DATABASE SYSTEM CONCEPTS AND ARCHITECTURE

CHAPTER 2

LECTURE OUTLINE

- Data Models
- Three-Schema Architecture and Data Independence
- Database Languages and Interfaces
- The Database System Environment
- DBMS Architectures
- Classification of Database Management Systems

DATA MODEL

- Collection of concepts that describe the structure of a database
- Provides means to achieve data abstraction
  - Suppression of details of data organization and storage
  - Highlighting of the essential features for an improved understanding of data
- Includes basic operations
  - Retrievals and updates on the database
- Dynamic aspect or behavior of a database application
  - Allows the database designer to specify a set of valid operations allowed on database objects

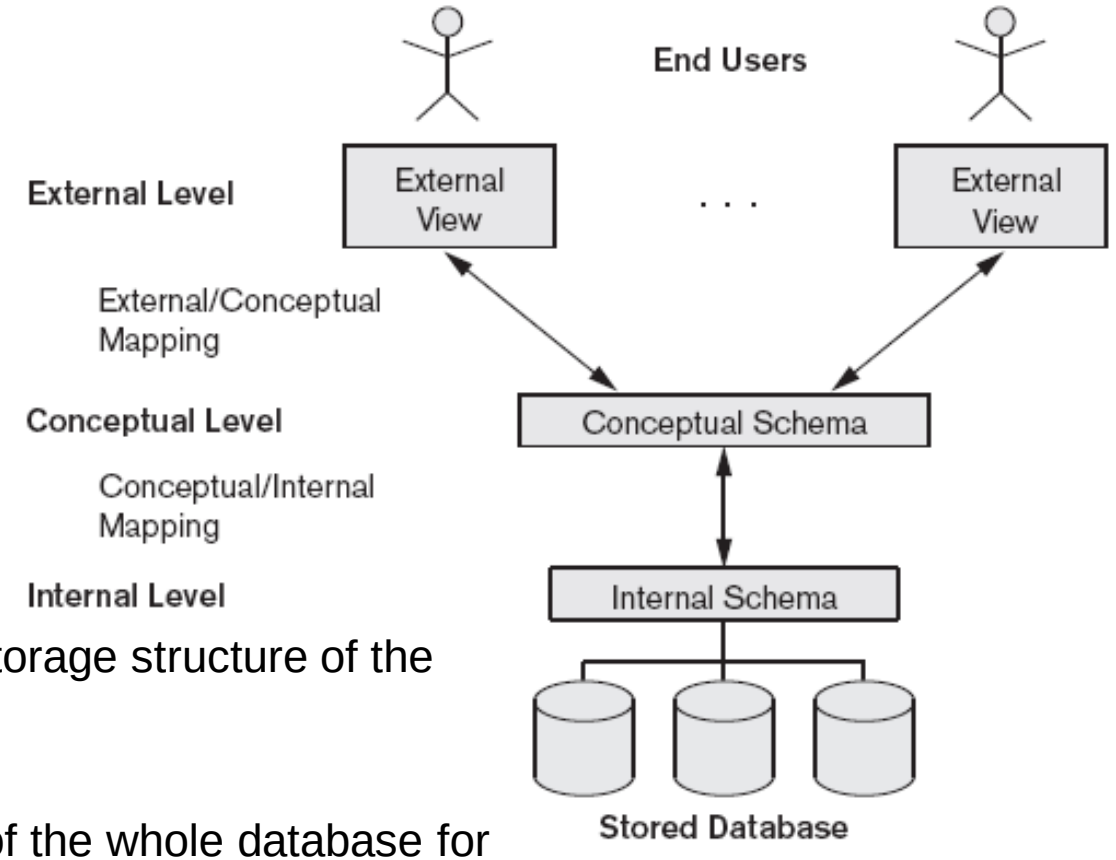
CATEGORIES OF DATA MODELS

- **High-level** or **conceptual** data models
  - Close to the way many users perceive data
  - For example, object-oriented models
- **Low-level** or **physical** data models
  - Describe the details of how data is stored on computer storage media
  - Include explicit access paths
    - Structure that makes locating particular database records efficient
    - Example: Index
      - Allows direct access to record by looking up a value
- Compromise: **Representational** data models
  - Abstract model of data
  - Emphasize aspects that should be understood by end users
  - Close enough to how data organized in computer storage that they can be implemented efficiently

Where does the relational data model fit?

THREE-SCHEMA ARCHITECTURE

Figure 2.2
The three-schema architecture.



- **Internal** level
  - Describes physical storage structure of the database
- **Conceptual** level
  - Describes structure of the whole database for the complete community of users
- **External** or **view** level
  - Describes part of the database of interest to a particular user group

DATA INDEPENDENCE

- Capacity to change the schema at one level of a database system without having to change the schema at the next higher level
  - Change the mappings between schemas
- Conceptual schema reflects the enterprise
  - Relatively stable
  - Serves as **Universe of Discourse**
  - **Physical** data independence achieved through conceptual/internal mapping
  - **Logical** data independence achieved through external/conceptual mappings

DBMS LANGUAGES

- **Data definition language (DDL)**
  - Defines conceptual schema
- **Storage definition language (SDL)**
  - Specifies the internal schema
- **View definition language (VDL)**
  - Specifies user views/mappings to conceptual schema
- **Data manipulation language (DML)**
  - Allows retrieval, insertion, deletion, modification
  - **Low-level** or **procedural** DML
    - Must be embedded in a general-purpose programming language
    - **Record-at-a-time**
  - **High-level** or **non-procedural** DML
    - Can be used on its own to specify complex database operations concisely
    - **Set-at-a-time** or **set-oriented**

DBMS INTERFACES

- Menu-based interfaces for Web clients or browsing
- Forms-based interfaces
- Graphical user interfaces
- Natural language interfaces
- Speech input and output
- Interfaces for parametric users
- Interfaces for the DBA

DBMS COMPONENT MODULES

- Stored data manager
- DDL compiler
- Interactive query interface
  - Query compiler
  - Query optimizer
- Precompiler
- Runtime database processor
- System catalog
- Concurrency control system
- Backup and recovery system

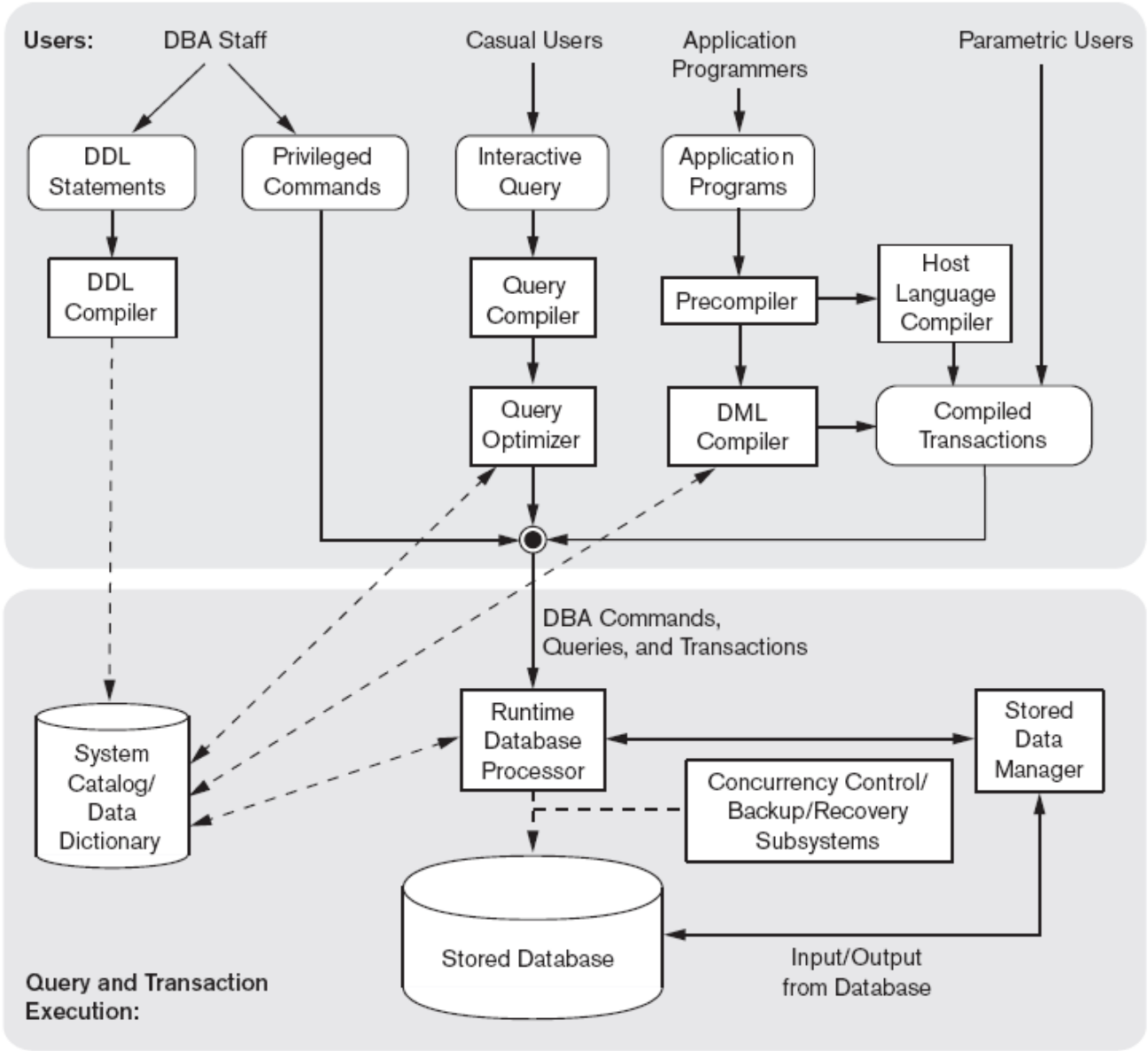


Figure 2.3
Component modules of a DBMS and their interactions.

DBMS UTILITIES

- **Loading**
  - Load existing data files
- **Backup**
  - Creates a backup copy of the database
- **Database storage reorganization**
  - Reorganize a set of database files into different file organizations
- **Performance monitoring**
  - Monitors database usage and provides statistics to the DBA

DBMS-RELATED FACILITIES

- CASE Tools
- **Data dictionary** (**data repository**) **system**
  - Stores design decisions, usage standards, application program descriptions, and user information
- **Application development environments**
- **Communications software**

CENTRALIZED DBMS ARCHITECTURE

- All DBMS functionality, application program execution, and user interface processing carried out on one machine

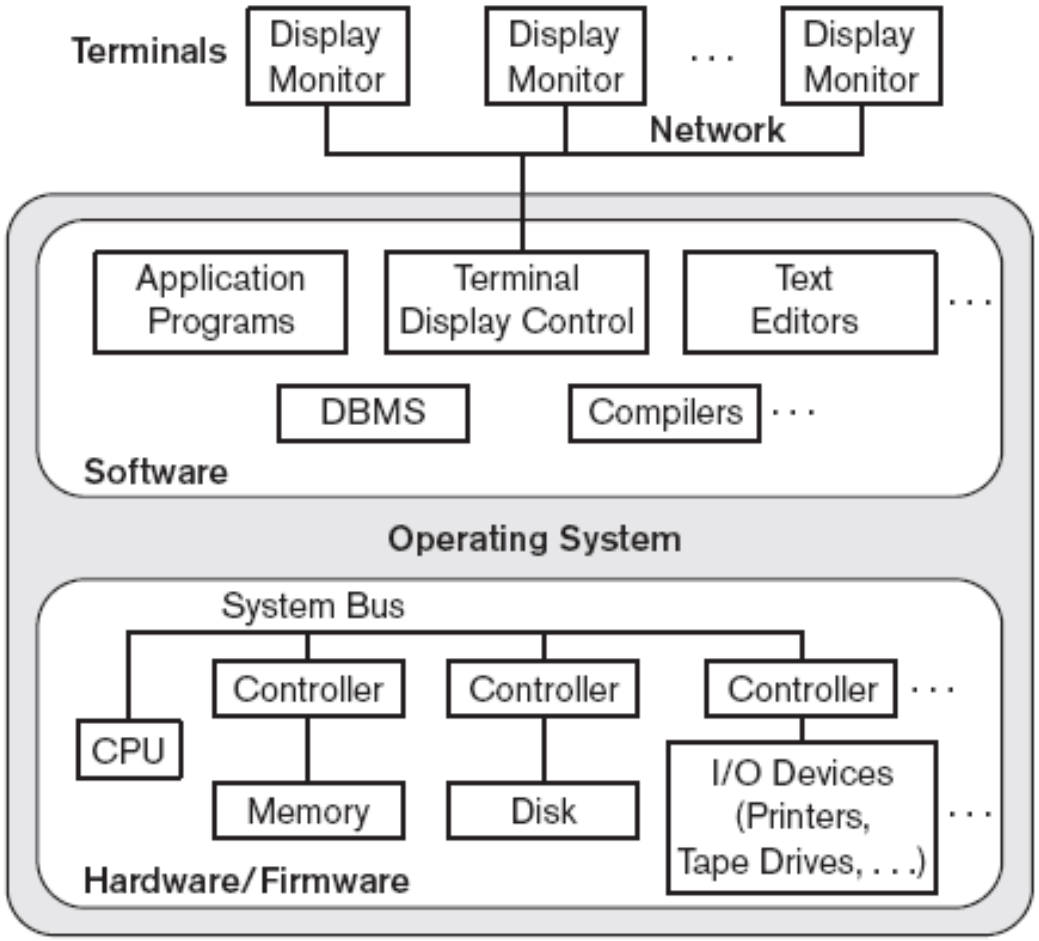


Figure 2.4
A physical centralized architecture.

BASIC CLIENT/SERVER ARCHITECTURES

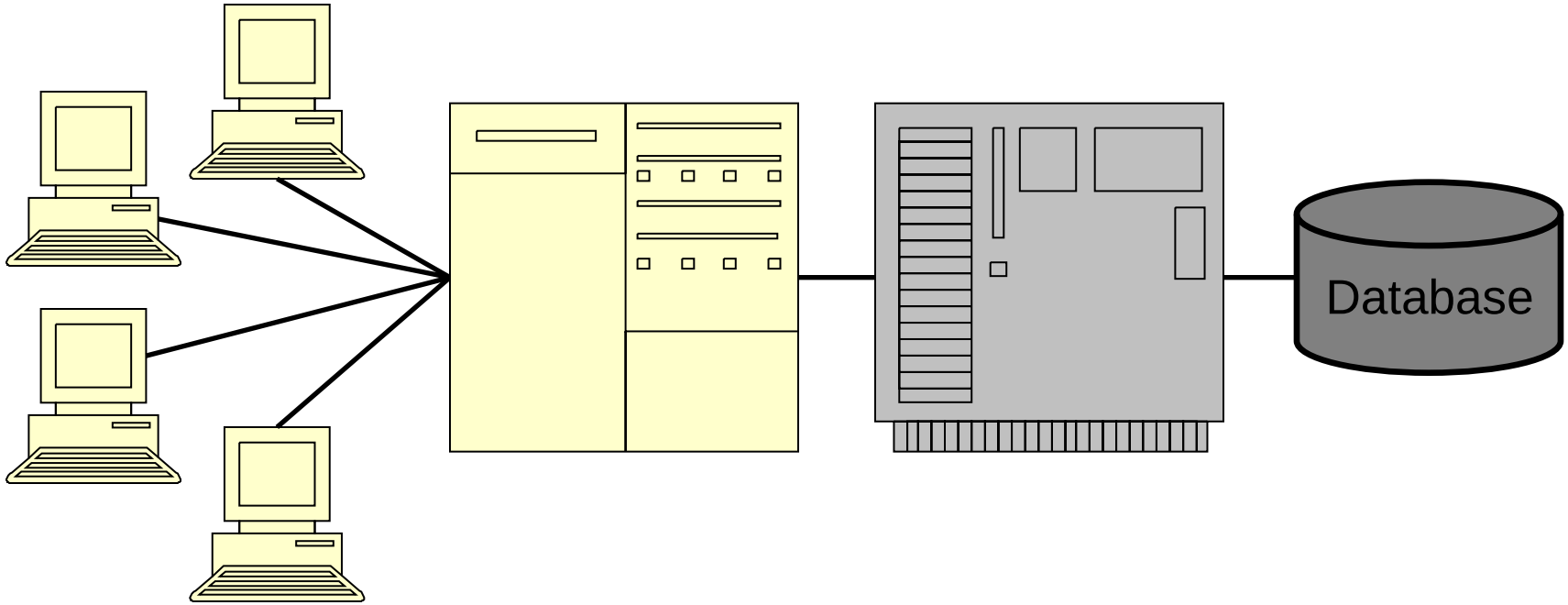
- **Servers** with specific functionalities
  - **File server**
    - Maintains the files of the client machines.
  - **Printer server**
    - Connected to various printers; all print requests by the clients are forwarded to this machine
  - **Web servers** or **e-mail servers**
- **Client machines**
  - Provide user with:
    - Appropriate interfaces to use these services
    - Local processing power to run local applications
- DBMS example
  - Server handles query, update, and transaction functionality
  - Client handles user interface programs and application programs

CLIENT/SERVER DBMS SOFTWARE

- **Open Database Connectivity** (**ODBC**)
  - Provides application programming interface (API) for C and C++
    - Call Level Interface (CLI)
  - Allows client-side programs to call the DBMS
    - Both client and server machines must have the necessary software installed
- **JDBC**
  - Allows Java client programs to access one or more DBMSs through a standard interface
- Alternative: Microsoft's **ADO.NET**

3-TIER CLIENT-SERVER DBMS ARCHITECTURE

- Code partitioned between clients (user interfaces), application server, and DBMS modules



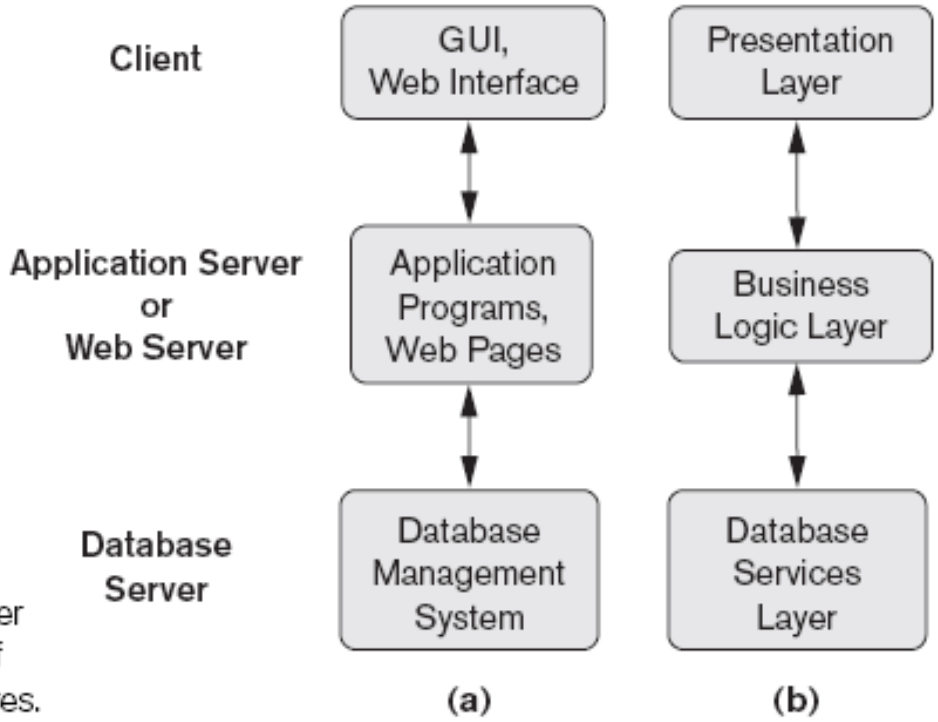


Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

CLASSIFICATION OF DBMSS

- General or special-purpose
- Data model
  - Relational
  - Object
  - Object-relational
  - Hierarchical and network (legacy)
  - Native XML
- Number of users
  - Single-user
  - Multiuser
- Number of sites
  - Centralized
  - Distributed
    - Homogeneous
    - Heterogeneous (Federated)
- Licensing
  - Open source
  - Proprietary

LECTURE SUMMARY

- Main categories of data models
- Three-schema architecture
- Types of languages and interfaces supported by DMBSs
- Components and services provided by the DBMS
- DBMS computing architectures
- DBMS classification criteria