EE4202 Database Systems

Dr. Nadeesha Sandamali

Module coordinator, Lecturer

Email address: Nadeesha@eie.ruh.ac.lk

INTRODUCTION

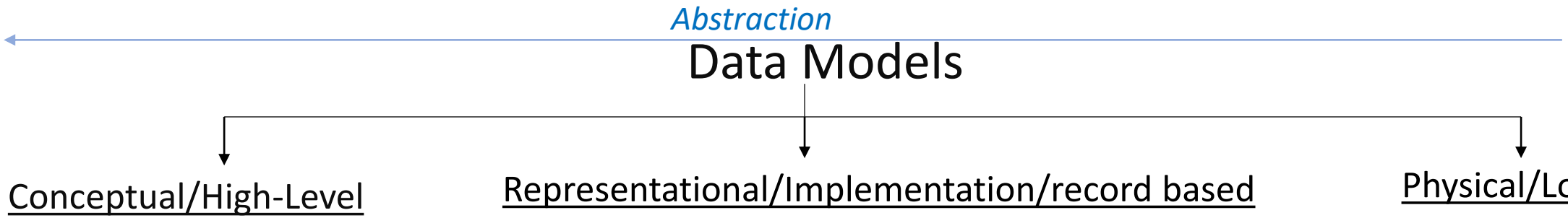
- A **Database** is a collection of data, typically describing the activities of one or more related organizations.
- For example, a university database might contain information about the following:
  - ❑Entities such as students, lecturers, courses, and classrooms.
  - ❑Relationships between entities, such as students' enrollment in courses, lecturers teaching courses, and the use of rooms for courses.
- A **Database Management System**, or DBMS, is a software designed to assist in maintaining and utilizing data.
- Four major functions of a DBMS are;
  - ❑ Add data
  - ❑ Delete data
  - ❑ Update data
  - ❑ Retrieve data

File system versus a DBMS

- To understand the need for a DBMS, let us consider the following scenario: A company has a large collection (say, 500 GB) of data on employees, departments, products, sales, and so on. This data is accessed concurrently by several employees. Questions about the data must be answered quickly, changes made to the data by different users must be applied consistently, and access to certain parts of the data (e.g., salaries) must be restricted. We can try to deal with this data management problem by storing the data in a collection of operating system files. This approach has many drawbacks.
- We probably do not have 500 GB of main memory to hold all the data. We must therefore store data in a storage device such as a disk or tape and bring relevant parts into main memory for processing as needed.
- We have to write special programs to answer each question that users may want to ask about the data. These programs are likely to be complex because of the large volume of data to be searched.
- We must protect the data from inconsistent changes made by different users accessing the data concurrently. If programs that access the data are written with such concurrent access in mind, this adds greatly to their complexity.
- We must ensure that data is restored to a consistent state if the system crashes while changes are being made.
- Operating systems provide only a password mechanism for security. This is not sufficiently flexible to enforce security policies in which different users have permission to access different subsets of the data.

DATA MODELS

- Data abstraction is suppressing storage and organization details while only highlighting essential features.
- Describes the structure (data types, relationships, constraints, behavior, operations) of a database.



- This model is more close to how users perceive data.
- Eg**: Entity - relationship**, object data model
- An entity is a real world object/concept and relationship is an association between the entities.
- This data model is used for database design

- Eg: **Relational (Tabular)**, **NoSQL - Hierarchical/Aggregation (Tree), Network/graph**
- Relational data models is a fully structured data models where relational model has no child nor parent records.
- Aggregation data model is a semi-structured data model having one parent record and multiple child records.
- Network data model – A record can have multiple parent records and child records.

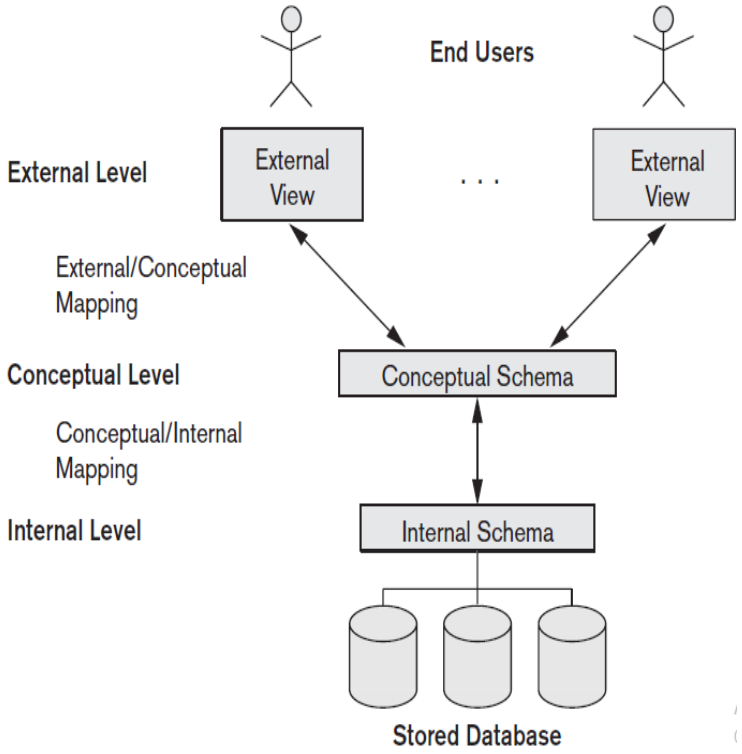
- Represent how data is stored in computer storage

  Eg: **files and access paths**
- Access path is a structure for searching database records efficiently

SCHEMAS AND INSTANCES

- **Database schema** is the description of the database. It is a design aspect which is not expected to change unless required. Schema describes the real-world objects/concepts (entity), their properties (attributes) and data types, association (relationship) between entities, constraints (limitations) etc.
- The database schema is stored in the database as meta-data.
- The set of data of database at a particular time is known as **database state/instance**. The database instance can be changed from time to time when new data is added, or existing data is deleted.

THREE SCHEMA ARCHITECTURE



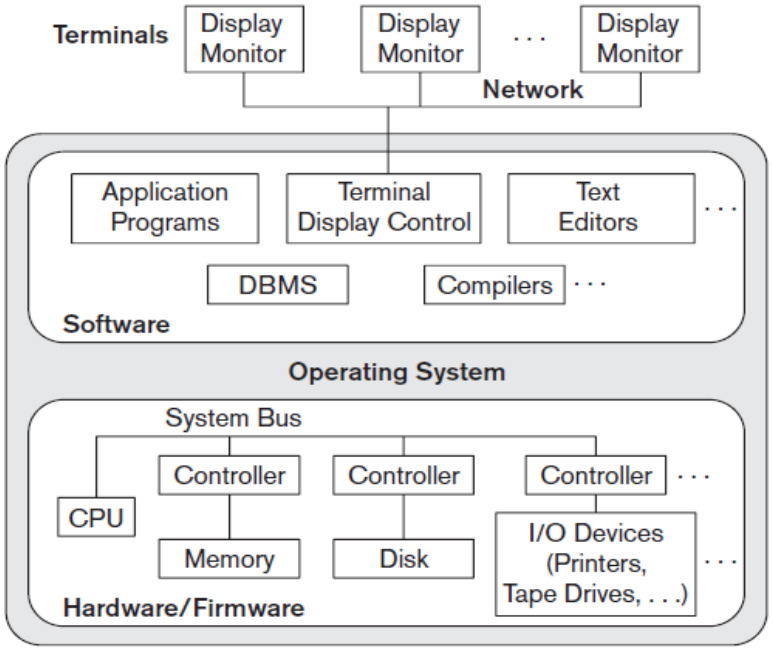
1. Internal schema: Uses the *physical data model* to describe the complete details of data storage and access paths.
2. Conceptual schema: Describe the database using *representational data model* hiding physical implementation details.
3. External schema/User Views: Describe the part of database using representational data model which the *particular user is interested/privileged*.

- **Data Independence** - A lower level schema can be changed without affecting the data of higher level schemas.
  - ❑ *Physical data independence*: Files can be reorganized and access paths can be changed in internal schema without affecting the conceptual schema.
  - ❑ *Logical data independence*: Structural and constraint changes in conceptual schema should not affect external views or application programs referring them.

DATABASE ARCHITECTURES

Centralized Architecture

- Database Management System functionality, application program execution and user interface processing are done in a single computer
- Used when users have some processing and memory resources



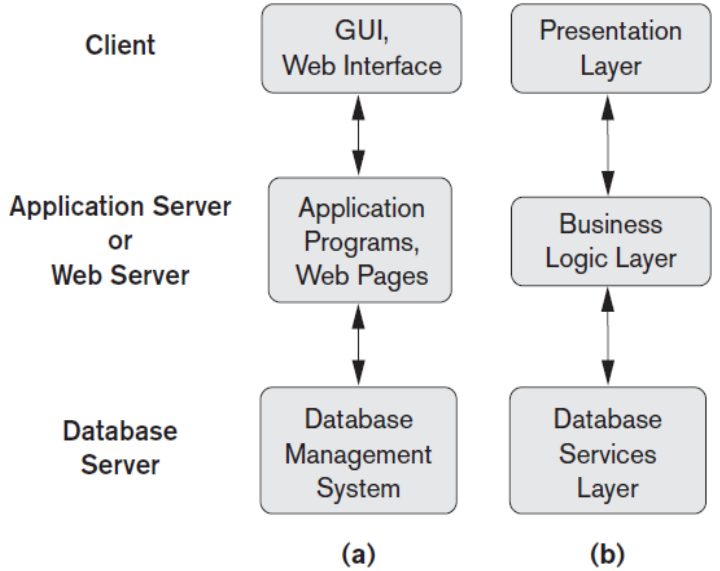
Client – Server Architecture

2 Tier

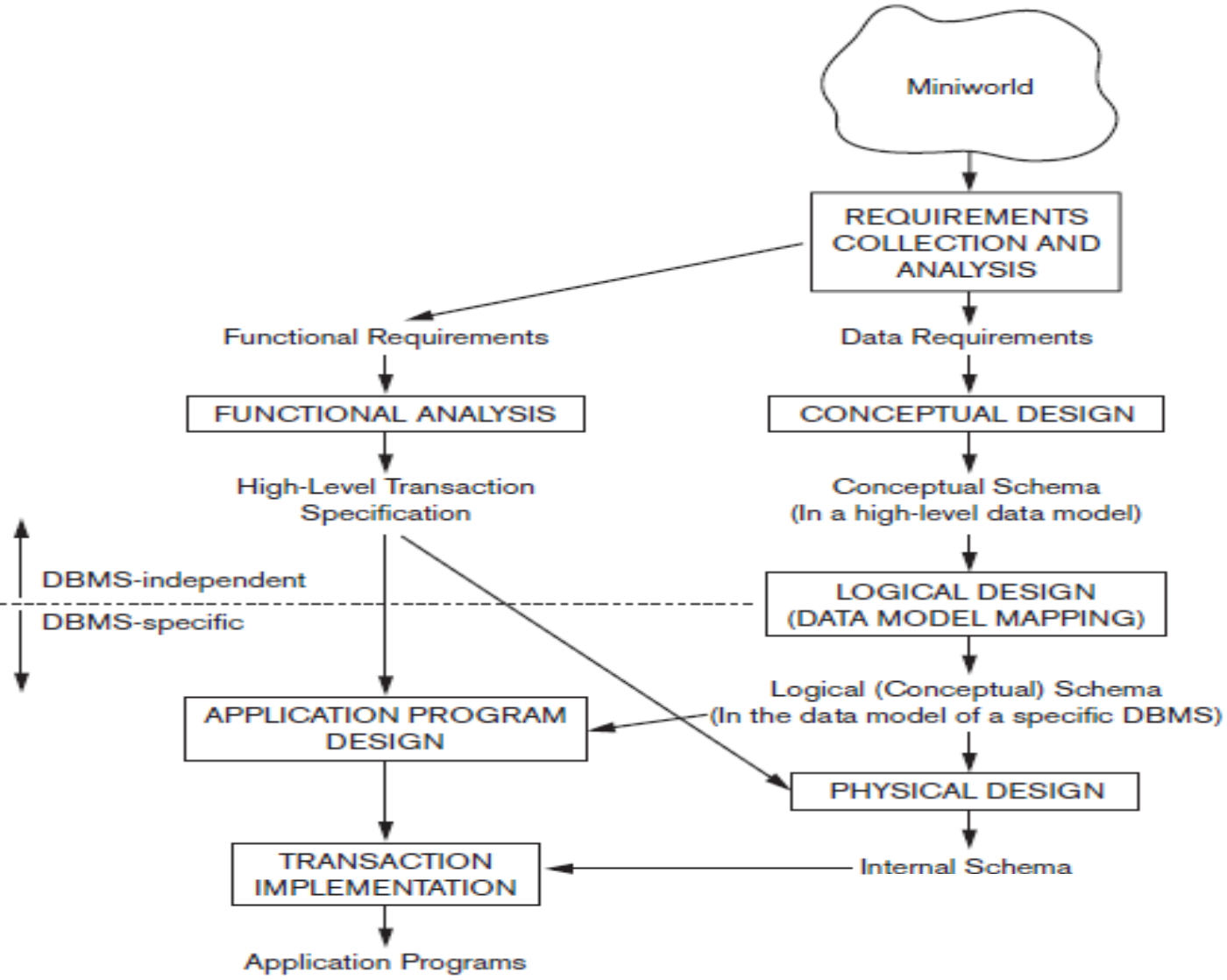
- Database Management System functionality, application program execution and user interface processing are shared between the client and the server.
- Used when users lack processing and memory resources
- The server mainly functions as a data server storing data.

3 Tier

- Additional Application/web server exists between the client and the server.
- Suitable for web applications.
- The middle layer stores business rules



DATABASE DESIGN PROCESS



- First, functional and data requirements of the user must be identified.
- Functional requirements are the user defined operations (transactions - queries etc.) such as database retrievals and updates.
- Based on data requirements, conceptual design is done using a high level data model like ER data model.
- After the conceptual schema is created, it should be mapped to a *implementation data model* like relational data model using a particular DBMS (like MySQL). It is still conceptual/logical schema in 3 schema architecture.
- Next, internal schema is derived from transaction specification and logical schema.
- Application programs are designed using transaction specification.
- Transactions are implemented using the internal schema and application programs.

REFERENCES

- Ramez Elmasri, Shamkant Navathe. Fundamentals of Database Systems, Boston: Pearson/Addison Wesley,2007, ISBN-10: 0133970779 ISBN- 13:9780133970777
- Garcia-Molina, Hector. Database systems: the complete book. Pearson Education India, second edition, 2008; ISBN-10:0131873253 ISBN- 13:9780131873254
- Pramod J. Sadalage and Martin Fowler, “NoSQL distilled”, ISBN-10:0321826620
- Thomas M. Connolly, Carolyn E. Begg. Database Systems: A practical approach to design, implementation and management, fourth edition, 2005; ISBN-10: 0321210255 ISBN-13: 978-0780134410951