

Toepassing in de praktijk bij de Belastingdienst

Het ontwerpen van een adaptief datamodel

Johan van der Graaf

Bij het ontwerpen van een datamodel is het soms noodzakelijk de traditionele paden te verlaten, wil men tot een gedegen resultaat komen. Dit speelt vooral wanneer de uitbreiding op het datamodel een periodiek (jaarlijks) terugkerend verschijnsel is. Vaak is de tijd om uitbreidingen te realiseren te kort. Wijzigingen op het datamodel betekenen vervolgens altijd wijzigingen in de toepassingen die op het datamodel werkzaam zijn.

Veranderingen in de toepassingen kunnen worden voorkomen als gebruik wordt gemaakt van een adaptief datamodel. Door middel van een tweetal voorbeelden wordt duidelijk gemaakt hoe zo'n datamodel tot stand komt.

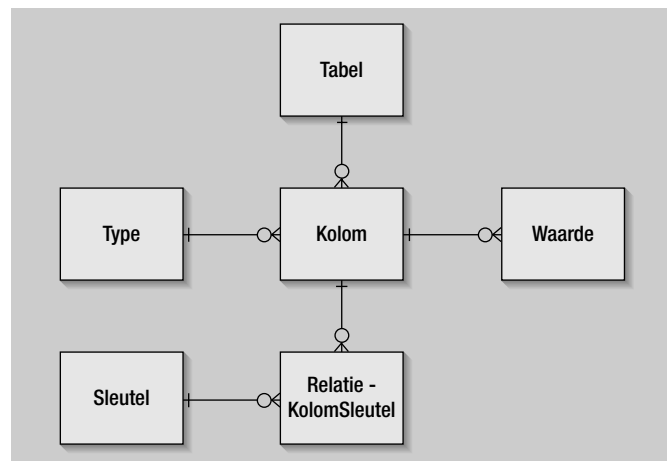
In het eerste voorbeeld wordt uitgegaan van een simpel datamodel voor de registratie van enkele persoonsgegevens. Deze persoonsgegevens bestaan uit het sofinummer, de naam en de geboortedatum van een persoon. In afbeelding 1 wordt een tabel 'Persoon' getoond met enkele persoonsgegevens. De lijn boven de kolom 'sofinummer' is een grafische weergave voor de sleutel van de tabel.

	sofinummer	naam	geboortedatum
	integer	char(64)	date
rij 1	572550078	Korporaal	19361206
rij 2	686600009	Sebus	19430409
rij 3	258091393	Rosmalen	19860708
rij 4	378456556	Pasman	19560208
rij 5	733620942	Bos	19560401

AFBEELDING 1: DE TABEL 'PERSOON'.

Om tot een adaptief datamodel te komen moeten de gegevens van het model en de gegevens van het bijbehorende metamodel in één datamodel worden samengevoegd. Populair gezegd is een metamodel een model van een model. In dit voorbeeld bestaat het metamodel, dat dus zelf ook weer een model is, uit de tabellen 'Tabel', 'Kolom', 'Type' en 'Sleutel'. Omdat de relatie tussen de tabellen 'Kolom' en 'Sleutel' een m-op-n-relatie betreft, is er ook een relatie-

tabel 'RelatieKolomSleutel' noodzakelijk. De data uit het model in afbeelding 1 wordt vastgelegd in de tabel 'Waarde'. Afbeelding 2 geeft het adaptief datamodel in een grafische vorm weer.



AFBEELDING 2: HET ADAPTIEF DATAMODEL.

De tabelgegevens bestaan uit een id (identificatienummer) en de naam. Het toegevoegde id functioneert als kunstmatige sleutel voor het realiseren van de relaties. De naam is de logische sleutel. Afbeeldingen 3a tot en met 3e tonen de vulling van de genoemde tabellen, gebaseerd op de tabel uit afbeelding 1.

id	naam
integer	char(64)
41	Persoon

AFBEELDING 3A: DE TABEL 'TABEL'.

De typegegevens bestaan uit een id, een typering en optioneel een lengte-aanduiding. De naam en lengte vormen tezamen de logische sleutel.

De kolomgegevens bestaan uit een id, een naam, een kolomnummer, een verwijzing naar een rij in de tabel 'Tabel' en een verwijzing naar een rij in de tabel 'Type'. Deze tabel heeft twee logische sleutels, zie afbeelding 3b.

id	typering	lengte
11	integer	NULL
12	char	64
13	date	NULL

AFBEELDING 3B: DE TABEL 'TYPE'.

id	naam	tabelId	kolomNum	typeld
31	sofinummer	41	1	11
32	naam	41	2	12
33	geboortedatum	41	3	13

AFBEELDING 3C: DE TABEL 'KOLOM'.

De sleutelgegevens bestaan uit een id en een naam. In dit voorbeeld is de naam van de tabel gekozen. De relatiegegevens tussen de tabel 'Sleutel' en de tabel 'Kolom' bestaan uit een verwijzing naar een rij in de tabel 'Sleutel' en een verwijzing naar een rij in de tabel 'Kolom'.

id	naam	sleutelId	kolomId
47	Persoon	47	31

AFBEELDING 3D: DE TABELLEN 'SLEUTEL' (RECHTS) EN 'RELATIESLEUTELKOLOM' (LINKS).

De waardegegevens bestaan uit een rijnummer, een waarde en een verwijzing naar een rij in tabel 'Kolom'. Een deel van de vulling voor deze tabel wordt getoond in afbeelding 3e.

kolomId	rijNummer	waarde
31	1	572550078
32	1	Korporaal
33	1	19361206
31	2	686600009
32	2	Sebus
33	2	19430409
...
33	5	19560401

AFBEELDING 3E : EEN DEEL VAN DE VULLING VOOR DE TABEL 'WAARDE'.

Het adaptief datamodel zoals getoond in afbeelding 2, kan worden gebruikt voor een toepassing waarin de gegevens bestaan uit één

of meer tabellen en een vulling van deze tabellen. Het voordeel van de getoonde aanpak is afhankelijk van de frequentie waarmee nieuwe tabellen aan de toepassing worden toegevoegd. In een traditionele aanpak moet voor iedere nieuwe tabel de toepassing worden aangepast.

VOORBEELD AANGIFTEFORMULIER

Het tweede voorbeeld is ontleend aan de praktijk binnen de Belastingdienst. Jaarlijks verwerkt de Belastingdienst miljoenen aangiften tot aanslagen. De gegevens die op een aangifteformulier worden gevraagd zijn jaarlijks aan wijzigingen onderhevig. Om de toepassing die de aangiften tot aanslagen verwerkt niet afhankelijk te maken van de jaaraanpassingen, dient een adaptief datamodel te worden gebruikt. In afbeeldingen 4a tot en met 4c worden delen van het O-biljet voor het belastingjaar 2001 getoond.



AFBEELDING 4A: VRAAG 1A UIT HET O-BILJET 2001.



AFBEELDING 4B: DE VRAGEN 5A EN 5B UIT HET O-BILJET 2001.



AFBEELDING 4C: DE VRAGEN 12A, 12B EN 12C UIT HET O-BILJET 2001.

Wordt er niet gekozen voor een adaptief datamodel maar een tabel, dan heeft deze mogelijk de vorm zoals getoond in afbeelding 5.

```
create table O_Biljet_2001
(
  id integer,
  sofinummer integer,
  ...,
  heelBelastingjaarGehuwd_Ja char(1),
  heelBelastingjaarGehuwd_Nee char(1),
  deelBelastingjaarGehuwd_Ja char(1),
  deelBelastingjaarGehuwd_Nee char(1),
  ...,
  naamWerkgever_1 char(64),
  ingehoudenLoonheffingTDB_11 integer,
  loon_1 integer,
  naamWerkgever_2 char(64),
  ingehoudenLoonheffingTDB_2 integer,
  loon_2 integer,
  ...,
  totaalIngehoudenLoonheffingTDB integer,
  totaalLoon integer,
  ...,
  waardeOnroerendeZaken_
  BeginBelastingjaar integer,
  waardeOnroerendeZaken_
  EindeBelastingjaar integer,
  waardeOverigeBezittingen_
  BeginBelastingjaar integer,
  waardeOverigeBezittingen_
  EindeBelastingjaar integer,
  ...,
)
```

AFBEELDING 5: EEN TABEL VOOR HET O-BILJET VAN HET BELASTING- JAAR 2001.

Volgens deze aanpak moet ieder jaar een nieuwe tabel worden gemaakt. Een alternatief is één tabel voor alle belastingjaren. Beide oplossingen hebben als nadeel dat de toepassing die de aangiften verwerkt jaarlijks aangepast dient te worden. Dit is niet het geval wanneer gekozen wordt voor een adaptief datamodel.

Een analyse van de gekozen fragmenten uit het O-biljet laat zien dat er drie soorten velden zijn: kruis-, tekst- en bedragvelden. Wanneer alle biljetten in de analyse worden betrokken zal dit aantal nog iets toenemen. Een manier om de tekens die in zo'n veld zijn toegestaan te beschrijven is om ze als reguliere expressies te beschouwen. Een voorbeeld van een reguliere expressie is [0-9]+, hetgeen een sequentie van één of meer cijfers betekent. Deze gegevens worden vastgelegd in een tabel 'DataType', zie afbeelding 6. Een bedragveld hierin is: ofwel het cijfer 0, ofwel optioneel een min-teken, gevolgd door één van de cijfers 1 t/m 9, gevolgd door nul of meer cijfers (0 t/m 9).

id	naam	reguliereExpressie
integer	char(64)	char(64)
1	kruisveld	X
2	tekstveld	.
3	bedragveld	0 -? [1-9][0-9]*

AFBEELDING 6: DE TABEL 'DATATYPE'.

Behalve deze typering van de velden kunnen de velden onderling een verband hebben. Zo staan er op de getoonde fragmenten ja/nee vragen (zie afbeelding 4a) en vragen over de waarde aan het begin en de waarde aan het einde van het belastingjaar (zie afbeelding 4c). In pseudocode kunnen deze samengestelde vragen als volgt worden beschreven:

```
type jaNeeVraag
(
  ja kruisveld,
  nee kruisveld
)
type waardeVraag
(
  waardeBeginBelastingjaar bedragveld,
  waardeEindeBelastingjaar bedragveld
)
```

De vragen uit afbeelding 4a en 4c kunnen nu in pseudocode als volgt worden beschreven:

```
type heelBelastingjaarGehuwd jaNeeVraag
type deelBelastingjaarGehuwd jaNeeVraag
...
type waardeOnroerendeZaken waardeVraag
type waardeOverigeBezittingen waardeVraag
...
```

Ook zijn er velden die een onderlinge samenhang hebben met een herhalend karakter (zie afbeelding 4b, vraag 5a en 5b). In pseudocode kan deze samenhang en de herhaling als volgt worden beschreven:

```
type array inkomstenUitTDB
(
  naamWerkgever tekstveld,
  ingehoudenLoonheffing bedragveld,
  loon bedragveld
)
type array inkomstenUitVDB2
(
  naamUitkeringsinstantie tekstveld,
  ingehoudenLoonheffing bedragveld,
  uitkering bedragveld
)
```

Tot slot zijn er velden (zie afbeelding 4b) die geen onderlinge samenhang hebben en die direct uitgedrukt kunnen worden in de datatyperingen, zie afbeelding 6. In pseudocode kunnen deze velden als volgt worden beschreven:

```
type totaalIngehoudenLoonheffingTDB  bedragveld
type totaalLoon                        bedragveld
type totaalIngehoudenLoonheffingVDB  bedragveld
type totaalUitkering                  bedragveld
```

Een formulering voor de vragen 5a en 5b in afbeelding 4b, die meer overeenkomt met het formulier, in pseudocode:

```
type inkomstenUitTDB
(
  array werkgever
  (
    naam                tekstveld,
    ingehoudenLoonheffing  bedragveld,
    loon                bedragveld
  ),
  totaalIngehoudenLoonheffing  bedragveld,
  totaalLoon                bedragveld
)
```

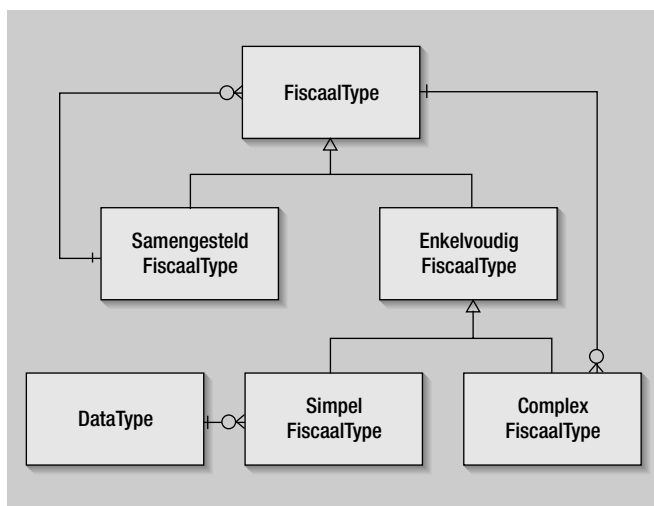
```
type inkomstenUitVDB
(
  array uitkeringsinstantie
  (
    naam                tekstveld,
    ingehoudenLoonheffing  bedragveld,
    uitkering            bedragveld
  ),
  totaalIngehoudenLoonheffing  bedragveld,
  totaalUitkering            bedragveld
)
```

De voorbeelden in pseudocode beschrijven fiscale typeringen. Een analyse van deze fiscale typeringen leert ons dat er twee verschillende vormen zijn: enkelvoudige fiscale typeringen, al dan niet herhalend en samengestelde fiscale typeringen, al dan niet herhalend. De enkelvoudige fiscale typeringen verwijzen naar ofwel een data-typering ofwel een fiscale typering (ofwel enkelvoudig, ofwel samengesteld).

De enkelvoudige fiscale typeringen die refereren aan een data-typering worden simpele fiscale typeringen genoemd.

De enkelvoudige fiscale typeringen die refereren aan een fiscale typering worden complexe fiscale typeringen genoemd. Een schematische weergave van de fiscale typeringen en hun relaties wordt getoond in afbeelding 7.

De besproken pseudocode omgezet in een tabel 'FiscaalType' wordt getoond in afbeelding 8.



AFBEELDING 7: SCHEMATISCHE WEERGAVE VAN DE FISCALE TYPERINGEN.

Via de kolom 'isOnderdeelVan' wordt aangegeven of een rij verwijst naar een ander rij. Op deze wijze wordt de samenstellingsrelatie aangegeven tussen 'Samengesteld FiscaalType' en 'FiscaalType'. Via de kolommen 'dataTypeId' en 'fiscaalTypeId' kan worden vastgesteld welke soort fiscale typering een rij voorstelt. In het geval dat beide kolommen de waarde 'NULL' hebben is het een samengesteld fiscaal type. Als dit niet zo is hebben we een enkelvoudig fiscaal type.

Zoals beschreven bestaan de enkelvoudige fiscale typeringen uit simpele en complexe fiscale typeringen. Voor een simpele fiscale typering heeft de kolom 'dataTypeId' een waarde ongelijk aan 'NULL'. Voor een complexe fiscale typering heeft de kolom 'fiscaleTypeId' een waarde ongelijk aan 'NULL'. Het is niet toegestaan dat beide kolommen 'dataTypeId' en 'fiscaleTypeId' een waarde hebben ongelijk aan 'NULL'.

Nu de fiscale typeringen in kaart zijn gebracht is het mogelijk een deel van een aangifte te gaan invullen. In pseudocode kan dit als volgt worden beschreven:

```
heelBelastingjaarGehuwd.ja          := 'X'
inkomstenUitTDB.werkgever[1].naam   := 'CGE&Y'
inkomstenUitTDB.werkgever[1].
  ingehoudenLoonheffing              := 35000
inkomstenUitTDB.werkgever[1].loon    := 80000
inkomstenUitTDB.totaalIngehouden
  Loonheffing                        := 35000
inkomstenUitTDB.totaalLoon           := 80000
```

De omschrijving links van het '='-teken is een fiscale variabele en de omschrijving rechts van dit teken is een fiscale waarde. In dit voorbeeld is de fiscale variabele afgeleid van de fiscale typering. De fout die nu gemakkelijk gemaakt wordt, is dat de variabele en de typering als gelijkwaardig worden beschouwd.

De omzetting van pseudocode naar tabellen wordt getoond in de afbeeldingen 9a tot en met 9d. De volgende stappen worden

id integer	isOnderdeelVan integer	naam char(128)	dataTypeld integer	fiscaalTypeld integer	isArray boolean
1	NULL	jaNeeVraag	NULL	NULL	false
2	1	ja	1	NULL	false
3	1	nee	1	NULL	false
4	NULL	heelBelastingjaarGehuwd	NULL	1	false
5	NULL	deelBelastingjaarGehuwd	NULL	1	false
6	NULL	waardeVraag	NULL	NULL	false
7	6	waardeBeginBelastingjaar	3	NULL	false
8	6	waardeBeginBelastingjaar	3	NULL	false
9	NULL	waardeOnroerendeZaken	NULL	6	false
10	NULL	waardeOverigeBezittingen	NULL	6	false
11	NULL	inkomstenUitTDB	NULL	NULL	false
12	11	werkgever	NULL	NULL	true
13	12	naam	2	NULL	false
14	12	ingehoudenLoonheffing	3	NULL	false
15	12	loon	3	NULL	false
16	11	totaalIngehoudenLoonheffing	3	NULL	false
17	11	totaalLoon	3	NULL	false
18	NULL	inkomstenUitVDB	NULL	NULL	false
19	18	uitkeringsinstantie	NULL	NULL	true
20	19	naam	2	NULL	false
21	19	ingehoudenLoonheffing	3	NULL	false
22	19	uitkering	3	NULL	false
23	18	totaalIngehoudenLoonheffing	3	NULL	false
24	18	totaalUitkering	3	NULL	false

AFBEELDING 8: DE TABEL 'FISCAALTYPE'.

hierbij doorlopen. Ken aan iedere fiscale variabele een unieke identificatie toe, bijvoorbeeld de identificatie '2' voor de fiscale variabele 'inkomstenUitTDB.werkgever[1].naam'. Substitueer de overeenkomstige fiscale typering-identificatie voor iedere naam in de fiscale variabele, dit geeft 11.12[1].13, omdat 13 onderdeel is van 12 en 12 onderdeel is van 11, zie afbeelding 8. De volgorde 11, 12 en 13 komt tot uitdrukking in de kolom 'sequenceNum' van de tabel 'RelatieFiscaleVariabeleFiscaleTypering', zie afbeelding 9b.

id integer	sequenceNum integer	fiscaalTypeld integer	index integer
1	1	4	NULL
1	2	2	NULL
2	1	11	NULL
2	2	12	1
2	3	13	NULL
3	1	11	NULL
3	2	12	1
3	3	14	NULL
4	1	11	NULL
4	2	12	1
4	3	15	NULL
5	1	11	NULL
5	2	16	NULL
6	1	11	NULL
6	2	17	NULL

AFBEELDING 9B: DE TABEL 'RELATIEFISCALEVARIABELEFISCALETYPERING'.

id integer	naam char(128)
1	heelBelastingjaarGehuwd.ja
2	inkomstenUitTDB.werkgever[1].naam
3	inkomstenUitTDB.werkgever[1].ingehoudenLoonheffing
4	inkomstenUitTDB.werkgever[1].loon
5	inkomstenUitTDB.totaalIngehoudenLoonheffing
6	inkomstenUitTDB.totaalLoon

AFBEELDING 9A: DE TABEL 'FISCALEVARIABELE'.

id	sofnummer	ontvangstdatum
integer	integer	date
	572550078	20020306

AFBEELDING 9C: DE TABEL 'AANGIFTE'.

aangifteld	fiscaleVariabeleld	waarde
integer	integer	char(255)
		X
	2	CGE&Y
	3	35000
	4	80000
	5	35000
	6	80000

AFBEELDING 9D: DE TABEL 'FISCALEWAARDE'.

TENSLOTTE

Met deze twee voorbeelden is inzichtelijk gemaakt hoe een adaptief datamodel kan worden toegepast. We geven tot slot nog enkele voor- en nadelen van het toepassen van een adaptief datamodel. Als voordelen kunnen we aanmerken:

- De periodieke aanpassingen betreffen alleen de tabelinhoud, niet de toepassing;
 - De periodieke aanpassingen kunnen in een relatief korte tijd worden gerealiseerd;
 - De periodieke aanpassingen kunnen door niet ICT-medewerkers worden uitgevoerd.
- En als nadelen kunnen we beschouwen:
- De tabel met de (fiscale) waarden kan zeer groot worden;
 - De periodieke aanpassingen kunnen uitsluitend uitgevoerd worden door voldoende geschoolde medewerkers (zij moeten bekend zijn met het concept);
 - Verder is een speciale ontwikkeling van bijbehorende beheer-toepassingen ten behoeve van de meta-definities nodig. ●

LITERATUUR

Universele Informatiekunde, prof. dr. ir. G.M. Nijssen, PNA Publishing B.V., ISBN 90-5540-001-7
 The theory of Relational Databases, David Maier, Computer Science Press, ISBN 0-914894-42-0

NOTEN

1. TDB - TegenwoordigeDienstBetrekking
2. VDB - VroegereDienstBetrekking

Johan van der Graaf (johan.vander.graaf@cgey.nl) is als consultant werkzaam bij Cap Gemini Ernst & Young.

A G E N D A



Cursussen, seminars e.d.

13/5: Modern Methods & Technology

Masterclass, Utrecht.
 Org./info: CIBIT Adviseurs | Opleiders, www.cibit.nl, 030 230 8989.

14-15/5: Ontwerpen van de nieuwe generatie datawarehouses

Masterclass Datawarehousing met Rick van der Lans, Leiden, Holiday Inn.
 Kosten: € 1250,- (€ 1175,- voor DB/M-abonnees). Org./info: Array Seminars, www.arrayseminars.nl, 036 540 9111.

14-15/5: A pragmatic approach to modeling with UML

Seminar met Sander Hoogendoorn, Brussels, Hotel Sofitel.
 Org./info: I.TWorks, www.itworks.be.

15-16/5: Trends in Databases en Business Intelligence

Cursus o.l.v. dr. E.O. de Brock, Apeldoorn.
 Org./info: PATO, www.pato.nl, 070 364 4957.

22/5: Nieuwe technologie voor software-ontwikkeling, internet en databases

Seminar met Rick van der Lans, Amsterdam, Novotel.
 Org./info: Array Seminars, www.arrayseminars.nl, 036 540 9111.

4-6/6: Informatie-analyse en logisch database-ontwerp

Workshop met Rick van der Lans, Amsterdam, Novotel.
 Org./info: Array Seminars, www.arrayseminars.nl, 036 540 9111.

25-26/6: Fysiek database-ontwerp

Workshop met Rick van der Lans, Amsterdam, Novotel.
 Org./info: Array Seminars, www.arrayseminars.nl, 036 540 9111.

26/6: De (on)zin van XML 'native' databases

Conferentie, Antwerpen, Crowne Plaza Hotel.
 Org./info: SAI, www.sai.be.

Alle vermelde bedragen zijn excl. BTW.