Regel 3: Systematische behandeling van NULL-waarden

Omgaan met ontbrekende informatie heet hangijzer

Frido van Orden

De aanhangers van het relationele model (en dat zijn er onder de DB/M-lezers ongetwijfeld velen) hebben tot nu toe in deze serie vergenoegd achterover kunnen hangen. Bij de behandeling van de eerste regels zijn immers vooral de pre-relationele systemen (hiërarchisch en netwerkmodel) en post-relationele systemen (objectgeoriënteerd model) onder vuur genomen. Het begon bijna saai te worden, want alle munitie is door Codd vrijwel panklaar aangeleverd. Deze keer gaat het echter over de behandeling van NULL-waarden en laat dat nu net het voornaamste twistpunt zijn tussen de twee relationele boegbeelden bij uitstek, Codd en Date. Tijd dus om partij te kiezen in deze relationele stammenstrijd!

De derde regel gaat over 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.

Logica

De behandeling van NULL-waarden is meer algemeen bekend als de problematiek van ontbrekende informatie (missing information). Deze problematiek is te herleiden tot de zogenaamde 'closed world assumption', die ten grondslag ligt aan onder meer predikatenlogica en databases. De closed world assumption stelt simpelweg dat alle informatie (feiten) die niet is vastgelegd en ook niet kan worden afgeleid, moet wordt beschouwd als zijnde onwaar.
Nemen we een personen-database als voorbeeld met daarin opgeslagen gegevens over personen met de naam Jansen, Pietersen en Klaassen. Indien aan deze database de vraag wordt gesteld 'Bestaat er een persoon met de naam De Roy van Zuidewijn?', dan is het antwoord op deze vraag ontkennend. De reden daarvoor is niet dat er ergens is vastgelegd dat De Roy van Zuidewijn niet bestaat, maar omdat er nergens is vastgelegd dat de betreffende persoon wel bestaat.

De closed world assumption is een onmisbaar fundament in de logica. Stelt u zich eens voor welke informatie over niet-bestaande personen er zou moeten worden vastgelegd als we de closed world assumption niet hadden. Zelfs in ons uitermate simpele voorbeeld mini-database, bestaande uit 1 tabel met 1 kolom, zouden we alle mogelijke namen moeten vastleggen met daarbij een indicatie of de naam wel of niet voorkomt. Dit is een volstrekt onwerkbare situatie.
Strikt redenerend vanuit de closed world assumption lijkt er geen enkel probleem rondom ontbrekende informatie. Informatie die er niet is wordt simpelweg niet vastgelegd. Vanwaar dan toch alle problemen?

Logica versus Structuur

Laten we het voorbeeld eens iets realistischer maken. We breiden de persoonsregistratie uit met gegevens over geslacht, geboortedatum en adres van de persoon, alsmede het aantal zwangerschappen dat de persoon gehad heeft. Zie de tabel in afbeelding 1.

Wat ogenblikkelijk opvalt is dat niet voor elk persoon elk attribuut een waarde heeft. Bij Pietersen is geen geboortedatum geregistreerd, terwijl zowel Jansen als Pietersen geen huisnummertoevoeging kennen. Het aantal zwangerschappen is alleen bij Pietersen ingevuld. Wat is hier aan de hand?
Van nature heeft ieder persoon een geboortedatum. Wellicht is die niet bij ons bekend, en wellicht zelfs niet bij de persoon zelf, maar in elk geval is de geboortedatum een eigenschap van ieder persoon. De betekenis van de niet ingevulde geboortedatum van Pietersen is dus vermoedelijk dat de geboortedatum ons niet bekend is.

Relational Rules (5)

De vorig jaar overleden dr. E.F. 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 schrijft in Database Magazine een serie artikelen over de betekenis en de erfenis van het gedachtengoed van Codd. Aflevering vijf gaat over de derde regel.

Naam	Geslacht	Geb.Datum	Straat	Huisnr.	Huisnr.Toev.	Woonplaats	Aant.Zwangersch.
Jansen	M	17-10-1958	Dorpsstraat	2		Ons Dorp	
Pietersen	V		Biltstraat	13		Utrecht	0
Klaassen	V	29-02-1976	Leidsestraat	14	A	Amsterdam	

Afbeelding 1.

Bij de huisnummertoevoeging is de situatie anders. Natuurlijk kan het zijn dat Pietersen feitelijk op de Biltstraat 13-bis woont, maar dat die informatie ons niet bekend is. Meer waarschijnlijk is het echter dat Jansen en Pietersen op een adres wonen zonder huisnummertoevoeging, met andere woorden, de huisnummertoevoeging is niet van toepassing. Het zou ook nog kunnen dat we zeker weten dat Jansen's adres geen toevoeging heeft, maar dat we het bij Pietersen niet weten.

Weer anders is het bij het aantal zwangerschappen. Jansen is een man, het aantal zwangerschappen is dus niet voor hem van toepassing. Van Pietersen weten we blijkbaar dat zij nooit zwanger is geweest, van Klaassen weten we dat niet. De betekenis van het niet geregistreerde aantal zwangerschappen bij Jansen is 'niet van toepassing', en dit weten we zeker omdat we weten dat Jansen een man is. Bij Klaassen weten we zeker dat het aantal zwangerschappen van toepassing is en is de betekenis derhalve dat het aantal zwangerschappen onbekend is.
We hebben dus vier verschillende betekenissen voor het niet geregistreerd zijn van bepaalde informatie:

- De informatie is zeker van toepassing, maar niet bekend (voorbeeld: geboortedatum, aantal zwangerschappen voor vrouwen);
- De informatie is wellicht van toepassing, maar we kunnen dit nergens uit afleiden. De informatie zou ook niet bekend kunnen zijn (voorbeeld: huisnummertoevoeging);
- De informatie is zeker niet van toepassing, maar we kunnen dit nergens uit afleiden. (voorbeeld: huisnummertoevoeging);
- De informatie is zeker niet van toepassing, en we kunnen dit afleiden uit andere informatie (voorbeeld: aantal zwangerschappen voor mannen).

De betekenis van de gegevens in onze database zou dus kunnen zijn zoals in de tabel in afbeelding 2.

De visie van Codd

De visie van Codd wordt goed samengevat in Regel 3 waarmee dit artikel opent. Codd maakt expliciet onderscheid tussen 'onbekend/ontbrekend' en 'niet van toepassing'. Ook stelt hij dat geen misbruik mag worden gemaakt van reguliere waarden. Dus geen geboortedatum '01-01-0001' of '31-12-9999' om aan te geven dat de geboortedatum onbekend is!
Ook moet de representatie onafhankelijk van het gegevenstype zijn, met andere woorden de representatie voor 'datum onbekend' en 'huisnummer toevoeging onbekend' zou identiek moeten zijn.

De behandeling van NULL-waarden is bekend als de problematiek van ontbrekende informatie

Merk op dat de optie '99-99-9999' voor 'geboortedatum onbekend' hierdoor afvalt. Dit is weliswaar geen reguliere datumwaarde, maar mogelijk wel een reguliere waarde voor andere gegevenstypen.
Tot slot stelt Codd dat de representaties door het DBMS systematisch moeten worden gemanipuleerd. Hiermee wordt bedoeld dat de representaties kunnen worden gemanipuleerd en uitgevraagd, net zoals dit geldt voor de reguliere waarden in de database.

De visie van Date

Chris Date is het op een aantal fundamentele punten oneens met Codd's visie op het omgaan met ontbrekende informatie. Zijn belangrijkste drie bezwaren zijn de volgende:

- De twee soorten ontbrekende informatie 'onbekend/ontbreekt' en 'niet van toepassing' die Codd onderscheidt zijn volstrekt willekeurig. Er zijn meer varianten denkbaar, zoals bijvoorbeeld 'voor ons niet interessant', 'wel bekend maar te twijfelachtig van kwaliteit om te registreren';
- Representaties voor 'waarde onbekend', 'waarde niet van toepassing' etcetera, zijn geen echte waarden ('de waarde voor geboortedatum is 17 oktober 1958') maar zijn eerder te beschouwen als meta-informatie (de waarde is onbekend/niet van toepassing);

Naam	Geslacht	Geb.Datum	Straat	Huisnr.	Huisnr.Toev.	Woonplaats	Aant.Zwangersch.
Jansen	M	17-10-1958	Dorpsstraat	2	*N.v.t.*	Ons Dorp	*N.v.t.*
Pietersen	V	*Onbekend*	Biltstraat	13	*Onbekend/n.v.t.*	Utrecht	0
Klaassen	V	29-02-1976	Leidsestraat	14	A	Amsterdam	*Onbekend*

Afbeelding 2.

Naam	Geslacht	Geb.Datum	Straat	Huisnr.	Huisnr.Toev.	Woonplaats
Jansen	M	17-10-1958	Dorpsstraat	2	*N.v.t.*	Ons Dorp
Pietersen	V	*Onbekend*	Biltstraat	13	*Onbekend/n.v.t.*	Utrecht
Klaassen	V	29-02-1976	Leidsestraat	14	A	Amsterdam

Naam	Aant.Zwangersch.
Pietersen	0

Afbeelding 3.

- De aparte representaties voor ontbrekende informatie zijn anders dan reguliere waarden (op zich prima) maar gedragen zich in operaties ook anders. Het antwoord op de vraag 'welke mensen hebben een geboortedatum groter of gelijk aan de datum van vandaag' bevat bijvoorbeeld niet de mensen waarvan de geboortedatum onbekend is, ook niet indien het systeem weet (door middel van een constraint) dat de geboortedatum altijd op of voor de huidige datum moet liggen.

In Date's perspectief zijn er betere oplossingen om om te gaan met ontbrekende informatie. Een daarvan is het simpelweg niet registreren van de informatie, de closed world assumption in gedachte. Zo kan bijvoorbeeld de informatie over het aantal zwangerschappen in ons voorbeeld in een aparte relatie worden vastgelegd, hetgeen leidt tot de database te zien in afbeelding 3. Date verwijst bij deze constructie expliciet naar de fysieke gegevensonafhankelijkheid die Codd als regel 8 formuleert. In dit verband wordt ermee bedoeld dat het heel goed mogelijk is om de informatie in de database fysiek vast te leggen als in het originele voorbeeld met 1 tabel, maar dat het logische gegevensmodel uit 2 relaties bestaat.
Een bezwaar aan deze constructie is dat de structuur van het gegevensmodel nogal verandert, enkel en alleen om netjes met ontbrekende informatie om te kunnen gaan. Overigens kan het model met 2 relaties natuurlijk ook prima functioneren zonder ontbrekende informatie, de beide relaties zijn dan 1:1 in plaats van 0-1:1.
Een andere oplossing die Date aandraagt is het definiëren van speciale waarden om ontbrekende informatie mee vast te leggen. Dit lijkt op Codd's definities van representaties voor ontbrekende informatie, maar het wezenlijke verschil is dat Date's speciale waarden echt als geldige waarde voor het onderliggende gegevenstype worden beschouwd. De speciale waarden zijn daarmee ook automatisch getypeerd (een ontbrekende datum is iets anders dan een ontbrekende huisnummertoevoeging), in afwijking van Codd's regel dat representaties onafhankelijk van gegevenstype moeten zijn.
Het voordeel van Date's benadering zit erin dat er vanuit gebruikersperspectief geen speciale behandeling is voor 'ontbrekende informatie' (vergelijk SQL "datum = '17-10-1958' " versus 'datum IS NULL'). Het probleem dat er echter niet mee wordt opgelost is het antwoord op de eerder geformuleerde vraag 'welke mensen hebben een geboortedatum groter of gelijk aan de datum van vandaag'. In het gegevenstype 'datum' zal moeten worden gedefinieerd wat de betekenis is van de vergelijkingsoperatoren groter dan, kleiner dan en is gelijk. Er is geen enkele definitie mogelijk waarin de expressie

```
geboortedatum < 'een datum' XOR geboortedatum >=
                                              'een datum'
```

altijd 'waar' als resultaat oplevert, ook indien de geboortedatum een speciale waarde voor ontbrekende informatie bevat. Date onderkent dit ook en geeft daarom sterk de voorkeur aan het helemaal niet vastleggen van de ontbrekende informatie, zoals eerder besproken.

> Informatie die er niet is wordt simpelweg niet vastgelegd

Een laatste alternatief van Date is het beschouwen van de ontbrekende informatie als meta-informatie. Inzake de huisnummertoevoeging zouden we bijvoorbeeld een attribuut 'Status huisnummertoevoeging' kunnen opnemen met als waarden 'Bekend', 'Onbekend' en 'Niet van toepassing'.

De praktijk

Los van alle theoretische bespiegelingen zullen we het in de alledaagse relationele praktijk gewoon met SQL moeten doen. SQL biedt ondersteuning voor ontbrekende informatie door middel van de NULL 'waarde' (waarde tussen aanhalingstekens omdat NULL zich geheel afwijkend van alle andere SQL-waarden gedraagt). Vergelijken we SQL NULL met Codd's regel dan zien we het volgende:

- Onderscheid tussen 'ontbrekend' en 'niet van toepassing': nee;
- Verschillend van alle reguliere waarden: ja;
- Onafhankelijk van gegevenstype: ja;
- Systematische manipulatie: ja (Date zou zeggen: ja, maar op een verschrikkelijke manier).

De systematische manipulatie van SQL bestaat er met name uit dat operaties waarbij minimaal een van beide operanden NULL is, altijd NULL als resultaat opleveren. Dus ook:

- NULL = anything ➔ false, in het bijzonder:
  NULL = NULL ➔ false;
- NOT(NULL) ➔ NULL;
- NULL OR anything ➔ NULL;
- NULL || 'any string' ➔ NULL.

Omdat vergelijkingen met NULL altijd NULL opleveren (zelfs indien de te vergelijken waarde zelf NULL is) definieert SQL speciale operatoren IS NULL en IS NOT NULL om te testen of een bepaalde waarde NULL is.

Om zaken extra moeilijk te maken biedt een aantal RDBMS'en, waaronder Oracle, als feature aan dat karakterwaarden van 0 tekens lang (lege strings) als NULL worden geïnterpreteerd. De gedachte hierachter is ongetwijfeld dat bij de representatie van gegevens in rapporten en op schermen, het onderscheid tussen een NULL en een lege string visueel niet te maken is. Daar valt op zich iets voor te zeggen, ware het niet dat we in de database nu met onvoorspelbare effecten worden geconfronteerd. Immers, zoeken op 'naam = ?', waarbij het vraagteken staat voor een nader in te vullen parameterwaarde, werkt prima zo lang de parameter maar uit minimaal 1 teken bestaat. Is de parameterwaarde een lege string, dan wordt nooit meer iets gevonden (naam = ' ' is identiek aan naam = NULL en dat levert altijd false op) en moet ineens 'naam IS NULL' worden gebruikt.
Wie deze complicaties niet als normaal ervaart (lees: wie niet is afgestompt door jarenlange besmetting met SQL) zal de suggestie van Date, om waar mogelijk ontbrekende informatie helemaal niet te registreren, allicht charmant vinden. We raken daarbij het al eerder genoemde aspect van fysieke gegevensonafhankelijkheid, want we willen natuurlijk geen performance inleveren als we de tabel splitsen in meerdere tabellen. We zullen hierop verder ingaan bij de bespreking van regel 8, maar noemen nu vast het Cluster feature zoals onder meer Oracle dat kent. Dit feature maakt het mogelijk om records uit verschillende tabellen fysiek bij elkaar op te slaan.

Conclusie

Het omgaan met ontbrekende informatie blijft een heet hangijzer. Een eenvoudige gegevensstructuur, voldoende uitdrukkingskracht van het gegevensmodel en een voorspelbare en voor iedereen begrijpelijke betekenis van gegevens en query's, lijken nauwelijks met elkaar te verenigen. Aan u, automatiserings-professional, om een optimale toepassing van de verschillende mogelijkheden uit te dokteren. Sterkte daarbij!

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