

Rick van der Lans over trends in de database-wereld

Tussen verandering en vasthouden aan oude principes

Robbert Hoeffnagel



Alsof nog benadrukt moest worden dat de wereld van de databasetechnologie flink in beweging is, kondigde IBM onlangs zijn lang verwachte

plannen rond 'on demand computing' aan. Een mooie aanleiding voor Database Magazine om auteur, spreker en consultant Rick van der Lans te vragen de belangrijkste trends die hij momenteel rond databases ziet, op een rij te zetten. Hij vertelt over vijf onderwerpen, variërend van de opkomst van 'carefree databases' tot de vraag of XML eigenlijk wel in de database thuishoort.

Rick van der Lans volgt als auteur, spreker, congresvoorzitter, opleider en columnist op het gebied van datawarehousing, XML, software-ontwikkeling en databasetechnologie de ontwikkelingen in de softwarebranche op de voet. Hij is niet altijd even gelukkig met wat hij dan ziet. "Waar mensen als Ted Codd of Chris Date vroeger

Bij XML-documenten heb je niets aan SQL

tijdens congressen volle zalen trokken, zie je nu dat sprekers die de ontwikkelingen rond databases vooral wetenschappelijk

benaderen, nauwelijks meer aandacht krijgen. Wie de ontwikkelingen nauwgezet volgt, ziet ook dat steeds vaker de marketingafdelingen van de aanbieders lijken te bepalen in welke richting de ontwikkeling van een database-product zich beweegt. Dat is niet alleen jammer, maar bovendien ook niet altijd zonder gevaar. Kijk bijvoorbeeld maar eens naar wat er momenteel rond XML en databases gebeurt."

TREND 1: XML HOORT NIET IN DE DATABASE THUIS

Dat de 'eXtensible Markup Language' (XML) een belangrijke trend in de wereld van de databasetechnologie is, zal niemand ontgaan zijn. Maar hoe kunnen we XML in de database opslaan met behoud van de structuur van het XML-document? De oplossingen die aanbieders hiervoor tot nu toe hebben aangeboden, noemt Van der Lans nogal simplistisch. "Veel aanbieders zijn hier snel – ik denk iets te snel – op ingesprongen. Inmiddels hebben ze beter naar hun klanten geluisterd en zijn betere methoden in aantocht."

Daarmee gaan zij echter voorbij, zo meent hij, aan een vraag die tot nu toe nu slechts hier en daar te beluisteren valt: hoort XML eigenlijk wel in de database thuis? "Met andere woorden, maken we niet een ernstige conceptuele denkfout door XML in de database te willen opslaan?"

Van der Lans legt uit: "We kunnen twee soorten XML-documenten onderscheiden. De eerste soort bevat veel tekst en is, met

De vijf trends van Van der Lans

TREND 1: XML HOORT NIET IN DE DATABASE THUIS

TREND 2: WEBSERVICES VERDWIJNEN ACHTER DE DATABASE

TREND 3: 'CAREFREE DATABASE-SERVERS' HEBBEN DE TOEKOMST

TREND 4: ER KOMEN NIEUWE TYPES IN-MEMORY DATABASE-SERVERS DIE NAUW SAMENWERKEN MET KLASSIEKE DATABASES

TREND 5: DE OPSLAGSTRUCTUUR VAN DATA IN EEN RELATIONELE DATABASE GAAT VERANDEREN

andere woorden, ongestructureerde informatie. Het vormt bovendien de kleinste groep van XML-documenten. De tweede en grootste groep bestaat uit gestructureerde documenten die tot doel hebben gegevensuitwisseling mogelijk te maken. Dit laatste type XML-document hoort niet, zo is mijn stellige overtuiging, in de database thuis."

De reden hiervoor ligt besloten in de klassieke indeling in een gegevenslaag,

Het lijkt soms wel 'technology for technology's sake'

een applicatielaag en een presentatielaag. "Waar zit in dat model nu de laag waar gegevensuitwisseling plaatsvindt? Naar mijn mening vervangt de gegevensuitwisselingslaag de presentatielaag en heeft het niets te maken met de data-laag. Dus moet dit soort XML-documenten simpelweg niet in de database opgeslagen worden. Klopt het dus wel wat veel aanbieders van databaseproducten momenteel aan het doen zijn?"

De situatie ligt uiteraard anders, wanneer we het over de eerste groep van XML-documenten hebben. "Ging het bij de tweede groep over XML als een mechanisme voor het uitwisselen van gegevens, bij deze groep gaat het om XML als een hulpmiddel om tekstuele informatie te structureren. In dit laatste geval horen XML-documenten natuurlijk wel in de database thuis."

Deze discussie wordt momenteel niet of nauwelijks gevoerd en dat is typerend voor de hiervoor reeds genoemde verandering, die Van der Lans de afgelopen jaren in de wereld van de databasetechnologie heeft waargenomen. "Voorheen waren de belangrijkste ontwikkelingen rond de database altijd erg wetenschappelijk gedreven. Tegenwoordig is het veel meer de marketing, die de toon zet. Het lijkt soms wel 'technology for technology's sake'. Dat is naar mijn mening een gevaarlijke ontwikkeling waar we ver van zouden moeten blijven."



TREND 2: WEBSERVICES VERDWIJNEN ACHTER DE DATABASE

Een belangrijke trend is natuurlijk ook het fenomeen Webservices. Tot nu toe hebben de aanbieders zich bij de koppeling van databases aan webservices beperkt tot eenvoudige faciliteiten. "Een 'stored procedure' kan nu gewoon als een webservice worden aangeroepen. Centraal staat het aloude idee van het modulair programmeren, wat nog maar eens duidelijk maakt dat oude principes nog altijd prima te gebruiken zijn. Het is natuurlijk wel mooi dat we nu met een betere en meer complete 'stack' kunnen werken om die stored procedures aan te roepen. Het is daardoor mogelijk transacties te draaien over meerdere systemen heen, waarbij die applicaties in gebruik kunnen zijn binnen onze eigen organisatie, maar evenzogoed bij een andere firma. Als concept is dat allemaal prachtig. De fundamenten voor webservices worden echter geleverd door SOAP en – een beetje – UDDI, maar daarboven gebeurt uiteraard het meest interessante van webservices. Het vervelende is alleen dat we webservices moeten aanroepen via low-level interfaces. Je kunt je afvragen of we webservices daarom niet veel beter achter de database zouden moeten verstoppen?"

"Het aardige aan webservices, is de demo die we altijd van aanbieders te zien krijgen. Dan wordt bijvoorbeeld de prijs van een cd opgevraagd via een SOAP-call. Maar waarom doen we dat eigenlijk niet gewoon via SQL, ook al staat de database waarin de prijs van die cd vastligt, opgesteld bij een ander bedrijf? Dan kunnen we bovendien heel eenvoudig alle prijzen van alle cd's van die ene artiest opvragen."

Een aantal aanbieders is momenteel bezig webservices min of meer achter de database te verstoppen. "De technologie die hiervoor nodig is, zit vaak al in de database-server. Want laten we wel wezen, een tabel hoeft natuurlijk niet altijd een

Waarom laten we oude principes los, die als belangrijke stappen vooruit werden ervaren?

tabel te zijn. Het is natuurlijk ook nu al heel goed mogelijk dat wat een bevraging van een tabel lijkt te zijn, in werkelijkheid door de database-server wordt doorge-sluist naar een andere applicatie. Net als we nu al OLAP-kubussen achter tabellen plaatsen."

Net als bij het omgaan met XML-documenten ziet Van der Lans ook bij dit



onderwerp meer fundamentele problemen. "Waarom laten we allerlei oude principes zomaar ineens los, die tot nu toe juist als belangrijke stappen vooruit werden ervaren? Enterprise Java Beans (EJB), ADO.Net, noem ze maar op: het zijn alle-

Misschien gaat de behoefte aan DBA's afnemen

maal low-level hulpmiddelen. Voorheen probeerden we juist naar een steeds hoger niveau toe te werken, waarbij de low-level aspecten door software werden geregeld. Het lijkt wel of we dat nu juist allemaal weer zelf willen gaan doen. Maar waarom eigenlijk? Is dat een gevolg van de opkomst van Java, waar we ook alles zelf willen kunnen regelen?"

Er is ook een enigszins cynisch antwoord mogelijk, maar daarin wil Van der Lans echter niet geloven. "Natuurlijk zou het waar kunnen zijn, dat dit soort ontwikke-

lingen vooral ingegeven wordt door aanbieders die op deze manier meer geld denken te kunnen verdienen, maar eigenlijk wil ik daar niet aan. Wat er dan wel aan de hand is, weet ik ook niet. Heeft het te maken met de kwaliteit of de manier van opleiden? Wat is er gebeurd met principes als declaratief programmeren of het gebruiken van een zo hoog mogelijke programmeertaal? Waar is het uitgangspunt gebleven dat integriteitsregels in de database geïntegreerd dienen te worden?" Vragen waarop van der Lans geen antwoord heeft.

TREND 3: 'CAREFREE DATABASE-SERVERS' HEBBEN DE TOEKOMST

"We zetten tegenwoordig zomaar ergens een database neer, in een omgeving waar niemand iets van deze technologie en deze producten afweet. Vroeger zouden we het niet in ons hoofd halen om een database buiten het rekencentrum te plaatsen. Dat kan alleen wanneer we

daadwerkelijk met 'carefree' database-servers kunnen gaan werken. De ontwikkeling van dat soort producten komt steeds meer op gang."

Er speelt echter meer rond het beheer van databases. "We hebben de laatste jaren heel wat nieuwe faciliteiten tot onze beschikking gekregen om databases optimaal te laten functioneren wat betreft snelheid, beschikbaarheid, noem maar op. Het gaat daarbij om tal van hele nuttige hulpmiddelen, die echter vaak wel één belangrijk nadeel kennen: we hebben mensen nodig om al die functionaliteit ook daadwerkelijk te kunnen gebruiken en steeds meer bedrijven kunnen dat simpelweg niet meer opbrengen."

Uiteraard zien ook de aanbieders van database-producten deze ontwikkeling gebeuren. Daarom wordt steeds meer tijd en energie gestoken in het ontwikkelen van functionaliteit die het mogelijk maakt dat een database-server zichzelf instelt en optimaliseert. "Het gaat dan om AI-achti-

We zoeken in toenemende mate via OLAP-technieken

ge technieken die in een database-server worden ingebouwd, zodat deze zichzelf kan instellen aan de hand van de kenmerken van de omgeving waarin gefunctioneerd moet worden. Sommige kenmerken geeft de database administrator op, andere worden zelf gemeten."

Het zal uiteindelijk een verlichting betekenen voor de database administrator.

"Misschien betekent het zelfs wel dat de behoefte aan DBA's gaat afnemen," meent Van der Lans. "Natuurlijk zullen er altijd parameters door de DBA zelf ingesteld moeten worden. Maar op basis van een reeks van standaardinstellingen en de genoemde AI-technieken kan de database-server, aan de hand van bijvoorbeeld workloads en dergelijke, bufferruimtes en dat soort zaken instellen. De DBA zal in dit soort situaties vooral de range aangeven waarbinnen de AI dient te blijven."

We zien momenteel reeds de eerste voorzichtige stappen in deze richting gezet

worden. "Kijk bijvoorbeeld eens naar database-servers waarbij de statistische informatie door de software zelf wordt bijgewerkt, zonder dat de DBA hier iets aan hoeft te doen. Het is naar mijn mening ook de enige manier waarop de aanbieders de

Met de opkomst van Java lukte het OO weer niet om een doorbraak te forceren

'Total Cost of Ownership' (TCO) van hun producten omlaag kunnen brengen."

Als voorbeeld van een carefree database-server, die zelf reeds een aantal beheertaken voor rekening neemt, noemt Van der Lans Solid, een database die in mobiele telefoons kan worden opgenomen (www.solidtech.com).

TREND 4: ER KOMEN NIEUWE TYPES IN-MEMORY DATABASE-SERVERS DIE NAUW SAMENWERKEN MET KLASSIEKE DATABASES

De klassieke database-server stamt nog uit de tijd dat I/O duur was en het werkgeheugen van een server klein. "De hardware waarmee destijds werd gewerkt, lijkt echter niet meer op de systemen waarmee we tegenwoordig aan de slag kunnen," zegt Van der Lans. "Wordt het niet tijd dat we overstappen op een aanpak waarbij het gehele 'hote' deel van de database altijd in het werkgeheugen is geplaatst? Met andere woorden: werken met een in-memory of main-memory database? Ik denk dat we steeds meer die kant uit gaan. Maar dan niet met dezelfde aanpak die we nu volgen. Een database gebruiken gebeurt momenteel eigenlijk heel inefficiënt. We kopiëren nu immers een opslagstructuur naar het werkgeheugen, die is geoptimaliseerd voor het werken met disks en het daardoor onvermijdelijke dataverkeer tussen werkgeheugen en schijf. Het is veel beter wanneer er voor in-memory databases nieuwe opslagstructuren komen, die zijn ontwikkeld met de

kenmerken van een werkgeheugen in het achterhoofd."

De eerste producten waarin dit reeds te zien is, zijn inmiddels op de markt verschenen. "Kijk bijvoorbeeld naar TimesTen van de gelijknamige firma (www.timesten.com) of PolyHedra van OSE Systems (www.polyhedra.com). Dit zijn pure softwareproducten. We zien ook al combinaties van hardware en software die voor hetzelfde doel zijn ontwikkeld: DataBlitz van Lucent (ook wel het Dali-project genoemd, www.bell-labs.com/project/dali) bijvoorbeeld of SuperSelect van Bull www.bull.com/servers/superselect/home/accueil_eng.htm). De eerste gebruikers zijn vooral in de telecom-wereld te vinden."

Het is naar de mening van Van der Lans echter niet zo dat in-memory of main-memory databases de klassieke database gaan vervangen. Er zal eerder sprake zijn van een hiërarchie van database-servers. "De in-memory database wordt daarbij boven de klassieke database geplaatst. Alle transacties worden opgevangen in

main-memory en worden daar razendsnel verwerkt. Vervolgens druppelt die data door, richting de klassieke database en disk." Er doet zich echter wel een probleem voor. "Deze manier van werken is lastig in een klassieke database in te passen. Het vereist nogal ingrijpende veranderingen. Nog niet erg duidelijk is hoe de aanbieders van klassieke database-servers daarmee om zullen gaan. We zien wel de eerste bewegingen ontstaan, kijk maar naar Sun Microsystems dat de aanbieder van clusteringtechnologie Clustra Systems heeft overgenomen. Maar waar ik met name benieuwd naar ben, is wat IBM, Oracle en Microsoft op dit gebied gaan doen."

TREND 5: DE OPSLAGSTRUCTUUR VAN DATA IN EEN RELATIONELE DATABASE GAAT VERANDEREN

"Wanneer we nu een tabel tekenen, dan ziet die er ook uit als een tabel. Kijken we naar de manier waarop zo'n tabel op schijf



FOTO: WIM VREEBURG, FOTOPPOINT

wordt vastgelegd, dan zien we wederom een aanpak die veel weg heeft van een tabelstructuur. Dat ging lange tijd goed, maar nu niet meer. We zoeken namelijk in toenemende mate op een andere manier, bijvoorbeeld via online analytical processing (OLAP) technieken. Hierdoor kunnen in een omgeving waarin data in tabellen vastligt, flinke wachtrijen en -tijden ontstaan omdat een traditionele relationele database niet goed met die manier van zoeken uit de voeten kan. Dat betekent dus dat de manier waarop we data vastleggen zal moeten veranderen."

Ik hoor ook vrijwel nooit vragen stellen over Xquery

Dit is een trend die duidelijke raakvlakken vertoont met wat Van der Lans eerder over webservices vertelde. "De oplossing die we nu zien ontstaan, is het 'verstoppen' van een OLAP-kubus achter een tabel. Dat is een vergelijkbare aanpak met waar we het eerder bij webservices over hadden. Achter een tabel gaat logica schuil, die bijvoorbeeld aangeeft dat de gegevens die de gebruiker eigenlijk had verwacht in de tabel tegen te komen, in werkelijkheid opgehaald dienen te worden uit een andere database of via een applicatie moeten worden berekend." Zoiets gebeurt nu dus ook al door OLAP-engines achter relationele tabellen te plaatsen. Het lijkt dus een tabel maar het is in werkelijkheid een kubus. Van der Lans gebruikt hiervoor termen als 'virtualisatie' of 'abstractie'.

KLEINERE TRENDS

Praten met Rick van der Lans levert een grote hoeveelheid informatie op over de ontwikkelingen in de wereld van de database-technologie. Hierbinnen vijf trends aangeven is geen probleem, maar het getal 'vijf' blijkt al snel discutabel te zijn. "Er gebeurt zoveel dat we ook wel tien trends zouden kunnen benoemen," meent Van der Lans, "al zullen die dan

wellicht niet allemaal even belangrijk zijn." Daarom nog enkele kleinere trends. De eerste is een trend die eigenlijk geen trend meer is: objectgeoriënteerde (OO) databases. "Is er een toekomst voor dit soort producten? Ik denk het niet. Er is hooguit in enkele kleine niche-markten een plek voor dit soort databases te vinden."

Wat is er dan fout gegaan met OO-databases, die toch lange tijd als veelbelovend te boek stonden? "Het concept is simpelweg niet aangeslagen," meent Van der Lans. "Begin jaren negentig hebben de aanbieders van dit soort producten een goede kans gekregen om door te breken. Ze hadden alle 'spotlights' op zich gericht staan. Toen bleek echter dat het gehele concept nog onvoldoende was uitgewerkt. Denk aan backup & recovery, waar men gewoon niet genoeg over had nagedacht. Later, met de opkomst van Java, kwam een tweede kans voor de OO-databases, maar weer lukte het niet om een doorbraak te forceren. IBM, Oracle en Microsoft hadden de database-markt toen al te stevig in handen om zo'n belangrijke ommezwaai toe te staan."

Daar komt nog een andere factor bij. "De houding van de markt – ook die van de gebruikers – is toch dat je SQL nodig hebt wanneer je een database gebruikt. Daar stapt men niet zomaar van af, ook niet wanneer zich een nieuwe ontwikkeling als OO-databases voordoet. Dat zijn cruciale beslissingen voor bedrijven en veranderingen daarin hebben verstrekken de gevolgen."

XQUERY GEEN BEDREIGING VOOR SQL

Het brengt Van der Lans op een vraag die hier direct verband mee houdt: hoe lang doen bedrijven eigenlijk met een database? De vraag leidt tot de nodige discussie, maar niet tot een duidelijk antwoord. Voortbordurend op die discussie stelt Van der Lans bovendien vast dat de 'XML Query Language' (Xquery) - waarvan momenteel versie 1.0 gereed is - in zijn visie geen bedreiging is voor SQL. "De reden daarvoor is in mijn ogen heel duidelijke:

SQL is een 'commodity' geworden. Iedereen gebruikt het. Overstappen op een alternatief als Xquery zou een enorme ommezwaai betekenen. Wanneer ik een voordracht houd of een seminar geef, hoor ik ook vrijwel nooit vragen stellen over Xquery. En als het een keer gebeurt, dan is het door iemand uit de wereld van documentmanagement, waar XML natuurlijk vooral als een broertje van SGML wordt gezien."

Heeft Xquery daarmee geen toekomst? Toch wel, vindt Van der Lans. "Verwacht echter niet op korte termijn belangrijke verschuivingen. De database is zo fundamenteel voor organisaties dat veranderingen op dit terrein een erg lange adem vereisen. Kijk ook naar de plannen die aanbieders ontvouwen: de eerste implementaties van Xquery worden pas over drie jaar verwacht. Misschien nog wel later."

Wat natuurlijk momenteel ook niet in het voordeel van Xquery werkt, is het feit

We willen steeds vaker dingen doen met SQL waarvoor het niet bedoeld is

dat de ICT-budgetten bij gebruikersorganisaties flink onder druk staan en ook de aanbieders niet erg ruim in hun geld zitten. Dit soort nieuwe technologieën met verstrekkende technische en financiële gevolgen heeft hierdoor niet direct de hoogste prioriteit, noch bij gebruikers noch bij aanbieders.

Het is echter wel degelijk een feit dat Xquery voordelen kan bieden boven SQL. "Zie Xquery maar als een optelsom van de functionaliteiten van SQL, de 'Object Query Language' (OQL) en XQL, de 'XML Query Language'. Bij data die vastligt in XML-documenten heb je niets aan SQL, dan heb je een ander hulpmiddel nodig. Wat dat betreft is de rek wel uit SQL. We willen steeds vaker dingen met SQL doen waarvoor het eigenlijk helemaal niet bedoeld is." ●

Robbert Hoeffnagel is freelance journalist