Nieuwe mogelijkheden via Oracle XML DB

Functionaliteit steeds meer gebruikt

Oracle's XML DB functionaliteit wordt meer en meer gebruikt in de op de Oracle database gebaseerde systemen. Met de komst van de laatste database patch versies voor de Oracle 10gR2 en 11gR1 versies, is deze functionaliteit, ook meer dan een reële optie om databaseapplicaties te voorzien van extra mogelijkheden. Vooral ten aanzien van ongestructureerde data-uitwisseling met de buitenwereld is de inzet van XML zeer geschikt. Data-uitwisseling is één van de belangrijkste fundamenten voor het gebruik van XML, ooit eens één van de uitgangspunten voor de inzet van XML, een (zo niet het) transportmedium voor het transport van data.

XML is niet zaligmakend. Ondanks dat XML door alle grote (software) fabrikanten is omarmd en er veel gedaan is om standaarden met elkaar af te spreken, is er nog steeds onenigheid en discussie ten aanzien van bijvoorbeeld XML Schema formaten, zoals die tussen Microsoft Office en Open Office. Het is slechts een klein voorbeeld, maar vanwege het achterblijven van dit soort belangrijke afspraken, hoe nu om te gaan met het hoge 'Free Format' karakter van XML, blijven belangrijke doorbraken zoals een databasenormalisatie (XML data) afspraak, zoals we die kennen uit de relationele database wereld, uit en is iedereen naarstig opzoek naar de ***oplossing*** voor de hieruit volgende performance problemen die gepaard kunnen gaan met het omgaan met XML data.

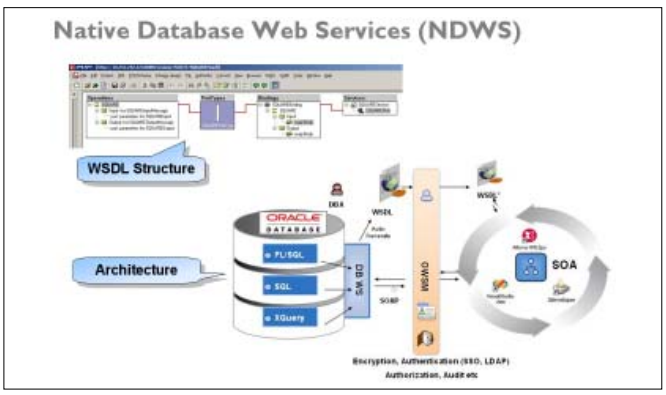
Positionering en mogelijkheden

Oracle is aan de behoefte van zijn klanten tegemoet gekomen om ook XML functionaliteit in zijn database product in te bouwen. Ooit eens begonnen als een XML Development Kit (XDK, Oracle 8i), XML tooling voor zowel in als buiten de database, maar hierna al snel ingebouwd als core onderdeel van de database en sinds Oracle database versie 9.2.0.3 officieel ondersteunt als 'Oracle XML DB'. De doorontwikkeling hiervan is snel gegaan, waarbij Oracle er naar gestreefd heeft om bijna elke W3C standaard voor ondersteuning van XML en/of die gebruikt wordt in de XML wereld, in de database te implementeren.

Een van de voordelen die de inzet van de Oracle heeft als XML database, ten opzichte van bijvoorbeeld een native XML database, is dat bij generieke XML (performance) problemen in de omgang met ongestructureerd, semi-gestructureerd of gestructureerde XML data, er altijd een alternatieve oplossingsrichting gekozen kan worden gebaseerd op de technologie die in de database al voorhanden is zoals: Java, PL/SQL, .Net, relationele of object georiënteerde structuren, partitionering van opslagstructuren, inzet van Oracle Text, Media of Spacial functionaliteit.

Die symbiose is wederzijds. Oracle XML DB heeft veel core functionaliteit aan de Oracle database toegevoegd, die ook heel goed inzetbaar is voor andere (relationele) doeleinden, waarbij onder andere maar niet alleen, XQuery, beveiliging op basis van Access Control Lists (ACL's), een Repository, toegang tot de database via HTTP, FTP en WebDAV of ondersteuning bij het creëren van web services via de in Oracle 11g ingebouwde Native Database Web Services (ondersteuning voor SOAP en WSDL).

Een goed voorbeeld van die symbiose is bijvoorbeeld ook de lokale Oracle APEX implementatie, een ontwikkel framework gebaseerd op klassieke relationele databasemethodieken en de XML DB Protocol Server Local Gateway ondersteuning, die zijn intrede heeft gemaakt bij de introductie van Oracle 11gR1. Hierdoor is een aparte webserver ten behoeve van de inzet van APEX niet meer per se noodzakelijk.

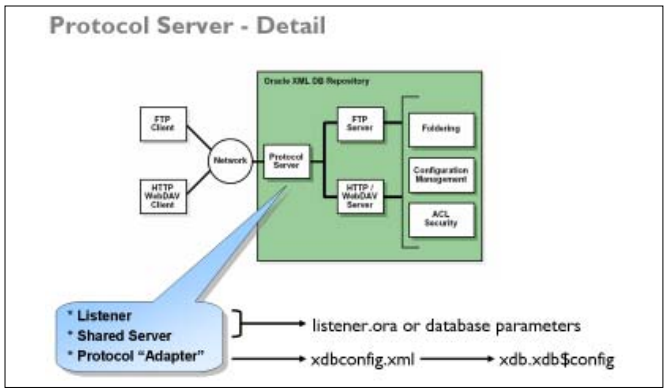


Data in de database kan via de Oracle 11g Native Database Web Services (NDWS) functionaliteit naar de buitenwereld worden aangeboden als een webservice en, hoewel niet noodzakelijk, worden beveiligd en onderhouden door de inzet van Oracle's Web Service Manager (onderdeel van de Oracle SOA Suite). Deze door Oracle meegeleverde NDWS functionaliteit maakt ook de weg vrij voor de inzet van de database als een SOA eindpunt of als onderdeel van een SOA architectuur op basis van SOAP.

De XMLDB Protocol Server

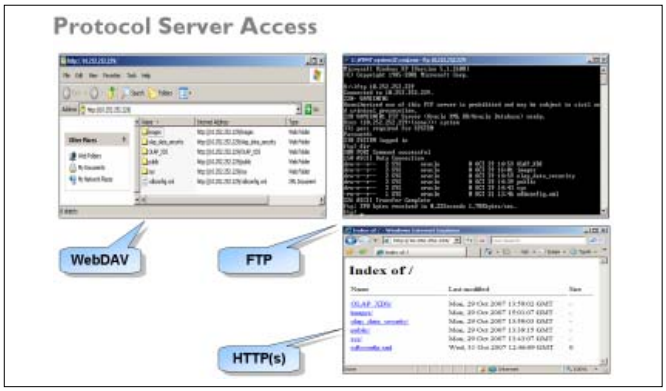
De Protocol Server functionaliteit is geïmplementeerd door het Oracle XML DB Development team, als onderdeel van Oracle XML DB, vanaf Oracle database versie 9.2.0.3. Het haakt in op de Oracle Shared Server, de vroeger geheten Multi Threaded Server, SQL*Net architectuur. Deze al in het Oracle 7 geïntroduceerde SQL*Net ondersteunende architectuur was ontwikkeld om honderden connecties tegelijkertijd af te kunnen handelen en het destijds dure geheugengebruik binnen de perken te houden.

De Protocol Server maakt het mogelijk om database objecten via FTP, HTTP(s) en WebDAV te manipuleren, maar ook om wat Oracle 'Foldering' noemt, directories en files (zogenaamde 'resources'), mogelijk te maken. Een van de vele XML-standaarden die Oracle heeft geïmplementeerd om een volwaardige XML-database omgeving te kunnen neerzetten.



De hierdoor ontsloten Repository omgeving voldoet onder andere aan deze 'Foldering', methodieken die de configuratie management ondersteunen, de beveiliging van objecten zijn via ACL's gewaarborgd en men heeft ook de mogelijkheid hieraan databaserollen te koppelen. Ondersteuning van versiebeheer en locking mechanismen en bijvoorbeeld check-in en check-out functionaliteit van files zijn ook in de database geïmplementeerd.

Out of the box voldoet deze XML DB Repository functionaliteit al aan een hoop lichtgewicht Content Management vereisten.



In de 11g database heeft Oracle ook Repository Events geïmplementeerd. Deze Repository Events zijn vergelijkbaar met database triggers, waar acties mee getriggerd kunnen worden, nadat bijvoorbeeld een file in de folderstructuur is gekopieerd. Zo is het mogelijk om, nadat een XML file in een directory is aangepast of gekopieerd (het event), de zich hierin bevindende data, volautomatisch (programeer technisch via PL/SQL of Java) te verwerken en hierna deze data bijvoorbeeld in een relationele tabel te zetten.

Het gedrag van de Protocol Server functionaliteit kan worden geregeld via een XML-configuratiefile met de naam **xdbconfig.xml**. Deze configuratiefile is te bewerken via WebDAV of via update statements vanuit de database. Deze statements maken tegenwoordig ook deel uit van database Packages, wat het beheer nog verder vergemakkelijkt. De Protocol Server configuratiefile xdbconfig.xml voldoet aan de vereisten vastgelegd in een XML Schema met de naam **xdbconfig.xsd** en regelt het gedrag dat je zou verwachten bij een webserver.

Onder andere is in de xdbconfig.xml file de ondersteuning van MIME-types beschreven. Het gedrag dat invloed heeft op locking, timeouts van dataverkeer via http of WebDAV, ondersteuning van de verschillende encoding en karakter sets en de aansturing van extra applicatie functionaliteit, die door Oracle 'Servlets' worden genoemd. Deze extra functionaliteit is echter lang niet altijd gebaseerd op Java zoals de naam 'Servlets'

zou doen vermoeden, maar kan ook gebaseerd zijn op de aansturing van PL/SQL code in de database of op C gebaseerde, in de database verankerde functionaliteit. Het is mogelijk om eigen Servlets te creëren en deze via de Protocol Server functionaliteit te ontsluiten voor de buitenwereld.

Vanuit de database is het mogelijk om alle structuren in de Repository te manipuleren. Dit zou kunnen betekenen dat er via eigen PL/SQL code files op en neer worden gestuurd, files worden gefilterd of gemanipuleerd op inhoud, naar behoefte files and directories worden aangemaakt en de inhoud gevuld wordt met de data uit tabellen. Een van de weinige beperkingen ten aanzien van files en directories op dit moment is nog wel dat een file in de Repository een maximale grootte kan hebben van 4 Gigabyte.

Enable and Status - The Protocol Server

```
SQL> select dbms_xdb.getHttpPort() from dual;
SQL> select dbms_xdb.getFtpPort() from dual;

SQL> call dbms_xdb.setHttpPort(8080);
SQL> call dbms_xdb.setFtpPort(2100);

SQL> alter system set shared_servers=5;
SQL> alter system register;
```

De Protocol Server functionaliteit kan simpel aan en uit worden gezet sinds database versie 10gR2, door middel van een DBMS_XDB.SETHTTPPORT package aanroep. Het zetten van een poort hoger dan de waarde nul (0) zal ervoor zorgen dat de HTTP/WebDAV functionaliteit zich automatische registreert tegen de Oracle Listener, zodat er vanaf dat moment HTTP verkeer kan plaatsvinden. Via een aanroep van DBMS_XDB.GETHTTPPORT package kan de poort worden gecontroleerd waarop de Protocol Server is geactiveerd.

Servlets in de XML DB HTTP Server

Zoals al vermeld ondersteunt de Protocol Server een aantal Servlets 'Out of the Box'. Als de APEX ontwikkelomgeving wordt geïnstalleerd, wordt er een functionaliteit geactiveerd die ingrijpt op de database via de zogenaamde PL/SQL Gateway, gebaseerd op de DBMS_EPG PL/SQL Package. De attributen die benoemd kunnen worden in de xdbconfig.xml configuratiefile van de Protocol Server, worden hierna doorgegeven aan de DBMS_EPG PL/SQL package in de database. Dataverkeer via HTTP met de database is vanaf dat moment mogelijk.

Het is ook denkbaar om een eigen webapplicatie te schrijven met HTML files in de Repository of gegenereerd uit de database, waarbij de database data via de deze functionaliteit wordt ontsloten.

Protocol Server HTTP API's

- Your Own…
  - Java Servlets, PL/SQL Gateway
- PL/SQL Gateways
  - APEX (/**apex**/*) PL/SQL
- "Servlets"
  - Native Database Web Service (/**orawsv**/*) C
  - *ReportFmwkServlet* (/**orarep**/*) C
  - TestServlet (/**Test**) Java
  - DBURI (/**oradb**/*) C

De in de Oracle 11g database geïntroduceerde Native Database Web Service is verankerd in de kernel van de database en roept via de Protocol Server functionaliteit rechtstreeks interne database code aan. Het aanbieden van data via een SOAP webservice hierna is erg gemakkelijk te implementeren.

Native Database WebService (Procedure)

1. Enable the Protocol Server for HTTP access
2. Enable the **orawsv** entry point in xdbconfig.xml
3. Create a user and grant access
   - DBMS_NETWORK_ACL_ADMIN
   - XDB_WEBSERVICES
   - XDB_WEBSERVICES_OVER_HTTP
   - XDB_WEBSERVICES_WITH_PUBLIC
4. Create your procedure
   - For instance a procedure called OBJECTTYPES
5. Test
   - http://localhost:8080/orawsv/SCOTT/OBJECTTYPES?wsdl

De Protocol Server moet worden geactiveerd zoals hiervoor beschreven door middel van een poort adressering via de DBMS_XDB.SETHTTPPORT package of een rechtstreekse aanpassing in de xdbconfig.xml file. De Native Database Web Service (NDWS) kan worden geactiveerd door het toevoegen van een aantal elementen in de xdbconfig.xml configuratiefile. Door een database user hierna specifieke XDB_WEBSERVICES databaserollen te geven kan deze via een eigen gemaakte procedure, die bijvoorbeeld data ophaalt uit een tabel, deze data publiceren via een webservice.

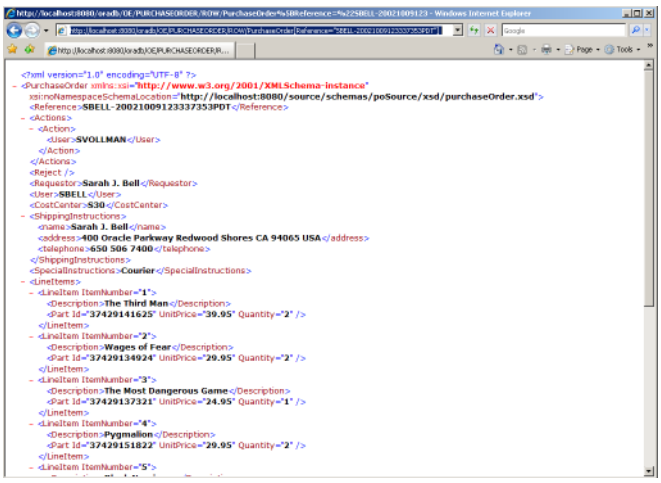
Oracle 11gR1 is strenger geworden ten aanzien van beveiliging van alle procedures en methodieken die iets rechtstreeks buiten de database kunnen doen zoals de packages: UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP en UTL_INADDR. Deze worden vanaf database versie 11g via Access Control List's (ACL's) beveiligd tegen ongeautoriseerd gebruik. De DBMS_NETWORK_ACL_ADMIN database package kan gebruikt worden om deze beveiliging te beheren, waarna de database ervoor zorgt dat er bij ongeautoriseerd gebruik een ORA-24247 foutmelding wordt gegenereerd, **'Network access denied by access control list (ACL)'**, als de privileges niet in orde zijn.

De laatste servlet die hier aan bod komt, de DBURI servlet, die ook via de Protocol Server standaard wordt aangeboden via HTTP verkeer, grijpt in op de database functionaliteit die ook bekend staat als de 'URI Factory'. Deze 'URI Factory' functionaliteit kent drie default URITypes. De DBURI servlet is gebaseerd op van die drie URITypes: de DBUriType.

Deze DBURI servlet zorgt ervoor dat men via het intikken van een URL, in een web browser zoals Internet Explorer, rechtstreeks in XML formaat, data uit de database kan worden opgevraagd. Eventueel kan deze XML-data verder worden getransformeerd door gebruik te maken van XPath statement toevoegingen in de URL en/of getransformeerd via XSL. De volgende URL, via de DBUri ingang /**oradb**/, vraagt **een** rij op uit de PurchaseOrder table, van de database user OE, waarbij het Reference element gelijk is aan 'SBELL-20021009123337353PDT'.

```
http://localhost:8080/oradb/OE/PURCHASEORDER/ROW/
PurchaseOrder[Reference="SBELL-20021009123337353PDT"]
```

Het antwoord op die vraag uit de PurchaseOrder tabel wordt hieronder getoond.



UriTypes

De Oracle database kent al langere tijd, sinds Oracle 9i, deze zogenaamde UriType datatypes/referenties . Er zijn default drie UriTypes met ieder hun eigen karakteristieke eigenschappen:

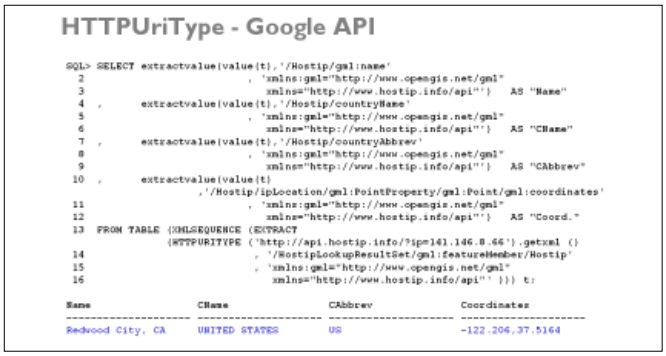
- **De DBUriType.** Deze vertegenwoordigt een Unified Resource Indicator (URI) waarmee je database data kunt aanroepen, zoals in tabellen of views.

- De **HTTPUriType.** Deze vertegenwoordigt een Unified Resource Locator (URL), waarbij je door middel van een http:// aanroep, buiten de database, data kunt aanroepen en is gebaseerd op de database package UTL_HTTP.

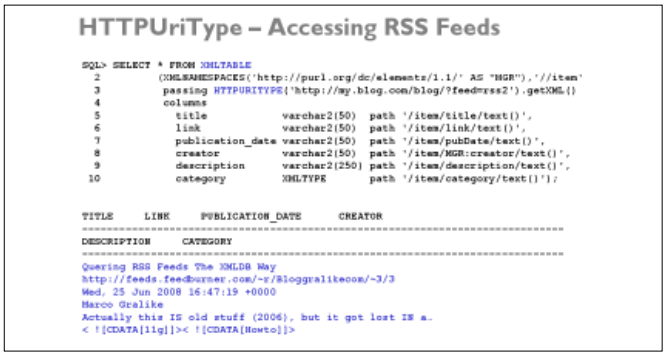
- De **XDBUriType.** Deze vertegenwoordigt Unified Resource Indicator (URI), waarbij je de files en folders in de XMLDB Repository kunt aanroepen.

Door 'XMLType' datatypes (Oracle's datatype voor XML) en XML DB functies of operatoren te combineren met de al in de database aanwezige UriTypes, kan er bijvoorbeeld data rechtstreeks opgehaald worden, zoals het volgende voorbeeld laat zien. In het voorbeeld wordt een webservice op het internet uitgevraagd om te achterhalen wat de longitude and latitude coördinaten zijn van het TCP/IP nummer, behorende bij een aangemelde webbrowser. Deze informatie kan worden gebruikt ten behoeve van een Google Map API, die de coördinaten als een plek op een geografische kaart zichtbaar kan maken.

De volgende twee voorbeelden zijn gebaseerd op de HTTPUriType.



Tegenwoordig is veel data op het internet gebaseerd op XML-principes of in XML-formaat voorradig. Denk hierbij aan webpagina's die voldoen aan XHTML en/of Web Blog 'feed' methodieken zoals RSS. Via het gebruik van een HTTPUriType in combinatie met XML DB functionaliteit kan vrij eenvoudig een RSS feed van een weblog worden uitgevraagd zoals het volgende voorbeeld aantoont.



De hierboven beschreven methodieken zien er indrukwekkender uit dan ze zijn.

Beide statements vragen data op via een website URL die XML data teruggeeft. Aan de hand van hun bijbehorende XML Schema's (RSS en GIS XML Schema's) wordt het teruggegeven XML Document getoetst op het juiste formaat en hierna, via XML operatoren zoals XMLTABLE en XPath selecties, verder ontleed naar een relationeel formaat dat naar behoefte kan worden verwerkt via DML statements.

De XDBUriType kan men gebruiken om XML DB Repository resources, files en folders, aan te roepen in de XML DB Repository. In de praktijk wordt de XDBUriType meestal gebruikt voor het opvragen van de inhoud in XML files of XML Schema's die zich in de XML DB Repository bevinden, om XML files te creëren in de XML DB Repository of om XML Schema's te registreren in de XML DB Repository.

Het volgende voorbeeld is gebaseerd op een XDBUriType aanroep en is een creatievere manier om met de XML DB Repository als opslagmedium om te gaan. Het voorbeeld laat zien hoe men een file aan kan maken in de XML DB Repository waarbij de inhoud bestaat uit een XQuery string. De XQuery string inhoud in de 'q1.xqy' file wordt hierna gebruikt in een daadwerkelijke XQuery vraag die wordt toegepast op een XMLType tabel (MY_XML_TABLE) die XML documenten bevat.

Storing your Scripts in the Repository

```
declare
  res BOOLEAN;
begin
  res := dbms_xdb.createResource
                          ('/public/q1.xqy',<My Xquery>);
  commit;
end;
/

SELECT xmlquery(xdburitype('/public/q1.xqy').getClob()
       passing OBJECT_VALUE
       returning content)
FROM   MY_XML_TABLE
/
```

External Tables via XML

Het schrijven van data buiten de database, en/of het lezen van deze data buiten de database, heeft meestal nog aardig wat programeer- of configuratiewerk in de Oracle database tot gevolg. De meest gebruikte manieren gaan uit van het gebruik van de UTL_FILE database PL/SQL package en/of configuratie van Oracle SQL*Loader methodieken om 'External Table' databasetabellen op te zetten. Alternatieven via Java-code in de database java kernel behoren ook tot de mogelijkheid om dataverkeer naar de buitenwereld mogelijk te maken.

Het is vrij eenvoudig om XML-files op een harde schijf, op de machine waar de database zich bevind, rechtstreeks uit te vragen of om XML files te creëren buiten de database. Een XML file kan gezien worden als een 'mini' tabel en deze kan veel 'rijen' bevatten. Als je hierover nadenkt, zou het in sommige gevallen een interessante manier kunnen zijn om data rechtstreeks uit een XML file te benaderen en of op een filesysteem buiten de database weg te schrijven. Het volgende voorbeeld laat zien hoe.

Selecting XML Data outside the Database

```
SQL> CREATE directory XMLSTORE as 'C:\TEMP';


SQL> SELECT extract((XMLTYPE(bfilename('XMLSTORE','data.xml')
                                      ,NLS_CHARSET_ID('AL32UTF8'))),'*') AS "XDATA"
  2  FROM dual;

XDATA
------------------------------------------------------------
<root>
  <id>0</id>
  <info>
    <info_id>0</info_id>
    <info_content>Text</info_content>
  </info>
</root>
```

Het eerste statement creëert een 'directory' alias met de naam XMLSTORE, die verwijst naar een directory buiten de database, bijvoorbeeld de directory 'C:\TEMP'. In deze directory buiten de database bevindt zich een XML file met de naam 'data.xml'. De inhoud van deze file wordt via een **bfilename** constructie, opgevraagd van schijf en hierna doorgegeven aan een XMLTYPE operator die er een XMLType datatype van maakt. Door hierna een EXTRACT XML operator te gebruiken, kan de XML inhoud verder naar wens worden uitgevraagd en/of bewerkt.

De XML DB database functionaliteit kent twee PL/SQL database packages, DBMS_XMLDOM en DBMS_XSLPROCESSOR, om XML te bewerken, XML documenten te creëren of deze XML documenten naar schijf te schrijven. Ook hier wordt weer vaak gebruik gemaakt van een database 'directory' alias die verwijst naar een locatie buiten de database.

- De DBMS_XMLDOM package wordt gebruikt voor XMLType objecten en maakt gebruik van het Document Object Model (DOM), een applicatie programmeer interface voor HTML en XML documenten.

- De DBMS_XSLPROCESSOR package vertegenwoordigt een interface die de inhoud en structuur van XML documenten beheersbaar maakt.

Saving XML Data outside the Database

```
SQL> DECLARE
2       rc  sys_refcursor;
3    BEGIN
4       OPEN rc FOR
          SELECT * FROM
          (SELECT rownum FROM dual connect BY level < 100);
5    dbms_xslprocessor.clob2file( xmltype(rc).getclobval( ),'XMLSTORE', 'data.xml');
6    END;
7 /



SQL> DECLARE
2       rc  sys_refcursor;
3       doc DBMS_XMLDOM.DOMDocument;
4    BEGIN
5       OPEN rc FOR
          SELECT * FROM
          (select rownum from dual connect by level < 100);
6       doc := DBMS_XMLDOM.NewDOMDocument(xmltype( rc ));
7       DBMS_XMLDOM.WRITETOFILE(doc, 'XMLSTORE/data.xml');
8    END;
9 /
```

Het bovenstaande voorbeeld laat een methodiek zien om gegenereerde inhoud op basis van een SQL statement in een XML document file weg te schrijven.

Nieuwe Mogelijkheden via XML Functies en Operatoren

Vanaf database versie 10.2 heeft ook voor XML-data belangrijke XQuery taal zijn intrede gedaan in de Oracle databasewereld. De XML DB database kent een grote hoeveelheid aan nieuwe functies, operatoren en nieuwe query of manipulatie talen zoals XPath, XSL en de genoemde XQuery taal. Deze nieuwe mogelijkheden zijn goed toepasbaar om gebruikt te worden samen met SQL of om programeertechnisch, binnen de relationele database kaders, 'onmogelijkheden' te realiseren of te vereenvoudigen.

De Oracle 10gR2 en 11gR1 database versies kennen over de twintig functies en operatoren, naast allerlei PL/SQL packages, om XML data te manipuleren en of te creëren. Hieronder volgt een klein overzicht van de beschikbare functies en operatoren in de twee genoemde database versies.

XML Operators & Functions in 11gR1

DELETEXML	XMLAGG	XMLPARSE
EXTRACTVALUE	XMLCAST	XMLPATCH
EXISTNODE	XMLCDATA	XMLPI
EXTRACT	XMLCOLATTVAL	XMLQUERY
INSERTCHILDXML	XMLCOMMENT	XMLROOT
INSERTXMLBEFORE	XMLCONCAT	XMLSEQUENCE
SYS_XMLAGG	XMLDIFF	XMLSERIALIZE
SYS_XMLGEN	XMLELEMENT	XMLTABLE
UPDATEXML	XMLEXISTS	XMLTRANSFORM
…	XMLFOREST	…

Een meester in het creatieve gebruik van XML binnen de SQL wereld is Laurent Schneider, een Oracle Certified Master uit Zwitserland. Op zijn website (http://laurentschneider.com/) en in zijn boek 'Advanced Oracle SQL Programming' geeft hij tal van voorbeelden, waaronder een pivot bewerking in een Oracle 10gR2 database. De Oracle officiële database SQL Pivot functie is pas beschikbaar vanaf de Oracle 11g database.

Het hier getoonde XML/SQL voorbeeld laat de kracht zien van het samenspel tussen XML functionaliteit en SQL en geeft een overzicht van het aantal rijen per tabel.

Using XML Operators with SQL

```
SQL> SELECT
  2    table_name,
  3    to_number(
  4      extractvalue(
  5        xmltype(dbms_xmlgen.getxml
  6                ('select count(*) C from '||table_name))
  7                ,'/ROWSET/ROW/C')) count
  8  FROM user_tables;

TABLE_NAME                     COUNT
------------------------------ ------
DEPT                                4
EMP                                14
BONUS                               0
SALGRADE                            5
```

Source **Laurent Schneider**: How do i store the counts of all tables ...

Oracle XML Database

De gratis meegeleverde Oracle XML DB functionaliteit wordt door Oracle steeds meer als core database functionaliteit ingezet. Vooral ten aanzien van datatransport beveiliging via Access Control Lists, zijn de eerste tekenen hiervan duidelijk in de Oracle 11g database release zichtbaar. De basisfunctionaliteit en concepten die in de 'Oracle XML DB Developers Guide' manual aan bod komen, bestrijken sinds de intrede in de Oracle 9.2.0.5 database versie, nu al meer dan negenhonderd pagina's. Een van de signalen dat er veel aandacht en energie in het product is gestoken.

De meeste in dit artikel besproken voorbeelden zijn ook goed toepasbaar in een Oracle 10gR2 database. Ze laten een tipje van de sluier zien wat er mogelijk is met een beetje creativiteit en de inzetbaarheid van XML DB functionaliteit met betrekking tot de bestaande relationele database methodieken.

De onderwerpen zij slechts enkele voorbeelden die in de 'Boost Your Environment with XMLDB' presentatie aan bod kwamen tijdens de UKOUG 2008 Conferentie in Birmingham en zijn geënt op een Oracle 11gR1 database. Hopelijk zetten zij aan tot nieuwe inspiratie en het gebruik van XML DB functionaliteit in een bestaande Oracle database omgeving.

Referentie

http://technology.amis.nl/blog/4204/ukoug-2008-boost-your-environment-with-xmldb

Marco Gralike is werkzaam bij AMIS Services BV in Nieuwegein als Senior Oracle Database beheerder met o.a. 15 jaar ervaring op het gebied van de Oracle database.

4 en 5 maart 2009 • Hotel Lapershoek Hilversum

Pragmatisch modelleren met UML

met Sander Hoogendoorn

- U leert het gebruik van use cases
- U krijgt inzicht in de samenhang tussen de verschillende modelleertechnieken
- U gaat zelf aan de slag met UML in een pragmatische case
- Zeer interactieve workshop met maximaal 20 deelnemers
- Voor iedere deelnemer gratis het boek ‘Pragmatisch modelleren met UML 2.0’
- Deelnemers waardeerden de vorige editie met een 7,9!

De modelleertaal UML is uitgegroeid tot de wereldwijde standaard voor het modelleren van requirements, functionaliteit, componenten en services. UML 2.0 kent een groot aantal modelleertechnieken zoals use case diagrammen, activity diagrammen, sequence diagrammen, communication diagrammen, class diagrammen en package diagrammen.

Tijdens de tweedaagse workshop leert u niet alleen een aantal essentiële vaardigheden voor het modelleren met UML (en andere modelleertechnieken), maar u zet tegelijk zo’n pragmatische werkwijze neer. In deze workshop met veel aandacht voor de praktijk kunnen de deelnemers met diverse oefeningen de verschillende gezichtspunten van een prikkelende case uitwerken – een online dating website. Daarbij wordt veel aandacht besteed aan de nauwe relatie tussen de verschillende modelleertechnieken en het gewenste detailniveau van de diagrammen. Bovendien wordt de link gelegd naar de ontwikkeling en het testen van de software op basis van de geproduceerde diagrammen en (servicegeoriënteerde) applicatiearchitecturen.

Kortom: in deze praktische en interactieve workshop wordt UML in al zijn facetten en toepassingsmogelijkheden behandeld.

KORTING!

Profiteer van vroegboekvoordeel en korting bij meerdere deelnemers van één bedrijf!

DATUM	**4 en 5 maart 2009**
LOCATIE	**Hotel Lapershoek Hilversum**
TIJD	**Van 9.30 uur tot 17.00 uur**
REGISTRATIE	**www.arrayseminars.nl**

Kijk snel op www.arrayseminars.nl voor het complete programma!

Array SEMINARS

in samenwerking met Computable