Relaties van hogere graad en hogere orde

Orthogonalen; een eerste introductie

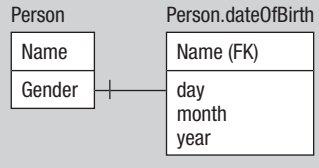
Maurice Gittens

In dit artikel wordt het concept orthogonaal geïntroduceerd als een generalisatie van de wel bekende identificerende relatie. Door middel van voorbeelden zal toegankelijk gemaakt worden dat orthogonalen een plaats verdienen in het palet van concepten die ter beschikking staan van de datamodelleur.

Concepten als complexe domeinen en hogere orde-relaties worden in dit artikel als verschijningsvormen van het meer algemene concept van een orthogonaal gepresenteerd. Tevens wordt door middel van voorbeelden geïllustreerd dat het *classes as domains* dogma, zoals deze door Chris Date en Darwen in 'The Third Manifesto' is gepropageerd, niet meer is dan een onnodige schending van de eerste normaalvorm. Ter afsluiting worden aan de hand van orthogonalen de door Codd geïntroduceerde *RN-Domains* en *E-Domains* onder de aandacht gebracht. Deze komen van pas bij de ondersteuning van hogere orde-relaties.

Identificerende relaties

Beschouw eens het eenvoudige datamodel waarin twee entiteiten door middel van een identificerende relatie aan elkaar gerelateerd worden, zie afbeelding 1. Een *Person* wordt via een identificerende één-op-één relatie gekoppeld aan een aan de betreffende persoon gerelateerde geboorte datum. De relatie heet identificerend, omdat de identificerende kenmerken van een Person volgens dit model een deelverzameling vormen van de identificerende kenmerken van de entiteit die in dit voorbeeld *Person.dateOfBirth* is genoemd. Het ligt voor de hand dat een additionele, optionele en identificerende relatie *Person.dateOfDecease* zou kunnen worden ingezet om de overlijdingsdatum te modelleren. In de volgende sectie wordt het bovenstaande gegeneraliseerd en in een datataal gepresenteerd. (Delen van deze taal zijn reeds geïmplementeerd.)



Afbeelding 1: Datamodel waarin twee entiteiten door middel van een identificerende relatie aan elkaar gerelateerd worden.

Orthogonalen als complexe domeinen

Het eerste soort orthogonaal dat de revue passeert staat bekend als een *complex domein*. (Volgens 'The Third Manifesto' komen deze overeen met objectgeoriënteerde classes.) Het volgende fragment toont de declaratie van een orthogonaal *Date* die als complex domein wordt ingezet.

```
declare orthogonal Date
    { day integer; month integer; year integer; }
create table Person
    { name char[20]; gender char[1];
      birth Date; decease Date optional;
      primary key(name)}
```

Dit gebeurt door bij de declaratie van de Person tabel een attribuut *birth* op te nemen die van het type Date is. Ter illustratie wordt ook een optioneel attribuut *decease* aan de declaratie van Person toegevoegd. Door middel van de volgende statements wordt de Person tabel gevuld.

```
insert Person
    { name("Joe"), gender("M"),
      birth { day(11), month(11), year(1911) },
      decease { day(12), month(12), year(1992) }};
insert Person
    { name("Jane"), gender("F"),
      birth { day(11), month(11), year(1911)}};
```

De resulterende relationele structuur kan in de onderstaande tabel worden weergegeven. De onderstaande tabel is dan ook het resultaat van het statement:

```
select * from Person;
```

NAME	gender	birth.day	birth.month	birth.year	decease.day	decease.month	decease.year
Joe	M	11	11	1911	12	12	1992
Jane	F	11	11	1911			

Het is in deze wel belangrijk om te appreciëren dat *Person.birth* en *Person.decease* beide normale tabellen zijn. Dus alle relationele operatoren kunnen op deze relaties worden uitgevoerd. Ook relationele constraints worden op idiomatische wijze ondersteund. Als voorbeeld: het statement

```
select * from Person.birth;
```

levert als resultaat de volgende verzameling op.

PERSON.NAME	day	month	year
Joe	11	11	1911
Jane	11	11	1911

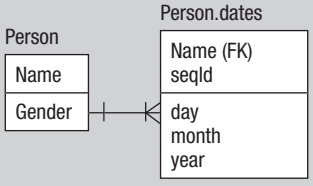
De relaties Person.birth en Person.decease kunnen dus als *weak entities* van Person gezien worden. De omgang met optionele attributen blijkt uit het volgende statement

```
select * from Person.decease;
```

dat in dit voorbeeld de volgende verzameling als resultaat oplevert.

PERSON.NAME	day	month	year
Joe	12	12	1992

Zoals in de bovenstaande tabel is te zien geldt dat slechts de persoon 'Joe' een overlijdingsdatum heeft. Het optionele attribuut decease is dus te zien als een optionele identificerende relatie tussen Person en Person.decease. Juist omdat Person.birth en Person.decease niet alleen als complexe domeinen maar ook als normale relaties te zien zijn, is er grond om te twijfelen aan de validiteit van de propositie van Chris Date en Darwen die erop neerkomt dat complexe domeinen en relaties in wezenlijke zin andere concepten zijn. (Zie bijvoorbeeld pagina 18 van de tweede editie van 'The Third Manifesto'.)



Afbeelding 2: Identificerende één-op-veel relaties.

Het is natuurlijk ook mogelijk om orthogonalen te definiëren die gebaseerd zijn op al dan niet optionele identificerende één-op-veel relaties, zie afbeelding 2. Eén-op-veel orthogonalen kunnen syntactisch worden onderscheiden door de toevoeging van additionele identificerende attributen, zoals in het volgende voorbeeld wordt geïllustreerd.

```
declare orthogonal Dates
    { seqId integer, day integer;
      month integer; year integer;
      primary key(sequenceId); }
create table Person
    { name char[20]; gender char[1];
      specialdays Date;
      primary key(name); }
```

In het bovenstaande voorbeeld zal de tabel *Person.specialdays* ook door het attribuut *sequenceId* worden geïdentificeerd. Met de volgende vulling

```
insert Person
    { name("Joe"), gender("M"),
      specialdays { seqId(0), day(11), month(11),
      year(1911)}};
insert Person
    { name("Joe"), gender("M"),
      specialdays { seqId(1), day(12), month(12),
      year(1992)}};
insert Person.days
    { name("Jane"),  gender("F"),
      specialdays { seqId(0), day(11), month(11),
      year(1911)}};
```

geeft het statement

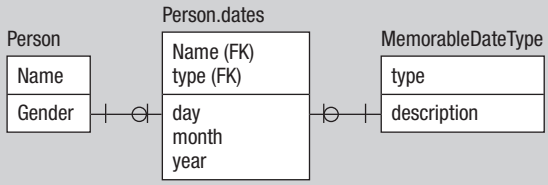
```
select * from Person;
```

als resultaat de volgende verzameling (specialdays wordt tot sdays afgekort).

NAME	gender	SDAY.seqid	sday.day	sday.month	sday.year
Joe	M	0	11	11	1911
Joe	M	1	12	12	1992
Jane	F	0	11	11	1911

Orthogonalen en relaties van hogere graad

De graad van een orthogonaal wordt bepaald door het aantal identificerende relaties dat als basis geldt voor de betreffende orthogonaal. In het volgende voorbeeld, zie afbeelding 3, wordt via twee identificerende relaties een aantal soorten datums, zoals geboortedatum, huwelijksdatum etcetera, aan een Person gekoppeld.



Afbeelding 3: Via twee identificerende relaties wordt een aantal soorten datums aan een Person gekoppeld.

In onderstaand fragment wordt via het gereserveerde woord *attach* de orthogonaal Date aan Person en *DateType* gekoppeld.

```
declare orthogonal Date { day integer;
                    month integer; year integer; };
create table Person { name char(20);
             gender char(1); primary key (name); };
create table DateType {type char(1);
               desc char(256); primary key (type)};
attach Date to Person.dates optional,
                        DateType.instances optional
```

Ter illustratie wordt een aantal tuples aan de relaties uit het bovenstaande voorbeeld toegevoegd.

```
insert Person { name("Joe"), gender("M"),
       dates{ day(23), month(7), year(1950)}},
       DateType{ type("B"), desc("Date of Birth") };
insert DateType{ type("M"), desc("Date of Marriage")},
       Person{ name("Joe"), gender("M"),
       dates { day(12), month(8), year(1992)}};
insert Person { name("Jane"), gender("F"),
       dates{ type("B"), day(8), month(8),
       year(1970)}};
```

De volgende query

```
select * from Person
```

geeft in dit geval het volgende resultaat.

NAME	dates.type	GENDER	dates.day	dates.month	dates.year
Joe	B	M	23	7	1950
Joe	M	M	12	8	1992
Jane	B	F	8	8	1970

Hierbij wordt benadrukt dat *Person.dates* en *DateType.instances* dezelfde relatie identificeren. Dit houdt in dat de statements

```
select * from Person.date;
select * from DateType.instances;
```

beide het volgende resultaat geven.

PERSON.NAME	DATEType.type	day	month	year
Joe	B	23	7	1987
Joe	M	12	8	1992
Jane	B	8	8	1970

Orthogonalen en relaties van hogere orde

In de publicatie 'Extending the Database Relational Model to Capture More Meaning' heeft Codd een aantal concepten geïntroduceerd die in zijn visie de semantische uitdrukkingskracht van het relationele model zouden uitbreiden.
In dit artikel heeft Codd onder andere zogenaamde *RN-Domains* en *E-Domains* aangedragen als bouwstenen die bijdragen tot het doel van het uitbreiden van het relationele model. In deze sectie wordt op eenvoudige wijze geïllustreerd hoe orthogonalen bijdragen tot het doel dat door Codd in het genoemde artikel is nagestreefd. Anders dan Chris Date en Darwen (zie pagina 6 van The Third Manifesto) zocht en vond Codd *wel* mogelijkheden om het relationele model qua semantische uitdrukkingskracht uit te breiden.

Beschouw als u wilt het volgende eenvoudige voorbeeld waarin de *Date* orthogonaal in drie verschillende tabellen wordt gebruikt.

```
declare orthogonal Date { day integer;
                    month integer; year integer; };
create table Person { name char(20);
            dateOfBirth Date; primary key(name); };
create table Organisation { name char(20);
           dateOfIncorp Date; primary key(name); };
create table Vehicle { id char (20);
                dateOfMan Date; primary key(id); };
```

Deze tabellen worden met de volgende statements van voorbeeldvulling voorzien.

```
insert Person { name("Joe"), dateOfBirth
                { day(23), month(7), year(1997)} };
insert Person { name("Jane"), dateOfBirth
                { day(2), month(12), year(1987)} };
insert Organisation { name("CompanyX"),
   dateOfIncorp { day(12), month(3), year(1980) }};
insert Vehicle { id("23-RD"), dateOfMan
                 { day(3), month(4), year(1970)} };
```

Dat orthogonalen overeenkomen met hogere orde-predikaten uit de predikatenlogica, is te appreciëren wanneer we het resultaat van het volgende statement beschouwen.

```
select * from Date;
```

TABLE ID	TUPLE ID	day	month	year
Person.dateOfBirth	oid{Joe}	23	7	1997
Person.dateOfBirth	oid{Jane}	2	12	1987
Organization.dateOfIncorp	oid{My Corp}	12	3	1980
Vehicle.dateOfMan	oid{23-RD}	3	4	1970

Dit is een relatie geïdentificeerd door de kolommen *Table Id* (een Table Id komt overeen met een predikaatconstante uit de hogere orde-logica) en *Tuple Id* (een tuple Id is onder verschillende namen bekend, bijvoorbeeld object identifier, surrogate, row identifier etcetera), die een rij bevat voor iedere instantie van een *Date* in onze voorbeeld database. Het domein van de kolommen table Id en tuple Id komt overeen met respectievelijk RN-Domain en E-Domain zoals deze door Codd zijn geïntroduceerd.
Uit het bovenstaande volgt dat een orthogonaal zoals *Date* in ons voorbeeld gezien kan worden als een relatie gedefinieerd als de vereniging van alle *Date* instanties in de relaties *Person.dateOfBirth*, *Organisation.dateOfIncorp* en *Vehicle.dateOfMan*.
Als laatste wordt benadrukt dat orthogonalen, volgens de definitie van Codd normale relaties zijn. Dit wil bijvoorbeeld zeggen dat de *Date* orthogonaal uit ons voorbeeld de rol van *parent* kan vervullen in een foreign key constraint.

Conclusie

In dit artikel is een eenvoudige orthogonaal *Date* aan de hand van voorbeelden geïntroduceerd. Door de inzet van de alom bekende identificerende relatie is aannemelijk gemaakt dat concepten als *complexe domeinen* in toekomstige databases geaccommodeerd kunnen worden zonder de door Chris Date en Darwen voorgestelde schending van de eerste normaalvorm. Op basis van dezelfde *Date* orthogonaal zijn relaties van hogere graad en relaties van hogere orde gepresenteerd. Over andere soorten orthogonalen, bijbehorende data modelleer-idiomen en toepassingen is natuurlijk veel meer te zeggen. Bij tijd en gelegenheid zal dan ook over deze onderwerpen worden uitgewijd.

Literatuur

E.F. Codd (1979), Extending the Database Relational Model to Capture more Meaning.
C.J. Date, Huge Darwen (2000), Foundation for Future Database Systems, Addison-Wesley Publishing Company.
Jan van Eijck en Elias Thijsse (1989), Logica voor alfa's en informatici, Academic Service.
Maurice Gittens, 'The Third Manifesto' kritisch bekeken, DB/M 5 en 6, 2004.

Maurice Gittens (maurice@gittens.nl) is zelfstandig IT-consultant.