

# Datenbanken 1

Dozent:  
Pro. Dr. Stefan Kramer

L<sup>A</sup>T<sub>E</sub>X von:  
Sven Bamberger

Zuletzt Aktualisiert:  
9. Oktober 2014



JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ



Dieses Skript wurde erstellt, um sich besser auf die Klausur vorzubereiten.

Dieses Dokument garantiert weder Richtigkeit noch Vollständigkeit, da es aus Mitschriften und Vorlesungsfolien gefertigt wurde und dabei immer Fehler entstehen können. Falls ein Fehler enthalten ist, bitte melden oder selbst korrigieren und neu hoch laden.



# Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>1</b>
1.1	Klausurtermin . . . . .	1
1.2	Material für die Klausur . . . . .	1
1.3	Empfohlene Lektüre: . . . . .	1
1.4	Verwendete Software: . . . . .	2
<b>2</b>	<b>Einleitung und Übersicht</b>	<b>3</b>
2.1	Empfohlene Lektüre: . . . . .	3
2.2	Motivation für ein DBMS . . . . .	3
2.3	Die Abstraktionsebenen eines Datenbanksystems . . . . .	4
2.4	Datenunabhängigkeit . . . . .	5
2.5	Datenmodellierung . . . . .	5
2.6	Logische Datenmodelle . . . . .	6



# 1 Allgemeines

## 1.1 Klausurtermin

Die Klausur findet am 14.08.2014 in Raum N1 und/oder N3 statt.

## 1.2 Material für die Klausur

Bis zum aktuellen Zeitpunkt sind keine Hilfsmittel zugelassen

## 1.3 Empfohlene Lektüre:

Für die Vorlesung wird das folgende Buch dringend empfohlen, da sich die Vorlesung an diesem orientieren wird.

Datenbanksysteme: Eine Einführung

Autoren: Alfons Kemper, André Eickler

Auflage: 9.

Oldenbourg Wissenschaftsverlag

(25. September 2013)

ISBN-10: 3486721399

ISBN-13: 978-3486721393

Die Auflage Nummer 8 wird für die Übungen verwendet. Auch Auflage 7 kann noch ausreichend sein. Jedoch ist dieses Buch äußerst lange aktuell und es lohnt sich dieses anzuschaffen. In dieser Vorlesung wird Kapitel 1 - 9 behandelt, wobei ein paar Themenbereiche wie z.B. JDBC ausgelassen werden.

### 1.3.1 Weiterführende Lektüre/Links:

1. A. Silberschatz, H. F. Korth und S. Sudarshan Database System Concepts, 4. Auflage, McGraw-Hill Book Co., 2002.
2. R. Elmasri, S.B. Navathe: Fundamentals of Database Systems, Benjamin Cummings, Redwood City, Ca, USA, 2. Auflage, 1994
3. R. Ramakrishnan, J. Gehrke: Database Management Systems, 3. Auflage, 2003.
4. G. Vossen : Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. Oldenbourg, 2001.
5. **D. Maier: The Theory of Relational Databases. Computer Science Press. 1983.**
6. S. M. Lang, P.C. Lockemann: Datenbankeinsatz. Springer Verlage, 1995.
7. C. Batini, S. Ceri, S.B. Navathe: Conceptual Database Design, Benjamin Cummings, Redwood City, Ca, USA, 1992.
8. **C. J. Date: An Introduction to Database Systems. McGraw-Hill, 8. Aufl., 2003.**

## 1 Allgemeines

9. J.D. Ullmann, J. Widom: A First Course in Database Systems, McGraw Hill, 2. Auflage, 2001. 8
10. A. Kemper, G. Moerkotte: Object-Oriented Database Management: Applications in Engineering and Computer Science, Prentice Hall, 1994
11. E. Rahm: Mehrrechner-Datenbanksysteme. Addison-Wesley, 1994.
12. P. Dadam: Verteilte Datenbanken und Client/Server Systeme. Springer Verlag, 1996
13. G. Weikum, G. Vossen: Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control. Morgan Kaufmann, 2001.
14. T. Härder, E. Rahm: Datenbanksysteme – Konzepte und Techniken der Implementierung, 2001.

### 1.4 Verwendete Software:

Es wird vorzugsweise auf PostgreSQL und MySQL als OpenSource Varianten zurück gegriffen. Um die Blätter auch ordentlich bearbeiten zu können wird die Installation dieser Systeme empfohlen. In den Prüfungen wurde SQLite verwendet.



## 2 Einleitung und Übersicht

*Datenbankverwaltungssysteme* (DBMS) gewinnen dank der digitalen Vernetzung und Kommunikation immer mehr an Bedeutung und sind heutzutage nicht mehr aus unserem Alltag wegzudenken. Zwar bemerken viele den Einsatz eines DBMS nicht jedoch sind diese Systeme überall

### 2.1 Empfohlene Lektüre:

Für die Vorlesung wird das folgende Buch dringend empfohlen, da sich die Vorlesung an diesem orientieren wird.

vertreten z.B. in Versicherungen, Banken, Universitäten (welche das Hauptbeispiel für diese Vorlesung bildet) noch bei vermeintlich nicht aufwendigen Blogs wie einem einfachem Blog.

Ein Datenbankverwaltungssystem besteht aus einer Menge von *Daten* und dem *Datenbankverwaltungssystem*

- Die *Daten* werden auch als Datenbasis bezeichnet welche in einer gewissen Beziehung zueinander stehen.
- Die Programme zum Zugriff, Modifikation und Kontrolle genutzt werden, werden als *Datenbankverwaltungssystem* bezeichnet

Diese Trennung wird häufig unterlassen, um eine Verwirrung zu vermeiden. Stattdessen werden Datenbankverwaltungssysteme (auch Datenbanksysteme) als Kombination der vorher genannten Unterschiede bezeichnet.

### 2.2 Motivation für ein DBMS

Es gibt einige Probleme wenn Firmen oder Organisationen auf ein DBMS verzichten müssten und auf Papier (Karteikarten, Akten, ...) oder separate Dateien zurück greifen müssten.

#### **Redundanz und Inkonsistenz:**

Wenn Daten isoliert gehalten werden, müssen diese mehrfach vorhanden sein (Zweigstelle einer Firma) und wenn Informationen in einer dieser Dateien geändert werden, geschieht dies lediglich in dieser einen Dateien, Hierdurch entstehen, aufgrund der vorhandenen Redundanz, Inkonsistenzen in den Informationen.

#### **Beschränkte Zugriffsmöglichkeiten:**

Es ist nahezu unmöglich separierte Dateien miteinander zu verknüpfen und logische Abhängigkeiten zu generieren. Mit einem DBMS können Informationen einer Firma/Organisation einheitlich modelliert werden (Datenmodell) . Dadurch lassen sich die Daten auf viele Arten verknüpfen.

#### **Probleme beim Mehrbenutzerbetrieb:**

Dateisysteme bieten nur wenige Kontrollmechanismen für den Mehrbenutzerbetrieb. Das bedeutet, falls mehrere Benutzer an einer Datei arbeiten, wird bei jedem Speicher Vorgang alle vorherigen Änderungen überschrieben. Dies nennt man ein „lost update“. DBMS bieten eine Mehrbenutzerkontrolle, welche solche Anomalien erkennt und verhindert.

### Verlust von Daten:

Bei isolierten Daten wird die Wiederherstellung eines konsistenten Zustandes äußerst schwierig. Datenbankverwaltungssysteme haben Möglichkeiten diese Daten wieder herzustellen.

### Integritätsverletzung:

Diese *Abhängigkeitsverletzungen* können auftreten, wenn Daten und Informationen isoliert bearbeitet und betrachtet werden. Die Überprüfung auf Integritätsverletzungen sollte vom System geprüft und bei Bedarf abgelehnt werden. DBMS führen Transaktionen nur aus, wenn sie die Datenbasis in einen konsistenten Zustand überführen.

### Sicherheitsprobleme:

Nicht jeder sollte Zugriff auf alle gespeicherten Daten haben, sondern lediglich auf bestimmte Bereiche und Daten, die unbedingt notwendig sind. Auch sollten nur bestimmte Leute das *Privileg* haben Daten zu verändern oder gar zu löschen. DBMS Systeme haben ausgefeilte Rechteverwaltungen für Benutzer und Benutzergruppen..

### hohe Entwicklungskosten für Anwendungsprogramme:

Anwendungsentwickler müssen sich meist mit einem Teil der vorher genannten Probleme auseinandersetzen um Informationen zu speichern und zu verwalten. Ein DBMS bietet eine einfache und erprobte Möglichkeit diese Probleme direkt zu erledigen.

## 2.3 Die Abstraktionsebenen eines Datenbanksystems

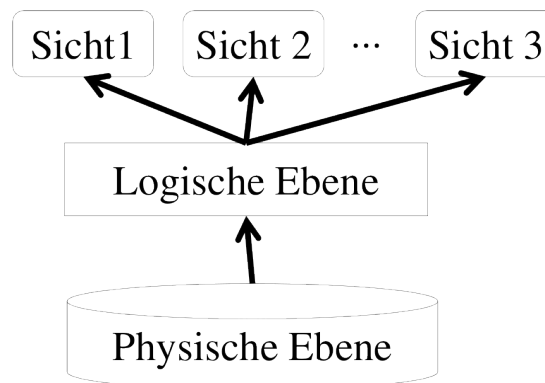


Abbildung 2.1: Drei Abstraktionsebenen eines Datenbanksystems

Bei einem DBMS unterscheidet man drei Abstraktionsebenen(siehe Abbildung 2.1).

**Die physische Ebene:** Auf dieser Ebene wird festgelegt, wie die Daten gespeichert sind. Im Allgemeinen sind die Daten auf einem Speichermedium (meistens als Plattenspeicher realisiert) abgelegt.

**Die logische Ebene:** Auf der logischen Ebene wird in einem sogenannten *Datenbankschema* festgelegt, welche Daten abgespeichert sind.

**Die Sichten:** Während das Datenbankschema der logischen Ebene ein integriertes Modell der gesamten Informationsmenge des jeweiligen Anwendungsbereichs (z.B. des gesamten Unternehmens) darstellt, werden in den Sichten Teilmenge der Informationen bereitgestellt. Diese Sichten sind auf Benutzer bzw. Benutzergruppen zugeschnitten, die lediglich diese Sichten benötigen. Denn nicht jeder Benutzer benötigt alle Daten die existieren.

## 2.4 Datenunabhängigkeit

Durch die drei Ebenen wird eine Datenunabhängigkeit erreicht, welche zu wohl definierten Schnittstellen führt, ermöglicht es die Realisierung der Datenbank zu variieren ohne die Benutzer der Schnittstellen in Mitleidenschaft zu ziehen. Aus den drei Schichten ergeben sich zwei Stufen der Datenunabhängigkeit im DBMS

**Physische Datenunabhängigkeit:** Die Modifikation der physischen Speicherstruktur belässt die logische Ebene (das Datenbankschema) invariant. Z.B. erlauben fast alle Datenbanksysteme das nachträgliche Anlegen eines Indexes, um die Datenbankobjekte schneller finden zu können. Dies darf keinen Einfluss auf bereits existierende Anwendungen in der logischen Ebene haben. Es sollte lediglich die Effizienz positiv beeinflussen.

**Logische Datenunabhängigkeit:** Die Anwendungen nehmen stets Bezug auf die logische Struktur der Datenbasis. Bei Änderungen der logischen Ebene (des Datenbankschemas) können diese in den Sichten vor dem Anwender verborgen werden. Dadurch wird zu einem gewissen Grad eine logische Datenbankunabhängigkeit erzielt.

Die heutigen Datenbanksysteme erfüllen mindestens die physische Datenbankunabhängigkeit. Die logische Datenunabhängigkeit kann schon rein konzeptuell nur für einfachste Modifikationen des Datenbankschemas gewährleistet werden.

## 2.5 Datenmodellierung

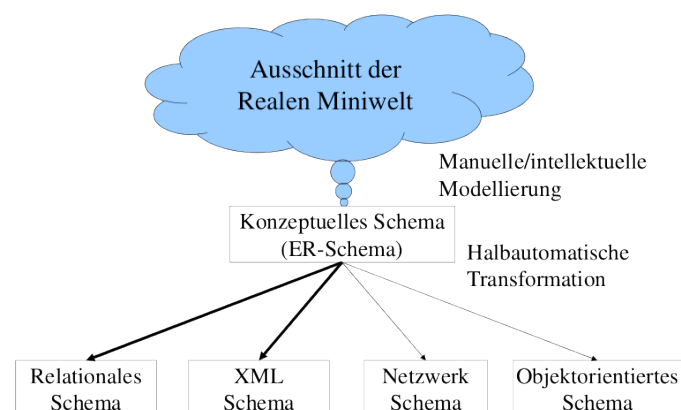


Abbildung 2.2: Datenmodellierung

Datenbankverwaltungssysteme basiert auf einem so genannten Datenmodell. Das Datenmodell stellt die Infrastruktur für die Modellierung der realen Welt zur Verfügung. Das Datenmodell beinhaltet die Möglichkeit zur

- Beschreibung der Datenobjekte und zur
- Festlegung der anwendbaren Operatoren und deren Wirkung

Das Datenmodell legt somit die generischen Strukturen und Operatoren fest, die man zur Modellierung einer bestimmten Anwendung ausnutzen kann. Eine Programmiersprache legt die Typkonstruktoren und Sprachkonstrukte fest, mit deren Hilfe man spezifische Anwendungsprogramme realisiert.

## 2 Einleitung und Übersicht

Das Datenmodell besteht demnach aus zwei Teilsprachen:

1. der *Datendefinitionssprache* (engl. *Data Definition Language*, DDL)
2. der *Datenmanipulationssprache* (engl. *Data Manipulation Language*, DML)

Die DDL wird benutzt, um die Struktur der abzuspeichernden Datenobjekte zu beschreiben. Die Strukturbeschreibung aller Datenobjekte des betrachteten Anwendungsbereichs nennt man das Datenbankschema.

Die Datenmanipulationssprache (DML) besteht aus

- der *Anfragesprache* (engl. *Query Language*)
- der „eigentlichen“ Datenmanipulationssprache zur Änderung von abgespeicherten Datenobjekten, zum Einfügen von Daten und zum Löschen von gespeicherten Daten.

Die DML (einschließlich der Anfragesprache) kann in zwei unterschiedlichen Arten genutzt werden:

- *interaktiv*, indem DML-Kommandos direkt am Arbeitsplatzrechner (oder Terminal) eingegeben werden
- in einem Programm einer höheren Programmiersprache, das „eingebettet“ DML-Kommandos enthält.

## 2.6 Logische Datenmodelle

- Netzwerkmodell
- Hierarchisches Datenmodell
- Relationales Datenmodell
- XML Schema
- Objektorientiertes Datenmodell
  - Objektrelationales Schema
- Deduktives Datenmodell

# Abbildungsverzeichnis

2.1	Abstraktionsebenen . . . . .	4
2.2	Datenmodellierung . . . . .	5