Selectie database voor datawarehouse wordt steeds lastiger

Database-platformen versus ETL-tools

Alexander van Helm en Erik-Jan Koning

In dit artikel wordt de relatie tussen datawarehouses, ETL-tools en databases beschouwd. Het blijkt dat de verschillende producten steeds meer in elkaar overlopen, waardoor een nette vergelijking en selectie steeds moeilijker wordt. Database en ETL-tool bieden vaak veel gelijke functionaliteit.

Om dit toe te lichten wordt eerst gekeken hoe databases zich in de loop der tijd ontwikkeld hebben. Daarna wordt dieper ingegaan op de eisen die een datawarehouse aan een database stelt en wat het gebruik van ETL-tools voor gevolgen heeft voor de gewenste functionaliteit van onderliggende databases.

> Met 'Linked servers' is het mogelijk te linken naar andere databronnen

Een (relationeel) database-systeem is eigenlijk niets meer dan een poging om de relationele theorie van Date en Codd te implementeren. Oorspronkelijk is een SQL-database gewoon een verzameling tabellen, relaties en views om de tabellen te bekijken. Stored procedures, triggers, sequences en een groot aantal functies zijn een uitbreiding op het strikt noodzakelijke, net als gepartitioneerde tabellen, bitmap- en allerlei andere indices. Een database biedt dus inmiddels veel meer dan aanvankelijk de bedoeling was.

Bewegingen in de markt

De leveranciers ontwikkelen door en hun databases worden steeds veelomvattender en krijgen steeds meer functionaliteit. Enkele voorbeelden volgen. ETL-tools bieden de mogelijkheid om van verschillende bronnen de gegevens te halen. Als reactie hierop zien we dat databases ook deze functionaliteit aanbieden. Met 'Linked servers' is het mogelijk te linken naar andere databronnen. Wanneer dit gedaan is kun je deze bronnen benaderen vanuit de database-omgeving met 'gewone SQL'. Er kan replicatie ingezet worden om data vanuit diverse bronnen in de datawarehouse-omgeving te krijgen. Databases bieden vaker tools om bijvoorbeeld XML- of CSV-bestanden te importeren. Soms zelfs wordt SQL uitgebreid met nieuwe statements zoals CSVREAD (H2 database) of LOAD (DB2). Daarnaast zien we steeds vaker MERGE-statements verschijnen, die meer en meer aansluiten op de dimensie-update functionaliteit van ETL-tools.

Database-platformen bieden vaker scheduling services aan, daar waar ETL-tools dat al bieden. Vaak is er integratie met het mail-systeem en kan een e-mail gestuurd worden wanneer een batch mislukt.
De markt voor OLAP-tools was hoofdzakelijk in handen van de leveranciers die ook ETL-tools aanbieden. Daar komt verandering in. Database-platformen bieden meer en meer OLAP-functionaliteit aan. SQL Server heeft dit al langer, IBM en Oracle bieden dit nu ook in de vorm van DB2 UDB Data Warehouse Edition en Oracle 10g OLAP.
Databases kruipen steeds dichter naar de BI-markt. Zo biedt Microsoft SQL Server 2005 het Unified Dimensional Model aan, een metalaag over de database. Reporting Services en Analysis Services spreken deze metalaag aan. Hiermee is SQL Server meer gericht op de uiteindelijke presentatie van de data. Tot slot: met de opkomst van het web zijn er webservices gekomen. Er is complete integratie met Java of .NET.

De eindsituatie is dat de grote leveranciers eigenlijk geen databases meer leveren maar database-platformen, waar de scheidingslijn tussen ETL, BI en de eigenlijke database steeds meer vervaagt. Sommige leveranciers gaan nog verder. Bij Netezza schaf je een compleet systeem inclusief hardware aan: een datawarehouse in a box, of zoals ze het zelf noemen: een datawarehouse appliance.
Aan de andere zijde is te signaleren dat er aan de instapzijde van de markt veel ontwikkeld wordt. Zeer serieuze database-producten worden aangeboden tegen lage kosten, soms zelfs gratis. Denk aan SQL Server 2005 Express, Oracle Database 10g Express Edition en IBM DB2 UDB Express-C. Hoewel er veelal beperkingen kleven aan het aantal CPU's, het geheugen en/of de totale grootte van de database, zijn dit complete databases. Ze bieden niet alle functionaliteit van hun grote, betaalde broers, maar zijn wel gebaseerd op dezelfde, bewezen, robuuste techniek.

Welke eisen stellen datawarehouses aan databases?

Een datawarehouse bevat meerdere logische gegevensverzamelingen en is opgebouwd uit meerdere lagen, zoals de staging-laag, de historische laag en de presentatielaag, vaak in de vorm van meerdere datamarts. Er zijn referentiedata of masterdata, er zijn stuurgegevens, performance-gegevens en metadata. Al deze gegevens moeten ergens opgeslagen liggen. Het datawarehouse stelt eisen aan deze opslag, maar vooral ook eisen aan de toegang tot de data. Het mechanisme dat hiervoor wordt gebruikt, moet dit snel afhandelen, zelfs op veel data. Het is een keuze deze verzamelingen op te slaan in databases. Dit is tegenwoordig uiteraard wel de meest gangbare keuze (relationeel of niet-relationeel, zoals bijvoorbeeld SAS), want een database biedt beheersbaarheid, schaalbaarheid en performance. Het datawarehouse stelt dus eisen aan de opslag, performance en natuurlijk de beveiliging (autorisatie, backup) van de data. Alle overige eisen hangen samen met 'ETL-en'.

Grofweg zijn er twee manieren om te 'ETL-en': met zelf gecreëerde laadscripts, of met aangeschafte ETL-tools. Voordelen van het zelf bouwen zijn de uitsparing van licentiekosten en het feit dat je volledige controle hebt over de uiteindelijke code.

De code is echter database-specifiek, omdat de SQL-variant van de desbetreffende database wordt gebruikt. Groot voordeel van gekochte ETL-tools is dat de voorgedefinieerde logica kan worden gebruikt. Hierdoor kan de bouw sneller en gecontroleerder plaatsvinden. Bovendien is er vaak ondersteuning voor het werken in projecten.

ETL-tools zijn er in twee soorten; de ene groep heeft een eigen engine (Informatica PowerCenter, Business Objects Data Integrator, Cognos Data manager, etcetera), de andere groep genereert scripts die door de engine van de onderliggende database worden uitgevoerd (bijvoorbeeld Oracle Warehouse Builder, Sunopsis Data Conductor, WhereScape RED) en waarbij gebruik gemaakt wordt van database-specifieke faciliteiten zoals linked servers en bulk loads. Deze laatste groep ligt dicht tegen de databases aan en vormt in wezen een uitbreiding van de functionaliteit van databases.

Als de keuze op het zelf schrijven van laadscripts valt, is het handig dat een database-platform veel functionaliteit biedt omdat geen gebruik gemaakt kan worden van de functionaliteit van een ETL-tool. Wanneer een ETL-tool wordt gebruikt met een eigen engine, dan is er minder behoefte aan functionaliteit van de database. Wanneer gekozen is voor generatie-software, dan hangt het af van de eisen die de software stelt aan het specifieke databaseplatform waarvoor gegenereerd wordt.

Keuze

Sprekend over Oracle, SQL Server of bijvoorbeeld DB2, dan gaat het eigenlijk niet meer over databases maar over databaseplatformen. De scheidingslijn tussen de eigenlijke database en ETL en BI wordt steeds moeilijker te trekken. De vraag die de datawarehouse-deskundige zich moet stellen is wat is voor hem een database is. Luidt het antwoord "een opslag- en querymechanisme" en doet een onafhankelijke ETL-tool de bewerkingen, dan kan een relatief 'kale' database volstaan. Extra functionaliteit van de database is dan een nice-to-have: leuk meegenomen maar geen must. Sterker nog, als de database te uitgebreid is, is er een flink stuk overlap in de functionaliteit die het database-platform en de ETL-tool aanbieden, en mogelijk wordt ook dubbel betaald.

> Groot voordeel van gekochte ETL-tools is dat de voorgedefinieerde logica kan worden gebruikt

Kleinere datawarehouses of onderdelen zouden dan misschien ondergebracht kunnen worden in Express databases. Wordt in de toekomst tegen een van de grenzen van de Express edities aangegroeid, dan kan er altijd nog een upgrade plaatsvinden naar een betaalde editie. Verder spelen de open source databases een steeds grotere rol (MySQL, PostgreSQL, Firebird, etcetera) en zullen deze steeds meer gebruikt gaan worden voor datawarehouses.

Worden databases echter beschouwd als compleet platform, dan kan de keuze anders uitpakken. In dat geval maakt de functionaliteit die het platform aanbiedt deel uit van de uiteindelijke oplossing. Generatie-software maakt bijvoorbeeld gebruik van deze functionaliteit in plaats van zelf functionaliteit aan te bieden.

Conclusie

De boodschap is dus "wees je bewust van je denkkader" en "maak een bewuste keuze en geen vooringenomen keuze". Dat klinkt als een open deur, maar is het zeker niet. Zoekt men achteraf gezien een component maar is al een heel platform gekocht? Heroverweeg dan de keuze of maak gebruik van de extra componenten die het platform biedt. Er bestaat geen losstaande keuze voor een database, een ETL-tool of een BI-suite meer. Ze maken alle deel uit van de gehele ICT-architectuur en de keuze dient in die context gemaakt te worden. Men moet zich afvragen uit welke componenten de ICT-architectuur bestaat, welke componenten er nog ontbreken voor het datawarehouse en hoe die kunnen worden ingevuld. De invulling van die keuze kan weer effect hebben op eerder gekozen componenten. De mogelijkheden zijn groot – dat maakt de keuze wel lastig.

Alexander van Helm en Erik-Jan Koning

Alexander van Helm (alexander.van.helm@kadenza.nl) en Erik-Jan Koning (erik.jan.koning@kadenza.nl) zijn beiden werkzaam als BI&DWH Architect bij Kadenza.