

OLTP en OLAP met inkomensgegevens

## De Polis Papers (5): Drill me up, Scotty!

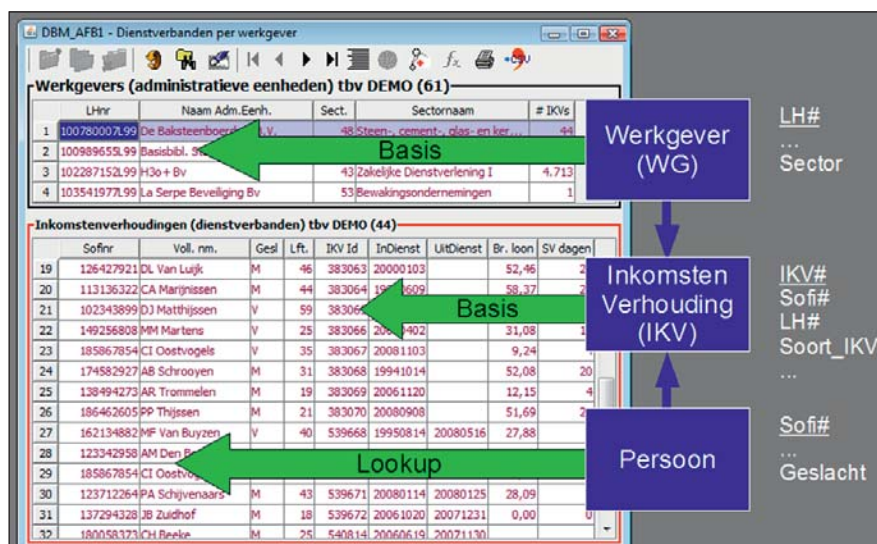
René Veldwijk en Robert Veldwijk

Na vier artikelen over de architectuur van de Polisadministratie (PA) richten we nu de blik op het interactief werken met de PA-gegevens. We hebben het dan over High-T, de modelgedreven front-end omgeving waarop de PA is gebaseerd. High-T verenigt de klassieke administratieve denkwijze over schermen (OLTP) en online analytical processing (OLAP) in één paradigma. Wij vragen uw aandacht voor een compleet nieuwe manier om aan te kijken tegen interactief gegevensgebruik en misschien een einde aan de gangbare aanpak van Business Intelligence. Zie ook de filmpjes op [www.youtube.com](http://www.youtube.com), zoektermen DBM FAAPartners High\_T PolisPapers.

In deze artikelserie vindt u wellicht een patroon. In alle vier voorgaande artikelen ging het over het opknippen van complexe functionaliteit in vrij combineerbare dimensies. In artikel 2 ging het over het omgaan met datakwaliteitsproblemen: de Good-Bad-Ugly lijn. Daarvoor besprak artikel 1 de omgang met tijdaspecten op basis van twee onafhankelijke tijdsdimensies: Geldige Tijd en Transactietijd. In artikel 3 rafelden we het autorisatieprobleem uiteen in drie onafhankelijke dimensies: Gegevens, Services en Afnemerindicaties. En in het laatste arti-

kel over logging betraden we zelfs de niet te filmen wereld van de vierde en de vijfde dimensie. Cruciaal is dat al die dimensies onderling combineerbaar zijn. Lukt dat dan is het resultaat een goedkoop, robuust en tegelijk flexibel systeem. Wat ons betreft is architectuur zoveel mogelijk functionaliteit peuren uit zo weinig mogelijk elkaar aanvullende concepten. Wiskundigen noemen dat *orthogonaliteit*, maar je zou het ook *informatica-economie* kunnen noemen. En zo werkt het ook echt: de PA zoals die hier wordt beschreven is gerealiseerd voor een bedrag dat nog niet de helft is van het bedrag waarvoor de oorspronkelijke PA was aanbesteed. Zo belangrijk is architectuur.

De architectuur van de PA is voor een groot deel gebaseerd op High-T, een met Java gebouwd modelgedreven ontwikkelplatform uit eigen stal. De basis van High-T en van elk met High-T gerealiseerd systeem is de zelfbeschrijvende datadictionary, waarvan de opzet lang geleden in DB/M is gepubliceerd in de 10 Geboden serie over database-ontwerp (te downloaden op [www.dbm.nl](http://www.dbm.nl)). Deze dictionary is het startpunt voor de ontwikkeling van flexibele software onder een open, orthogonale architectuur. Het ideaal daarbij is een extreem modelgedreven, maximaal declaratieve werkwijze. In leken termen: niet programmeren maar metadata aanklikken of intikken. Nergens wordt dat ideaal zo dicht benaderd als bij de ontwikkeling van schermen. De circa 150 schermen die voor de PA zijn ontwikkeld zijn niet geprogrammeerd maar in elkaar geklikt. In dit artikel behandelen we



Afbeelding 1: Van datamodel naar scherm.

Werkgevers (administratieve eenheden) tbv DEMO (61)				
LHNr	Naam Adm.Eenh.	Sect.	Sectornaam	# IKVs
6	106314772L99	40	Uitgeverij	1
7	108060568L99	45	Zakelijke Dienstverlening III	65
8	110416284L99	54	Culturele instellingen	109
9	111478270L99	62	Overheid, rijk, politie en recht...	34
10	112950899L99	43	Zakelijke Dienstverlening I	8
11	113812393L99	17	Detailhandel en aanbachten	1

Inkomstenverhoudingen (dienstverbanden) tbv DEMO (10)									
LHNr	Sofinr	Voll. nm.	Gesl	Lft.	IKV Id	InDienst	UitDienst	(On)Bep.	SV dagen
1	112950899L99	153660756	J Dupuis	V	31	316410	19990101	O	
2	112950899L99	171239248	HS Marcic	V	37	316412	19990101	O	14
3	112950899L99	164302902	AM Motos	V	38	316414	19990101	O	18
4	112950899L99	136106385	FM Meijer	V	27	317467	20070110	20070411	O
5	112950899L99	177969246	EA Kleuters	V	53	325971	20070223	20080821	O
6	112950899L99	160151228	M Baartz	V	19	385947	20080904		O
7	112950899L99	171862071	ML Schat	V	18	385948	20081007		O
8	106314772L99	154845586	T Test	M	18	498210	20000101	20090930	O
9	112950899L99	145767700	HR Vellinga	V	20	539954	20080118	20080627	O
10	113812393L99	148712054	T Test	M	18	730810	20000101		O

**Afbeelding 2:** Een MultiCurrent (MuCu) Master/Detailscherm.

die schermen, echter niet vanuit het perspectief van de 'programmeur' maar vanuit de eindgebruiker. De eindgebruiker die naar de PA kijkt ziet, jawel, onafhankelijke dimensies. Hoeveel dimensies? In beginsel zoveel als hij of zij nodig heeft. We gaan proberen om dat in één artikel uiteen te zetten. Omdat we moeten woekeren met ruimte verwijzen we daarbij naar de artikelen 1 en 2 in DB/M 1 en 2 van dit jaar, ook te downloaden van [www.dbm.nl](http://www.dbm.nl). We beginnen echter met een recapitulatie van het huidige OLTP paradigma.

## Terug naar af: de master/detail

Lang geleden kwam Ted Codd ongeveer tegelijkertijd met het idee van database-normalisatie en met het relationele gegevensmodel. Normalisatie is een manier om gegevensstructuren te disciplineren en wordt bijkans afgedwongen door relationele DBMS'en – althans voor transactiedatabases (OLTP). Met de komst van gedisciplineerd opgezette relationele databases werd het mogelijk om slimme tools te ontwikkelen om gegevens te presenteren en te manipuleren: de schermgenerator (en de rapportgenerator) was geboren. Het basisidee was om schermen op te bouwen uit gegevensblokken die steeds een afbeelding zijn van een onderliggende database tabel of view. Tabellen (en views) zijn fundamenteel aan elkaar gerelateerd via constraints die we aanduiden als verwijzende sleutels. En laat nu in bijna alle gevallen de behoefte om gegevensblokken aan elkaar te relateren samenvallen met de beschikbare verwijzende sleutels. Zo wordt het mogelijk om uitgaande van een gegevensmodel (modelgedreven) en zonder te programmeren (declaratief) redelijk complexe schermen te maken.

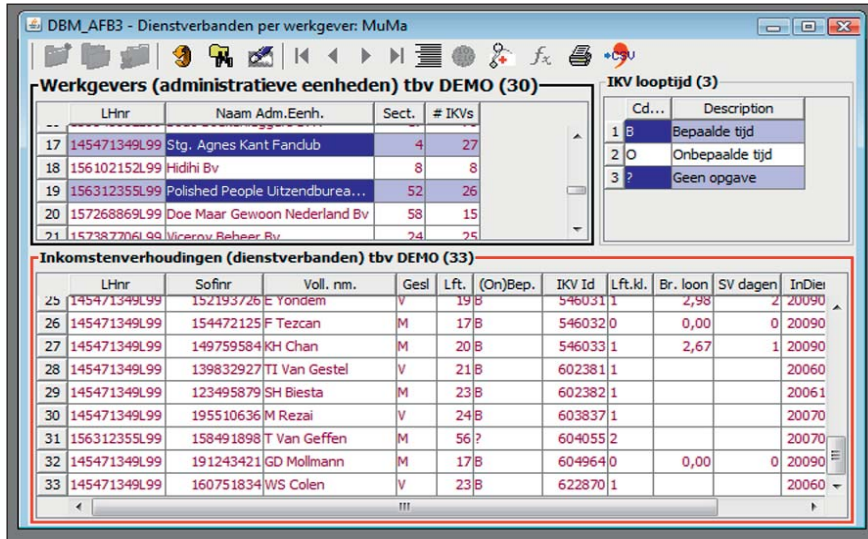
Afbeelding 1 toont een voorbeeld van een PA-scherm. We zien een scherm met twee gegevensblokken, één gebaseerd op de tabel/view Werkgever en één op de tabel/view Inkomstenverhouding. We selecteren één werkgever en we krijgen van die werkgever de inkomstenverhoudingen (de *details*) te zien – hele-

maal volgens de regel dat een inkomstenverhouding toebehoort aan één werkgever maar dat één werkgever best meer inkomstenverhoudingen mag hebben. We zien overigens ook dat in de detailgegevens uit de tabel Persoon worden getoond. Ook dat gebeurt weer met behulp van een verwijzende sleutel. De database tabellen Inkomstenverhouding en Persoon worden in het detail-gegevensblok gedenormaliseerd tot een gecombineerd gegevensblok waarvan de records nog steeds corresponderen met inkomstenverhoudingen. Op basis van een fatsoenlijke data-dictionary en wat slimme grafische programmatuur kan zelfs een eindgebruiker een scherm ontwikkelen. Tot zover niets nieuws.

## Twee innovaties: MuCu en MuMa!

Op weg naar een nieuw paradigma zetten we eerst een schijnbaar kleine maar fundamentele stap: we nemen afscheid van het idee dat gegevens die in een *detail*-gegevensblok staan, alle betrekking hebben op één record in het *master*-gegevensblok. We introduceren de *MultiCurrent Master* – zie afbeelding 2. Een MuCu master is een extreem handig construct en wie ermee heeft gewerkt kan moeilijk wennen aan single-current masters. In afbeelding 2 zien we in één oogopslag wie er bij drie geselecteerde bedrijven werken. Zodra er meer dan één record in de master wordt geselecteerd, tonen we dynamisch de identificatie van de master (kolom LHNr) zodat de eindgebruiker kan zien wie waar werkt. Als er in dit scherm updates zouden zijn toegestaan, dan kan een inkomstenverhouding ook worden 'omgehangen' naar een andere werkgever.

MuCu is handig en prettig maar op zich niet revolutionair. Er is echter nóg een vreemde conventie in de klassieke schermenwereld. Behalve dat er nooit meer dan één master record actief mag zijn, valt een detail-gegevensblok altijd onder één master-gegevensblok. Waarom eigenlijk? Met High-T laten we ook deze beperking varen: wij maken de relatie tussen gegevensblokken veel-op-veel. Afbeelding 3 geeft een voorbeeld van zo'n



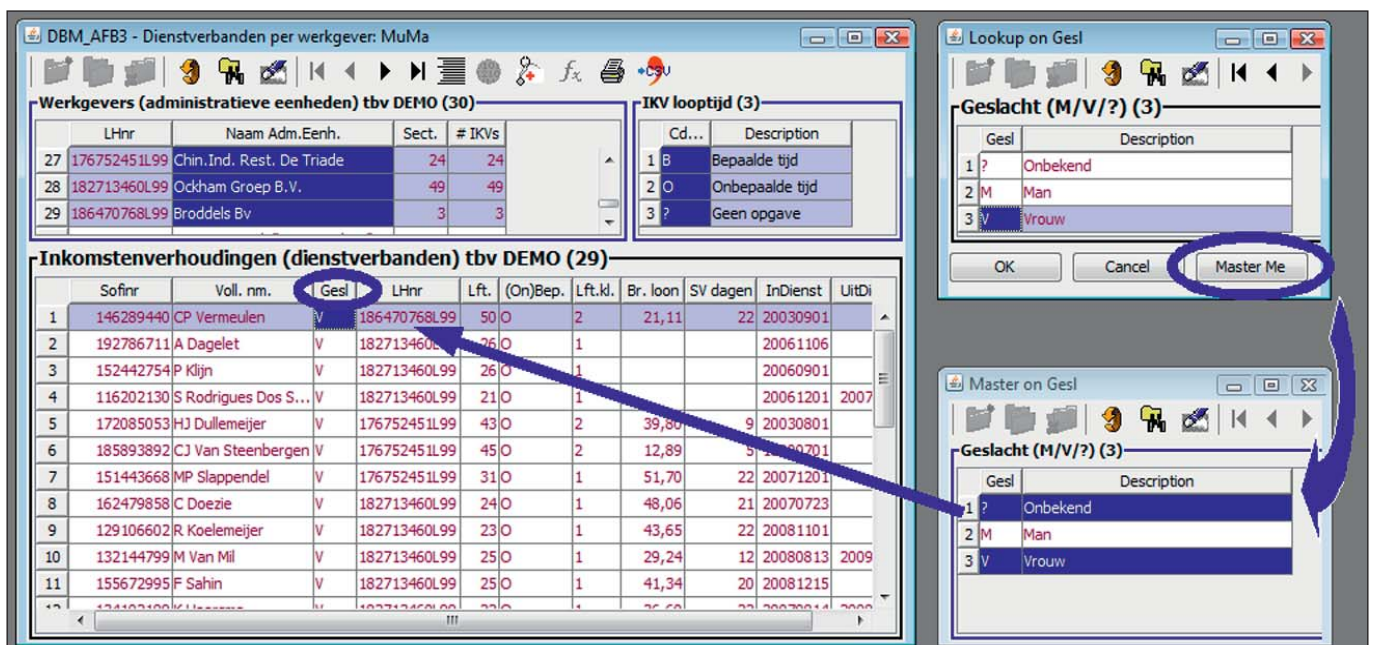
**Afbeelding 3:** Een MultiMaster (MuMa) Master/Detailscherm.

MultiMaster scherm. We zien nu alleen die inkomstenverhoudingen van de geselecteerde werkgevers voorzover die niet voor onbepaalde tijd zijn aangegaan. We kunnen zowel in de werkgever master als in de 'looptijd' master bladeren en de detail loopt mee. Uiteraard staat of valt het gebruik van een MultiMaster bij de MuCu-functionaliteit. Zonder MuCu kunnen we bijvoorbeeld niet in één detail-gegevensblok meerdere soorten dienstverbanden selecteren. Met MuCu én MuMa gecombineerd maken we een enorme sprong voorwaarts in het empoweren van eindgebruikers. Bedenk daarbij dat het niet bij twee masters hoeft te blijven: drie of meer masters mogen ook. Afbeelding 1 van artikel 1 (zie DB/M 1) toonde al een detail met drie masters: loonbedragen per inkomstenverhouding, per tijdvak en per mutatiemoment. Wellicht denkt u dat MuCu-MuMa schermen niet iets zijn voor 'gewone eindgebruikers', maar iets

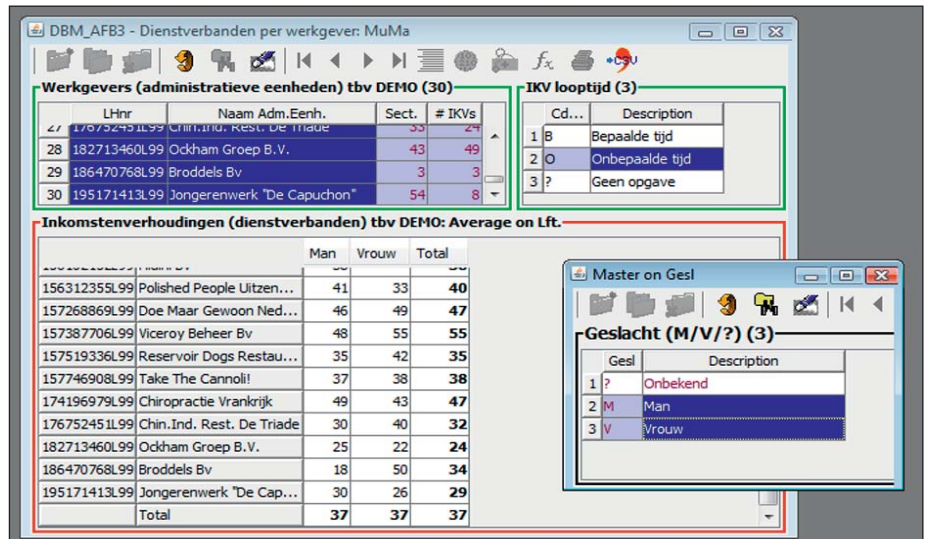
voor analisten en kenniswerkers. Niet dus. Wie één keer heen en weer in de tijd is gereisd met het inkomensscherm van artikel 1 weet beter. Daarbij komt ook dat secundaire masters zoals de 'Looptijd-master' van afbeelding 3 op een ander tabblad worden geplaatst, zodat er voor gewoon gebruik schijnbaar sprake is van een klassieke master/detail. Het scherm van afbeelding 3 oogt dan voor de argeloze gebruiker weer als afbeelding 2, terwijl de echte kenniswerker als het zo uitkomt haar eigen parametrisering verzorgt. Programmeurs zijn exit.

## MuMa on the fly: MasterMe!

Vermoedelijk is het u opgevallen dat de 'Looptijd master' in afbeelding 3 niet terug te vinden is als tabel in afbeelding 1. Dat klopt: attributen die behoren tot een zogenaamd enumeratief domein zoals 'geslacht' en 'sectorcode', gedragen zich in High-T



**Afbeelding 4:** Master on the fly: MasterMe.



**Afbeelding 5:** MasterMorph: gemiddelde leeftijd per sexe per werkgever.

schermen als code/waarde-tabellen en kunnen net als elke 'echte' tabel of view terugkomen als master, als detail of als *lookup* veld, zoals het veld 'Sexe'. Voor de eindgebruiker wordt zelfs een eenvoudig gegevensmodel zoals dat van de PA als het ware omgeven door een wolk van virtuele code/waarde-tabellen die zich voor de eindgebruiker niet onderscheiden van echte tabellen. En daarmee explodeert het aantal zinvolle schermconstructies dat declaratief (dus zonder programmeur) kan worden ingeregeld. Omdat in High-T alles dynamisch is, kan de geautomatiseerde eindgebruiker ook op ieder schermveld dat onderdeel uitmaakt van een verwijzende sleutel of dat behoort tot een enumeratief domein, een *lookup* schermpje genereren. Behalve dat daarvoor niets hoeft te worden geprogrammeerd, is ook dit weer niets nieuws. Wel nieuw, zonet uniek, is dat de eindgebruiker dat *lookup* schermpje met één muisklik kan transformeren in een extra master door de *MasterMe* button aan te klikken. In afbeelding 4 zien we zo'n situatie: de eindgebruiker heeft een *lookup* op het veld Persoon.Geslacht gemaakt en deze daarna gepromoveerd tot een derde master-gegevensblok. We zien nu van de geselecteerde bedrijven alle vrouwelijke medewerkers. Een klik en we zien alleen de mannen. Control-A en we zien weer alle medewerkers ongeacht geslacht. Zelfs een eenvoudige eindgebruiker wordt tot een kenniswerker.

## OLAP zonder 'P': de MasterMorph

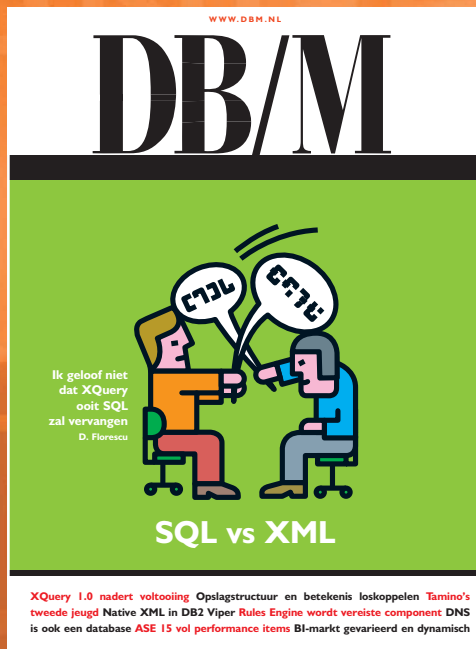
Wat we tot nu toe hebben beschreven is op z'n minst vernieuwend, maar wel binnen het klassieke paradigma van front-end software voor administratieve toepassingen. Maar wie dit artikel leest met een OLAP- of BI-bril op, die ziet iets anders. Multi-Current-MultiMaster schermen zoals in afbeeldingen 3 en 4 zijn ook te zien als platte weergaven van multidimensionale informatiemodellen – sterschema's zo u wilt. Om precies te zijn: afbeelding 3 toont een matrix en afbeelding 4 een kubus. Doordat we ons in de administratieve wereld altijd hebben beperkt tot details met één master en één actief master record – Single Current en Single Master – is er een compleet afgescheiden wereld van

OLAP en BI ontstaan met eigen gespecialiseerde tools en zelfs eigen databasestructuren. Het zal nog even duren, maar met MuCu en MuMa luiden we het einde in van het OLTP/OLAP-schisma en brengen we iedereen weer onder de klassieke relationele paraplu. Een enkel artikel is veel te weinig om dit hard te onderbouwen, maar we maken wel een begin. We kijken terug naar afbeelding 4 en we zien dat de cursor staat op het veld 'Leeftijd', een veld op een detail gegevensblok met drie masters. Enkele muisklikken verder heeft de eindgebruiker dit gegevensblok omgetoverd in de matrix van afbeelding 5.

Afbeelding 5 laat zien hoe dat uitpakt. Door 'Werkgever' te selecteren als rij-ingang, (dimensiemaster 1) en 'Geslacht' als kolom-ingang (dimensiemaster 2) en de AVG-functie aan te klikken (hier niet getoond), ontstaat een matrix met per werkgever per sexe een overzicht van de gemiddelde leeftijd van de medewerkers met een dienstverband voor onbepaalde tijd. De derde ingang, de 'looptijd master' zien we niet terug in deze matrix maar fungeert als extra restrictie (restrictiemaster). De matrix van afbeelding 5 toont per werkgever per sexe de gemiddelde leeftijd van alle medewerkers met een dienstverband voor onbepaalde tijd. We hebben de conceptuele kubus van afbeelding 4 omgezet in een tweedimensionale matrix. Natuurlijk kunnen we probleemloos omschakelen van de ene naar de andere matrix of zelfs tegelijk meerdere matrices tonen. Goed beschouwd is er in het geval van afbeelding 4 zelfs vrijheid van beweging in vijf dimensies: behalve Werkgever, Looptijd en Geslacht is er ook nog een dimensie die wordt bepaald door het aantal velden van het detail-gegevensblok (Inkomstenverhouding-plus-persoon) en een dimensie die wordt gedefinieerd door de voorgedefinieerde functies die door High-T worden ondersteund: Sum, Avg, Count, Percentage, enzovoort. Op basis van het voorbeeld van afbeelding 4 kunnen we vele tientallen verschillende matrices maken. En al die matrices zijn potentieel zinvol, want zo iets betekenisloos als het gemiddelde sofinummer wordt door High-T niet toegestaan. Het aantal rijen en kolommen in een matrix kan

Neem nu een abonnement op

# Database Magazine



## Méér dan een vakblad!

Database Magazine is al sinds 1990 een **kwaliteitsvakblad** op het gebied van databeheer, databases, datawarehousing en Business Intelligence. Onderwerpen als relationele en object-georiënteerde DBMS-en, datamodellering, database-ontwerp, datakwaliteit, datawarehousing, datamanagement, metadata, tools, datamigratie, data-integratie, performance, OLAP, ETL, Web Services, Business Intelligence, Analytic Applications, SQL, XML... u komt ze acht maal per jaar tegen in DB/M. Met vaak praktische tips en voorbeelden. De praktijk staat ook voorop in zo'n 30 **seminars, masterclasses en congressen** die speciaal voor u worden georganiseerd.

Database Magazine heeft een uitgebreide **website** met onder andere:

- actueel nieuws uit de database-markt,
- een actuele agenda met relevante events,
- compleet overzicht van alle producten in de markt,

- een database met detail-informatie van BI-tools: de BI-Matrix,
- het online archief met alle artikelen uit het blad, te downloaden door abonnees,
- een overzicht van relevante vacatures op uw vakgebied.

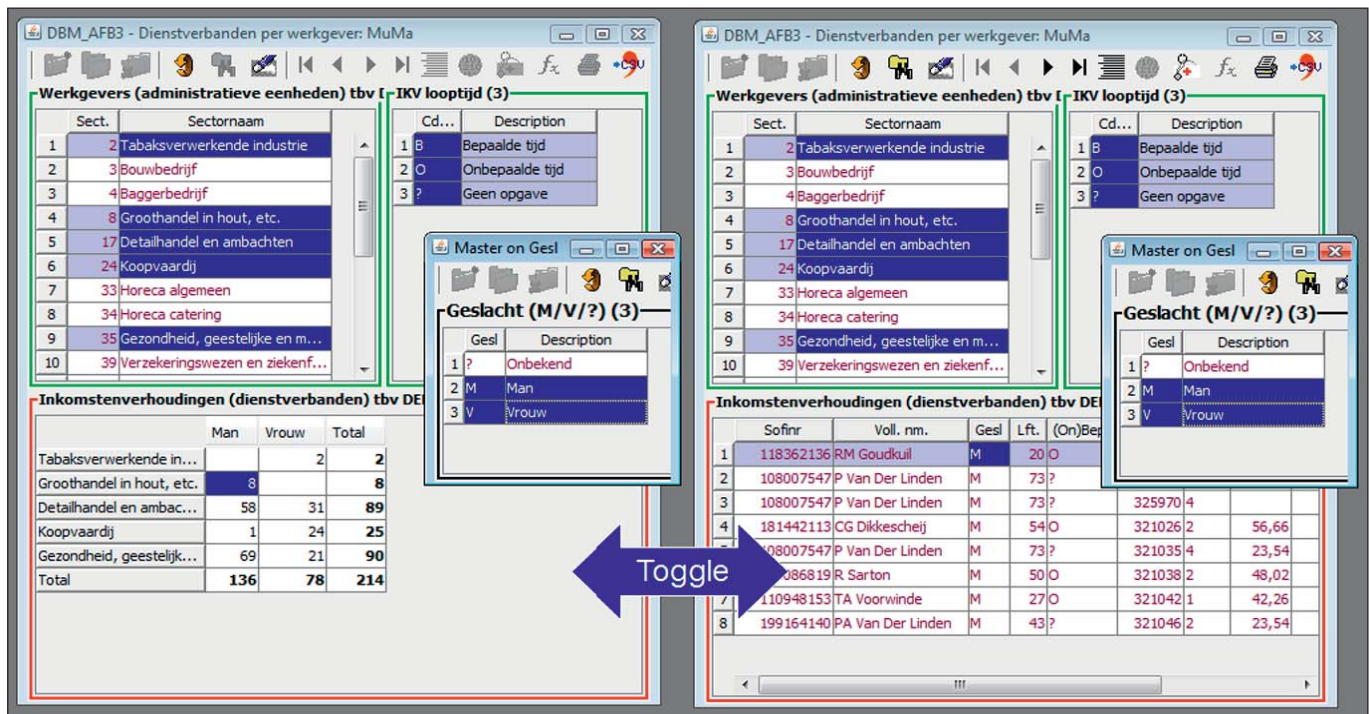
Ook kunt u zich kosteloos abonneren op de **e-mail nieuwsbrief** die eenmaal per 3 weken verschijnt en geheel in het teken staat van Business Intelligence en Datawarehousing.

U ziet: DB/M is méér dan een vakblad en is uitgegroeid tot een onmisbare vraagbaak voor IT-managers, DBA's, analisten, consultants, datamanagers, database-specialisten en datawarehouse-ontwerpers en –bouwers.

### **Nog geen abonnee?**

Meld u online aan op **www.dbm.nl**. Het eerste jaar profiteert u van bijna 50% korting voor nieuwe abonnees.

[www.dbm.nl](http://www.dbm.nl)



**Afbeelding 6:** Drill Up matrix: sexe per sector – heen en weer tussen OLTP en OLAP.

door de gebruiker ook dynamisch worden aangepast door de MuCu-selectie in de twee dimensiemasters te veranderen. Tenslotte kan een nuttige matrix ook als vast onderdeel van een scherm worden gedefinieerd en als zodanig worden opgeslagen: Uiterst handig indien de matrixingang vaak wordt gebruikt of wanneer de eindgebruiker een digibeeft is.

### Drill Me Up: de MetaMasterMorph

In de wereld van OLAP en BI die we nu hebben betreden is de *drill down*, de mogelijkheid om vanuit een matrix in te zoomen op onderliggende details een centraal concept. High-T ondersteunt dat concept natuurlijk ook, maar via een omweg die *drill up* heet. Daarbij vallen we terug op een oud en haast vergeten basisconcept van het relationele model: bij een projectie verdwijnen eventuele dubbele records. Stel we pakken het scherm van afbeelding 4 en we besluiten om alle velden uit het Werkgevergegevensblok te verwijderen met uitzondering van de sector. Wat we dan krijgen is afgebeeld links-boven in afbeelding 6: een gegevensblok waarin we geen individuele werkgevers meer zien maar sectoren. Daaronder gaan nog steeds de geselecteerde werkgevers schuil, maar als we nu een matrix maken met 'Werkgever' als dimensiemaster zijn de masterrijen geen werkgevers meer maar de sectoren waartoe deze werkgevers behoren. Het is dus mogelijk om vanuit de details meer ingedikte matrices te maken, zoals de sector/geslacht matrix die we zien in afbeelding 6. Ook *drill up* matrices kunnen worden opgeslagen en daarmee permanent beschikbaar gesteld. Op basis hiervan gaan we nu aan de slag met de klassieke *drill down*: enerzijds kunnen we de matrixen dynamisch in- en uitzoomen maar ook kunnen we de gegevens onder een matrix met een druk op de knop op maat tonen.

Weer hebben we een OLAP-concept bij de lurven: *slice & dice*. In afbeelding 6 hebben we één matrixveld geselecteerd. Twee klikken verder en we zien alle mannen van werkgevers in de Groothandel in hout, etc. En anders dan bij OLAP-tools kunnen we via matrices ook (bulk)mutaties doorvoeren, zoals het verhogen van alle lonen met 3 procent. Zo profiteren we ook bij het doorvoeren van mutaties van het samengaan van OLTP en OLAP.

### Terug op de grond...

De mogelijkheden van het gecombineerde administratieve/OLAP-paradigma dat we in dit artikel hebben geïntroduceerd zijn adembenemend. In dit artikel hebben we slechts het topje van de ijsberg kunnen tonen. Om één ding te noemen: wat we hier hebben laten zien is de toepassing op kleine hoeveelheden gegevens. Bij *hardcore* OLAP gaat juist om grote hoeveelheden gegevens. Zeker bij de enorme omvang van de PA zou opschaaling van duizenden naar honderden miljoenen *detail*-records een enorme sprong vooruit betekenen voor data-analisten en statistici. Over dit soort functionaliteit wordt nu nagedacht.

*De PA wordt grootschalig gemuteerd en geraadpleegd, groeit als kool en wordt een steeds belangrijkere schakel in overheidsland. Het volgende artikel behandelt de belangrijkste overwegingen en keuzes rond de implementatie van de PA en bespreekt de strategie die wordt gevolgd om het systeem werkbaar te houden.*

**René Veldwijk** is partner bij FAA Partners, onderdeel van de Ockham Groep. **Robert Veldwijk** is software engineer bij What-If software Solutions.