The Third Manifesto: setting the Record Straight (2)

Integriteit en toekenning

H. Darwen en C.J. Date

In deze bijdrage bespreken de auteurs van The Third Manifesto (TTM) de opmerkingen van Maurice Gittens over het naar zijn mening ontbreken van integriteit bij toekenningen voor relvars.

In zijn artikel [2] geeft Gittens een voorbeeld met een relvar AUTHOR met key SURNAME en de volgende waarde:

SURNAME	FIRST_NAME
Date	Chris
Darwen	Hugh

Dit voorbeeld krijgt een update waardoor de waarde verandert in:

SURNAME	FIRST_NAME
Darwen	Chris
Date	Hugh

In DB/M 2 van dit jaar uitte Maurice Gittens nogal wat kritiek op de derde editie van het standaardwerk 'Databases, Types, and the Relational Model: The Third Manifesto' van Hugh Darwen en Chris Date. Gittens vraagt zich af wat de bijdrage van Darwen en Date aan het relationele model de afgelopen jaren is geweest – de derde editie van TTM acht hij zelfs een regressie ten opzichte van de ideeën en principes van wijlen Ted Codd, grondlegger van het relationele model. Hij onderbouwt dit met zes argumenten. Het genoemde artikel 'Twijfels over logische correctheid' is een afgeleide van Gittens' publicatie op zijn website www.gittens.nl, zie [2].
Hugh Darwen en C.J. Date hebben Gittens' commentaar uiterst serieus genomen en krijgen van DB/M de gelegenheid hun standpunten en meningen ten aanzien van Gittens' opmerkingen diepgaand toe te lichten. Leerzame stof over de basisregels van het relationele database-model. Dit is de tweede bijdrage in een serie.

De update wordt uitgevoerd door middel van het volgende 'Double UPDATE' statement:

```
UPDATE AUTHOR WHERE SURNAME = 'Date'
                            ( SURNAME := 'Darwen' ) ,
UPDATE AUTHOR WHERE SURNAME = 'Darwen'
                              ( SURNAME := 'Date' ) ;
```

Dit statement is een meervoudig toekenning en de twee enkelvoudige toekenningen (de individuele UPDATE's) die het bevat zijn relationele toekenningen. We vermelden deze feiten omdat:

- de titel van de desbetreffende paragraaf in Gittens' paper luidt: "Geen semantische integriteit in de aanwezigheid van een relationele toekenning", daarmee aangevend dat zijn kritiek specifiek gaat om relationele toekenningen;
- in tegenstelling daarmee impliceert de verdere tekst "Doordat de integriteitscontrole wordt uitgesteld tot de gehele meervoudige toekenning is afgewerkt" dat zijn kritiek zich richt op meervoudige toekenningen. (Zoals bekend zijn de bekende INSERT, DELETE en UPDATE operators alleen maar verkorte, en daarom logische equivalenten van zekere relationele toekenningen.)

Hoe het ook zij, Gittens stelt vervolgens:
Gegeven de relationele values van de relvar AUTHOR voor en na dit enkelvoudige toekenning en de wetenschap dat slechts één toekenning is uitgevoerd (sic), veronderstel een forensische applicatie die moet uitzoeken wat er door dit toekenning statement precies is veranderd. Het niet onderkennen van deze fundamentele barst in de door TTM vereenvoudigde integriteit, zal waarschijnlijk leiden tot volkomen verkeerde conclusies zoals:

- *De voornaam van de AUTHOR met de achternaam "Date" is gewijzigd in "Hugh", en;*
- *De voornaam van de AUTHOR met de achternaam "Darwen" is gewijzigd in "Chris".*

Dit is een verwarrende situatie. We zullen Gittens' punten één voor één bekijken.

Wat is er veranderd?

Gittens zegt "Veronderstel een forensische applicatie die moet uitzoeken wat er door dit toekenning statement precies is veranderd", doelend op de eerdergenoemde "double UPDATE". Wel,

de wijziging op zich is volkomen duidelijk: de oorspronkelijke waarde van de relvar is gewijzigd in (of beter: is vervangen door) een andere waarde, en zowel de oorspronkelijke waarde als de vervangende waarde zijn volkomen expliciet.

Welke bres in de integriteit?

Gittens wijst op een "fundamentele bres in de integriteit". Die is er helemaal niet. Na de update voldoet de relvar aan de enige genoemde integriteitsbeperking, namelijk dat achternamen uniek zijn. Inderdaad, want als dat niet het het geval was, zou de update worden afgewezen.

We voegen hier aan toe dat een van de redenen waarom meervoudig toekenning nuttig is, is in verband met soortgelijke situaties als in het onderhavige voorbeeld,. Denk aan het probleem om de waarden van de variabelen X en Y onderling te wijzigen. Het lijkt voor de hand te liggen om dit te doen zonder een meervoudig toekenning, maar met een tijdelijke variabele Z:

```
Z := X ; X := Y ; Y := Z ;
```

Met een meervoudig toekenning zou het gewenste effect echter veel eenvoudiger worden verkregen:

```
X := Y , Y := X ;
```

Welke verkeerde conclusies?

Gittens refereert aan "volkomen verkeerde conclusies". In welke zin zijn de conclusies die Gittens noemt dan precies zo "volkomen verkeerd"? Dit vraagt om verduidelijking. Eigenlijk vermoeden we dat er sprake is van grote verwarring: ergens in dezelfde paragraaf verwijst Gittens naar de dubbele UPDATE als "het omwisselen van de waarden van de twee tuples". Maar tuples ZIJN waarden en kunnen dus per definitie niet worden gewijzigd. We vermoeden dat dit is wat hij bedoelt:

- Elke tuple in de AUTHOR relvar staat voor een echt bestaande auteur. Nog specifieker: de SURNAME waarde in zo'n tuple identificeert de betreffende auteur op voor de hand liggende wijze;
- Stel dat tuple t in de AUTHOR relvar staat voor auteur x, oftewel de echt bestaande auteur met achternaam x.
- Het updaten van de AUTHOR relvar op zo'n manier dat: a. de waarde na de update alleen hierin verschilt van de vorige waarde dat tuple t is vervangen door t1, en; b. tuple t1 alleen hierin verschilt van tuple t dat het een andere FIRST_NAME component heeft, kan gezien worden als het weergeven van een wijziging van de voornaam van auteur x.
- Al we het zo interpreteren, willen we waarschijnlijk niet dat men key's kan updaten. Want, stel dat we de AUTHOR relvar zo zouden willen updaten dat: a. de waarde na de update alleen hierin verschilt dat tuple t is vervangen door tuple t2, en; b. tuple t2 alleen hierin verschilt van tuple t dat het een andere SURNAME component heeft (zeg y in plaats van x). Dan kunnen we die update moeilijk zien als een weergave van een wijziging in de achternaam van auteur x – omdat auteurs worden *geïdentificeerd* door hun achternaam, en de regel "auteur x" na de update duidelijk helemaal niet aan een echte bestaande auteur refereert. (Het duidt zeker niet op een echte bestaande auteur die nu wordt voorgesteld in de AUTHOR relvar).

> Na de update voldoet de relvar aan de enige genoemde integriteitsbeperking

- Dus willen we een conventie introduceren, volgens welke zekere attributen – in het bijzonder zekere key attributen – expliciet worden gedefinieerd als niet-updatable. Stel dat SURNAME zo'n attribuut is. Dan zou blijken dat een UPDATE statement zoals

```
UPDATE AUTHOR WHERE SURNAME = x ( SURNAME := y ) ;
```

  onrechtmatig is. (We merken hierbij op, dat als onze interpretatie van Gittens' kritiek klopt, het helemaal niet nodig is om meervoudige toekenningen in de discussie te betrekken. Om het eenvoudig te houden beperken we ons daarom tot enkelvoudige toekenningen en het vervolg van de discussie).
- Gittens zou dan te berde kunnen brengen dat een relationele toekenning (de enige relationele update operator die momenteel door TTM wordt voorgeschreven) gewoon niet fijnkorrelig genoeg is voor regels als "attribuut A is niet-updatable", omdat het alleen maar de volledige waarde van een target relvar vervangt door een andere waarde. Het is inderdaad moeilijk om precies vast te stellen welke toekenningen onder zo'n regel onrechtmatig zouden zijn.
- Vervolgens zou Gittens kunnen opbrengen dat een expliciet UPDATE statement ondersteund moet worden (in plaats van alleen maar een optionele ingekorte vorm van een toekenning te zijn, wat het nu in TTM is) om regels als "Attribuut A is niet-updatable" te kunnen toepassen.
- Maar zelfs als we dit argument accepteren, bereik je met de bepaling dat een attribuut 'niet-updatable' is helemaal niets! Stel dat de tuple met SURNAME waarde x FIRST_NAME waarde z heeft. Dan kan het gevolg van de UPDATE

```
UPDATE AUTHOR WHERE SURNAME = x ( SURNAME := y ) ;
```

  (die vermoedelijk onder de voorgestelde niet-updatabale regel niet zal werken) duidelijk worden verkregen door de volgende, volkomen rechtmatige DELETE / INSERT sequence:

```
DELETE AUTHOR WHERE SURNAME = x ;
INSERT AUTHOR
```

```
RELATION { TUPLE { SURNAME y, FIRST_NAME z } } ;
```

(of vanzelfsprekend door een ander logisch equivalent paar expliciete relationele toekenningen).

Een mogelijke discipline

Gittens vervolgt: "Het probleem is dat een toekenning niet de mogelijkheden heeft om specifieke tuples bij te houden, omdat Darwen en Date ervoor hebben gekozen om het concept van tuple-identiteit af te wijzen."

Op het eerste gezicht lijkt deze uitspraak niet erg zinnig; om preciezer te zijn: de zinsnede "het bijhouden van specifieke tuples" snijdt geen hout. Om Gertrude Stein te parafraseren: een tuple is een tuple is een tuple; het is een *waarde* en net als alle waarden IS het alleen maar – het heeft geen locatie in tijd en ruimte en de vraag om het dus "bij te houden" doet zich dus niet voor. Wat Gittens eigenlijk wil, zo denken wij, is het op de een of andere manier bijhouden van de historie van waarden van tuple-*variabelen*. Maar we zijn het met Codd eens dat we behalve *relatie*-variabelen geen enkele andere soort variabelen in de database toestaan, en we wijzen deze suggestie dus van de hand.

Dat allemaal gezegd zijnde: er is niets in TTM dat de adoptie van een conventie of discipline, die bewerkstelligt wat Gittens schijnt te willen, tegenhoudt. In termen van zijn voorbeeld:

1. We zouden de conventie kunnen adopteren dat alle tuples die ooit, dus op elk moment, binnen de AUTHOR relvar verschijnen met dezelfde specifieke SURNAME waarde x, allemaal refereren aan 'dezelfde' echt bestaande auteur (of, beter gezegd, de conventie dat we alle dergelijke tuples interpreteren als refererend aan dezelfde echte, bestaande auteur).
2. We zouden een log kunnen bijhouden, dat laat zien wanneer dergelijke tuples verschijnen en verdwijnen uit de relvar.
3. Daarbij zou dat log de waarden van de andere attributen kunnen aangeven. (In het voorbeeld is er natuurlijk maar één dergelijk attribuut, namelijk FIRST_NAME).
4. Dat log zou ook kunnen tonen wie en wat de oorzaak is van de verschijningen en verdwijningen.
5. We zouden ook een surrogaatsleutel aan de AUTHOR relvar kunnen toevoegen, en de stappen 1 tot en met 4 hiervoor herinterpreteren in termen van waarden van die sleutel, in plaats van SURNAME waarden. (Hoewel TTM op dit moment de ondersteuning van surrogaatsleutels niet vereist, wordt er wel sterk op aangedrongen dat te doen. Zie [1], hoofdstuk 10, RM Very Strong Suggestion 1 (we hebben geprobeerd erachter te komen welke redenen Gittens zou kunnen hebben om deze 'Very Strong Suggestion' af te wijzen als adequate

oplossing voor het probleem dat hij waarneemt, maar dat is ons niet gelukt). Op die manier kan een echte bestaande auteur 'dezelfde auteur' blijven, zelfs als zijn of haar achternaam verandert: feitelijk een realistische, wenselijke gang van zaken, omdat de achternamen van mensen in werkelijkheid wel eens veranderen.

Wij zijn niet alleen van mening dat een conventie zoals hierboven beschreven eenvoudig te adopteren is, maar vinden het bovendien een goed idee om dat vaker te doen.
We vinden echter niet dat TTM dergelijke zaken dwingend kan of moet voorschrijven; zulke zaken vallen per definitie buiten het werkveld.

Literatuur
1. *C.J. Date en Hugh Darwen: Databases, Types and the Relational Model: The Third Manifesto (3rd edition). Boston, Mass.: Addison-Wesley (2006).*
2. *Maurice Gittens: "The Third Manifesto Revisited," http://www.gittens.nl/TheTTMRevisited.pdf.*

Hugh Darwen en Chris Date zijn auteurs van de derde editie van The Third Manifesto.

Dit is een bewerkte en vertaalde versie van de originele Engelse tekst, die u kunt vinden op onze website www.dbm.nl onder 'specials', 'extra materiaal'. In geval van discussies is de originele Engelse tekst doorslaggevend.

Update

MarketCap: combinatie BO/SAP heeft 43 procent aandeel in Nederlandse BI-markt

De aangekondigde overname van Business Objects door SAP zal een aardige verschuiving van de krachtvelden binnen de Nederlandse Business Intelligence markt tot gevolg hebben, constateert MarketCap in het op 8 oktober 2007 gepubliceerde speciale Strategic Market Report 'The Impact of the Business Objects take-over (Benelux)'.
Met de overname koopt SAP binnen de Nederlandse markt de marktleider: Business Objects kent medio 2007 een marktpenetratie van 35,7 procent. Wanneer de twee klantenbases van Business Objects en SAP puur op gebied van Business Intelligence op elkaar worden gelegd, dan is de nieuwe marktpenetratie 42,8 procent. Op het gebied van Business Intelligence heeft slechts een te verwaarlozen aantal bedrijven en instellingen een BI-applicatie van beide organisaties, stelt MarketCap.
Business Objects kende op de Nederlandse markt een geduchte concurrent in Cognos, na de overname wordt Cognos in een keer op afstand gezet. Wel zal Cognos veruit de belangrijkste concurrent van het nieuwe SAP/BO blijven. Oracle heeft met de overname van Hyperion per medio 2007 een totale marktpenetratie verkregen van circa 7 procent. Daar SAP en Business Objects praktisch geen overlap kennen in de BI klantenbase, is de impact van deze overname binnen de Nederlandse markt aanzienlijk. Binnen de sectoren ICT & Utilities en de bouw zal de totale marktpenetratie van de nieuwe organisatie zelfs boven de 60 procent uit gaan komen. De Nederlandse Business Intelligence markt komt met deze overname zeer stevig in handen van SAP, aldus MarketCap.

In het rapport stelt MarketCap dat de overname van één van de grote BI-spelers zoals Business Objects al vanaf medio 2005 werd verwacht. Hyperion wilde een overname overleven c.q. voorkomen door Information Builders over te nemen. Oracle was echter sneller en haalde deze vooraanstaande BI-speler aan boord. Hoewel SAP niet snel tot grote overnames bereid is geweest in het verleden, is de overname van Business Objects wel een zeer logische volgende stap in de groei naar allround leverancier op gebied van Business Critical Applications. SAP heeft hiermee in één keer de markt van Business Intelligence stevig in handen. Hierbij dient wel opgemerkt te worden dat het met name om een zeer sterke positie binnen de Europese markt gaat. De overname van Hyperion door Oracle heeft daarentegen juist minder effect gehad op de BI-positie van Oracle in Europa. Hyperion is duidelijk minder sterk aanwezig binnen Europa, constateert MarketCap.

1000ste certificaat cursus 'Dimensionaal Modelleren'

Op 1 oktober 2007 reikte docent Dr. Harm van der Lek het 1000ste certificaat van de cursus 'Dimensionaal Modelleren' uit aan Raymond Buis van Delta Lloyd Verzekeringen. Bij deze cursus worden de deelnemers in staat gesteld het geleerde operationeel te maken aan de hand van uitgekiende opgaven. Harm van der Lek is naast data-architect en docent ook auteur, onder meer van het DB/M-boek 'Sterren en dimensies', en ontwikkelaar van de datawarehouse-generator BIReady.

Teller CBIP-gecertificeerde consultants staat op 120

In de zomer van dit jaar bereikte het aantal gecertificeerde BI-consultants (Certified Business Intelligence Professional of CBIP) in Nederland de mijlpaal van 100. Hoewel de teller inmiddels op 120 staat, is de 100ste Nederlandse CBIP'er op dinsdag 11 september 2007 gehuldigd te Putten.