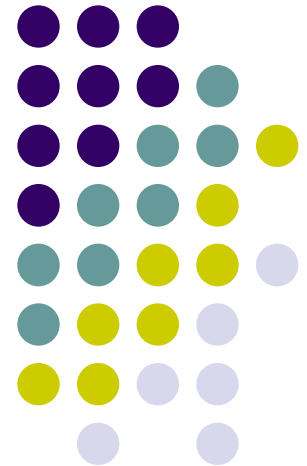


Advanced Aspects and New Trends in XML (and Related) Technologies

RNDr. Irena Holubová, Ph.D.

holubova@ksi.mff.cuni.cz

Labs 2. XML support in Oracle

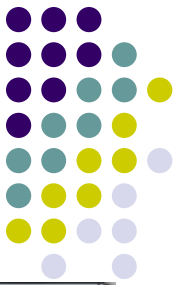




Access

- Oracle SQL Developer
<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>
- Unofficially:
<http://tirpitz.ms.mff.cuni.cz/sqldeveloper64.zip>
<http://tirpitz.ms.mff.cuni.cz/sqldeveloper.zip>
- Unpack + run
 - Asks for path to Java SDK

Access



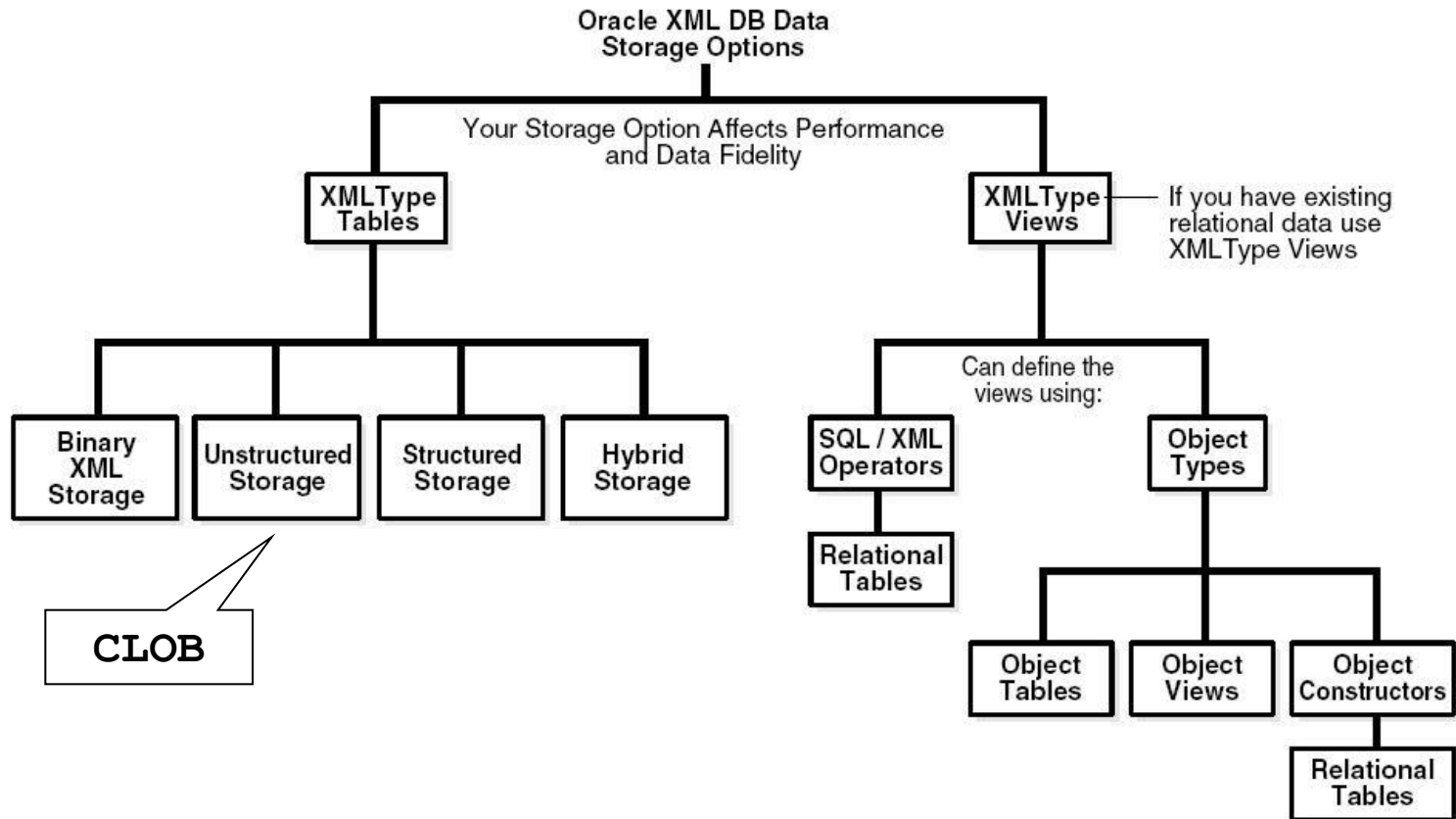
The screenshot shows the 'New / Select Database Connection' dialog box. On the left, there is a table with two columns: 'Connection Name' and 'Connection Details'. The first row is selected and contains 'tirpitz.ms.mff.cuni.cz' and 'holubova@//tirpitz...'. On the right, the configuration fields are as follows:

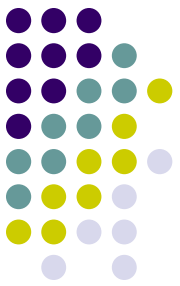
- Connection Name: tirpitz.ms.mff.cuni.cz
- Username: holubova
- Password: [masked with dots]
- Save Password
- Tab: Oracle Access
- Connection Type: Basic
- Role: default
- Hostname: tirpitz.ms.mff.cuni.cz
- Port: 1511
- SID: jedenact
- Service name: [empty]
- OS Authentication
- Kerberos Authentication
- Proxy Connection

At the bottom, there are buttons for 'Nápověda', 'Save', 'Clear', 'Test', 'Connect', and 'Zrušit'. The 'Status:' label is also present.

- Login + password: as usual
 - Change by: **ALTER USER login IDENTIFIED BY new_password**

Storage Strategies





Which storage model?



	Data-Centric		Document-Centric	
Use Case	XML schema-based data, with little variation and little structural change over time	XML schema-based data, with some embedded variable data	Variable, free-form data, with some fixed embedded structures	Variable, free-form data
Typical Data	Employee record	Employee record that includes a free-form resume	Technical article, with author, date, and title fields	Web document or book chapter
Storage Model	Object-Relational (Structured)	Hybrid	CLOB (Unstructured) or Binary XML	
Indexing	B-tree index	Index the structured and unstructured parts separately	Function-based index	XMLIndex index

Unstructured (CLOB) storage

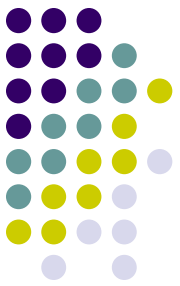


```
CREATE TABLE product (  
  id VARCHAR(10),  
  name VARCHAR2(100),  
  description XMLTYPE)  
XMLTYPE COLUMN description STORE AS CLOB;
```




```
INSERT INTO product (id, name, description)  
VALUES ('XDK', 'XML Developer's Kit',  
  XMLTYPE('<DESCRIPTION><KEYWORD>xdk</KEYWORD> is a  
set of standards-based utilities that help to build XML  
applications.</DESCRIPTION>'));
```

Structured Storage

Registration of an XSD



```
BEGIN
```

```
DBMS_XMLSCHEMA.registerSchema (  
  SCHEMAURL=>'http://xmlns.oracle.com/xml/content.xsd' ,  
  SCHEMADOC=>'<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="DESCRIPTION">  
    <xs:complexType mixed="true">  
      <xs:choice minOccurs="0" maxOccurs="unbounded">  
        <xs:element name="KEYWORD" type="xs:string"  
          maxOccurs="unbounded"/>  
      </xs:choice>  
    </xs:complexType>  
  </xs:element>  
</xs:schema>',  
  LOCAL=>TRUE,   
  GENTYPES=>TRUE,   
  GENTABLES=>FALSE) ; 
```

```
END;
```



Oracle SQL Developer : XML Schema HOLUBOVA.http://xmlns.oracle.com/xml/content.xsd@tirpitz.ms.mff.cuni.cz

File Edit View Navigate Run Versioning Tools Help

ask
Top

Connections x

- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized Views Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories
- Editions
- Application Express
- Java
- XML Schemas
 - http://xmlns.oracle.com/xml/cont
- XML DB Repository
- Scheduler
- Recycle Bin
- Other Users

Cloud Connections

tirpitz.ms.mff.cuni.cz~1 x http://xmlns.oracle.com/xml/content.xsd x

Actions...

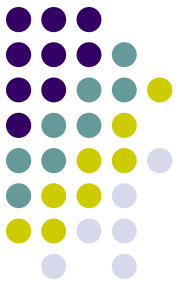
```
<?xml version="1.0" encoding="ISO-8859-2" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:oraxdb="http://xml
  <xs:element name="DESCRIPTION" oraxdb:propNumber="4081" oraxdb:global="true"
    <xs:complexType mixed="true" oraxdb:SQLType="DESCRIPTION561_T" oraxdb:SQLSc
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="KEYWORD" type="xs:string" maxOccurs="unbounded" oraxd
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Log

tirpitz.ms.mff.cuni.cz | HOLUBOVA | http://xmlns.orade.com/xml/content.xsd Editing

Structured Storage

During Registration



- SQL objects to store XMLType are generated
- Default tables are created

```
SELECT object_name, object_type FROM user_objects;
```

```
OBJECT_NAME  OBJECT_TYPE
-----
KEYWORD209_COLL  TYPE
DESCRIPTION208_T  TYPE
...
```

```
describe KEYWORD562_COLL;
```

```
"KEYWORD209_COLL" AS VARRAY(2147483647) OF VARCHAR2(4000 CHAR)
```

Structured Storage



```
CREATE TABLE product2 (  
  id VARCHAR(10),  
  name VARCHAR2(100),  
  description XMLTYPE )  
XMLType COLUMN description  
XMLSCHEMA "http://xmlns.oracle.com/xml/content.xsd"  
ELEMENT "DESCRIPTION"
```

```
INSERT INTO product2 (id, name, description)  
VALUES ('XDK', 'XML Developer's Kit',  
XMLTYPE('<DESCRIPTION><KEYWORD>xdk</KEYWORD> is a set of  
standards-based utilities that help to build XML  
applications.</DESCRIPTION>').createSchemaBasedXML  
( 'http://xmlns.oracle.com/xml/content.xsd' ));
```

Binary Storage

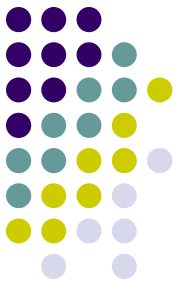


```
CREATE TABLE product3 (  
  id VARCHAR(10),  
  name VARCHAR2(100),  
  description XMLTYPE)  
XMLTYPE COLUMN description STORE AS BINARY XML;
```

```
INSERT INTO product3 (id, name, description)  
VALUES ('XDK', 'XML Developer's Kit',  
  XMLTYPE('<DESCRIPTION><KEYWORD>xdk</KEYWORD> is a  
set of standards-based utilities that help to build XML  
applications.</DESCRIPTION>'));
```

Binary Storage

Options



- CREATE TABLE **options:**

- STORE AS BINARY XML

- STORE AS BINARY XML XMLSCHEMA ...

- STORE AS BINARY XML XMLSCHEMA ...
ALLOW NONSCHEMA

- STORE AS BINARY XML ALLOW ANYSCHEMA

- STORE AS BINARY XML ALLOW ANYSCHEMA
ALLOW NONSCHEMA



Data Upload

- Trivial: string + INSERT INTO
 - see previous examples
- `HttpURIType` ←

```
utl_http.set_body_charset('EE8MSWIN1250');  
  
insert into customer  
values  
(HttpURIType('http://www.ksi.mff.cuni.cz/~mlynkova/tmp/toxgene/purc  
hasel.xml').GetXML());
```

- SQL Loader
 - Part of client
- Oracle XML DB Repository
 - Must be installed
- ...



Sample Data

- Data
 - schema.xsd
 - data.xml
- Copy-based evolution
 - schemaNew1.xsd
 - dataNew1.xml
 - trans.xsl
- In-place evolution
 - schemaNew2.xsd
 - dataNew2.xml
- Script
 - script.sql



XML Data Evolution

- **Copy-based**

- Any change → XML data must be re-validated using an XSLT script
- `DBMS_XMLSCHEMA.copyEvolve`
 - Copies current data to auxiliary tables
 - Deletes original schema S_{old} and related data
 - Registers new schema S_{new}
 - Copies the data from auxiliary tables (+ applies XSLT script on them to re-validate against S_{new})
- Indexes, triggers etc. need to be re-created manually

```
procedure copyEvolve(schemaURLs      IN XDB$STRING_LIST_T,  
                    newSchemas      IN XMLSequenceType,  
                    transforms      IN XMLSequenceType := NULL,  
                    ...) ;
```



XML Data Evolution

- **In-place**

- Backward compatibility
- `DBMS_XMLSCHEMA.inPlaceEvolve`
 - New schema S_{new} is created using changes from **diffXML** document
 - Validates S_{new}
 - Modifies data structures for storing XML data
 - Replaces S_{old} with S_{new}
- diffXML – XML document
 - Created using function `XMLDiff`

```
procedure inPlaceEvolve(schemaURL IN VARCHAR2,  
                        diffXML    IN XMLType,  
                        ...);
```


Creating diffXML



```
SELECT XMLDIFF(  
XMLTYPE('<?xml version="1.0"?>  
<bk:book xmlns:bk="http://nosuchsite.com">  
  <bk:tr>  
    <bk:td>  
      <bk:chapter>Chapter 1.</bk:chapter>  
    </bk:td>  
    <bk:td>  
      <bk:chapter>Chapter 2.</bk:chapter>  
    </bk:td>  
  </bk:tr>  
</bk:book>'),  
XMLTYPE('<?xml version="1.0"?>  
<bk:book xmlns:bk="http://nosuchsite.com">  
  <bk:tr>  
    <bk:td>  
      <bk:chapter>Chapter 1.</bk:chapter>  
    </bk:td>  
    <bk:td/>  
  </bk:tr>  
</bk:book>'))  
FROM DUAL;
```

Result



```
<xd:xdiff xsi:schemaLocation="http://xmlns.oracle.com/xdb/xdiff.xsd
    http://xmlns.oracle.com/xdb/xdiff.xsd"
    xmlns:xd="http://xmlns.oracle.com/xdb/xdiff.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:bk="http://nosuchsite.com">
  <?oracle-xmldiff operations-in-docorder="true" output-model="snapshot"
    diff-algorithm="global"?>
  <xd:delete-node xd:node-type="element"
    xd:xpath="/bk:book[1]/bk:tr[1]/bk:td[2]/bk:chapter[1]"/>
</xd:xdiff>
```

- Note: We will apply it on XML evolving schemas

diffXML Bigger Example

```
<?xml version="1.0" encoding="UTF-8"?>
<xd:xdiff xsi:schemaLocation="http://xmlns.oracle.com/xdb/xdiff.xsd
      xmlns:xd="http://xmlns.oracle.com/xdb/xdiff.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:oraxdfns_0="http://booklist.oracle.com">
  <?oracle-xmldiff operations-in-docorder="true" output-model="snapshot"
    diff-algorithm="global"?>
  <xd:delete-node xd:node-type="element" xd:xpath="/oraxdfns_0
    :booklist[1]/oraxdfns_0:book[2]"/>
  <xd:update-node xd:node-type="attribute"
    xd:parent-xpath="/oraxdfns_0:booklist[1]/oraxdfns_0:book[3]/oraxdfns_0
    :title[1]" xd:attr-local="country">
    <xd:content>US</xd:content>
  </xd:update-node>
  <xd:append-node xd:node-type="element" xd:parent-xpath="/oraxdfns_0
    :booklist[1]/oraxdfns_0:book[4]">
    <xd:content>
      <oraxdfns_0:description> This is a classic </oraxdfns_0:description>
    </xd:content>
  </xd:append-node>
  <xd:insert-node-before xd:node-type="element" xd:xpath="/oraxdfns_0
    :booklist[1]/oraxdfns_0:book[5]/oraxdfns_0:author[1]">
    <xd:content>
      <oraxdfns_0:edition>Hardcover</oraxdfns_0:edition>
    </xd:content>
  </xd:insert-node-before>
  <xd:update-node xd:node-type="text" xd:xpath="/oraxdfns_0
    :booklist[1]/oraxdfns_0:book[5]/oraxdfns_0:price[1]/text()[1]">
    <xd:content>12.99</xd:content>
  </xd:update-node>
</xd:xdiff>
```



Task

- Experiment with Oracle XML data evolution:
 - Create a bigger example of XML schema
 - Preferably, use the data from lab 1.
 - Annotate the XML schema so that the XML data are stored using structured storage optimally
 - Store the XML data and check the created data structures
 - Modify the XML schema and ensure correct:
 - In-place evolution
 - Copy-based evolution