

Redundantie genuanceerd gebruiken bij modelleren van de werkelijkheid

Normaliseren? Voor wie niet kan modelleren!

Gerrit Wolters

In dit artikel wil ik enkele noten kraken ten aanzien van de toepassing van de normalisatiestappen zoals die in 1970 door Codd zijn geïntroduceerd. Een bezinning is naar mijn mening gewenst omdat het normaliseren gemakkelijk leidt tot minder handige modellen. Minder handig? Ja. Ook modellen die ontstaan op basis van normalisatieregels zijn maar een afbeelding van de werkelijkheid, zij geven niet de werkelijkheid weer.

Er zijn voor mij twee redenen om de resultaten van het normalisatieproces eens kritisch te bezien. Ten eerste een gevoel dat de wijze waarop we doorgaans zaken als orders en bestellingen modelleren eenvoudiger kan. In de tweede plaats de (betwiste) noodzaak van datawarehouses naast productiesystemen. Dat domweg normaliseren op basis van formulieren en andere overzichten niet verstandig is, wordt vrij algemeen onderschreven. Met name zaken als generalisaties¹ worden bij zo'n aanpak vaak niet onderkend. In dit artikel wordt daar niet verder op ingegaan. Ik richt me op de twee genoemde punten en met name het eerste. Zij hebben gemeenschappelijk dat het begrip redundantie genuanceerd wordt.

ORDERS EN/OF ORDERREGELS

In cursussen voor het normaliseren van gegevens is een van de schoolvoorbeelden het normaliseren van orders of bestellingen.

Jan Krediet	20 juli 2002			
Rijksweg 12				
1234 AB Amsterdam				
Artikelcode	Omschrijving	Aantal	Prijs p/s	Totaal
A12	Whiteboardmarkerset	4	€ 5,00	€ 20,00
F7	Whiteboard 1,5 * 1 m	1	€ 24,95	€ 24,95
Totaal				€ 44,95

FIGUUR 1.

In figuur 1 is een bon weergegeven van een fictieve order. De vertaling van deze bon volgens de normalisatiestappen leidt dan doorgaans tot een gegevensmodel zoals afgebeeld² in figuur 2 en waarvan figuur 3 de schematische weergave is.

Volgens het normalisatieproces is dit een correct model. Doordat uitgegaan is van een bepaald orderformulier, is het resultaat echter inrichtingsafhankelijk en onvolledig. Een ander gegevensmodel voor het modelleren van orders zou kunnen worden weergegeven door figuur 4. Dit model geeft aan dat voor elk besteld artikel een aparte order wordt onderkend. Ik zal de verschillen tussen deze twee zienswijzen bespreken.

Eerst moeten we definiëren wat een order is. Wie of wat bepaalt wat een order is? Ik wil de volgende twee definities in bespreking brengen⁶:

1. Een order is het verzoek van een KLANT tot het leveren van artikelen van een of meer ARTIKELSOORTEN.
 2. Een order is het verzoek van een KLANT tot het leveren van artikelen van een ARTIKELSOORT.
- Het onderwerp van bespreking is niet of

we nu spreken over een verzoek, een opdracht of een overeenkomst, of over de gebeurtenis van het plaatsen van de order, maar het gaat ons om de vraag of er sprake is van één artikelsoort of van meer artikelsoorten. Een keuze voor de ene of de andere definitie (en daarmee het bijbehorende plaatje) is niet goed of fout; eerder meer of minder handig.

De tweede definitie wordt vaak als eerste gegeven. Het is de definitie die past bij het tweede plaatje. De eerste definitie past bij het eerste plaatje.

ORDER EN ORDERREGEL

Welke gegevens horen er bij een order volgens de eerste definitie (en het eerste model)?

- klant-id
- orderdatum (ordertijdstip)
- ordernummer

Ordernummer is de sleutel. Een kunstmatige sleutel; semantisch gezien vormen de klantidentificatie en de orderdatum, al dan niet met het tijdstip waarop de order is

Basisvorm³:

Klantnaam, klantadres, orderdatum, {artikelcode, artikelnaam, aantal, prijs per stuk, totaal}, totaalbedrag}

1^e normaalvorm:

Order klantnaam⁴, klantadres, orderdatum, totaalbedrag
 Orderregel artikelcode, artikelnaam, aantal, prijs per stuk, totaal, klantnaam, orderdatum

2^e normaalvorm (hier tevens 3^e normaalvorm):

Klant klantnaam, klantadres
 Order klantnummer, orderdatum, totaalbedrag
 Orderregel artikelcode, aantal, prijs per stuk⁵, totaal, klantnaam, orderdatum
 Artikelsoort artikelcode, artikelnaam, prijs per stuk

FIGUUR 2.

geaccepteerd, de sleutel. Het entiteitstype bestaat op deze manier uit slechts sleutelgegevens, en het heeft geen zelfstandig bestaansrecht. Orders zonder orderregels bestaan niet. Andere gegevens die opgenomen zouden kunnen worden zijn:

- afleverdatum
- afleveradres
- orderidentificatie van de klant
- identificatie van het orderdocument

Voor het afleveradres en de afleverdatum geldt dat zij nogal inrichtingsafhankelijk zijn. Door ze bij de order op te nemen, geeft men aan dat ze voor de hele order gelden. Als voor een bepaald artikel een afwijkende leverdatum of een afwijkend adres geldt, zal daarvoor een nieuwe order moeten worden opgenomen. Door deze gegevens als eigenschappen van de orderregel op te nemen, bereik je een grotere mate van inrichtingsonafhankelijkheid. De orderidentificatie van de klant en de identificatie van het orderdocument zijn verwijzingen naar onderwerpen die soms wel en soms niet zelf onderwerp van administratie zijn. De discussie of ze bij

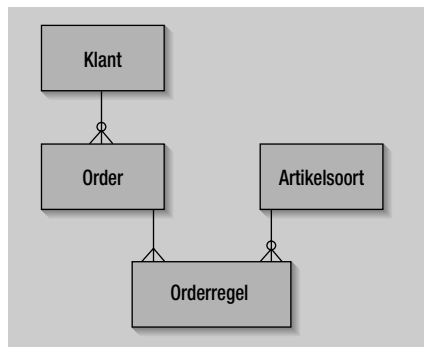
de order dan wel bij de orderregel horen, heeft slechts relevantie voor het begrip redundantie. Dit onderwerp komt later in dit artikel aan de orde.

Verder zou je bij gegevens voor een order kunnen denken aan:

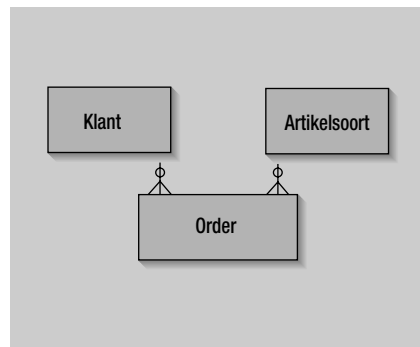
- orderstatus
- statusdatum

Deze gegevens zijn van groot belang voor het volgen van de order tijdens de verwerking binnen het bedrijf. Ze zijn afhankelijk van (en afleidbaar uit) de processen⁷ die er op de orderregels hebben plaatsgevonden. Door ze als gegeven van de order op te nemen, gelden ze voor alle bijbehorende orderregels en zijn ze minder nauwkeurig dan wanneer ze per orderregel worden bijgehouden: de order is pas afgehandeld als alle orderregels zijn afgehandeld; het betekent ook dat de order in zijn geheel, of niet geannuleerd kan worden.

Voor het orderpicken, en in zekere mate ook voor het afleveren van een order, is de oorspronkelijke samenstelling van de order niet van belang: het gaat om de orderregels.



FIGUUR 3.



FIGUUR 4.

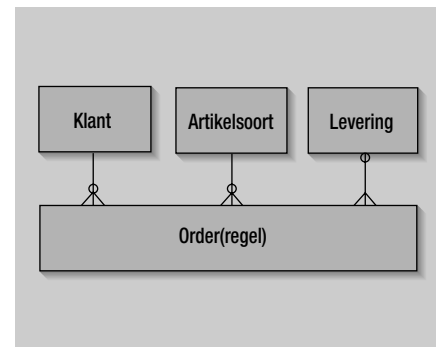
Tenslotte worden aan een order (en andere entiteitstypen) vaak nog de volgende gegevens gekoppeld:

- registratiedatum
- user-id

Dit zijn echter niet zozeer gegevens over de order, maar over de behandeling en de beschrijving van de order. Ook voor deze gegevens geldt dat zij de werkelijkheid preciezer beschrijven als zij per orderregel worden opgenomen: voor nabestellingen speelt dan de vraag, of zij bij de oorspronkelijke order opgenomen moeten en kunnen worden, dan niet meer.

De conclusie van het bovenstaande is dat het entiteitstype ORDER geen wezenlijk bestaansrecht heeft naast orderregel omdat het van zichzelf geen onderwerp van gesprek is: het bevat alleen sleutels of gegevens die inrichtingsafhankelijk zijn (en ook anders of elders kunnen worden vastgelegd). Door de inrichtingsafhankelijkheid worden we gedwongen om in situaties die (nog) niet gangbaar zijn, nieuwe orders te registreren terwijl het toch om één order gaat. Bijvoorbeeld wanneer een klant één artikelsoort op verschillende dagen of locaties afgeleverd wil hebben. Bovendien zijn we voor het raadplegen van orders genoodzaakt om twee tabellen te raadplegen, iets dat onnodig complex is. Voor de verwerking van de order is in veel gevallen de oorspronkelijke order nauwelijks van belang en gaat het steeds om de orderregels.

Het tweede model (zie figuur 4) lijkt dus duidelijk voordelen te bieden boven het eerste, klassieke model. Zijn er ook nadelen aan dat tweede model? De volgende bezwaren worden er vaak primair tegen in gebracht:



FIGUUR 5.

Klant-id	Artikel-soort-id	Orderdatum	Hoeveelheid besteld	Order bedrag	Postcode	Huis-nummer	Gewenste afleverdatum	Ordernummer klant
J1	A123	02-09-2002	20	160	1234 AB	10	15-09-2002	20023245
J1	A234	02-09-2002	2	235	1234 AB	10	15-09-2002	20023245
J1	A235	02-09-2002	1	235	1234 AB	10	15-09-2002	20023245
E43	B763	04-09-2002	150	75	4527 NM	2	16-09-2002	H216
E43	C45	04-09-2002	200	850	4527 NM	2	16-09-2002	H216

FIGUUR 6.

1. Het strookt niet met de werkelijkheid of het beeld van de gebruiker;
2. Het strookt niet met het beeld van de klant;
3. Het leidt tot redundantie.

De tabel ORDER zou er uit kunnen zien zoals in figuur 6. De orderdatum, het afleveradres, de gewenste afleverdatum en het ordernummer klant lijken redundant en zouden 'uitgenormaliseerd' kunnen worden. Hiermee winnen we echter niets en dus laten we dat achterwege.

HET BEELD VAN DE GEBRUIKER OP DE ORDER

Allereerst het tegenargument dat het tweede model niet aansluit bij de wereld van de gebruiker. Dit is waar voorzover de wereld van 'de gebruiker' nog is gevormd door de klassieke orderformulieren. De verandering is echter vrij natuurlijk en niet groot. Voor de aansluiting op het eigen inkoop-, productie- en leveringsproces is de verandering echter zeer logisch omdat bij die processen verschillende orders gecombineerd kunnen worden. De volgende verschillen zijn te merken:

1. Toename van het aantal orders.
Dit is een logisch gevolg van de nieuwe zienswijze. Het aantal orders neemt toe zonder dat de omzet stijgt.
2. Verandering van de gegevensverwerking. Orderinvoer: het registratieproces wordt niet arbeidsintensiever. Desgewenst kunnen de invoerschermen zelfs hetzelfde blijven. Het gegevenstype 'ordernummer' is niet meer nodig. Het raadplegen, wijzigen en verwijderen van orders zal lichtelijk wijzigen: er moet nu gerefereerd worden aan de 'orderregel' in plaats van aan de order. In de praktijk gebeurde dat echter ook al,

omdat die bewerkingen betrekking hebben op een orderregel. Feitelijk gaat het om een vereenvoudiging.

Het is overigens nog steeds mogelijk om orders die gelijktijdig door een klant zijn geplaatst, in de gegevensverzameling te herkennen en terug te halen. Dit kan zelfs zonder gebruik te hoeven maken van een kunstmatig ordernummer. De communicatie met de klant kan dus ook eenvoudiger plaatsvinden.

Als een klant een bepaalde levertijd afspreekt voor een combinatie van artikelen, dan zal het gegeven dat aangeeft wanneer de orders uitgeleverd worden, voor die verschillende orders gelijk moeten zijn. Het gaat hier om de relatie naar de levering. Desgewenst kunnen ook later geplaatste orders aan diezelfde levering gerelateerd worden. Ook in de situatie waarin orderregels worden onderkend, zal niet de order, maar zullen de orderregels aan de levering gerelateerd worden (al dan niet in een n:m relatie), zie figuur 5.

HET BEELD VAN DE KLANT OP DE ORDER

De klanten communiceren met ons over hun orders en als wij ons beeld aanpassen, dan zal dat gevolgen hebben voor de klant. Hoe wij de orders administreren is voor de klant echter niet van belang. De communicatie wordt zonder het onderkennen van orderregels echter eenvoudiger: we hoeven de klant niet meer te vertellen dat hij in zijn correspondentie of mondelinge contacten het ordernummer moet geven. En als de klant zelf ook gebruik maakt van een eigen ordernummer, dan kunnen wij dat gegeven ook nog kwijt in onze administratie.

In het geval dat er kortingsregelingen worden toegepast op basis van de grootte van tegelijkertijd uitgebrachte orders, dan kan dat nog steeds. Het kortingsbedrag is echter niet zozeer een gegeven van de order, maar van de kortingsregeling. Kortingsregelingen zijn altijd inrichtingsafhankelijk. Hoe meer elementair de order wordt onderkend, hoe specifiekere en eenvoudigere kortingsregelingen ook kunnen worden ingevoerd.

REDUNDANTIE

Als tegenargument voor het vereenvoudigde model wordt ook de toename van redundantie genoemd. Daar zullen we nu op ingaan. Hebben we eigenlijk wel zo'n hekel aan redundantie en waarom? Redundantie is overtolligheid. Het duidt op iets dat niet noodzakelijk, maar wel handig is. Het is dus niet gelijk aan overbodigheid. Iets is overbodig als het teveel is en gemist kan worden. Het niet noodzakelijk zijn wordt in administraties veroorzaakt door het feit dat een gegeven ook al elders bekend is, of afgeleid kan worden uit andere gegevens. Voor de beheerder van gegevens is redundantie lastig, of in ieder geval een extra aandachtspunt, omdat hij ervoor moet zorgen dat de gegevens consistent zijn. Voor de gebruikers van de gegevens is redundantie echter handig en informatief. Gebruikers willen van een orderregel ook weten bij welke klant zij hoort en op welke datum zij geplaatst is. In de klassieke wijze van modelleren moet daarvoor altijd een ander entiteitstype geraadpleegd worden.

Is er eigenlijk wel sprake van redundantie? Welke gegevens zijn er redundant in het vereenvoudigde model? Als we order en orderregel samenvoegen, zouden

de gegevens van order redundant kunnen zijn. Het gaat dan om gegevens als klantnummer, orderdatum en misschien een orderidentificatie van de klant. Deze gegevens zijn echter niet redundant. Immers: ze komen weliswaar voor bepaalde orders overeen, maar ze zullen nooit wijzigen omdat het generieke gegevens van de order zijn. De order kan niet aan een andere klant gehangen worden, en de orderdatum kan niet wijzigen⁸. Er kunnen dus geen 'verwerkingsanomalieën' plaatsvinden. In het voorbeeld is ook de prijs van het artikel opgenomen als eigenschap van de order(regel). Ook dit gegeven is zoals aangegeven niet redundant. Ook andere gegevens, zoals bijvoorbeeld de NAW-gegevens van de klant of de artikelnaam zouden op gelijke wijze meegenomen kunnen worden; zij geven de toestand aan zoals die was op het moment van bestellen. Er is geen redundantie. Het blote feit dat gegevens met elkaar overeenkomen, zegt dus nog niet dat er sprake is van redundantie.

DATAWAREHOUSES EN PRODUCTIESYSTEMEN

Ik denk dat we het er over eens zijn dat genormaliseerde modellen niet geschikt zijn voor datawarehouse toepassingen; deze toepassingen zijn juist mede ontstaan doordat de genormaliseerde modellen niet preformeerden bij omvangrijke queries. Tegelijkertijd zijn er ook deskundigen die van mening zijn dat datawarehouses niet nodig zijn naast de gewone productiesystemen, mits deze goed ontworpen zijn. Het grote verschil tussen productiesystemen en datawarehouses ligt met name in de 'redundantie' in de datawarehouses. Datawarehouses zijn vaak 'gedenormaliseerd': er wordt weer redundantie aan toegevoegd. Het normalisatieproces, zoals dat door Codd is bedacht, is een abstracte, wiskundige benadering. Het belangrijkste doel hierbij was het voorkomen van redundantie; met name in verband met het voorkomen van verwerkingsanomalieën (inconsistenties ten gevolge van gegevensmutaties die onvoorziene gevolgen hebben voor andere gegevens). In de praktijk

wordt vaak elk dubbel vastgelegd gegeven als redundant bestempeld. Denk bijvoorbeeld aan artikelprijs in order(regel). Deze 'dubbel vastgelegde' gegevens zijn lang niet altijd redundant, ten minste niet in de zin van: gevoelig voor verwerkingsanomalieën. Ze geven in veel gevallen een moment of toestand weer. Er is geen gevaar van inconsistenties, hooguit van verkeerd gebruik: de prijs van een artikel in een order is niet de prijs van het artikel, maar de prijs van het artikel zoals die voor de order gold of gevraagd werd.

Dit soort van redundantie was vroeger kostbaar omdat ook de opslagcapaciteit duur was. Het 'uitnormaliseren' van dit soort van gegevens was dan ook meer dan meegenomen. Nu speelt dat echter niet meer. Door dit soort van gegevens niet meer uit te normaliseren of zelfs toe te voegen, wordt de noodzaak van datawarehouses voor een belangrijk deel verkleind. Bovendien geldt dat toepassingen op het productiesysteem in bepaalde gevallen sneller kunnen worden omdat ook daar de raadplegingen eenvoudiger worden.

CONCLUSIES

Bij het modelleren van gegevens is het van belang om te abstraheren vanuit de inrichting. Het uitgaan van bestaande documenten houdt het risico van inrichtingsafhankelijkheid in. Het begrip redundantie dient genuanceerder te worden gebruikt bij het modelleren van de werkelijkheid. Door een order te definiëren als 'het verzoek van een KLANT tot het leveren van artikelen van een ARTIKELSOORT', (in plaats van ARTIKELSOORTEN) ontstaat een eenvoudiger, herkenbaar en correct model. Dit heeft zowel voor de functionele als de technische kant voordelen:

- specificaties en applicatiecomponenten worden eenvoudiger;
- applicaties worden sneller;
- groter flexibiliteit zonder toename van de complexiteit.

Doordat er minder tabellen zijn, zijn er minder of eenvoudiger beheerfuncties nodig. Er vervalt een relatief zware integriteitsregel (een verplichte 1:n relatie). De argumentatie die hier gegeven wordt

voor orders, geldt ook voor bestellingen, leveringen en betalingen. Deze entiteiten zijn door de voorgestelde vereenvoudiging ook beter aan elkaar te relateren.

Ook voor andere situaties waarin een verplichte 1:n relatie voorkomt, geldt vermoedelijk vaak bovenstaande redenering. Te denken valt bijvoorbeeld aan ledenadministraties waar de NAW-gegevens van leden die op één adres in gezinsverband wonen, worden 'uitgenormaliseerd'. Ook invoer- en uitvoeraangiftes bij de douane (waar op één aangifte één of meer in of uit te voeren artikelsoorten vermeld staan), komen hiervoor in aanmerking. Saillant detail hierbij is dat de wet de aangifte ook definieert als betrekking hebbend op één artikelsoort. Het modelleren van gegevens is vooral een zaak van *common sense*. Toepassing van normalisatiestappen zou hooguit iets voor beginners moeten zijn, maar heeft het risico van onnodige complexiteit. Vaak is men in eerste instantie wel gecharmeerd van de constructie van orderregel als naam voor de herhalende groep bij order, maar het houdt iets gekunstelds. ●

Gerrit Wolters (gerrit.wolters@ordina.nl) is als Gegevensarchitect en Informatiekundige werkzaam bij Ordina Public B.V.

Noten

- 1 Te denken valt bijvoorbeeld aan het generaliseren van LEVERANCIERS en KLANTEN tot RELATIES, of het generaliseren van AFDELINGEN en DIVISIES tot ORGANISATORISCHE EENHEDEN.
- 2 Sleutelgegevens worden onderstreept.
- 3 De afgeleide totaalbedragen van regels en order worden gewoon meegenomen.
- 4 Klantnaam wordt als identificatie aangenomen, dit is meestal een nummer, dat echter niet voorkomt op het orderformulier.
- 5 De prijs van het artikel die voor de order geldt, is op het moment van ontstaan van de orderregel wel afleidbaar, maar later niet meer (tenzij er historie van de artikelprijs wordt bijgehouden), en dus niet redundant.
- 6 We gaan in de bespreking uit van klantorders, voor eigen bedrijfsorders geldt hetzelfde.
- 7 Bijvoorbeeld: 'Orderpicken', afleveren en factureren.
- 8 Tenzij bij fouten. Daarvoor zijn echter verschillende en eenvoudige oplossingen denkbaar.