

# XML DB

## SQL en XML in hybride Oracle database

*Met de introductie van Oracle9i heeft Oracle de fundamenteën gelegd van een nieuwe generatie Database Management Systemen. De Oracle database evolueert naar een hybride product dat het midden lijkt te houden tussen de traditionele RDBMS en de recentere generatie native XML DBMS. In dit artikel licht Harold Gerritsen de stand van zaken toe op het gebied van XML in de Oracle database, vanaf Oracle9iR2 aangeduid met XML DB.*

Het gebruik van RDBMS'en is al jaren gemeengoed voor vrijwel elk serieus opererend bedrijf. Je kunt dan ook wel stellen dat het zuivere databaseaspect van de recente RDBMS'en niet verder geïnnoveerd kan worden. Bovendien zijn revolutionaire ontwikkelingen, zoals jaren geleden de introductie van het eerste Object DBMS ('Illustra'), amper aangeslagen in de markt. Oracle, die niet verder ging dan het implementeren van het objectrelationele principe in Oracle8, had dat kennelijk juist ingeschat.

Hoe anders gesteld is het met de recentere ontwikkelingen op databasegebied. Voortkomend uit de hype van XML ontstond er behoefte aan het opslaan in de database en vervolgens kun-

***Vanaf Oracle9iR2 kun je het Oracle RDBMS met hetzelfde recht een Oracle XDBMS noemen***

nen benaderen van complete XML documenten en fragmenten ervan. Een RDBMS is echter niet bij uitstek geschikt is voor deze functionaliteit, er bestaan niet voor niets aparte textservers in het portfolio van de meeste RDBMS leveranciers. Dankzij de relatieve achterstand van de gevestigde orde RDBMS-leveranciers voor ondersteuning van XML documen-

ten, ontstond er al snel een nieuwe categorie databaseleveranciers en bijbehorende databaseproducten<sup>1</sup>. De eerste zogenaamde native XML database, momenteel marktleider in dit segment, is de eXtensible Information Server, van het Amerikaanse bedrijf eXcelon Corporation. De grootste concurrenten in de Nederlandse markt zijn het Duitse Software AG met Tamino XML Server en het Nederlandse X-Hive met X-Hive/DB. Deze laatste krijgt overigens wereldwijd steeds meer bekendheid en wordt veelvuldig genoemd in performance benchmarks. In no time creëerde deze categorie databaseleveranciers marktverraag van ongekende omvang voor de XML DBMS'en. En ziedaar, deze ontwikkeling wordt door Oracle wel degelijk serieus genomen.

### Onbegrensde mogelijkheden

Vanaf Oracle9iR2 kun je het Oracle RDBMS met hetzelfde recht een Oracle XDBMS noemen (al zal dit uiteraard te vuur en te zwaard worden bestreden door de inmiddels gevestigde orde van native XML DBMS leveranciers). In deze release kan de database namelijk gebruikt worden als een heuse repository voor XML documenten en bestanden, voorzien van een echte bestandsstructurering (overeenkomstig het principe van directories in een Operating System), met mogelijkheden voor versiebeheer en autorisatie (door middel van het vastleggen van Access Control Lists), en last but not least, te benaderen met behulp van de standaard internet protocollen HTTP, FTP en WebDAV.

Als je hierbij bedenkt dat vrijwel alle moderne office en internet development suites native ondersteuning bieden voor de genoemde internetprotocollen, begin je langzaam onder de indruk te raken van de onbegrensde mogelijkheden die de jongste generatie database lijkt te bieden. Zo kan een ontwikkelaar die gebruikmaakt van bijvoorbeeld JDeveloper, of een contentbeheerder die de Macromedia of Adobe content management tools gebruikt, de XML-bestanden die tijdens zijn werkzaamheden worden geproduceerd, transparant met behulp van WebDAV in de Oracle database vastleggen (als ware het een bestandssysteem). Een andere persoon zou de gegevens

in deze documenten (mits hij daartoe is geautoriseerd) met behulp van SQL kunnen bewerken. Daarnaast zou de inhoud ook nog zichtbaar gemaakt kunnen worden door de betreffende bestanden, slechts gebruikmakend van een recente versie van Microsoft's Internet Explorer, met behulp van het protocol HTTP te benaderen.

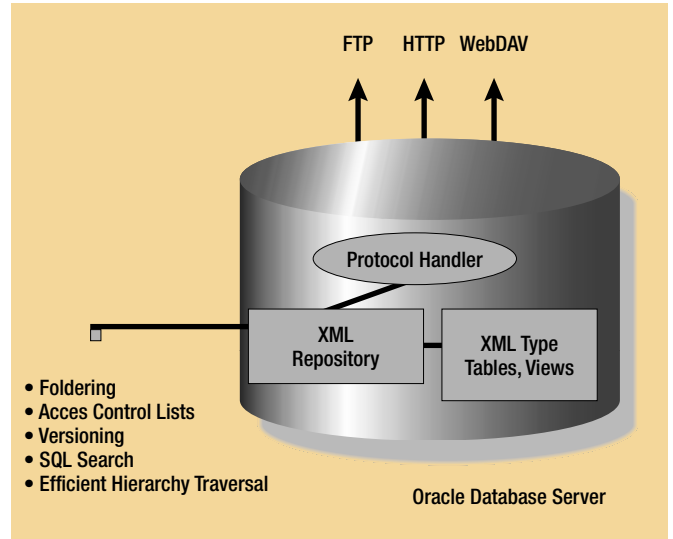
Nieuwsgierig geworden? In de rest van dit artikel zullen we wat dieper op de details ingaan.

## XML support in Oracle

Een veelgestelde vraag door klanten is: "in hoeverre wordt XML eigenlijk ondersteund door Oracle?". Het antwoord hierop is echter niet eenduidig. Als we Oracle als synoniem voor het bedrijf interpreteren kunnen we zeggen dat XML zeer sterk wordt ondersteund. Oracle heeft het leveren van een platform voor XML en Java zelfs tot haar missie verheven. Als we Oracle echter interpreteren als een synoniem voor haar RDBMS dan hebben we een heel ander verhaal. Uiteraard, Oracle's marketingafdeling wil ons al jaren doen geloven dat Oracle's database alle denkbare mogelijkheden biedt voor XML. Maar wat je als ontwikkelaar moet doen om daar gebruik van te maken is toch wel sterk afhankelijk van de gebruikte database release.

Als we allereerst kijken naar Oracle8i hebben we met name mogelijkheden voor het koppelen van de database aan de buitenwereld, waarbij we met die buitenwereld in de vorm van XML berichten communiceren (de zogenaamde XML middleware). De interfaces die deze berichten samenstellen of ontleden dienen zelf ontwikkeld te worden. Hiervoor levert Oracle al jaren de gratis van OTN te downloaden 'XML Developers Kit (XDK)'. Deze bevat de class generators (programmaroutines die vanuit de database data kunnen halen en in het juiste XML berichtformaat kunnen plaatsen) en parsers (die het ontleden van het bericht voor hun rekening nemen en daarmee de tegengestelde richting ondersteunen), zowel beschikbaar in PL/SQL als Java. Voor transformatiedoeleinden van XML naar bijvoorbeeld HTML is er ook een specifieke XSLT processor opgenomen in de toolkit. Daarop voortbordurend bevat de XDK de zogenaamde XSQL servlet, waarmee XML, SQL en XSLT in de server worden gecombineerd om dynamisch web content te genereren. Een ander onderdeel van Oracle8i is het internet File System (IFS). Met dit product kun je het RDBMS benaderen als ware het een File System. Doordat het IFS een veelvoud aan standaarden ondersteund is het bijvoorbeeld mogelijk binnen Microsoft Windows een Netwerkdriev te mappen op c.q. te koppelen aan een Oracle database.

Hierdoor wordt een van de basisfunctionaliteiten van het Operating System, nl. File Management, overgenomen door de Oracle software. In feite is dit de eerste voorloper van



Afbeelding 1. Overzicht XML support voor Content Management in de Oracle database

Content Management support zoals dat in XML DB (9iR2) is geïmplementeerd.

## Ondersteuning in 9i

Kijken we vervolgens naar XML ondersteuning in Oracle9i dan zien we dat de XML Developers Kit verder ontwikkeld is, maar vooral dat er daadwerkelijke ondersteuning is gekomen voor het opslaan van XML documenten in de database. Hiervoor is het datatype XMLType aan de standaard datatypes van Oracle toegevoegd. En is in Oracle9iR1 het XML Type nog geïmplementeerd als een binair datatype met een reguliere textindex, vanaf 9iR2 kan er ook voor worden gekozen het XMLType te implementeren als een object-relatieve datatype, met als gevolg een veel performantere oplossing. XML documenten welke eenmaal zijn vastgelegd in de structuur van het XMLType datatype, kunnen op een veelvoud van manieren benaderd worden. Hierbij zijn de benaderingswijzen globaal te verdelen in twee categorieën, de wijze van benaderen vanuit de Content Management optiek, en vanuit de Data Management optiek.

## Content Management

Als we XML DB beschouwen vanuit een content management optiek, dat kunnen we dat schematisch weergeven zoals in afbeelding 1. Hierin zien we dat de Oracle9i database in feite een repository is voor XML documenten, waarbij geldt dat we deze repository met behulp van een drietal protocollen kunnen benaderen. Deze protocollen zullen we eerst wat uitgebreider bekijken. Twee van deze protocollen zullen de meeste ontwikkelaars zeker meer dan bekend voorkomen. Het protocol HTTP (HyperText Transfer Protocol) is het standaard protocol voor overdracht over TCP/IP netwerken van inter-, intra- en extranet gerelateerde documenten ('content'), zoals pagina's

van websites. FTP (File Transfer Protocol) is naar omvang van gebruik de grote tweede voor overdracht over dergelijke netwerken, maar bestond in feite al ver voor het internet z'n enorme vlucht had genomen. Zoals de naam al aangeeft werd dit protocol al vroeg gebruikt door automatiseerders en systeem-beheerders om bestanden over te zetten van de ene machine op de andere. Hoewel deze protocollen zoals gezegd al geruime tijd bestaan, zijn ze redelijk beperkt in functionaliteit. HTTP is vooral bedoeld om content 'op te halen', en kan eigenlijk alleen met behulp van het zogenaamde 'POST' commando data versturen naar de HTTP Server. FTP is wat dat betreft uitgebreider. Met FTP is dezelfde functionaliteit beschikbaar om data op te halen als te versturen, maar kan bovendien vanaf de client genavigeerd worden door het filestelsel van de FTP Server, zodat op die wijze het juiste bestand gevonden kan worden.

## WebDAV

Het derde protocol dat XML DB ondersteunt is WebDAV, een extensie van HTTP, en is te beschouwen als de opvolger van HTTP en FTP. Erg toegankelijke informatie over WebDAV is overigens terug te vinden op de WebDAV Resources site<sup>2</sup> van de WebDAV community. In feite kun je WebDAV zien als een faciliteit voor groupware functionaliteit op het internet. WebDAV staat voor World Wide Web Distributed Authoring and Versioning. Het is de standaard van het Internet Engineering Task Force (IETF) die het in samenwerkingsverband bewerken van teksten en content mogelijk maakt en file management faciliteiten biedt voor gebruikers die onderling verbonden zijn via het internet. De belofte van WebDAV is dat het het ontstaan van virtuele ondernemingen een enorme impuls zal geven. Daar steeds meer tools WebDAV compliant zijn kan met dergelijke tools daadwerkelijk samenwerking ontstaan via het internet. Simpel gezegd kun je stellen dat WebDAV de leesmogelijkheden van HTTP uitbreidt met schrijfmogelijkheden.

***De belofte van WebDAV is dat het het ontstaan van virtuele ondernemingen een enorme impuls zal geven***

Hierdoor wordt het internet in combinatie met WebDAV Servers als platform net zo veelzijdig als lokale netwerken in combinatie met File Servers. Hoe transparant WebDAV is voor eindgebruikers wordt het best geïllustreerd door de WebDAV mogelijkheden van bijvoorbeeld Microsoft XP. In de Explorer kunnen verschillende typen netwerk locaties (zoals FTP of

WebDAV Servers) worden toegevoegd, waarna deze locaties als reguliere filesystemen gebruikt kunnen worden. Ook in de moderne Internet Explorer vind je ondersteuning voor WebDAV. Hiervoor is een node met de naam 'Web Folders' zichtbaar. In een dergelijke browser kan via deze node genavigeerd worden naar een WebDAV Server (bijvoorbeeld een Oracle database met XML DB), en kan binnen de filestructuur van die server verder genavigeerd worden.

## Versiebeheer

Zoals we nu hebben gezien biedt het WebDAV protocol de basisfunctionaliteit om als fileserver op het internet te fungeren. Maar een WebDAV server is veel moderner dan dat. Vergelijken we een WebDAV server met een 'gewone fileserver' uitgerust met een 'gewoon operating system', dan zien we dat WebDAV uitstekende faciliteiten heeft voor versiebeheer. De grote kloof die er is wat betreft versiebeheermogelijkheden tussen het Open VMS en het UNIX besturingssysteem, kun je vergelijken met de kloof tussen een WebDAV server en een Microsoft Windows File server. Het WebDAV protocol biedt uitgebreide mogelijkheden voor versiebeheer op documenten, en het toekennen van rechten op deze documenten. Aangezien XML DB WebDAV compliant is, zijn deze mogelijkheden volledig geïmplementeerd. Uit al het voorgaande kunnen we afleiden dat XML DB uitstekend ingezet kan worden als repository voor tekstdocumenten. De link met XML is echter nog niet echt gelegd. Welnu, door met behulp van de Content Management protocollen (als we ze even zo mogen noemen) XML content vast te leggen in de repository van XML DB, daarmee onder versiebeheer geplaatst en benaderbaar volgens de ingerichte autorisatiestructuur, kan er gebruik gemaakt worden van een groot scala aan functies in Oracle die het mogelijk maken de inhoud van de XML documenten te benaderen. Daarbij is er op verschillende manieren door de inhoud van de documenten te navigeren. Vanuit de Content Management optiek willen we vooral de opslag van de XML bestanden goed kunnen structureren en managen. Het gemakkelijk kunnen terugvinden van bestanden is daarbij essentieel. Hiervoor wil je eigenlijk zoekmogelijkheden hebben vergelijkbaar met de zoekfunctionaliteit die Microsoft in Windows biedt om File Searches te doen. De File Search van Windows biedt de mogelijkheid binnen een bepaald pad te zoeken, op bestanden met bepaalde eigenschappen en/of met een bepaalde inhoud. In XML DB kunnen we dit allemaal doen met SQL, welke speciaal hiervoor is uitgebreid met functies zoals UNDER\_PATH(), EQUALS\_PATH(), DEPTH() etc. Met deze functies kunnen we bijvoorbeeld heel gericht vanaf een bepaalde plaats in de folderstructuur gaan zoeken, tot een bepaalde diepte, naar documenten met bepaalde eigenschappen of een bepaalde inhoud. Een voorbeeld van een statement waarin deze functies worden gebruikt luidt als volgt: Met EXTRACT\_VALUE() kan van een geselecteerd element de

# 10: Advertentie Whitehorses

```

select      extractvalue(res
            , '/' || Voornaam' )
,
            path(1)
from        path_view pvw
where       under_path(pvw
            , 10 /* diepte */
            , '/klanten/cursisten' /* root folder */) > 0
and         extractvalue(res
            , '/' || Achternaam' ) = 'GERRITSEN'
/

```

waarde worden opgevraagd. PATH() geeft de relatieve padnaam van de geselecteerde instantie. De view PATH\_VIEW bevat een rij voor elk unieke padnaam in de XML DB repository. De operator UNDER\_PATH() tot slot geeft TRUE voor elke gevonden instantie onder de aangegeven root folder binnen de optioneel aangegeven diepte. Het voorbeeld statement retourneert dus alle instanties die minder dan tien niveaus diep in de "/klanten/cursisten" folder zijn vastgelegd, en die een element bevatten met de naam 'Achternaam', en de waarde 'GERRITSEN'.

Soms willen we de inhoud van XML bestanden niet zozeer bekijken ten behoeve van File Management, maar willen we veel gericht bezig gaan met fragmenten van de XML bestanden. We vergeten dan als het ware het concept van 'files', en beschouwen de inhoud als gewone data, maar in de vorm van XML in de database gebracht. Het op die wijze ontginnen en bewerken van de inhoud van deze documenten kunnen we scharen onder de noemer Data Management.

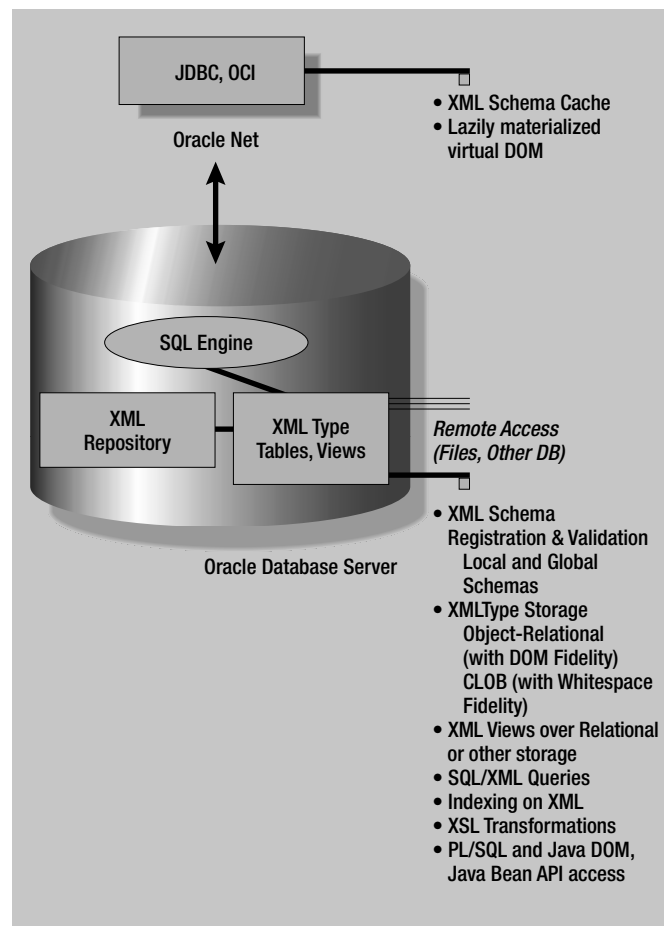
## Data Management

Als we XML DB beschouwen vanuit een Data Management optiek, dat kunnen we dat schematisch weergeven zoals in afbeelding 2. Hierin zien we dat de Oracle9i database een verzameling tabellen en views is, geoptimaliseerd om XML data in op te slaan. De inhoud van deze tabellen kunnen we op vele wijzen bevragen en bewerken met behulp van zowel SQL als met methoden die in de XML wereld gebruikelijk zijn. Hetzelfde geldt voor het creëren van de definitie van de tabellen. De definitie kan met behulp van een conventioneel Data Definition Language (DDL) statement worden gecreëerd, of via het zogenaamde 'registreren van een XML Schema'. Om dit te verduidelijken zullen we nog even een zijstapje maken naar XML.

## Tabeldefinitie

Voor een schemadefinitie in Oracle (vastgelegd in de data-dictionary, die daarmee de metastructuur van de vast te leggen data bevat), bestaat er een equivalent in de XML wereld. Bij het ontstaan van de XML standaard waren hiervoor zogenaamde Document Type Definitions (DTD's) erg in trek. Deze DTD's vormen eigenlijk documenten waarin de structuur van de metagegevens wordt gedefinieerd, vergelijkbaar met de data-

dictionary in Oracle. In XML files met feitelijke data kan simpelweg met behulp van een URL naar de bijbehorende DTD worden verwezen. Op die manier is het voor een geautomatiseerd tool mogelijk validaties te doen van de feitelijke data tegen de gedefinieerde metastructuur. Zo zul je in de DTD ook onderdelen tegenkomen die de equivalent vormen van het begrip 'datatype' uit de databasewereld. In de definitie van DTD's is het specificeren van datatypes echter niet in zulk detail mogelijk als in de DDL van hedendaagse RDBMS'en. Daarom is het belang van DTD's de afgelopen jaren afgenomen, waarvoor XML Schema in de plaats is gekomen, een geavanceerdere opvolger. XML Schema is een officiële standaard van het World Wide Web Consortium (W3C). Oracle9iR2 maakt het mogelijk om een schema in te richten door het opgeven van een XML Schema definition. Het opgeven daarvan, het zogenaamde 'registreren', gebeurt met behulp van een PL/SQL package. Het enige dat daarvoor nodig is, is de URL van het document met de XML Schema definitie. XML DB laadt de schema definitie, compileert de informatie om runtime snel beschikbaar te hebben, en maakt daarbij eventueel object relationele tabellen aan, alsmede indexen en Java Classes. Wat tijdens de compilatie



Afbeelding 2. Overzicht XML support voor Data Management in de Oracle database

eigenlijk gebeurt, is het aanvullen van de geladen XML Schema definitie met informatie over de mapping van de verschillende schema-elementen op de gecreëerde object-relatieve database-objecten. Deze mapping is overigens eenvoudig aan te passen met XML Schema editing tools zoals XML Spy van Altova Corporation. Hierbij moet wel worden beseft dat de flexibiliteit van de mapping niet hetzelfde is als flexibiliteit ten aanzien van de structuur waarop gemapt wordt, bijvoorbeeld de structuur van de object-relatieve tabellen die tijdens de XML Schema registratie default zijn gegenereerd. Voor het wijzigen daarvan gelden dezelfde beperkingen als voor reguliere DDL operaties.

Naast het registreren van een XML Schema kunnen we uiteraard ook nog op de conventionele wijze een tabel definiëren waarin XML gegevens zullen worden vastgelegd. Als de structuur van de vast te leggen XML gegevens is gedefinieerd in een XML Schema definition, dan kunnen we daar in de DDL naar verwijzen. Een voorbeeld van een statement waarin dit wordt toegepast luidt als volgt:

```
create      table xml_aanmeldingen
(info xmltype)
xmlschema  'http://www.anewlink.nl/xdb/CursusSchema.xsd'
element    'Aanmelding'
/
```

Door een kolom van het datatype XMLType te definiëren, kan hierin een compleet XML document worden vastgelegd. In dit geval is de structuur van dat XML document beperkt tot de structuur van het element 'Aanmelding' in de XML Schema definitie 'CursusSchema.xsd'.

Met betrekking tot het datatype XMLType valt overigens wel het een en ander toe te lichten. Zoals reeds eerder is aangegeven, kan het XMLType geïmplementeerd worden als een binair datatype met een textindex (CLOB) en vanaf 9iR2 ook als een object-relatieve datatype. Hoewel de fysieke implementatie binnen Oracle dus erg verschillend kan zijn, is dit transparant voor de applicatie. Het benaderen van XML in een kolom van het type XMLType kan voor beide implementatievarianten op dezelfde manier geschieden. Zodoende kan bijvoorbeeld een applicatie die is geschreven voor Oracle9iR1 zonder aanpassing blijven draaien als de toegepaste variant van XMLType gewijzigd wordt in de object-relatieve implementatie van 9iR2. De optimizer van Oracle herschrijft indien nodig de statements tot object-relatieve syntax.

## XPath en XQuery

Nu Oracle9i met XML DB ook de XML wereld 'naar binnen haalt', dienen voor het opvragen, navigeren en bewerken van XML data uiteraard ook standards ondersteund te worden

die in de XML wereld zijn ontstaan. De belangrijkste hiervan is XPath (XML Path Language), gestandaardiseerd door het W3C. Voor geïnteresseerden is hier meer over terug te vinden op hun officiële website<sup>2</sup>. XPath is een taal die een manier beschrijft om onderwerpen in XML documenten te lokaliseren en bewerken. Hierbij wordt een adresseringssyntax gebruikt die is gebaseerd op een pad door de logische structuur of hiërarchie van een document. Voor querydoeleinden op XML data implementeert Oracle XPath in twee opvraagmethoden: SQL/XML en XQuery. SQL/XML is een uitbreiding van SQL, waarbij XPath notaties gebruikt kunnen worden om als het ware 'in een XML document te stappen'. Met SQL/XML heeft Oracle als eerste leverancier een implementatie gereed van de SQL/XML standaard van de SQLX group. Een voorbeeld van een SQL/XML statement luidt als volgt:

```
select      xml_aanmeldingen
from        aanmeldingen
where       existsnode(xml_aanmeldingen
                        , '//Aanmelding/Cursuscode') > 0
/
```

Hierin scant EXISTSNODE het XMLType object om te zien of er een node is die matcht met de gespecificeerde XPath expressie. De grote kracht van SQL/XML is dat het de inhoud van XML documenten en data via de SQL abstractie laag benaderbaar maakt. Want hoewel de statements niet altijd even simpel zijn voor de beginnende ontwikkelaar blijft het wel SQL, met alle voordelen van dien. Zo kunnen er views worden gedefinieerd waar de SQL/XML statements in zijn verwerkt. Hierdoor kan bepaalde data zuiver relationeel via views worden gerepresenteerd, terwijl deze zich in feite diep verborgen in bepaalde XML documenten bevindt.

De andere toepassing van XPath die Oracle implementeert

**Oracle9iR2 maakt het mogelijk om een schema in te richten door het opgeven van een XML Schema definition**

heet XQuery. Omdat deze implementatie is gebaseerd op een W3C standaard die nog in draft is, is er hiervoor uitsluitend een in Java gebouwd prototype beschikbaar. Deze is uiteraard te downloaden van OTN. XQuery zal met name toegepast worden als XML DB uitsluitend via de XML abstractie laag

wordt benaderd. Dat wil zeggen, wanneer er geen behoefte bestaat om SQL gebaseerde enterprise tooling in te zetten.

## Voorbeelden

De in dit artikel gegeven informatie is grotendeels beschouwend van aard en dient vooral als context om XML DB als productfeature van Oracle9i te kunnen begrijpen. Het echte begrip van XML DB komt echter pas na bestudering van uitgebreide voorbeelden van de beschreven functionaliteit. Voor dit doel is de whitepaper van Mark Drake: 'XML in the Database: Storing XML with all your other critical data' uitermate

## *Het is mogelijk XML bewerkingen uit te voeren op gegevens in tabellen en SQL bewerkingen op XML documenten*

geschikt. Mark Drake is een medewerker van Oracle die vorig jaar op de Oracle Open World een uitgebreide presentatie heeft gegeven van de CLOB implementatie van XMLType in (toen nog) Oracle9iRI. De 29 pagina's tellende whitepaper bestaat vrijwel volledig uit voorbeelden en is voor de verandering vrij van elke vorm van marketing. Het whitepaper is terug te vinden op OTN, in de XML DB sectie<sup>2,3</sup>.

## Resumerend

In de historie van Oracle hebben we meermalen kunnen aanschouwen dat de marketingafdeling veel mist wist te creëren bij de annoncering van nieuwe producten. In de praktijk bleek dit dan vaak te gaan om een nieuwe naam voor een bestaand product of feature: oude wijn in nieuwe zakken dus. Wat betreft de functionaliteit van Oracle9i XML DB is dat echter geenszins het geval. Met het beschikbaar komen van Oracle 9iR2 hebben we de mogelijkheid gestructureerde en ongestructureerde data in het W3C XML model vast te leggen. XML DB biedt complete transparantie en uitwisselbaarheid van XML en SQL metaforen. Het is mogelijk XML bewerkingen uit te voeren op gegevens in tabellen en SQL bewerkingen op XML documenten. XML DB biedt beter beheer van ongestructureerde XML data door updatemogelijkheden op fragmenten van documenten, door indexerend, zoeken en meervoudige views op de data. Hierdoor kunnen we met recht spreken van een hybride database. Daarnaast biedt XML DB de functionaliteit van een repository voor XML resources, compleet met mogelijkheden voor foldering, versioning en autorisatie. Deze

repository is benaderbaar via het WebDAV protocol, dat ondersteund wordt door alle moderne besturingssystemen en Content Management omgevingen.

Bedrijven die reeds XML technologie toepassen of overwegen dat te gaan doen, dienen dan ook zeker het fenomeen XML DB nader te bestuderen.

## Noten

1 Sites van de 'gevestigde orde' native XML database leveranciers:

<http://www.exln.com> , <http://www.excelon.nl>

<http://www.softwareag.com> , <http://www.softwareag.nl>

<http://www.x-hive.com> , <http://www.x-hive.nl>

2 Meer informatie over genoemde onderwerpen is terug te vinden op:

<http://www.webdav.org>

<http://www.w3.org/TR/xpath>

<http://otn.oracle.com/tech/xml/content.html>

<http://otn.oracle.com/tech/xml/xmlldb/content.html>

3 En als aanrader voor voorbeelden en uitleg van XMLType in 9iRI de paper van Mark Drake:

<http://otn.oracle.com/tech/xml/pdf/Oracle9iXMLI.pdf>

## Harold Gerritsen

is senior consultant bij A New Link bv. Hij heeft meer dan twaalf jaar ervaring in het adviseren over effectief inzetten van Oracle technologie: <http://www.anewlink.nl>.

E-mail: [h.gerritsen@anewlink.nl](mailto:h.gerritsen@anewlink.nl)