Fundamentals of Database Systems

Chapter 2. Database System Concepts and Architecture

Myungsun Kim

Information_Security@The_University_of_Suwon

2017, Fall

Contents

§2.1 Data Models, Schemas, and Instances

- The main characteristics of the database approach
  - Self-describing nature of a database system
  - Insulation between programs and data, and **data abstraction**
  - Multiple views of the data
  - Sharing of data and multiuser transaction processing

Data abstraction

1. Suppression of details of data organization and storage
2. Highlighting of the essential features for an improved understanding of data

Data Model

A collection of concepts that can be used to describe the structure of a database

§2.1 Data Models, Schemas, and Instances

Structure of a database

A structure of a database consists of the data types, relationships, and constraints that apply to the data.

- Data abstraction vs. Data model
  - Data model offers various necessary tools to get the data abstraction
  - Data model uses logical concepts (e.g., relations, tables, and objects) that are easier for general users
- Two aspects of abstraction
  1. Program-data independence
  2. Program-operation independence

  $\Longrightarrow$ Data abstraction = Program-(data,operation) independence

§2.1.1 Categories of Data Models

- Classification by the types of concepts
  1. Conceptual (or high-level) data model
     - Example. Entity, attribute, & relationship (in Chap. 7 & 9)
  2. Physical (or low-level) data model
- Classification by representation
  1. Hierarchical model
  2. Network model
  3. Object-oriented model
  4. **Relational model** $\Leftarrow$ our target

§2.1.2 Schemas, Instances, and Database State

Database schema

The description of a database

Database state

The data in the database at a particular moment in time.

Valid state

A state that satisfies the structure and constraints specified in the schema

- STUDENT Schema

STUDENT

Name	Student_number	Class	Major

- A valid STUDENT state

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

- An Invalid STUDENT state

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	17	2	CS

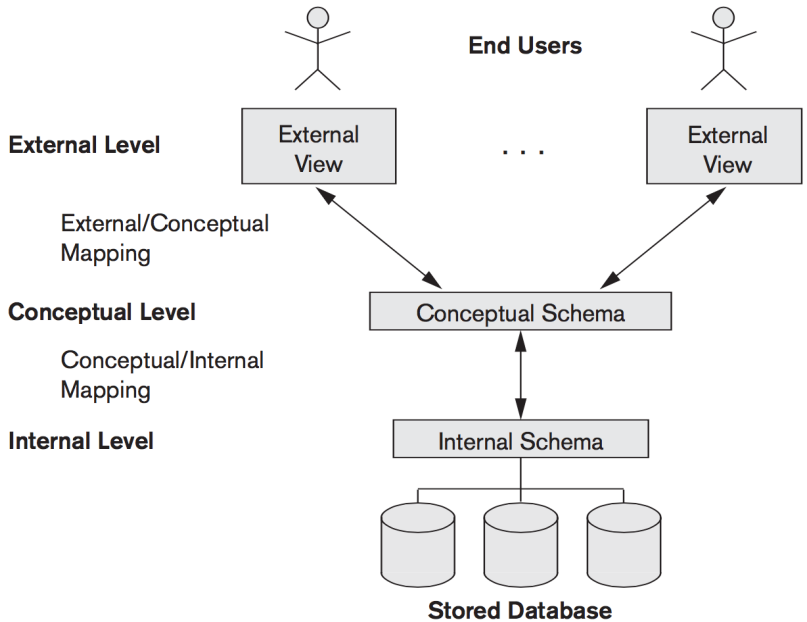
§2.2 Three-Schema Architecture and Data Independence

§2.2.1 The Three-Schema Architecture

Three-schema architecture

An architecture for database systems that allows

- Self-describing
- Data abstraction
- Multiple user views



- Internal schema: describes the physical storage structure of the database, using a physical data model
- Conceptual schema: describes the structure of the whole database for a community of users, hiding physical details
- View (external) schema: describes the part of the database that a particular user group is interested in

§2.2.2 Data Independence

Logical data independence

- Capacity to change the conceptual schema without having to change view schemas or applications
- Changes to constraints can be applied to the conceptual schema without affecting the view schemas or applications

Physical data independence

- Capacity to change the internal schema without having to change the conceptual schema
- Some physical files were reorganized to improve the performance, but the conceptual schema should be kept unchanged

§2.3 Database Languages and Interfaces

§2.3.1 DBMS Languages

The first step is to specify conceptual and internal schemas for the database, once after the design of a database is completed

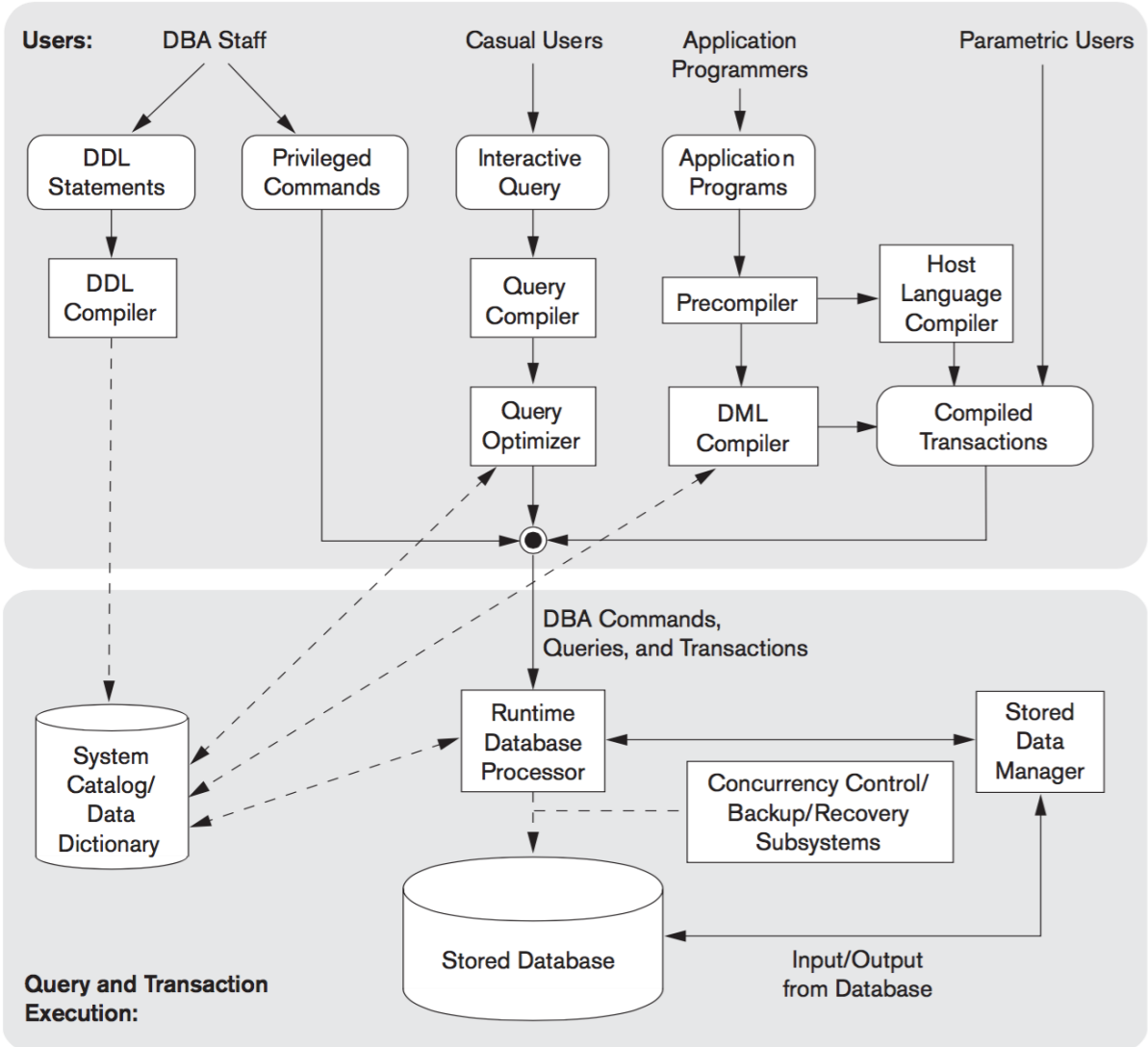
- Data definition language (**DDL**): specify the conceptual schema only
- Storage definition language (SDL)
- View definition language (VDL)
- Data manipulation language (**DML**): provide a set of operations such as retrieval, insertion, deletion, and modification of the data
- Whenever DML commands, are embedded in a general-purpose programming language, that language is called the **host language** and the DML is called the **data sublanguage**

§2.3.2 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
  - Parametric users, such as bank tellers, often have a small set of operations that they must perform repeatedly
  - A special interface for each known class of naive (or parametric) users
- Interfaces for the DBA

§2.4 The Database System Environment

§2.4.1 DBMS Component Modules



§2.4.2 Database System Utilities

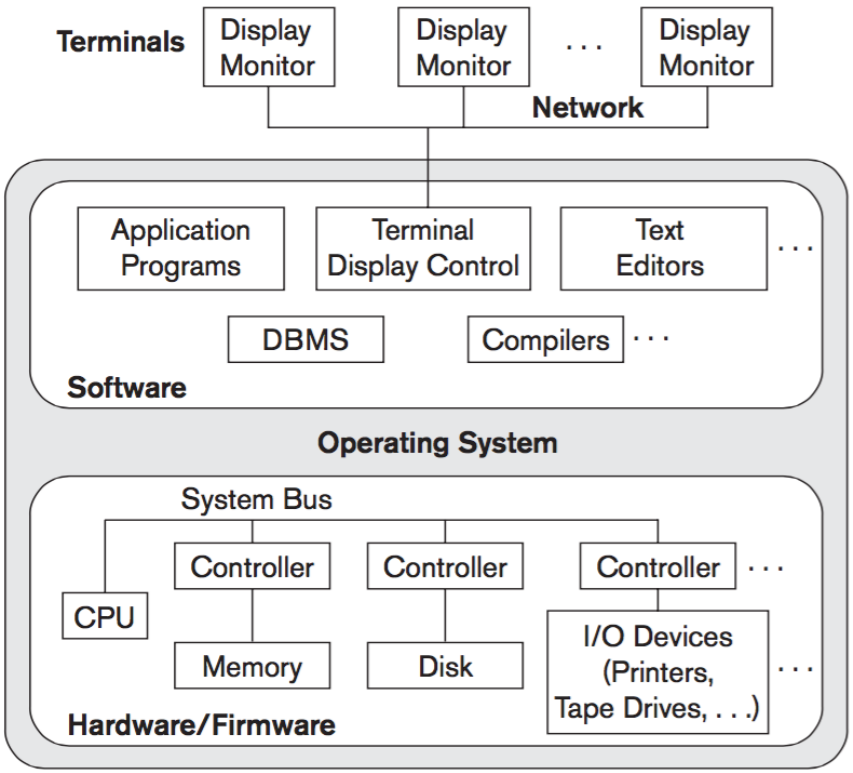
- Loading: load existing data file such as text files or sequential files, into the database
- Backup: create a backup copy of the database
- Database storage reorganization: reorganize a set of database files into different file organizations and create new access paths to improve performance
- Performance monitoring: provide statistics to the DBA

§2.4.3 Tools, Application Environments, and Communications Facilities

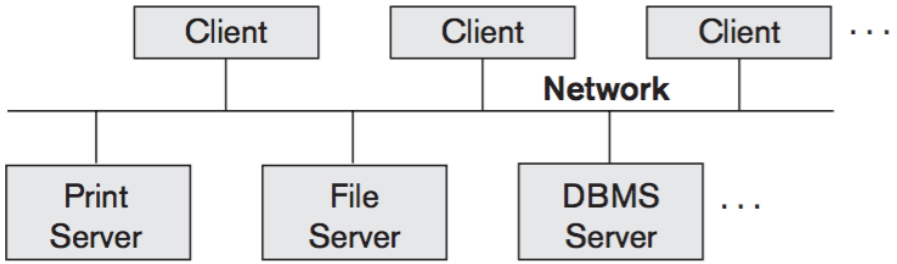
- Other tools are often available to database designers, users, and the DBMS
  - CASE tools
  - Expanded data dictionary system

§2.5 Centralized and Client/Server Architectures for DBMSs

§2.5.1 Centralized DBMSs Architecture

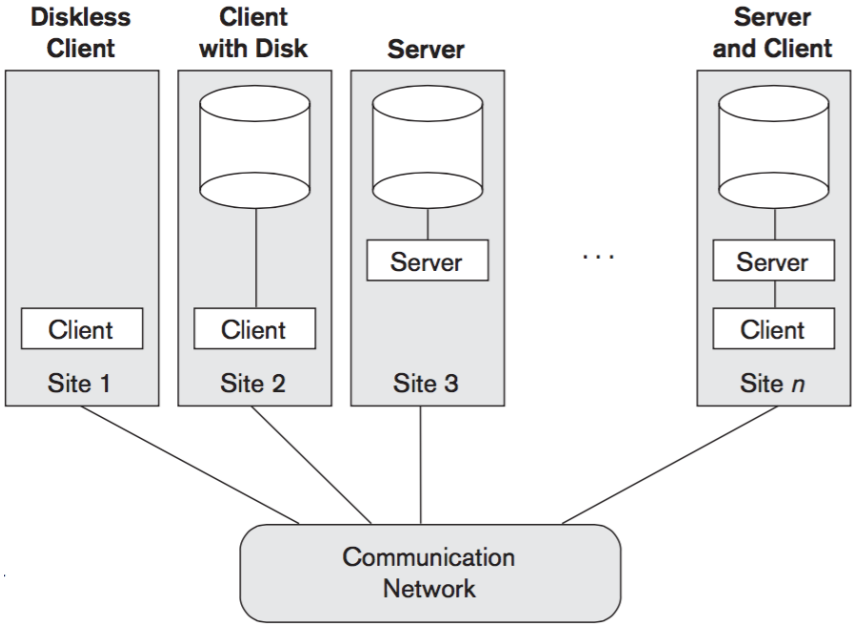


§2.5.2 Basic Client/Server Architectures



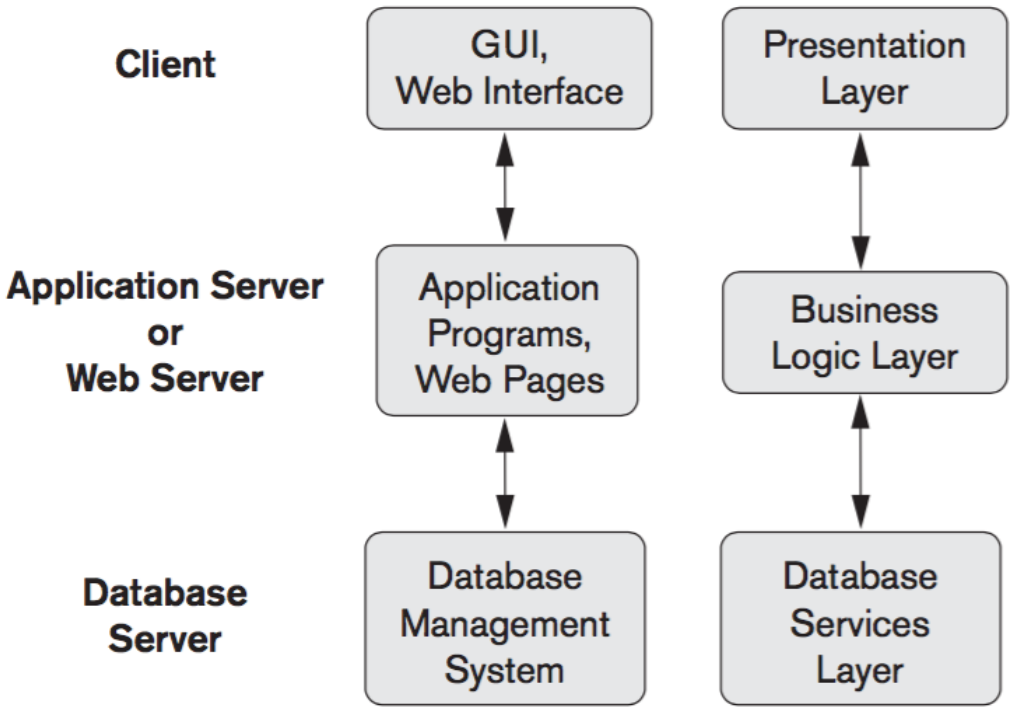
- DBMS systems started to exploit the available processing power at the user side
- The basic idea is to define specialized servers with specific functionalities
- The resources provided by specialized servers can be accessed by many client machines
- The client machines provide the user with the appropriate interfaces to utilize these servers

§2.5.3 Two-Tier Client/Server Architectures for DBMSs



- Software modules of the DBMS were divided between client and server
  - At server level: handle data storage on disk pages, local concurrency control and recovery, buffering and caching of disk pages, and so on
  - At client level: handling user interface, data dictionary functions, global query optimization, concurrency control, and so on

§2.5.4 Three-Tier and *n*-Tier Architectures for Web Applications



§2.6 Classification of Database Management Systems

- Hierarchical data model
  - Represent data as hierarchical tree structures
  - Each hierarchy represents a number of related records
  - Ex. IBM IMS DL/1
- Network data model
  - Represent data as record types and represents a limited type of $1 : N$ relationship
  - A $1 : N$ relationship relates a record to many records
  - Ex. IMAGE (Hewlett-Packard)
- Relational data model
  - Represent a database as a collection of tables
- Object data model
  - Represent a database in terms of objects, their properties, and their operations
  - Objects with the same structure and behavior belong to a class
  - Classes are organized into acyclic graphs

Wrap-up & Questions

정리

- Independence among schemas: The Three-Schema Architecture
- Some terminologies: DDL, DML, etc
- Some other sections: read them carefully!!

********** Thanks & Question? **********