

EINFÜHRUNG IN DIE DATENBANKPROGRAMMIERUNG

(Datenbanksysteme I)

Prof. Dr. Udo Lipeck

Sommersemester 2015

Vorlesung: Di 14:15–15:45, F 102

Übungen: M. Sc. Michael Schäfers und Tutoren
Mi 9-10, Mi 13-14, Do 9-10 oder Fr 11-12

Beginn: Mi, 15.04.14

Gruppenübungen, wöchentliche Hausübungen mit Punkten und Bonus, sowie mit DB-Zugang über WWW

Adresse: Welfengarten 1, Räume C 102 | C 103

Telefon: 4951 | 4599

E-Mail: { ul | mms }@dbs.uni-hannover.de

WWW: www.dbs.uni-hannover.de

Web-Begleitung: Bitte tragen Sie sich im StudIP-System ein:

- für die Vorlesung: Zugriff auf Begleitmaterialien
- in eine Übungsgruppe¹:
Zugriff auf Aufgaben und Lösungen

¹erst 10 Minuten nach Ende der ersten Vorlesung möglich

Literaturauswahl

– meist auch zur Folgevorlesung, ohne reine SQL-Bücher –

- Date, C.J.: An Introduction to Database Systems. 8th Edition, Addison-Wesley, 2004

** Elmasri, R. / Navathe, S.B.: Grundlagen von Datenbanksystemen – Bachelorausgabe. 3. Auflage, Pearson Studium, 2009 (34,95 EUR)

oder Fundamentals of Database Systems. 6th Edition, Pearson Education, 2011²

- Garcia-Molina, H. / Ullman, J.D. / Widom, J.: Database Systems – The Complete Book. 2nd Edition, Prentice Hall, 2009/2013

* Kemper, A. / Eickler, A.: Datenbanksysteme – Eine Einführung. 9. Auflage, Oldenbourg, 2013 (39,95 EUR)

dazu Kemper/Wimmer: Übungsbuch Datenbanksysteme, 3. Auflage, Oldenbourg, 2012

** Saake, G. / Sattler, K.-U. / Heuer, A.: Datenbanken: Konzepte und Sprachen. 5. Auflage, MITP-Verlag, 2013 (39,95 EUR)

* Silberschatz, A. / Korth, H.F. / Sudarshan, S.: Database System Concepts. 6th Edition, McGraw-Hill, 2010

- Vossen, G.: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 5. Auflage, Oldenbourg, 2008

²Die gegenüber der Bachelorausgabe fehlenden Kapitel aus der deutschen Vollausgabe von 2002 sind als PDF-Texte elektronisch verfügbar; siehe (im Uni-Domain 130.75.) http://www.dbs.uni-hannover.de/lehre/dbs2-1112/Elmasri_Differenz

Kapitel 1

Einführung

Anwendungsbereiche von Datenbanksystemen

traditionell: betriebswirtschaftlich-administrativ

- z. B. ein Unternehmen:
 - Einkauf, Lagerhaltung, Produktion, Verkauf, Personal, Lohnabrechnung, Buchhaltung, ...
- z. B. eine Bibliothek:
 - Buchzugang, Katalogisierung, Benutzerverwaltung, Buchausleihe, Mahnungen, Buchhaltung, ...
- z. B. eine Bank:
 - Kunden-/Kontenverwaltung, Kontoführung/Buchungen, Kurse, Zahlungen von Gebühren/Zinsen/Raten u.a., ...

⇒ viele Teilanwendungen mit überlappendem Datenbedarf, aber (noch) überwiegend nur Strings, Zahlen, Kalenderdaten;
viele einfach strukturierte Datensätze, relativ einfache Algorithmen

Anwendungsbereiche von Datenbanksystemen (Forts.)

neuartig: “Nicht-Standard”-Anwendungen

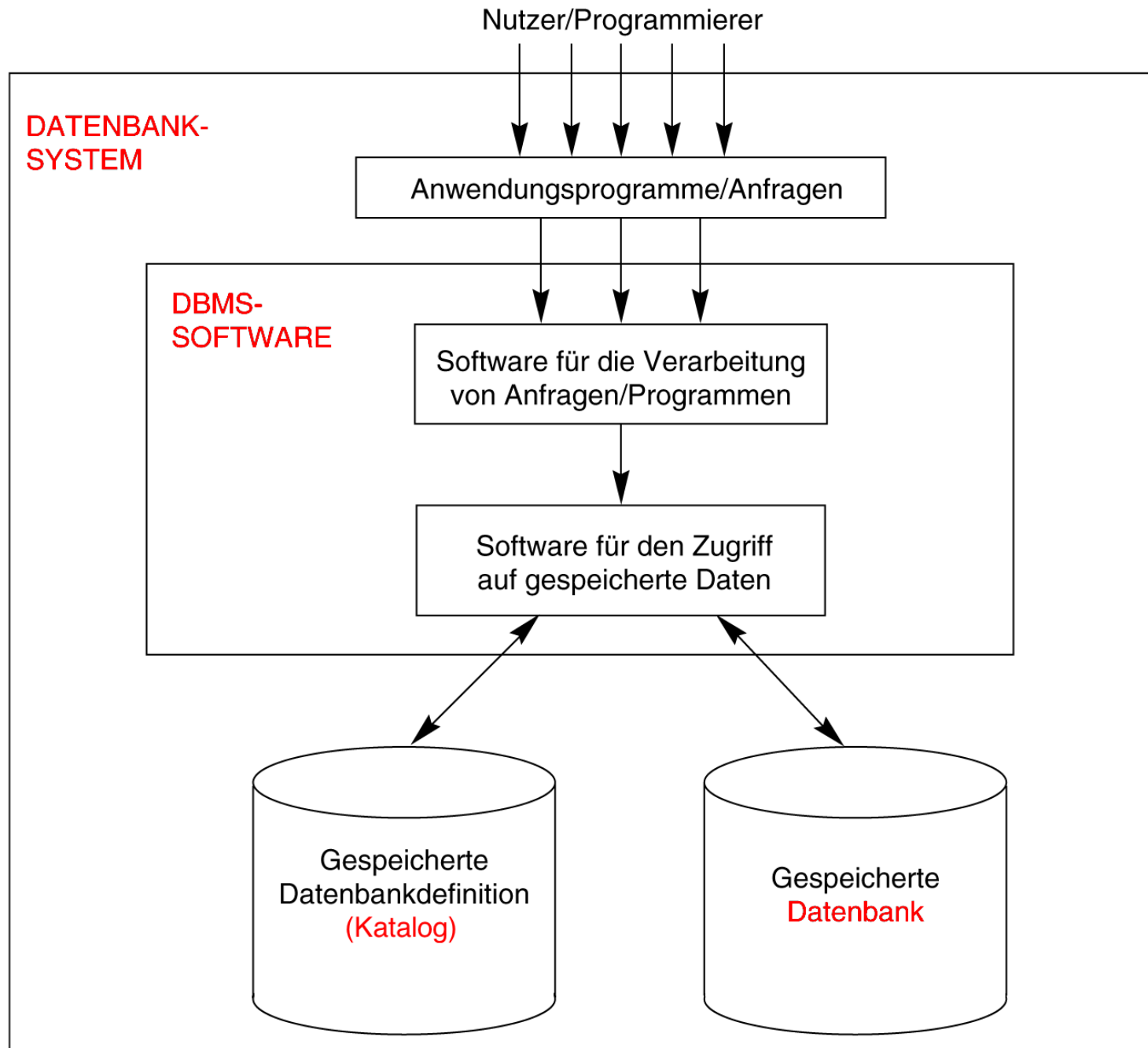
- Multimedia-Datenbanken, Elektronische Bibliotheken, ...
(Texte, Bilder, Audio, Video, WWW-Seiten ...)
- CAD-Systeme, Geographische Informationssysteme (GIS), ...;
allgemeiner: ingenieur-/naturwissenschaftliche Anwendungen
(Konstruktionszeichnungen/-graphiken, Karten, 2D/3D-Geometrien ...)

⇒ komplexe Objekte, komplexe Algorithmen, lange Transaktionen

Was ist ein Datenbanksystem ?

Ein **Datenbanksystem (DBS)** besteht aus:

- einer **Datenbank (DB)**, einem integrierten Datenbestand:
 - einheitlich gemäß einem Datenmodell strukturiert,
z. B. durch “Relationen” $\hat{=}$ Tabellen
- einem **Datenbank-Managementsystem (DBMS)**:
Software zur Erzeugung, Benutzung und Wartung von Datenbanken:
 - universell, d. h. anwendungsunabhängig !
 - mit bequemen Benutzerschnittstellen,
insbes. einer DB-Sprache wie z. B. SQL
 - aber auch mit möglichst effizienten Implementierungen
- grundlegenden Anwendungsprogrammen (Transaktionen)
- und einem **Katalog** für die Definition der jeweiligen Datenbank



Grundoperationen eines DBMS:

- Adhoc-Anfragen an die gespeicherten Daten
(in SQL: “select ... from ... where ...”)
- Zusammenstellungen, Berichte, Auswertungen mit Hilfe
relativ “einfacher” Datenverknüpfungen
(Logik, Arithmetik, Stringverarbeitung u. a.; evtl. sogar geome-
trische Operatoren — je nach angebotenen Datentypen)
- Änderungen bzw. “Manipulationen” der gespeicherten Daten:
Erzeugen, Einfügen, Aktualisieren, Löschen
(in SQL: create, insert, update, delete)

Beispielanfrage:

RECHNUNG	RNr	KName	Betrag	RDatum	bezahlt?
(Datenbank)		:			
	208	Marvin	127,63	15.07.2014	nein
	612	Ford	42,00	10.10.2012	nein
	1204	Marvin	2047,99	04.04.2015	ja
	1248	Marvin	1412,19	12.02.2015	nein
		:			

- Anfrage in SQL: “Welche Rechnungen in welcher Höhe hat Marvin noch nicht bezahlt?”

```
select RNr, Betrag
from RECHNUNG
where KName = 'Marvin' and bezahlt? = 'nein'
```

- Ergebnis:

	RNr	Betrag
	208	127,63
	1248	1412,19

Kern – Anforderungen an ein DBMS

- Abbildung von Daten einer *Mini-Welt*, d. h. eines zusammenhängenden Ausschnitts der realen Welt, z. B. eines Unternehmens
- **Integration** aller Anwendungsdaten in einer Datenbank: einheitliche Verwaltung der Daten und ihrer Beziehungen
→ keine bzw. *kontrollierte Redundanz*
(Zentralisierung im logischen Sinne, schließt physische Verteilung nicht aus)
- **Operationen:**
Datenspeicherung und -änderungen (“updates”),
Datensuchen (!) und -auswahlen durch Anfragen (“queries”)
- **Persistenz:** dauerhafte Speicherung
- **Katalog** (“data dictionary”):
Verwaltung der eigenen Datenbeschreibungen = *Metadaten*
(idealerweise mit gleichen Zugriffsmöglichkeiten wie auf echte Daten)

Benutzungs – Anforderungen an ein DBMS

- **Benutzerschnittstellen** für mehrere Benutzerklassen:
naive (parametrische) ... gelegentliche ... erfahrene ... Endbenutzer, Anwendungsprogrammierer, DB-Administratoren, DB-Entwerfer
 - mehrere **Benutzersichten** (“views”) auf die gleichen Daten
→ Gewährleistung der *logischen Datenunabhängigkeit*
d.h. die Benutzerschnittstelle einer Anwendung soll stabil sein gegen Änderungen anderer Anwendungen.
- ↓ zentrale Kontrolle aller Datenbank-Benutzungen, insbesondere:
- **Integritätsüberwachung** (“integrity”):
Gewährleistung korrekter (konsistenter) DB-Inhalte und -Änderungen
 - **Datenschutz** (“security”):
Ausschluss unauthorisierter Zugriffe
 - **Datensicherung** (“backup/recovery”)
zur Wiederherstellung von Daten nach Systemfehlern

Ablauf-/Datenorganisations – Anforderungen an ein DBMS

- Unterstützung gleichzeitiger Benutzungen → *Mehrbenutzerfähigkeit* insbes. **Synchronisation** von konkurrierenden Zugriffen mehrerer Benutzer (“concurrency control”)
- **Transaktionskonzept:**
Zusammenfassung von einzelnen DB-Änderungen zu Einheiten (der Integritätsüberwachung, Datensicherung, Synchronisation usw.)
- **Datenorganisation**, für Benutzer möglichst transparent:
→ Gewährleistung der *physischen Datenunabhängigkeit*
d. h. Benutzerschnittstelle(n) sollen unabhängig sein von
 - Geräten wie z. B. Plattenspeichern, Betriebssystemen,
 - Speicherstrukturen wie z. B. Satzadressen/-formaten,
 - Zugriffspfade wie z. B. Suchbäumen oder Hashtabellen, usw.
- ⇒ möglichst automatische **Anfrageoptimierung**

Beachte: Solch vielfältige Funktionalität kostet Overhead (und Geld).

Architekturprinzip I:

Unterscheidung zwischen **Meta-Daten** und **Daten**

Meta-Daten $\hat{=}$ **DB-Schema** (Beschreibung der DB)

Daten $\hat{=}$ **DB-Zustand** / “DB-Ausprägung”
(Inhalt der DB zu einem Zeitpunkt)

... analog zu Deklarationen und Inhalten von Programmvariablen

Das DBMS

- verwaltet DB-Schemata (im Katalog) und
- verwaltet DB-Zustände (in der Datenbank), d.h. es verarbeitet und kontrolliert DB-Operationen gemäß dem jeweiligem DB-Schema.

Meta-Daten (Forts.)

Typische Bestandteile eines DB-Schemas:

1. Struktur von DB-Zuständen, z. B. Tabelle:

PERSON	Name	Familienstand	Ehegatte

2. Integritätsbedingungen, d. h. Regeln für konsistente DB-Zustände und DB-Änderungen, z. B.:

a) Personen sind durch Namen eindeutig identifiziert.

b) Famstand = ledig \Rightarrow Ehegatte **is null** (undefiniert)

c) **old** Famstand = verheiratet \Rightarrow **new** Famstand \neq ledig

3. Zugriffsrechte, d. h. Regeln für autorisierte DB-Zugriffe, z. B.:

Die Abteilung Statistik darf nur auf das Feld Famstand zugreifen, und zwar nur lesend.

Architekturprinzip II: 3-Ebenen-Architektur

Konsequenz aus der Forderung nach log./phys. Datenunabhängigkeit:

