SIE04 – Sistem Basis Data

Capaian Pembelajaran Mata Kuliah

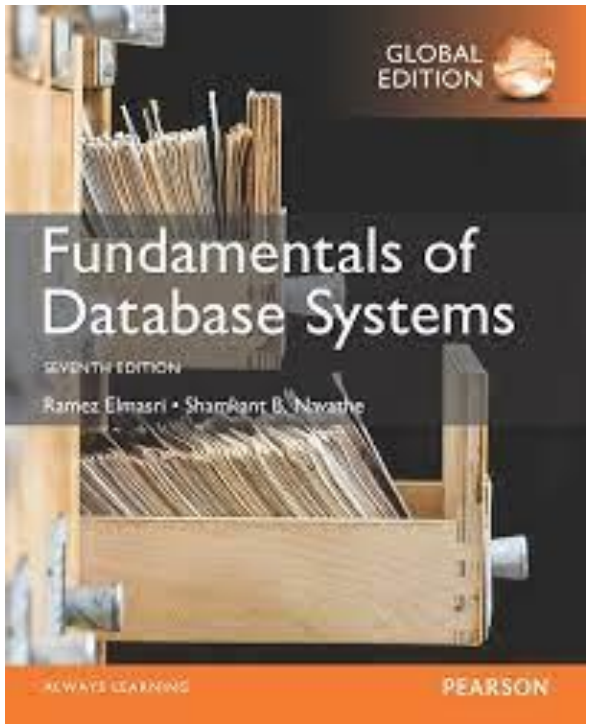
- Mahasiswa mampu menjelaskan tentang komponen utama sistem database, fungsi utama DBMS, aplikasi sistem database, dan berbagai arsitektur database pada lingkungan sistem basis data (C2, A1).
- Mahasiswa mampu menggunakan perintah dan fungsi dasar SQL untuk membuat database dengan perintah DDL, DML, query lanjutan, penggabungan data dan subquery (C3, A2).
- Mahasiswa mampu menyusun rancangan database sesuai kebutuhan organisasi (C4, A3).

Referensi

Silberschatz, Abraham 2020. *Database System Concepts,* 7th edition. McGraw Hill.

Referensi (Cont.)

Elmasri, Ramez. Navathe , Shamkant B., 2017. *Fundamentals of Database Systems,* 7th edition. Pearson.



Arsitektur dan Konsep Sistem Database

Sesi 1 & 2

Sub - CPMK

Mahasiswa mampu menjelaskan tentang komponen utama sistem database, fungsi utama DBMS, aplikasi sistem database, dan berbagai arsitektur database pada lingkungan sistem basis data (C2, A2)

Materi

1. Data
2. Database System
3. Simplified database system environment
4. Components of DBMS Environment
5. Purpose of Database System
6. Database Applications Examples
7. Data Model
8. Three-Schema Architecture
9. Data Independence
10. Centralized and Client-Server DBMS Architectures

1. Data

Data

- Data are raw facts, meaning facts that have not been processed for meaning.

- Information is the result of processing raw data to derive meaning. To get meaning, information needs context.

Data (Cont.)

- Data must be properly formatted for storage, processing and presentation.

- Examples:
  - Respondents' answers in the form of **Yes/No** need to be converted into **Y/N** or **0/1** format for storage.
  - Dates must be stored in Julian calendar format in the database, but can be displayed in various formats, such as **dd-mm-yyyy** or **mm-dd-yyyy**

2. Database System

Database System

- A database is an integrated computer structure that stores a collection of :
  - End user data: raw facts needed by end users.
  - Metadata, or data about data, where end user data is aggregated and managed.

- Metadata describes the characteristics of the data and the relationships that relate the data found in the database.

Database System (Cont.)

- Database Management System (DBMS) is a collection of programs that manage the structure of the database and control access to the data stored in the database.

- A database resembles a highly organized electronic cabinet where there is software (DBMS) that manages the contents of the cabinet.

Database System (Cont.)

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use

Database System (Cont.)

- Database systems are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple users and applications, often at the same time.

Database System (Cont.)

- A modern database system is a complex software system whose task is to manage a large, complex collection of data.

- Databases touch all aspects of our lives.

3. Simplified Database System Environment

Simplified Database System Environment

- User / Programmer
- Database System
- DBMS Software

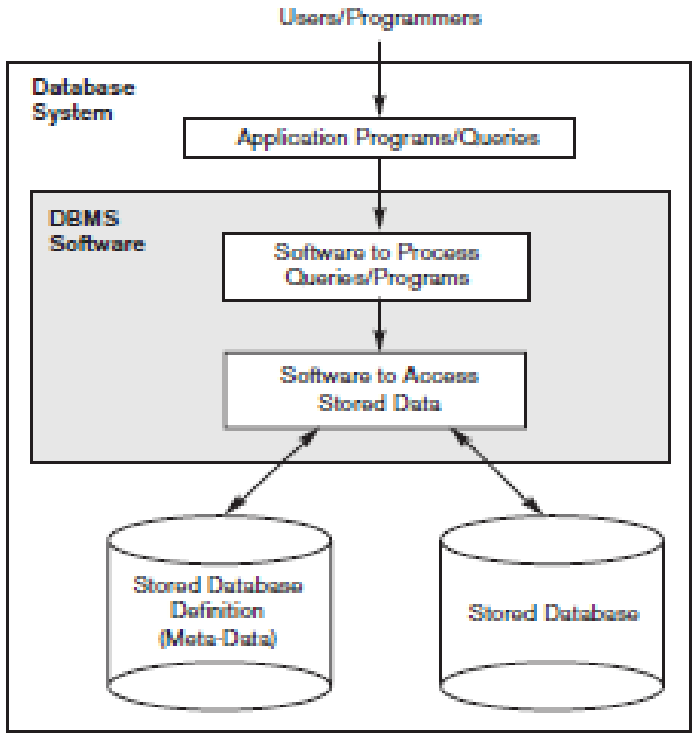


Figure 1.1
A simplified database system environment.

Source : Elmasri, Ramez. 2017. Fundamentals of Database Systems. 7th edition., Pearson (Chapter 1, Page 37)

4. Components of DBMS Environment

Components of DBMS Environment

- Hardware
  - Can range from a PC to a network of computers.
- Software
  - DBMS, operating system, network software (if necessary) and also the application programs.
- Data
  - Used by the organization and a description of this data called the schema.

Components of DBMS Environment (Cont.)

- Procedures
  - Instructions and rules that should be applied to the design and use of the database and DBMS.

Components of DBMS Environment (Cont.)

- People
  - Administrator system
  - Administrator database
  - Designer database
  - System Analyst & programmer
  - End user

5. Purpose of Database System

Purpose of Database Systems

In the early days, database applications were built directly on top of file systems, which leads to:

- Data redundancy and inconsistency: data is stored in multiple file formats resulting induplication of information in different files.

- Difficulty in accessing data
  - Need to write a new program to carry out each new task.

Purpose of Database Systems (Cont.)

- Data isolation
  - Multiple files and formats.

- Integrity problems
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly.
  - Hard to add new constraints or change existing ones.

Purpose of Database Systems (Cont.)

- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out.
  - Example: Transfer of funds from one account to another should either complete or not happen at all.

Purpose of Database Systems (Cont.)

- Concurrent access by multiple users
  - Concurrent access needed for performance.
  - Uncontrolled concurrent accesses can lead to inconsistencies.
    - Ex: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time.

Purpose of Database Systems (Cont.)

- Security problems
  - Hard to provide user access to some, but not all, data

6. Database Applications Examples

Database Applications Examples

- Enterprise Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.

- Manufacturing: management of production, inventory, orders, supply chain.

Database Applications Examples (Cont.)

- Banking and finance
  - customer information, accounts, loans, and banking transactions.
  - Credit card transactions.
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data.
- Universities: registration, grades.

Database Applications Examples (Cont.)

- Airlines: reservations, schedules.
- Telecommunication: records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards.

Database Applications Examples (Cont.)

- Web-based services
  - Online retailers: order tracking, customized recommendations
  - Online advertisements
- Document databases

Database Applications Examples (Cont.)

- Navigation systems: For maintaining the locations of varies places of interest along with the exact routes of roads, train systems, buses, etc.

7. Data Model

Data Model

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints

- Relational model.

Data Model (Cont.)

- Entity-Relationship data model (mainly for database design).
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)

Data Model (Cont.)

- Other older models:
  - Network model
  - Hierarchical model

8. Three-Schema Architecture

Three-Schema Architecture

- Proposed to support DBMS characteristics of:
  - **Program-data independence.**
  - Support of **multiple views** of the data.

- Not explicitly used in commercial DBMS products, but has been useful in explaining database system organization

Three-Schema Architecture (Cont.)

- Defines DBMS schemas at ***three*** levels:
  - **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
    - Typically uses a **physical** data model.

Three-Schema Architecture (Cont.)

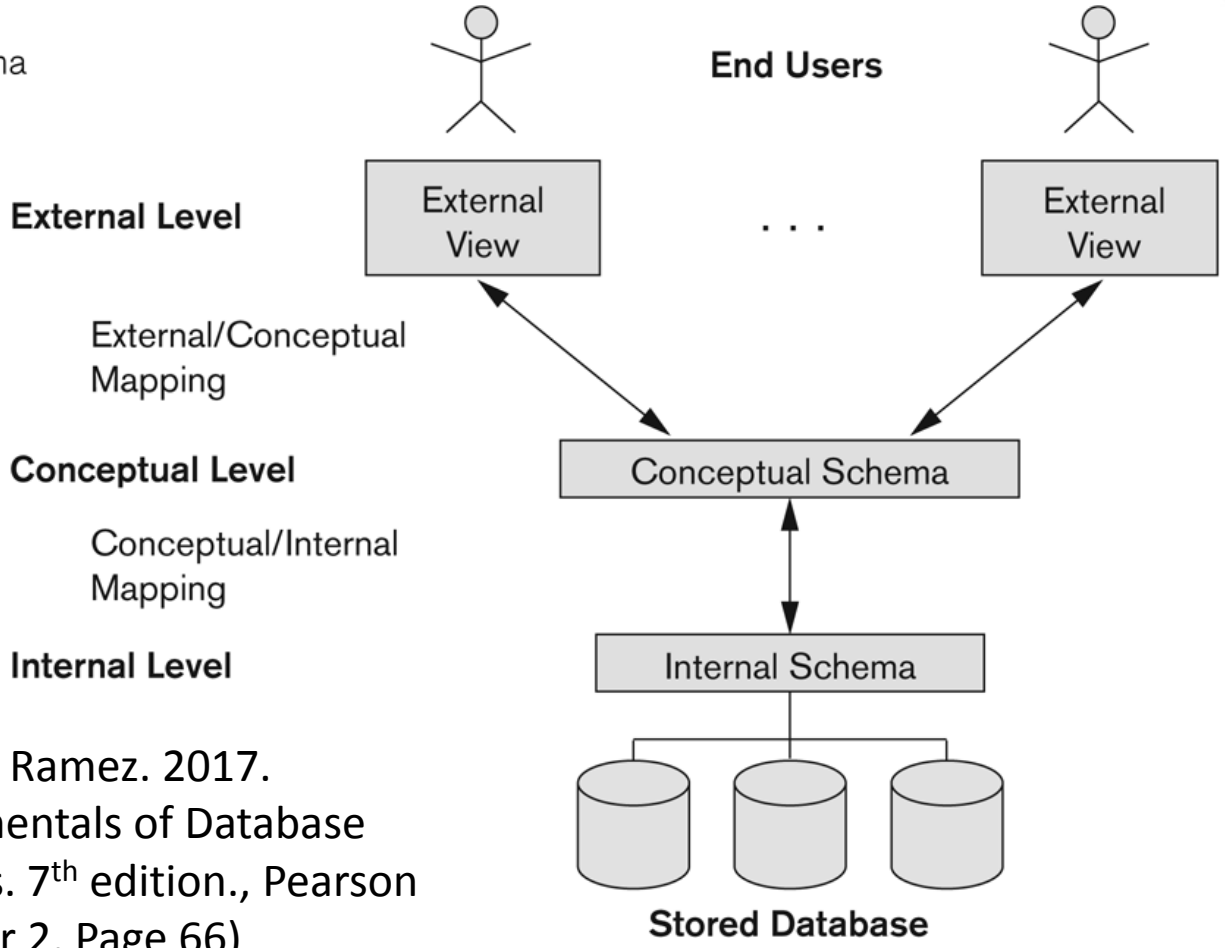
- **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
  - Uses a **conceptual** or an **implementation** data model.

Three-Schema Architecture (Cont.)

- **External schemas** at the external level to describe the various user views.
  - Usually uses the same data model as the conceptual schema

Three-Schema Architecture (Cont.)

Figure 2.2
The three-schema architecture.



Source : Elmasri, Ramez. 2017. Fundamentals of Database Systems. 7th edition., Pearson (Chapter 2, Page 66)

Three-Schema Architecture (Cont.)

- Mappings among schema levels are needed to transform requests and data.
  - Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.
  - Data extracted from the internal DBMS level is reformatted to match the user's external view (e.g. formatting the results of an SQL query for display in a Web page).

9. Data Independence

Data Independence

- **Logical Data Independence:**
  - The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.

Data Independence (Cont.)

- **Physical Data Independence:**
  - The capacity to change the internal schema without having to change the conceptual schema.
  - For example, the internal schema may be changed when certain file structures are reorganized or new indexes are created to improve database performance

Data Independence (Cont.)

- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence.

- The higher-level schemas themselves are **unchanged**.
  - Hence, the application programs need not be changed since they refer to the external schemas.

10. Centralized and Client-Server DBMS Architectures

10.1 Centralized and Client-Server DBMS Architectures

- Centralized DBMS:
  - Combines everything into single system including-DBMS software, hardware, application programs, and user interface processing software.
  - User can still connect through a remote terminal – however, all processing is done at centralized site.

10.2 A Physical Centralized Architecture

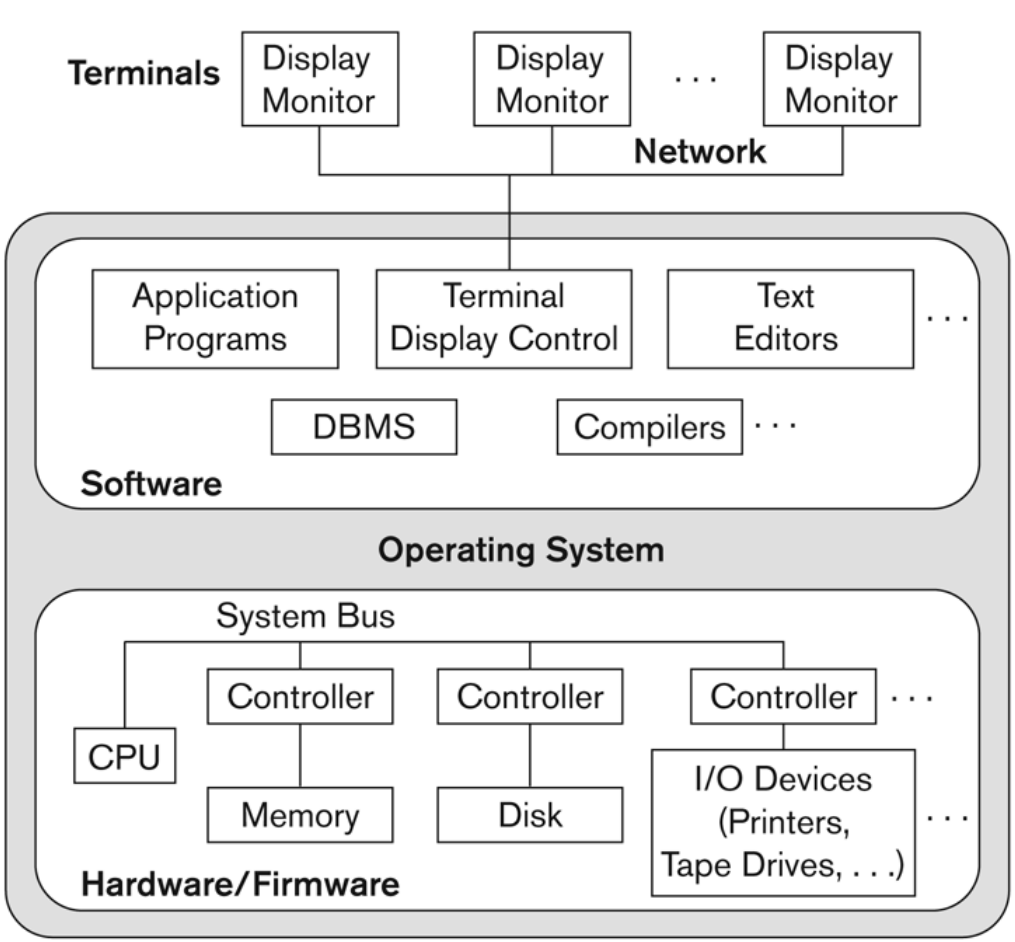


Figure 2.4
A physical centralized architecture.

Source : Elmasri, Ramez. 2017. Fundamentals of Database Systems. 7th edition., Pearson (Chapter 2, Page 77)

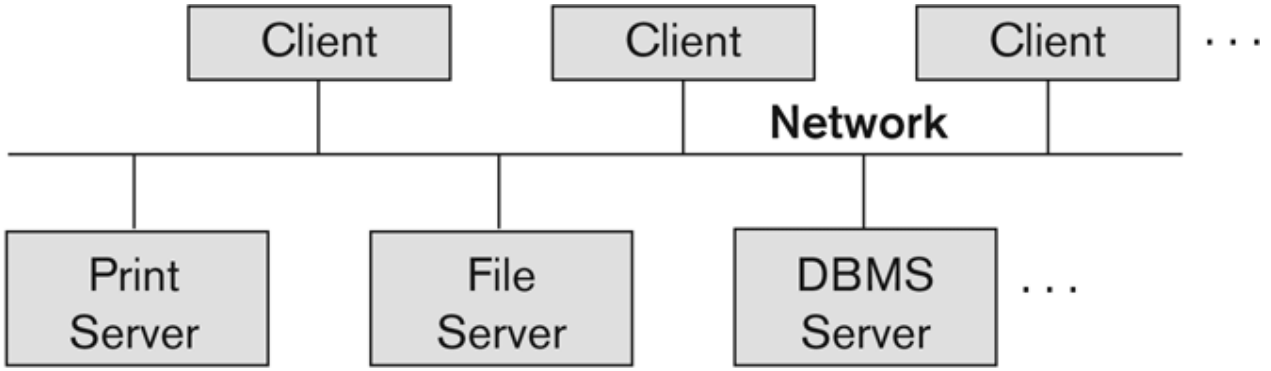
10.3. Basic 2-tier Client-Server Architectures

- Specialized Servers with Specialized functions
  - Print server
  - File server
  - DBMS server
  - Web server
  - Email server

- Clients can access the specialized servers as needed.

10.3. Basic 2-tier Client-Server Architectures (Cont.)

Figure 2.5
Logical two-tier client/server architecture.



Source : Elmasri, Ramez. 2017. Fundamentals of Database Systems. 7th edition., Pearson (Chapter 2, Page 78)

10.3. Basic 2-tier Client-Server Architectures (Cont.)

Clients

- Provide appropriate interfaces through a client software module to access and utilize the various server resources.

- Clients may be diskless machines or PCs or Workstations with disks with only the client software installed.

10.3. Basic 2-tier Client-Server Architectures (Cont.)

Clients

- Connected to the servers via some form of a network.
  - (LAN: local area network, wireless network, etc.)

10.3. Basic 2-tier Client-Server Architectures (Cont.)

DBMS Server

- Provides database query and transaction services to the clients.

- Relational DBMS servers are often called SQL servers, query servers, or transaction servers.

10.3. Basic 2-tier Client-Server Architectures (Cont.)

DBMS Server

- Applications running on clients utilize an Application Program Interface (**API**) to access server databases via standard interface such as:
  - ODBC: Open Database Connectivity standard
  - JDBC: for Java programming access

10.3. Basic 2-tier Client-Server Architectures (Cont.)

DBMS Server

- Client and server must install appropriate client module and server module software for ODBC or JDBC.

10.4. Two Tier Client-Server Architecture

- A client program may connect to several DBMSs, sometimes called the data sources.

- In general, data sources can be files or other non-DBMS software that manages data.

10.4. Two Tier Client-Server Architecture (Cont.)

- Other variations of clients are possible: e.g., in some object DBMSs, more functionality is transferred to clients including data dictionary functions, optimization and recovery across multiple servers, etc.

10.4. Two Tier Client-Server Architecture (Cont.)

- Common for Web applications.

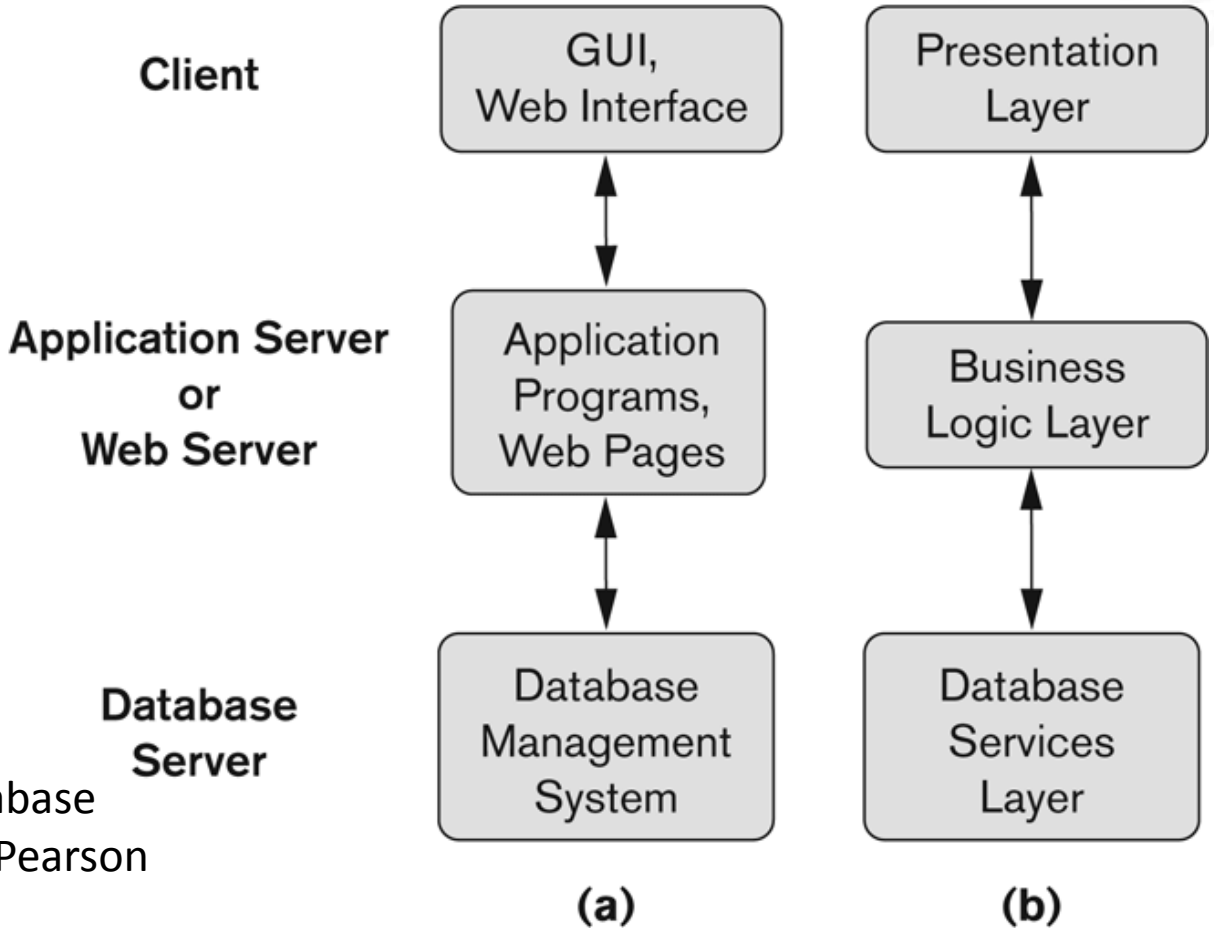
- Intermediate Layer called Application Server or Web Server:
  - Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server
  - Acts like a conduit for sending partially processed data between the database server and the client.

10.4. Two Tier Client-Server Architecture (Cont.)

- Three-tier Architecture Can Enhance Security:
  - Database server only accessible via middle tier
  - Clients cannot directly access database server

10.4. Two Tier Client-Server Architecture (Cont.)

Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.



Source : Elmasri, Ramez. 2017. Fundamentals of Database Systems. 7th edition., Pearson (Chapter 2, Page 80)

Summary

- Data are raw facts, meaning facts that have not been processed for meaning.

- Database Management System (DBMS) is a collection of programs that manage the structure of the database and control access to the data stored in the database.

- Components of DBMS Environment: Hardware, Software, People, Procedure, Data.

Summary (Cont.)

- Three-Schema Architecture :
  - Internal Schema
  - Conceptual Schema
  - External Schema

- Simplified database system environment :
  - User
  - Database System
  - DBMS Software

Thank You