1.3 Fundamental Terms

**Database (DB)**
- integrated and structured repository of large collections of persistent data, which serves for all users of an application area as a common and reliable basis of up-to-date information
- frequently used synonymous term: *data basis*

**Database Management System (DBMS)**
- all-purpose software system, which supports the user in the definition, construction and manipulation of databases for different applications in an application-neutral and efficient manner
- set of programs for the management of and access to the data in the DB
- software level between physical database and user

**Database System (DBS)**
- DBS = DBMS + DB
Data Model

- mathematical formalism consisting of a notation for describing the data of interest and of a set of operations for manipulating these data
- description of the structure of a database (data types, relationships, conditions)
1.4 Data Model

- a data model offers facilities
  - for the specification of data objects
  - for the specification of the relationship between data and
  - for the specification of operations on data objects together with their semantics

- usually a DBS has at least two data models
  - **physical data models** for the storage-oriented representation of data
  - **logical data models** for the user-oriented representation of data

- logical data models
  - object-based, e.g.
    - entity-relationship model
    - object-oriented data model
    - object-relational data model
  - record-based, e.g.
    - object-relational data model
    - relational data model
    - network data model
    - hierarchical data model
1.5 Data Abstraction: The Three-Level Model

DBS has several abstraction levels:

- **external/view levels** describe the part of the DB, which is relevant for the user.
- **conceptual/logical level** gives information about existing data and relationships in the DB.
- **physical/internal level** describes how data are physically stored.

**Database schema and state**

- A schema describes the structure/the design of a DB.
- A state describes a concrete instance of a DB.
1.6 Data Independence

- denotes the property that higher levels of the model are not influenced by changes of lower levels

- **logical data independence**
  - changes of the conceptual schema (e.g., information about new types of entities, further information about existing entities) do not have impact on external schemas (e.g., existing application programs)
  - example: extension of the class data for an additional boolean value expressing whether the merits required for a master thesis have been performed

- **physical data independence**
  - changes of the physical schema (e.g., change of an access structure to a more efficient one, use of other data structures, exchange of algorithms) do not have impact on the conceptual schema and thus also not to external schemas
  - example: the class data, which are so far stored in an unsorted file, are to be reorganized in a B-tree to enable efficient access by “registration number”.