

1 Einführung

1.1 Was ist eine Datenbank?

Eine sehr grosse integrierte Sammlung von Daten

Modelliert ein reeller Unternehmen

- Entitäten (z.B., Studierende, Kurse)
- Relationen (z.B., Sabine nimmt den Kurs dbII)

Ein Datenverwaltungssystem [Database Management System (DBMS)] ist ein Programm, das konzipiert worden ist, um Datenbanken zu speichern und zu verwalten.

1.2 Wieso DBMS und nicht einfach Dateien?

Anwendungen bewegen grosse Datenmengen zwischen Hauptspeicher und Disks. Für grosse Datenmengen ist dieses Hin und Her eine bedeutende Beschränkung der Effizienz. Deshalb müssen Daten:

- zwischengespeichert sein (Buffering);
- Seitenweise zugegriffen werden.

Weitere wichtige Aspekte der Verwaltung grosser Datenmengen:

- Daten müssen gegen Inkonsistenz geschützt werden
- Weil Daten gleichzeitig von mehrere Benutzer und Benutzerinnen angewendet werden, müssen auch Aspekte der
 - o Crash recovery
 - o Sicherheit und Zugriffsberechtigungenbetrachtet werden

1.3 Was bringt ein DBMS aus der Sicht der Applikation?

- Datenunabhängigkeit - abstrakte Sicht der Daten, siehe nachher;
- Effizienter Zugriff;
- Geringere Entwicklungszeit von Anwendungen;
- Datenintegrität (constraints) ;
- Datensicherheit;
- Zentrale, einheitliche DB Administration;
- Concurrent access;
- Crash recovery.

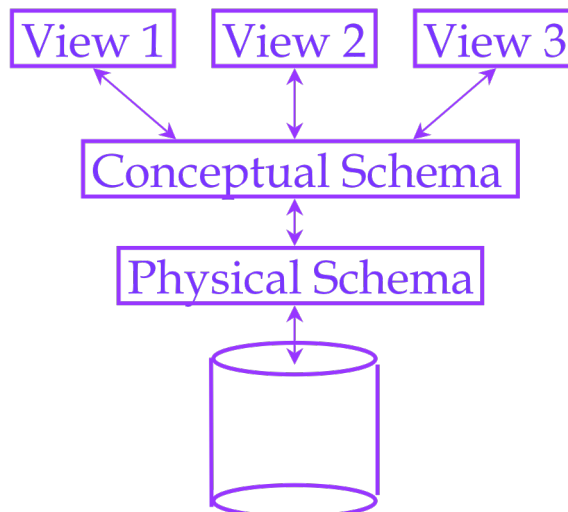
1.4 Datenmodell

- Ein Datenmodell ist eine Sammlung von Konzepte, die Daten beschreiben.
 - o Z.B: Netzwerkmodell, hierarchisches Modell, relationales Modell

- Ein Schema ist die Beschreibung einer bestimmten Sammlung von Daten, unter Anwendung eines bestimmten Datenmodells und seine Datendefinitionssprache.
- Das relationale Datenmodell ist das meistgebrauchte Modell heute.
 - o Hauptkonzept: Relation, im Grunde genommen eine Tabelle mit Zeilen und Kolonnen.
 - o Jede Relation hat ein Schema, das die Zeilen und Spalten beschreibt.

1.5 Abstraktionsstufen

- Verschiedene Sichten, einziges konzeptuelle (logische) Schema und physische Schema.
 - o Sichten beschreiben wie die Benutzer und Benutzerinnen die Daten sehen
 - o Das konzeptuelle Schema definiert die logische Struktur
 - o Das physische Schema beschreibt die benutzten Dateien und Indexen



Figur 1: Abstraktionsstufen der DBMS

☛ Schemas sind mit **Data Definition Language** definiert;

☛ Daten sind mit **Data Manipulation Language** modifiziert/abgefragt

SQL enthält beides DDL und DML

Beispiel: Fachhochschul-Datenbank

- Konzeptuelles Schema:
 - o *Studenten(sid: string, name: string, login: string, alter: integer, gesamtnote:real)*
 - o *Kurse(cid: string, cname:string, credits:integer)*
 - o *Eingeschrieben(sid:string, cid:string, note:string)*
- Physisches Schema:
 - o Relationen sind als unsortierte Dateien gespeichert

- Index auf der ersten Kolonne von „*Studenten*“.
- Externes Schema (View):
 - *Kurs_Info(cid:string,anzahl_anmeldungen:integer)*

1.6 Datenunabhängigkeit [data independence]

Anwendungen sind isoliert von Betrachtungen über wie Daten strukturiert und gespeichert sind

–Logische Datenunabhängigkeit: Schützt gegen Änderungen in der logischen Struktur der Daten (z.B. neue Felder addieren).

–Physische Datenunabhängigkeit: Schützt gegen Änderungen in der physischen Struktur der Daten (z.B., Anordnung, Indizierung, Komprimierung).

Ein der wichtigsten Vorteile der DBMS !

1.7 Gleichzeitige Ausführung von Benutzerprogramme: Concurrency Control

Die gleichzeitige Ausführung von Benutzerprogrammen ist für gute DBMS Performanz unbedingt nötig.

Weil Diskzugriffe häufig und **relativ langsam sind**, es ist wichtig, dass man die CPU mit mehreren Programmen beschäftigt.

Deshalb werden Benutzerprogramme *verschachtelt [interleaved]* ausgeführt

Die Wirkung der Verschachtelung [Interleaving] kann aber auf Inkonsistenzen führen, z.B. eine Geldsumme wird auf ein Konto gutgeschrieben, während der Saldo berechnet wird.

Die DBMS macht es sicher, dass solche Probleme nicht vorkommen können. Die Anwender und Anwenderinnen benutzen das System, wie wenn sie allein auf dem System wären

1.8 Transaktion: die Ausführung eines DB-Programmes

Hauptkonzept ist die Transaktion, eine atomare Sequenz von Datenbank-Handlungen (Lese- und Schreibebefehle).

Jede Transaktion, wenn völlig ausgeführt, muss die DB in einem konsistenten Zustand lassen, wenn die DB in einem konsistenten Zustand vor der Transaktion war.

Benutzer können einfache Integritätseinschränkungen auf den Daten spezifizieren, und der DBMS wird diese Einschränkungen überprüfen.

Aber: der DBMS versteht die Semantik der Daten nicht, d.h. schlussendlich ist die Konsistenz unter der Verantwortung der Anwender!