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 decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

THE INFORMATION COMPANY

ORACLE

Geoff Lee
Principal Product Manager
Oracle Corporation
Robert Couch
Managing Director
Nextance Inc.

ORACLE

XQuery Support in Oracle Database 10g Release 2

ORACLE

- **Overview**
- Oracle Database-Native XQuery
- Nextance Case Study
- Conclusion

What is XQuery?

- A W3C standards effort to enable data to be extracted from XML documents
- XQuery is designed to work with the XML data model
- A comprehensive query language for data expressed in XML, as
  - SQL for structured relational data
  - Keyword searches for unstructured information access on the Internet
- Also useful for constructing and transforming XML documents
- Oracle Database 10g Release 2 debuts a full-featured database-native XQuery engine

Oracle XML DB

- A native XML store introduced in Oracle Database 9i Release 2
- Useful for a certain data-management tasks
  - Content Management – standards-based life-cycle management (e.g., technical manuals, multi-media messages, legal statutes, regulatory filings)
  - Exchange and Storage – generation and storage of template-based business documents (e.g., purchase orders, bills, invoices, and reports to exchange between applications)
  - Data Integration – querying across different types of information assets (e.g., database records, files, web servers, news feeds)
- The rapidly maturing Oracle XML DB technologies are being used for a variety of applications incorporating the above tasks

Why XQuery?

- The initial use of XML as a storage format is broadening into the need for querying the stored XML
- Conciseness and simplicity
  - One-fifth the code of XSLT-oriented approaches
  - One-twentieth the code of DOM-based approaches
- Heterogeneous queries
  - XML can map to a large number of different data models, and therefore XQuery can be used to search databases, files, and web-services simultaneously
- Semi-structured XML data model
  - the XML Schema-based data model is best suited for representing semi-structured data and queried by XQuery
- XML construction and transformation
  - XQuery can construct XML as the result of evaluating query expressions, in many cases more expressively and efficiently than XSLT

The Structure of XQuery

A simple document named emp.xml:

```
<empset>
 <emp empno="21" ename="SCOTT" salary="50000"/>
 <emp empno="22" ename="JONES" salary="100000"/>
</empset>
```

To get the names of those employees with salary > 80000,

```
for $i in doc("emp.xml")/empset
let $j = 80000
where $i/@salary > $j
return $i/@ename
```

The result is the following attribute node: Jones

FLWOR Expressions

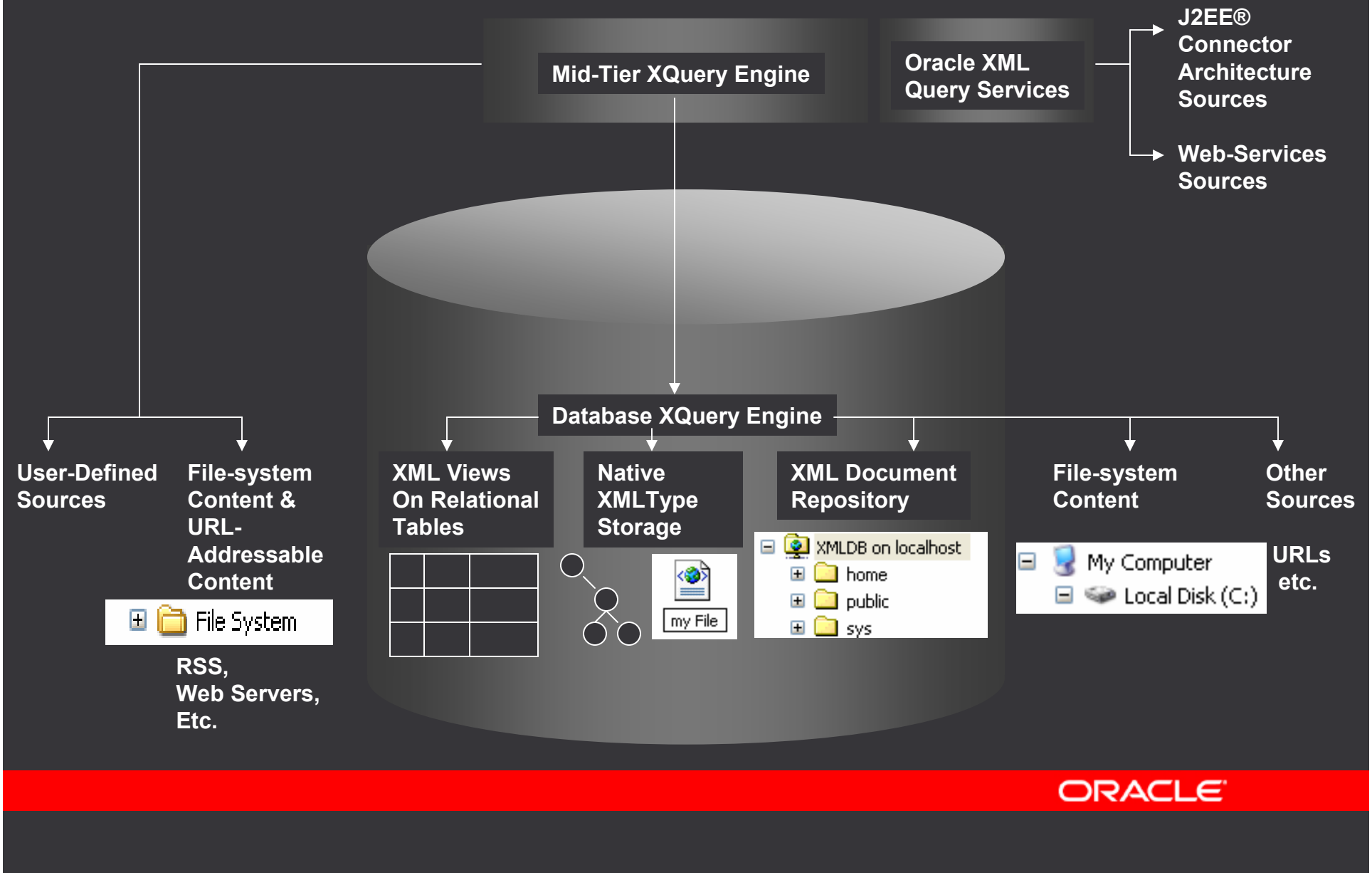
- For: Analogous to the SQL FROM clause
  - Iterates across a range of sequence values
- Let:
  - Let clause lets you define variables and assign them in turn during iteration through a FOR clause
- Where: Analogous to the SQL WHERE clause
  - Provides a set of conditions that filter or limit the initial selection in a For statement
- Order-By: Analogous to ORDER BY in SQL
  - Provides the ordering constraints on a sequence.
- Return: Analogous to the SQL SELECT clause
  - Creates output in a custom formatting language

Other XQuery Expressions

- Namespaces: the *declare namespace* prolog declaration
  - Associates a namespace URI with a prefix, crucial for indicating scoping rules on functionality
- Functions and Operators: Analogous to SQL functions and operators
  - Built-in functions and operators: numerics, strings, etc.
  - User-defined functions
- XPath: XQuery and XPath 2.0 share the same data model and the functions and operators specifications

- XQuery Overview
- Oracle Database-Native XQuery
  - **Architecture and implementation**
  - Usage Scenarios
    - Querying XML DB Repository
    - Querying Relational Data
    - Querying Native XML Data
    - Querying Diverse Data Sources
  - Database XQuery Environment
- Nextance Case Study
- Conclusion

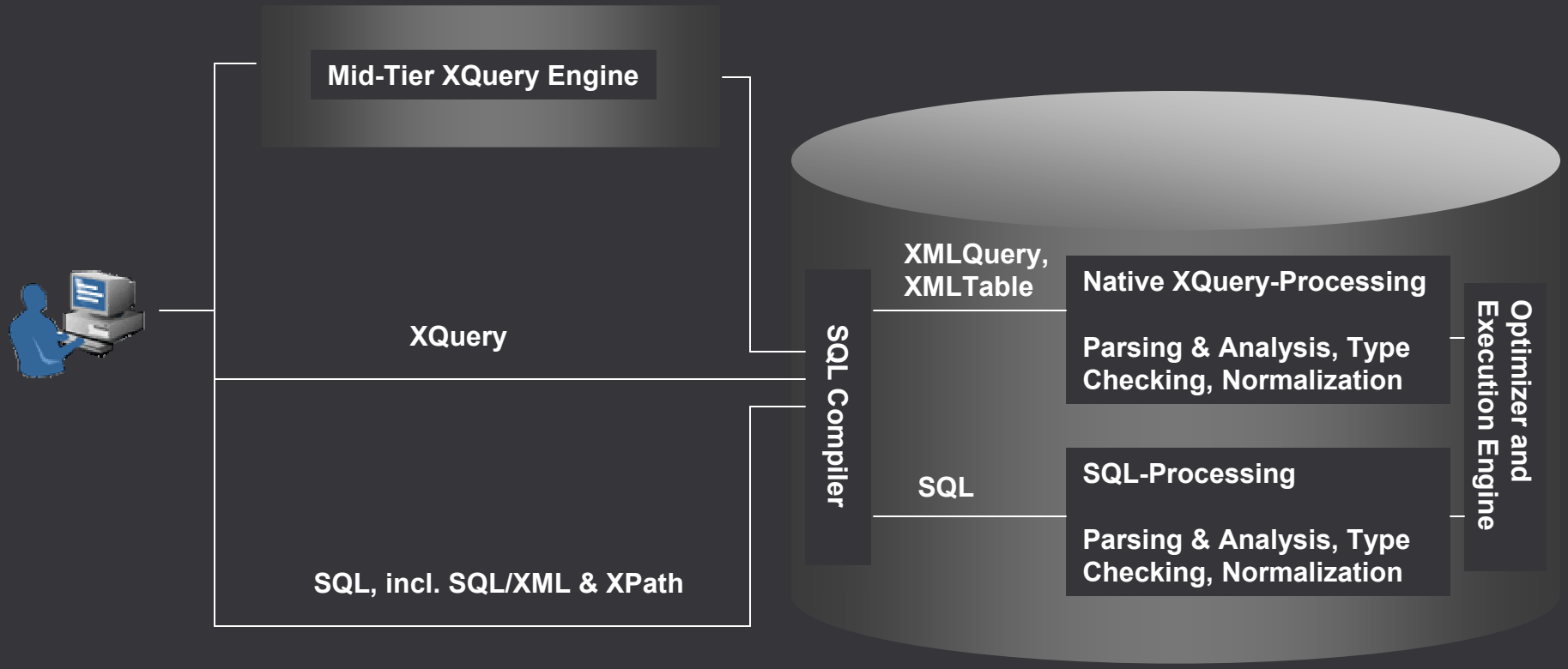
Oracle XQuery Architecture



Oracle XQuery Architecture

- Oracle supports XQuery running in both database and mid-tier
  - The database XQuery engine: For large amounts of persistent data (relational or XML)
  - The mid-tier XQuery engine: For small amounts of metadata, transient messages, or configuration information; or a large number of heterogeneous sources
- Oracle Database is the leading platform for storing all your data
- Oracle Database supports XML data model, XQuery, Web Services, etc.
- With ‘duality’ at the heart of Oracle XML DB, operational data and content, can be
  - Stored in databases
  - Modeled as XML or SQL
  - Queried by whichever metaphor is more appropriate

Oracle's Native XQuery Implementation



Native XQuery Implementation

- Goes deeper than the co-processor approach
- Directly generates low-level database query execution structures and query sub-blocks
  - Optimizes XQuery expressions in the context of the global SQL statement
- Further optimized by the underlying database optimizer
- Efficiently executed by the Oracle execution engine
- Tightly integrates XQuery and SQL/XML within the database kernel
- Delivers performance that is orders of magnitude faster than the co-processor approach
- Utilizes standard indexes
- Eliminates unnecessary materializations of XML values
- It is *transparent* to end users

Using XQuery in The Database

- **Querying the Oracle XML DB Repository**
  - XQuery is a natural way to query documents stored in Oracle
- **Querying a Relational Table or View as XML data**
  - Oracle XQuery function *ora:view()* creates an XML view over the relational data
- **Querying native XMLType data in Oracle XML DB**
  - XQuery can operate on natively stored XMLType data
- **Querying across Databases, File Systems, and Websites**
  - Call out from the database
  - Read in a file using the *doc() or the collection()* function
  - Perform a query across both data in the database and a document read in from an external source

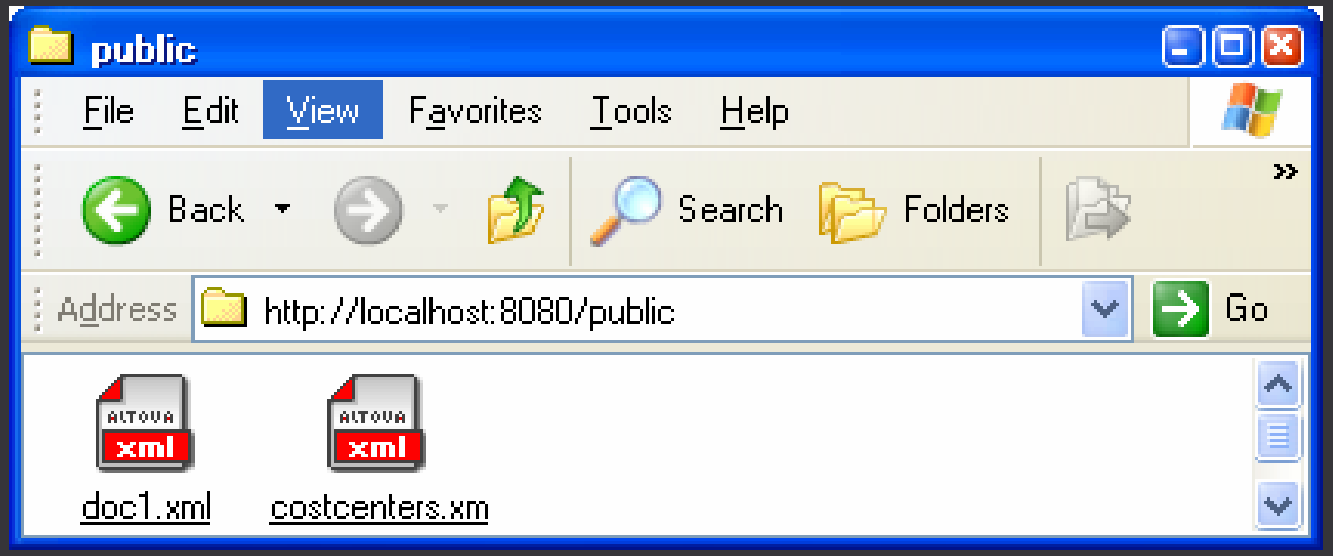
- XQuery Overview
- Oracle Database-Native XQuery
  - Architecture and implementation
  - Usage Scenarios
    - **Querying XML DB Repository**
    - Querying Relational Data
    - Querying Native XML Data
    - Querying Diverse Data Sources
  - Database XQuery Environment
- Nextance Case Study
- Conclusion

Querying the Oracle XML DB Repository

- Oracle XML DB uses the term ‘resource’ to describe content – documents, messages, metadata – stored in the Repository
- As an example, let us create two ‘resources’ in the Repository
  - A project summaries document carries summary information of the projects
  - A document that describes the cost centers
- These resources can be thought of documents managed as ‘XML files’ in Oracle XML DB
  - Oracle builds hierarchical indexes to traverse the directory-structure of these documents efficiently
  - Content-based indexes (full-text, B-Tree, XPath) can also be built

An Example (Cont'd)

- The new resources are shown as files in a WebDAV client



Using XQuery for the Example

```
SELECT XMLQuery(
'for $p in
  doc("/public/doc1.xml")/doc1/body/project
 let $c :=
  doc("/public/costcenters.xml")//costcenter
       [@costcenterno = $p/@costcenterno]/@name
 where $p/@budget > 4000
 order by $p/@projectno
 return <project Name="{$p/summary}" CC="{$c}"/>'
  RETURNING CONTENT) FROM DUAL;

<project Name="Web Service Implementation for Self
  Service Vacation Reporting  " CC="Application
  Development"></project>
```

Full-Text Searches of XML

- XQuery 1.0 does not support full-text searching of XML documents
- A related standards effort, *XQuery 1.0 and XPath 2.0 Full-Text* is still in the early stages of formulation
- Oracle adds a function *ora:contains()* to enable full-text keyword searches as part of XQuery
  - Leverages an Oracle Text index on the XML documents if one has been created
  - Otherwise, it will be evaluated functionally using in-memory indexing

An Example

```
for $p in doc("/public/doc1.xml")/doc1/body/project,
 $c in doc("/public/costcenters.xml")//costcenter
where $p/@costcenterno = $c/@costcenterno and
   ora:contains($p/summary,''Migration or "web
   service"'') > 0
order by $p/@projectno
return <project Name="{$p/summary}" CC="{$c/@name}"
</project>

<project Name="Oracle 10gR1 to 10gR2 Migration for
   Project Management Database " CC="Database
   Administration">
</project>
<project Name="Web Service Implementation for Self
   Service Vacation Reporting    " CC="Application
   Development">
</project>
```

- XQuery Overview
- Oracle Database-Native XQuery
  - Architecture and implementation
  - Usage Scenarios
    - Querying XML DB Repository
    - **Querying Relational Data**
    - Querying Native XML Data
    - Querying Diverse Data Sources
  - Database XQuery Environment
- Nextance Case Study
- Conclusion

Querying a Relational Table or View as XML Data

- Querying across relational data is common for Application-to-Application data Exchange scenarios
- SQL will be usually be more expressive and intuitive for querying relational data,
- In certain cases you may want to use XQuery
  - For instance, if you have to feed the results to an XML-consumer or a web service

Using XQuery

```
SELECT XMLQuery('
       for $i in ora:view("REGIONS"),
           $j in ora:view("COUNTRIES")
       where $i/ROW/REGION_ID = $j/ROW/REGION_ID
         and $i/ROW/REGION_NAME = "Asia"
       return $j'
     RETURNING CONTENT) AS asian_countries
FROM DUAL;

<ROW>
  <COUNTRY_ID>AU</COUNTRY_ID>
  <COUNTRY_NAME>Australia</COUNTRY_NAME>
  <REGION_ID>3</REGION_ID>
</ROW>
...
```

The XQuery Explain Plan

```
-----------------------------------------------------
| Id  | Operation                    | Name           |
-----------------------------------------------------
|   0 | SELECT STATEMENT             |                |
|   1 |  SORT AGGREGATE              |                |
|   2 |   NESTED LOOPS               |                |
|   3 |    INDEX FULL SCAN           | COUNTRY_C_ID_PK |
|*  4 |    TABLE ACCESS BY INDEX ROWID| REGIONS        |
|*  5 |     INDEX UNIQUE SCAN        | REG_ID_PK      |
|   6 |  FAST DUAL                   |                |
-----------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   4 - filter("REGION_NAME"='Asia')
   5 - access("REGION_ID"="REGION_ID")
```

- XQuery Overview
- Oracle Database-Native XQuery
  - Architecture and implementation
  - Usage Scenarios
    - Querying XML DB Repository
    - Querying Relational Data
    - **Querying Native XML Data**
    - Querying Diverse Data Sources
  - Database XQuery Environment
- Nextance Case Study
- Conclusion

Querying Native XMLType Data in Oracle XML DB

- XMLType data stored natively in Oracle XML DB can be based on an XML Schema
- For data based on XML Schema (e.g., *purchaseorder* in the *oe* sample schema),
  - Use the XMLQuery or the XMLTable function
  - Along with the PASSING clause to provide the *purchaseorder* table as the context item for the XQuery execution

An Example

```
SELECT xtab.COLUMN_VALUE
  FROM purchaseorder, XMLTable('for $i in
  /PurchaseOrder
        where $i/CostCenter eq "A10"
        and $i/User eq "SMCCAIN"
        return <A10po pono="{$i/Reference}"/>'
        PASSING OBJECT_VALUE) xtab;

<A10po pono="SMCCAIN-20021009123336151PDT"></A10po>
<A10po pono="SMCCAIN-20021009123336341PDT"></A10po>
...
```

- The pseudocolumn COLUMN_VALUE of the resulting virtual table holds a constructed element, A10po

- XQuery Overview
- Oracle Database-Native XQuery
  - Architecture and implementation
  - Usage Scenarios
    - Querying XML DB Repository
    - Querying Relational Data
    - Querying Native XML Data
    - **Querying Diverse Data Sources**
  - Database XQuery Environment
- Nextance Case Study
- Conclusion

Querying Across Data Sources

- A database may contain information that references other content stored externally (e.g., file systems, web sites)
- Some queries need to address both the internal and external content
- As XML becomes the an über model that can be used to map a variety of other models, XQuery can be used to ‘join’ them with each other

An Example

```
SELECT XMLQuery(
'for $i in ora:view("COSTCENTERS")/ROW,
     $j in $ext/doc1/body/project
where $i/COSTCENTERNO eq $j/@costcenterno
return $j'
passing xmlparse (document httpuritype
  ('http://localhost:80/public/doc1.xml').getclob(
  ))as "ext"
returning content) from dual;

<project projectno="1278" budget="1000"
  costcenterno="A01">
  <summary>Oracle 10gR1 to 10gR2 Migration for
  Project Management Database
  </summary>
</project>
...
```

Querying a RSS Feed

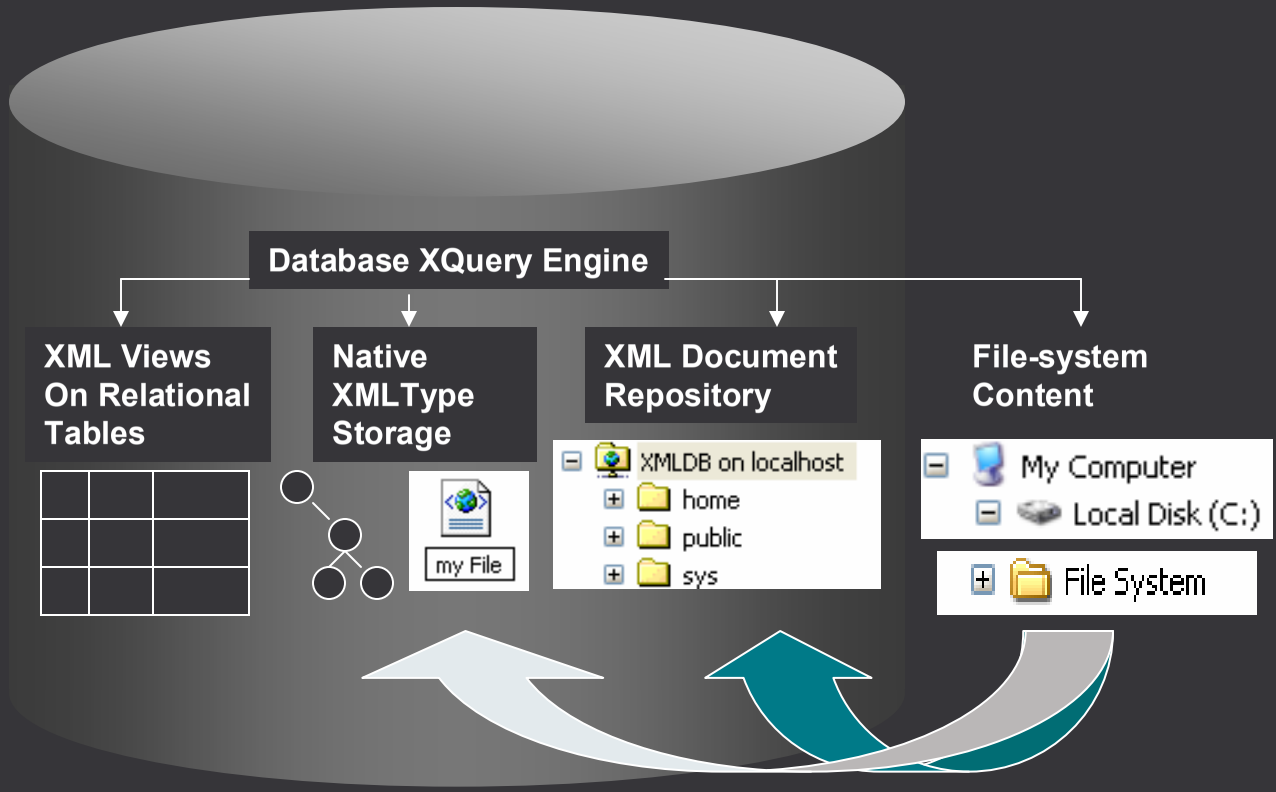
- You can query an XML RSS feed (Really Simple Syndication)
  - A format for syndicating news and the content of news-like sites like web-logs
  - Also for publishing web-site content with richer metadata than HTML
  - Any XML sources that you reach through an URL, including dynamic pages generated by applications, can be queried in this manner

An XQuery Example

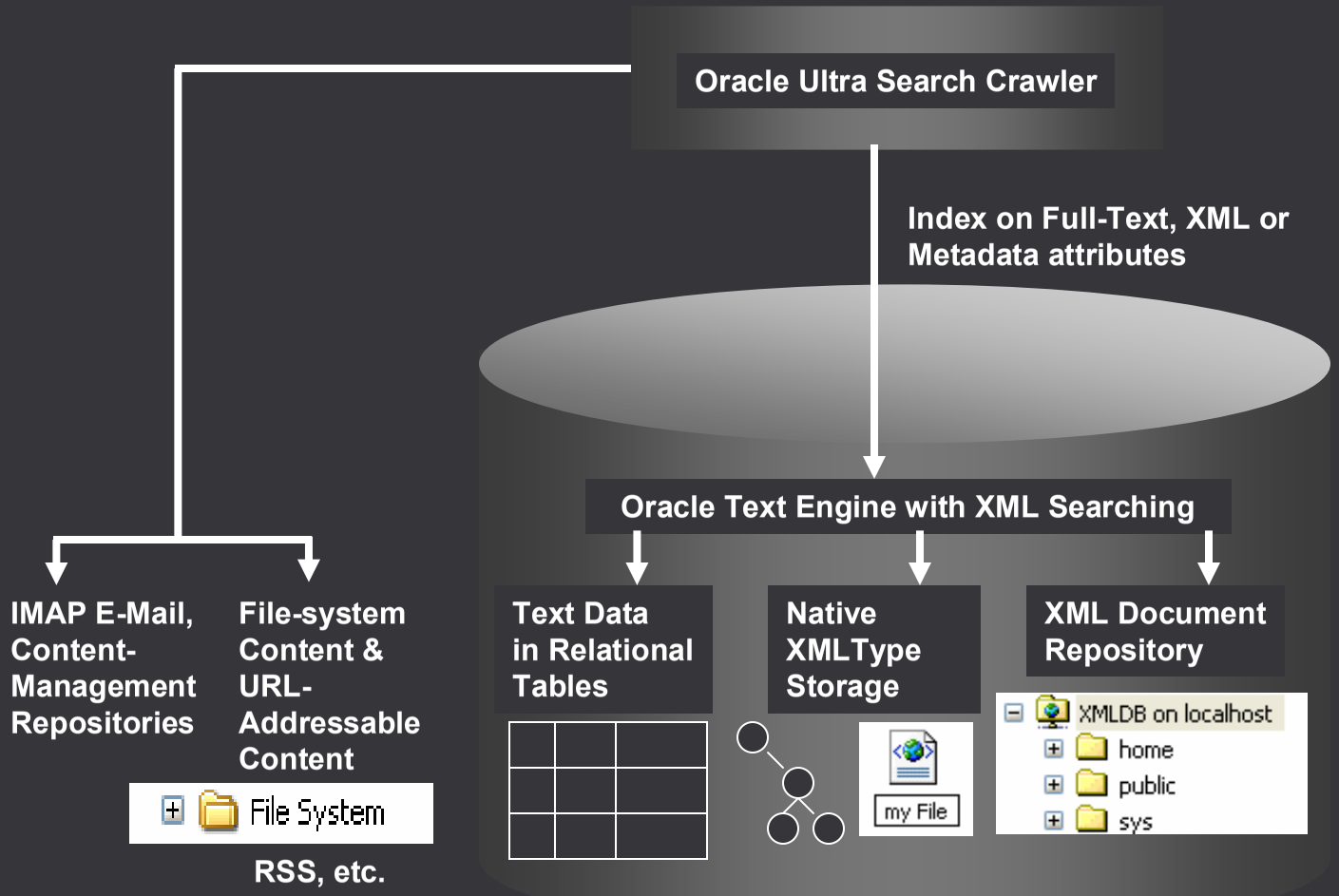
```
SELECT XMLQuery('for $i in $k//item  where contains
  ($i/title ,"Film") return $i'
  passing xmlparse( document
  httpuritype('http://www.rediff.com/rss/inrss.xml')
  .getclob()) as "k" returning content') from dual;
```

```
<item>
  <title>Filmmaker Ismail Merchant dies </title>
  <link>
  http://www.rediff.com/rss/redirect.php?url=http://
  www.rediff.com/movies/2005/may/25ismail.htm
  </link>
  <description> The filmmaker was 68.
  </description>
</item>
```

Oracle XML DB Repository



Oracle Enterprise Search



Benefits of XQuery

- There are three major advantages to using XQuery for querying XML data:
  - Conciseness and Simplicity
  - Heterogeneous Queries
  - Leveraging the XML Data Model
    - XML construction
    - XML transformation

XML Construction and Transformation

```
CONNECT HR/HR
GRANT SELECT ON LOCATIONS TO OE
CONNECT OE/OE
SELECT XMLQuery(
  'for $i in ora:view("OE", "WAREHOUSES")/ROW
    return
    <Warehouse id="{$i/WAREHOUSE_ID}">
      <Location>
        {for $j in ora:view("HR", "LOCATIONS")/ROW
         where $j/LOCATION_ID eq $i/LOCATION_ID
         return ($j/STREET_ADDRESS, $j/CITY,
                 $j/STATE_PROVINCE)}
      </Location>
    </Warehouse>'
RETURNING CONTENT) FROM DUAL;
```

XQuery vs. XSLT

- Constructing a new XML document based on a source document may be thought of either as
  - Transformation from the source to a result (XSLT), or
  - Building new document basing on a query on the source (XQuery)
- Oracle provides a native XSLT processor as part of Oracle XML DB
- XQuery will be more efficient for XML construction where optimizations can be done dynamically
  - What the type is? How the data flowing is through expressions? Where data it comes from and where is it going?
  - An example might be path expressions which have to be sorted
  - Dataflow analysis is easier to do in XQuery (using expressions) than in XSLT (using templates)

- XQuery Overview
- Oracle Database-Native XQuery
  - Architecture and implementation
  - Usage Scenarios
    - Querying XML DB Repository
    - Querying Relational Data
    - Querying Native XML Data
    - Querying Diverse Data Sources
  - **Database XQuery Environment**
- Nextance Case Study
- Conclusion

The Database XQuery Environment

- In addition to the database execution runtime, XQuery support in Oracle Database 10g Release2 is well integrated with the rest of the Oracle environment
  - SQL*Plus
  - Web Services functionality in the database
  - The XML Developer's Kit (XDK)
  - APIs (PL/SQL, JDBC, ODP.NET, etc.)

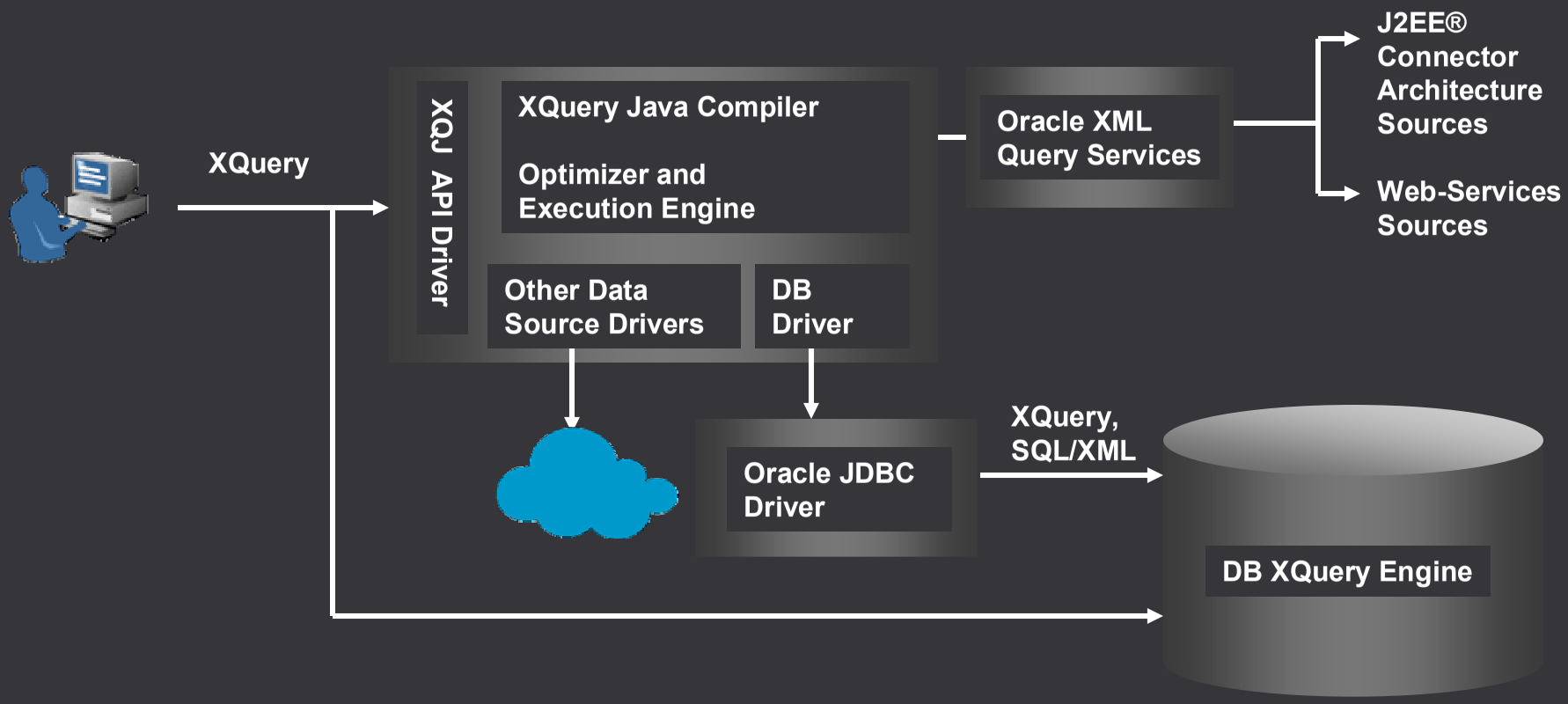
Top Level Query Support through SQL*Plus

```
SQLPLUS > xquery
for $p in doc("/public/doc1.xml")/doc1/body/project,
    $c in doc("/public/costcenters.xml")//costcenter
where $p/@costcenterno = $c/@costcenterno and
      $p/@budget > 4000
order by $p/@projectno
return <project Name="{$p/summary}“ CC="{$c/@name}"/>
SQLPLUS > /
```

XQuery APIs

- XDK
  - The XML Developer's Kit in Oracle Database 10g Release 2 supports the XPATH 2.0 and XQuery 1.0 Functions and Operators in Java
- XQuery in the Mid-Tier (XML Query Services)
  - Oracle also provides a mid-tier XQuery implementation in Java
  - Available for download from the Oracle Technology Network
  - Will be part of the Oracle Application Server 10g release
- XQJ (XQuery API for Java)
  - Currently under development as JSR 225
  - Allows invocation of and integration with functionality not part of XQuery – such a DOM, SAX, StAX, and Web Services
  - What JDBC is to SQL, XQJ is to XQuery

Query Pushdown



D E M O N S T R A T I O N

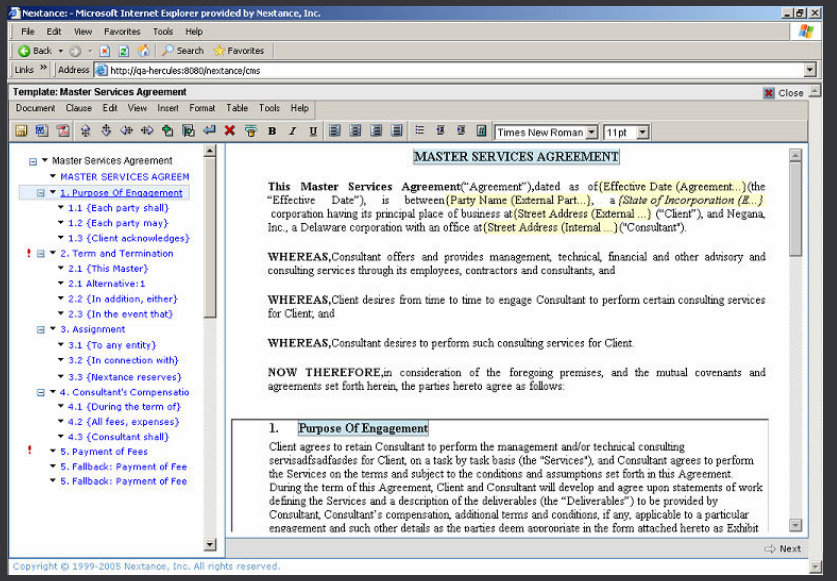
Oracle XQuery

ORACLE

- Overview
- Oracle Database-Native XQuery
- **Nextance Case Study**
- Conclusion

Nextance

- Founded 2000 to help companies optimize their Contract Lifecycle Management and Contract Lifetime Value
- Procurement, Sales & Intellectual Property Contracts
- Proven Success in the Market
  - Over 40 Customers - majority Global 1000
  - Global deployments in over 40 countries
  - Over 80,000 users
  - Optimizing over 200,000 contracts



ECM Overview

- What is ECM ?
  - Enterprise Contract Management (ECM) enables you to
    - Create, Store, Track, Manage, Analyze, Optimize, and Renew your Sales, IP, and Procurement Contracts with Customers, Partners, and Suppliers
  - Features of an ECM
    - Contract Repository
    - Searching (both structure and full text search)
    - Reporting
    - Automatic notifications and escalations based on contractual obligations or milestones
    - Approval Workflow
    - Contract Authoring and Publishing
    - Integration to ERP, CRM etc

ECM Architecture

- XML (End to End)
  - Data model, Data, Metadata, Business rules, UI Markup
- XML Schema for strong typing
- Standards-based middleware - J2EE platform
- Standards-based XML technologies - XML Schema, XSLT, SQL/XML + XQUERY, XMLDB, DOM, STAX, SAX
- Web services API for integration

Why XML ?

- Content self describing and readable by both humans and computers
- Hierarchical structure
  - Express data more naturally (e.g. contract has multiple parties, each party has a name, address and multiple contacts)
- Open standard and medium neutral structure
  - Seamless integration with ERP, CRM, Web portals, Dashboards
- XML schema
  - Rich grammar for flexible data models
  - Evolve quickly with minimum impact
  - Unified data model for data and content
- Unstructured content
  - Random shape
  - Difficult to model in a relational database
- XML based markup
  - Improves search, processing, organization and integration
- Contracts
  - Structured business terms and Unstructured content
  - Vary in structure based on the contract type and industry
  - XML a natural fit

Why XQuery ?

- Industry standard query language for searching XML documents, offers significant query functionality beyond what SQL can do
- Designed to extract data from an XML data model
- Supports XML fragments to be constructed thereby reducing bandwidth requirements
- Supports Joins across multiple XML documents

Why Oracle XML DB/XQuery ?

- Native Support for XML Type
- Enforces validity and integrity of XML data using XML Schema
- Multiple storage mechanisms (CLOB, Structured storage etc)
- XML Schema evolution which allows easy data upgrades
- XMLType views
  - Report XML with SQL Report writers
- XQuery engine in the database
  - Query optimization, scalability and performance
- Support for SQL/XML
  - Query structured data and text in a single SQL statement
  - Advanced text search with Oracle Text

D E M O N S T R A T I O N

Nextance ECM Using Oracle XQuery

ORACLE

Conclusion

- Introducing a native XQuery processing engine
  - C*ompiles* XQuery expressions into the same underlying structures as SQL queries
  - Leverages both relational- and XQuery-specific optimization
- With Oracle's XQuery support in the database and in the middle-tier, you can:
  - Search with any language –XQuery/Xpath/SQL/XSL/Full-Text
  - Search anywhere - mid tier or backend
  - Search anything - files, relational data, message, web-service etc.
  - … and search everything – structured tables, semi-structured XML, unstructured documents
  - Search any size - a native, scalable implementation
  - Search any time - an unbreakable solution, built on Oracle's robust enterprise platform.

Next Steps…

- Visit the Oracle XQuery Demo pod (A28) in the Oracle Database Demoground
- Check out the OTN Oracle XQuery page:
  http://otn.oracle.com/tech/xml/xquery
  http://otn.oracle.com/tech/xml/xmldb
- Download Oracle Database 10g Release 2:
  http://otn.oracle.com/products/database/oracle10g


QUESTIONS
ANSWERS
ORACLE


ORACLE®