



Kommunikation und Datenhaltung

Datenhaltungsteil

Frank Eichinger, Mirco Stern

Charakteristika von Datenbanken

Einleitung

Motivation

Historie

Einsatz

- Eine Bank:



- Langfristige **Aufbewahrung** von Werten (hier: Daten)
- Werte werden zur **Sicherheit vor Verlusten** auf die Bank gebracht
- Eine Bank bietet **Dienstleistung für mehrere Kunden** an um effizient arbeiten zu können

Einsatz von Datenbanken

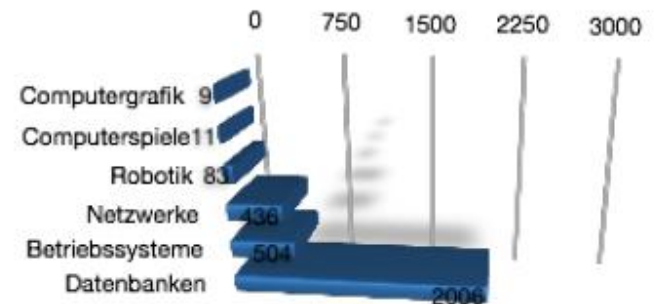
Einleitung

Motivation

Historie

Einsatz

- Datenbanksysteme sind Herzstück heutiger IT-Infrastrukturen
- ...allgegenwärtig
- Datenbankspezialisten sind gefragt



Stellenangebote (monster.de, August 2008)

Ohne Datenbanken: Datenredundanz

Einleitung

Motivation

Historie

Einsatz

- Basis- oder Anwendungssoftware verwaltet eigenen Daten in eigenen (Datei-)Formaten, z.B. in Unternehmen
 - Buchhaltung: Artikel, Adressen
 - Lagerverwaltung: Artikel, Aufträge
 - Auftragsverwaltung: Aufträge, Artikel, Adressen
 - Produktion, Bestelleingang, Kalkulation, etc. benötigen z.T. ebenfalls diese Daten
 - Daten sind redundant
 - D.h. mehrfach gespeichert
 - Probleme:
 - Verschwendung von Speicherplatz
 - „Vergessen“ von Änderungen
- ⇒ (Ein) Ziel von DBMS: Beseitigung von Datenredundanz



Ohne Datenbanken: Weitere Probleme

Einleitung

Motivation

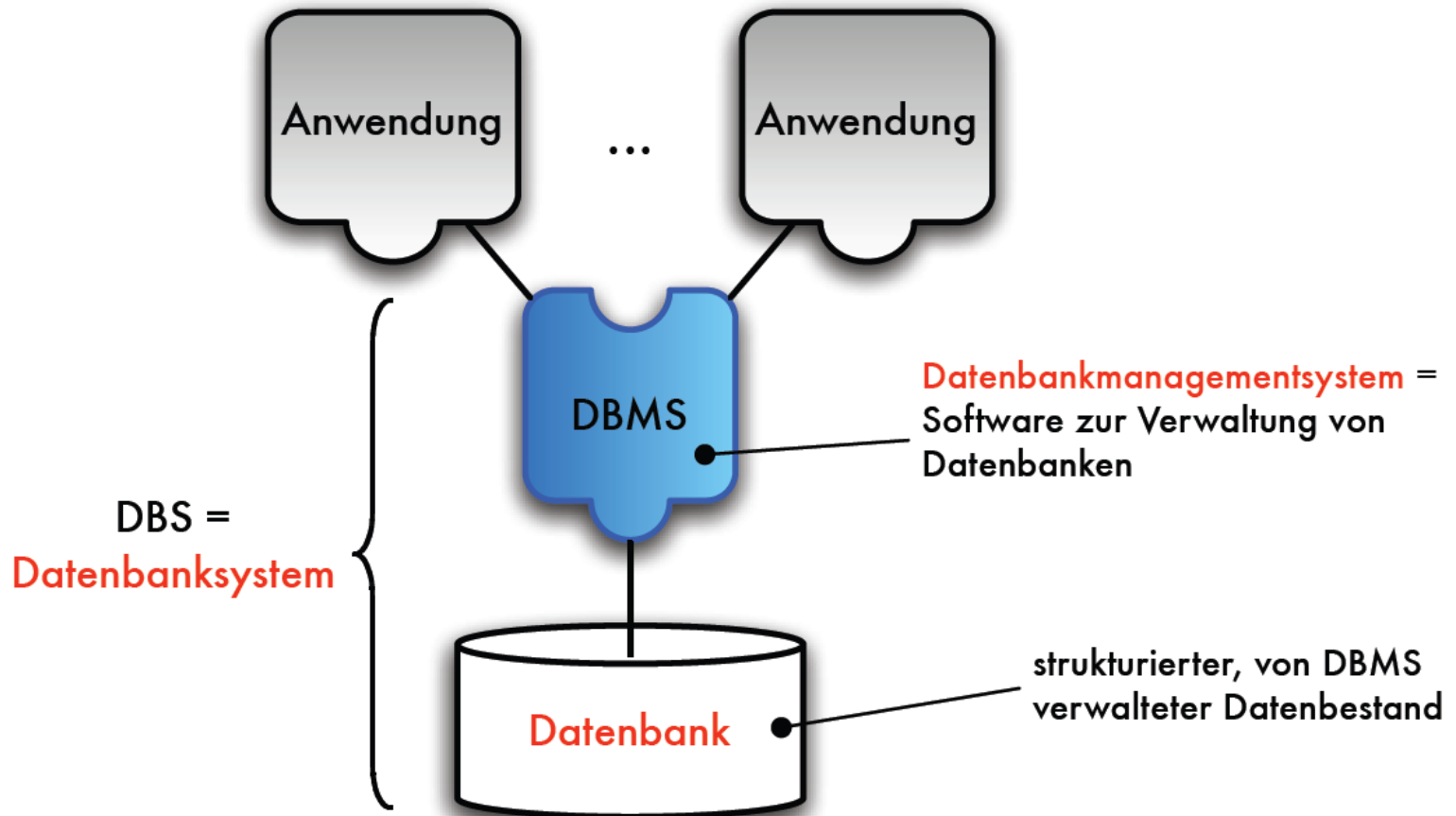
Historie

Einsatz

- Andere Software-Systeme können große Mengen von Daten nicht effizient verarbeiten
- Mehrere Benutzer oder Anwendungen können nicht parallel auf den gleichen Daten arbeiten, ohne sich zu stören (**Beispiel?**)
- Anwendungsprogrammierer / Benutzer können Anwendungen nicht programmieren / benutzen, ohne
 - Interne Darstellung der Daten
 - Speichermedien oder Rechner zu kennen (**Datenunabhängigkeit** nicht gewährleistet)
- Zugriffskontrolle und Datensicherheit sind nicht gewährleistet ohne zentrale Datenhaltung

Idee: Datenintegration durch Datenbanksysteme

- Einleitung
- Motivation
- Historie
- Einsatz



Mit Datenbanken: Datenintegration

- Gesamte Basis- und Anwendungssoftware arbeitet auf denselben Daten
(z.B. Adressen und Artikel nur einmal gespeichert)
 - Datenbanksysteme können große Datenmengen effizient verwalten
(Deklarative Anfragesprachen, Optimierung)
 - Benutzer können parallel auf Datenbanken arbeiten
(Transaktionskonzept)
 - Datenunabhängigkeit durch 3-Ebenen-Konzept
 - Integrität/ Konsistenz der Daten
 - Zugriffskontrolle (kein unbefugter Zugriff) und Datensicherheit (kein ungewollter Datenverlust) werden vom System gewährleistet
 - (Kosten der Anwendungsentwicklung)

Einleitung

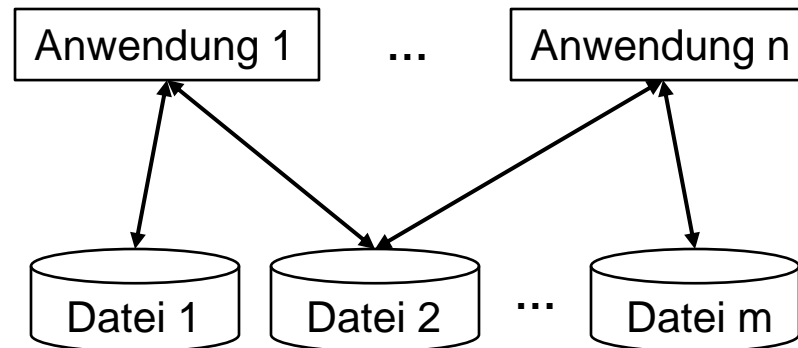
Motivation

Historie

Einsatz

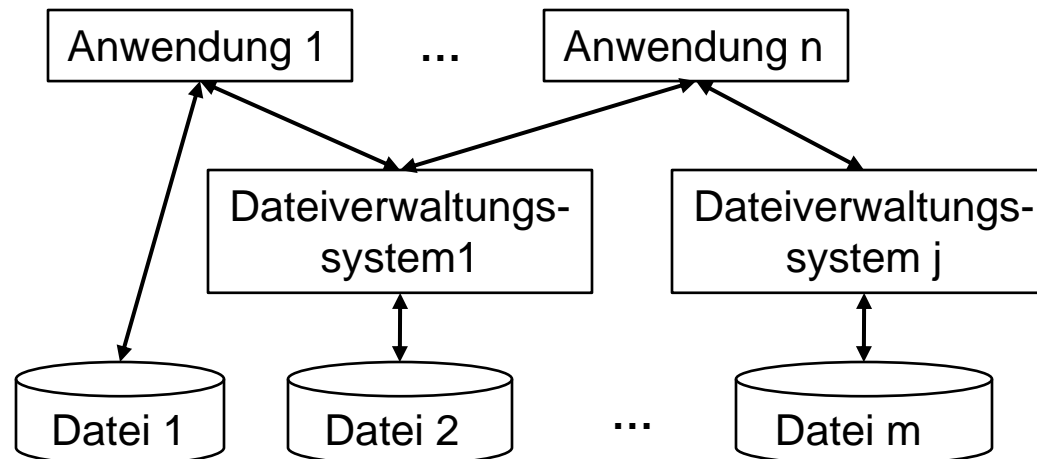
Historie: 60er Jahre

- Anfang 60er
 - Elementare Dateien
 - Anwendungsspezifische Datenorganisation (geräteabhängig, redundant \Rightarrow Inkonsistenzen)



Historie: 60er Jahre

- Anfang 60er
 - Elementare Dateien
 - Anwendungsspezifische Datenorganisation (geräteabhängig, redundant \Rightarrow Inkonsistenzen)
- Ende 60er Jahre
 - Dateiverwaltungssysteme (SAM, ISAM)
 - Mit Dienstprogrammen (Sortieren) (geräteunabhängig, aber: redundant \Rightarrow Inkonsistenzen)





Historie: 60er Jahre

- Anfang 60er
 - Elementare Dateien
 - Anwendungsspezifische Datenorganisation (geräteabhängig, redundant \Rightarrow Inkonsistenzen)
- Ende 60er Jahre
 - Dateiverwaltungssysteme (SAM, ISAM)
 - Mit Dienstprogrammen (Sortieren) (geräteunabhängig, aber: redundant \Rightarrow Inkonsistenzen)
- Erste DBS basierend auf hierarchischem Modell, Netzwerkmodell
 - Zeigerstrukturen zwischen Daten
 - Navigierende DML („Data Manipulation Language“ - Anfragesprache)
 - Trennung DML/ Programmiersprache

Einleitung

Motivation

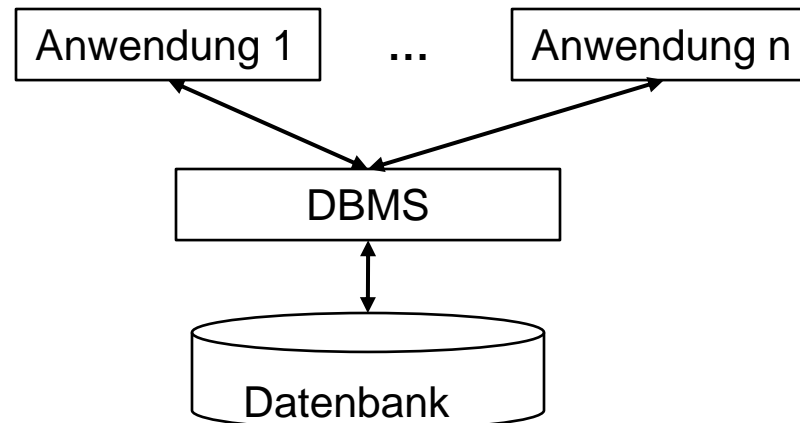
Historie

Einsatz

Historie: 70er Jahre

- 70er Jahre: Relationale Datenbanksysteme

- Geräte- und Datenunabhängigkeit
- Redundanzfrei
- Konsistent



Einleitung

Motivation

Historie

Einsatz

Historie: 70er Jahre

- 70er Jahre: Relationale Datenbanksysteme
 - Geräte- und Datenunabhängigkeit
 - Redundanzfrei
 - Konsistent
- Daten in Tabellenstruktur
 - Beispiel: Musikversand
 - Künstlertabelle
 - Speichert Informationen zu Künstlern (Name, Herkunftsland, Geburtsjahr, ...)
 - Identifikation über KünstlerID
 - Titeltabelle
 - Speichert Informationen zu den Titeln (Name, Art, Größe, ...)
 - Identifikation über Titelnummer
 - Weitere Tabellen (Alben, Bewertungen, ...)

Einleitung

Motivation

Historie

Einsatz

Daten in Tabellenform - Beispiel

- Einleitung
- Motivation
- Historie
- Einsatz

Künstler

KID	Name	Land	Jahr
1012	Neil Young	Kanada	1945
1014	Rammstein	Deutschland	1994
1015	The Ramones	USA	1974
1016	Eric Fish	Deutschland	1969

Titel

TitleID	Name	Art	Größe	KID
102	Neil Young – Heart of Gold	mp3	2.920kb	1012
103	Rammstein – Ich liebe Neil Young	wma	4.234kb	1014
104	Neil Young – Old Man	mp3	3.161kb	1012
105	Neil Young – Four Strong Winds	wma	5.125kb	1012



Historie: 70er Jahre

- 70er Jahre: Relationale Datenbanksysteme
 - Geräte- und Datenunabhängigkeit
 - Redundanzfrei
 - Konsistent
- Daten in Tabellenstruktur
- Deklarative DML
- Trennung DML/ Programmiersprache

Einleitung

Motivation

Historie

Einsatz



Historie von RDBMS

- Einleitung
- Motivation
- Historie
- Einsatz

- 1970: Ted Codd (IBM)
 - Relationenmodell als konzeptionelle Grundlage relationaler DBS
- 1974: System R (IBM): erster Prototyp eines RDBMS
 - Ca. 80.000 LOC (PL/1, PL/S, Assembler)
 - Anfragesprache SEQUEL
 - Erste Installation 1977
- 1975: University of California at Berkeley (UCB) – Ingres
 - Anfragesprache QUEL
 - Vorgänger von Postgres, Sybase, ...
- 1979: Oracle Version 2



Historie: (80er und) 90er Jahre

Einleitung
Motivation
Historie
Einsatz

- Objektorientierte Datenbanksysteme
 - Daten in komplexen Objektstrukturen (Trennung Objekt und seine Daten)
 - Deklarative oder navigierende DML
 - Oft integrierte Datenbankprogrammiersprache
- Objektrelationale Datenbanksysteme
 - OO-Konzepte erweitern RDBMS
 - Von Herstellern unterstützt (→ SQL 2003)



Aktuelle Entwicklungen

Einleitung

Motivation

Historie

Einsatz

- Unterstützung für spezielle Anwendungen
 - XML-Datenbanken: Verwaltung semistrukturierter Daten (XML-Dokumente)
 - Datenstrommanagementsysteme (kontinuierliche Anfragen auf Datenströmen)
 - Multimediadatenbanken: Verwaltung multimedialer Objekte (Bilder, Audio, Video)
 - Verteilte Datenbanken: Verteilung von Daten auf verschiedene Rechnerknoten
 - Föderierte Datenbanken, Multidatenbanken, Mediatoren: Integration von Daten aus heterogenen Quellen (Datenbanken, Dateien, Web-Quellen)
 - Mobile Datenbanken: Datenverwaltung auf Kleinstgeräten (PDA, Handy, ...)



Einige konkrete Systeme

Einleitung

Motivation

Historie

Einsatz

- (Objekt-)Relationale DBMS
 - Oracle11g, IBM DB2 V.9, Microsoft SQL Server 2008
 - MySQL (www.mysql.org), PostgreSQL (www.postgresql.org), Ingres (www.ingres.com)
- Objektorientierte DBMS
 - Poet, Versant, ObjectStore
- XML-DBMS
 - Tamino (Software AG), eXcelon

Einsatzgebiete

Einleitung

Motivation

Historie

Einsatz

- Klassische Einsatzgebiete: Buchhaltungs- und Katalogisierungsprobleme
 - Viele Objekte
(15000 Bücher, 300 Benutzer, 1000 Ausleihen/Tag, ...)
 - Wenige Objekttypen
(BUCH, BENUTZER, AUSLEIHUNG)
 - Beispiele
 - Buchhaltungssysteme
 - Auftragserfassungssysteme
 - Bibliothekssysteme
 - Personaldatenbanken...
- Aktuelle Anwendungen
 - E-Commerce
 - Entscheidungsunterstützende Systeme (Data Warehouse)
 - Data Mining

Datenbankgrößen

- Einleitung
- Motivation
- Historie
- Einsatz

- **Yahoo! Data Warehouse** **100TB**
Oracle DBMS, Unix
- **WalMart Data Warehouse** **0,5 PB (?)**
NCR TeraData;
Produktinfos (Verkäufe etc.) von 2900 Märkten;
50000 Anfragen/Woche
- **Amazon.com** **25 TB**
Oracle RAC DBMS, Linux



Überblick über den Datenhaltungsteil

- Einleitung
 - Motivation und Grundlagen
 - Architektur von Datenbanksystemen
- Datenbankabfragen
 - Relationenmodell und Relationenalgebra
 - Relationale Datenbanksprachen (SQL)
- Datenbankentwurf
 - ER- und EER-Modell
 - Abbildung von ER-Modellen auf das Relationenmodell
 - Relationaler Entwurf
 - Sprachen zur Datenbankdefinition
- Transaktionsverwaltung
- Anfrageoptimierung
- Datenbankanwendungsentwicklung



Zusammenfassung

- DBS
 - Verwaltung großer Datenmengen
 - Einheitlich für verschiedene Anwendungen
 - Redundanzfrei
 - Effizienz/Performanz (Optimierung)
 - Transaktionskonzept (Mehrbenutzerbetrieb, Fehlerresistenz, Integrität)
 - Datenunabhängigkeit durch 3-Ebenen-Konzept
 - Integrität/ Konsistenz der Daten
 - Zugriffskontrolle (kein unbefugter Zugriff)
 - Datensicherheit/ Dauerhaftigkeit (kein ungewollter Datenverlust)