

Een gebruikersgericht onderzoek naar relationele database management systemen

Citation for published version (APA):

Remmen, F., Heemstra, F. J., & Heijst, van, J. G. M. (1987). Een gebruikersgericht onderzoek naar relationele database management systemen. *Informatie*, 27(11), 956-963.

Document status and date:

Gepubliceerd: 01/01/1987

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

EEN GEBRUIKERSGERICHT ONDERZOEK NAAR RELATIONELE DATABASE MANAGEMENT SYSTEMEN

door ir. F. J. Heemstra, drs. J. G. M. van Heijst en
drs. F. Remmen.

Het aanbod van database management systemen met het voorvoegsel 'Relationeel' wordt alsmat groter. Bij de gebruiker wordt de indruk gewekt dat deze systemen de oplossing bieden voor zijn informatieproblemen. Zo wordt beweerd dat met behulp van een krachtige en gebruiksvriendelijke vraag- en onderhoudstaal allerlei soorten gegevens, die op eenvoudige wijze in de database gedefinieerd kunnen worden, snel opgevraagd en gewijzigd kunnen worden. In vorige artikelen, gepubliceerd in dit tijdschrift, zijn reeds kritische kanttekeningen geplaatst bij deze optimistische beweringen. In dit artikel wordt een onderzoek beschreven dat als doel heeft een aantal aspecten, te weten definitiekracht, constraintbewaking en gebruiksvriendelijkheid van een aantal Relationele Database Management Systemen (RDMS'en) kritisch op hun werkelijke waarde te toetsen. In dit onderzoek is de niet gebruikelijke weg gevolgd van een casus die voorgelegd is aan leveranciers van RDMS'en. De eerste, voorlopige conclusies die uit de resultaten van het onderzoek getrokken mogen worden zijn dat met name het definiëren en bewaken van constraints (nog) een zwakke schakel vormt in deze pakketten en bovendien dat bij complexere vraagstellingen de querytaal in formuleringskracht vermoedelijk te kort schiet.

1 INLEIDING

Vele leveranciers van een Data Base Management Systeem bieden momenteel hun produkt aan met het adjectief 'relationeel'. Deze groei in het aanbod van (relationele) Data Base Management Systemen is voor de gebruiker (de consument) een plezierige ontwikkeling: zijn keuzemogelijkheden worden groter. Anderzijds kleeft aan deze groei ook een nadeel: de keuze welk 'databasepakket' voor een bepaald bedrijf of voor een bepaalde toepassing het meest geschikte is wordt erdoor bemoeilijkt. Bij de aanschaf van een databasepakket, zowel voor kleine(re) als grote(re) systemen, bestaat steeds meer de tendens om de selectie te doen plaatsvinden uit het aanbod 'Relationele' Data Base Management Systemen (RDBMS).

Van de klassieke DBMS'en (hiërarchisch en netwerk) is bekend, dat een gebruiker dient te beschikken over uitgebreide analytische en programmeertechnische vaardigheden om bepaalde gegevens uit een database op te vragen. Betekent het inslaan van een 'relationele' weg nu, dat de gebruiker op eenvoudige wijze kan voorzien in zijn informatiebehoefte? Zijn de queries inderdaad zo eenvoudig als wel gesuggereerd wordt? Zijn de RDBMS'en eigenlijk wel in staat om gegevensstructuren met door de gebruiker gedefinieerde constraints te realiseren?

De hoofddoelstelling van het onderzoek, dat in deze bijdrage wordt beschreven, is een antwoord proberen te krijgen op de onderstaande vier kernvragen:

1. Hoe groot is de 'definitiekracht' van de 'rationele' DBMS'en?
2. Kunnen de constraints bewaakt worden door het RDBMS?
3. Hoe 'user-friendly' is de definitietaal/querytaal/onderhoudstaal?

4. Welke 'built-in' functies worden ondersteund door het RDBMS?

Bovendien trachten we een (enigszins) bevredigend antwoord te vinden op meer algemene vragen zoals:

- Hoe is in de praktijk de performance van het RDBMS bij grotere databases?
- Wat verstaat de leverancier van het RDBMS onder de term 'relationeel'?

Om het begin van een antwoord te kunnen geven op bovengenoemde kernvragen is een casus opgesteld en verstuurd naar een aantal leveranciers van RDBMS'en. In deze casus (zie paragraaf 3) wordt een eenvoudige database beschreven. De leverancier wordt verzocht deze database te creëren en te vullen met door ons verstrekte gegevens. Vervolgens dienen een 18-tal queries en onderhoudsopdrachten op deze database uitgevoerd te worden door de leverancier. Op basis van de reacties kan een antwoord worden gegeven op de vier kernvragen, die aan dit onderzoek ten grondslag liggen.

Bij de opzet van dit onderzoek hebben wij gekozen voor de vorm van een casus met een aantal query- en onderhoudsopdrachten. Op deze wijze hebben wij (op een eenvoudige manier) de praktische situatie willen uitbeelden, waarin vele gebruikers zich geplaatst zien. Een gebruiker heeft op de eerste plaats een informatieprobleem en vraagt zich af hoe en met welke middelen dit probleem kan worden opgelost. De eigenschappen van RDMS'en is niet primair voorwerp van onderzoek geweest. Via diverse enquêtes zijn deze eigenschappen uitgebreid vastgelegd. Een bekend voorbeeld is het onderzoek van Schmidt en Brodie [Schm-Br 83].

In paragraaf 4 wordt een toelichting gegeven op genoemde query- en onderhoudsopdrachten. Er wordt aangegeven wat de achterliggende bedoeling van elke opdracht is, waarom de betreffende opdracht in het onderzoek is

opgenomen. In paragraaf 5 gaan we inhoudelijk in op de zeer uitvoerige reactie van Computer Associates met het DB-pakket CA-Universe. Een summier bespreking van een aantal andere, korte, reacties staat beschreven in paragraaf 5.3. In paragraaf 6 verwoorden we een algemene conclusie en een aantal aanbevelingen voor verder onderzoek.

2 TOELICHTING OP DE KERNVRAGEN

Alvorens in paragraaf 3 de opzet en uitvoering van het onderzoek meer in detail te beschrijven, willen wij een toelichting geven op de 4 genoemde cruciale vragen. De eerste kernvraag heeft betrekking op de definitiekracht van een RDBMS, op de mogelijkheid tot structuurbeschrijving. Met andere woorden kan een door de gebruiker gedefinieerde gegevensstructuur met de daarbij behorende constraints worden gerealiseerd door het RDBMS?

In kernvraag twee, die onlosmakelijk verbonden is met de eerste kernvraag, wordt de problematiek van de bewaking van constraints centraal gesteld. In hoeverre is het mogelijk dat de gewenste constraints door het RDBMS worden bewaakt; of zal deze bewaking via applicatie-programma's moeten worden gerealiseerd met, indien dat niet goed gebeurt, het risico van gegevensvervuiling [Rem 85,1] In dit kader is het van belang onderscheid te maken tussen een viertal soorten constraints [Rem 82] te weten:

1. **Attribuutconstraints**
Deze soort beperking maakt het mogelijk dat bepaalde waarden voor een attribuut worden uitgesloten. In de casus in paragraaf 3 wordt o.a. het volgende voorbeeld van zo'n constraint gegeven:
Het klantnummer kan variëren van 1 t/m 350.
2. **Tupelconstraints**
Met behulp van dit soort constraints wordt het mogelijk combinaties van attribuutwaarden uit te sluiten. Tupelconstraints zijn derhalve constraints van een hogere orde dan attribuutconstraints. Een voorbeeld uit de casus is:
Van het artikel drop is de prijs 2,20 of 7,10. Het gewicht van het artikel bedraagt minder dan 5000.
3. **Tabelconstraints**
Dit soort constraints zijn wéér van een hogere orde en maken het mogelijk om een bepaalde verzameling tupels (records) in een tabel uit te sluiten. Bijvoorbeeld: er zijn binnen een artikelbestand minimaal 2 artikelen met de artikelomschrijving Drop. (Dat wil zeggen dat al die tabellen, waarin het artikel drop 0 of 1 maal voorkomt, geen toegestane tabellen in de database zijn).
Een andere, meer bekende tabelconstraint luidt: 'het nummer van de klant is uniek en fungeert als sleutel'. Dat wil zeggen, dat een klantentabel, waarin twee verschillende records voorkomen met hetzelfde klantnummer, niet toegestaan is.
4. **Databaseconstraints**
In een informatiesysteem bestaan er meestal verbanden tussen tupels in verschillende tabellen. Zo bestaat er in de casus een relatie tussen het klantenbestand en het orderbestand.
Bijvoorbeeld: bij elk orderrecord behoort een klantentablet aanwezig te zijn (met dezelfde waarde

voor het attribuut klantnummer): Onbekende klanten kunnen geen order plaatsen.

Beperkingen, die opgelegd worden aan relaties tussen verschillende tabellen of tupels uit verschillende tabellen, worden databaseconstraints genoemd. In het bovengenoemd voorbeeld wordt een bepaald soort databaseconstraint gegeven, n.l. een subset-requirement.

Het moge duidelijk zijn dat constraints die niet in het RDBMS gedefinieerd kunnen worden, ook niet door datzelfde RDBMS bewaakt kunnen worden. Als zodanig zouden de kernvragen twee en drie gecombineerd kunnen worden tot een vraag.

In de casus komen onderhoudsopdrachten voor waarbij bewaking op de vier genoemde constraints een rol speelt.

In kernvraag drie worden een aantal aspecten van de query en onderhoudstaal centraal gesteld, met name:

- a. In hoeverre kunnen alle operaties uit de relationele algebra worden weergegeven. (Een van de voorwaarden die Codd [Schm-Br 83] noemt om een RDBMS de titel 'fully relational' te geven).
- b. Is het mogelijk om met behulp van de querytaal/ onderhoudstaal in kwestie de genoemde opdrachten te formuleren of wanneer worden zoals Reitsma [Rei 85] het formuleert, de grenzen van de query- en onderhoudstaal overschreden. Vooral bij de meer ingewikkelde query-opdrachten uit de casus hadden wij vooraf het bange vermoeden dat de mogelijkheden van de querytaal wel eens te kort zouden kunnen schieten.
- c. Als het mogelijk is om met behulp van de query- en onderhoudstaal de opdracht(en) te formuleren, is deze formulering dan eenvoudig en, vanuit gebruikers standpunt gezien, snel te leren en te gebruiken. Met andere woorden hoe gebruiksvriendelijk is de querytaal in de toepassingsfeer. Het is uiteraard moeilijk objectieve normen vast te stellen voor wat betreft dit aspect van gebruiksvriendelijkheid. Bij de formulering van een query voor een complexe raadpleging spreekt het voor zich dat er een groot beroep wordt gedaan op de analytische vaardigheden van een gebruiker. De uiteindelijke formulering van een dergelijke query zal, evenals de nederlandse vraagstelling, complex zijn en bestaan uit een nesting van verschillende subqueries. Het is ons inziens niet terecht om voor dergelijke queries een hoge mate van gebruiksvriendelijkheid te eisen. Anders ligt het wanneer ogenschijnlijk eenvoudige en voor de hand liggende vraagstellingen alleen met complexe query formuleringen te realiseren zijn. Tegen deze achtergrond zal het aspect gebruiksvriendelijkheid in de toepassingsfeer worden belicht.

Kernvraag vier heeft als doelstelling na te gaan welke built-in functions door het RDBMS worden ondersteund, of het mogelijk is functies als minimum, maximum, gemiddelde, som e.d. in de queryformulering op te nemen.

Tenslotte wordt aan het einde van de casus een aantal algemene vragen gesteld. Een belangrijk aspect, dat in deze vragen terloops wordt aangesneden is het aspect

performance; een uiterst belangrijk criterium bij de beoordeling van een RDBMS. Dit aspect valt evenwel buiten de doelstelling van dit onderzoek (en zal door ons voorlopig niet verder worden bestudeerd).

We konden niet nalaten de leveranciers van RDBMS'en te vragen naar hun interpretatie van het begrip relationeel; wat er zo relationeel aan hun RDBMS is.

3 ONDERZOEK

3.1 Inleiding

Het onderzoek is uitgevoerd in de vorm van een casus met een opdrachten-, c.q. vragenlijst. Deze casus, zo veel mogelijk ontdaan van elk theoretisch en/of wetenschappelijk tintje, is verzonden naar een aantal leveranciers van RDBMS'en. Er is geen onderscheid gemaakt in pakketten voor kleine(re) en grote(re) computers. We zijn ervan overtuigd dat de dure databasepakketten meer mogelijkheden in zich hebben dan de goedkopere pakketten voor bijv. een microcomputer: het uitgevoerde onderzoek is zeer zeker geen vergelijkend warenonderzoek. Als gebruiker zitten we met een aantal vragen, die we opgelost willen hebben door middel van queries op een door ons gedefinieerde database! Het uitgangspunt van het onderzoek is dus de 'gebruikersbehoefte', die op een fictieve manier tot uiting wordt gebracht in een casus met een database van geringe omvang en geringe complexiteit. Afhankelijk van de reacties op dit onderzoek kan een uitgebreider onderzoek worden overwogen.

De leveranciers, die wij verzocht hebben om aan het onderzoek mee te werken, is gevraagd om dit eenvoudig voorbeeld van een database in te voeren met behulp van hun 'relationeel' databasepakket. (In par. 3.2 wordt dit voorbeeld in casusvorm gepresenteerd). Vervolgens is hen gevraagd een aantal query- en onderhoudsopdrachten (par. 3.3) op de database (stap voor stap) uit te voeren. Op basis van de ontvangen reacties kan men dan op meer objectieve wijze een antwoord geven op de kernvragen 1, 2 en 4 (resp. definitiekracht, constraints bewaking en beschikbaarheid van built-in functions) uit paragraaf 1. T.a.v. kernvraag 3 (te weten gebruiksvriendelijkheid) is de interpretatie van de verstrekte antwoorden op de queries niet ontdaan van enige subjectiviteit: de normen voor 'user-friendly-queries' zijn nog niet op een goede manier gedefinieerd!

3.2 Casus met opdrachten- en vragenlijst

De casus met de opdrachten- en vragenlijst is verstuurd naar 14 leveranciers van RDBMS'en. In een begeleidend schrijven is uiteengezet welke de doelstellingen van het onderzoek zijn en op welke wijze de ontvangen reacties gebruikt zullen worden.

Casusbeschrijving

Een handelsbedrijf maakt gebruik van een geautomatiseerd systeem om de gegevens over klanten, orders en artikelen te registreren. Hiertoe wordt gebruik gemaakt van een database met vier bestanden, namelijk een klantenbestand, orderbestand, orderregelbestand en artikelbestand.

Hieronder worden de attributen (items) van de verschillende records beschreven samen met de beperkende voorwaarden (we noemen dit de 'constraints') waaraan

de attributen, de records, de bestanden en de database moeten voldoen.

Per recorditem worden achtereenvolgens gegeven: symbolische naam, korte omschrijving, numeriek/alfanumeriek (N/A) en aantal posities.

Klantenbestand

Recordindeling:

item	omschrijving	N/A	aantal posities
KLNR	klantnummer	N	3
KLNAAM	naam van de klant	A	30
KLWPL	woonplaats van de klant	A	30
KRW	kredietwaardigheid	N	1

Constraints:

1. De klantnummers kunnen variëren van 1 t/m 350.
2. Als het klantnummer (KLNR) kleiner of gelijk aan 100 is, dan komt de klant uit Eindhoven.
3. Het klantnummer is uniek, d.w.z. KLNR fungeert als sleutelgegeven.
4. Het aantal klanten uit Breda is (altijd) kleiner dan het aantal klanten van buiten Breda.
5. KRW kan de waarde 1, 2, 3, 4 of 5 hebben. Een hogere waarde betekent een betere kredietwaardigheid.

Orderbestand

Recordindeling:

item	omschrijving	N/A	aantal posities
ORDNR	ordernummer	N	5
ORDDAT	orderdatum	N	6
KLNR	klantnummer	N	3
ORDBEDR	orderbedrag	N	6,2

(onder 6,2 wordt verstaan: 6 cijfers voor en 2 cijfers achter de 'komma').

Constraints:

1. Het ordernummer (ORDNR) is uniek, dus ORDNR fungeert als een key voor het orderbestand.
2. De combinatie van orderdatum (ORDDAT) en klantnummer (KLNR) is ook uniek. In feite betekent dit, dat een klant op een bepaalde dag hoogstens een order kan plaatsen.
3. Een klant mag nooit meer dan 5000,00 aan totaalbedrag bij uitstaande orders hebben.

Orderregelbestand

Recordindeling:

item	omschrijving	N/A	aantal posities
ORDNR	ordernummer	N	5
REGNR	regelnummer	N	1
ARTNR	artikelnummer	N	5
AANTAL	aantal stuks besteld	N	3

Constraints:

1. Het regelnummer varieert van 1 t/m 9.
2. De combinatie ORDNR, REGNR fungeert als sleutel voor het orderregelbestand.

Artikelenbestand**Recordindeling**

item	omschrijving	N/A	aantal posities
ARTNR	artikelnummer	N	5
ARTOM	artikelomschrijving	A	30
ARTPR	artikelprijs	N	6,2
ARTGEW	artikelgewicht	N	5

Constraints:

1. Van het artikel drop is de prijs 2,20 of 7,10. Het gewicht van het artikel bedraagt minder dan 5000.
2. Het artikelnummer (ARTNR) fungeert als key voor het artikelbestand.
3. Er zijn minimaal 2 artikelen met de artikelomschrijving 'drop'.

Naast bovenvermelde 'constraints', die beperkingen opleggen aan items, records en bestanden, dienen in het handelsbedrijf de opgeslagen gegevens ook nog te voldoen aan de volgende voorwaarden:

1. Bij elk orderregel, die opgeslagen is, hoort (uiteraard) ook een orderrecord en een artikelrecord opgeslagen te zijn in de desbetreffende bestanden.
2. Zo hoort ook bij elk orderrecord een klantenrecord aanwezig te zijn: onbekende klanten kunnen geen orders plaatsen.
3. Het orderbedrag van een klant met een kredietwaardigheid kleiner dan 3 moet onder 1000,00 liggen.
4. Elke order bevat minimaal 1 orderregel.

Voorbeeld database**Klantenbestand**

KLNR	KLNAAM	KLWPL	KRW
201	Jansen	Breda	1
312	Sax	Tilburg	2
30	Donker	Eindhoven	5
134	V. Dun	Breda	4
347	Karels	Tilburg	3
234	Pieters	Amsterdam	5

Artikelenbestand

ARTNR	ARTOM	ARTPR	ARTGEW
101	drop	2,20	3000
102	pepermunt	3,50	2250
103	drop	7,10	4100
104	jujubes	4,10	2000
105	rang	3,70	2500
106	kauwgom	6,80	4000
107	lolly	1,50	1500
108	marsepein	9,20	6500

Orderbestand

ORDNR	ORDDAT	KLNR	ORDBEDR
301	830619	312	708,00
302	830519	201	110,00
303	830520	312	633,00
304	830723	201	572,00
305	830724	134	184,00
306	830802	30	112,00
307	830812	347	287,00
308	830814	234	552,00

309	830907	234	1068,00
310	831101	30	159,00
311	831102	347	90,00

Orderregelbestand

ORDNR	REGNR	ARTNR	AANTAL
301	1	103	80
301	2	102	40
302	1	101	50
303	1	101	30
303	2	102	20
303	3	103	70
304	1	104	40
304	2	106	60
305	1	108	20
306	1	104	10
306	2	103	10
307	1	104	70
308	1	108	60
309	1	106	90
309	2	101	80
309	3	102	80
310	1	101	40
310	2	103	10
311	1	107	60

3.3 Queries- en onderhoudsopdrachten

Voordat u de volgende opdrachten gaat uitvoeren dient u de bestanden, zoals hiervoor omschreven te creëren en te vullen met de gegevens uit het voorbeeld.

Bovendien dient u, voor zover dat in uw databasesysteem mogelijk is, *alle* constraints in te bouwen in het model.

U wordt verzocht om onderstaande opdrachten in de aangegeven volgorde uit te voeren met de voor uw database(management) systeem gehanteerde *querytaal* (vraagtaal) en/of *onderhoudstaal*.

Wij verzoeken u vriendelijk om van elk van onderstaande queries te geven:

1. de formulering van de query, waarmee u de opdracht hebt uitgevoerd;
 2. het antwoord op de query, zoals dit gegeven wordt door uw database-systeem.
01. Geef de nummers en omschrijvingen van de artikelen met een prijs van meer dan 2,00.
 02. Geef een overzicht van de woonplaatsen van alle klanten, waarbij een bepaalde woonplaats in het overzicht maar *een* keer mag voorkomen.
 03. Geef een overzicht van alle orders met een orderdatum in augustus 1983 en wel zodanig, dat in het overzicht bij elke order ook de bijbehorende orderregels zijn opgenomen.
 04. Geef de nummers en de omschrijvingen van alle artikelen, waarvoor in augustus 1983 een order is geplaatst.
 05. Geef het ordernummer, de orderdatum en het orderbedrag van de orders waarop het artikel jujubes of het artikel pepermunt is besteld.
 06. Geef het ordernummer, de orderdatum en het klantnummer van de orders, waarop *niet* het artikel drop is besteld.

07. Geef de ordernummers en de orderbedragen van de orders, waarop wel pepermunt is besteld, maar geen kauwgom.
08. Geef van elke klant: klantnummer, naam van de klant, laagste orderbedrag, hoogste orderbedrag en het gemiddelde orderbedrag.
09. Geef nummer en naam van de klanten, die een artikel hebben besteld met een prijs lager dan 5,00, maar die geen enkel artikel hebben besteld met een gewicht lager dan 2500.
10. Geef de namen van de klanten uit Breda, die in mei 1983 geen enkele order geplaatst hebben, en geef tevens (per klant) het totaal aantal artikelen, dat door hem besteld is.
11. Geef de nummers en namen van de klanten, die in het tweede kwartaal van 1983 een order hebben geplaatst voor een artikel, dat in het derde kwartaal van 1983 niet is besteld door een klant uit Breda of Eindhoven, tenzij die klant in het vierde kwartaal van 1983 een artikel heeft besteld met een prijs van meer dan 5,00.

De volgende reeks opdrachten heeft betrekking op een aantal mutaties in de bestanden van de database.

Wij verzoeken u vriendelijk van elk van onderstaande onderhoudsopdrachten te geven:

1. de formulering van de 'onderhoudsopdracht', in uw databasesysteem;
2. het antwoord op de 'onderhoudsopdracht', zoals dit gegeven wordt door uw database-systeem.

Bovendien verzoeken wij u om *na het uitvoeren van elk der onderstaande opdrachten* een uitdraai te geven van de complete inhoud van de bestanden in de database.

12. De volgende klantgegevens moeten toegevoegd worden aan het klantenbestand:
 - KLNR : 444
 - KLNAAM : Derks
 - KLWPL : Den Bosch
 - KRW : 5

13. De volgende order dient toegevoegd te worden aan de reeds opgeslagen gegevens:
 - ORDNR : 312
 - ORDDAT : 831212
 - KLNR : 347
 - ORDBEDR : 1375,00

Deze order bevat de volgende twee orderregels:

- ORDNR : 312
- REGNR : 1
- ARTNR : 108
- AANTAL : 80

- ORDNR : 312
- REGNR : 2
- ARTNR : 103
- AANTAL : 90

14. De order met ordernummer 301 wordt geannuleerd. De desbetreffende order dient uit de database verwijderd te worden.
15. De volgende klantgegevens moeten aan het klantenbestand worden toegevoegd:
 - KLNR : 22
 - KLNAAM : Fritsen
 - KLWPL : Den Bosch
 - KRW : 2

16. Verwijder het artikelrecord van artikelnummer 106.
17. Voeg een nieuw artikelrecord toe. Van het artikel zijn de volgende gegevens bekend:
 - ARTNR : 104
 - ARTOM : caramel
 - ARTPR : 3,60
 - ARTGEW : 7000

18. De volgende order dient toegevoegd te worden aan de reeds opgeslagen gegevens:
 - ORDNR : 315
 - ORDDAT : 831101
 - KLNR : 30
 - ORDBEDR : 370,00
 Deze order bevat de volgende orderregel:
 - ORDNR : 315
 - REGNR : 1
 - ARTNR : 105
 - AANTAL : 100

3.4 Algemene vragen

1. In ons voorbeeld is gebruik gemaakt van 'constraints': beperkingen die opgelegd worden aan attributen, records en bestanden of aan de gehele database. Voorziet uw databasesysteem in dergelijke constraints en zo ja, welke?
2. Voorziet uw database(management)systeem ook in zogenaamde built-in functions, zoals bijv. sum, average, maximum en minimum? Zo ja, wilt u dan een opsomming geven van de mogelijkheden? Zo nee, hoe pakt u dan de daarmee samenhangende queries aan?
3. Het database-voorbeeld aan de hand waarvan u de erbij horende opdrachten heeft uitgevoerd, is uiteraard vrij klein van omvang. Veronderstel nu, dat we in ons handelsbedrijf te maken krijgen met een vrij omvangrijke database. Problemen, die dan gaan ontstaan, liggen op het terrein van de 'performance', bijv. is de responstijd nog acceptabel bij allerlei ad-hoc queries. Welke voorzieningen zijn/moeten/kunnen in uw databasesysteem getroffen (worden) om de performance in de hand te houden? Zijn er testen uitgevoerd met betrekking tot de performance van uw database(management)systeem? Zo ja, wilt u dan de desbetreffende testresultaten ons mededelen of aangeven waar de testresultaten te verkrijgen zijn.
4. Uw database(management)systeem is geannoneerd als 'relationeel'. Wat verstaat u onder de term 'relationeel'? Wat zijn volgens u de typische kenmerken van een relationeel databasesysteem?
5. Kunt u een kort overzicht geven van de mogelijke opdrachten in uw querytaal (vraagtaal) en/of onderhoudstaal?
6. Op welke computer(s) en onder welk operating system(s) draait uw database(management)systeem en wat is de prijs van het pakket?
7. De kans is vrij groot dat dit onderzoek voortgezet wordt. Bent u bereid mee te werken aan een uitgebreider onderzoek, waarbij een 'echte' database (± 3 MB) op tape aangeleverd wordt?

Tot slot nog een vriendelijk verzoek om zo gedetailleerd mogelijke produktinformatie in te sluiten.

Wij danken u hartelijk voor uw bereidwillige medewerking.

4 TOELICHTING OP DE OPDRACHTEN- EN VRAGENLIJST

In paragraaf 1 van dit artikel zijn een viertal kernvragen geformuleerd, die aan dit onderzoek ten grondslag liggen. Middels de query- en onderhoudsopdrachten, beschreven in de opdrachtenlijst (par. 3.3), trachten wij een antwoord op de vier kernvragen te krijgen. In deze paragraaf willen wij aangeven welke opdracht(en) een antwoord zouden moeten geven op welke kernvraag/-vragen; m.a.w. wat de achterliggende bedoelingen van de verschillende opdrachten zijn.

In de opdrachtenlijst wordt expliciet gevraagd om de bewuste tabellen(bestanden) te creëren en te vullen met de gegevens uit het voorbeeld. Bovendien wordt gevraagd om *alle* constraints, voor zover mogelijk, in het model in te bouwen. De uitvoering van deze opdracht kan aldus een antwoord geven op de eerste kernvraag in paragraaf 1, betreffende de definitiekracht van een RDBMS.

In kernvraag twee wordt de bewaking van de constraints door het RDBMS centraal gesteld. Met behulp van de opdrachten 12 t/m 18 uit de opdrachtenlijst wordt nagegaan of en zo ja, op welke wijze de constraintbewaking wordt gerealiseerd. Van elk van deze zeven opdrachten zullen wij aangeven op welke constraintbewaking wordt gecontroleerd.

Opdracht 12

Het klantenrecord met klantnummer 444 mag niet opgenomen worden in het klantenbestand. Er is namelijk een de attribuutconstraint gedefinieerd, die stelt dat de waarde van een klantnummer ligt tussen 0 en 351.

Opdracht 13

Controle op de bewaking van een databaseconstraint. Een klant met een kredietwaardigheid kleiner dan 3 mag niet meer dan 1000 aan openstaande orderbedragen hebben. De bewuste order (met een orderbedrag > 1000) heeft betrekking op een dergelijke klant, met andere woorden de bewuste order, met bijbehorende orderregels, mag niet worden opgenomen in de database.

Opdracht 14

Controle op de bewaking van een databaseconstraint. Bij verwijdering van de order met ordernummer 301, moeten automatisch ook de twee orderregels van de zojuist verwijderde order uit het orderbestand worden verwijderd.

Opdracht 15

Controle op de bewaking van een tupelconstraint. Toevoeging van dit klantenrecord in het klantenbestand is in strijd met de beperking dat een klant met een klantnummer kleiner dan 100 uit Eindhoven komt.

Opdracht 16

Een soortgelijke controle als in opdracht 14. Bij elk orderregelrecord behoort een artikelrecord opgeslagen te zijn. Het artikelrecord mag niet worden verwijderd omdat er nog orderregels voor dat artikel in de database zitten.

Opdracht 17

Een controle op de bewaking van de tabelconstraint die eist dat het attribuut artikelnummer in het artikelbestand fungeert als sleutel. Het bewuste artikelrecord mag derhalve niet worden toegevoegd; het artikel met artikelnummer 104 ligt immers al opgeslagen in het artikelbestand.

Opdracht 18

Ook deze opdracht controleert op de bewaking van een tabelconstraint, namelijk dat de combinatie van de waarde van de attributen ORDNR en KLNDR uniek is in het orderbestand.

In kernvraag drie wordt de formuleringskracht van de querytaal centraal gesteld. Met behulp van de eerste elf opdrachten uit de opdrachtenlijst trachten wij hierin inzicht te krijgen. Van elk van deze opdrachten wordt aangegeven op welk aspect van de formuleringskracht van de querytaal een beroep wordt gedaan. De elf opdrachten zijn gerangschikt naar oplopende complexiteit; zo wordt in de eerste drie opdrachten achtereenvolgens gecontroleerd of het mogelijk is eenvoudige selectie-, projectie- en joinformuleringen te realiseren. De daarop volgende opdrachten kunnen alleen gerealiseerd worden door in de query formulering gebruik te maken van een combinatie van een aantal relationele operatoren en van subqueries. Een extra moeilijkheid vormen die opdrachten waarbij in de vraagstelling een ontkenning is gebruikt (bijvoorbeeld: geef nummers en namen van de klanten, die *wel* ..., maar *niet* ...). In Rem [Rem 85,2] wordt deze problematiek ook uiteengezet. Er zijn ook opdrachten opgenomen waarbij het uiteindelijke resultaat van de query aan een aantal voorwaarden is gebonden; Con-junctie en disjunctie van voorwaarden (bijvoorbeeld: geef nummers en namen van de klanten, die voldoen aan ... *en* ... *en* ... *of* ... *en* *niet* ...) In dit voorbeeld zien we een combinatie van ontkenning, conjunctie en disjunctie. Juist door het stellen van deze minder eenvoudige vragen wordt het interessant om na te gaan of de veel gehoorde (commerciële) slogan: 'Een essentieel onderdeel van een RDBMS is een gebruiksvriendelijke, 'relational-based' en zeer krachtige vraagtaal' overeind kan blijven.

Met behulp van opdracht 8 en 10 wordt nagegaan welke built-in functions door het RDBMS worden ondersteund; het centrale thema van kernvraag vier. Met behulp van opdracht 8 wordt nagegaan of de built-in functions minimum, maximum en gemiddelde gerealiseerd kunnen worden en via opdracht 10 wordt deze controle uitgevoerd voor de built-in function som. In de lijst met algemene vragen (paragraaf 3.4) wordt expliciet gevraagd welke built-in functions gerealiseerd kunnen worden.

5 REACTIES VAN LEVERANCIERS

5.1 Inleiding

We hebben moeten constateren dat er maar weinig leveranciers van RDBMS'en zijn geweest, die ondanks een

herhaalde aansporing van onze kant, de uitdaging om hun medewerking te verlenen aan dit onderzoek hebben aangenomen. Een gunstige uitzondering vormt Computer Associates. In een zeer uitgebreide reactie verschaft deze leverancier ons een uitstekend inzicht in de mogelijkheden van het relationeel database management pakket CA-Universe, door op uiterst nauwkeurige wijze de richtlijnen bij de opdrachten in de opdrachtenlijst op te volgen. In paragraaf 5.2 wordt uitvoerig aandacht besteed aan de evaluatie van het pakket CA-Universe. Daarnaast hebben de leveranciers CBA-computerservice (pakket: Optimum DBMS), IBM (pakket: IBM DB2) en CINCOM Systems Benelux (pakket: Ultra) gereageerd. Deze reacties beperken zich echter tot definiëring van de tabellen, formulering van de queries en onderhoudsopdrachten en een kort antwoord op de algemene vragen. Antwoorden van het systeem op een bepaalde query/onderhoudsopdracht worden helaas niet gegeven. Op de problematiek van constraintformulering en -bewaking wordt in het geheel niet gereageerd. In paragraaf 5.3 worden deze reacties kort besproken.

5.2 CA-Universe

1. Definitiekracht

* Door CA-Universe wordt alle informatie in de vorm van tabellen opgebouwd.

* Inbouwen van constraints in het RDBMS.

- attribuutconstraints

Bij de definitie van de tabellen wordt van elk attribuut het type, het maximaal aantal karakters per attribuut, waarde, aantal cijfers voor en achter de decimale punt opgegeven. Beperkingen op het waardebereik van een attribuut (bijv. $0 \leq \text{KLNR} \leq 350$) worden bij definiëring van de tabellen opgenomen in het model.

- tupelconstraints

De tupelconstraints uit de vragenlijst kunnen niet opgenomen worden in het database model

- tabelconstraints

De tabelconstraints die betrekking hebben op het definiëren van de uniciteit van een attribuutwaarde, kunnen wel gerealiseerd worden, zij het indirect door middel van het definiëren van een index. Merkwaardig was evenwel dat er problemen ontstonden bij het definiëren van een andere key (in de ordertabel). Alle overige tabelconstraints uit de vragenlijst kunnen echter niet worden ingebouwd in het model.

- databaseconstraints

Hiervoor geldt hetzelfde als de tupelconstraints; Computer Associates maakt evenwel de aantekening dat een aantal van deze constraints in een toekomstige release wel opgenomen kunnen worden.

Alle 'gemiste' constraints kunnen (zoals al vooraf werd vermoed) op applicatieniveau worden ingebracht. We zijn bijzonder gecharmeerd van de introductie van variabelen bij de opdrachtformuleringen. Op deze wijze wordt het mogelijk de formulering kort en kernachtig te houden.

2. Constraintbewaking

opdracht 12: attribuutconstraint wordt bewaakt.

opdracht 13: de bewuste order en bijbehorende orderre-

gels worden probleemloos in de database opgenomen. Dit ligt in de lijn der verwachtingen. De bewuste database constraint kan niet worden ingebouwd in het model. Bewaking door het RDBMS is dan ook niet mogelijk.

opdracht 14: omdat ook deze database constraint niet kan worden ingebouwd in het model, zal de gebruiker een expliciete opdracht moeten geven om de bewuste orderregels te verwijderen.

opdracht 15: bij de uitvoering van deze opdracht treedt een soortgelijke situatie op als bij uitvoering van opdracht 13.

opdracht 16: evenals bij opdracht 14 moet bij uitvoering van deze 'verwijder record'-opdracht een bewaking van een niet gedefinieerde database constraint worden gerealiseerd. Een en ander wil zeggen dat bij verwijdering van het artikel met nr. 106, de gebruiker *zelf* de bewakingsfunctie moet overnemen door de orderregels met betrekking tot artikelnummer 106 uit de database te verwijderen.

opdracht 17: op deze opdracht reageert het pakket CA-Universe correct. Het bewuste artikel wordt vanwege de tabelconstraint niet in de database opgenomen.

opdracht 18: bij uitvoering van deze opdracht treedt het eigenaardige verschijnsel op dat de bewuste order niet en de bijbehorende orderregels wel in de database worden opgenomen. Door gebruikmaking van een 'kunstgreep' zorgt CA-Universe ervoor dat het resultaat van dergelijke onlogisch uitgevoerde opdrachten ongedaan wordt gemaakt.

3. Formuleringkracht querytaal

- Bij de uitvoering van zowel opdracht 2 als 3 ontstaat het probleem om de waarde van respectievelijk plaatsnaam en ordernummer maar éénmaal in het overzicht te laten voorkomen, ondanks de queryformulering *retrieve unique*.

- CA-Universe heeft niet in het minst problemen met de formulering en uitvoering van queries, waarin meerdere operaties en subqueries noodzakelijk zijn. Ook met opdrachten waarin een ontkenning in de vraagstelling is gebruikt, heeft CA-Universe alerminst moeite. Zelfs opdracht 11 wordt, teliswaar met een complexe queryformulering, (volgens onze verwachtingen in) probleemloos uitgevoerd. Opvallend is tevens de eenvoud van de queryformulering. Ook de complexere opdrachten (met uitzondering van opdracht 11) kunnen kort en inzichtelijk in een queryformulering worden omgezet.

4. Built-in functions

Door het RDBMS CA-Universe wordt een groot aantal built-in functions ondersteund (count, sum, average, minimum, maximum, any en tevens countunique, sumunique en average-unique). Met de formulering en de uitvoering van de query 8, die expliciet een beroep doet op deze built-in functions, blijkt CA-Universe echter problemen te hebben. Pas na een aantal pogingen en 'met het werken naar de goede oplossing' kan de opdracht met succes worden uitgevoerd.

5. Algemene vragen

- Constraints kunnen slechts in beperkte mate in het model worden ingebouwd.

- Performance problemen treden op wanneer niet verstandig te werk wordt gegaan bij
 1. zeer grote relaties (meer dan 100 000 records).
 2. joins op meer dan 3 relaties en/of
 3. zgn. nested aggregate functions.
 Opgemerkt wordt dat er gewerkt wordt aan faciliteiten ten behoeve van query-optimalisatie.

5.3 Overige reacties

- Uit de beantwoording van de vragen blijkt dat constraints slechts in beperkte mate in het model kunnen worden ingebouwd. Evenals bij CA-Universen kunnen bij DB2 en Ultra alleen constraints worden opgelegd aan attributen en de uniciteit van sleutels. Bij Optimum ontbreekt zelfs de eerste mogelijkheid. Controle op naleving van deze constraints was gezien de aard van de reacties niet mogelijk. Opvallend is wel dat bij beantwoording van de opdrachten 12 t/m 18, juist die onderhoudsopdrachten niet geformuleerd zijn, waarbij problemen met constraintbewaking te verwachten waren.
- De querytalen van het pakket Ultra en SQL van DB2 blijken in problemen te komen bij de formulering van de queries voor de complexere opdrachten uit de reeks 1 t/m 11. Wat dat betreft onderschrijven wij volledig de ervaringen van Reitsma [Rei 85] met SQL. Het pakket Ultra zoekt al zeer snel haar toevlucht tot applicaties bij de formulering van de queries. Van Optimum zijn geen queryformuleringen ontvangen.
- Zowel in Ultra als SQL zijn built-in functions voorhanden, hoewel Ultra niet wijst op de beschikbaarheid van de unique-variant.
- Alle reacties zijn vaag in hun antwoord op de vraag wat verstaan moet worden onder het begrip 'relationeel'. Definities van Codd [Schm-Br 83] en Date [Date 77] worden aangehaald. Nog vager worden de antwoorden op de vraag naar wat er zo relationeel is aan het pakket in kwestie. Begrijpelijk, gezien de algemene worsteling om het begrip relationeel eenduidig te definiëren.

6 CONCLUSIES

Het is uiteraard jammer te moeten constateren dat de response op de voorgelegde casus vooralsnog mager te noemen is. Van de 4 leveranciers, die benaderd zijn, hebben er slechts 4 gereageerd. Als motivering wordt veelal aangedragen dat de bereidheid er wel is, maar niet de beschikbare mankracht, dat het pakket deze vorm van reclame niet nodig heeft of dat men geen interesse heeft in het onderzoek. Hopelijk geeft deze publikatie aanleiding tot meer reacties.

Gezien de geringe mate van response dragen onze conclusies een zeer voorlopig karakter. De belangrijkste conclusie die tot nu toe uit de resultaten van het onderzoek getrokken kan worden is dat het inbouwen van tuple-, tabel en databaseconstraints in het model niet of nauwelijks realiseerbaar is. Bewaking van dergelijke constraints door het RDBMS is derhalve niet mogelijk.

Een en ander houdt in dat deze bewaking gerealiseerd zal moeten worden door applicatieprogramma's. Verzaakt een van deze applicaties in zijn bewakingsplicht, dan staat (in de bewoordingen van Remmen [Rem 85,1]) de deur voor gegevensvervuiling wagenwijd open.

Een tweede, meer voorzichtige, conclusie is dat bij complexere vraagstellingen de querytaal in formuleringkracht te kort schiet. Door de teneur van het verhaal zou de indruk gewekt kunnen worden dat RDBMS'en alles behalve deugdelijke produkten zouden zijn. We wensen dit met klem te ontkennen. Bij de evaluatie moet men er rekening mee houden dat het hier vaak gaat om produkten waarvan nog maar een of twee versies zijn uitgebracht en die gebaseerd zijn op een geheel nieuwe technologie. Het is te verwachten, en uit de reacties die we hebben ontvangen kwam dat ook naar voren, dat er steeds meer verbeteringen zullen worden aangebracht. Een vergelijking met niet-relatieve systemen is ons inziens niet geheel terecht; alleen al om het simpele feit dat deze systemen circa 15 jaren in gebruik zijn en gedurende die tijd steeds verbeterd en aangepast zijn. Bovendien moet betwijfeld worden of er nog belangrijke verbeteringen zijn aan te brengen in niet-relatieve systemen, terwijl dat ongetwijfeld bij hun relationele concurrenten wél het geval is [Date 84].

Wat ons inziens wel overeind blijft staan is de bewering dat bij potentiële gebruikers van RDBMS'en hogere verwachtingen gewekt worden dan op dit moment door de RDBMS'en waar gemaakt kunnen worden.

Vlak voor het moment van publikatie ontvingen wij van Computer Associates de resultaten van de uitvoering van enkele opdrachten met behulp van de nieuwe release van het pakket CA-Universen.

- De aanvankelijke problemen met betrekking tot opdracht 8 zijn nu opgelost.
- Eenvoudige tupleconstraints (zoals in opdracht 15) kunnen worden geformuleerd en bewaakt. Eenvoudig wil in dit verband zeggen: geen 'aggregate functions', geen joining over meerdere relaties.
- Ook is het in de nieuwe release mogelijk om subset requirements in het model te definiëren en te bewaken.

7 REFERENTIES

- [Date 77], Date, C. J., *An introduction to database systems*; second edition; Addison-Wesley, 1977
- [Date 84], Date, C. J., De werking van Relationele systemen. In: *Computerworld*, 2 maart 1984.
- [Rei 85], Reitsma, E. H., Grenzen en beperkingen van SQL. In: *Informatie*, jrg. 27, nr. 3, blz. 198-201, maart 1985.
- [Rem 82], Remmen, F., *Database, grondslagen voor de logische structuur*. Den Haag, Academic Service, 1982.
- [Rem 85,1], Remmen, F., Relationeel, dan is het goed. In: *Informatie*, jrg. 27, nr. 4, blz. 278-280, april 1985.
- [Rem 85,2], Remmen, F., Hoe vriendelijk zijn gebruikersvriendelijke talen. In: *Informatie*, jrg. 27, nr. 7, juli/augustus 1985.
- [Schm-Br 83], Schmidt, J. W., Brodie, M. L.; *Relationeel Database systems*, Springer, 1983.