

XML-Datenbanken

Übungsaufgaben WS 2005/06

Betrachten Sie die existierende Company-Datenbank, die unter Ihrem Account vorinstalliert ist. Speziell die Tabellen `emp` und `dept` sollen in den nachfolgenden Aufgaben als Company-Datenbank bearbeitet werden.

1. DTD-Definition

Definieren Sie mittels eines geeigneten XML-Editors (z.B. XMLSpy, Seeburger) eine Document Type Definition (DTD) `Company.dtd`, wobei die hierarchische Beziehung zwischen Abteilungen und Angestellten im Dokument darzustellen ist.

2. XML Schema-Definition

Definieren Sie in gleicher Weise ein XML Schema-Dokument `Company.xsd`, das die Speicherung der Company-Datenbank in der Oracle XDB ermöglicht.

Definieren Sie weiterhin ein XML Schema für ein Dokument `EmpDept.xsd`, das die Daten einer einzelnen Abteilung beinhaltet. Die zwei Spalten für Gehalt (`sal`) und Kommission (`comm`) sind als Subelemente eines neu einzuführenden Elements `payment` zu betrachten.

Zur Abspeicherung der XML-Dokumente in der Oracle XDB sind verschiedene Speicherungsoptionen vorzusehen:

- Speicherung der XML-Objekte als Ganzes (CLOB) in einer Tabelle (`company_lobtab`)
- Speicherung der XML-Objekte als Ganzes in einer schemabasierenden Tabelle vom Typ XMLTYPE (`companytab`)
- Speicherung der XML-Objekte in einer schemabasierenden Spalte vom Typ XMLTYPE in einer Tabelle (`emptab`)

Die Speicherung als XMLTYPE soll objektrelational umgesetzt werden. Im objektrelationalen Ansatz gibt es verschiedene Optionen für die Abspeicherung mengenwertiger Elemente (Collections). Dies ist am Beispiel der Speicherung des Dokuments `EmpDept.xsd` in der Tabelle `emptab` auszuführen:

Folgende Speicherungsvarianten sind dabei zu realisieren:

- 1) Inline VARRAYS (Standardspeicherung)
- 2) Nested Object Tables
- 3) Separate XML Type Table

Passen Sie zu diesem Zweck jeweils die Oracle XDB Annotations in der XML-Schemadefinition von `EmpDept.xsd` an. Führen Sie eine bestimmte Namenskonvention zur Benennung ein, um die Speicherungsvarianten zu unterscheiden (z.B. angehängte Nr.).

3. XML Schema-Registrierung

Registrieren Sie die XML Schema Dokumente in der Oracle XDB.

Die Struktur der XML Schemas sowie die zugehörigen XDB Annotations (z.B. Spei-

cherungsoptionen) steuern die automatische Definition der korrespondierenden Object Types in Oracle. Betrachten Sie die aus dem Schema erzeugten Oracle-Objekttypen und analysieren Sie diese.

4. Einrichten der Datenbankobjekte

Erzeugen Sie die notwendigen Datenbanktabellen, um die XML-Objekte in allen drei Speicherungsvarianten zu unterstützen, d.h.: Abspeicherung als CLOB, Abspeicherung in einer XML-Schemabasierenden Tabelle, Abspeicherung in einer XML-Schema basierenden Spalte.

5. Laden von XML-Dokumenten

Erzeugen Sie mittels eines Texteditors ein gültiges XML-Dokument `Company.xml`, das dem Schema `Company.xsd` entspricht (d.h. alle Abteilungen und ihre Mitarbeiter). Der Inhalt der Dokumentelemente soll den Daten aus den Beispieltabellen `emp` und `dept` entsprechen.

Erzeugen Sie weiterhin mittels Texteditor für jede Abteilung ein gültiges XML-Dokument `EmpDept<xx>.xml` ($xx = 10, 20, \dots$), das dem Schema `EmpDept.xsd` entspricht.

Diese Dokumente sind anschließend in den dafür vorgesehenen Tabellen abzuspeichern.

6. Generieren von XML-Dokumenten aus Tabellen

Erzeugen Sie sich ein XML-Dokument `Emp.xml` aus der Tabelle `emp` mittels Oracle XSU (`getxml`)

7. Strukturierte Speicherung von XML-Dokumenten in Tabellen

Erzeugen Sie sich eine leere Kopie `dept_from_xml` der Originaltabelle `dept`. Definieren Sie ein XML-Dokument `Dept.xml` im kanonischen XML-Format (d.h. ohne Strukturierung der Elemente), das eine Liste aller Abteilungen beinhaltet. Speichern Sie den Inhalt dieses Dokuments in der Tabelle `dept_from_xml` mittels Oracle XSU (`putxml`).

Zur Benutzung der Oracle XSU haben Sie prinzipiell drei Möglichkeiten

a) Verwenden des Command Line Utility.

Dabei genügt die Verwendung der XDK-Klassen aus dem JDeveloper. Aufruf:

```
java -classpath "E:\Oracle\JDeveloper 3.1.1.2\jdbc\
lib\oracle8.1.6\classes12.zip;E:\Oracle\JDeveloper
3.1.1.2\lib\xmlparserv2.jar;E:\Oracle\JDeveloper
3.1.1.2\lib\oraclexmlsql.jar" OracleXML putXML
- conn "jdbc:oracle:thin:@abraham.imn.htwk-
leipzig.de:1521:dblv04"
-user "username/password"
-fileName input.xml
...
```

Alternativ können auch die Oracle XDK-Klassen verwendet

b) Verwenden Sie alternativ das PL/SQL-Package `DBMS_XMLSave` zum Import der Daten aus dem Dokument.

c) Verwenden Sie alternativ die Java-Klasse `OracleXMLSave`.

8. Behandlung von geschachteltem XML

Im Unterschied zur Generierung flacher XML-Dokumente (vgl. Aufgabe 7) ist bei der Erzeugung komplex strukturierter XML-Dokumente die Verknüpfung mehrerer Tabellen

erforderlich, die eine hierarchische Beziehung repräsentieren.

Das zu erzeugende XML-Dokument `Company.xml` sollte sich an die XML Schemadefinition anlehnen, die bereits spezifiziert wurde (vgl. Aufgabe 3). In etwa so:

```
<DEPTLIST>
  <DEPT>
    <EMPLIST>
      <EMP> ..... </EMP>
    </Emplist>
  </DEPT>
  ...
</DEPTLIST>
```

Zur Erzeugung einer solchen Dokumentstruktur können zwei Lösungsvarianten gewählt werden:

- a) Erzeugen Sie eine View, die die Menge aller Abteilungen mit ihren jeweiligen Angestellten ausliest, d.h. es ist eine mengenwertige Spalte für die Liste der Angestellten einer Abteilung in der Sicht zu erzeugen. Die Syntax hierfür lautet:

```
CAST (MULTISET(query) AS collection_type_name) column_alias
```

Diese View

- b) Realisieren Sie die gleiche Aufgabe unter Verwendung von XSQL. Tip: Nutzen diesmal den SQL cursor-Operator zur Erzeugung mengenwertiger Spalten.

9. XSLT

Schreiben Sie ein XSLT-Stylesheet, das das in der Aufgabe 9 generierte XML-Dokument in eine äquivalente HTML-Darstellung transformiert.

Anfragen in XML-Datenbanken

10. Schreiben Sie eine Anfrage, die eine Liste der Angestellten ausgibt, die mehr als x verdienen (z. B. $x = 2000$). Führen Sie diese Anfrage auf den XML-Objekttabellen `emptab` und `companytab` aus. Vergleichen Sie dabei die Möglichkeiten von XPath und Query Rewrite (bei objektrelationaler Speicherung).
11. Schreiben Sie eine Anfrage, um die Angestellten auszugeben, die eine Kommission bekommen, d.h. das Element `comm` ist mit einem Wert belegt. Führen Sie diese Anfrage wiederum auf den Tabellen `companytab` und `emptab` aus.
12. Schreiben Sie eine Suchprozedur, die eine Anfrage in den XML-Objekttabellen `companytab` und `emptab` zur Ermittlung der Angestellten, die von Beruf 'CLERK' sind. Parametrisieren Sie die Anfrage (Parameter `job`). Stellen Sie die Ergebnisse in einem neuen Dokument `emp_<xx>.xml` dar ($xx = job$). Zur Verfügung stehen hierfür SQL XML Funktionen bzw. XML Type Views (ab 10g XQuery).

Referenz:

Oracle9i XML Database Developer's Guide - Oracle XML DB
Release 2 (9.2)
Part Number A96620-01