XML und Datenbanken

(Winter 2017/18)

Prof. Dr. Stefan Brass

Institut für Informatik

Geplante Themen

- XML Syntax (Kurzeinführung / Wiederholung)
- XML Schema
- XPath / XQuery Data Model (XDM), XML Infoset
- XML Parser Bibliotheken in Java: DOM, SAX (kurz)
- XPath
- XQuery
- XSLT (kurz)
- XML-Unterstützung in SQL (Oracle, DB2)

  Falls noch Zeit (unwahrscheinlich): Speicherstrukturen für XML.
  Ggf. mehr zum semantischen Web: RDF, SPARQL

Voraussetzungen

Dies ist eine fortgeschrittene DB-Vorlesung (für Master-Studenten). Vorausgesetzt werden:

- Relationales Modell, ER-Diagramme
- SQL (komplexere Anfragen, `CREATE TABLE`)
- Programmierkenntnisse, Syntaxdiagramme
- (mehr oder weniger:) Englische Sprachkenntnisse

Nicht (unbedingt) vorausgesetzt: XML

- Wer hat "Grundlagen des WWW" gehört?
- Wer kennt XML? Wer kennt HTML?

Semistrukturierte Daten (1)

- Relationale Daten gelten als stark strukturiert:
  - ◇ Das Schema ist DBMS und Nutzern bekannt, sehr stabil (ändert sich nur minimal).
  - ◇ Die einzelnen Tabelleneinträge sind atomar, Auswertungen ohne manuelle Hilfe möglich.
- Texte (auch Bilder etc.) gelten als unstrukturiert:
  - ◇ DBMS-Sicht: nur Folge von Zeichen/Worten.
  - ◇ Die inhaltlich interessanten Strukturen sind dem DBMS nicht bekannt → Keine Hilfe bei Suche.

    Z.B. Warenbezeichnungen/zugehörige Preise, falls Katalog.

Semistrukturierte Daten (2)

- Mit dem Web und XML sind semistrukturierte Daten aufgekommen:
  - ◇ Zum Teil sind inhaltlich interessante Strukturen mit Tags markiert, zum Teil einfach Text.
  - ◇ Mit den Tags wird häufig sehr frei umgegangen, die Strukturen sind unregelmäßig.
  - ◇ Die Tags (Element-Typen) / das Schema sind nicht unbedingt vorab bekannt.
  - ◇ Das Schema ist ständiger Änderung unterworfen.

    Daten entstehen häufig durch Integration autonomer Quellen.

Semistrukturierte Daten (3)

- In Anfragen an relationale Datenbanken kann man sich nur auf bekannte Spalten fester Tabellen beziehen.

  Bei Bedarf kann man das Data Dictionary abfragen, um sich zuerst die Schema-Information zu beschaffen. Dies ist aber ein getrennter Schritt: Man braucht erst die Ergebnisse dieser Abfrage, um die Abfrage an die eigentliche Datenbank zu formulieren.

- Bei XML-Daten macht es dagegen Sinn, dass man sich den Inhalt von beliebigen Tags, die auf "name" enden, an beliebiger Stelle im Dokument anzeigen lassen möchte.

  Mit einem intelligenteren Texteditor könnte man auch danach suchen.

Motivation (1)

- Es geht viel schneller, Daten im XML-Format zu erfassen, als eine relationale Datenbank anzulegen.

  Solange die Datensammlung klein ist, reicht ein Texteditor. Trotzdem können die Daten so strukturiert sein, dass man mit XQuery alles an Anfragen/Auswertungen berechnen könnte, was auch mit einem relationalen DBMS möglich wäre. Das Risiko ist allerdings, dass die Daten im Laufe der Zeit immer schlechter strukturiert werden (wenn man nicht bewusst eine DTD/ein Schema entworfen hat und die Einhaltung erzwingt).

- XML ist ein wichtiges Daten-Austauschformat.

  Selbst wenn man seine Unternehmens-Daten in einer relationalen DB hält, wird man in der Kommunikation mit Geschäftspartnern etc. XML benutzen. Es ist dann nützlich, wenn man manche Auswertungen auch direkt auf den XML-Dateien durchführen kann.

Motivation (2)

- XML eignet sich auch zur langfristigen Aufbewahrung (Archivierung) von Daten.

  XML ist ein firmenunabhängiger Standard, der allgemein anerkannt, verbreitet, und sehr stabil ist. Man kann davon ausgehen, dass es auch in 10–20 Jahren noch funktionierende Software geben wird, die heute erstellte XML-Dateien lesen kann. Das wird auch dadurch unterstützt, dass viele XML-Software "Open Source" ist, man sie also ggf. auf einem dann aktuellen Betriebssystem neu compilieren könnte, bzw. erforderlichenfalls anpassen. Die Struktur von XML-Daten ist auch relativ einfach. Wichtig ist natürlich, dass die konkrete Anwendung von XML gut dokumentiert ist (Bedeutung der verwendeten Tags etc.).

  Im Gegensatz dazu sind die Dateien von Datenbanksystemen immer systemspezifisch, und können sich von Version zu Version ändern. Datenbank-Software ist auch sehr komplex.

Motivation (3)

- XML unterstützt komplex strukturierte Objekte.

  Am relationalen Modell wird kritisiert, dass man die Objekte zur Speicherung in einfache Tupel zerlegen muss.

- Manche Leute meinen, XML sei "die Zukunft" auch im DB-Bereich: XQuery wird als das "SQL des 21. Jahrhunderts" bezeichnet.

  Ich halte das für stark übertrieben. Obwohl man feststellen muss, dass vieles von der schönen Einfachheit, die das relationale Modell gebracht hat, heute kaputt gemacht wird: Alles wird immer komplexer (meiner persönlichen Meinung nach häufig ohne rechte Not).

- Es gibt für XML sehr viele freie Werkzeuge.
- Interessante neue Forschungsprobleme.

Zeit und Ort

Vorlesung (2 SWS):

- Donnerstags, 10^{15}–11^{45}, Raum 3.28.

  Eine kurze Pause (5 Minuten) nach 45 Minuten gilt als förderlich. Der Raum wurde gewählt, weil er die Technik für eine automatische Video-Aufnahme hat.

Übung:

- Donnerstags, 12^{15}–13^{45}, Raum 3.04.
- Teilweise im PC-Pool (Raum 3.32).

  Universitäts-Accounts müssen dafür freigeschaltet werden.

Zeitliche Belastung

- Diese Vorlesung hat 5 Leistungspunkte.

  Auch "credit points" genannt.

- Entspricht 150 Stunden studentischer Arbeitszeit:

Lernform	SWS	Stunden
Vorlesung	2	30
Selbststudium	0	60/45
Tafelübung	1	15
Praktische Übung (z.T. HA)	1	15
Hausaufgaben	0	30/45

Aufteilung zwischen Tafelübung und praktischer Übung ist variabel.

Prüfung (1)

Studienleistung:

- Bedingung für erfolgreichen Abschluss des Moduls.
- Mindestens 50% der Übungspunkte.

  Hausaufgaben müssen einzeln bearbeitet werden. Zu ähnliche oder offensichtlich schlechte Lösungen zählen als nicht bearbeitet.

- Voraussichtlich kurzer Seminarvortrag (10–15min).
- Regelmäßige und aktive Mitarbeit in den Übungen.

  Höchstens drei Mal fehlen oder mit Übungsleiter besprechen. Man muss Hausaufgaben in der Übung an der Tafel präsentieren, dabei auch Fragen zum Umfeld beantworten. Ggf. werden die Hausaufgabenpunkte wieder aberkannt. Präsenzaufgaben (auch in Vorlesung) bearbeiten, dafür kann es auch Übungspunkte geben. Mitdiskutieren.

Prüfung (2)

Modulleistung:

- Klausur: 01.03.2018, 10^{00}–12^{00} (Vorschlag).

  Der Termin kann sich eventuell noch ändern. Bitte achten Sie auf Ankündigungen auf der Webseite. Eventuell mündliche Prüfung.

- Zweiter Termin: mündl. Prüfung, 19.-23.03.2018

  Falls Klausur nicht bestanden: ggf. auch im April.

- Bei der Klausur sind drei DIN A4-Blätter mit Notizen erlaubt, keine Bücher, Aktenordner, Rechner.

  Praktische Anwendung, Verstehen, wenig Auswendiglernen.
  Z.B. XML DTD und Datendatei für gegebene relationale DB, XML Schema (ggf. nur Teilstück), Anfragen in XPath und XQuery. Eventuell XDM Datenstruktur zeichnen. Kurzer Aufsatz zu Frage.

Ansprechpartner (1)

Dozent: Prof. Dr. Stefan Brass

- Email: `brass@informatik.uni-halle.de`

  Betreff-Zeile sollte mit `[xml17]` beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Dienstags, 12^{00}–13^{00}
- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.
- Oracle8 Certified Database Administrator.
- IBM Certified Advanced DBA (DB2 UDB 8.1).

Ansprechpartner (2)

Weitere Mitarbeiter der Gruppe:

- PD Dr. Alexander Hinneburg

  Raum 314, Telefon: 0345/55-24732, Email: `hinneburg@...`

- (weitere Mitarbeiterstelle ist unbesetzt)

  Das erklärt, warum ich die Übung selbst halte mit gewissen Einschränkungen bei der Korrektur der Hausaufgaben.

Sekretärin: Ramona Vahrenhold

- Büro: Von-Seckendorff-Platz 1, Raum 324
- Telefon: 0345/55-24750, Fax: 0345/55-27333
- Email: `vahrenho@informatik.uni-halle.de`

WWW-Seite

`http://www.informatik.uni-halle.de/~brass/xml17/`

- Aktuelle Ankündigungen
- Folien der Vorlesung (PDF oder ps 4:1)
- Übungsblätter
- Alte Klausuren
- Verweise auf Literatur im WWW

  Zu WWW-Themen gibt es sehr viel nützliche Literatur im WWW selbst (z.B. Standards, Tutorials). Falls Sie empfehlenswerte Quellen finden, schicken Sie mir bitte eine EMail mit der URL.

- Verweise auf Software im WWW

Lehrbücher (1)

- Erhard Rahm, Gottfried Vossen (Hrsg.):
  Web & Datenbanken.
  Konzepte, Architekturen, Anwendungen.
  dpunkt.verlag, 2003, ISBN 3-89864-189-9, 488 Seiten.
- Meike Klettke, Holger Meyer:
  XML & Datenbanken.
  Konzepte, Sprachen, Systeme.
  dpunkt.Verlag, 2003, ISBN 3-89864-148-1, 428 Seiten.
- Georg Lausen:
  Datenbanken. Grundlagen und XML-Technologien.
  Spektrum Akademischer Verlag, 2005, ISBN 3827414881, 281 Seiten.

Lehrbücher (2)

- Harald Schöning:
  XML und Datenbanken. Konzepte und Systeme.
  Hanser Fachbuchverlag, 2002, ISBN 3446220089, 300 Seiten.
- Wassilios Kazakos, Andreas Schmidt, Peter Tomczyk: Datenbanken und XML.
  Konzepte, Anwendungen, Systeme.
  Springer, 2002, ISBN 354041956X, 352 Seiten.
- Akmal B. Chaudhri, Awais Rashid, Roberto Zicari:
  XML Data Management.
  Native XML and XML-Enabled Database Systems.
  Addison-Wesley, 2003, ISBN 0201844524, 688 Seiten.

Lehrbücher (3)

- Priscilla Walmsley:
  Definitive XML Schema, 2nd Ed.
  Prentice Hall, 2012, ISBN 0-13-288672-3, 768 Seiten.
- Eric van der Vlist:
  XML Schema.
  O'Reilly, 2002, ISBN 0596002521, 400 Seiten.

Lehrbücher (4)

- Wolfgang Lehner, Harald Schöning: XQuery:
  Grundlagen und fortgeschrittene Methoden.
  dpunkt.verlag, 2004, ISBN 3898642666, 304 Seiten.
- Howard Katz (Editor):
  XQuery from the Experts.
  A Guide to the W3C XML Query Language.
  Addison-Wesley, 2003, ISBN 0321180607, 512 Seiten.
- Rudolf Jansen:
  XQuery, Eine praxisorierentierte Einführung.
  Software & Support Verlag, 2004, ISBN 3-935042-65-5, 167 Seiten.

Lehrbücher (5)

- Jim Melton, Stephen Buxton:
  Querying XML.
  Morgan Kaufmann, 2006, ISBN 1-55860-711-0, 848 Seiten.
- Michael Seemann:
  Native XML Datenbanken im Praxiseinsatz.
  Software & Support Verlag, 2003, ISBN 3-935042-35-3, 316 Seiten, mit CD.
- Bastian Gorke:
  XML-Datenbanken in der Praxis.
  bomots verlag, 2006, ISBN 3-939316-19-9, 130 Seiten.

Lehrbücher (6)

- Michael Kay:
  XPath 2.0 Programmer's Reference.
  Wiley/Wrox, 2004, ISBN 0-7645-6910-4, 530 Seiten.
  Es gibt auch: XSLT 2.0 and XPath 2.0 Prog. Ref., 4th Ed., 2008, 1368 Seiten.
- Bob DuCharme:
  XML: The Annotated Specification.
  Prentice-Hall, 1998, ISBN 0-13-082676-6, 339 Seiten.
- Elliotte Rusty Harold, W. Scott Means:
  XML in a Nutshell, A Desktop Quick Ref., 3rd Ed.
  O'Reilly, Okt. 2004, ISBN 0-596-00764-7, 689 Seiten.

Eine Bitte

- Das Gebiet ist auch für mich noch ziemlich neu.
  Es entwickelt sich ja auch recht schnell.
- Wahrscheinlich weiß mancher von Ihnen zumindest über manches Detail mehr als ich.
  Das ist mir nicht peinlich, ich lerne gerne.
- Teilen Sie Ihr Wissen mit uns allen!
  Korrigieren Sie Fehler, falls Sie sie bemerken.
- Stellen Sie Fragen!
- Nehmen Sie an den Übungen in der Vorlesung teil.
  Bleiben Sie nicht einfach nur passiver Zuhörer!