1 Motivation

The contents of this chapter should actually already be known by everyone in the class. We are using this as a start into the subject so that everyone remembers some basic facts that we need to build on later.

Learning Goals:
- Remembering the definition of DBMSs and their general architecture
- remembering the dependency of database system performance on hardware parameters
- remembering classes of database applications

Definition of a database:
A collection of logically related data (and a description of this data, aka meta data), designed to meet the information needs of an organization.

Definition of a database management system:
A software system that manages a database on a computer.
Sometimes also called database system.

Definition of meta data
Meta data are data which describe other data.

Examples for data and meta data:

Meta data of a WhatsApp message are:
- date/time when the message was sent/delivered/read
- who sent the message to whom
- size and format of the message

The content of the message itself, however, is NOT part of the meta data. The message‘s content is the data.

In a database, data may be stored for instance in the format of tables. The actual data per se consists of the table entries (rows). The pertaining meta data are for instance:
- table names
- column names and the column data types
- the number of table entries (rows)
- the creator of a table
- access rights to the tables
- and so on.

One DBMS can manage several databases.

Example:
In the DBMS by the name MySQL many databases can be created and managed.

For the lab exercises in this class, you will have access to 3 different databases within MySQL:

- kochUniversityDB
- information_schema
- a database carrying your own user name

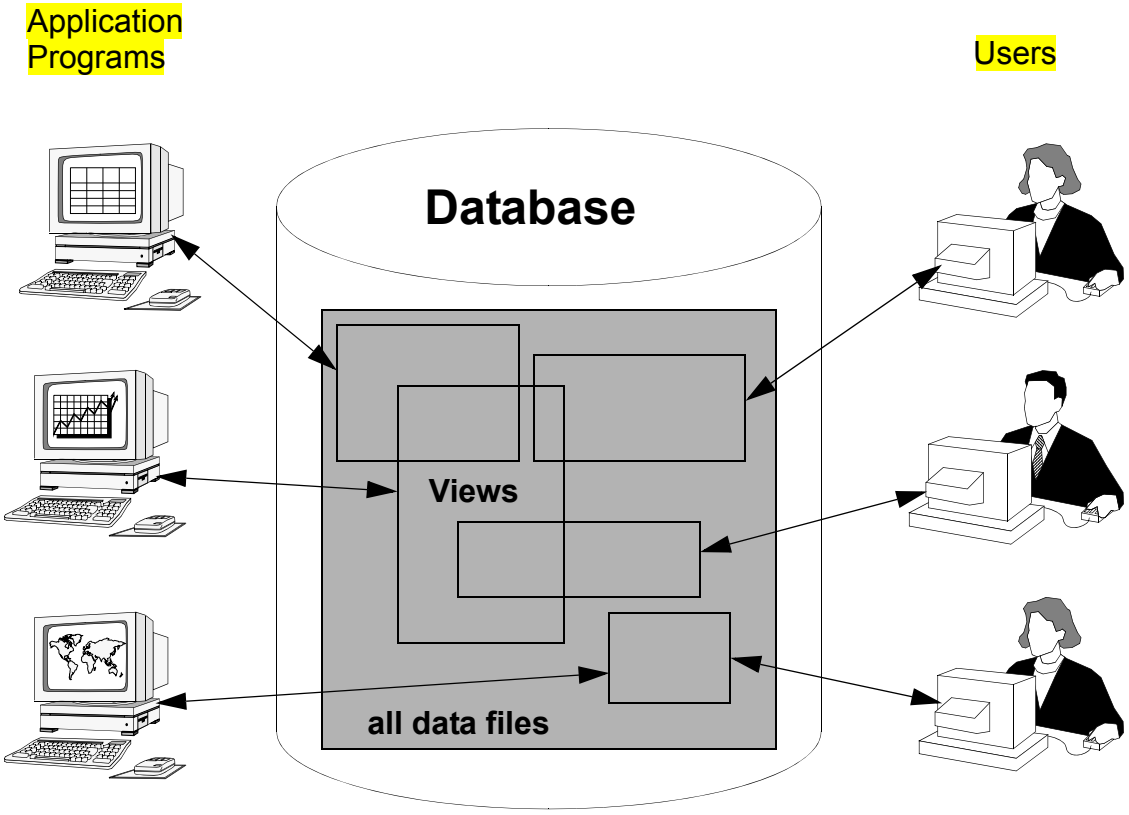
Examples where databases are typically used:

- library: card index
- bank: account information
- insurance company: customers, premiums, claims
- hospital: patients and treatments
- airline: airplanes, seat reservations, crews
- industrial company: product data, bills of materials, customer information
- any company: personnel data
- residents' registrations office: persons and addresses
- university: student data
- many institutions: address data
- ...

Small exercise: Consider which data supposedly should be stored in databases at HFT? At a supermarket chain like Edeka? At companies like Daimler or Porsche? At a small IT company with 3 employees?

For the lab exercises and also to explain concepts in this class, a database with university data will be used in most chapters. You find its description (entity relationship model plus a table schema complete with tuples) in a separate document in the Moodle. It will be useful to keep this file handy for class work. This database is also available in MySQL on our lab server.

A database usually consists of a set of data files.



Important features of a database:

- Integration:
  All database data is stored in one single model that contains everything, with as little redundancy as possible.
- Shared access:
  Several users and applications access the same database. They may use different views of the data. A view is a subset of the data or data that can be derived from the actually stored data or a combination of both.

Redundancy:

Redundancy means that the same information is stored in two or more different places.

Example:
You have stored a friend‘s phone number on your landline phone as well as on your mobile. That is redundant.
Advantages:

- You can quickly access the number in both places. (=> higher availability)
- If one of your devices breaks, you still have the number on the other device. (=> backup).

Disadvantages:

- If your friend changes their number, you need to update it on both devides. (=> more effort)

- If you forget to update the number in all places, you will have contradictory, inconsistent data. This may lead to confusion and mistakes. (=> source of errors)

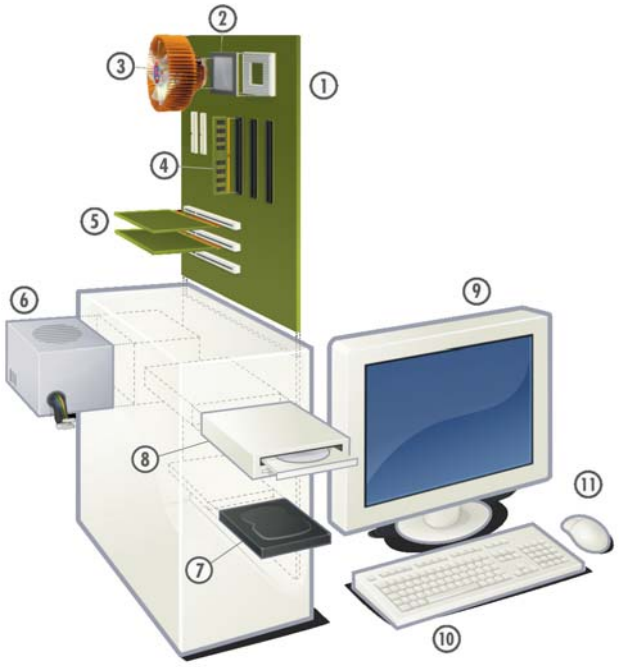
Redundancy thus has two faces:
A negative: it always causes more work to be done in order to prevent mistakes from happening.
A positive: it increases availability and safety of the data.

From this follows:
Redundancy should be used in all situations where its positive aspects are needed. In these cases, we will put up with the extra work. Redundancy should be eliminated in all other situations.

Quiz:
Which situations in the context of database systems do you remember, where redundancy is used or eliminated?

Answer:
- Entity relationship modelling
- foreign keys
- normal forms
- materialized views
- backups
- any query (because data is being copied between secondary storage and memory)
- ...

Computer architecture has a direct influence on DBMS architecture:



1 Hauptplatine
2 Prozessor
3 Kühler
4 Hauptspeicher
5 Grafikkarten
6 Stromanschluss
7 Festplatte
8 CD-ROM-Laufwerk
9 Bildschirm
10 Tastatur
11 Maus

Teile eines Rechners

Hauptplatine
mit allen Bauteilen
(Motherboard)

Hauptspeicher
(Memory)

Prozessor (CPU)

Prozessorkühler

Festplatte
Harddisk, Storage

Secondary storage is:

Festplatte, 2008
Foto: Christian asky, Wikipedia

- persistent (data is still stored even without power)
- cheap (because of the used materials and technology)
- and therefore large
- slow (because of the used technology, sometimes even mechanical movement)

Main memory is:

RAM für Notebook, 2008
Foto von Gemgem, Wikipedia

- volatile (dependant on power supply)
- expensive (because of the used materials and technology)
- and therefore smaller
- fast (because of the materials and technology, no mechanics, just power flowing through circuits)

This implies the following conclusions for database systems:

1. Databases are usually stored on secondary storage (HDD[1] or SSD[2]). This is because they need to be available even when users log out, when a computer is shut down, when there is a power failure, etc.
Moreover, a database typically requires large amounts of storage space and is therefore too big to be kept in main memory.

2. The CPU (and thus all processes running on the computer) can only directly access data that is present in memory. Thus, data needs to be copied from secondary storage into memory, before they can be processed. Newly entered data will be first placed into memory, before it can be copied to disk.
This means

- that in using a database system, data will always be redundant - at least temporarily (in memory and on disk). In order to prevent inconsistencies from happening, a DBMS requires extensive functionality.
- that one fundamental task of a DBMS is to shuffle data between disk and memory (CRUD operations: Create, Retrieve, Update, Delete).

Types of database applications
Database applications can be classified according to the number of their users:

- Personal database
  A system for only one user.
  Examples: Customer contacts maintained by a salesperson. Accounting for a sports club.
- Workgroup database
  A system used by a relatively small group of people, typically less than 25 people, for instance people working on a shared project for which the data is stored in the database. The database resides on a server, the team members have their own computers that are linked to the server and to each other by a LAN.
- Department database
  A system used by a whole department. Departments are typically larger than workgroups and are responsible for a wider range of tasks. Databases are used to support the departmental functions and activities, for instance for personnel, for budgeting information, or for administration of equipment used in the department.
- Enterprise database
  A system comprising data that is relevant enterprise-wide.
  Examples:
  ERP systems (enterprise resource planning: system integrating enterprise-wide functions as human resource, finance, sales, marketing, manufacturing, inventory, accounting)
  Data warehouses (support analysis and decision-making (for instance tracking costs and profits of the organization over given periods of time)).
- Internet / Intranet / Extranet database
  A system that can be accessed via the Internet, an intranet, or an extranet. E-Com-

1. Hard disk drive: a secondary storage device with a rotating hard disk
2. Solid State Disk: a seondary storage device without mechanical parts

merce (electronic shopping, auctions, online travel arrangements, etc.) requires databases.

Type of database	Typical number of users	Typical Architecture	Typical size of database
Personal	1	Desktop, laptop, tablet	Megabytes
Workgroup	5-25	Client/Server (2-tier)	Megabytes - Gigabytes
Department	25-100	Client/Server (3-tier)	Gigabytes
Enterprise	> 100	Client/Server (distributed or parallel server)	Gigabytes - Terabytes
Internet	> 1000	Web server and application servers	Gigabytes - Petabytes or larger