Technische Hochschule Rosenheim

Chapter 5 – Database design process

Databases lectures

Prof. Dr Kai Höfig

Chapter 3: Database design

Objective of database design

- The better the design of the data storage, the longer the data can be used in the coming years and decades.
  - Change in the applications/application scenarios
  - Change in the non-functional requirements, e.g. regarding performance, availability or distribution
  - Particular importance of avoiding redundancy to prevent anomalies and save storage space.

- In addition to the question of HOW data is stored as sensibly as possible, there is also the question of WHAT should be stored
  - Too little information might not meet a company's future information needs
  - Storing all the data that arises might be interesting (big data, machine learning), but expensive.

Database design

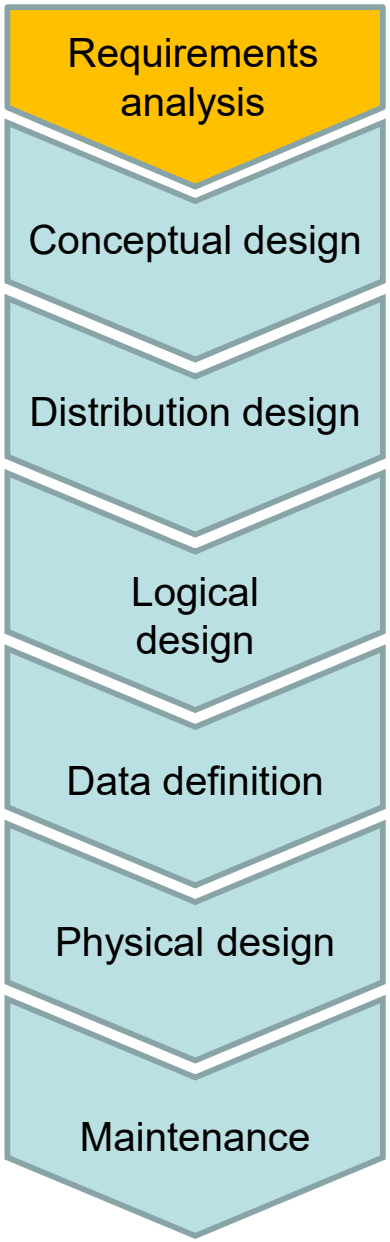
- Database design
  - Sequence of design documents (modelling)
  - Starting with abstract, application-oriented description level
  - Ending with the actual realisation of a database
  - Design step: from one model to the next (manual, automatic or semi-automatic)

- Most important characteristics of design steps
  1) Information preservation
     All data that could be saved in the previous model can also be saved in the new model
  2) Consistency preservation
     Rules and restrictions in the previous model can also be ensured in the new model

Quality criteria for design documents

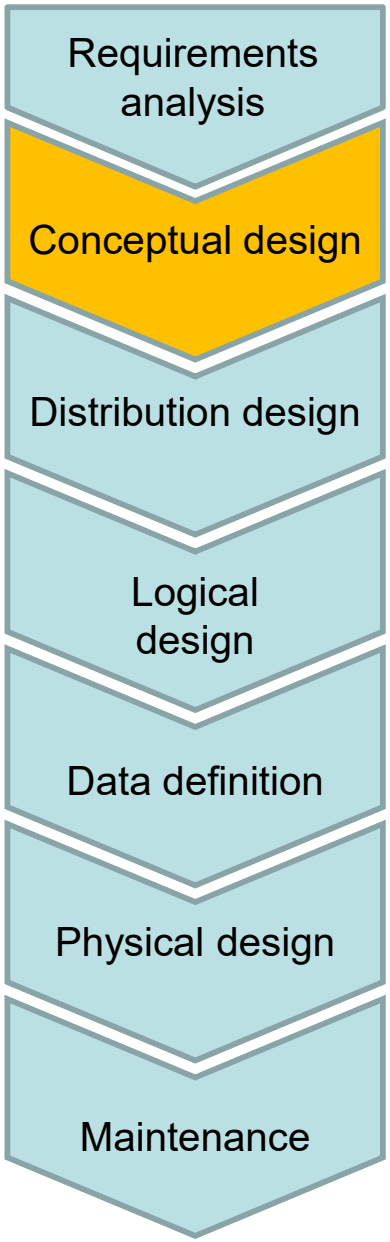
- No redundancy
- Completeness with regard to requirements analysis
- Consistency of the description document
- Expressiveness, comprehensibility of the formalism used (formal semantics of the descriptive constructs)
- Readability of the documents
- Other quality characteristics: extensibility, modularisation, reusability, tool support, etc.

Phases of database design



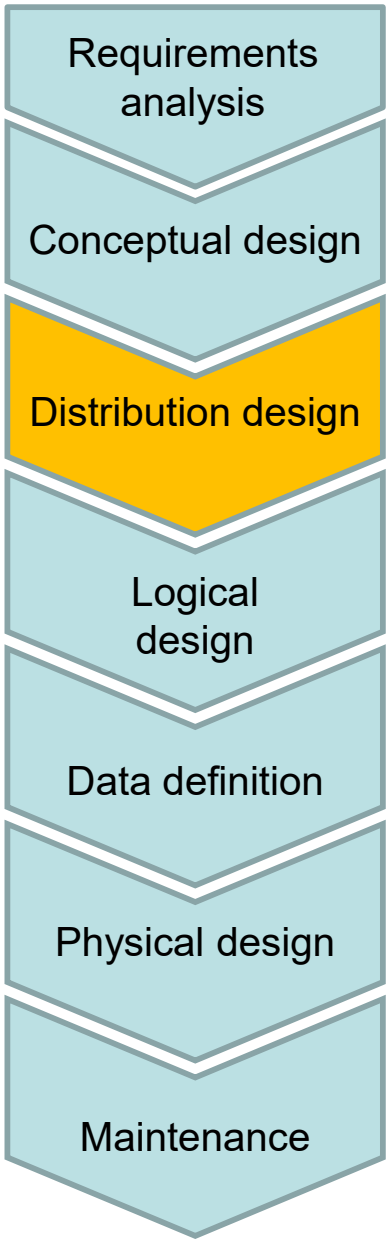
- Aim: Collect and analyse the requirements for the database system to be realised
- Method: Collecting the information requirements in the specialist departments
- Results:
  - Informal description (texts, tabular lists, forms, etc.) of the specialist problem
  - Separation of the information about data (data analysis) from the information about functions (functional analysis)
- Only touched on during these lectures, for typical methods see the **Software Engineering** lecture

Phases of database design



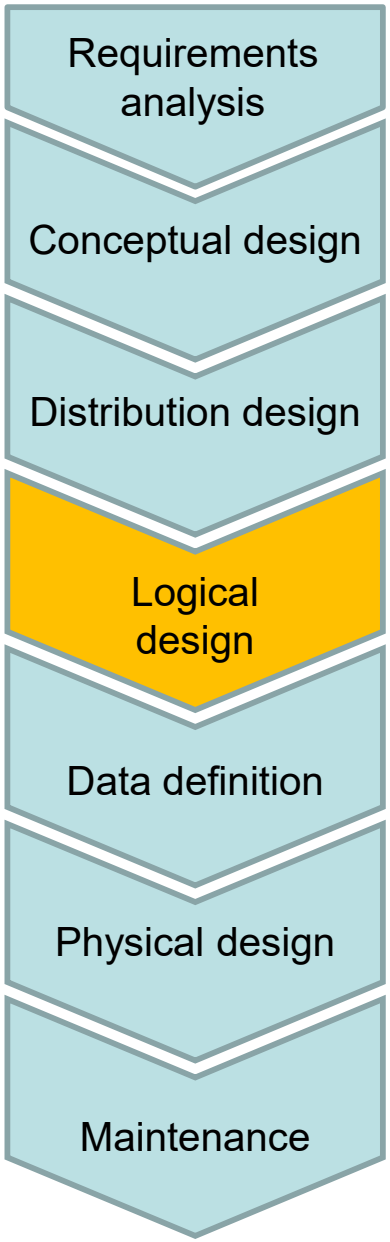
- Aim: Initial formal description of the specialist problem, *regardless* of the system to be used later
- Linguistic devices: Abstract (semantic) data model
- Method:
  - Modelling views, e.g. for different specialist departments
  - Analysis/comparison of the existing views with respect to conflicts, such as
    - Naming conflicts: homonyms / synonyms
    - Type conflicts: different structures for the same element
    - Value range conflicts: different value ranges for an element
    - Constraint conflicts, e.g. different keys for an element
    - Structural conflicts: the same issue is expressed by different constructs
  - Integration of the views into a consistent and contradiction-free overall schema
- Result: Conceptual data model, typically an ER or UML diagram
- **Integral part of these lectures**

Phases of database design



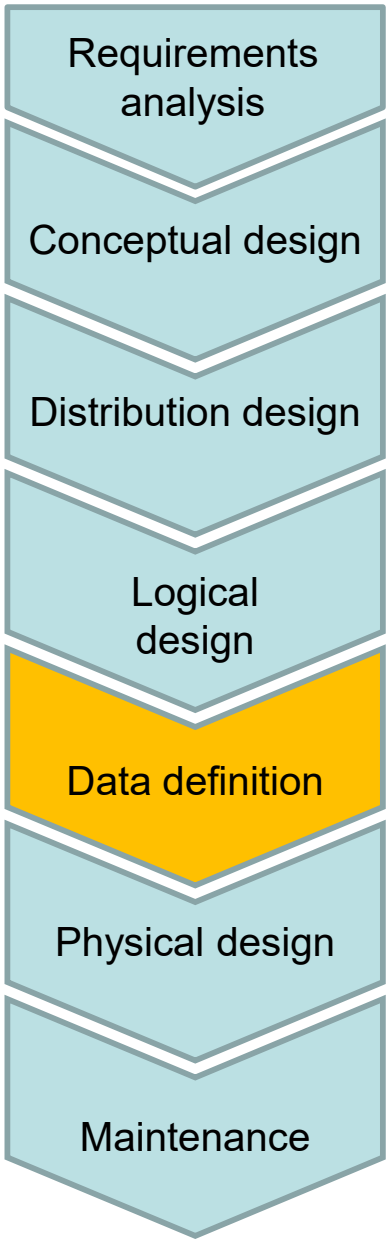
- Optional
- If data is to be distributed on several computers, the type and method of distributed storage must be defined
- Maybe only in the context of the physical design
- Not considered further in these lectures, see the **Data warehouse** lecture

Phases of database design



- Method:
  - (Automatic) transformation of the conceptual schema
    e.g. ER → relational model
  - Improvement of the relational schema based on quality criteria
    (keyword: normalisation, see later)
    Design goals: avoiding redundancy, . . .
- Linguistic devices: Data model of the selected DBMS implementation, e.g. relational model
- Result: Logical schema, e.g. collection of relational schemas
- **Integral part of these lectures**

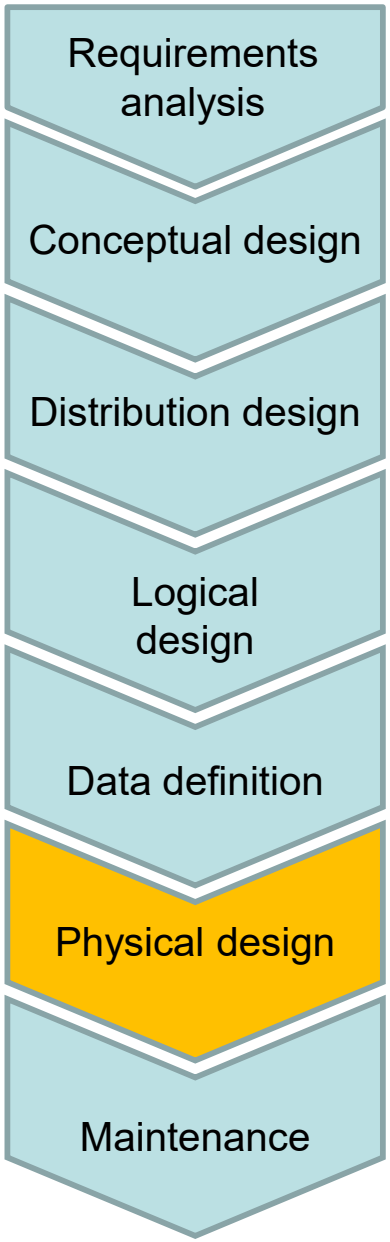
Phases of database design



- Aim: Implementation of the logical schema in a specific schema
- Linguistic devices: DDL and DML of a DBMS, e.g. Oracle, IBM DB2, Microsoft SQL Server
  - Database declaration in the DDL of the DBMS
  - Implementation of the integrity assurance
  - Definition of the user views

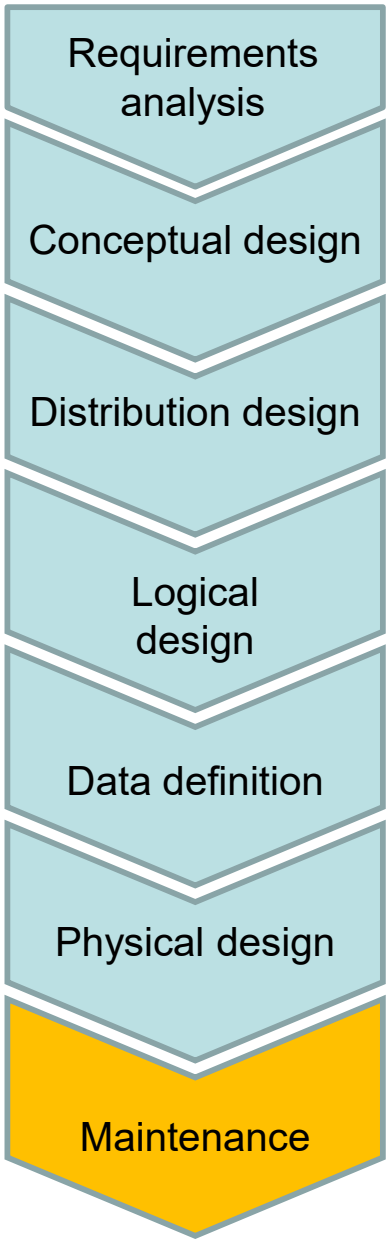
- **Integral part of these lectures**

Phases of database design



- Aim: Complement the physical design with access support for efficiency improvement, e.g. definition of indexes
- Index
  - Access path: data structure for additional key-based access to tuples (<key attribute value, tuple address>)
  - mostly realised as a B* tree
- Linguistic devices: Storage structure language (e.g. as part of SQL)
- Necessity for access paths
  - Example: table with 100 GB data, hard drive transfer rate approx. 50 MB/s.
  - Operation: search for a tuple (selection)
  - Implementation: sequential search
- **Basics as part of the lectures**

Phases of database design



- Typical challenges of the maintenance phase:
  - Further optimisation of the physical layer
  - Adaptation to new requirements and system platforms
  - Porting to new database management systems
  - etc.
- Not considered further in these lectures

Methods supporting the phases

Methods supporting the phases to ensure design requirements are met (especially information and consistency preservation)

- **Verification**: the formal proof of aspects such as schema properties
- **Prototyping**: working with the database on a trial basis before the final implementation
- **Validation with test data**: verification of design correctness using real or artificial test data

Summary

- Database design as a **sequence of design steps** (modelling)
- Start with the abstract requirements of the specialist departments, work towards a specific, implemented DBS
- Phases of database design
  - Requirements analysis
  - Conceptual design
  - Distribution design
  - Logical design
  - Data definition
  - Physical design
  - Maintenance