XML en relationeel vloeien in elkaar over

Native XML in DB2 Viper

Klaas Brant

DB2 Versie 9 is de eerste versie van DB2 die het mogelijk maakt om native XML in DB2 te gebruiken. Op de platformen Linux, Unix en Windows (LUW) is versie 9 reeds enige maanden beschikbaar, op het mainframe zal deze functionaliteit binnenkort beschikbaar komen. Voor de duidelijkheid: DB2 for z/OS (mainframe) en DB2 LUW zijn verschillende producten met verschillende implementaties en mogelijkheden.
Toch zal de XML implementatie van DB2 for z/OS sterk lijken op die van LUW. Dit artikel beschrijft de mogelijkheden van DB2 LUW.

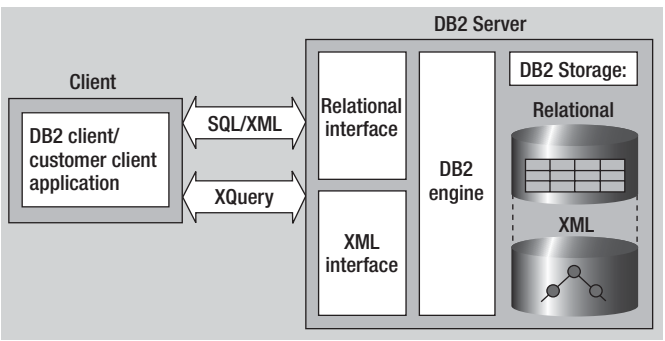
DB2 Versie 9 kreeg bij de ontwikkeling de naam 'Viper'. Viper betekende een enorme ingreep in DB2. Van huis uit is DB2 een relationele database met SQL als querytaal. Naast de relationele data kan DB nu ook XML data opslaan. XML is van origine hiërarchisch en heeft geen vaste structuur. Overigens is in sommige gevallen het ongestructureerd zijn van data niet te vermijden (denk bijvoorbeeld maar eens aan de Windows-registry). XML voegt een nieuwe querytaal, een nieuwe opslagtechnologie en een nieuwe indextechnologie toe aan DB2. IBM noemt Viper dan ook een *hybrid database*. Beide technologieën leven in dezelfde database naast elkaar. Het bijzondere is dat relationeel en XML in elkaar overvloeien.

Architectuur van Viper

XML is vaak een file met daarin de XML data. Bij de verwerking van XML moet een keuze gemaakt worden: of het XML document gaat in de database en wordt opgeslagen in een kolom van een tabel (bijvoorbeeld een BLOB). Of het XML document wordt afgebroken (parsing met DOM of SAX) en tijdens dit proces worden data uit het XML document in een relationele database opgeslagen. In de vorige releases van DB2 was er een extender van DB2 die beide varianten aanbood. Deze extender had vele beperkingen.
Ondanks dat XML totaal anders van structuur is dan relationele data was het uitgangspunt dat de XML en relationele data naadloos in elkaar over moeten kunnen gaan. Zo kunnen relationele en XML data gecombineerd worden in een enkele SQL query.

> Het bijzondere is dat relationeel en XML in elkaar overvloeien

Viper kent een nieuwe opslagmethode voor XML data met een nieuwe soort XML index. De client interface van DB2 is aangepast en nu kunnen applicaties zowel relationele data als XML hiërarchische documenten benaderen. De applicatie heeft hierbij de keuze tussen SQL (uitgebreid met iso-XML extensies, nu vaak SQL/XML genoemd) of XQuery. Intern in DB2 komen beide interfaces uit in dezelfde engine, zie afbeelding 1.



Afbeelding 1: Integratie van XML en relationeel in DB2.

Logical Storage

Intern in DB2 worden XML documenten opgeslagen in relationele columns. Je zou haast zeggen: soortgelijk als bij de extender, maar het grote verschil is dat de column geen BLOB is maar een nieuw datatype. In de vorige release was dit nieuwe XML datatype reeds aanwezig als de SQL XML extensies gebruikt werden om vanuit relationele data XML documenten op te bouwen. Was het XML datatype in de vorige release alleen een *runtime* datatype dat via functies tot *character data* omgezet kon worden, nu is XML een datatype dat opgeslagen kan worden (persistent) en zijn er query-faciliteiten om in deze persistente data te zoeken. Omdat XML

een SQL datatype is, gaat het aanmaken van XML structuren in de database dan ook met normale SQL DDL. Bijvoorbeeld:

```
CREATE TABLE ITEMS (
        ID              INT     PRIMARY KEY NOT NULL,
        BRANDNAME       VARCHAR(30),
        ITEMNAME        VARCHAR(30),
        SKU             INT,
        SRP             DECIMAL(7,2),
        COMMENTS        XML
)
```

De interne opslag van XML is verre van relationeel maar geheel transparant voor de gebruiker. De enige zorg van de gebruiker is de vraag of de opgeslagen XML wel conform een schema is.
Het datatype XML kent in de DDL geen verdere parameters voor storage of variaties. Validatie van XML is dus geen DDL-optie maar wordt op een andere manier geregeld.

Physical Storage

De data worden niet één-op-één opgeslagen, maar door de DB2 parser ingelezen en intern opgeslagen in een hiërarchische techniek die een afspiegeling is van de XML *node tree*. Het is tijdens het parsen dat DB2 eventueel een schemavalidatie kan doen (niet verplicht). Door de speciale opslag is het mogelijk dat DB2 door de XML navigeert zonder deze geheel in storage te hebben. Dit is belangrijk, omdat XML documenten erg groot kunnen worden. Als we deze werkwijze vergelijken met een applicatie, dan is de methode van DB2 een duidelijke verbetering. Applicaties moeten de XML data volledig inlezen (DOM) of geheel sequentieel doorlopen (SAX) om data in de nodes te vinden. DB2 kan door de nodes navigeren zonder deze in storage te halen. De Xquery optimizer kan dus zeer grote XML documenten verwerken. Een nadeel van het parsen en opslaan in intern formaat is, dat het document er qua formattering anders uit kan zien als het weer geformeerd moet worden om aan de gebruiker te laten zien.

> DB2 kan door de nodes navigeren zonder deze in storage te halen

DB2 ondersteunt dus niet de optie die vaak *preserve white space* wordt genoemd. In DB2 LUW behoord XML tot de long datatypes zoals long varchars; dat wil zeggen dat de data intern over meerdere datapages verspreid kunnen liggen. Net als bij relationele data is het wenselijk om indexen aan te leggen om een goede performance te krijgen. Bij het parsen worden de XML nodes genummerd. Deze nummering wordt gebruikt in de indexen en bij de navigatie. De gebruiker zal van deze nummering niets merken.

Query voorbeelden

Hier volgen enkele voorbeelden van query's op de mix van relationeel en XML.
XML en SQL data met SQL:

```
SELECT * FROM ITEMS WHERE SKU = 112233
```

DB2 geeft de XML column terug als XML document. Applicaties kunnen bijvoorbeeld DOM gebruiken om het XML document verder te processen.
XML data met Xquery (simple Xpath expressie):

```
XQUERY db2-fn:xmlcolumn('ITEMS.COMMENTS')/
                        Comments/Comment/Message
```

In de XML documenten die zijn opgeslagen in de items table kolom comments wordt gezocht naar nodes die voldoen aan de XPath expressie "/Comments/Comment/Message".
XML data met XQuery (FLWOR expressie):

```
XQUERY FOR $Y IN DB2-FN:XMLCOLUMN('ITEMS.
                    COMMENTS')/COMMENTS/COMMENT
RETURN ($Y/MESSAGE)
```

Dit is exact dezelfde query maar nu als FLWOR expressie. Hoewel dit een simpel voorbeeld is, zijn flower expressies vele malen krachtiger dan simpele Xpath expressies.
XML data met Xquery waarbinnen SQL expressie:

```
XQUERY DB2-FN:SQLQUERY('SELECT COMMENTS FROM
                                          ITEMS
WHERE SRP > 100')/COMMENTS/COMMENT/MESSAGE
```

In deze query wordt eerst de SQL query als een sort nested subquery uitgevoerd. De rijen die terugkomen uit de SQL Query gaan vervolgens de XPath selectie in.

XML Indexing

Soortgelijk als bij relationele data worden XML indexen aangemaakt met DDL. Bijvoorbeeld:

```
CREATE INDEX MYINDEX ON ITEMS(COMMENTS) GENERATE KEY
USING XMLPATTERN '/COMMENTS/COMMENT/COMMENTID'
                                        AS SQL DOUBLE
```

Worden er bij relationele indexen kolommen genoemd voor de index, bij XML is dit een Xpath expressie (zonder predicates). Op die manier kan een aantal nodes die aan de expressie voldoen uit de XML tree getild worden. Enige voorzichtigheid is geboden

omdat de Xquery expressie case sensitive is. Omdat DB2 geen idee heeft wat voor soort data in de nodes liggen opgeslagen, moet de gebruiker dit aangeven bij het aanmaken van de index. Is dit datatype in de nodes niet consistent, dan is indexing een probleem. Echter, in de praktijk zullen de meeste XML documenten well formed (conform schema dus) zijn en zal dit geen probleem moeten zijn.

> Er zijn geen gebieden waar de XML functionaliteit ontbreekt

Er zijn twee opmerkelijke verschillen met SQL indexen. Ten eerste is het aantal data types binnen de nodes van XML vele malen minder dan bij relationele data. Ten tweede zal bij relationele indexen één index entry naar één rij wijzen; bij XML indexen kunnen vele index entry's naar dezelfde rij wijzen, maar binnen het XML document van die rij maar naar één node in de XML tree. De nummering van nodes komt hier dus goed van pas, anders zou men de XML tree weer opnieuw moeten parsen.

XQuery interface

Naast de traditionele relationele SQL query-taal, kent DB2 nu ook XQuery om data uit XML documenten te halen. Zelfs een mix van beide query-talen kan gebruikt worden. Natuurlijk kunnen we in dit artikel niet een complete uiteenzetting geven van de mogelijkheden, maar enkele voorbeelden geven een aardig idee. XQuery is een taal die op twee manieren gebruikt kan worden: als een simpele Xpath expressie of met een zogenaamde FLWOR expressie. Om het wat gemakkelijker uit te spreken noemt men dit ook vaak een *flower* expressie. FLWOR komt van for, let, where, order by, return. Een van de grondleggers van FLWOR is Don Chamberlin, die ook aan de wieg stond van SQL. De flower expressies doen sterk aan SQL denken maar zijn toch net even anders. Omdat XQuery voor veel mensen nieuw is zal de *graphical query builder* een welkom tool zijn. Met dit reeds bestaande tool kan men op een simpele manier Xquery query's bouwen. Naast de de bekende SQL optimizer bevat DB2 nu ook een speciale optimizer voor Xquery, zie het kader op pagina 19.

XML schema's en validatie

In de praktijk zal de kwaliteit van data waarschijnlijk een aandachtspunt zijn dat hoog op de agenda staat. In veel gevallen zal de open structuur van XML zeer ongewenst zijn. Om er voor te zorgen dat de XML data aan bepaalde eisen voldoen, zijn de zogenaamde schema's uitgevonden. In de schema's staan de hiërarchie, de datatypes en diverse andere kenmerken (bijvoorbeeld komt een node 0,1 of meerdere malen voor en optionaliteit). Het is dan ook zeer wenselijk dat data gevalideerd worden met een schema, voor ze opgeslagen worden in de database. DB2 ondersteunt de schemavalidatie. De schema's worden opgeslagen in een interne repository. Een enkel schema kan gebruikt worden om meerdere XML kolomen in meerdere tabellen te valideren. Dit is natuurlijk zeer wenselijk, anders zou het kunnen gebeuren dat dezelfde XML structuur (bijvoorbeeld een order) meerdere malen wordt beschreven. Als we relationele databases opnieuw zouden bouwen, dan zouden we het wellicht ook mogelijk maken om een bepaalde table layout een naam te geven en die meerdere malen te hergebruiken. Schema's zijn optioneel, als ze ontbreken dan staat DB2 iedere XML structuur toe. Men kan dit geen 'foute data' noemen, want er is immers geen schema voor validatie. Het is toegestaan om schema's na verloop van tijd te vervangen door nieuwe schema's. Hierdoor worden reeds opgeslagen data niet geïnvalideerd. Hoewel dit een wenselijk feature is – net als het wijzigen van SQL table layouts – kan het ook tot rare bijverschijnselen leiden in applicaties. Schema of geen schema, XML data zijn minder 'robuust' dan relationele data.

DBA-tools en programmeeromgeving

Ook de toolset van DB2 is aangepast om XML te ondersteunen. Zo werden er aanpassingen gedaan aan backup, restore en datareplicatie om XML te ondersteunen. Database Import en Export kregen nieuwe faciliteiten om XML documenten te behandelen. XML documenten komen voor deze utility's uit aparte files, net als bij BLOB's). De grafische toolset, zoals control center en command center, kregen uitbreidingen voor XML data en XML indexen. Snapshot, Runstats en Explain geven nu ook uitgebreide informatie omtrent XML en de XML indexen. Er zijn geen gebieden waar de XML functionaliteit ontbreekt.
Alle programma-interfaces, van Cobol embedded SQL tot PHP kregen nieuwe interfaces om de XML data te kunnen aanbieden of ontvangen. De interfaces verschillen per programmeeromgeving. Daar waar de programmeeromgeving ook XML DOM ondersteunt, zal DB2 API's bieden voor DOM. De Developer Works website van IBM (www.ibm.com/developerworks/db2) heeft vele voorbeelden van de diverse mogelijkheden.

Conclusie

Vroeg of laat doet XML zijn intrede bij veel bedrijven. Op dat moment moet er een keuze gemaakt worden of XML wel of niet in de database opgeslagen wordt. Met de vele faciliteiten en de goede query faciliteiten van DB2 Viper zijn er waarschijnlijk meer voordelen dan nadelen voor opslag in de database. Voor degene die DB2 met zijn XML faciliteiten verder wil onderzoeken is de gratis versie DB2 Express Editie een aanrader. Voor wie meer over DB2 en XML wil lezen is het redbook *DB2 9: pureXML Overview and Fast Start* (SG24-7298-00) dat gratis gedownload kan worden van www.redbooks.ibm.com een goede start.

Klaas Brant
Klaas Brant (kbrant@kbce.nl) is database-specialist en oprichter van KBCE b.v.