

The Third Manifesto: setting the Record Straight (3)

# Relation Valued Attributen (1)

H. Darwen en C.J. Date

**In deze bedrage – die in drie stukken geknipt 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.**

De opmerkingen van Gittens over dit onderwerp [11] beginnen als volgt:

*In tegenstelling tot Codd, staan Date en Darwen relation valued attributen (RVA's) toe. Dat roept bij mij de volgende vragen op:*

- *welk probleem wordt er door het ondersteunen van deze attributen opgelost dat niet op een andere manier had kunnen worden opgelost?*
- *welke proposities kunnen worden voorgesteld door relvars inclusief RVA's? Kunnen deze proposities niet op een andere relationele manier worden voorgesteld?*
- *welke positieve eigenschappen van de alternatieve oplossingen zijn niet beschikbaar binnen oplossingen die gebaseerd zijn op RAV's?*

De eerste twee paragrafen van dit artikel behandelen deze vragen. Noot: argumenten die onze mening in deze ondersteunen

werden eerder uitgesproken in [3]. Het huidige artikel kan worden gezien als een verdere uitwerking van die originele argumenten.

### Een eenvoudig voorbeeld

Natuurlijk heeft Gittens helemaal gelijk als hij zegt dat we relaties toestaan om relation valued attributen (RVA's) te hebben. Afbeelding 1 laat een voorbeeld van zo'n relatie zien, waaraan we formeel referen als spq maar informeel als 'shipments'. Relatie spq heeft twee attributen, S # en PQ; de waarden van S # zijn leveranciersnummers en PQ-waarden zijn relaties en dus is attribuut PQ een RVA. Deze PQ-relaties hebben op hun beurt twee attributen, P # en QTY, waarbij de waarden van P # onderdeelnummers zijn en QTY-waarden hoeveelheden. De beoogde interpretatie of betekenis van relatie spq – oftewel het relatie-predicaat van die relatie – is: de specifieke leverancier levert de specifieke aandelen in de gespecificeerde hoeveelheden.

Bijvoorbeeld: de tuple voor leverancier S4 representeert de propositie: Leverancier S4 levert onderdeel P2 in hoeveelheid 200, deel P4 in hoeveelheid 300, en deel P5 in hoeveelheid 400.

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](http://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 derde bijdrage in een serie.

S#	PQ	
S3	P#	QTY
	P2	200
S4	P#	QTY
	P2	200
	P4	300
	P5	400
S5	P#	QTY

**Afbeelding 1:** relatie spq ('shipments' met een RVA).

Noot: onze formuleringen van het predicaat en voorbeeldpropositie zijn beide opzettelijk nogal losjes. In het geval van het predicaat is een precieze – maar veel minder idiomatische – formulering:

Voor een gegeven leverancier, geïdentificeerd door een leveranciersnummer (S #) wordt de verzameling onderdelen die door die leverancier worden geleverd, gerepresenteerd door de corresponderende PQ-waarde; elk onderdeel wordt geïdentificeerd door een onderdeelnummer (P #) en de corresponderende hoeveelheid wordt geïdentificeerd door de corresponderende QTY-waarde.

De reden waarom wij deze precieze formulering geven, zal in de volgende paragraaf duidelijk worden.

## Representatie van proposities (I)

Nu gaan we terug naar Gittens' vragen. We hebben ervoor gekozen ze in omgekeerde volgorde te behandelen. De laatste is:

- Welke positieve eigenschappen van de alternatieve oplossingen zijn niet beschikbaar binnen oplossingen die gebaseerd zijn op RAV's?

We beginnen met deze vraag om hem direct te kunnen afhandelen. Feitelijk kan de vraag niet op een absolute manier worden beantwoord – soms zijn oplossingen gebaseerd op RVA's beter, soms de alternatieven; met andere woorden, het antwoord hangt af van de context. De rest van dit artikel kan beschouwd worden als een verdere uitwerking hiervan.

De tweede vraag – die eigenlijk uit twee vragen bestaat – van Gittens luidt:

- welke proposities kunnen worden voorgesteld door relaties inclusief RVA's? Kunnen deze proposities niet op een andere relationele manier worden voorgesteld? (Gittens giet zijn vraag in termen van relvars, niet in relaties, maar relvars zijn hier een dwaalspoor; zijn vraag kan bediscussieerd worden in termen van relaties, zonder relvars hierbij te betrekken. We komen later terug op het fenomeen relvars.)

We kijken in afbeelding 1 naar de tuple voor leverancier S5.

Verwijzend naar de meer precieze vorm van het predicaat, is het duidelijk dat de tuple de propositie voorstelt:

- voor leverancier S5 is de set van onderdeelnummer/hoeveelheid-paar voor alle onderdelen geleverd door die leverancier leeg.

Meer idiomatisch gesteld: leverancier S5 levert helemaal geen onderdelen.

Terzijde: eigenlijk is deze laatste formulering tamelijk slordig.

Als we dat een beetje abstracter maken: als we alleen maar weten dat er geen y-z paar bestaat voor een gegeven x, kunnen we daaruit logischerwijs niet de conclusie trekken dat er geen y's bestaan voor die x; we kunnen alleen afleiden dat er of geen y's bestaan of geen z's, of beide, voor x. Wetende dat x'en, y's en z's respectievelijk leveranciersnummers, onderdeelnummers en

hoeveelheden zijn, kunnen we echter wel naar een zeker feit in de echte wereld verwijzen – namelijk het feit dat als een leverancier een onderdeel levert, hij dat moet doen in enige hoeveelheid – om te kunnen afleiden dat als er geen paar onderdeelnummer/hoeveelheid voor een gegeven leveranciersnummer bestaat, dat er dan geen onderdelen voor die leverancier zijn.

Stel dat we moeten kiezen (zoals we in de praktijk normaliter zouden doen) hoe we 'shipments' zouden weergeven, niet door een relatie zoals spq met een RVA, maar liever door een relatie – laten we die sp noemen – zoals die in afbeelding 2 staat, zonder RVA. Het predicaat is: leverancier S# levert onderdeel P# in hoeveelheid QTY. Dan zien we dat zo'n relatie niet expliciet een propositie kan voorstellen zoals die gegeven is voor leverancier S5 ('Leverancier S5 levert helemaal geen onderdelen'). Waarom niet? Omdat voor een dergelijke leverancier, zeg s, er geen onderdeel p en geen hoeveelheid q voorkomt die nodig is om het predicaat 'leverancier s levert onderdeel p in hoeveelheid q' te vervullen, en er dus geen mogelijke tuple is die logischerwijs kan verschijnen in de relatie.

S#	P#	QTY
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Afbeelding 2: relatie sp ('shipments' zonder RVA).

We hebben dus nu een gedeeltelijk antwoord verkregen op Gittens' vraag; om precies te zijn, we hebben een propositie getoond die expliciet kan worden gerepresenteerd door een relatie mét een RVA en die niet expliciet kan worden gerepresenteerd zónder een RVA. Let op het gebruik van het woord 'expliciet' in de vorige zin. Het punt is dat geredeneerd kan worden dat relatie sp (degeen zónder een RVA) minstens de bedoelde propositie impliciet representeert. Om precies te zijn worden relaties conventioneel geïnterpreteerd in overeenstemming met wat genoemd wordt de 'Closed World Assumption'; en de Closed World Assumption zegt onder meer dat als tuple t in relatie r zou kunnen verschijnen maar het niet doet, dat dan de door tuple t voorgestelde propositie niet klopt. (Zie voor meer details over de 'Closed World Assumption' referentie [6]). In het onderhavige geval kunnen we daarom het volgende beredeneren:

- relatie sp bevat geen tuple voor leverancier S5;
- dus zal het predicaat 'Leverancier S5 levert onderdeel P# in

hoeveelheid QTY' als onrechtmatig worden aangemerkt voor alle mogelijke P#/QTY-paren;

- dus is er geen P#/QTY-paar waarvoor het predicaat 'Leverancier S5 levert onderdeel P# in hoeveelheid QTY' waarheid wordt;
- dus is er geen onderdeel dat leverancier S5 levert;
- dus levert leverancier S5 geen enkel onderdeel;
- dus representeert relatie sp de onderhavige propositie zoals de claim – het doet dat alleen niet expliciet.

**Representatie van proposities (2)**

Ondanks de argumenten in de vorige paragraaf blijft de vraag: kan de propositie 'Leverancier S5 levert geen enkel onderdeel' expliciet worden gerepresenteerd zónder een RVA te gebruiken? Natuurlijk is het antwoord: ja. Het enige dat we moeten doen is – in aanvulling op relatie sp (die laat zien welke leveranciers welke onderdelen leveren) – voorzien in een andere relatie snp die toont welke leveranciers helemaal geen onderdelen leveren. Zie afbeelding 3.

S#	P#	QTY
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

S#
S5

**Afbeelding 3:** Relaties sp en snp.

De afbeeldingen 1 en 3 zijn natuurlijk logische equivalenten in die zin, dat de relaties in afbeelding 3 kunnen worden afgeleid uit de relatie in afbeelding 1 en vice versa, door middel van expressies uit de relationele algebra. Er bestaan echter situaties

waarin er niet zulke equivalentie bestaat; er bestaan relaties met RVA's die geen equivalent hebben in termen van relaties zonder RVA's. Bijvoorbeeld: relatie sibs (gehaald uit referentie [5]) wordt getoond in afbeelding 4. De bedoelde betekenis van die relatie is dat de personen geïdentificeerd binnen elke gegeven PERSONS-waarde allemaal familie van elkaar zijn (en geen andere familie hebben). Op die manier zijn Amy en Bob familie van elkaar; Cal, Don en Eve zijn familie; en Fay is enig kind. Het enige attribuut van relatie sibs is een RVA.

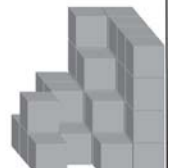
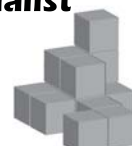
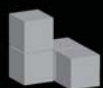
PERSONS
SIB
Amy
Bob
SIB
Cal
Don
Eve
SIB
Fay

**Afbeelding 4:** Relatie sibs.

# Bouw mee aan de In Summa oplossing

**SSAS, SSIS, SSMS, SSRS....**

**Zie jij kansen i.p.v afkortingen? Wil jij onze juniors & mediors al jouw SQL 2005 kennis bijbrengen? Bekijk onze vacature voor BI&DWH Specialist op [www.insumma.nl/specialist](http://www.insumma.nl/specialist)**



Nogmaals, er is geen relatie zonder een RVA die een logisch equivalent is van (dat wil zeggen, draagt exact dezelfde informatie als) relatie sibs.

Als we op de volgende wijze het attribuut PERSONS uit relatie sibs halen:

```
UNGROUP sibs ( PERSONS )
```

verkrijgen we de relatie zoals getoond in afbeelding 5, een relatie die duidelijk niet laat zien wie familie is van wie. (Met andere woorden, door het uiteennemen heeft het informatie verloren.)

SIB
Amy
Bob
Cal
Don
Eve
Fay

**Afbeelding 5:** Degroepering van relatie sibs op attribuut PERSONS.

Het is natuurlijk mogelijk om met een relatie te komen zonder RVA's die min of meer dezelfde informatie representeren als relatie sibs doet, zie afbeelding 6. Maar let op dat 'min of meer'! Het feit is dat de relaties in afbeelding 4 en 6 echt verschillende dingen voorstellen; om precies te zijn, de relatie in afbeelding 6 betekent echt dat de gespecificeerde personen 'behoren tot de gespecificeerde familie', een concept dat in afbeelding 4 helemaal niet voorkomt. Nog preciezer: er is geen relationele expres-

sie (in het algemeen) waardoor een relatie zoals in afbeelding 6 kan worden afgeleid uit een zoals in afbeelding 4.

FAMILY	SIB
Mozart	Amy
Mozart	Bob
Walton	Cal
Walton	Don
Walton	Eve
Dvořák	Fay

**Afbeelding 6:** een relatie die familieleden toont.

Nu hebben we dus een andere vraag van Gittens beantwoord: 'Kunnen deze proposities niet op een andere relationele manier worden voorgesteld?' Het antwoord is: ja, dat kan, maar niet altijd op een precies equivalente wijze. (In elk geval zien we dat een gegeven propositie meestal gerepresenteerd kan worden in vele verschillende relationele vormen, zelfs zonder RVA's te gebruiken – daarom komen verschillende ontwerpers met verschillende database-ontwerpen voor dezelfde informatie – en dus is de onderhavige vraag misschien niet zo belangrijk).

## OUTER JOIN overbodig

Gittens' eerste vraag is: welk probleem wordt er door het ondersteunen van deze attributen (RVA's) opgelost dat niet op een andere manier had kunnen worden opgelost?

Wel, we geloven dat we nu minstens één goed antwoord op deze vraag hebben gegeven. Maar er valt nog veel meer nuttigs over te zeggen. Stel dat we vinden dat het een goed idee is om RVA's te vermijden in basis relvars (een discipline die we toevallig ook



the-ict-job.nl

onderschrijven en waarover we later nog wat meer zullen zeggen). In overeenstemming met deze discipline zouden we een ontwerp kunnen bedenken voor leveranciers en leveringen die eruit ziet zoals de voorbeelden in afbeelding 7.

Wat opvalt is dat er in die afbeelding geen expliciete representatie is van het feit dat leverancier S5 geen onderdelen levert; in plaats daarvan verwijzen we – zoals we eigenlijk altijd doen – naar de Closed World Assumption om dat feit impliciet te herleiden.

S	S#	CITY	SP	S#	P#	QTY
	S3	Paris		S3	P2	200
	S4	London		S4	P2	200
	S5	Athens		S4	P4	300
				S4	P5	400

**Afbeelding 7:** Leveranciers en leveringen – conventioneel ontwerp.

Bekijk dan eens de volgende query: 'vind voor elke leverancier het leveranciersnummer, stad, geleverde onderdelen en bijbehorende hoeveelheden.'

De meest voor de hand liggende formulering voor deze query is:

```
S JOIN SP
```

Maar deze formulering levert natuurlijk niet op wat we willen, omdat leverancier S5 ontbreekt (het resultaat staat in afbeelding 8). Om die reden zal een gebruiker die bekend is met SQL waarschijnlijk de query in termen van een outer join formuleren, misschien als volgt:

```
S NATURAL OUTER JOIN SP
```

Afbeelding 9 toont het resultaat van deze SQL-expressie.

S#	CITY	P#	QTY
S3	Paris	P2	200
S4	London	P2	200
S4	London	P4	300
S4	London	P5	400

**Afbeelding 8:** Join van relaties uit afbeelding 7.

## Ambitieuze BI & ECM specialisten zoeken nieuwe collega's met passie voor hun vak



### Ambitieus

Onze medewerkers zijn ons grootste onderscheidend vermogen: ambitieuze (tool)specialisten met een afspraak = afspraak mentaliteit en passie voor hun vak. We hebben de ambitie om daar op korte termijn een aantal specialisten aan toe te voegen. Ben jij de collega die we zoeken?

### Passie

VLC is een ICT dienstverlener met passie voor Business Intelligence en Enterprise Content Management. Twee dynamische vakgebieden die steeds meer naar elkaar toegroeien. We volgen de ontwikkelingen op de voet en stoppen veel tijd en energie in het volgen van cursussen, het bezoeken van seminars, het lezen van vakliteratuur etc. Op deze manier zorgen we ervoor dat onze specialisten specialist blijven.

### Jij zoekt

Ambitieuze collega's, passie voor het vakgebied, uitdagende opdrachten, persoonlijke ontwikkeling & uitstekende arbeidsvoorwaarden.

### Jij bent

Een Business Intelligence en/of Enterprise Content Management specialist, met een afgeronde HBO/academische opleiding, goede communicatieve vaardigheden en minimaal 2 jaar ervaring met een of meer van de volgende BI tools: WebFOCUS, Business Objects, Cognos, PowerCenter, Oracle Warehouse Builder, CMS tools: Tridion, GX WebManager en/of ECM tools: Documentum.

### Interesse?

Stuur dan een reactie naar [vlc@vlc.nl](mailto:vlc@vlc.nl) of kijk op [www.vlc.nl](http://www.vlc.nl).



S#	CITY	P#	QTY
S3	Paris	P2	200
S4	London	P2	200
S4	London	P4	300
S4	London	P5	400
S5	Athens	-	-

- = nulls

**Afbeelding 9:** Outer join van relaties uit afbeelding 7.

Wat opvalt is dat dit resultaat geen relatie is (en we refereren er niet als zodanig aan) vanwege de nulls. (Zie [4] voor gedetailleerde argumenten die de zienswijze ondersteunen dat een 'relatie' die 'een NULL bevat' eigenlijk helemaal geen relatie is.) Als tegenstelling is hier een formulering van de query die wel een relatie als resultaat heeft – om precies te zijn de relatie getoond in afbeelding 10 met een RVA:

```
WITH ( SP RENAME ( S# AS SNO ) ) AS xyz :
EXTEND S ADD ( ( xyz WHERE SNO = S# ) { P#, QTY }
              AS PQ )
```

Uitleg: De eerste stap (met WITH) introduceert een tijdelijke naam xyz om een probleem met de naamgeving te voorkomen, die zich anders in de volgende stap zou kunnen voordoen; de naam xyz duidt een relatie aan die identiek is aan SP, behalve dat attribuut S# hernoemd is naar SNO.

De tweede stap breidt elke tuple van S uit met een extra attribuut, PQ genaamd, waarvan de waarde in elke tuple t een relatie is, afgeleid van xyz (eigenlijk van SP) en die P#/QTY-paren bevat voor alle leveringen corresponderend met het leveranciersnummer binnen die tuple t. We merken hierbij op, dat de expressie (xyz WHERE SNO = S#) { P#, QTY } in de tweede stap 'relation valued' is.

S#	CITY	PQ	
S3	Paris	P#	QTY
		P2	200
S4	London	P#	QTY
		P2	200
		P4	300
		P5	400
S5	Paris	P#	QTY

**Afbeelding 10:** relationele analogie van afbeelding 9.

In de relatie getoond in afbeelding 10 wordt de lege set van onderdeelnummer/hoeveelheid-paren horend bij leverancier S5 gerepresenteerd door een lege set – en niet door de een of andere rare NULL-constructie zoals in afbeelding 9. Om een lege set voor te stellen als een lege set is duidelijk een geweldig idee! Eigenlijk suggereert het voorbeeld, dat een OUTER JOIN eigenlijk helemaal niet nodig is als RVA's zouden worden ondersteund – een feit dat we ook als sterk argument in het voordeel van RVA's zien.

Einde deel 1 van 3.

#### 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](http://www.gittens.nl/TheTTMRevisited.pdf).

**Hugh Darwen en C.J. 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](http://www.dbm.nl) onder 'specials', 'extra materiaal'. In geval van discussies is de originele Engelse tekst doorslaggevend.