

Hinweise

1. Einrichten und Erproben des OracleXML SQL Utility (XSU)

Klassen des XDK werden standardmäßig mit Oracle 9i ausgeliefert und sind bereits auf dem Server installiert. Die Packages sind JAVA-basiert. Für die serverseitige Nutzung der PL/SQL-Packages muss für den Nutzer.

```
GRANT JAVAUSERPRIV TO USER
```

ausgeführt werden.

Package XMLGEN

Konvertiert das Ergebnis einer SQL Query in kanonisches XML. Das Package nimmt eine beliebige SQL Query als Input, konvertiert sie zu XML und gibt das Ergebnis als CLOB zurück. Das Package entspricht dem DBMX_XMLQUERY Package, außer dass es in C geschrieben ist und in den Kernel compiliert wurde. Das Package kann nur serverseitig ausgeführt werden.

http://www.psoug.org/reference/dbms_xmlgen.html

Für die clientseitige Nutzung von PL/SQL muss ein aktueller JDK installiert sein. Weiterhin müssen die Klassenpakete des XDK

- classes12.jar
- xdb.jar
- xmlparserv2.jar
- xsu12.jar

verfügbar sein. Die Pakete stehen in einer ZIP-Datei zur Verfügung. Der Ordner, der die Pakete enthält, müssen dem Klassenpfad des JDK hinzugefügt werden.

2. Directory erstellen für serverseitige Nutzung der Klassen

Um Dateien aus einem bestimmten Verzeichnis heraus in die Datenbank zu importieren, muss ein Verzeichnislink in der Datenbank erstellt werden. Dafür ist das Recht:

create any directory	zum Erstellen und
drop any directory	zum Löschen des Verzeichnislinks notwendig.

Erstellt wird der Verzeichnislink mit:

```
create directory XML as '/home/david/99I/username/<Ordnername>';
```

Das im Beispiel angegebene Verzeichnis muss zuvor auf dem Datenbankserver (Goliath) erstellt werden und es müssen auf dieses Verzeichnis leserechte für alle Benutzer erlaubt sein.

3. Benötigte Prozeduren

-- CLOB aus XML-Datei in Directory erzeugen

```
create or replace function getDocument(filename varchar2,dir varchar2)
return clob
is
xbfile bfile;
xclob clob;
begin
    xbfile := bfilename(dir,filename);
    dbms_lob.open(xbfile);

    dbms_lob.createtemporary(xclob,TRUE,dbms_lob.session);
    dbms_lob.loadfromfile(xclob,xbfile, dbms_lob.getlength(xbfile));
    dbms_lob.close(xbfile);
    return xclob;
end;
/
```

-- Funktion, die kanonisches XML einfügt

```
create or replace procedure insProc(xmlDoc IN CLOB, tableName IN VARCHAR2)
is
    insCtx DBMS_XMLSave.ctxType;
    rows number;
begin
    insCtx := DBMS_XMLSave.newContext(tableName);
-- context handle erzeugen
    rows := DBMS_XMLSave.insertXML(insCtx,xmlDoc);
-- Dokument einfügen
    DBMS_XMLSave.closeContext(insCtx);
-- schliesst den handle
end;
/
```

4. XML per PL/SQL in eine Tabelle einfügen

Die XML-Datei enthält die in die Tabelle einzufügenden Zeilen. Diese Datei muss in das in Punkt 2 erstellte Directory kopiert werden. Der Aufruf erfolgt so:

```
Exec insProc(getDocument('<file>`,`XML``,`<table>');
```

Hinweis: Das Directory muss immer groß geschrieben werden.

5. Ganze XML-Dokumente einfügen

Die Vorgehensweise zum Importieren von Dateien in eine Tabelle der Datenbank Oracle soll an einem Beispiel erklärt werden. Zuerst muss eine Tabelle erzeugt werden, die eine Spalte vom Typ XMLType enthält, in welche die Dateien gespeichert werden sollen. Die Beispieltabelle wird wie folgt angelegt:

```
CREATE TABLE XMLTABLE(id integer,doc XMLType);

create sequence doc_id increment by 1;

create or replace trigger xml_before_insert
```

```

before insert on xmltable for each row
BEGIN
    SELECT doc_id.nextval
    INTO :new.id FROM dual;
END;
/
create index doc_ix on xmltable(doc)
indextype is ctxsys.context
parameters ('section group ctxsys.auto_section_group');

```

Das Importieren aus dem in Punkt 2 erstellten Directory erfolgt mit:

```
INSERT INTO XMLTABLE(doc) VALUES (XMLTYPE(getDocument('lear.xml','XML')));
```

Hinweis: Das Directory muss immer groß geschrieben werden.

6. Clientseitiges Nutzen des XSU

Die einzufügenden Zeilen müssen in kanonischem XML vorliegen

```

<?xml version="1.0"?>
<ROWSET>
  <ROW>
    <SPALTE1>Wert1</SPALTE1>
    <SPALTE2>Wert2</SPALTE2>
  </ROW>
</ROWSET>

```

Bei den folgenden Kommando-Zeilen wird davon ausgegangen, dass die benötigten Packages im Verzeichnis c:\libs abgelegt sind.

Einfügen von Zeilen:

```

Java -classpath
"C:\libs\classes12.jar;C:\libs\xmlparserv2.jar;C:\libs\xdb.jar;C:\libs\xsu12.jar"
OracleXML putXML -conn "jdbc:oracle:thin:@abraham.imn.htwk-leipzig.de:1521:dblv04"
-user "xd0401/oracle" -filename <file> <tabelle>

```

Auslesen von Zeilen:

```

java -classpath
"c:\libs\classes12.jar;c:\libs\xmlparserv2.jar;c:\libs\xsu12.jar;c:\libs\xdb.jar"
OracleXML getXML -conn "jdbc:oracle:thin:@abraham.imn.htwk-leipzig.de:1521:dblv04"
-encoding "iso-8859-1" -user "xd0401/oracle" <query>

```

7. Installation der Tamino-Benutzer und Datenbanken für die Vorlesung

8. Testen der Taminoclients im PC-Pool (L107)

Der Taminoclient im PC-Pool ist vollständig installiert. Damit alle Tamino-Tools und auch die Dokumentation richtig funktionieren muss der Internet Explorer als Standardbrowser eingerichtet sein! Um einen Connect zur Tamino-Datenbank herzustellen, sind folgende Angaben nötig:

Server:	http://141.57.9.57/tamino	oder	http://timon.imn.htwk-leipzig.de/tamino
Database:	tamtest		
User:	Username des Sun-Pools		
Password:	Passwort des Sun-Pools		

revoke execute on xmlgen from tkrumbe;