



Hoe krijgt men metagegevens in de database?

Zet de eerste normaalvorm overboord!

Martin Misseyer

Informatie dient in herkenbare begrippen en context te worden gepresenteerd aan een business user. Het is essentieel dat de technische naamgeving waarmee gegevens in een informatiesysteem worden vastgelegd, wordt vertaald in functionele naamgeving en definities, die de gebruiker is gewend. Definities van gehanteerde termen en begrippen zijn slechts één categorie uit een heel scala waarvoor men beschrijvende gegevens kan vastleggen om te gebruiken in selecties, rubricering, en wat al niet meer. Welkom in de wereld van metagegevens.

Aanvullende metagegevens zijn die metagegevens die men voor presentatie en informatieve doeleinden gebruikt in rapportages en bijvoorbeeld in bevraging van de database. Er zijn verschillende alternatieven voor het vastleggen en het beheren van aanvullende metagegevens. Dit artikel behandelt een in de praktijk beproefde methode, die in essentie is gebaseerd op een aantal acties:

- Bepaal de database-objecten waarvoor aanvullende metagegevens gewenst zijn;
- Definieer de vast te leggen aanvullende metagegevens;
- Leg aanvullende metagegevens vast in de logische en fysieke datamodellen;
- Neem bij fysieke implementatie van het datamodel aanvullende metagegevens over in de database;
- Stel de aanvullende metagegevens beschikbaar voor eindgebruik(ers).

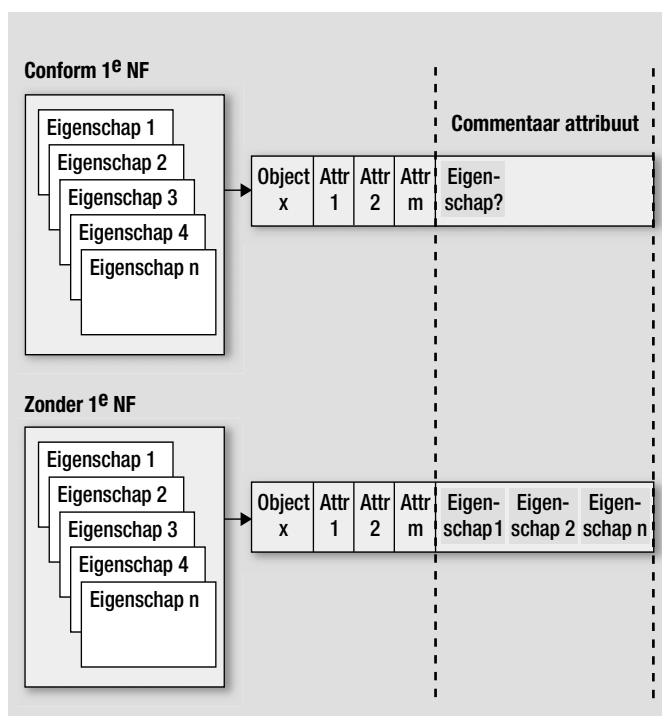
Deze manier voor vastlegging van aanvullende metagegevens is, in vergelijking met andere alternatieven, eenvoudig, efficiënt, goedkoop en effectief.

<p>Logische naam Klant van de organisatie</p> <p>Omschrijving Onder klant van de bank worden alle betalende personen, zowel private als rechtspersonen gerekend</p> <p>Objecttype Functioneel</p> <p>Domein N.v.t.</p> <p>Documentatie DWH_Functioneel_Ontwerp_1_3.doc</p>

Afbeelding 1: De vast te leggen aanvullende metagegevens.

Behoeft aan aanvullende metagegevens

Om informatie aan bijvoorbeeld een business user te kunnen presenteren, dienen uit een database opgevraagde gegevens te worden vertaald in voor de gebruiker bekende (bedrijfs-)begrippen. Bovendien is het wenselijk om, in het geval dat de gebruiker er om zou vragen, deze begrippen nader te verklaren of zelfs toe te kunnen lichten met een voorbeeld, of met een referentie te kunnen verwijzen naar een meer gedetailleerd document dat eventueel online beschikbaar is.



Afbeelding 2: Zonder een structuurwijziging zijn toch aanvullende metagegevens vast te leggen.

```

<scheidingsstring><item><scheidingsstring><cr>
<waarde><cr>
....
<scheidingsstring><item><scheidingsstring><cr>
<waarde><cr>
<scheidingsstring>

Waarbij
<scheidingsstring> een unieke reeks karakters;
<item> te registreren metagegeven;
<waarde> waarde van het te registreren metagegeven, of
indien geen waarde de waarde N.v.t.;
<cr> een carriage return

```

Afbeelding 3: De voorgestelde syntax.

In moderne informatiesystemen, vooral management-informatie-systemen, waarin standaardrapportages zijn te raadplegen of via allerlei 'subjecten' zijn te genereren, is beschikbaarheid van dergelijke definities en omschrijvingen onontbeerlijk. Bij gebrek aan metagegevens blijven waardevolle standaardrapportages steken op een tamelijk technisch niveau of kunnen om dezelfde reden ad hoc rapportages niet door eindgebruikers worden gegenereerd. Laat staan dat eindgebruikers van meer geavanceerde vraag- en navigatie-mogelijkheden gebruik kunnen maken. Afbeelding 1 toont een voorbeeld van vast te leggen aanvullende metagegevens. In dit geval worden er vijf aanvullende metagegevens getoond. Deze gegevens zijn uitermate handig voor rapportagedoeleinden, zoals de *logische naam* en de bijbehorende *omschrijving* van het onderliggende database-object. Daarnaast is het tevens handig om over een *classificatieobject* te beschikken, waarmee de functionele objecten kunnen worden losgeweekt van de niet-functionele. Immers, de doorsnee gebruiker heeft in een rapport niets aan *technische objecten* (versleuteling, historie objecten) of *operationele objecten* (markering, tijdstempeling). Voor ontwikkelaars en beheerders daarentegen zijn juist deze objecten wel essentieel. Tenslotte is het handig om ook een *documentatieverwijzing* op te nemen naar het onderliggende ontwerpdocument waarop het onderliggende object waarvoor de aanvullende metagegevens is gebaseerd.

Mogelijkheden

Wanneer men de genoemde eigenschappen wil vastleggen ten behoeve van query- en rapportagedoeleinden, dan is er een aantal alternatieven beschikbaar. De belangrijkste worden hierna opgesomd.

1. Ontwikkelen van een 'mini' metagegevens repository.

In dit alternatief wordt een specifiek voor de vastlegging van aanvullende metagegevens een 'mini'-repository en bijbehorende

<pre> &&Logische naam&& Klant van de bank &&Omschrijving&& Alle betalende personen, zowel private als rechtspersonen gerekend &&Object&& Functioneel &&Domein&& N.v.t. &&Documentatie&& DWH_FO_1_3.doc && </pre>	<pre> &&Logische naam&& Klantnummer &&Omschrijving&& Uniek 9 cijferig rekeningnummer &&Object&& Functioneel &&Domein&& N.v.t. && </pre>
Tabel/Views	Attributen

Afbeelding 4: Specificatie van aanvullende metagegevens volgens de syntax.

beheerapplicatie en -processen zelf ontwikkeld. Ze vergt een extra inspanning omdat *inhoud* (metagegevens) en *structuur* (autorisaties, ITIL en dergelijke) ingericht en beheerd moeten worden. Een niet onbelangrijk detail is de aandacht voor de vereiste synchronisatie met datamodellen en database dictionary.

2. Aanschaffen metagegevens repository.

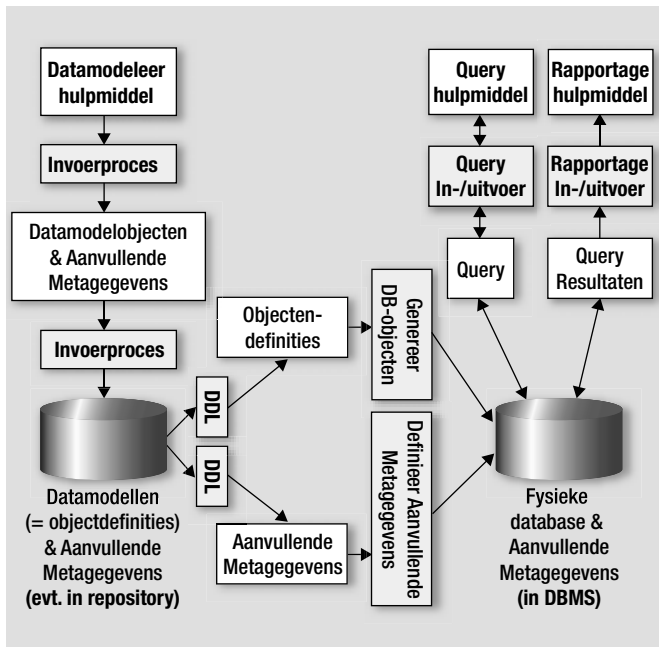
In dit alternatief schaft men, indien men het geld er voor over heeft, een metagegevens repository aan. Hierbij worden dan

Aanvullende metagegevens zijn direct met behulp van SQL uit de catalogusobjecten te bevragen

diverse hulpmiddelen meegeleverd waarmee metagegevens, inclusief aanvullende, worden vastgelegd en ontsloten. In een gekochte metagegevens-repository kunnen nagenoeg alle metagegevens van de organisatie aan elkaar worden geknoopt. Dit maakt een 'kant-en-klare' metagegevens-repository complex (inrichting, beheer, gebruik) en duur (aanschaf, exploitatie). Dit vergt een behoorlijke investering in architectuur en infrastructuur en vereist dat er een strategische keuze ten grondslag ligt aan een beslissing tot aanschaf van een metagegevens-repository, en niet een eenvoudige behoefte aan aanvullende metagegevens.

3. Gebruiken extra functionaliteit datamodelleerhulpmiddel.

In dit alternatief maakt men maximaal gebruik van de functionaliteit van het gebruikte datamodelleerhulpmiddel. Immers, deze functionaliteit is de laatste jaren sterk gegroeid. Enerzijds kennen steeds meer datamodelleerhulpmiddelen een eigen meta-



Afbeelding 5: Het vastleggings- en ontsluitingsproces.

gegevens-repository. Anderzijds is men steeds meer in staat om aanvullende metagegevens 'mee te modelleren' door bijvoorbeeld *extended attributes* (extra eigenschappen) van gegevensobjecten vast te leggen.

In theorie zijn deze extra eigenschappen geschikt om aanvullende metagegevens te bevatten. Echter, er kleven praktische bezwaren aan dit alternatief. Deze hebben voornamelijk te maken met de specifieke inrichting (datamodelleerhulpmiddel-repository), het noodzakelijke maatwerk en de toenemende complexiteit met betrekking tot ontwikkeling en beheer. Wanneer men aanvullende metagegevens uit de datamodelleerhulpmiddel-repository wil rapporteren, dan zijn er doorgaans extra werk en/of additionele hulpmiddelen nodig om dit te realiseren. Dit maakt dit alternatief in aanschaf en exploitatie wederom tamelijk duur.

Bovenstaande alternatieven zijn niet de oplossing voor een simpel probleem van het vastleggen van aanvullende metagegevens. Het kan echter ook anders en vooral veel simpeler. Zie hier het ontstaan van het vierde alternatief.

```

SELECT name,
       SUBSTR(remarks,
              POSSTR(remarks, '&&Logische naam&&') + LENGTH( '&&Logische naam&&') + 2,
              POSSTR(remarks, '&&Omschrijving&&') - LENGTH('&&Logische naam&&') - 5)
       AS "Logische naam",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Omschrijving&&') + LENGTH( '&&Omschrijving&&') + 2,
              POSSTR(remarks, '&&Object&&') - POSSTR(remarks, '&&Omschrijving&&') - LENGTH('&&Omschrijving&&') - 4) AS "Omschrijving",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Object&&') + LENGTH( '&&Object&&') + 2,
              POSSTR(remarks, '&&Domein&&') - POSSTR(remarks, '&&Object&&') - LENGTH('&&Object&&') - 4)
       AS "Object",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Domein&&') + LENGTH( '&&Domein&&') + 2,
              POSSTR(remarks, '&&Documentatie&&') - POSSTR(remarks, '&&Domein&&') - LENGTH('&&Domein&&') - 4)
       AS "Domein",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Documentatie&&') + LENGTH( '&&Documentatie&&') + 2,
              LENGTH(remarks) - POSSTR(remarks, '&&Documentatie&&') - LENGTH('&&Documentatie&&') - 5)
       AS "Documentatie"
FROM SYSIBM.SYSTABLES
WHERE remarks LIKE '%&&%';
    
```

```

SELECT name,
       SUBSTR(remarks,
              POSSTR(remarks, '&&Logische naam&&') + LENGTH( '&&Logische naam&&') + 2,
              POSSTR(remarks, '&&Omschrijving&&') - LENGTH('&&Logische naam&&') - 5)
       AS "Logische naam",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Omschrijving&&') + LENGTH( '&&Omschrijving&&') + 2,
              POSSTR(remarks, '&&Object&&') - POSSTR(remarks, '&&Omschrijving&&') - LENGTH('&&Omschrijving&&') - 4) AS "Omschrijving",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Object&&') + LENGTH( '&&Object&&') + 2,
              POSSTR(remarks, '&&Domein&&') - POSSTR(remarks, '&&Object&&') - LENGTH('&&Object&&') - 4)
       AS "Object",
       SUBSTR(remarks,
              POSSTR(remarks, '&&Domein&&') + LENGTH( '&&Domein&&') + 2,
              LENGTH(remarks) - POSSTR(remarks, '&&Domein&&') - LENGTH('&&Domein&&') - 5)
       AS "Domein"
FROM SYSIBM.SYSCOLUMNS
WHERE remarks LIKE '%&&%';
    
```

Afbeelding 6: SQL DML om resp. tabel en attribuut COMMENTS-strings te ontrafelen in de afzonderlijke aanvullende metagegevens (Voorbeeld DB2 UDB).

Vastlegging

Moderne DBMS'en bieden de mogelijkheid om bij database-objecten als tabellen, views en attributen, specifiek commentaar op te nemen, bijvoorbeeld een omschrijving. Hiervoor biedt SQL het 'COMMENT ON' statement. Omdat de verschillende DBMS'en de betreffende data als karakterreeks bewaren, maakt het technisch niet uit wat er in staat.

Oracle bewaart commentaar in aparte commentaar-catalog-objecten; voor tabellen, views is dit ALL_TAB_COMMENTS en voor op attributen is dit ALL_COL_COMMENTS. DB2 bewaart commentaar in reguliere catalogtabellen, respectievelijk SYSTABLES en SYSCOLUMNS. Dit is mogelijk doordat deze catalogtabellen het REMARKS-attribuut kennen. Overigens ondersteunt Microsoft in SQL Server geen SQL-gebaseerd commentaar.

Elk attribuut in een (R)DBMS dient te voldoen aan de eerste normaalvorm. De eerste normaalvorm bepaalt dat een attribuut atomair is, dat wil zeggen slechts één waarde kan bevatten. Dit is weergegeven in het bovenste deel van afbeelding 2. Echter, in het kader van de vastlegging van aanvullende metagegevens wordt gesteld dat het geen halszaak is om de eerste normaalvorm te schenden. Beter gezegd: de eerste normaalvorm is in het onderhavige geval niet van toepassing. Dit maakt het mogelijk om zonder een structuurwijziging, dat wil zeggen met een additionele metagegevens-repository, toch aanvullende metagegevens vast te



Afbeelding 7: Blik op de tabel COMMENTS-strings in de DB2 Catalog via het DB2 UDB Control Center.

leggen. Immers, in die ene tekenreeks zijn, technisch gezien, meerdere aanvullende metagegevens vast te leggen. Dit is weergegeven in het onderste deel van afbeelding 2. Voorwaarden zijn echter wel dat dit absoluut betrouwbaar, consistent en gecontroleerd gebeurt.

FYSIEKE NAAM	LOGISCHE NAAM	OMSCHRIJVING	OBJECT	DOMEIN	DOCUMENTATIE
DEBITEUR	Debiteur	Klanten die bekend staan als debiteur. Vanaf 61 dagen wordt men dubieuze debiteur..	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
DEBITEUR_DUBIEUS	Debiteur Dubieus	Klant die meer dan 60 dagen factu(ur)en open heeft staan.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
DOELGROEP	Doelgroep.	Doelgroep is een clustering of categorisering van klant.	Functioneel	Intern	Functioneel_Ontwerp_DWH_release0_2_1.doc
FACTUUR	Factuur	De rekening voor de producten (aantal x prijs) die zijn aangeschaft.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
FINANCIELE_INFORMATIE_EXTERN	Financiële informatie extern	Bevat de belangrijkste te rapporteren KPIs per maand van de afgelopen 5 jaar.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
FINANCIELE_INFORMATIE_JAARVERSLAG	Financiële informatie extern	Bevat de belangrijkste te rapporteren KPIs van de afgelopen 5 jaar, per jaar.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
KALENDER	Kalender	Bevat alle data/datum van de afgelopen en komende 10 jaar.	Functioneel	Intern	Functioneel_Ontwerp_DWH_release0_1_1.doc
KALENDER_VAKANTIE_EN_FEESTDAGEN	Kalender vakantie en feestdagen	Bevat alle dagen waarop er in het bedrijf niet wordt gewerkt..	Functioneel	Intern	Functioneel_Ontwerp_DWH_release0_1_1.doc
KLANT	Klant	De klant verkoopt de organisatie haar producten aan.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
LAND	Land	Officieel land in de wereld.	Functioneel	ISO	Functioneel_Ontwerp_DWH_release0_1_5.doc
LEVERANCIER	Leverancier	Leveranciers zijn de bedrijven die de producten/grondstoffen leveren.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
LEVERANCIERKLASSE	Leverancierklasse	Dit is een kwalificatie van een leverancier.	Functioneel	Intern	Functioneel_Ontwerp_DWH_release0_1_3.doc
MANAGEMENT_INFORMATIE_INTERN	Management informatie extern	De belangrijkste interne te rapporteren KPIs per maand.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
MEDEWERKER	Medewerker	Medewerker bevat alle relevante persoonsgegevens van de medewerker van de organisatie.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
MEDEWERKER_ZIEKTEVERZUIM	Medewerker ziekteverzuim	Ziekteverzuim per medewerker.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
OFFERTE	Offerte	Offertes zijn aan de klant aangeboden aanbiedingen, inclusief korting(en).	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
OFFERTE_REGEL	Offerteregulering	Offerteregels zijn de detailinformatie (potentiele) productverkoop...	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
ORGANISATIE	Organisatie.	De organisatie bestaat uit vele eenheden, onderdelen, afdelingen, en dergelijke.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
POSTCODE	Postcode	Dit is de postcode tabel van KPN. De tabel wordt maandelijks geactualiseerd.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_Domeinen_release1_1_3.doc
PRODUCT	Product	Producten zijn de objecten die de organisatie verkoopt aan klanten.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
RECHTSPERSOON	Rechtspersoon	Dit is de juridische vorm van de betreffende organisatie of klant.	Functioneel	Intern	Functioneel_Ontwerp_DWH_release0_1_2.doc
TITEL	Titel	Titels die gevoerd kunnen worden.	Functioneel	Algemeen	Functioneel_Ontwerp_DWH_release0_2_1.doc
VERKOOPTEAMS	Verkoopteams	Dit zijn de verkoopteams van de verkooporganisatie.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc
VERKOOPTEAMS_DETAIL	Verkoopteams individueel	Dit zijn medewerkers per verkoopteam.	Functioneel	N.v.t.	Functioneel_Ontwerp_DWH_release1_1_0.doc

Afbeelding 8: De aanvullende metagegevens zoals deze feitelijk in de catalog van het DBMS worden geadmistreerd.

Derhalve is het noodzakelijk dat er een strikte syntax wordt afgesproken waarmee meerdere aanvullende metagegevens aan elkaar kunnen worden geregen tot één tekenreeks.

Afbeelding 3 toont de voorgestelde syntax. De syntax wordt als volgt toegelicht:

- een aanvullend metagegeven bestaat uit een naam en een waarde;
- de scheidingsstring is vrij te kiezen, in dit artikel wordt && gehanteerd;
- de scheidingsstring bestaat uit meerdere gezamenlijk unieke karakters;
- de scheidingsstring bestaat uit een karaktercombinatie die niet voorkomt in gewone tekst;
- een scheidingsstring wordt eenmalig gedefinieerd;
- de volgorde van vastlegging van aanvullende metagegevens is altijd hetzelfde;
- er kunnen geen lege regels worden gedefinieerd, de standaardwaarde luidt dan 'n.v.t.';
- elke regel dient te worden afgesloten met een 'volgende-regel-teken';
- de laatste regel bevat alleen een specifiek scheidingsstring;
- neem geen onderhoudsgevoelige zaken op in aanvullende metagegevens;

Ten aanzien van dit laatste kan men bijvoorbeeld denken aan het verwerken van een URL in een documentnaam. Of het onnodig specificeren van het aanvullend metagegeven 'Documentatie' voor

attributen. Dit is overbodig, immers het volstaat wanneer men het aanvullend metagegeven 'Documentatie' voor tabel/view definieert. Afbeelding 4 toont een voorbeeld van de specificatie van aanvullende metagegevens volgens de zojuist behandelde syntax. Zoals uit het voorbeeld is af te leiden, is deze gebaseerd op de in afbeelding 1 geschetste behoefte aan aanvullende metagegevens.

Ontsluiting

Het gehele vastleggings- en ontsluitingsproces is te zien in afbeelding 5. Allereerst worden in datamodellen aanvullende metagegevens conform de opmaak vastgelegd. Met betrekking tot aanvullende metagegevensinvoer is het handig om vooraf templates te definiëren met standaardwaarden. Deze templates kunnen dan telkens in een datamodel en fysiek datamodelobject worden toegevoegd, en kunnen die waarden die specifiek zijn worden aangepast en/of ingevuld. De tijd die nodig is voor de vastlegging van aanvullende metagegevens hangt af van het aantal datamodelobjecten, het aantal items en de omvang van de tekst.

In het kader van de vastlegging van aanvullende metagegevens is het geen halszaak om de eerste normaalvorm te schenden

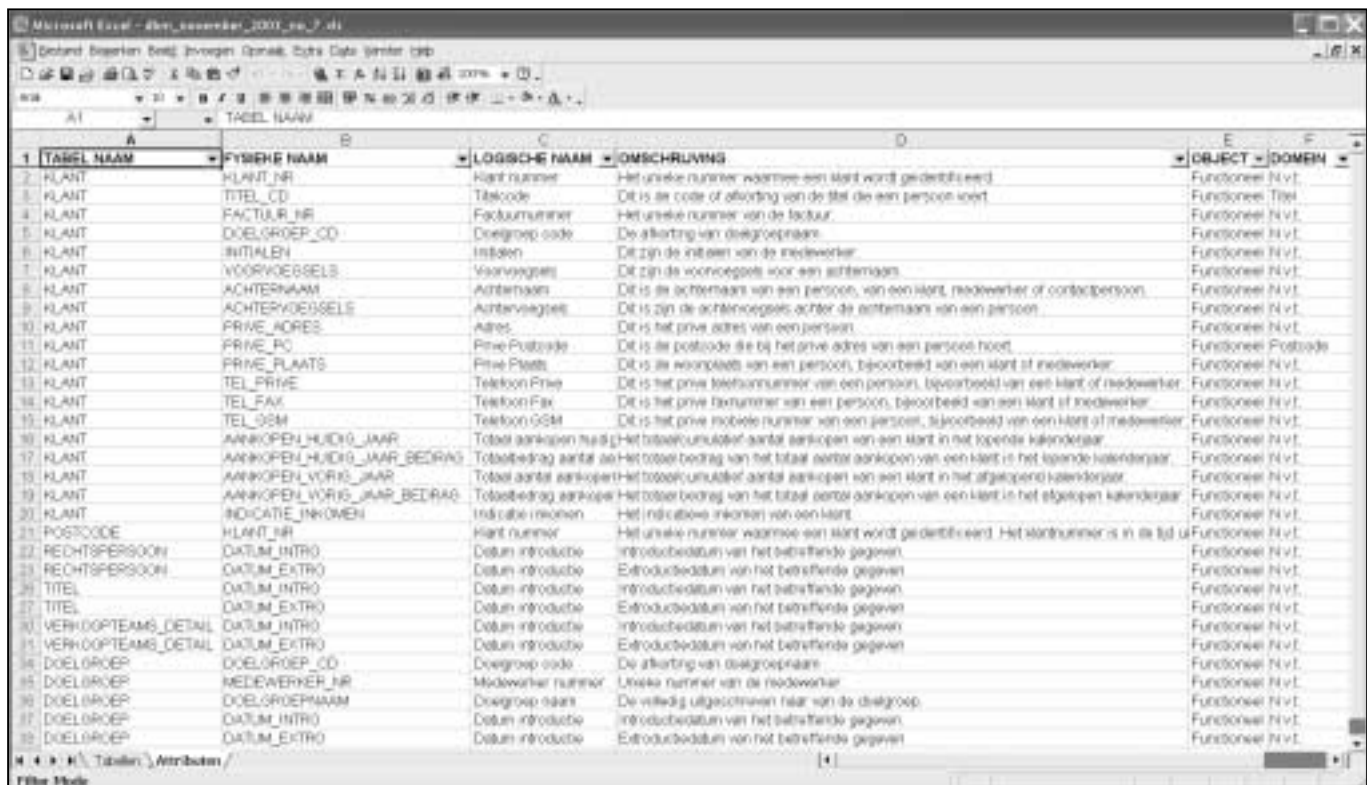
Vanuit een fysiek datamodel wordt uiteindelijk een database uitgebreid en/of gewijzigd. Om de vastgelegde aanvullende metagegevens in de database te krijgen worden de COMMENT-statements mee gegenereerd. Om te voorkomen dat men in elke individuele query of rapport de aanvullende metagegevens uiteen moet rafelen, is het wenselijk om aangepaste views te definiëren op de commentaarobjecten van het DBMS. Afbeelding 6 toont twee SQL DML-statements. Afbeelding 7 toont dat aanvullende metagegevens zijn vastgelegd in de database. Afbeelding 8 toont de aanvullende metagegevens zoals deze feitelijk in de catalog van het DBMS worden geadministreerd.

Aanvullende metagegevens zijn direct met behulp van SQL uit de catalogusobjecten, of indirect via de hiervoor besproken views, te bevragen. Deze aanvullende metagegevens zijn zonder probleem te combineren met andere (meta)gegevens. Vanuit een ontwikkel- en beheeroogpunt is het wenselijk om standaard-rapportages te definiëren, bijvoorbeeld een setje per datamodel, per interface, of per gebruikersgroep. Deze standaardrapportages kunnen in verschillende systeem- en acceptatietesten worden gebruikt om te controleren of alle relevante aanvullende metagegevens op een correcte wijze zijn vastgelegd en beschikbaar zijn. Afbeelding 10 laat een 'voorbeeldrapportage' zien in Microsoft Excel. Natuurlijk zijn er met behulp van BI-tools veel fraaiere

- 1 = Eigen repository;
 2 = Gekochte repository;
 3 = Extra functionaliteit data modelleerhulpmiddel;
 4 = Via datamodel als database COMMENT vastleggen.

Item	Alternatief	1	2	3	4
Hergebruik bestaande procedure		--	--	--	+
Hergebruik bestaande hulpmiddelen		-/+	--	-/+	++
Initiële kosten		-	--	-	++
Exploitatiekosten		-/+	--	-/+	+
Ontsluitbaarheid		++	++	-/+	++
Inbedding in ITIL		+	+	+	++
Theoretische zuiverheid (INF)		++	++	++	--
Mate van handwerk		+	-/+	-/+	+
Beperkingen DBMS (Comments)		n.v.t.	n.v.t.	n.v.t.	+/-
Complexiteit (aantal componenten)		-	--	--	+
Mate van flexibiliteit		++	+	-/+	+
Schaalbaarheid		+	++	+	-/+
Autorisatiemechanisme (hoev. werk)		-	--	-	++
Voorkeursalternatief		2	4	3	1

Afbeelding 9: Globaal beeld van de scoring per besproken alternatief.



Afbeelding 10: Microsoft Excel als frontend. Via Menu-item 'Data' -> 'Externe gegevens ophalen' -> 'Kies Gegevensbron'; naar de juiste database om een aanvullende metagegevens-query te draaien of een geprefabriceerde aanvullende metagegevens-view te openen.

rapportages te creëren. Overigens, dit voorbeeld is gebaseerd op gecreëerde views op de betreffende DBMS-catalogtabellen, waarin het technische commentaarattribuut reeds uiteten is gerafeld. Als men de reguliere beheer- en onderhoudsprocedure volgt, worden wijzigingen doorgevoerd in de logische en fysieke datamodellen. Dit impliceert dat dan overeenkomstig het betreffende beheer en/of onderhoud aanvullende metagegevens worden toegevoegd, gewijzigd en/of verwijderd. Vanzelfsprekend wordt dit dan conform het hierboven beschreven proces doorgevoerd in de database.

Voorkeursalternatief

Afbeelding 9 geeft een globaal beeld van de scoring per besproken alternatief. Als men een efficiënte en effectieve methode wenst voor het vastleggen van aanvullende metagegevens, dan is de in dit artikel beschreven methode zeer geschikt. De belangrijkste voordelen van dit alternatief, nogmaals op een rijtje gezet, zijn:

- Geen extra metagegevens-repository en bijbehorende hulpmiddelen;
- Geen beheerapplicatie hoeven te ontwikkelen;
- Geen geavanceerde datamodelleer-hulpmiddelfunctionaliteit hoeven te gebruiken;
- Geen additioneel ontwikkel- en/of beheerproces;
- Maximaal hergebruik beschikbare hulpmiddelen.

Dit zegt overigens niet dat deze methode altijd dient te worden verkozen boven een van de andere. De behoefte aan metagegevens in een organisatie kan (veel) groter zijn, er kunnen reeds zwaardere hulpmiddelen in de organisatie aanwezig zijn, enzovoort.

Aanbevelingen

Als men er voor kiest om de beschreven methode voor de vastlegging van aanvullende metagegevens te willen implementeren, dan is een aantal aanbevelingen op zijn plaats.

In de eerste plaats is het mechanisme uit te breiden voor meer type data(base)-objecten (stored procedures, functies en dergelijke), maar ook voor basistabellen, basis-views en gebruikspecifieke views.

In de tweede plaats betreft het ruimtebesparing. Omdat sommige DBMS-leveranciers niet al te scheutig zijn met de toebedeling van commentaar-ruimte, is het aan te bevelen om de te benoemen technische velden zo compact mogelijk te houden. Men kan er voor kiezen om hiervoor een (alfa)numeriek karakter te gebruiken.

Martin Misseyer

Dr. Martin P. Misseyer (martin.misseyer@ordina.nl) is Profession Leader bij Ordina Finance SDS b.v.