Semistructured Data & XML
(Summer Term 2003)

(c) Prof Dr. Wolfgang May
Universität Göttingen, Germany

may@informatik.uni-goettingen.de

Wahlpflichtvorlesung Informatik; 4 SWS, 6 Kreditpunkte.

ZIELE DER VORLESUNG

- Am Ende sollen Sie sich in der XML-Welt auskennen, und mit einigen Konzepten bereits praktisch experimentiert haben
  ⇒ Anwendungsorientierung
  (erfordert selbständiges Arbeiten)
- Hintergründe, warum XML entstanden ist, und warum es so entstanden ist
  ⇒ Verstehen von Konzepten und Entwicklungen
- zugrundeliegende Metakonzepte
  ⇒ “Informatik” als ganzes

ÜBERBLICK

- andere Folien ...

TABLE OF CONTENTS

Kapitel 1
Einführung

KONTEXT AND ÜBERBLICK

- Datenbanken werden in vielen Bereichen genutzt ... Wirtschaft, Verwaltung, Forschung ...
- ursprünglich: Speichern von Informationen
  Relationales Modell, SQL “Einführung in Datenbanken und SQL”
- Entwicklung: Informationssysteme, Kombination von Datenbanken und Anwendungen, verteilte Datenbanken, Datenintegration
- heute: Web-basierte Informationssysteme, Electronic Data Interchange
  → neue Herausforderungen, semistrukturierte Daten, XML

1.1 Grundbegriffe

Im Grunde genommen basieren alle DB-Ansätze auf dem Konzept des “Datensatzes”.

Datenbankschema und -Zustand

Üblicherweise betrachtet man bei einer Datenbank ein *Schema* und einen *Zustand*:

Datenbankschema: Das Schema beinhaltet die *Metadaten* über die Datenbank, d.h., beschreibt die Konzepte (z.B. Objekttypen und Beziehungstypen).

Datenbankzustand: Der Zustand einer Datenbank ist durch die Menge der aktuell gespeicherten Daten gegeben. Dort sind alle Objekte und Beziehungen beschrieben, die in der Anwendungswelt zu einem gegebenen *Zeitpunkt* gelten.

Im Laufe der Zeit ändert sich der Datenbankzustand.

Die Menge der zulässigen Zustände wird im konzeptuellen Modell beschrieben (z.B. durch Integritätsbedingungen).

Data Dictionary: enthält die Definitionen des Datenbankschemas.

DATENMODELLE

Ein *Datenmodell* definiert Modellierungskonstrukte die dann zur Modellierung der Anwendungsdomäne genutzt werden können

- Definition von *Datenstrukturen*:
  - ER-Modell: Objekttypen und Beziehungstypen
  - Relationales Modell: Relationen
- einschliesslich Möglichkeiten zur Spezifikation von Integritätsbedingungen
- Definitionen von Operationen und ihrer Semantik (abstrakter Datentyp)
  - SQL, OQL (ähnlich wie SQL für objektorientierte Daten) ...

Es besteht im allgemeinen aus

- **D**ata **D**efinition **L**anguage (DDL) zur Schemadefinition,
- **D**ata **M**anipulation **L**anguage (DML) um Datenbankzustände zu bearbeiten (Anfragen, Datenmanipulation).

Datenmodelle

... es gibt viele verschiedene Datenmodelle und Modellierungssprachen

- Konzeptuelle Modellierung
  - Entity-Relationship-Model (nur statische Konzepte, grafisch)
  - Unified Modeling Language (UML)
    umfassender grafischer Formalismus zur Modellierung von Prozessen, basierend auf objektorientierten Ideen
- Logische Datenmodelle und Implementierungsmodelle
  - Netzwerkmodell (IMS (IBM), Ende 60er; CODASYL 1971), hierarchisches Modell (vorher)
  - Relationales Modell (Codd 1970, mit SQL (IBM System R 1974; Standard 1986))
  - Objektorientiertes Modell (ODMG 1993; OQL)
- Dokument-Datenmodell (SGML)
- Semistrukturierte Datenmodelle, XML
  - warum?
  - Entwicklung und aktuelle Situation

1.2 Relationales Modell

- *Relationales Modell* von Codd (1970): mathematische Grundlage: Mengentheorie
- nur ein einzelnes Strukturkonzept: *Relation* für Entitäten/Objekte und Beziehungstypen
- Eigenschaften von Entitäten/Objekte und Beziehungstypen durch *Attribute* dargestellt
- Ein Relationsschema bsteht aus einem Namen und einer Menge von Attributen
  Continent: Name, Area
- Jedem Attribut ist eine *Domäne (Wertebereich)* zugewiesen, die die erlaubten Werte des Attributes enthält. Attribute können oft auch *Nullwerte* haben.
  Continent: Name: VARCHAR(25), Area: NUMBER
- Ein **(relationales) Datenbankschema** ist durch eine (endliche) Menge von (Relations)schemata gegeben.
  Continent: ...; Country: ...; City: ...; encompasses: ...

Relationen

- Ein **(Datenbank) zustand** weist jedem **Relationsschema** eine **Relation** zu.
- Die Elemente einer Relation werden als *Tupel* bezeichnet.
  Jedes solche Tupel bezeichnet ein Objekt oder eine Beziehung.
  (Name: Asia, area: 4.5E7)

Example:

Continent	
<u>Name</u>	**Area**
VARCHAR(20)	NUMBER
Europe	9562489.6
Africa	3.02547e+07
Asia	4.50953e+07
America	3.9872e+07
Australia	8503474.56

Relationen: Beispiel

Continent	
<u>Name</u>	**Area**
Europe	9562489.6
Africa	3.02547e+07
Asia	4.50953e+07
America	3.9872e+07
Australia	8503474.56

Country				
Name	**<u>code</u>**	**Population**	**Capital**	**...**
Germany	D	83536115	Berlin	
Sweden	S	8900954	Stockholm	
Canada	CDN	28820671	Ottawa	
Poland	PL	38642565	Warsaw	
Bolivia	BOL	7165257	La Paz	
..	..	..	..	

encompasses		
<u>Country</u>	**<u>Continent</u>**	**Percent**
VARCHAR(4)	VARCHAR(20)	NUMBER
R	Europe	20
R	Asia	80
D	Europe	100
...	...	...

- mit referentiellen Integritätsbedingungen

ANFRAGESPRACHE: SQL

- Seit ca. 1974 in IBM System R etc.; Standard SQL-1: 1986
- 1995: 80% aller Datenbanken verwenden das relationale Modell und SQL

```
SELECT name,percent
FROM country,encompasses
WHERE country.code = encompasses.country
   AND encompasses.continent = 'Europe';
```

- intuitiv verständlich
- klauselbasierte, *deklarative* Sprache
- *funktionales Programmierparadigma*
- *mengenorientiert*: Ergebnis (fast) jedes Ausdrucks ist wieder eine Relation
- *orthogonale Konstrukte*, (fast) beliebig schachtelbar

HINWEIS

"Datenbank" bedeutet nicht nur relationales Modell & SQL, sondern allgemein

- Persistente Speicherung
- Massendaten
- Mehrbenutzerkonzept
  - Zugriffskontrolle/Sicherheit
- Transaktionskonzepte
  - Korrektheit/Konsistenz
  - Datensicherheit: Rollback, Recovery

... alles *allgemeine Konzepte*, die für Netzwerk-Modell, relationales Modell, objektorientiertes Modell, und auch XML gültig sind.