrische links (of satlinks) die ik in loop van de tijd van diverse kanten gehoord heb, eerst eens op:

A. *Het onderscheid is gebaseerd op een 'business rule' en aangezien business regels kunnen veranderen, moet men het ontwerp van DV hierop niet baseren;*
B. *Dit is een vorm van historie en dat moet worden geregeld in een satelliet van de link;*
C. *De combinatie van A_SQN en B_SQN kan niet langer een unieke index meer zijn;*
D. *Aangezien de key-structuren verschillen heb je een nieuw concept geïntroduceerd.*

De lezer krijgt een poosje de tijd om zelf na te denken over deze bezwaren, want we eindigen dit eerste deel met een ware cliffhanger: pas in deel 2 gaan we deze tegenwerpingen één voor één onder de loep nemen en de vloer ermee aanvegen. Mocht iemand nog meer bezwaren kunnen verzinnen, dan hoor ik dat graag. Wellicht kan ik die in het volgende deel dan ook nog meenemen.

Harm van der Lek (vdlek@vdlek.nl) is BI Architect bij BinckBank en zelfstandig Docent.