Oracle Database 11g XML DB

Geoff Lee
Principal Product Manager
Oracle XML DB
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remain at the sole discretion of Oracle.
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
XML Use Cases

- Common paradigm for managing structured and unstructured data
  - Flexible data model equally suited to data-centric and document centric application development
- Open, vendor-neutral standards, driven by W3C
  - XML, XMLSchema, XQuery, SOAP
- Widely adopted in multiple industries
  - Growth of XML based industry standards
    - XBRL, FIXML, FpML, ACORD, MISMO, HL7, NIEM,…
- Primary storage format for next generation office productivity suites
Why XML in the database?

- XML being used to manage mission critical information
  - Interchange with external organizations
  - Web Services
- Need to manage XML effective and efficiently
  - Number and size of documents increasing
  - Reliability, Scalability, Availability
  - Security
  - Compliance
- Accurate and fast information location and retrieval
Evolution of Oracle’s XML Support

- 1998: XML API’s
- 2001: XML Storage & Repository
- 2004: XQuery
- 2007: Binary XML Storage & Indexing

Performance
Oracle XML DB Firsts

- Native storage of XML and XML standards
  - XML Namespaces, XPath, XML Schema, XSLT, WebDAV
- XML Repository for managing XML content
- Implementation of W3C’s XQuery specification
- SQL/XML:2003, 2006 compliant XML publishing
- SQL/XML and XML/SQL interoperability.
XML Standards supported

- XML, Namespaces
- XQuery, XPath
- XML Schema
- XSL and XSLT
- DOM

- DAV, HTTP, FTP
- SQL/XML
XML DB value propositions

• Fast and easy native XML application development
• Hybrid database
  • SQL centric access to XML content
  • XML centric access to relational content
• XML DB repository enables document centric integrity and security models
• Multiple XML storage options allow tuning for optimal application performance
  • Application code is totally independent of storage model
  • Optimized storage and indexing for structured and unstructured XML
SQL ↔ XML

XML Apps

XQuery/XPath

SQL/XML

XML data model

Document centric

Data centric
Oracle XML DB Overview

- **Schema-Based XML**
  - Object-Relational Storage
  - Binary Storage

- **Schema-Less XML**
  - Binary Storage
  - Text Storage

- **Relational Content**
  - XML Views

- **XML Indexing**
  - B-Tree Indexing
  - XML Indexing
  - Functional Indexing
  - Full-Text Indexing

- **Access Paradigms**
  - Table / Row
  - File / Folder
  - Content and Meta-data

- **XML Document or Message**
- **XML Application**
- **JDBC**
- **.NET**
- **OCI**
- **SOAP**
- **HTTP**
- **FTP**
- **WebDav**
- **Files**
- **Folders**
- **ACLS**
- **Versioning**
- **Metadata**
- **XML Schema**
- **SQL/XML**
- **XSLT**
- **DOM**
- **XML Type**

- **XML Application**
- **XDK**
XML Storage Models

- **XML-CLOB**: Unstructured, Flexibility
- **XML-Binary**: Structured, Performance
- **XML-OR**: Schema Optimized Persistence
- **Schema Less Persistence**

Performance vs. Flexibility diagram.
Oracle XML DB Overview

- Persistence
- DML
- Repository
- Publishing
Native XML data type

- Abstraction for Storing XML in the database
  - Use as Table, Column, Variable, Argument or Return Value
- Supports both Schema-based and Schemaless XML
- Multiple native persistence models for storing XML in the database
- Guarantees XML Fidelity
  - XML Content model is maintained
  - XML Schema Validation
  - SQL Style Referential Integrity
Versatile Persistence Models

- Applications require flexible persistence models
  - Different models to meet different application requirements
  - Model can be tuned to deliver application performance
- Persistence model is transparent to application
  - Developer should code independent of the persistence model
  - Changes to the persistence model must not require changes to application code
XML Schema-Based Persistence

• WC3 Standard
  • Defines the structure of XML documents
  • Authored by Tools like Oracle’s JDeveloper, Altova’s XMLSpy
  • XML Editors, Tools and Database can validate XML content against XML Schema

• XML DB provides XML Schema based optimizations
  • Automatic mapping between XML and SQL object models
  • XQuery operations over XML automatically re-written into SQL query over SQL objects
XML Schema in XML DB

- XML Schema registration with XML DB
- XML columns can be constrained by registered XML Schema
- XML Schema for integrity
  - Ensures data integrity by constraining what can be stored
  - Allows users to perform complete XML Schema validation
- XML Schema controls OR Storage
  - Storage mappings
  - In-memory representations
  - Enables XML queries to map to relational tables & columns
- XML Schema enables optimizations
  - Minimizes data conversions
  - Enables queries and updates to perform better
OR Storage

- Attributes and single-valued elements
  - Stored as columns in single row
  - SQL data types correspond to XML Schema types
  - SQL constraints correspond to XML Schema constraints
- Multi-valued elements (collections) stored in separate nested tables
  - One row per item in collection
  - Nested table row stores parent key
  - Array Index column stores the position information
- Supports multiple levels of nesting
  - Embedded object types
  - Embedded collection types with multiple nested tables
Versatile XML Indexing

- **B-TREE Indexing**
  - Allows indexing of specific scalar values
  - Requires Schema based XML persistence

- **Functional Indexing**
  - Allows index of specific element or attribute values
  - Path Expression must be unique within document

- **Full text indexing**
  - Document and node level text indexing
XML Schemaless Persistence

- Binary XML Storage Model
- CLOB Storage Model
  - Document-fidelity
Versatile XML Indexing

- XMLIndex Indexing
  - Allows indexing of XML sub-trees
  - XMLTable Indexing
- Secondary full text indexing
  - Document and node level text indexing
Oracle XML DB Overview

- Persistence
- Repository
- Publishing
- DML
XML Operations

• Document Level operations
  • Create, Read, Update and Delete
  • Schema Validation
  • XSL Based Transformation

• Full support for W3C XQuery Specification
  • Enable XQuery over persisted XML content and XML Views of relational data
XML Operations

• Node Level operations
  • Selection based on document content
  • Extraction of scalar values and Fragments
  • Update of scalar values or fragments
  • Deletion of nodes or Fragments
  • Insertion of nodes or Fragments

• DOM API
  • PL/SQL, Java and C
Oracle XML DB Overview

- Persistence
- DML
- Repository
- Publishing

XQuery
What is XQuery?

- W3C language for querying and generating XML
  - Natural query language for XML content
- Evolved from XPath and XSLT
- Analogous to SQL in the relational world
XQuery Usages

- Transformation
  - XML to XML Transformation
- Templating
  - XML Construction
- Querying XML documents
  - Selection, Iteration
XQuery Expressions

- Everything in XQuery is an “expression”
- Every expression returns a new value
- Major Expressions
  - FLWOR expressions –
    - For, Let, Where, Order-by Return
  - Path Expressions - /a/b/c
- Powerful, extensible Functions and Operators Library
  - Shared with XSLT 2.0
XQuery Status

- XQuery 1.0 has been a W3C recommendation since Jan. 23, 2007
- Proposed XQuery extensions
  - Grouping, Exception handling
- XQuery Full-Text extensions
- XQuery update DML extensions
  - Insert, Update, Delete
XQuery and SQL:2007

• XMLQuery() Function
  • Used in SQL SELECT clause
  • XML-centric
  • Applies an XQuery to one or more XML documents
  • Returns the result of evaluating the XQuery expression as an XML document

• XMLTable() Function
  • Used in SQL FROM clause
  • SQL-centric
  • Maps the result of an XQuery evaluation into relational rows and columns
  • Enables SQL based operations on XML content
XQuery and SQL:2007

- XMLExists() Function
  - Used in SQL WHERE clause
  - Query filtering
- XMLCast() Function
  - Used in SQL SELECT clause
  - Datatype conversion
Oracle’s XQuery support

- XQuery supported in Database and Application Server
- First RDBMS Vendor to market with an XQuery implementation
- Oracle Database 10g Release 2 supports XMLQuery() and XMLTable() operators
- Oracle Database 11g adds XMLExists() and XMLCast() operators
- Oracle’s XQuery implementation will conform to the W3C XQuery standard
- Direct execution of XQuery via SQL*Plus XQUERY command
XQuery database support

- Native ‘C’ and Java execution paths
- Native Execution path leverages Oracle XML DB technology
- JSR 225 (XQJ) API for XQuery forthcoming
Native Execution Path

- Optimized XQuery performance for native data sources
  - Schema-based XMLType tables and views
  - SQL/XML views over relational tables
  - XML DB repository
- Rewrites XQuery to native relational structures
- Leverages relational engine and optimizer
XMLQuery() Operator

- Primarily used in the Select List of a SQL statement
- Inputs
  - XQuery expression,
  - Context item (XML)
  - Bind variables
- Output
  - XMLType containing the result of evaluating the XQuery expression
- Logical evolution of Oracle’s extract() operator
XMLQuery() Function

```sql
SQL> SELECT warehouse_name, XMLQuery(
    'for $i in /Warehouse where $i/Area > 10000 return
    <Details><Docks num="{$i/Docks}"/>
    <Rail>{
        if ($i/RailAccess = "Y") then "true" else "false"
    }
    </Rail>
    </Details>'
passing warehouse_spec returning content) big_warehouses
from warehouses;

WAREHOUSE_NAME          BIG_WAREHOUSES
--------------------- --------------------
Southlake, Texas
<Details>
    <Docks num="2"/>
    <Rail>false</Rail>
</Details>
...
```
XMLTable() Operator

- SQL-centric
- Used in the From Clause of a SQL Expression
- Inputs
  - A Set of XQuery Expressions
  - Context Item (XML)
  - Bind Variables
- Output
  - Relational row set which can be manipulated via SQL
- Logical evolution of Oracle’s table(xmlsequence(extract(...))))
XMLTable() example

CREATE VIEW PURCHASEORDER_VI EW AS SELECT extractvalue(pocol, '//PurchaseOrder/@Reference') as REF, lines.* FROM purchaseorder, XMLTable('//LineItems[Quantity > 200]' columns lino number path 'LineNum', part varchar2(20) path 'PartName', quantity number path 'Quantity') lines;

Select * from PURCHASEORDER_VI EW where pono in (10, 20);

REF   lino   part          quantity
-----   ------  -------------  ---------
10      100    CDROM         201
10      101    Monitor      223
20      101    CPU Fan      300
Oracle Functions and Operators

- **ora:View**
  - Enables XQuery operations on relational tables

- **ora:Contains**
  - Enables XQuery operations to leverage Oracle Text

- **ora:Doc**
  - Enables XQuery operations on content stored in XML DB repository

- **ora:Collection**
  - Enables XQuery operations on the contents of a folder in the XML DB repository
Oracle XML DB Overview

- Persistence
- DML
- Repository
- Publishing
XML DB repository

- Enables Hierarchical organization of content
  - Based on IETF DAV standard
  - File / Folder organization of content
- Allows URL based as well as table/row based access
  - Required to model XLink, XPointer and XInclude
- Supports basic Content Management features
  - Access control
  - Versioning
  - Events
  - User defined Metadata
Oracle XML DB Repository

• Support for WebDAV, HTTP(S) and FTP protocols
  • Office productivity tools, XML editors and Windows Explorer have direct access to content stored in XML DB
• Repository and Path based access and update are available from SQL
  • Allows programmatic access via multiple APIs
• Hierarchical Index
  • Patented, high performance folder-traversal operations and queries
Oracle XML DB Overview

- Persistence
- DML
- Repository
- Publishing
XML Publishing

- SQL/XML Publishing functions incorporated in the SQL:2003 Standard
- Allow the result of a SQL query to one or more XML documents
- Defines new SQL Operators
  - XMLElement, XMLAttributes, XMLAgg, etc.
- Supported by Oracle since Oracle 9i Release 2
- XQuery can accomplish the same
SQL / XML Support

- SQL/XML allows complex XML documents to be generated from relational queries
  - E.g., XML Compliant with Microsoft’s SpreadsheetML
- XMLType views provide XML access to relational content
  - Rows in an XML type view can be exposed as an XML document in the repository.
  - Content can be access by Office Productivity tools using WebDAV
Key ISVs and SIs helping to advance XQuery in conjunction with Oracle products

- Tools: XMLSpy, oXygen, DataDirect, HP/Systinet
- Apps: JustSystems, UBMatrix, Nextance, WMG
Public References

- **State of California’s Legislative Counsel Bureau** Deploys Oracle XML DB for the Re-vamped State Legislative Publishing and Drafting Process Mar 2007
- **Motorola** Relies on Oracle XML DB for Its Biometrics Product Lines Nov 2006
- **UPS** Streamlines Its High Volume Daily Field Transactions with Oracle XML DB Jul 2006
- **ThyssenKrupp Steel** Uses Oracle XML DB to Integrate XML Documents into Data Warehouse Jul 2006
- **Robert Bosch India** Deploys an Integrated Oracle XML DB and Forms Application (PDF) Apr 2006
- **Nextance** Delivers a CPM Solution Using Oracle XML DB and XQuery Mar 2006
- **Temenos**' Customer Went Live with an Integrated Banking System Running on Oracle XML DB Feb 2006
XML DB Customers

• SQL Centric
  • WMG, Thyssen, UPS
  • Temenos

• XML Centric
  • Energysys/BP, Motorola

• Document Centric
  • State of California’s Legislative Counsel Bureau
  • Nextance
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
What’s new in Oracle Database 11g?

- Increased Flexibility for Schema-optimized storage
  - Zero down-time for schema change
  - Partitioning support
  - Intelligent defaults
  - General / Performance Enhancements
- Better management and performance for Schema-Less XML scenarios
  - Binary XML
  - XML Index
- Direct integration in to Web Service Architectures
- Programmable Repository and Compound Documents
- Improved Security
- XDK Enhancements
11gR1 : In Place Schema Evolution

• Allows simple changes to registered XML schemas with zero down-time
  • No data copy required
  • Schema change takes a few seconds regardless of amount of data.
• Changes to the XML must not invalidate existing documents.
  • Add optional elements and attributes
  • Adding new values to enumerations
  • Increase in length
11gR1 XMLType partitioning

- Leverages the 11gR1 “Ref-Based” partitioning feature
- Nested Tables are partitioned using the same key as the top level XMLType table
  - Includes supports for “out-of-line” storage models
- Improved Manageability
  - “partition maintenance” operations
  - All operations are performed on the XMLType table
- Improved query optimization
  - “partition pruning”
11gR1: Intelligent Defaults

- Avoids most common limitations and bottlenecks
  - Default storage model is now collections stored as nested-tables
- Default organization is heap-organized nested-tables
  - Optimal re-write for XQuery expressions and XMLTable-based operations
  - Partitioning and indexing
  - Text based indexing
- Optional, automatic horizontal partitioning for wide structures
11gR1 General Improvements

- XMLCast and XMLExists operators
  - XMLCast: XQuery for fragment extraction
  - XMLExists: XQuery in where clause
- Streaming XSLT
  - Significant performance improvements when generating large documents
  - XSLT output no-longer restricted to well-formed XML
- Large node handling eliminates current 64K limit on size of a text-node.
- Stream based replication support for Text-based XML storage
11gR1 Performance Improvements

- Significant performance improvements
  - Up to 10x performance improvement when ingesting XML
  - Up to 10x performance improvement when generating XML
    - XMLAgg optimization
  - General improved optimization of XQuery expression on Schema-Optimized storage
  - Significant optimization for fragment level DML operations on collections
11gR1: Binary XML

[Diagram of 11gR1 architecture with Binary XML highlighted throughout the database, app server, and web cache.]
11gR1 : Binary XML Storage Option

- Supports Schema-Less and Schema-based XML
- Post Parse representation of XML
- Single format for “on-disk”, “in-memory” and “on-the-wire”
- Optimized for indexing and fragment extraction
- Flexible XML Schema Support
  - No “Schema-Evolution” issues
  - Reduces storage requirements
  - Heterogeneous XML documents in a single table or column
- Tight integration with Oracle Secure Files
11gR1 Binary XML advantages

• **Compact Storage Format**
  • Tag-names are tokenized
  • Text-nodes and attribute values are stored in native representation, rather than text format

• **Reduced CPU and Memory overhead**
  • Pull parser, streaming validation and streaming XPath avoid use of DOM Tree
  • Single format avoids parse and serialize issue when moving XML between application tiers

• **Reduced Network Overhead**
  • Post-parse format used when XML moves between tiers
11gR1 Binary XML advantages

- High performance fragment access and extraction
  - Streaming XPath allows multiple nodes to be accessed in a single operation
- High performance update
  - Sliding inserts
  - Partial re-validation following update
- Locale Sensitive
  - Support XLiff, allowing locale-sensitive retrieval of content
11gR1: XML Index

• New universal index for Binary and Text based XMLType storage models
• Addresses all known issues with CTX-XPath index
• Optimizes most common classes of Path Expressions
  • Recursive, Relative, Lazy (/)
• Accelerates path & value based predicates
• Fully type aware
  • Optimizes numeric and date range predicates
• Fully namespace aware
11gR1 : Totally transparent

- Support fragment extraction as well as fragment existence
- Supports all XQuery operators
  - XMLQuery, XMLTable, XMLExists, XMLCast
- Supports DML operators and legacy XPath operators
  - updateXML, insertChildXML,
  - existsNode, extractValue, xmlSequence …
- No code changes required
- Certain queries are executed totally against the index
11gR1: XML Index Optimizations

- XML Index Can get large / expensive to maintain
  - By default all possible paths are indexed by default
  - High disk usage
  - Impact on throughput
- XML index optimization
  - Asynchronous Operation
  - Path Sub-setting
  - Partial re-indexing
11gR1 : XML Index Optimization

- Synchronous and Asynchronous indexes
  - Determined at Index creation time
- Synchronous Mode
  - Insert operation does not complete until indexing complete
  - Query results always include DML changes
- Asynchronous Mode
  - Insert operation does not wait for indexing to take place
  - Query results may not reflect latest DML changes
  - Index maintain takes place automatically in near-real time
- No Fragmentation Issues in either mode
XML Index Optimizations

• Path Subsetting
  • Provides control over which nodes within the document are indexed
  • Trade index size, throughput against query performance
  • Index can only optimizes performance for the indexed Paths
  • Can add or remove paths dynamically

• Partial re-indexing
  • Only re-index modified content
  • Requires Binary XML and Secure Files
11gR1 : XML Index features

- Repository integration
  - Create XML Index on XML DB repository
  - Optimize queries on meta-data and content

- Hybrid Storage
  - Create XML index on fragments in Schema-Optimized storage that are mapped to CLOB
  - Enhanced query capabilities

- ‘Data Guide’
  - Index can provide meta-data about the XML it has indexed

- Tight integration with Oracle Text
  - Index driven full-text search
Enhanced XQuery Support

• XQuery supported in conjunction with XMLIndex and Binary XML
• Support of SQL/XML 2006 standards
  • New operators: XMLCast(), XMLExists()
  • W3C XQuery 1.0 recommendation
• Performance improvements
  • Improve XQuery-rewrite for schema-based storage
  • Enhanced support for operations on recursive structures
  • Optimization of XML update operations
Database-native Web Services

- WSDL Request
- WSDL
- SOAP Request
- SOAP Response
- PL/SQL
- SQL
- XQuery
Database-native Web Services

  - Any package method, function or procedure can be accessed as a SOAP end-point
- Leverages the Oracle XML DB HTTP Server
  - No additional infrastructure required
- Automatic generation of WSDL
  - URL to Package, Function or Procedure mapping scheme
- Uses XML DB infrastructure for processing request and generating response
- Includes ‘SQL Query’ and ‘XQuery’ Services
XML DB repository events

• ‘Triggers’ for the XML DB repository.
  • Enables ‘server-enforced’ rules for XML DB repository
  • PL/SQL procedures automatically executed as a result of repository activity

• Multiple-models
  • Repository Wide
  • Folder or Folder-Tree specific
  • Content specific

• Associate code with
  • CRUD (Create, Replace, Update and Delete) operations on documents
  • Folder Link and Unlink
XML DB repository enhancements

- XLink and XInclude
  - Support for compound documents
  - XML centric referential integrity
  - Enforcement of XLink based references
- Extended Links
  - Support for ‘Soft’ links
XML DB Content Connector: JSR-170

- Standardized APIs accessing XML DB content repository
  - Manage structure, unstructured and semi-structured content in one repository
  - Traverse, query, access, and operate data using standard APIs
- Support JSR 170 Level 1 and Level 2
XML DB Security enhancements

- Support for DAV ACL
  - Security Interoperability with emerging DAV Clients
- ACL inheritance
  - Define Common Security Model
    - Organization-Wide policies
    - Content-Specific policies
- User Defined ACLs
  - Use ACL mechanism to manage access to non XML DB objects
XDK Enhancements

• XDK “C”
  • Pull Parser
  • Streaming Validator
  • Binary XML integration

• XDK “J”
  • Binary XML integration
  • Scaleable ‘DOM’

• XML Diff and XMLPatch
  • Stand alone utility
  • ‘C’ API
  • PL/SQL API
  • Java ‘forthcoming’
XML Diff and Patch

- Compare two XML documents
- Represent the difference in XML
- Apply and control the changes
XMLDiff Output

```xml
<xddf xsi:schemaLocation="http://xmlns.oracle.com/xdb/xdiff.xsd
 http://xmlns.oracle.com/xdb/xdiff.xsd">
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xdb="http://xmlns.oracle.com/xdb"
  xmlns:xd="http://xmlns.oracle.com/xdb/xdiff.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <?oracle-xmlendiff operations-in-docorder="true" output-model="snapshot" diff-
algorithm="global"?>
  <xd:append-node xd:node-type="element" xd:parent-xpath=
"/xs:schema[1]/xs:simpleType[6]/xs:restriction[1]">
    <xd:content>
      <xs:enumeration value="E999"/>
    </xd:content>
  </xd:append-node>
</xd:xdiff>
```
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
## XML Storage/Index Options

<table>
<thead>
<tr>
<th>Primary Content vs Embedded Content</th>
<th>Embedded Unstructured Content</th>
<th>Embedded Structured Content</th>
</tr>
</thead>
</table>
| **Data Centric**                   | Hybrid Storage with B-Tree, XML Indexes  
(e.g. employee record with resume) | Object Relational Storage with B-Tree Indexes  
(e.g. employee record in XML) |
| **Document Centric**               | Binary XML Storage with XML Indexes  
(e.g. XML document from the web) | Binary XML Storage with XML Indexes  
(e.g. Functional Specification with author, date, title fields) |
## Use Case Guidelines

<table>
<thead>
<tr>
<th>Primary Content vs Embedded Content</th>
<th>Embedded Unstructured Content</th>
<th>Embedded Structured Content</th>
</tr>
</thead>
</table>
| Data Centric                       | Hybrid Storage with B-Tree & XML Indexes  
(e.g. employee record with resume) | Object Relational Storage with B-Tree Indexes  
(e.g. employee record in XML)  
NCPA, WMG, UPS, etc. |
| Document Centric                   | Binary XML Storage with XML Indexes  
(e.g. XML document from the web) | Binary XML Storage with XML Indexes  
(e.g. Functional Specification with author, date, title fields) |
Data-Centric with Structured Components

- Well-defined XML schema
- Non-sparse XML instances
- Schema changes can be handled by In-place or copyEvolve schema evolution
- Schema-based (O-R) storage model
- B-tree indexes
## XML Storage/Index Options

<table>
<thead>
<tr>
<th>Primary Content vs Embedded Content</th>
<th>Embedded Unstructured Content</th>
<th>Embedded Structured Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Centric</strong></td>
<td>Hybrid Storage with B-Tree &amp; XML Indexes (e.g. employee record with resume)</td>
<td>Object Relational Storage with B-Tree Indexes (e.g. employee record in XML)</td>
</tr>
<tr>
<td><strong>Document Centric</strong></td>
<td>Binary XML Storage with XML Indexes (e.g. XML document from the web)</td>
<td>Binary XML Storage with XML Indexes (e.g. Functional Specification with author, date, title fields)</td>
</tr>
</tbody>
</table>
Data-Centric with Unstructured Components

- Well-defined XML schema
- Non-sparse XML instances
- Schema changes can be handled by In-place schema evolution
- Hybrid storage
  - Schema-based storage model
  - Supplemented with CLOBs
- B-tree indexes for structured components
- XML indexes and secondary full-text indexes for unstructured components
## XML Storage/Index Options

<table>
<thead>
<tr>
<th>Primary Content vs Embedded Content</th>
<th>Embedded Unstructured Content</th>
<th>Embedded Structured Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Centric</strong></td>
<td>Hybrid Storage with B-Tree &amp; XML Indexes <em>(e.g. employee record with resume)</em></td>
<td>Object Relational Storage with B-Tree Indexes <em>(e.g. employee record in XML)</em></td>
</tr>
</tbody>
</table>
| **Document Centric**                | Binary XML Storage with XML Indexes *(e.g. XML document from the web)* | Binary XML Storage with XML Indexes *(e.g. Functional Specification with author, date, title fields)*
|                                    |                             | **UBMatrix, Waters, EnergySys, Motorola** |
Document-Centric with Structured Components

- Schemaless or …
  - Very complex schema with sparse XML instances
  - Multiple schemas used by XML docs in one table
- Schema evolution unnecessary
  - Document versioning
- Binary XML (schemaless or schema-based) storage model
- XML indexes
# XML Storage/Index Options

<table>
<thead>
<tr>
<th>Primary Content vs Embedded Content</th>
<th>Embedded Unstructured Content</th>
<th>Embedded Structured Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Centric</td>
<td>Hybrid Storage with B-Tree &amp; XML Indexes (e.g. employee record with resume)</td>
<td>Object Relational Storage with B-Tree Indexes (e.g. employee record in XML)</td>
</tr>
<tr>
<td>Document Centric</td>
<td>Binary XML Storage with XML Indexes (e.g. XML document from the web) Nextance, LDC</td>
<td>Binary XML Storage with XML Indexes (e.g. Functional Specification with author, date, title fields)</td>
</tr>
</tbody>
</table>
Document-Centric with Unstructured Components

- Schemaless
- Schema evolution unnecessary
  - Document versioning
- Binary XML storage model
- XML indexes
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
Future Roadmap

- Performance improvements in general from internal cases
- XQuery
  - Modules
  - XQuery Update (actively involved in the committee, DB2 using earlier version still in working draft, not in CR yet, 2nd half of 2008?)
- XQJ (completed reference implementation, targeting DB connection in 11.2)
Agenda

- Introduction
- Overview
- Oracle Database 11g XML DB
- Best Practices
- Future Roadmap
- Hands-on Lab
  - Basics (OR Storage, XQuery, etc.)
  - Binary XML Storage and XMLIndex
  - Web Services
  - Schema Evolution
- Summary
- Q & A
Agenda

• Introduction
• Overview
• Oracle Database 11g XML DB
• Best Practices
• Future Roadmap
• Hands-on Lab
• Summary
• Q & A
Summary

- XML use cases are diverse
- An XML solution needs to match a particular use case
- Oracle XML DB provides one XMLType abstraction with multiple use-case-specific storage and indexing options
- Take advantage of Oracle internal and external resources
  - OTN Oracle XML DB page
  - Oracle internal XML DB Web Site
  - helpxdb_us@oracle.com