

Native XML Databases

Ketan Palshikar

Database and XML – A brief history

- Term 'Relational Database Management System (RDBMS)' was coined by E.F. Codd in early 1970s
- RDBMS is a table structure with tuples and attributes
- XML was developed in late 1990s with the advent of the Web
- XML is a rework on SGML to answer some of the problems the Web faced as it was growing
- XML is a tree structure with nodes and branches

XML Databases

- Relational Databases are data-centric
- XML is document centric
- This is a challenge in converting XML into Relational Database
- XML Databases is a solution
- Offers two methods for converting XML into relational database
 - XML-enabled Databases
 - Native XML Databases

Database and XML

<Flights>

<Flight>

<Airline airlineID = "12345">American Airlines</Airline>
<Airplane>Boeing</Airplane>
<Origin airportID = "123">DAL</Origin>
<Destination airportID = "456">CHT</Destination>
<Departure>09:15</Departure>
<Arrival>11:19</Arrival>

</Flight>

<Flight>

<Airline airlineID = "98765">Delta Airlines</Airline>
<Airplane>Airbus</Airplane>
<Origin airportID = "456">CHT</Origin>
<Destination airportID = "678">RDU</Destination>
<Departure>12:15</Departure>
<Arrival>13:46</Arrival>

</Flight>

<Flight>

<Airline airlineID = "98765">Delta Airlines</Airline>
<Airplane>Boeing</Airplane>
<Origin airportID = "678">RDU</Origin>
<Destination airportID = "890">JFK</Destination>
<Departure>15:30</Departure>
<Arrival>19:50</Arrival>

</Flight>

</Flights>

Database and XML

airportID	Airport Abbr
123	DAL
456	CHT
678	RDU
890	JFK

Airline ID	Airline Name
12345	American Airlines
98765	Delta Airlines

Airplane
Boeing
Airbus

XML-enabled Databases Vs Native XML Databases

XML-enabled Database

- Maps XML schema to database schema
- Data is transferred using this mapping
- It has its own data model to map XML instances into database
- Creates tables like Airlines, Airports, Airplanes, etc.

Native XML Database (NXD)

- NXD has fixed (native) schema
- Uses XML model directly to map XML instances into database
- Uses tables for holding arbitrary XML document like Elements, Attributes, Text, etc.

Native XML Database

Element	Node Type
Flights	Root
Flight	First node
Airline	Second node
Airplane	Second node
Origin	Second node
Destination	Second node
Departure	Second node
Arrival	Second node

Attribute name
AirlineID
CityID

Text	Text Type
American Airlines	Character
Delta Airlines	Character
DAL	Character
CHT	Character
RDU	Character
JFK	Character
Boeing	Character
Airbus	Character
9:15	Time
11:19	Time
12:15	Time
13:46	Time
15:30	Time
19:50	Time

What is Native XML Database (NXD)?

Native XML databases have following properties:

- Defines a model for an XML document as opposed to the data in that document. Stores and retrieves documents according to this model. At a minimum, this model includes elements, attributes, PCDATA, and document order
- Has an XML document as its fundamental unit of storage, just as a relational database has a row in a table as its fundamental unit of storage
- Not required to have any particular underlying physical storage model. Built on a relational, hierarchical, or object-oriented database

Necessity of NXDs

- Native XML Databases (NXDs) are not meant to replace existing databases or XML but they intend to provide storage and manipulation of XML documents
- XML provides many characteristics of relational databases like storage in the form of XML document, schemas in the form of DTDs and XML Schema, query languages like XQuery and XPath and finally APIs like DOM and JDOM
- XML lacks many of the other characteristics of DBMS such as indexing, transactions, data integrity, triggers, normalization and updates
- Native XML Databases, while maintaining the already existing characteristics of XML provides most of the features or relational databases giving it advantages of both XML and RDBMS

Features

- Storage
- Querying
- Updates
- Indexing
- Normalization
- Referential Integrity
- APIs

Storage

- NXDs store data in the form of XML document
- This is useful for semi-structured data as storing semi-structure data into a relational database is difficult
- Advantage: Retrieval is faster as there are no joins while retrieving the document
- Disadvantage: Difficult to retrieve a different view of the data

Example:

- Retrieving a particular flight instance is faster
- Retrieving a list of all airline companies whose flights are flying from RDU airport is difficult

Querying and Updates

Querying

- Most NXDs support XPath and XQuery for querying
- XPath is most commonly used query language for NXDs. But XPath lacks functionality like grouping, sorting, cross document joins, etc.
- XQuery has overcome these shortcomings

Updates

- XUpdate is used for updating Native XML Databases
- Uses XPath to identify a set of nodes and then specifies whether to insert or delete these nodes, or insert new nodes before or after them.

Indexing

Three types of indexes

- *Value indexes*
 - They index text and attribute values
 - Used to resolve queries such as, “Find all elements or attributes whose value is ‘American Airlines’”
- *Structural indexes*
 - They index location of elements and attributes
 - Used to resolve queries such as, “Find all Airline elements”
 - *Value* and *structural* indexes are combined to resolve queries such as, “Find all Airline elements whose value is ‘American Airlines’”
- *Full-text indexes*
 - They index individual tokens in text and attribute values
 - Used to resolve queries such as, “Find all documents that contain the words ‘American Airlines’”
 - Used with structural indexes for queries like, “Find all documents that contain the words “American Airways” inside an Airline element”

Normalization

- Similar to relational databases, normalization can be done on NXDs as well but....
- XML supports multi-valued properties . Thus NXDs are normalized even when they have multi-valued attributes
- 1NF of relational database meaningless in context of NXDs
- For most of the collections, very small amount of data is repeated (corporate addresses, phone numbers, product logos, etc.)
- Thus normalization is a non-issue for many NXDs

Referential Integrity

- “Pointers” like key/keyref fields, Xlinks are used to achieve referential integrity
- Referential integrity, in context of NXDs, means ensuring that “pointers” point to valid documents
- Two types of referential integrities:
 - Integrity of internal pointers i.e. pointers within a document
 - Integrity of external pointers i.e. pointers between documents.
- Integrity of internal pointers is only partially supported
- Validation is performed only when a document is inserted into the database
- When updates are performed at the document level (e.g. deleting and replacing the document) internal pointers can perform validation

Referential Integrity

- When updates are performed at the node level (e.g inserting, modifying or deleting individual nodes) then internal pointers *may* not work
- Most NXDs support external pointers. BUT
- Referential integrity of external pointers is not supported
- External pointer points to a different document
- These pointers have no control over this document

Application Programming Interfaces (APIs)

- Native XML Databases support APIs
- They are generally similar to ODBC-like interface with methods for connecting to the databases and retrieving results
- Results are returned in the form of XML string or DOM tree or XML Reader
- Two commonly used APIs are:
 - XML:DB API from XML:DB.org
 - This is a programming language-neutral API
 - Uses XPath as its query language and is being extended to support Xquery
 - JSR 225: XQuery API for Java (XQJ)
 - It is based on JDBC
 - Uses XQuery as its query language
 - Developed through Sun's Java Community Process (JCP)

NXD Products

Product	Developer	License
4Suite, 4Suite Server	FourThought	Open Source
Berkeley DB XML	Oracle	Open Source
Birdstep RDM Mobile	Birdstep Technology / Raima	Commercial
DBDOM	K. Ari Krupnikov	Open Source
dbXML	dbXML Group	Open Source
Dieselpoint	Dieselpoint, Inc.	Commercial
DOMSafeXML	Ellipsis	Commercial
eXist	Wolfgang Meier	Open Source
eXtc	M/Gateway Developments Ltd.	Free
Extraway	3D Informatica	Commercial
Infonyte DB	Infonyte	Commercial
Ipedo XML Database	Ipedo	Commercial
Lore	Stanford University	Research
MarkLogic Server	Mark Logic Corp.	Commercial

NXD Products contd...

Product	Developer	License
M/DB:X	M/Gateway Developments Ltd.	Free
myXMLDB	Mladen Adamovic	Open Source
Natix	University of Mannheim	Free / non-commercial
ozone	ozone-db.org	Open Source
Qizx/db	XMLMind	Commercial
Sedna XML DBMS	ISP RAS MODIS	Free
Sekaiju / Yggdrasill	Media Fusion	Commercial
SQL/XML-IMDB	QuiLogic	Commercial
Sonic XML Server	Sonic Software	Commercial
Tamino	Software AG	Commercial
TeraText DBS	TeraText Solutions	Commercial
TEXTML Server	IXIASOFT, Inc.	Commercial
TigerLogic XDMS	Raining Data	Commercial
Timber	University of Michigan	Open Source (non-commercial only)

NXD Products still contd...

Product	Developer	License
TOTAL XML	Cincom	Commercial
Virtuoso	OpenLink Software	Commercial
XDBM	Matthew Parry, Paul Sokolovsky	Open Source
XDB	ZVON.org	Open Source
XediX TeraSolution	AM2 Systems	Commercial
X-Hive/DB	X-Hive Corporation	Commercial
Xindice	Apache Software Foundation	Open Source
xml.gax.com	GAX Technologies	Commercial
Xpiori XMS	Xpiori	Commercial
XQuantum XML Database Server	Cognetic Systems	Commercial
XStreamDB Native XML Database	Bluestream Database Software Corp.	Commercial
Xyleme Zone Server	Xyleme SA	Commercial

NXD Product – MarkLogic Server

- MarkLogic Server is a native XML database that can convert Word, PowerPoint, Excel, PDF, and HTML documents to XML
- Supports Xquery and full-text search using wildcards, stemming, spell checking and so on
- It has a single index that combines *full-text index*, *value index* and *structural index*
- Uses Java and .NET APIs
- Supports transactions, triggers, journaling, role-based security, clustering, and backup
- <http://www.marklogic.com/product/demos.html>

References

1. <http://xmldb-org.sourceforge.net/faqs.html>
2. <http://www.rpbouret.com/xml/XMLAndDatabases.htm#datavdocs>
3. <http://www.xml.com/pub/a/2001/10/31/nativexmldb.html>
4. <http://www.rpbouret.com/xml/XMLDatabaseProds.htm>
5. http://swing.felk.cvut.cz/index.php?option=com_docman&task=doc_view&gid=5&Itemid=62

Thank you