VAN DER LANS

Houd logisch en fysiek gescheiden

Zo'n dertig jaar geleden is mij in het researchlaboratorium van Control Data in Brussel, waar toen onder leiding van Professor Nijssen NIAM ontwikkeld werd, geleerd dat als een database ontworpen moet worden, dit in minimaal twee stappen moet gebeuren.

Tijdens de eerste stap creëren we een model dat zich richt op conceptuele of noem het logische aspecten. Dit leidt tot een niet-technische specificatie van de data die opgeslagen moeten worden. Voor deze stap worden in de literatuur verschillende termen gehanteerd, zoals logisch datamodelleren, informatiemodellering en conceptueel datamodellering. Voor de duidelijkheid gebruiken we in deze column de eerste term. Om deze stap goed te kunnen uitvoeren, is veel business kennis nodig. Technische kennis is nagenoeg irrelevant. Het doel is een model te creëren dat onder andere duidelijk aangeeft welke data-elementen er zijn, wat hun onderlinge relaties zijn en wat hun respectievelijke definities zijn. Ook het identificeren van formules om bepaalde waarden te berekenen behoort tot deze stap, zoals: wat is precies de formule voor het berekenen van jaaromzet.
In de tweede stap gaan we broeden op hoe we dit logische model het beste kunnen implementeren met behulp van een bepaalde databasetechnologie. Dit is de fase waarin we ons buigen over zaken als performance van query's, snelheid van laden, toegangssnelheid en opslag. Laten we deze stap betitelen met de term fysiek databasemodelleren, al worden in de literatuur ook hier veel verschillende termen voor gehanteerd. Deze stap vereist absoluut veel kennis van de databasetechnologie die ingezet zal worden en is daar ook sterk afhankelijk van.
Of een database geïmplementeerd moet worden met klassieke relationele databasetechnologie, een moderne datawarehouse appliance of een multidimensionale database server, dat bepaalt in grote mate hoe het fysieke databasemodel er zal uitzien. Business kennis is tijdens deze stap van ondergeschikt belang. Niets nieuws zult u zeggen. En dit hoort ook niets nieuws te zijn. Volgens mij werken veel analisten en ontwerpers op deze manier. Sommigen maken er zelfs drie of vier stappen van. Toch zie ik in menig datawarehouseproject dat we die twee stappen niet altijd duidelijk uit elkaar houden. Ook in sommige artikelen en boeken wordt deze scheiding niet altijd aangehouden. Bijvoorbeeld, als we het databasemodel van een datawarehouse ontwerpen dat ontwikkeld zal worden met relationele databasetechnologie, kunnen we globaal uit vier verschillende oplossingen kiezen. We kunnen voor een volledig genormaliseerde databasestructuur kiezen, we kunnen onze databasestructuur een sterschema-ontwerp geven, we kunnen voor een sneeuwvlokmodel kiezen, of we gaan voor de wat modernere Data Vault oplossing. Er zijn trouwens nog meer alternatieven, maar we beperken ons in deze column tot deze vier. Maar is de keuze voor een van deze vier oplossingen nu een aspect van logisch datamodelleren of van fysiek databasemodelleren? Vaak wordt deze keuze geheel onterecht tijdens logisch datamodelleren bepaald. Of we een sterschema of een sneeuwvlok kiezen heeft niets te maken met het logische datamodelleren waar we alleen proberen te doorgronden wat de structuur van de gegevens is en wat de informatiebehoeften zijn. Deze keuze is voornamelijk een technische. Het heeft te maken met de gewenste performance van query's, de benodigde opslag en de flexibiliteit van de databasestructuur. Omdat het dus een aspect is van fysiek databasemodelleren hoort het besproken te worden door de databasespecialisten die veel afweten van hoe de database server intern werkt. Het hoort geen onderwerp te zijn voor analisten die zich bezig houden met het logisch modelleren en voornamelijk met gebruikers in gesprek zijn.

De keuze van de techniek is ook niet onafhankelijk van de ingezette databasetechnologie. Als we bijvoorbeeld een multidimensionale database gaan gebruiken, waarbij alle data in kubussen opgeslagen worden, zullen alternatieven als Data Vault en normaliseren niet echt relevant zijn. Een ander voorbeeld is dat als bekend is dat een snelle datawarehouse appliance ingezet zal worden, dat dan de beoogde performancewinst van een sterschema minder zal zijn dan bij een klassieke database server. Kortom, deze technieken horen echt thuis in de wereld van fysiek databasemodelleren.
Toch zien we te vaak bij het opzetten van een logisch model dat er al bijvoorbeeld een keuze voor een sterschema of Data Vault gemaakt wordt. Dit is onverstandig. Uiteraard realiseer ik mij dat we het nooit voor elkaar krijgen de twee stappen volledig te scheiden, de stappen horen nu eenmaal bij elkaar. Maar het dient wel een doelstelling te zijn om ze zoveel mogelijk gescheiden te houden. Het degelijk uitvoeren van logisch datamodelleren mag niet worden vervuild met typische aspecten die behoren bij fysiek databasemodelleren. Logisch datamodelleren is daarvoor te belangrijk.

Rick van der Lans is zelfstandig IT-consultant.