

The Third Manifesto: setting the Record Straight (3)

# Relation Valued Attributen (2)

H. Darwen en C. J. Date

**In deze bedrage – in drie stukken geknipt, waarvan dit het tweede 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.**

Op basis van de voorbeelden zoals we die in het vorige deel hebben gegeven, claimen we dat relation valued attributes (RVA's) zeker nuttig zijn voor query-doeleinden; met andere woorden, afgeleide relaties moeten tenminste RVA's kunnen hebben. Maar hoe zit dat met basisrelaties? Of preciezer: als we een database ontwerpen en vooral als we besluiten welke basis relvars die database gaat bevatten, mogen die relvars dan RVA's hebben? Noot: Basis relvars zijn de 'gegeven' relvars (dat zijn relvars die niet gedefinieerd zijn in termen van andere) en een basisrelatie is de waarde van een gegeven basis relvar op een bepaald moment. Zie referentie [9] voor verdere uitleg.

Het korte antwoord op de vraag is ja – in overeenstemming met het 'Principle of Interchangeability' [9], als relaties in het alge-

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 het tweede deel van de derde bijdrage in een serie.

meen RVA's mogen hebben, dan moeten juist de basisrelaties (en dus de basis relvars) ook RVA's kunnen hebben. Kunnen, want hoewel basis relvars zeker RVA's mogen hebben, voegen we daar direct aan toe dat zulke RVA's naar onze mening meestal afgeraden moeten worden; RVA's in basis relvars zijn meestal een slecht idee. Gedetailleerde argumenten die deze zienswijze ondersteunen kunnen worden gevonden in referentie [5].

Denk erom dat we het hier hebben over database-ontwerp items – of RVA's in basis relvars mogen voorkomen is een vraagstuk bij het database-ontwerp en gaat niet over wat wel en niet mag volgens de onderliggende theorie (of model). Database-ontwerp items zijn natuurlijk belangrijk, maar ze vallen buiten het bestek van TTM op zich; TTM houdt zich bezig met vraagstukken rond het ontwerpen van database-taal (en dus ook DBMS-ontwerp items) en niet met database-ontwerp items.

Ondanks het voorgaande willen we nog een paar dingen zeggen over het specifieke vraagstuk omtrent het ontwerp van de onderhavige database. We hebben al gesteld dat we dergelijke database-ontwerpen doorgaans niet zo'n goed idee vinden (en er staat niets in TTM dat anders suggereert). Maar we moeten duidelijk maken dat deze zienswijze alleen maar een leidraad is – het is geen wet van Meden en Perzen. Eigenlijk hebben we maar één tamelijk dwingend voorbeeld gevonden waarbij een basis relvar met een RVA precies het goede ontwerp lijkt te zijn. Het voorbeeld (dat afkomstig is uit referentie [5]) gaat over een catalog relvar KEYS die de relvars in de database en hun sleutels documenteert. Een voorbeeldwaarde van de catalog relvar wordt getoond in afbeelding 1; daarin nemen we aan dat: a. een van de relvars in de database MARRIAGE heet met als attributen HUSBAND, WIFE en DATE (trouwdag) en; b. relvar MARRIAGE drie onderscheidende sleutels heeft, die elk bestaan uit twee van de drie attributen.

We doen in deze paragraaf nog een laatste waarneming. Stel dat we zouden besluiten een database-ontwerp te aanvaarden waarbij we RVA's niet toestaan in basisrelaties maar wél in afgeleiden daarvan. Dan is die zienswijze analoog aan wat vele ontwerpers doen: welke basisrelaties zijn nodig om aan de beperkingen van, zeg, de Boyce/Codd normaalvorm te voldoen, maar de resultaten van query's doen dat meestal niet.

RELVAR	KEY			
MARRIAGE	<table border="1"> <tr><td>ATTRIB</td></tr> <tr><td>HUSBAND</td></tr> <tr><td>DATE</td></tr> </table>	ATTRIB	HUSBAND	DATE
ATTRIB				
HUSBAND				
DATE				
MARRIAGE	<table border="1"> <tr><td>ATTRIB</td></tr> <tr><td>DATE</td></tr> <tr><td>WIFE</td></tr> </table>	ATTRIB	DATE	WIFE
ATTRIB				
DATE				
WIFE				
MARRIAGE	<table border="1"> <tr><td>ATTRIB</td></tr> <tr><td>WIFE</td></tr> <tr><td>HUSBAND</td></tr> </table>	ATTRIB	WIFE	HUSBAND
ATTRIB				
WIFE				
HUSBAND				

**Afbeelding 1:** De catalogus-tabel KEYS – voorbeeld (en gedeeltelijke) waarde.

## Het ‘samenvatten’ van query’s

We hebben nóg een antwoord op de vraag ‘Welk probleem wordt er door het ondersteunen van deze attributen (RVA's) opgelost dat niet op een andere manier had kunnen worden opgelost?’ Het antwoord is: summarizing. Bijvoorbeeld: leg de volgende query eens naast het conventionele ontwerp voor leveranciers en leveringen zoals getoond in afbeelding 7 in het vorige deel in DB/M 7 (Een alternatieve en gebruiksvriendelijker formulering is: EXTEND s{S#} } PER sp ADD (SUM(QTY) AS SQ.):

```
SUMMARIZE sp PER ( s { S# } ) ADD ( SUM ( QTY )
AS SQ )
```

Deze expressie representeert de query: ‘vind van elke leverancier het leveranciersnummer en de totale bijbehorende geleverde hoeveelheid.’ Het resultaat, gegeven de voorbeeldwaarden in de bovengenoemde afbeelding 7, wordt getoond in afbeelding 2.

S#	QTY
S3	200
S4	900
S5	0

**Afbeelding 2:** Leveranciersnummers en totale geleverde hoeveelheid.

Intuïtief is duidelijk dat de semantiek in de zojuist getoonde expressie als volgt gedefinieerd is (in outline):

- Leid van relaties  $s$  en  $sp$  een relatie  $r$  af met attributen  $S\#$  en  $PQ$ , waarbij  $PQ$  een RVA is (het resultaat lijkt op dat van afbeelding 1 in het vorige deel in DB/M 7);
- Leid van relatie  $r$  het gewenste resultaat af door de aggregaat operator SUM toe te passen op het attribuut QTY van elk van de relaties die toevallig waarden zijn van de RVA  $PQ$ .

Met andere woorden: de SUMMARIZE operator wordt fundamenteel gedefinieerd in termen van RVA's.

Opmerking: Referentie [10], door Nikos Lorentzos en de huidige auteurs, bevat een gedetailleerde verzameling voorstellen voor de toepassing van het relationele model op het probleem van tijdelijke data. Een cruciaal aspect van die voorstellen is de definitie van twee nieuwe relationele operators, genaamd PACK en UNPACK. We noemen dit feit hier, omdat deze operators – net als SUMMARIZE – ook gedefinieerd worden in termen van RVA's.

## Zienswijze

Is onze zienswijze tegengesteld aan die van Codd? Gittens begint zijn commentaar met de stelling dat onze zienswijze om RVA's toe te staan in strijd zou zijn met de ideeën van Codd. Deze stelling is in feite niet accuraat; Codd veranderde in de loop der tijd van gedachten over dit probleem. In zijn eerste paper [1] stond hij wél RVA's toe. In zijn tweede [2] zei hij dat de mogelijkheid om ze te elimineren de moeite waard lijkt om te onderzoeken, maar hij drong niet aan op eliminatie. Zeker, in latere papers verbood hij RVA's; wij geloven echter dat zijn beweegredenen gebaseerd waren op een misvatting – namelijk dat het concept van ‘data value atomicity’ een of andere absolute betekenis had. Wij wijzen die redenen af en ondersteunen RVA's wel. Voor verder discussie hierover verwijzen we naar referentie [5].

## Een opmerking over constraints

Na zijn eerste vragen vervolgt Gittens: *[Voorgaande] vragen doen er toe, in het licht van het feit dat [waarden van] deze RVA's [relaties] zijn en als zodanig eraan verbonden kandidaat-sleutels missen. Bovendien kunnen RVA's geen rol spelen bij foreign key constraints zoals TTM dat wil. Dat wil zeggen dat alternatieven voor RVA's meer mogelijkheden hebben om accuraat databases te beheersen bij het vervullen van de eisen van de business en andere applicaties.*

We reageren als volgt op deze opmerking. Allereerst is het natuurlijk waar – per definitie! – dat RVA-waarden relaties zijn en geen relvars, en dat ze dus geen kandidaat-sleutels hebben (we noemen dat verder sleutels). Met het zeggen dat iets – zeg  $R$  – een sleutel heeft zeg je tegelijk dat: a.  $R$  een variabele is (opnieuw per definitie) en; b. dat updates van  $R$  op een bepaalde manier worden beperkt (ze mislukken als ze proberen een waarde toe te kennen aan  $R$  die niet voldoet aan de key constraint). Dus heeft een relatie – nogmaals: in de betekenis van relatie-waarde – per definitie geen sleutel. Het snijdt wel hout om van

een gegeven relatie te zeggen of hij wel of niet moet voldoen aan enige key constraint. We kunnen zelfs verder gaan en ruwweg zeggen dat als de onderhavige relatie moet voldoen aan een key constraint, dat die relatie dan die key 'heeft' – ofschoon een dergelijke uitspraak waarschijnlijk alleen maar voor verwarring zorgt, en we bevelen dat dus niet aan.

TTM vereist zeker ondersteuning van elke vorm van constraint die kan worden uitgedrukt in termen van relationele vergelijkingen, en de 'RVA constraints' waar Gittens zo bezorgd over is, kunnen dat zeer zeker; dus mist zijn bezorgdheid elke grond.

### TTM moet op zijn eigen merites worden beoordeeld

(Het is waar dat Tutorial D – in TTM [9] wordt als taal Tutorial D gebruikt als basis voor illustratieve voorbeelden – niet voorziet in syntactische afkortingen die het uitdrukken van zulke RVA constraints gemakkelijker maken, maar dat komt omdat wij van mening zijn – en we stelden dat al eerder – dat basisrelaties met RVA's ontmoedigd moeten worden. TTM verbiedt echter de toepassing van zulke afkortingen niet, zolang ze maar logisch en psychologisch goed ontworpen zijn).

#### Complicatie

Maken wij het relationele model ingewikkelder? Gittens vervolgt: *De vraag blijft dus: waarom zou het relationele model ingewikkelder gemaakt moeten worden om dubieuze winst te boeken ten opzichte van Codd's alternatief? Appendix B van TTM werkt verschillende mogelijkheden hiervan verder uit. Dit hoofdstuk weidt dan wel flink uit, maar het biedt weinig substantieels.*

*Ik citeer: "Wat is dan het criterium om iets een type te maken en geen relvar? Naar onze mening is deze vraag nog enigszins open".*

*Anders gezegd; Date en Darwen schijnen niet precies te weten wat de voordelen zijn van relation valued attributes ten opzichte van alternatieve oplossingen. De nadelen zijn duidelijk: Het werken met RVA's zoals geïntroduceerd wordt in TTM biedt minder mogelijkheden voor de expressie van constraints voor kandidaat-sleutels en foreign keys in databases, dan de alternatieven zonder RVA.*

*Is het niet aan Date en Darwen om met goede argumenten te komen waarom RVA's moeten worden toegevoegd aan het relationele model? Tot het moment dat de logische en valide voordelen van RVA's ten opzichte van de alternatieven kunnen worden aangetoond, lijkt het ondersteunen van zulke attributen, zoals in TTM wordt voorgesteld, niet alleen op het zoeken naar problemen, maar ook een nodeloze en zinloze complicatie.*

*Dientengevolge, gegeven de huidige stand van zaken rond dit*

*item, blijf ik erbij dat ondersteuning van RVA's zoals gedefinieerd in TTM strijdig is met op z'n minst de 'parsimony requirement' van RM voorschrift 26.*

Ten eerste kan beargumenteerd worden dat het toestaan van RVA's niet alleen het relationele model niet compliceren, maar zelfs vereenvoudigen, door een restrictie weg te halen bij de toegestane attribuut-typen. (We hebben al geconstateerd dat Codd's regel dat attributen 'atomic' typen kunnen hebben, naar onze mening niet overleefd blijft omdat het hele concept van 'data value atomicity' al niet stand houdt). Wat betreft de 'dubieuze winst' zijn we van mening dat we een paar van de voordelen van het werken met RVA's in de voorafgaande paragrafen van dit artikel hebben laten zien.

Ten tweede raakt Gittens' verwijzing naar Appendix B van TTM kant noch wal. Die appendix heeft helemaal niets van doen met RVA's. De titel is 'Een ontwerp dilemma' en heeft te maken met vragen van de volgende aard: Stel dat we te maken hebben met employees, waarbij elke employee een employee-nummer heeft (EMP#), een naam (ENAME), een afdelingsnummer (DEPT#) en een salaris (SALARY). Het is duidelijk dat we dan een EMP type kunnen definiëren (Design T) of een EMP relvar (Design R). De vraag die we in deze appendix behandelen is: welke aanpak is beter? Met andere woorden zijn er redenen om de een boven de ander te verkiezen?

Hoewel men dus inderdaad zou kunnen claimen dat de appendix "weinig substantieels biedt" (hoewel we het daar dus helemaal niet mee eens zijn), kan Gittens' conclusie dat "Date en Darwen niet precies schijnen te weten wat de voordelen zijn van relation valued attributes ten opzichte van alternatieve oplossingen" daar absoluut niet uit getrokken worden. We weten dat wél precies en we illustreren dat in dit artikel. Sterker nog, we hebben ook aangetoond dat Gittens' nadelen slechts *schijnbaar* waar zijn.

Gittens vervolgt: "Het werken met RVA's zoals geïntroduceerd wordt in TTM biedt minder mogelijkheden voor de expressie van constraints voor kandidaat-sleutels en foreign keys in databases, dan de alternatieven zonder RVA." Zoals we hebben laten zien is ook deze claim onjuist.

Gittens: "Is het niet aan Date en Darwen om met goede argumenten te komen waarom RVA's moeten worden toegevoegd aan het relationele model?"

Welnu: ten eerste, we voegen ze helemaal niet toe. Het relationele model bevat ze al vanaf het begin; het is waar dat ze vervolgens verwijderd zijn, maar wij vinden dat dat op onjuiste gronden is gebeurd.

Ten tweede, we hebben jaren geleden in referentie [4] al 'goede argumenten' in het voordeel van RVA's gegeven. Het is niet onze bedoeling om die argumenten in TTM nog eens te herhalen; TTM is een gedetailleerd voorschrift voor database-taal en DBMS-ontwerp, gebaseerd op principes en zienswijzen die ergens anders op adequate wijze zijn gedocumenteerd.

TTM moet op zijn eigen merites worden beoordeeld. Als we ieder aspect van elk voorschrift apart zouden moeten rechtvaardigen in TTM zelf, zou het een nog veel immenser omvang hebben dan het al heeft.

Tot slot stelt Gittens: "Dientengevolge, gegeven de huidige stand van zaken rond dit item, blijf ik erbij dat ondersteuning van RVA's zoals gedefinieerd in TTM strijdig is met op z'n minst de 'parsimony requirement' van RM voorschrift 26".

Nee, dat is het helemaal niet, zoals afdoende is aangetoond in dit artikel en elders.

Einde deel 2 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 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](http://www.dbm.nl) onder 'specials', 'extra materiaal'. In geval van discussies is de originele Engelse tekst doorslaggevend.

# Hier geen nummer

# 0800-5432101

Werken bij Valid is werken voor een ICT dienstverlener waar persoonlijke aandacht nog de normaalste zaak van de wereld is. Voor onze collega's én voor onze klanten. Bij Valid krijg je wat je verdient: uitdagende projecten bij toonaangevende klanten, een uitmuntend salaris, een uitdagend bonussysteem en een individueel budget voor opleidingen en trainingen.

Ben je een ervaren BI Consultant, Oracle Software Engineer of DBA en toe aan een op het lijf geschreven uitdaging in Utrecht, Eindhoven of Maastricht? Neem dan contact op met Bart Meex via bovenstaand telefoonnummer of mail je CV naar [work@valid.nl](mailto:work@valid.nl).

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

