Bepaal de juiste datawarehouse-architectuur voor uw organisatie

Mag het een (opslag)laagje meer zijn?

Frank Habers

Elke periode in de IT kent zijn hypes, trends en discussies. Op het gebied van datawarehousing heeft het afgelopen decennium een architectuurdiscussie tussen twee kampen gewoed: het 'Inmon kamp' versus het 'Kimball kamp'. Nu de meeste discussies zijn gevoerd en de meeste stof is neergedaald, wordt het tijd om de discussie vanuit het huidige tijdsbeeld te evalueren en te toetsen aan de hand van courante en aan de praktijk getoetste ideeën. Hoe bepaalt u de juiste datawarehouse-architectuur voor uw organisatie?

Datawarehouses zijn er in allerlei soorten en maten, maar er zijn twee dominante datawarehouse-architecturen. De ene is de Inmon architectuur met een centraal genormaliseerd datawarehouse en (dimensionele) datamarts. De ander is de Kimball architectuur, waarbij een centraal dimensioneel gemodelleerd datawarehouse wordt gehanteerd. Beide architecturen kennen hierbij nog een data staging area, zie afbeelding 1.

Overeenkomsten

Voordat ik inga op de verschillen tussen beide architecturen, wil ik allereerst stilstaan bij de overeenkomsten van de architecturen, want deze worden vaak onderbelicht. Beide architecturen gaan uit van een centraal datawarehouse, gebaseerd op een relationele database (want zowel een dimensioneel datawarehouse als een genormaliseerd datawarehouse is gebaseerd op het relationele model!) waarbij relevante gegevens op het laagste detailniveau worden opgeslagen. Tevens gaan beide architecturen er vanuit dat dimensionele modellen beter bruikbaar zijn voor de eindgebruiker vanwege eenvoud en performance. Bill Inmon brengt alleen een nuance aan door te stellen dat een dimensioneel model niet voor

> Een dimensioneel model legt de hiërarchische verbanden tussen meetwaarden niet vast

iedere BI-toepassing geschikt is (verderop volgt hiervan een voorbeeld). Een andere overeenkomst is dat beide architecturen een data staging area vereisen, waarin de gegevens worden verzameld in een 'voorportaal' (al dan niet opgeslagen in een relationele database). Deze data staging area heeft verschillende voordelen, zoals de ontkoppeling van de bronsystemen en de vereenvoudiging van de ETL-processen in de vervolgstappen. Daarnaast hebben beide architecturen, ondanks de andere suggesties die nog wel eens gewekt wordt, een enterprise focus en zijn beide architecturen bottom-up en top-down te realiseren, waarbij iteratief en incrementeel ontwerpen mogelijk is. Immers, bij beide architecturen gaat het om het vormgeven van een set aan gegevens, dat leidt niet tot een beperking in aanpak of mogelijkheden.

Historie

In essentie is er één belangrijk achterliggend principieel verschil tussen de architectuur van Ralph Kimball en Bill Inmon. Bij de Kimball architectuur worden bij het modelleren van het centraal datawarehouse meer keuzes vooraf gemaakt. Een belangrijk voorbeeld daarvan is de keuze voor de opslag van historie. Voor dimensies worden keuzes gemaakt voor welke tabellen (en attributen) de waarden historisch worden vastgelegd (via het principe van *slowly changing dimensions*). Dat is een typisch vraaggestuurde redenering, omdat alleen die informatie wordt vastgelegd waarvan verwacht wordt dat die nodig is ten behoeve van analyses. Inmon redeneert veel meer vanuit het aanbod, door in het centrale datawarehouse alle historie vast te leggen, zonder zich te laten sturen (en te beperken) door de vraag. Het grote voordeel hiervan is dat indien de informatiebehoefte met betrekking tot historie verandert, de historie in ieder geval beschikbaar is in het centrale datawarehouse. Bij de Kimball architectuur is een keuze gemaakt voor het al dan niet vastleggen van historie en dat kan worden gezien als een tekortkoming. Dat leidt ook vaak tot de suggestie dat deze architectuur geen enterprise focus heeft. Echter, dan wordt de vraag: is dit hét overtuigende argument om altijd een genormaliseerd datawarehouse te implementeren? Dat is allereerst een kosten/baten-afweging, wil een organisatie dusdanig veel extra investeren (want een extra genormaliseerde laag

is een substantiële extra inspanning!) om de historie die op dat moment niet relevant wordt geacht, toch vast te gaan leggen? Uit mijn ervaring blijkt dat veel organisaties op basis van kennis, ervaring en pragmatische houding een goede keuze ten aanzien van historie kunnen maken, zonder daar achteraf op terug te hoeven komen.

Historical Data Area

Indien het volledig vastleggen van alle historie vanuit aanbodzijde van belang is, betekent dit dan dat er altijd een genormaliseerd centraal datawarehouse vereist is? Nee, er is namelijk een eenvoudig alternatief: het vastleggen van alle historie in de data staging area. Het principe bestaat daarbij uit niets anders dan de historie via tijdstroken (begindatum tot en met einddatum) volgens de tabelstructuur van de bron vast te leggen in de data staging area. Om het verschil te benadrukken met een traditionele staging area zonder historie, wordt deze omgeving ook wel een Historical Data Area (HDA) genoemd. Omdat in de HDA dus alle historie wordt vastgelegd, kan in de toekomst de keuze worden gemaakt om alsnog deze historie beschikbaar te stellen in het dimensionele datawarehouse. Vanzelfsprekend moeten hiervoor wel het datamodel van het dimensionele datawarehouse en de bijbehorende ETL-processen worden aangepast. Meer informatie over de HDA vindt u in het kader. Geconcludeerd kan worden dat voor het beschikbaar houden van historie vanuit een aanbod-gerichte aanpak niet een genormaliseerd centraal datawarehouse vereist is.

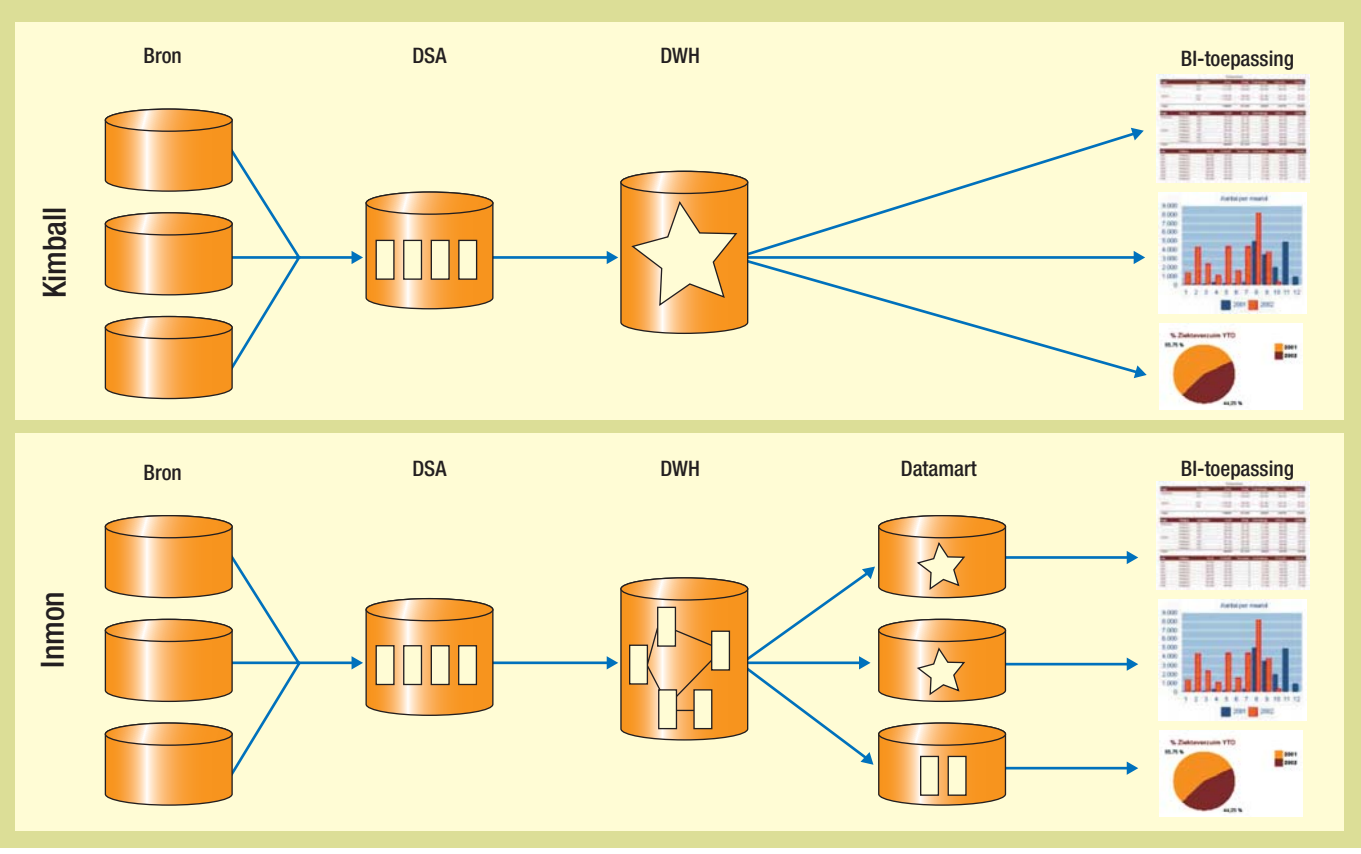
HDA

De gebruikelijke implementatie van een data staging area is het opslaan van de brongegevens volgens de structuur van de bron in een relationele database. Een HDA doet exact hetzelfde, maar voegt een tijdstrook (begindatum tot en met einddatum) toe aan de tabellen, zie afbeelding 2, en daarnaast worden (vanzelfsprekend) de gegevens niet verwijderd. Dit in tegenstelling tot een DSA, waarbij de gegevens tijdelijk worden opgeslagen. Omdat dit een standaardprincipe is, kan ook het ETL-proces voor het laden van de HDA worden gestandaardiseerd, of nog mooier, automatisch worden gegenereerd.

Een bijkomend voordeel is dat door het compleet beschikbaar hebben van de historie, bepaalde ETL-processen richting het datawarehouse eenvoudiger kunnen worden opgezet, evenals het uitvoeren van herstelwerkzaamheden.

Extra laag: datamarts

Een ander verschil tussen de oplossing van Kimball en Inmon is dat Inmon per definitie nog een extra gebruikerslaag (datamarts) hanteert, terwijl Kimball aangeeft dat het datawarehouse fungeert als gebruikerslaag[1]. Maar wat is nu de juiste architectuur, wel of geen extra datamarts?



Afbeelding 1: Architecturen van Kimball en Inmon.

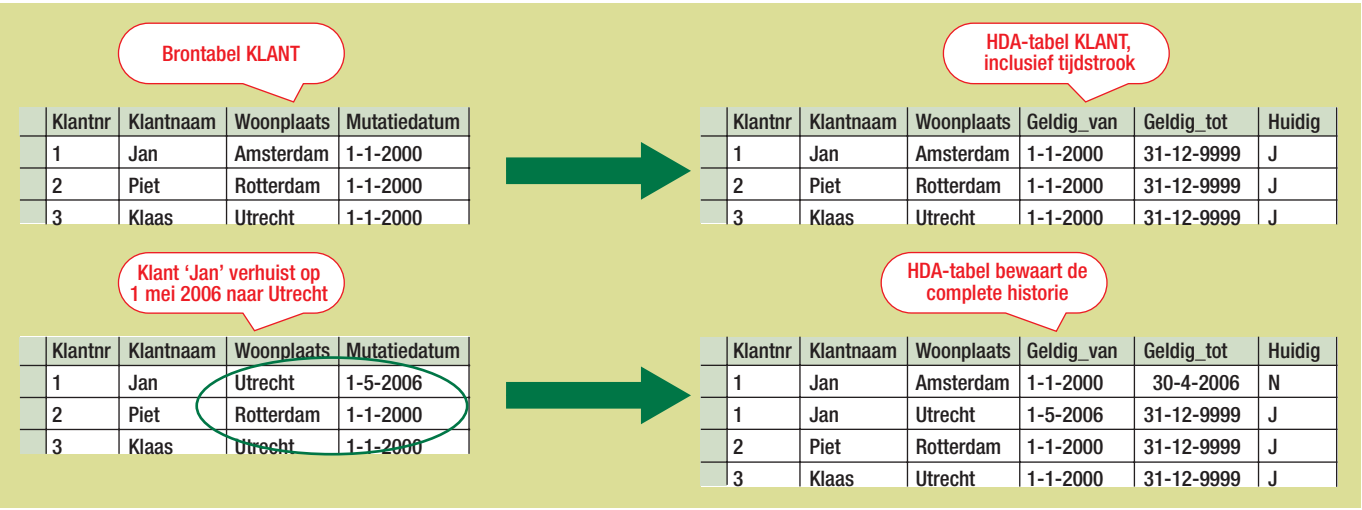
Op deze vraag is geen eenduidig antwoord te geven, want het antwoord wordt situationeel bepaald. Echter, wel kan duidelijkheid worden geschapen over wat het criterium is om tot een keuze te komen. Dat criterium is als volgt: "Elke extra (opslag)laag in een datawarehouse-architectuur moet toegevoegde waarde hebben". De toegevoegde waarde van de extra laag moet zodanig zijn dat de ontwikkel-, beheer- en onderhoudsinspanning van het totale datawarehouse afneemt of de functionaliteit voldoende toeneemt.
Volgens dit principe kan tevens het bestaansrecht van een datawarehouse worden gelegitimeerd. Immers, het rapporteren en analyseren rechtstreeks op de administratieve bronsystemen leidt uiteindelijk (vaak) tot meer inspanning en levert minder functionaliteit op dan dit via een datawarehouse te doen (waarbij veel complexiteit van de bronsystemen wordt 'verborgen' en extra functionaliteit, bijvoorbeeld integratie van gegevens, voor de gebruiker beschikbaar wordt gesteld). In afbeelding 3 is het hierboven genoemde criterium in een fictief voorbeeld toegepast. Iedere organisatie zal soortgelijke cijfers kritisch moeten invullen wanneer een keuze in architectuur wordt gemaakt. Overigens is mijn praktijkervaring dat een extra genormaliseerde laag in geen enkel geval voldoende toegevoegde waarde biedt, gezien de hoge kosten die deze architectuur met zich meebrengt.

Terug naar de vraag in welke situatie een datamart vereist is. Dit is ook afhankelijk van de toegevoegde waarde van deze datamart. Veelal wordt impliciet aangenomen dat datamarts noodzakelijk zijn, terwijl de argumentatie moet worden omgedraaid. Er moet een goede reden zijn om een datamart te implementeren, anders moet de BI-toepassing gebruik maken van het dimensionele datawarehouse. Een voorbeeld van de noodzaak van een datamart is de situatie waarbij een belangrijke BI-applicatie de gegevens in een ander formaat en/of structuur vereist dan de dimensionele structuur in een relationele database. Bijvoorbeeld een KPI-oplossing, waarbij tussen de Key Performance Indicatoren (de meetwaarden van een dimensioneel model) een hiërarchisch verband bestaat (dat voor de gebruiker ook moet worden gevisualiseerd). In afbeelding 4 zien we een voorbeeld van een retailer die KPI's wil zien over leverprestaties van de leveranciers. Een dimensioneel model legt de hiërarchische verbanden tussen meetwaarden niet vast, dus moeten de gegevens in een ander formaat en/of structuur worden vastgelegd voor de KPI-oplossing. Dat is een goede reden voor het implementeren van een extra opslaglaag, een datamart. Overigens blijkt in de praktijk dat voor de meeste rapportage- en analysetoepassingen het dimensionele datawarehouse wel volstaat.

Extra laag: ODS

Ook bij de keuze voor een operational data store (ODS) kan het criterium van de extra laag worden gebruikt. In de praktijk wordt nog wel eens geredeneerd dat een ODS simpelweg vereist is, omdat operationele gegevens *near real-time* beschikbaar moeten worden gesteld. Echter, allereerst moet worden getoetst of dit niet kan worden gefaciliteerd via de bestaande architectuur. Als dan blijkt dat dit niet haalbaar is, bijvoorbeeld omdat de ETL-performance niet volstaat, dan is een (vaak dure) extra laag (de ODS) gelegitimeerd. Belangrijk is dus dat de keuze voor deze extra laag een bewuste kosten/baten-afweging moet zijn.

Terug naar de vraag hoe het centrale datawarehouse gemodelleerd moet worden. Waarom wordt in veel literatuur vaak de suggestie gewekt dat een genormaliseerd centraal datawarehouse meer mogelijkheden biedt voor een bedrijfsbrede oplossing (enterprise focus) en derhalve een verstandige keuze is? Dat wordt mijn inziens voornamelijk ingegeven door drie aspecten: historie; flexibiliteit door middel van genericiteit; toepassen van business rules. Voor wat betreft historie blijkt er een eenvoudig, maar onbekend, alternatief te zijn (de HDA), zodat de bedrijfsbrede oplossing wordt gewaarborgd. Historie is dus geen goede reden voor een genormaliseerd bedrijfsbreed centraal datawarehouse. Het tweede argument, flexibiliteit door middel van genericiteit, vergt een nadere toelichting. Een genormaliseerd datawarehouse



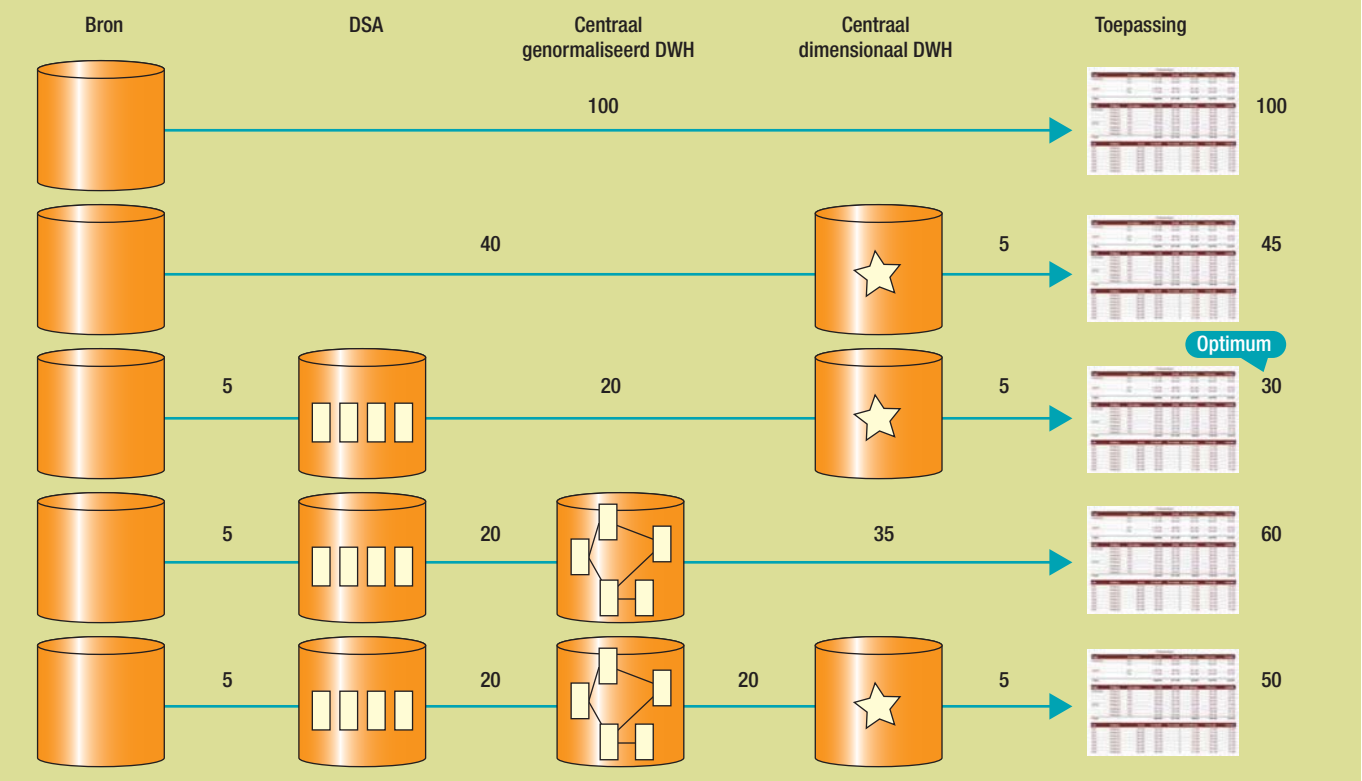
Klantnr	Klantnaam	Woonplaats	Mutatiedatum
1	Jan	Amsterdam	1-1-2000
2	Piet	Rotterdam	1-1-2000
3	Klaas	Utrecht	1-1-2000

Klantnr	Klantnaam	Woonplaats	Geldig_van	Geldig_tot	Huidig
1	Jan	Amsterdam	1-1-2000	31-12-9999	J
2	Piet	Rotterdam	1-1-2000	31-12-9999	J
3	Klaas	Utrecht	1-1-2000	31-12-9999	J

Klantnr	Klantnaam	Woonplaats	Mutatiedatum
1	Jan	Utrecht	1-5-2006
2	Piet	Rotterdam	1-1-2000
3	Klaas	Utrecht	1-1-2000

Klantnr	Klantnaam	Woonplaats	Geldig_van	Geldig_tot	Huidig
1	Jan	Amsterdam	1-1-2000	30-4-2006	N
1	Jan	Utrecht	1-5-2006	31-12-9999	J
2	Piet	Rotterdam	1-1-2000	31-12-9999	J
3	Klaas	Utrecht	1-1-2000	31-12-9999	J

Afbeelding 2: Historical Data Area.



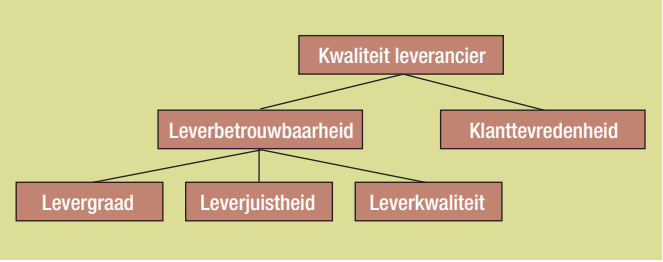
Afbeelding 3: Inspanning per architectuur.

biedt de mogelijkheid om generieker te modelleren, omdat het model niet vanuit gebruikersperspectief hoeft te worden ontworpen. Als het model generiek kan worden opgesteld, lijkt de oplossing daarmee flexibeler te worden.
Er kan bijvoorbeeld een generieke (supertype) tabel 'partij' worden ontworpen in het genormaliseerd centrale datawarehouse, waarbij alle 'partijen' (leveranciers, klanten, medewerkers, etcetera) kunnen worden vastgelegd. Eventuele nieuwe partijen (bijvoorbeeld managers) kunnen worden toegevoegd aan het datawarehouse zonder dat dit tot aanpassingen van het datamodel leidt. Dit soort generieke entiteiten wordt (al dan niet bewust) niet toegepast in dimensionele datawarehouses, omdat dit ten koste gaat van de eenvoud van het model (een belangrijke eis van het datawarehouse!).

Dit voorbeeld suggereert dat het voordelig is een generiek genormaliseerd model op te zetten, want het bedrijfsbrede datamodel is bestand tegen verandering en daarmee dus flexibeler. Dit is echter een misverstand, want het gaat niet om de flexibiliteit van het datamodel alleen, maar de flexibiliteit van de totale oplossing. Ofwel, de flexibiliteit om wijzigingen door te voeren in alle lagen van de architectuur. In bovenstaand voorbeeld kan het datawarehouse de wijziging eenvoudig opvangen, maar nog steeds moet in de vervolglaag (de dimensionele datamarts) de wijziging worden doorgevoerd. Want de belevingswereld van gebruikers is niet 'partij', maar bijvoorbeeld klant, leverancier, medewerker of manager. Ofwel, bij een generiek genormaliseerd datawarehouse wordt het probleem verschoven naar de datamart. Anders gezegd, bij een centraal genormaliseerd datawarehouse is het datawarehouse model wel flexibeler, maar dit geldt niet voor de totale oplossing, omdat een impact van een wijziging in zowel de generieke genormaliseerde laag als in de datamart in ogenschouw moet worden genomen. Hieruit blijkt dus dat flexibiliteit door middel van generiek modelleren geen toegevoegde waarde heeft. Daarmee kan dus geconcludeerd worden dat dit geen geldig argument is voor een genormaliseerd centraal datawarehouse. Deze conclusie is ook belangrijk bij de vraag of standaard gegevensmodellen die in de markt beschikbaar zijn ook als implementatiemodel kunnen worden gebruikt. Omdat deze standaardmodellen per definitie generiek moeten worden opgezet, blijken deze niet als implementatiemodel te voldoen, zie het kader 'Standaard datamodellen'.

Business rules

Het derde in de literatuur aangevoerde argument dat voor een bedrijfsbreed datawarehouse een genormaliseerd datamodel



Afbeelding 4: Hiërarchie Key Performance Indicatoren.

vereist is, is dat een dergelijk datawarehouse betrouwbaarder en consistenter is. De achterliggende argumentatie die hiervoor wordt gebruikt is dat meer business rules worden toegepast. Een vereenvoudigd voorbeeld hiervan staat in afbeelding 5. Het getoonde dimensionele model is gedenormaliseerd waardoor in het datamodel een aantal business rules ontbreekt, zoals:

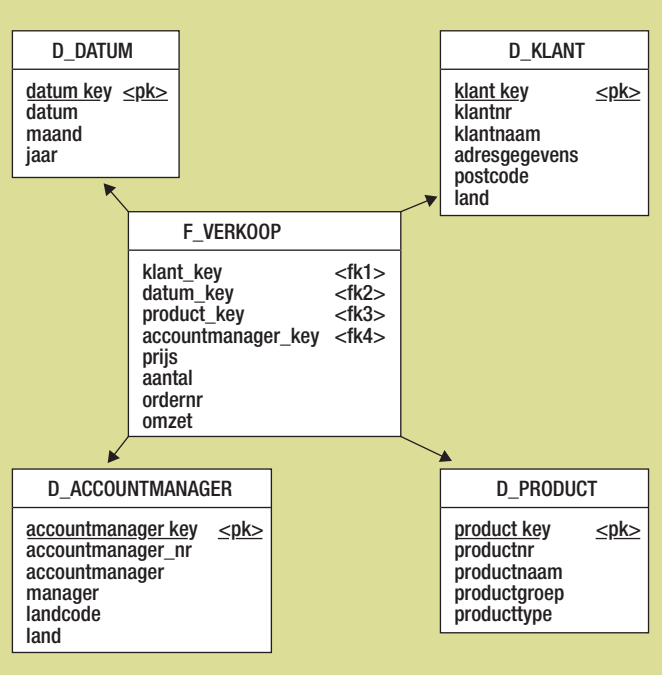
- de toets of de child/parent-relatie van de producthiërarchie consistent is;
- het controleren of de naamgeving van het land van de klant identiek is aan de naamgeving van de account manager;
- het afdwingen van een consistente relatie tussen order en orderregel.

Deze business rules kunnen wel consequent worden toegepast in een genormaliseerd model en dus niet in een dimensioneel datawarehouse. Het is dus een voordeel dat een genormaliseerd datawarehouse meer business rules kan afdwingen. Echter, het is te kort door de bocht om te stellen dat deze architectuur daarmee dus een betere oplossing is. Er is namelijk een eenvoudig alternatief, namelijk het vastleggen van de business rules in het ETL-proces. Hiermee kan ook de kwaliteit van de data worden gewaarborgd. Daarnaast kan het juist een bewuste keuze zijn om het datawarehouse een afspiegeling van de bron te laten zijn en dus deze business rules niet af te dwingen. Hierbij wordt het aloude GIGO-principe (garbage in = garbage out) gehanteerd. Dus ook dit derde argument is geen valide argument om te kiezen voor een genormaliseerd centraal datawarehouse. Anders gezegd, ook met een dimensioneel datawarehouse kan een bedrijfsbrede oplossing worden geïmplementeerd, maar dan tegen lagere kosten.

Standaard datamodellen

In de markt zijn in de loop der jaren standaard datamodellen voor datawarehousing beschikbaar gekomen. Deze datamodellen richten zich specifiek op een branche/domein (retail-model, banking-model, HR-model, etcetera) en zijn onafhankelijk van administratieve systemen, procesinrichting en informatiebehoefte opgezet. De vraag is of deze standaardmodellen met minimale aanpassingen kunnen worden geïmplementeerd als centraal datawarehouse-model. Dit blijkt in de praktijk echter geen goede keuze te zijn, omdat het kenmerk van deze modellen is dat ze zeer generiek zijn opgezet, zodat alle situaties 'passen' binnen dit model.

Het nadeel van deze generieke opzet van de modellen is dat dit ten koste gaat van de gebruiksvriendelijkheid, inzichtelijkheid en performance van de oplossing. Deze nadelen zullen zwaarder wegen dan het voordeel, waardoor het niet aan te bevelen is standaardmodellen fysiek te implementeren. De standaardmodellen kunnen echter wel veel toegevoegde waarde bieden als referentiemodel bij het ontwerpen van een centraal datawarehouse.



Afbeelding 5: Business Rules.

Tenslotte

Terugkijkend op het afgelopen decennium kunnen we concluderen dat op het vakgebied van datawarehousing goede, nuttige en bewezen oplossingen zijn ontstaan, maar ook 'ideologische verblinding' heeft plaatsgevonden. Ideologische verblinding is het niet toetsen van courante ideeën op nuttigheid, waarbij een patch wordt verward met een structurele oplossing. Bij dit fenomeen worden IT-oplossingen ontworpen omdat 'het nu eenmaal zo hoort', zonder kritisch stil te staan bij de achterliggende redenen van bepaalde constructieprincipes. Bij datawarehouses ontstaan hierdoor oplossingen met een nodeloos complexe gegevenslogistiek, wat ten koste gaat van flexibiliteit en leidt tot hoge ontwikkel-, beheer- en onderhoudskosten.
Dit fenomeen is treffend verwoord door René Veldwijk in zijn column 'Modulair verknipt' (DB/M 3-2006). Hierin geeft Veldwijk aan dat op database-gebied bepaalde ideeën tijdelijk zeer populair zijn, maar bij nader inzien schadelijk zijn als leidraad voor systeemontwerp. Het fenomeen ontstaat als gevolg van ideologische verblinding, waarbij het middel belangrijker wordt dan het doel. Om deze ideologische verblinding te voorkomen is het dus belangrijk het principe van 'minimaal aantal lagen' te hanteren. Ofwel, niet meer vragen of het een (opslag)laagje meer mag zijn, maar nagaan of het een (opslag)laagje minder kan!

Noot

1. Kimball's officiële definitie is dat de verzameling van logische geconformeerde datamarts het datawarehouse betreft. In de praktijk betekent dit veelal een implementatie van één fysieke database. Voor beide oplossingen is het datawarehouse direct benaderbaar voor gebruikers.

Frank Habers is architect en directeur bij Inergy Analytical Solutions, leverancier van oplossingen op het gebied van BI en datawarehousing.