SQL-basierte Datenbankzugriffe und XML: Eine 4-Schichten-Architektur

Christoph Kiewitt und *Thomas Müller*
Friedrich-Schiller-Universität Jena
Lehrstuhl für Datenbanken und Informationssysteme
Ernst-Abbe-Platz 2
07743 Jena
{christoph.kiewitt, mueller}@informatik.uni-jena.de

Zusammenfassung

SQL:2007, die sich momentan noch in Entwicklung befindende nächste Version der SQL-Norm, erweitert den aus SQL:2003 bekannten SQL-Datentyp „XML“. Als Instanzen dieses Datentyps sind nun auch komplette XQuery-Sequenzen erlaubt. Infolgedessen können SQL:2007-Anfrageergebnisse vollständige XQuery-Sequenzen als Attributwerte besitzen. Damit stellt sich die Frage, wie XQuery-Sequenzen, die im Anfrageergebnis enthalten sind, von Anwendungsprogrammen verarbeitet werden können. Eine Möglichkeit besteht darin, Ausschnitte der (möglicherweise sehr großen) Sequenzen ins Anwendungsprogramm zu übertragen, um diese dort lokal zu verarbeiten. In diesem Beitrag wird eine zur Unterstützung dieses Vorgehens geeignete 4-Schichten-Architektur vorgestellt. Eine umfangreiche Prototypentwicklung auf dieser Basis ist aktuell im Entstehen.

1 Einführung

In der derzeit gültigen SQL-Norm, SQL:2003 [Tü03, ISO03b], wurde durch den Basisdatentyp `XML` [ISO03a] erstmals eine Unterstützung der Verarbeitung von XML-Dokumenten in relationalen Datenbanken eingeführt. Die sich momentan in Bearbeitung befindende Nachfolgeversion der SQL-Norm, gemeinhin als SQL:2007 bezeichnet, wird diesen Datentyp bezüglich des zugrunde liegenden Datenmodells erweitern [EM04]. Anstelle des bisher benutzten XML-Infosets [W3C04] wird das XQuery-Datenmodell [W3C05b] verwendet, wodurch komplette XQuery-Sequenzen [LS04] als Attributwerte auftreten können. SQL:2007 erlaubt zudem die Validierung von XQuery-Sequenzen gegen XML-Schemata. Bei einer solchen Validierung werden die Sequenzen mit Typ-Informationen angereichert.

Ein SQL:2007-Anfrageergebnis kann Spalten des neuen XML-Datentyps besitzen. Damit enthalten die Ergebnistupel dann komplette XQuery-Sequenzen. Aus Performance- bzw. Handhabbarkeitsgründen kann es sinnvoll sein, diese XQuery-Sequenzen, oder insbesondere auch Teile davon, direkt innerhalb des Anwendungsprogramms zu verarbeiten [Mü05]. Eine solche Verarbeitung kann *lesend* oder *ändernd* erfolgen. Im Falle einer ändernden Verarbeitung soll es natürlich möglich sein, die Änderungen in die Datenbank einzubringen, indem sie auf die zugrunde liegende(n) Basistabelle(n) (rück)abgebildet werden.

Der vorliegende Beitrag stellt eine 4-Schichten-Architektur vor, welche die anwendungsprogramm-seitige Verarbeitung von Sequenzausschnitten ermöglicht. Neben dem prinzipiellen Verarbeitungsablauf (Abschnitt 2) wird diese Architektur in Abschnitt 3 beschrieben. In Abschnitt 4 schlagen wir eine Funktion vor, die gezielte Änderungen innerhalb einer XQuery-Sequenz ermöglicht. Abschnitt 5 fasst den Beitrag zusammen und gibt einen Ausblick auf künftige Forschungsschwerpunkte.

2 Verarbeitungsablauf

Das in [Mü05] motivierte Verfahren zur Verarbeitung von in SQL:2007-Anfrageergebnissen enthaltenen XQuery-Sequenzen lässt sich in folgende Verarbeitungsschritte untergliedern:

0. Ausgangspunkt der Verarbeitung ist eine SQL:2007-Tabelle (Abbildung 1) mit einer oder mehreren Spalten des *neuen* SQL-Basisdatentyps XML. Im gezeigten Beispiel enthalten die Tabellenspalten C und D XQuery-Sequenzen, A und B seien „normale“ Spalten.

1. Mit Hilfe einer SQL:2007-Anfrage wird ein Anfrageergebnis (Abbildung 2) ermittelt, welches eine oder mehrere XML-Spalten besitzt. Die Ergebnistupel enthalten als Attributwerte u. a. komplette XQuery-Sequenzen. Diese müssen nicht „1:1“ mit den XQuery-Sequenzen in der Basistabelle übereinstimmen, sondern können durch die SQL:2007-Anfrage verändert worden sein.
2. Auf dem SQL:2007-Anfrageergebnis wird „klassisch“ mittels Tupel-Cursor navigiert. Der Tupel-Cursor ist nun auf dem zu verarbeitenden Tupel positioniert, welches im Folgenden als „aktuelles Ergebnistupel“ bezeichnet wird.

3. Mittels sogenannter Sequenz-Cursor [Rab05] werden Ausschnitte der Sequenz(en) des aktuellen Ergebnistupels definiert (Abbildung 3). Die Navigation der Sequenz-Cursor basiert dabei auf der navigationsbasis-orientierten Darstellung [MR05] der Sequenzen. Diese Repräsentation zeichnet sich dadurch aus, dass jeder im getypten Wert von Element- oder Attributknoten enthaltene atomare Wert durch einen separaten Knoten dargestellt wird.

4. Der Sequenz-Ausschnitt wird ins Anwendungsprogramm übertragen.
5. Das Anwendungsprogramm arbeitet mittels spezieller Methoden auf dem Sequenz-Ausschnitt. Die Verarbeitung wird wie im Punkt 2 beschrieben fortgesetzt.

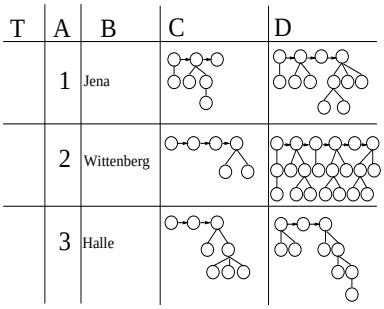

Abbildung 1: SQL:2007-Tabelle

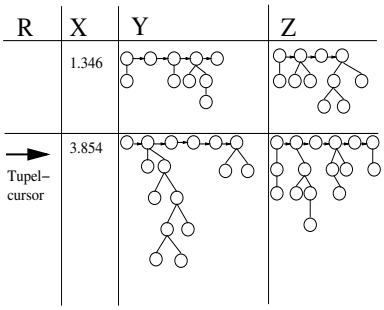

Abbildung 2: SQL:2007-Anfrageergebnis

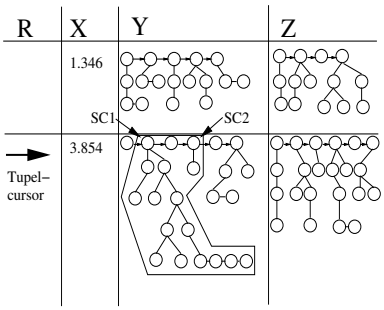

Abbildung 3: SQL:2007-Anfrageergebnis mit XQuery-Sequenzen in navigationsbasis-orientierter Darstellung sowie Sequenzcursorn und Sequenz-Ausschnitt

3 4-Schichten-Architektur

Zur Unterstützung des im vorigen Abschnitt kurz beschriebenen Verarbeitungsablaufs schlagen wir eine 4-Schichten-Architektur vor (Abbildung 4). Die unterste Schicht ist ein DBMS, welches XML-Funktionalität entsprechend SQL:2007 bereitstellt.

Die nächste Schicht besteht aus der *Navigationsbasis-Komponente* (NBK), welche die im Anfrageergebnis enthaltenen XQuery-Sequenzen in die navigationsbasis-orientierte Darstellung überführt und der nächsthöheren Schicht zur Verfügung stellt. Andererseits nimmt die Navigationsbasis-Komponente die Änderungsinformation von der Sequenz-Cursor-Verwaltung entgegen und transformiert diese in eine positionierte SQL:2007-`UPDATE`-Operation, welche dann vom SQL:2007-DBMS verarbeitet wird.

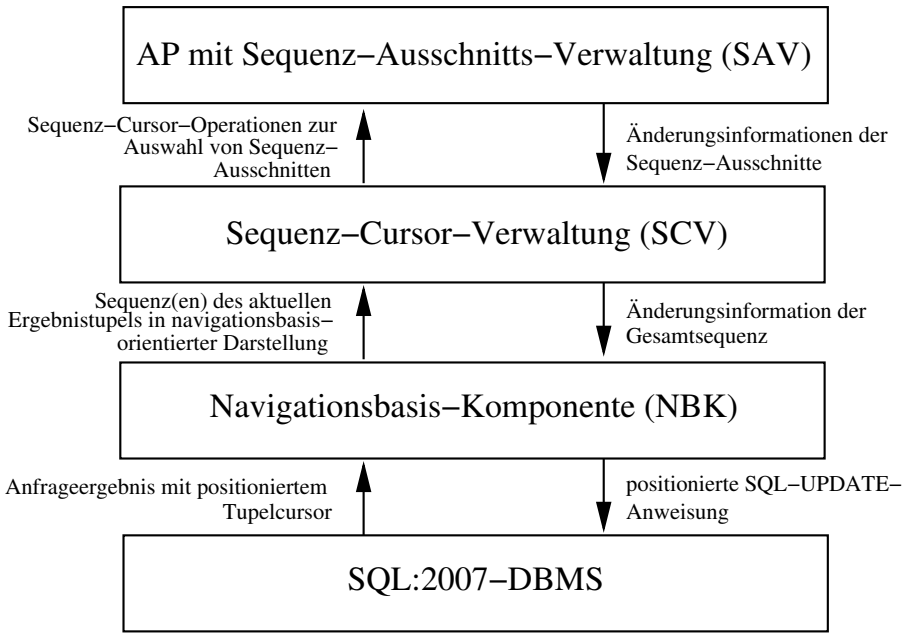

Abbildung 4: 4-Schichten-Architektur

Die *Sequenz-Cursor-Verwaltung* (SCV) ermöglicht dem Anwendungsprogramm die Auswahl eines Sequenz-Ausschnitts basierend auf der navigationsbasis-orientierten Darstellung. Die SCV nimmt hierfür vom Anwendungsprogramm Sequenz-Cursor-Operationen entgegen und verarbeitet diese. Im Anschluss wird der Sequenz-Ausschnitt an die Sequenz-Ausschnitts-Verwaltung übergeben, welche Bestandteil des Anwendungsprogramms ist. Bei Bedarf fordert die SCV von der Sequenz-Ausschnitts-Verwaltung die Änderungsinformationen der einzelnen Sequenz-Ausschnitte an und integriert diese zu einer mit der Gesamtsequenz assoziierten Änderungsinformation, welche daraufhin der NBK zur Verfügung gestellt wird.

Die oberste Schicht der Architektur ist das Anwendungsprogramm (AP), welches die *Sequenz-Ausschnitts-Verwaltung* (SAV) enthält. Das Anwendungsprogramm definiert Sequenz-Ausschnitte, um diese lokal zu verarbeiten. Die Auswahl erfolgt mit Hilfe von Sequenz-Cursor-Operationen, welche von der SCV verarbeitet werden. Die ins Anwendungsprogramm übertragenen Sequenz-Ausschnitte werden dort von der SAV verwaltet. Diese stellt dem Anwendungsprogramm geeignete Methoden zur Verfügung, um auf den Sequenz-Ausschnitten arbeiten zu können.

4 Erweitertes positioniertes SQL-Update

Wie im vorigen Abschnitt beschrieben, überführt die Navigationsbasis-Komponente die Änderungsinformation der Sequenz in eine positionierte SQL-`UPDATE`-Operation, welche von einem SQL:2007-DBMS ausgeführt wird. Ein positioniertes `UPDATE` entsprechend den bisher im Normungsentwurf vorgesehenen Möglichkeiten, würde allerdings das u. U. aufwendige Überschreiben der kompletten Sequenz (also ein „replace“) nach sich ziehen:

```
UPDATE T
SET B = neueSequenz
WHERE CURRENT OF Tupelcursor;
```

Aus Performancegründen ist es deshalb sinnvoll, dem (erweiterten) DBMS-Optimizer zu ermöglichen, anstelle des Überschreibens kompletter Sequenzen, gezielte Änderungen an diesen vorzunehmen. Hierzu wird auf der rechten Seite der SET-Klausel der positionierten `UPDATE`-Anweisung ein Ausdruck benötigt, der vom Optimizer entsprechend ausgewertet werden kann.

Hierfür schlagen wir die Funktion `XMLMANIPULATE` vor. Mit dieser können alle Eigenschaften der in XQuery-Sequenzen enthaltenen Knoten oder atomaren Werte (inklusive der Struktur der Sequenz selbst) verändert werden. Die Funktion `XMLMANIPULATE` besitzt die folgende (an die Oracle-Funktion `updateXML` [Ora03] angelehnte) Syntax:

```
XMLMANIPULATE(Spalte, erweiterter XPath-Ausdruck, Operation, Werteausdruck
                                                  [,validity check]
        {,erweiterter XPath-Ausdruck, Operation, Werteausdruck
                                                  [,validity check]})
```

Das Ergebnis der (innerhalb der positionierten SQL-`UPDATE`-Anweisung genutzten) Funktion ist die XQuery-Sequenz, die entsteht, wenn die angegebenen Änderungsoperationen auf die (mit Hilfe des Spaltennamens identifizierte) XQuery-Sequenz des aktuellen Ergebnis-Tupels angewendet werden. Der zu ändernde Teil der Sequenz wird mit einem um die Eintragsnummer erweiterten XPath-Ausdruck [W3C05a] (z. B. [4]//Tagung/Ort) beschrieben. Die angestrebte Änderung selbst wird durch die Angabe einer Operation (z. B. `replace`) und eines zu dieser Operation passenden Werteausdrucks (z. B. <Ort>Wittenberg</Ort>) spezifiziert. Der (Schema-)Typ eines Knotens lässt sich allerdings nicht durch die Anwendung von `XMLMANIPULATE` verändern. Dazu muss die XQuery-Sequenz gegen ein XML-Schema validiert werden. Der letzte (optionale) Parameter „validity check“ bewirkt die Validitätsprüfung der auszuführenden Operation, d. h. es wird geprüft, ob durch die vorzunehmenden Änderungen die Validität der Sequenz verletzt werden würde. Im Falle einer Validitätsverletzung würde die positionierte `UPDATE`-Anweisung zurückgewiesen. Wird der optionale Parameter `validity check` nicht angegeben, wird nicht geprüft, ob die vorzunehmenden Änderungen zu einer Validitätsverletzung führen. In diesem Fall werden die Typ-Informationen aller von der `UPDATE`-Operation betroffenen Knoten auf `untyped` bzw. `untypedAtomic` gesetzt, was einem Löschen der Typinformationen gleichkommt. Dies ist notwendig, da ohne Überprüfung der Validität nicht sichergestellt ist, dass die resultierende Sequenz den mit ihr assoziierten Schemata genügt. Um die Sequenz wieder mit Typ-Informationen anzureichern, muss sie mit Hilfe der SQL-Funktion `XMLVALIDATE` erneut validiert werden.

5 Zusammenfassung und Ausblick

Seit SQL:2003 werden XML-Werte von der SQL-Norm unterstützt[1]. Hierzu wurde von den Normungsgremien der SQL-Basisdatentyp XML eingeführt, welcher (in SQL:2003) auf dem XML-Infoset basiert. Die noch zu verabschiedende nächste Version der SQL-Norm — voraussichtlich als SQL:2007 bezeichnet — erweitert den Datentyp XML, indem sie ihm das XQuery-Datenmodell zugrunde legt. Somit können Resultate von SQL-Anfragen nunmehr komplette XQuery-Sequenzen enthalten. In diesem Beitrag haben wir eine 4-Schichten-Architektur vorgestellt, welche die anwendungsprogramm-seitige Verarbeitung von in Anfrageergebnissen enthaltenen XQuery-Sequenzen ermöglicht.

Hierbei werden Ausschnitte der im Anfrageergebnis enthaltenen Sequenzen ins Anwendungsprogramm übertragen und dort lokal verarbeitet. Lokal vorgenommene Änderungen werden in die der Anfrage zugrunde liegende(n) Basistabelle(n) eingebracht. Hierzu werden die Änderungen in positionierte SQL-`UPDATE`-Operationen überführt, welche dann vom DBMS verarbeitet werden. Um dabei ein komplettes Überschreiben der Sequenzen zu vermeiden, schlagen wir die SQL-Funktion `XMLMANIPULATE` vor, welche es erlaubt, gezielt Änderungen an Sequenzen vorzunehmen.

Die einzelnen Schichten der vorgestellten 4-Schichten-Architektur sollten in folgenden Forschungsarbeiten weiter konkretisiert werden. Zudem ist geplant, die 4-Schichten-Architektur prototypisch zu implementieren und anschließend zu evaluieren. Da derzeit kein SQL:2007-fähiges

[1] In verschiedenen DBMS-Produkten gibt es ebenfalls seit einigen Jahren XML-Unterstützung, hier liegen jedoch Abweichungen von der Norm vor.

DBMS verfügbar ist, muss die unterste Schicht geeignet simuliert werden. Mit der Implementierung dieser und anderer Schichten wurde bereits begonnen [Kie06, Jat06].

Literatur

[EM04] A. Eisenberg und J. Melton. Advancements in SQL/XML. *SIGMOD Record, 33(3):79-86*, 2004.

[ISO03a] World Wide Web Consortium. *Information technology - Database languages - SQL - Part14: XML-Related Specifications (SQL/XML)*, Dezember 2003. ISO/IEC 9075-14:2003, International Standard (IS).

[ISO03b] World Wide Web Consortium. *Information technology - Database languages - SQL - Part2: Foundation (SQL/Foundation)*, Dezember 2003. ISO/IEC 9075-2:2003, International Standard (IS).

[Jat06] M. Jatho. Konzeption und Implementierung einer Navigationsbasiskomponente zur Unterstützung einer datenbankorientierten Verarbeitung von XML-Werten. Diplomarbeit, Friedrich-Schiller-Universität Jena, 2006. In Vorbereitung.

[Kie06] C. Kiewitt. Konzeption und Implementierung einer Sequenzsimulationskomponente zur Unterstützung einer datenbankorientierten Verarbeitung von XML-Werten. Diplomarbeit, Friedrich-Schiller-Universität Jena, 2006. In Vorbereitung.

[LS04] W. Lehner und H. Schöning. *XQuery: Grundlagen und fortgeschrittene Methoden*. dpunkt.verlag, Heidelberg, 2004.

[MR05] T. Müller und G. Rabinovitch. Das Navigationsbasis-Modell zur Unterstützung der cursorbasierten Übergabe von XQuery-Sequenz-Ausschnitten zwischen Datenbanksystem und Anwendungsprogramm. *Berliner XML-Tage - Tagungsband, Humboldt-Universität zu Berlin und Freie Universität Berlin, Seiten 135-146, Berlin*, September 2005.

[Mü05] T. Müller. SQL-basierte Datenbankzugriffe und XML: Verarbeitung von Anfrageergebnissen in Anwendungsprogrammen. *Tagungsband zum 17. Workshop "Grundlagen von Datenbanken", Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, Seiten 94-98, Wörlitz*, Mai 2005.

[Ora03] Oracle Corporation, Redwood City, CA. *Oracle XML DB Developer's Guide 10g Release 1 (10.1)*, Dezember 2003.

[Rab05] G. Rabinovitch. Sequenzcursor-Konzept zur Übergabe von XML-Werten zwischen Datenbanksystem und Anwendungsprogramm. Diplomarbeit, Friedrich-Schiller-Universität Jena, Juli 2005.

[Tü03] C. Türker. *SQL:1999 & SQL:2003*. dpunkt.verlag, Heidelberg, 2003.

[W3C04] World Wide Web Consortium. *XML Information Set (Second Edition)*, Februar 2004. W3C Recommendation.

[W3C05a] World Wide Web Consortium. *XML Path Language (XPath) 2.0*, November 2005. W3C Recommendation.

[W3C05b] World Wide Web Consortium. *XQuery 1.0 and XPath 2.0 Data Model*, November 2005. W3C Candidate Recommendation.