

# 1 Einleitung

**XML und Datenbanken.** Der Titel des Buchs lautet »XML und Datenbanken«.

XML steht für Extensible Markup Language und bezeichnet ein Dokumentenformat zur Darstellung semistrukturierter Daten. XML genießt zweifellos gegenwärtig eine große Popularität. Die Anwendungen sind vielfältig und reichen vom Zwischenformat zur Datenrepräsentation oder zum Datenaustausch bis hin zur Markup-Sprache für Volltextdokumente. Datenbanken hingegen sind eine seit Jahrzehnten bewährte Technologie zur Speicherung und Anfrage von großen strukturierten Datenbeständen.

Auf den ersten Blick lassen sich die Unterschiede beider Gebiete erkennen. XML und Datenbanken sind zwei sehr verschiedene Konzepte, die in unterschiedlichen Bereichen der Informatik entwickelt und angewendet werden.

Datenbanken werden eingesetzt, um große Datenmengen sicher zu speichern und effizient anfragen zu können. Sie dienen in erster Linie der Speicherung von strukturierten Daten. XML ermöglicht neben der Darstellung strukturierter Daten auch die Repräsentation von semistrukturierten Daten, also Daten, deren Struktur unregelmäßig ist, wechseln kann oder gar nicht explizit vorhanden ist. Auch Dokumente, die Volltextinformationen beinhalten, können mit XML-Syntax dargestellt werden.

Wie sieht nun der Zusammenhang zwischen beiden Gebieten aus? Die Forderung nach konsistenter und zuverlässiger Speicherung von XML-Dokumenten suggeriert den Einsatz von Datenbanken oder die Verwendung und Anpassung von Datenbanktechnologien. Wir werden im Verlauf der weiteren Kapitel betrachten, welche Technologien aus dem Gebiet Datenbanken für XML adaptiert wurden und inwiefern sie angepasst und erweitert wurden, um für XML geeignet zu sein. XML ist ein relativ neues Format zur Darstellung und Speicherung von Daten, während Datenbanken eine schon seit sehr langer Zeit bestehende und vielfältig eingesetzte Technologie sind. Es bietet sich bei der Entwicklung von Technologien für XML an, auch die verfügbaren Technologien im Bereich Datenbanken anzusehen und teilweise in angepasster Form zum Einsatz zu bringen. Dafür können bei der Speicherung und Indizierung von XML-Dokumenten Datenbanktechniken eingesetzt oder

zugrunde gelegt werden. Im Bereich Anforderungen an XML-Anfragesprachen und XML-Anfragesprachen selbst sind ebenfalls viele Parallelen zu Datenbank-Anfragesprachen sichtbar. Auch bei der Entwicklung von Updatesprachen sind die Updatefunktionen der Datenbank-Anfragesprachen eine nützliche Basis. Wir werden im weiteren Verlauf daher Fragen untersuchen, wie

- ❑ Welche Speicherungstechniken existieren für XML-Dokumente?
- ❑ Wie erfolgt eine geeignete Indizierung der gespeicherten XML-Dokumente?
- ❑ Welche Anfragesprachen und Updatemöglichkeiten für XML-Dokumente gibt es?
- ❑ Wie kann die Abbildung von XML-Dokumenten auf Datenbankinformationen erfolgen?

**Datenbanken und XML.** Drehen wir den Buchtitel doch einmal um, dann erhalten wir: »Datenbanken und XML« — und damit die zweite Sichtweise auf diese Problematik. Hierbei werden wir beleuchten, wie XML die klassische Datenbankwelt ergänzt.

Die Anforderungen an Datenbanksysteme steigen und es wird bedingt durch verschiedene Anwendungen notwendig, neue Arten von Informationen zu speichern. Über strukturierte Daten hinaus, die zum Beispiel für technische und ökonomische Anwendungen gespeichert werden, besteht der Wunsch, weitere Informationsarten, z.B. Multimediatypen (Bild, Audio, Video und Volltextdaten), innerhalb von Datenbanksystemen zu speichern. Mit der ExtenderTechnologie wird in verschiedenen Datenbanksystemen versucht, die Speicherung neuer Datentypen zu realisieren und Anfragen an die neuen Datentypen umzusetzen. Die Anfragen an Multimediatypen sind in den SQL99-Standard eingegangen. XML-Dokumente stellen ebenfalls einen neuen Datentyp dar, der innerhalb von Datenbanken gespeichert, verwaltet und angefragt werden soll.

XML ermöglicht die Speicherung von semistrukturierten Informationen. Durch das eingesetzte Markup enthalten XML-Dokumente Informationen über die Struktur, die in Anfragen einbezogen werden sollen. Man erreicht durch die Speicherung von XML-Dokumenten innerhalb von Datenbanksystemen, dass Datenbanken um neue, irreguläre Anteile erweitert werden. Daraus ergeben sich jetzt Fragen wie

- ❑ Welche neuen Anforderungen stellt XML an Datenbanksysteme?
- ❑ Welche Erweiterungen haben Datenbanksysteme, um XML-Dokumente zu speichern und zu verarbeiten?

Ein weiterer Aspekt ist, dass die Daten aus Datenbanken für andere Anwendungen verfügbar gemacht werden sollen. XML bietet sich dazu

als Zwischen- und Austauschformat an. Die Transformation von Daten aus Datenbanken in XML-Syntax ist also eine weitere relevante Aufgabe, die sich durch die folgende Frage ausdrücken lässt:

- Wie können XML-Dokumente aus den Inhalten von Datenbanken generiert werden?

Im Verlauf des Buchs werden wir uns diesen und weiteren Fragen widmen.

## 1.1 Aufbau des Buchs

Im Folgenden sollen die einzelnen Kapitel des Buchs kurz vorgestellt werden.

**Kapitel 2: Anwendungsszenarien** Kapitel 2 stellt einige typische Anwendungen vor, zeigt dabei bestehende Teilaufgaben und gibt an, in welchen nachfolgenden Kapiteln des Buchs eine genaue Darstellung von Lösungen dafür erfolgt.

**Kapitel 3: Grundlagen — XML** In Kapitel 3 wird eine kurze XML-Einführung gegeben. Dabei werden Begriffe wie Element, Attribut, Entity, usw. erklärt und anhand vieler Beispiele erläutert. DTDs (Document Type Definitions) werden hier ebenfalls eingeführt.

**Kapitel 4: XML-Prozessoren** Der Zugriff auf XML-Dokumente wird durch so genannte XML-Prozessoren unterstützt. Generell sind zwei verschiedene Varianten von Parsern verfügbar. Sie werden in diesem Kapitel vorgestellt.

**Kapitel 5: Datenrepräsentation von XML** In Kapitel 5 wird dargestellt, wie sich aus Datenbankinhalten XML-Dokumente generieren lassen. Dabei können aus vollständigen Datenbanken XML-Dokumente mit feststehender Syntax generiert werden. Aus Datenbankausschnitten, die durch Anfragen oder Sichten spezifiziert werden, lassen sich XML-Dokumente mit fester Syntax erzeugen, und es können variable Methoden eingesetzt werden, die aus Ausschnitten von Datenbanken spezielle XML-Dokumente mit einer vom Anwender festgelegten Syntax generieren. Alle diese Verfahren werden ausführlich dargestellt. Verfügbare Implementierungen in kommerziellen Datenbanksystemen und frei verfügbare Tools werden am Ende des Kapitels beschrieben.

**Kapitel 6: XML-Datenbanken — Architekturmodelle** In diesem Kapitel wird zunächst dargestellt, wie XML-Dokumente klassifiziert werden können. Auf der Klassifikation aufbauend werden verschiedene Verfahren für die konzeptuelle, die logische und die physische Ebene der Speicherung von XML-Dokumenten beschrieben. Die dabei eingesetzten Verfahren sind teilweise an Verfahren aus den Bereichen Datenbanken und Dokumentverarbeitung angelehnt, teilweise mit diesen auch identisch. Zusätzlich haben sich neue Verfahren etabliert, die speziell auf XML-Anwendungen zugeschnitten sind. Das Kapitel dient als Systematisierung der Ansätze, XML-Dokumente zu speichern. Die hier eingeführten Verfahren werden in den folgenden drei Kapiteln (7–11) ausführlicher dargestellt.

**Kapitel 7: Schemabeschreibung und konzeptueller Entwurf von Schemata** Dieses Kapitel ist zweigeteilt: Zum einen werden hier zwei Methoden vorgestellt, mit denen das Schema von XML-Dokumenten dargestellt werden kann — Document Type Definition und XML Schema. Im zweiten Teil des Kapitels wird gezeigt, wie eine konzeptuelle Modellierung der Schemabeschreibungen erfolgen kann. Es besteht die Möglichkeit, das Entity-Relationship-Modell, das Object Role Model oder die Unified Modeling Language (UML) zu adaptieren. Weiterhin kann die Struktur von XML-Dokumenten und Schemabeschreibung graphisch visualisiert werden.

**Kapitel 8: XML-Datenbanken — Speicherungstechniken** Die logische und physische Speicherung von XML-Dokumenten ist Inhalt dieses Kapitels. Dabei werden drei grundsätzliche Verfahren dargestellt, die die Abbildung auf interne Strukturen beschreiben. Die XML-Dokumente können unverändert gespeichert und indiziert werden. Es ist möglich, die Graphenstruktur der XML-Dokumente als Basis für die Speicherung zu verwenden. Die dritte Variante ist die strukturierte Speicherung von XML-Dokumenten in Datenbanken. Es wird ebenfalls vorgestellt, welche Verfahren für welche Arten von Dokumenten geeignet sind. Existierende Implementierungen in kommerziellen Systemen und frei verfügbare Tools werden zum Abschluss des Kapitels beschrieben.

**Kapitel 9: Speicherorganisation und Indexstrukturen** Nachdem im vorherigen Kapitel mögliche Speicherungs- und Indizierungsvarianten für XML-Dokumente dargestellt wurden, erfolgt in Kapitel 9 die Darstellung, wie die Speicherungsverfahren genau aussehen. Dazu werden allgemeine Indexstrukturen vorgestellt und Möglichkeiten zur Speicherung eines XML-Indexes angegeben.

**Kapitel 10: XML-Anfragesprachen** Ausgehend von generellen Anforderungen an Anfragesprachen werden diese für spezielle XML-Anfragesprachen dargelegt. Viele Vorschläge und Realisierungen stützen sich auf XPath 1.0, das als eine Anfragemöglichkeit an XML-Dokumente betrachtet wird. Mit XQuery wird ausführlich der absehbare Standard im Bereich Anfragesprachen beschrieben. Eine Übersicht zu weiteren Sprachen sowie verfügbaren Implementierungen beschließt das Kapitel.

**Kapitel 11: Systeme** In vielen Kapiteln des Buchs werden bestimmte Verfahren und Techniken vorgestellt und am Ende findet sich eine Aufzählung, in welchen frei verfügbaren Tools und kommerziellen Systemen diese Techniken umgesetzt sind. In diesem Kapitel jedoch wird die Betrachtung von der anderen Seite vorgenommen. Es werden verschiedene kommerzielle Systeme zum Entwurf, zur Verwaltung und zur Speicherung von XML-Dokumenten vorgestellt.

## 1.2 Beispiel, Konventionen

Im weiteren Verlauf des Buchs werden wir vieles anhand von Beispielen erläutern. Dabei werden wir überwiegend mit Anwendungen aus dem Tourismusbereich arbeiten. Diese erscheinen uns für das Thema geeignet, da Tourismusinformationen einerseits typische XML-Anwendungen für die Darstellung im World Wide Web sind, die auch vielfach multimediale Elemente enthalten. Andererseits existieren im Tourismusbereich Hoteldatenbanken für Buchungssysteme, in denen ähnliche Informationen stehen. Der Einsatz von XML- und Datenbank-Anwendungen ist also typisch für diese Branche. Transformationen zwischen XML-Dokumenten und Datenbankinhalten lassen sich anhand der Beispielanwendung ebenfalls gut beschreiben.

Ein weiterer Grund für die Auswahl des Anwendungsbeispiels ist, dass wir in Projekte involviert waren, die sich mit diesem Thema beschäftigten. Das ist in unserem Bundesland naheliegend, in dem der Tourismus der größte Wirtschaftszweig ist.

Alle beschriebenen Beispiele (Hotels/Museen/Veranstaltungen) kennen wir persönlich, die Auswahl wurde jedoch aus »XML-Sicht« getroffen und stellt somit keine Wertung dar. Wir profitieren auch nicht von der Verwendung der Beispiele. :-)