

35 jaar na de geboorte van het relationele model

# Een actueel in memoriam

Frido van Orden

**In de tijd dat Codd aan zijn baanbrekende werk op het gebied van gegevensbeheer begon, eind jaren zestig, was gegevensverwerking min of meer synoniem met platte bestanden of hiërarchische database management-systemen, in het bijzonder IBM's product IMS. Netwerkdatabases (met als bekendste representant IDMS) waren in die tijd nog 'cutting edge' technologie.**

Beide technologieën hadden gemeen dat het schrijven van database-applicaties kennis vereiste van de wijze waarop het gegevensmodel technisch was geïmplementeerd. Typische operaties daarbij zijn 'doorloop een bestand sequentieel', 'zoek via een index' en 'doorloop een link (pointer-ketting)'. De programmeer-interface was daarmee feitelijk een afgeleide van de technische implementatie van het product.

## Relational rules, deel I

*It was thirty years ago today*

*Dr. Edgar showed the world the way*

*His relations won't go out of style*

*They're guaranteed to last a while*

*So may I introduce to you*

*The act you've known for all these years*

*Dr. Edgar's Data Model Band!*

Chris Date - vrij naar Lennon & McCartney.

In april van dit jaar overleed dr. E.F. Codd op 79-jarige leeftijd. Codd werd wereldberoemd met zijn serie publicaties over een gegevens(meta)model dat later bekend zou worden onder de naam 'Relationeel Model'. De eerste uit die serie publicaties was een intern IBM Research Report dat uitkwam in 1969. 2004 zal dus gelden als een lustrum – 35 jaar Relationeel Model. Frido van Orden zal in Database Magazine een serie artikelen schrijven over de betekenis en de erfenis van het gedachtegoed van Codd. Hij begint met diens regels.

## Radicaal anders

Codd's benadering was radicaal anders. Hij begon met het definiëren van een abstract, 'machinevrij' model (misschien nog beter: theorie) waarin hij een 'relationele kijk' op gegevens beschreef. Dit model baseerde hij op de uit de wiskunde reeds bekende verzamelingenleer en predicatenlogica. Op basis van dit model ontwikkelde hij vervolgens twee verzamelingen operaties (de relationele algebra en de relationele calculus) waarmee hij raadpleeg- en mutatieacties op de gegevensverzameling definieerde.

## Hem komt de eer toe om database management van technologie tot wetenschap te hebben verheven

Inmiddels is bijna vergeten dat Codd naast het relationele model *an sich* nog twee andere concepten introduceerde:

- Het idee 'gegevensmodel' in algemene zin (dus eigenlijk 'gegevensmetamodel'), namelijk een formele beschrijving van de bouwstenen waarmee een gegevensmodel in enge zin (de beschrijving van gegevensstructuren in een bepaald domein) kan worden gespecificeerd. Met opzet is hierboven de term 'programmeer-interface' gebruikt voor de operaties op hiërarchische- en netwerk-databases: het 'hiërarchische model' en 'netwerkmmodel' zijn feitelijk uit de implementatie *reversed engineered* gegevensmodellen;
- Introductie van het concept van functionele afhankelijkheden en normalisatie. Wederom betreft het hier een formele beschrijving van een concept dat ook in de pre-relationele wereld gemeengoed was. De voordelen van normalisatie staan immers los van het relationele model en de principes die aan normalisatie ten grondslag liggen (met name het elimineren van redundantie van gegevens) werden reeds langer toegepast door praktijkdeskundigen.

Codd wilde met zijn relationele model (de term hanteerde hij overigens zelf pas in 1979) een aantal zaken bewerkstelligen:

- Benadering van gegevensverwerking op een theoretisch verant-

# Codd's regels

## 0.

Een systeem dat zich wil kwalificeren als een RELATIONEEL, DATABASE, MANAGEMENT systeem moet zijn RELATIONELE faciliteiten (en niets anders) gebruiken om de DATABASE te MANAGEN.

## 1. De informatie-regel.

De informatie-regel vereist simpelweg dat alle informatie in de database wordt gerepresenteerd op één, en slechts één, manier, namelijk door middel van waarden op kolomposities binnen rijen van tabellen

## 2. De gegarandeerde toegang-regel.

Deze regel is feitelijk een herformulering van de fundamentele eis van primaire sleutels. De regel zegt dat iedere individuele scalaire waarde in de database logisch adresseerbaar moet zijn door specificatie van de naam van de tabel, de naam van de kolom en de primaire sleutel waarde(n) van de regel die de waarde bevat.

## 3. Systematische behandeling van null-waarden.

Het DBMS dient een representatie te ondersteunen voor 'ontbrekende informatie' en 'niet van toepassing zijnde informatie', en wel systematisch, verschillend van alle reguliere waarden (bijvoorbeeld, 'verschillend van nul of ieder ander getal', in geval van numerieke waarden) en onafhankelijk van gegevenstype. Eveneens is vereist dat dergelijke representaties door het DBMS op een systematische wijze worden gemanipuleerd.

## 4. Actieve online catalogus gebaseerd op het relationele model.

Het systeem dient een online, inline, relationele catalogus te ondersteunen, die toegankelijk is voor geautoriseerde gebruikers door middel van hun reguliere vraagtaal.

## 5. De veelomvattende data subtaal-regel.

Het systeem dient minstens één relationele taal te ondersteunen die

- a. een lineaire syntaxis heeft;
- b. zowel interactief gebruikt kan worden als binnen applicatie-programma's;

c. ondersteuning biedt voor datadefinitie-operatoren (inclusief view-definities), datamanipulatie-operaties (mutaties en raadplegingen), beveiliging- en integriteitregels en transactiebeheeroperaties (begin, commit en rollback).

## 6. De view-mutatieregel.

Alle views die theoretisch muteerbaar zijn, dienen muteerbaar te zijn door het systeem.

## 7. Set-level insert, update, en delete.

Het systeem dient set-level INSERT-, UPDATE-, en DELETE-operatoren te ondersteunen.

## 8. Fysieke gegevensafhankelijkheid.

De logische laag van de architectuur wordt afgebeeld op de fysieke laag van de architectuur. Gebruikers en programma's zijn niet afhankelijk van de fysieke structuur van de database.

## 9. Logische gegevensafhankelijkheid.

Gebruikers en programma's zijn zoveel mogelijk onafhankelijk van de logische structuur van de database, dat wil zeggen, de logische structuur van gegevens kan wijzigen met minimale invloed op de applicatie.

## 10. Integriteit-onafhankelijkheid.

Integriteitregels dienen te worden gespecificeerd separaat van applicatieprogramma's en te worden opgeslagen in de catalogus. Het moet mogelijk zijn dergelijke regels te veranderen wanneer dat wenselijk is, zonder bestaande applicaties onnodig te beïnvloeden.

## 11. Distributie-onafhankelijkheid.

Bestaande applicaties dienen te blijven werken

- a. indien een gedistribueerde versie van het DBMS wordt geïntroduceerd;
- b. indien bestaande gedistribueerde gegevens worden herdistribueerd over het systeem.

## 12. De non-ondermijning regel.

Indien het systeem een laag niveau (per-record) interface ondersteunt, dan kan die interface niet worden gebruikt om het systeem te ondermijnen, bijvoorbeeld door een relationele beveiliging- of integriteitregel te passeren.

woorde, a-technologische manier. Feitelijk komt hem dan ook de eer toe om database management van technologie tot wetenschap te hebben verheven;

- Volledige scheiding van model en implementatie. Natuurlijk waren de hiërarchische en netwerk-DBMS'en vergeleken met het in binaire codes rechtstreeks aanspreken van de hardware van de Eniac (de allereerste computer uit 1946) reeds een enorme vooruitgang. Codd wilde echter ook de laatste stap zetten,

en wel naar een echt 'implementatie-vrij' model. Met de vertaling van relationeel model naar implementatie in een RDBMS heeft hij zich pas bezig gehouden toen hij tot de conclusie kwam dat de RDBMS-leveranciers daar niet veel van bakten;

- Ontsluiting van gegevens ten behoeve van eindgebruikers, zonder tussenkomst van programmeurs, door middel van het beschikbaar stellen van een voor iedereen (...) begrijpelijk model met bijbehorende vraagtaal.

### Codd's erfenis

In 1969 ondervond Codd binnen IBM weinig enthousiasme voor zijn ideeën. Niet verwonderlijk, aangezien IBM's succes voor een niet onaanzienlijk deel gebaseerd was op IMS! Pas in 1981 kwam IBM met zijn eerste relationele product, SQL/DS, nadat onder meer de garage company van ene Larry Ellison de eerste relationele producten op de markt had gebracht.

Hoe anders ziet de wereld er anno 2003 uit. Hoewel IMS en IDMS, de mastodonten van weleer, nog een stevige poot aan de grond hebben in de wereld van stokoude systemen waar niemand meer aan durft te komen, is het relationeel wat de klok slaat voor vrijwel iedere nieuwe vorm van gegevensverwerking: van desktop tot *enterprise-level*. De voorspelde aanval van object-georiënteerde databases is uitgebleven en het antwoord van de gevestigde relationelen daarop – de 'object-relationele' features van onder meer Oracle en DB2 – worden in de praktijk nauwelijks gebruikt. Hebben de relationelen daarmee de eindoverwinning behaald? Dat hangt er maar vanaf hoe men de term 'relationeel' precies definieert. Codd voorzag reeds in 1985 – dus luttele jaren nadat de eerste relationele probeersels op de markt kwamen – dat de term 'relationeel' het risico liep om uitgehold te worden, doordat allerhande lieden hun producten als 'relationeel' zouden gaan promoten.

### De garage company van ene Larry Ellison had de eerste relationele producten op de markt gebracht

Hij ontwikkelde daarom een lijst van 12 – eigenlijk 13 – regels waaraan een DBMS naar zijn mening zou moeten voldoen, alvorens het zich 'relationeel' mag noemen, zie het bijgaand kader. Het zal u wellicht niet verbazen dat anno 1985 geen van de toenmalige producten aan alle regels voldeed. En het zal Codd zaliger allicht treurig stemmen dat dit, alle vooruitgang ten spijt, bijna 20 jaar later nog steeds het geval is. Codd's offensief tegen omgebouwde pre-relationele databases (door hem minachtend "born again relational" genoemd) heeft echter wel bijgedragen aan het om zeep brengen van relationele wanproducten als IDMS/R(ational) van Cullinet: een netwerk-RDBMS met een relationeel laagje er overheen, vergezeld van folders die riepen dat Cullinet ook relationeel was geworden en white papers die beschreven hoe men Codd's relationele model had aangepast en verbeterd.

Nog triester echter, zeker voor een man die zijn leven lang wetenschapper bleef en anderen rijk zag worden met zijn ideeën, is het dat 'het relationele model' steeds meer synoniem wordt met 'de huidige relationele producten' of 'SQL'. Dat mogen zich enerzijds



Ted Codd: database management van technologie tot wetenschap verheven.

diegenen aantrekken die deze synonymisering op hun geweten hebben (met name in de object-georiënteerde wereld zijn zij te vinden), maar anderzijds zeker ook de leveranciers van op relationele concepten gebaseerde producten, die er de afgelopen 20 (of zelfs 35) jaar niet in geslaagd zijn om hun producten écht relationeel te maken.

Dat wil overigens niet zeggen dat het daarmee slechte producten zijn, integendeel, zie ook de eerder gemaakte vergelijking tussen IMS en de Eniac.

### Living by the rules

Het relationele model: velen menen het te kennen, slechts weinigen kennen en begrijpen het ook werkelijk. Alle aanleiding dus om in het lustrumjaar 2004 een artikelserie in DB/M te wijden aan de kwaliteiten van het relationele model, de (non-)ondersteuning daarvan in de bekende relationele producten en de claims van post-relationele technologieën. Als leidraad daarbij zullen we Codd's lijst van 13 regels hanteren. Niet als 'rood boekje', maar wel als een geweldige bron van inspiratie tot denken over gegevensbeheer. Ted Codd is dood, zijn geest leeft voort.

### Literatuur

*C.J. Date – The Database Relational Model, A Retrospective Review and Analysis, Addison Wesley, 2001. Een beknopte samenvatting van Codd's verschillende publicaties over het relationele model, waarvan de meeste nog maar lastig te vinden zijn. Zeker voor de prijs van \$10,- een must voor iedere serieuze automatiseerder.*

*C.J. Date & H. Darwen – Foundation for Future Database Systems: The Third Manifesto – 2nd edition, Addison Wesley, 2000. Chris Date als Codd's natuurlijke opvolger in optima forma: een grondige verkenning van concepten uit de object oriëntatie, zowel op zichzelf als in relatie tot het relationele model.*

**Frido van Orden** (frido.van.orden@faapartners.com) is partner bij FAA Partners.