Historie sla je op in historietabellen

Asymmetrische links in Data Vault (2)

Harm van der Lek

In het eerste deel (DB/M 1, 2011) hebben we het begrip asymmetrische link geïntroduceerd. Daarbij bleek dat de structurele verschillen erg klein waren. Wel hebben velen (waaronder ikzelf) de neiging om een einddatum toe te voegen aan een dergelijke linktabel. Dit was echter een overtreding van de voorschriften van Dan Linstedt. Nu kun je het natuurlijk 'stiekem' doen, omdat het in de praktijk zo handig is, maar ik ben van mening dat het ook theoretisch te verantwoorden is. Daarom gaan we in dit deel de bezwaren tegen 'end-dating links' eens grondig onder de loep nemen en weerleggen.

Laten we nog even kort het eerste deel samenvatten. Een linktabel geeft per definitie een veel-op-veel relatie tussen de twee HUB-tabellen[1]. Bij een asymmetrische linktabel komt dit echter alleen maar omdat er historie wordt opgeslagen en was de relatie, zonder het tijdsaspect, een één-op-veel. We hebben toen gezien dat dit tot een zeer subtiel verschil leidt in de structuur van de linktabel: er ontstaat een nieuwe alternatieve sleutel, namelijk één van beide verwijzingen naar de HUB (zie hier de asymmetrie!) en de LOAD_DATE. Bovendien is men geneigd om een einddatum op te nemen in de tabel, omdat men dan goed de historie kan volgen van één van beide objecten, bijvoorbeeld Truus en haar carrière langs verschillende afdelingen. Zie afbeelding 6 in combinatie met 4a en 4c. (Zie deel 1 in DB/M 1). Het eerste deel eindigde met een opsomming van argumenten tegen 'end-dating links' die ik in loop van de tijd van diverse kanten had gehoord:

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.*

We gaan deze tegenwerpingen één voor één weerleggen. Voor tegenwerping D zal nog wat uitstel van executie gelden.

Herontwerp nodig?

Ad A. De achtergrond hiervan is dat men geen (of in ieder geval zo weinig mogelijk) herontwerp wil bij dit soort wijzigingen. Let wel: als we het over herontwerp hebben dan bedoelen we: wijzigen van de structuur van het met data gevulde productie-datawarehouse en (niet te vergeten!) de ETL-code. We moeten dit zeer serieus nemen, want het beperken van de inspanningen voor 'herontwerp' is nu eenmaal één van de belangrijkste sterke punten van DV. Er zijn minstens drie antwoorden op deze tegenwerping A:

1. We moeten ons realiseren dat het ontwerp van een DV datawarehouse gebaseerd is op tonnen van dit soort business rules als het gaat om gewone beschrijvende attributen. Immers, als we in een satelliet bijvoorbeeld de (historie van de) postcode van een klant opslaan, dan is dat gebaseerd op de aanname dat, op zijn minst in het source systeem, er op één moment in tijd maar één postcode is. Dus waarom moeilijk doen in het geval van een asymmetrische link? Als men informatie wil opslaan zonder gebruik te maken van business rules, dan moet men niet een database management systeem gebruiken maar een tekstverwerker. Structuur = business rules;
2. Gelukkig is juist in dit geval het herontwerp (van de database!) minimaal. De toegevoegde kolom LOAD_EDTS kan zonder bezwaar weer worden verwijderd, of op metaniveau worden gevlagd als 'no longer in use'. Wel moeten eventueel toepassingen worden aangepast of zelfs verwijderd. We zullen bijvoorbeeld de gebruikers er op moeten attenderen dat hun rapport over de aantallen uren per afdeling nu niet meer gemaakt kan worden (of grondige herbezinning behoeft);
3. Zoals eerder opgemerkt zijn dit soort links vaak gebaseerd op verwijzende sleutels in de bron en hoe vaak komt het nu voor dat deze worden gewijzigd in veel-op-veel relaties? Ja, het komt voor, heel af en toe, daarom gaan we dadelijk nog even op een voorbeeld in en zullen zien dat er dan nog veel meer aan de hand is.

Men zou ook nog de gedachte kunnen opperen om Key 3 in afbeelding 3 niet te implementeren (en Key 2 uiteraard ook niet) in het geval van een asymmetrische link. Wel zou je dan op een andere manier moeten blijven checken of de combinatie van A_SQN en LOAD_DTS uniek blijft. Dit heeft ook het voordeel

dat, mocht dit onverhoopt niet het geval zijn, het ETL-proces niet stopt op een keiharde error. En het suggereert dat men helemaal geen herontwerp (behalve het niet meer achteraf checken) meer nodig heeft als de business rule op een gegeven moment overboord wordt gezet (veel-op-veel wordt). Helemaal geen gekke gedachte en dus zeker doen, maar pas op, je houdt jezelf wel voor de gek als je denkt dat je op deze manier automatisch ontdekt dat de business rule is gewijzigd (dat de link symmetrisch is geworden). We hebben immers gezien dat asymmetrie de uniciteit impliceert, maar niet andersom! Met andere woorden: de regel kan bij de business allang zijn gewijzigd zonder dat we dat (op deze manier althans) in het datawarehouse merken. En dat is pas echt eng.

Het is wel boeiend om even stil te staan bij de vraag of en zo ja wat er nu valt te herontwerpen. We zullen bovendien zien dat het in ieder geval erg onwaarschijnlijk is dat we dit als DWH-team niet zouden merken. Neem maar het geval dat men gaat toestaan dat werknemers op één moment in tijd toch voor meer dan één afdeling aan de slag zouden mogen zijn. Niet eens zo onwaarschijnlijk. Truus is weliswaar overgestapt naar Marketing, maar ze blijft ook nog voor Financiën klussen, met andere woorden: ze werkt voor twee afdelingen tegelijk. Zo op het oog is geen herontwerp nodig (zie punt 2 boven). We zullen het overigens wel te weten komen (wellicht 'the hard way'), want onze load routines zullen moeten zijn aangepast. Het bronsysteem zal namelijk zijn veranderd. Toen de business kenbaar maakte dat de ambitie van Truus om voor twee afdelingen tegelijk te werken mogelijk moest zijn, kwam IT er achter dat er wel een wijziging in het systeem nodig was. Waar het vroeger een verwijzende sleutel van de werknemertabel naar de afdelingtabel was, heeft men dit in het operationele systeem moeten vervangen door een nieuwe tabel waar de (nu) veel-op-veel relatie wordt bijgehouden. En wij zullen dit als DWH-team merken (hopelijk op tijd).

a.

A_B_SQN	LOAD_DTS	LOAD_EDTS	STATUS
1	2011-04-23	2011-05-16	Bestaand
2	2011-04-23	9999-12-31	Bestaand
3	2011-04-23	9999-12-31	Bestaand
4	2011-05-16	9999-12-31	Bestaand
1	2011-05-16	9999-12-31	Niet meer bestaand

b.

A_B_SQN	LOAD_DTS	LOAD_EDTS	STATUS
1	2011-04-23	2011-05-16	Bestaand
4	2011-05-16	9999-12-31	Bestaand
1	2011-05-16	9999-12-31	Niet meer bestaand

Afbeelding 7: Historie van relatie in satelliet?

> Elke foreign key in de source geeft aanleiding tot een asymmetrische link

Conclusie: de herontwerpactiviteiten zullen voornamelijk in de ETL-sfeer zitten en dat betekent dat we dan in ieder geval iets moeten doen, ook al hoeven we niets te veranderen aan het database-ontwerp. Er is nog een, wellicht veel boeiender, verschijnsel dat zich in de praktijk zal voordoen. Voordat de wispelturige business met haar verandering in de regel kwam hadden we twee attributen in de werknemertabel zitten: 'functie' en 'aantal werkuren per week'. Als u er even over nadenkt zal het u niet verbazen dat, bij de informatieanalyse in verband met het wijzigingsverzoek, bleek dat deze twee attributen opgenomen moesten worden in de nieuwe tabel die de veel-op-veel relatie ging realiseren. Immers, deze twee attributen kunnen (en zullen) verschillen voor dezelfde persoon werkend voor verschillende afdelingen. Het zijn dus attributen van de veel-op-veel relatie en die zullen moeten verhuizen naar een nieuw te creëren satelliet onder de link. Hoezo geen herontwerp in Data Vault? Uiteraard komt het een enkele maal voor dat een geniale DV ontwerper dit allemaal heeft voorzien en al in het eerste ontwerp deze attributen in een satelliet onder de (toen nog asymmetrische) link heeft gehangen.

De andere tegenwerpingen

Ad B. Inderdaad hebben we dit principe al toegepast in het voorbeeld waarmee we in deel 1 begonnen (historie van de mate van interesse). We antwoorden op deze tegenwerping als volgt. In de eerste plaats wordt hier natuurlijk alleen maar een regel geponeerd (gij zult ...), zonder dat wordt uitgelegd waar de regel goed voor is. Laten we desondanks gehoorzaam zijn en dit proberen te doen. We krijgen dan zoiets als in afbeelding 7a. Dit ziet er niet al te prettig uit. Laten we ons weer beperken tot de rijen in deze satelliet die wijzen naar rijen in de link die A_SQN=5 hebben (1 en 4) in de hoop dat het er dan wat duidelijker op wordt. We krijgen afbeelding 7b. Als men heel goed kijkt (combineer afbeeldingen 7b en 3 en eventueel 4a en c), dan kan men hier de verhuizing per 2011-05-16 van Truus van Financiën naar Marketing in zien, maar onhandig is het wel, zeker als je het vergelijkt met 6b. Nu kan men natuurlijk zeggen: "Dan maar onhandig, het is in ieder geval netjes volgens de standaard theorie", maar ik heb zelf de neiging om in zo'n geval te kijken of de theorie wel goed is. Immers 'niets zo praktisch als een goede theorie', dus wellicht is er wat mis met de standaard. En dat is het mijns inziens zeker, want ik zou willen beweren dat

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	2011-11-01
5	5	8	2011-11-01	9999-12-31

Afbeelding 8:A-B-A Probleem.

het fundamenteel fout is om te proberen de historie van Truus en met name haar omzwervingen langs de afdelingen op deze manier vast te leggen. Laten we namelijk weer eens goed kijken waarvan nu precies de historie wordt bijgehouden als we een satelliet onder een link hangen. Daar is eigenlijk maar één antwoord op mogelijk: van die 'dingen' die worden gerepresenteerd door de rijen in de linktabel. In het interesse voorbeeld was dat ook duidelijk: daar stond in de eerste twee rijen van afbeelding 5 de historie genoteerd van rij 1 van afbeelding 3 en dat was het object 'De belangstelling van Truus voor een Auto'. Wat staat er dan in afbeelding 7? Welnu, ook weer de historie van de 'dingen' die worden gerepresenteerd door de rijen in de linktabel. Bijvoorbeeld weer rij 1. Dit is nu: 'Het lidmaatschap van Truus van de afdeling Financiën'. Van dit 'ding' houden we de historie vast, met name het attribuut STATUS dat op 2011-05-16 van 'Bestaand' in 'Niet (meer) bestaand' veranderde. Let wel: hieruit is uiteindelijk wel de historie van Truus zelf te herleiden, maar het is indirect en onhandig (om over de performance-aspecten nog maar te zwijgen). En, nogmaals, dit zal erg vaak voorkomen, want elke foreign key in de source geeft aanleiding tot een asymmetrische link.

Ad C. In het begin van het artikel hadden we als vanzelfsprekend aangenomen dat de combinatie van A_SQN en B_SQN een alternatieve sleutel (Key 2 in afbeelding 3) zou zijn. Dit valt inderdaad niet meer te handhaven als we overgaan op een satlink. Stel immers dat Truus toch niet op haar plaats was bij Marketing en dus op 2011-11-01 weer terugkeert naar Financiën. Dan willen we toch graag dat de link eruit ziet als in afbeelding 8. Hoe erg is het dat we Key 2 niet kunnen handhaven? Het lijkt erop dat we een soort regel hebben dat de combinatie van alle verwijzingen in een linktabel een sleutel zouden moeten vormen. Zo'n regel bestaat echter helemaal niet. We geven een ander voorbeeld. Stel we hebben een standaard voorbeeld met orders en orderregels. Als (zoals het Northwind voorbeeld van Microsoft) een orderdetail wordt geïdentificeerd door een ordernummer en het product dat op die regel staat, dan kan inderdaad een keurige link worden gedefinieerd waarin de twee hub-verwijzingen samen uniek zijn. Maar het komt ook wel voor dat een orderregel uniek wordt geïdentificeerd door het ordernummer en een regelnummertje, zodat hetzelfde product wel twee keer op een order kan voorkomen. Dan zullen we in de linktabel (een kenmerkende transactielink) als alternatieve sleutel de verwijzing naar de order hub plus het regelnummertje moeten nemen.

Ad D. Dit is een zeer serieuze tegenwerping. De kracht van DV is nu juist het geringe aantal basisconcepten en daarmee de standaardisatie. Het is dan vervelend als allerlei mensen concepten of variaties op concepten beginnen toe te voegen. Ook het idee om een einddatum in een link op te nemen lijkt de DV wereld weer onoverzichtelijker te maken. Immers, in den beginne was de situatie simpel: *historie houd je bij in satellieten en alleen daar*. Nu wordt het ineens: *historie sla je op in satellieten en uh, soms in links en dat noem je dan asymmetrische links*.

Ik wil in een volgend artikel hierop verder ingaan. De essentie is dat je hier op een net andere manier tegenaan kunt kijken. Om een klein tipje van de sluier vast op de lichten: ik wil niet drie basistypen van tabellen onderscheiden, maar slechts twee: niet historietabellen (ook wel primaire tabellen genoemd) en ... historietabellen. Een historietabel wijst naar een primaire tabel en houdt historie bij van de 'dingen' die in die primaire tabel staan. Binnen deze twee basistypen ontstaat op een natuurlijke manier nog een secondaire onderverdeling:
Primaire (non-historie-) tabellen;
- *HUB-tabellen*: onderliggende 'dingen' worden zelfstandig geïdentificeerd door een (combinatie van) code(s), de zogenaamde 'business key';
- *Symmetrische links*: voor de identificatie van de 'dingen' zijn andere 'dingen' nodig;

Historietabellen;
- *Satellieten*: wanneer de bijbehorende primaire tabel de enige is waar hij naar wijst;
- *Asymmetrische links*: wanneer hij ook nog naar andere primaire tabellen wijst.

En nu is de wereld plotsklaps heel simpel: *historie sla je op in historietabellen*, PUNT.

Conclusie

Een linktabel waarin je historie bijhoudt van een veel-op-één relatie is een eenvoudige en verantwoorde constructie en tast de fundamenten van Data Vault niet aan. De meest serieuze tegenwerping, namelijk dat DV complexer wordt gemaakt, kan worden weerlegd door op een iets andere, en in wezen simpeler, manier tegen de fundamenten aan te kijken. Dit laatste behoeft echter nog wel meer toelichting dan in dit artikel is gegeven.

Noot

1. *We beperken ons tot het binaire geval, maar de theorie is gemakkelijk uit te breiden naar linktabellen die meer dan twee referenties naar hub- (of andere link-) tabellen hebben. De asymmetrie van zo'n n-aire link zit in het feit dat één van de hub-referenties een bijzondere rol vervult.*

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