

The Third Manifesto: Setting the Record Straight (3)

Relation Valued Attributen (3)

H. Darwen en C. J. Date

In deze bijdrage – in drie stukken geknipt, waarvan dit het derde deel is – gaan de auteurs van The Third Manifesto (TTM) zeer diep in op de opmerkingen van Maurice Gittens over het nut van relation valued attributen, in het artikel afgekort tot RVA's. Een afdaling naar de diepste krochten van de relationele database.

Appendix A van referentie [9] bevat een formele definitie van een relationele algebra, die A heet (beschreven in referentie [7]). Met de definitie van die algebra beoogden wij een onwrikbaar fundament te leggen voor het ontwerp van andere relationele talen; referentie [9] zelf gebruikt het als basis voor de definitie van Tutorial D, en we hopen dat ontwerpers van andere relationele talen het zullen toepassen voor soortgelijke doeleinden. Maar Gittens' kritiek op de Relation Valued Attributes (RVA's) bevat ook enkele opmerkingen over A, die gebaseerd lijken op enige misvattingen over die algebra;

Algebra A blijkt geen operators te hebben waarmee RVA's kunnen worden geïsoleerd voor verdere bewerking. Daaruit volgt dat vanuit het perspectief van algebra A RVA's in het geheel geen relatiewaarden zijn. Dit blijkt overduidelijk, omdat deze RVA's niet kunnen worden gemanipuleerd door de verzameling van

In DB/M 2 van 2007 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 [11].

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 het derde deel van de derde bijdrage in een serie. De vierde en laatste bijdrage verschijnt in DB/M 2.

relationele operators, die speciaal ontworpen zijn om de transformatie van relationele waarden mogelijk te maken. Gelet op de transformaties waarin algebra A voorziet, zijn RVA's blijkbaar een doel op zich. Vanuit algebra A gezien kunnen bijvoorbeeld de volgende vragen gesteld worden.

- *Op welke wijze zou algebra A logisch anders zijn, als de GROUP operator een XML waarde produceerde?*
- *In zijn algemeenheid, op welke manier verschilt de ondersteuning van RVA's logisch gezien van bijvoorbeeld ingebouwde ondersteuning van XML valued attributen?*

De meest serieuze claim – waarmee we als eerste willen afrekenen – is hier dat relaties die toevallig waarden zijn van RVA's op de een of andere manier speciaal zijn, in zoverre dat operators die worden toegepast op relaties in het algemeen, niet toepasbaar zijn op zulke relaties in het bijzonder. Dat is niet zo. Kijk eens naar het volgende voorbeeld:

```
spq WHERE TUPLE { P# P# ('P2) } e PQ { P# }
```

Deze expressie heeft het effect van een beperkende relatie spq, zoals getoond in afbeelding 1 gepubliceerd in DB/M 7 2007, alleen voor de tupels waarin de PQ-waarde (een relatie, vanzelfsprekend) een tupel voor deel P2 bevat. De expressie PQ{P#} representeert de projectie van de PQ-waarde boven P#; met andere woorden, het is een voorbeeld van de toepassing van een conventionele operator – viz., projection – op een relatie die toevallig de waarde is van een RVA binnen (een zekere tupel binnen) een bepaalde relatie. In het algemeen stelt referentie [9] feitelijk op pagina 155 dat "TTM ondersteuning vraagt voor relaties met RVA's, en alle operators die toegepast worden op relaties in het algemeen voor waarden van de onderhavige attributen."

Relaties zijn vanzelfsprekend waarden en zijn "operators die toegepast worden op relaties in het algemeen" per definitie read-only operators. Om precies te zijn, de operators van A zijn allemaal read-only, omdat A algebra is en besef heeft van noch variabelen noch operators (relationeel of anderszins). Maar er is niets dat een relationele taal ervan weerhoudt om te zorgen voor handige, korte operators voor de updating van relvars met RVA's. Met name Tutorial D voorziet in zulke korte operators. Hier zijn wat voorbeelden (afkomstig uit referentie [9] pagina 155):

```
UPDATE SPQ WHERE S# = S# ('S2')
( UPDATE PQ WHERE P# = P# ('P3') ( QTY : =
QTY * 2 ) )
```

```
UPDATE SPQ WHERE S# = S# ('S2')
( INSERT PQ RELATION { TUPLE { P# P# ( 'P5' ),
QTY QTY ( 500 ) } } ) ;
```

(We nemen in deze voorbeelden aan dat SPQ een relvar is van dezelfde soort als relatie spq uit afbeelding 1, zie DB/M 7 2007 pag. 40.)

Manipulatie

Er zijn nog enkele dingen die we kwijt willen over Gittens' kritiek op A en aanverwante zaken. Ten eerste begrijpen we niet echt wat het voor zin zou hebben om (waarden van) RVA's "te kunnen isoleren voor verdere bewerking". Als we dat willen, kunnen we wel degelijk individuele RVA-waarden extraheren. De volgende expressie geeft de relatie aan die de PQ-waarde is voor toeleverancier S4 binnen relatie spq:

```
PQ FROM ( TUPLE FROM ( spq WHERE S# = S# ('S4') ) )
```

Sterker nog: deze expressie kan gebruikt worden om – volgens het bekende principe van taalontwerp genaamd orthogonaliteit – de gespecificeerde relatie aan te duiden en elke context waar de regels van de taal vereisen dat er een relatie van het bijpassende type is. Bijvoorbeeld, binnen een expressie die opdracht geeft een JOIN uit te voeren met een andere relatie. Dus als dat is wat Gittens bedoelt met het "isoleren voor verdere bewerking" van een bepaalde waarde van een zekere RVA, dan kunnen we dat zeker doen. Met andere woorden, RVA-waarden kunnen wel degelijk "worden gemanipuleerd door de verzameling van relationele operators, die speciaal ontworpen zijn om de transformatie van relationele waarden mogelijk te maken" – in tegenstelling tot wat Gittens beweert.

Of bedoelt Gittens met 'manipulatie' eigenlijk 'update'? (De term 'manipulatie' wordt vaak in die zin gebruikt; het is geen erg gepaste term, maar het is door gebruik gemeengoed geworden). We hebben al laten zien hoe syntactische, korte operators kunnen worden gedefinieerd die de illusie wekken dat een RVA direct een update krijgt. Dat kunnen we dus óók.

Gittens vraagt ook: "Op welke wijze zou algebra A logisch anders zijn, als de GROUP operator een XML waarde produceerde?" Het is verleidelijk om een wedervraag te stellen: "Op welke manier zou een kat anders zijn als het een hond was?" Om het wat beleefder te formuleren: we moeten toegeven dat we de bedoeling van de vraag niet begrijpen. Als we RVA's gebruiken, hebben we natuurlijk operators nodig voor de mapping tussen relaties die dergelijke attributen bevatten en relaties die dat niet doen. Dat is wat GROUP en UNGROUP doen – GROUP transformeert een relatie zonder een RVA in één mét, en UNGROUP doet het omgekeerde (in beide gevallen grofweg gesteld).

Dit alles gezegd zijnde, voegen we daar nu aan toe dat algebra A toch helemaal geen GROUP en UNGROUP bevat! Een van de bedoelingen van Appendix A van referentie [9] is precies om dat aan te tonen, terwijl zulke operators erg nuttig zijn in een concrete relationele taal, ze zijn op de keper beschouwd alleen maar ingekorte versies van bepaalde combinaties van andere operators.

We willen er graag aan toevoegen, dat er in principe niets mis is met het idee om een operator te definiëren die een relatie in een XML document kan omzetten. Echter: a. zo'n operator kan als zodanig geen deel uitmaken van relationele algebra, juist omdat het iets produceert dat geen relatie is; en b. het is waarschijnlijk niet zo'n goed idee om die operator GROUP te noemen, omdat de naam GROUP al gebruikt is, als het ware.

Tot slot vraagt Gittens: "In zijn algemeenheid, op welke manier verschilt de ondersteuning van RVA's logisch gezien van bijvoorbeeld ingebouwde ondersteuning van XML valued attributen?" Opnieuw moeten we toegeven dat we de vraag niet begrijpen. Een van de bedoelingen van TTM is om een theorie van typen te definiëren, een die begeleidt maar orthogonaal is aan de theorie die het relationele model heet. En een van de doelstellingen van die theorie van typen is om vervolgens te zeggen, dat relationele attributen elk willekeurig type kunnen zijn. (Met twee kleine uitzonderingen, die we alleen noemen voor de compleetheid. Ten eerste staan we geen recursief gedefinieerde typen toe – typen die zijn gedefinieerd in termen van zichzelf; ten tweede staat het relationele model niet toe dat relaties attributen van *type pointer* hebben – waar we met type pointer een type bedoelen met eraan verbonden operators voor referentiëring en de-referentiëring.)

Een van de bedoelingen van TTM is om een theorie van typen te definiëren

Attributen 'van het type XML' – of beter, van het type XML document – zijn dus helemaal niet verboden. Of de ondersteuning van dat type is ingebouwd of door de gebruiker wordt gedefinieerd maakt wat ons betreft niets uit. Daarentegen eisen we wel dat 'support voor RVA's' wordt ingebouwd, in zoverre als relationele types zelf zijn ingebouwd.

Opmerking: de laatste zin van de voorgaande paragraaf is enigszins overgesimplificeerd: relatietypen worden meestal niet ingebouwd – tenminste, niet precies. We kunnen ze echter behandelen alsof ze dat wel zijn, tenminste bij benadering, en dat is goed genoeg voor hedendaagse doeleinden. Het hele verhaal is een beetje gecompliceerd en om dat helemaal uit te leggen moeten we ver buiten de scope van dit artikel treden.

Slotopmerkingen

Gittens is niet de eerste die kritiek heeft op de manier waarop TTM omgaat met RVA's. Merkwaaardig genoeg beperkten de vorige critici zich echter tot een of beide van de volgende twee kwesties, terwijl Gittens geen van beide noemt (tenminste, niet expliciet):

- zijn RVA's niet in strijd met de eerste normaalvorm;
- voeren RVA's ons niet naar logica van de tweede orde?

Omdat Gittens geen van beide vragen expliciet heeft gesteld, hebben we ervoor gekozen om ze hier niet in detail te beantwoorden. Voor de goede orde: het antwoord op de eerste vraag is nee, het antwoord op de tweede vraag is misschien, met de aantekening dat we niet zeker weten of het een goede vraag is. Gedetailleerde discussies over deze kwesties staan in de referenties [5] en [8].

Literatuur

1. E. F. Codd: *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*, IBM Research Report RJ599 (August 19th, 1969).
2. E. F. Codd: *A Relational Model of Data for Large Shared Data Banks*, CACM 13, No. 6 (June 1970). Republished in *Milestones of Research*, CACM 26, No. 1 (January 1982).
3. Hugh Darwen: *Relation Valued Attributes; or, Will the Real First Normal Form Please Stand Up?*, in C.J. Date and Hugh Darwen, *Relational Database Writings 1989-1991*. Reading, Mass.: Addison-Wesley (1992).

4. C.J. Date: *Missing Information*, in *An Introduction to Database Systems (8th edition)*. Boston, Mass.: Addison-Wesley (2004).
5. C.J. Date: *What First Normal Form Really Means*, in *Date on Database: Writings 2000-2006*. Berkeley, Calif.: Apress (2006).
6. C.J. Date: *The Closed World Assumption*, in *Logic and Databases: The Roots of Relational Theory*. Victoria, BC: Trafford Publishing (2007).
7. C.J. Date: *Why Is It Called Relational Algebra?*, in *Logic and Databases: The Roots of Relational Theory*. Victoria, BC: Trafford Publishing (2007).
8. C.J. Date: *Frequently Asked Questions*, in *Logic and Databases: The Roots of Relational Theory*. Victoria, BC: Trafford Publishing (2007).
9. C.J. Date and Hugh Darwen: *Databases, Types, and the Relational Model: The Third Manifesto (3rd edition)*. Boston, Mass.: Addison-Wesley (2006).
10. C.J. Date, Hugh Darwen, and Nikos A. Lorentzos: *Temporal Data and the Relational Model*. San Francisco, Calif.: Morgan Kaufmann (2003).
11. Maurice Gittens: *The Third Manifesto Revisited*, 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.



You have a **Point**.
You make your **Point**..
You get the **Point**...
You want to prove your **Point**....

Join **FourPoints** !!