

Database Management Systems Introduction

Data Model

- a collection of concepts that can be used to define the structure (data, data types, relations and constraints) of a database.
- Examples:
 - Entity Relationship model
 - Relational Model
 - hierarchical & network models
 - object-data models

Levels of Data Abstraction

External Schema

Conceptual Schema

Physical Schema

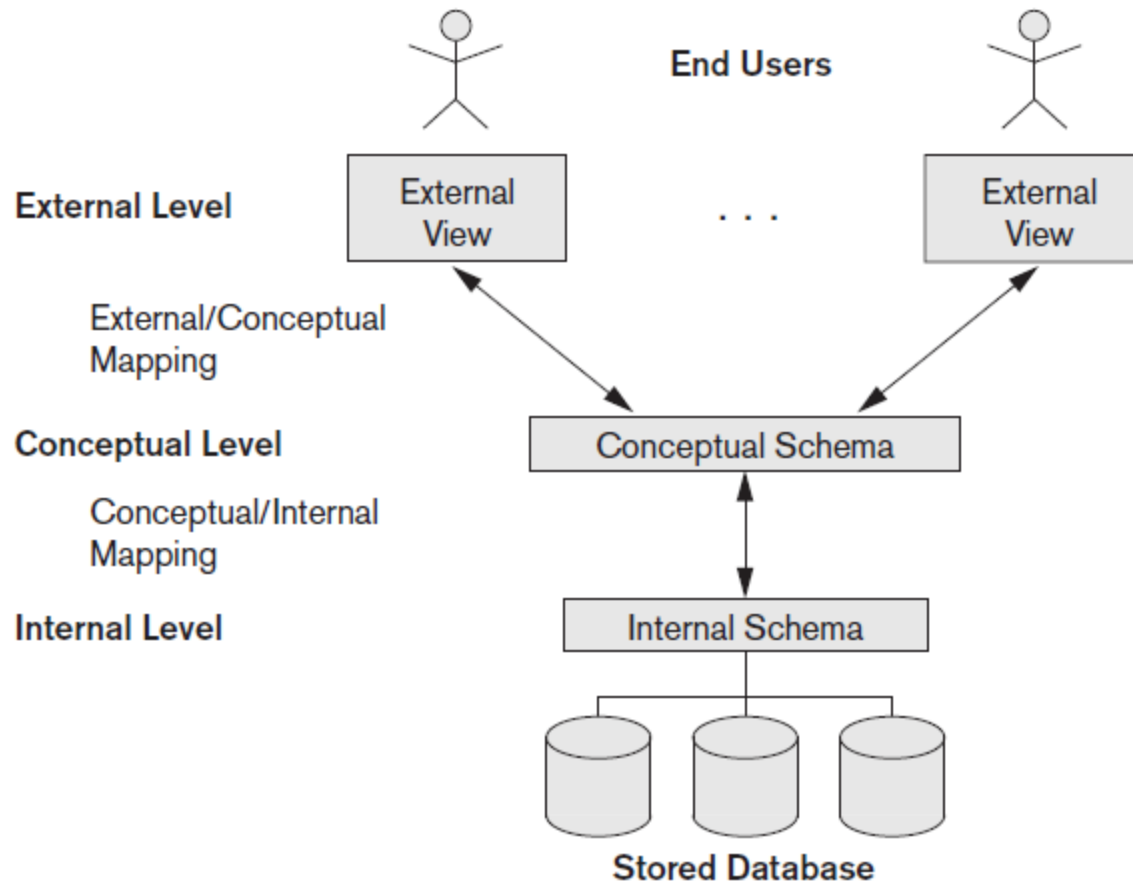
High Level



Low Level

Figure 2.2

The three-schema architecture.



Schemas & Instances

- Important to distinguish between
 - **database schema**: the description of the database
 - **database**: the stored data

Database Schema

Financial Records

(Company, Type, Name, Date, Amt, NumShares, Broker)

Database

Company	Type	Name	Date	Amt	NumShares	Broker
Trimark	Mutual Fund	Trimark Fund	01/01/84	49.75	100	C. Harris
AGF	Mutual Fund	Foreign Equity	01/01/94	62.25	1000	C. Harris

Characteristics of the Database Approach

- insulation of programs & data
- support of multiple user views
- use of a catalog to support database descriptions

Database States

- Empty State -- database is empty when we first define the database schema
- Initial State -- database is first populated or loaded with data
- Current State -- snapshot in time

Data Independence

- **Logical data independence** - ability to change the conceptual schema without having to change the external schemas or application programs
- **Physical data independence** - ability to change the internal schema without having to change the conceptual (or external) schemas.

DBMS Languages

- **DDL: Data Definition Language**
 - used to define/change the structure of the database
 - SDL – for storage...
 - VDL – for external views...
- **DML: Data Manipulation Language**
 - used to query the database, insert data, change data or delete data
 - Low level/procedural – embedded in host language

DBMS Interfaces

- Menu driven
- Mobile Apps
- Form based
- GUI
- Natural Language interfaces
- Keyword-based
- Speech – as I/P and O/P
- Interface for parametric user (canned transactions)

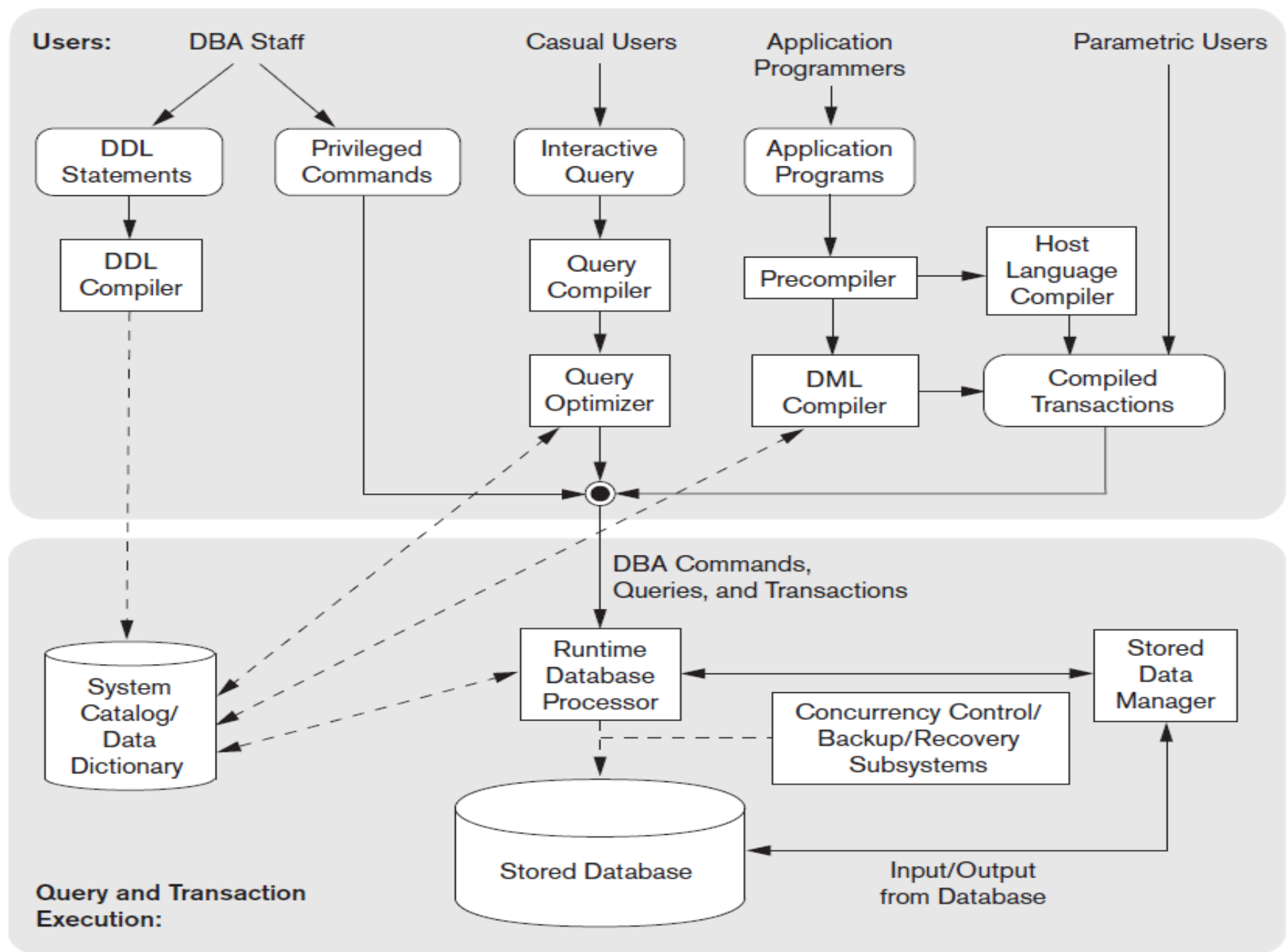


Figure 2.3
Component modules of a DBMS and their interactions.

Database Utilities

- Loading (exports/imports across FS)
- Backup
- Storage reorganization
- Performance monitoring

Tools, Application environments

- CASE tools
- Application development environments
- Communication softwares

Centralized and Client/Server Architectures for DBMSs

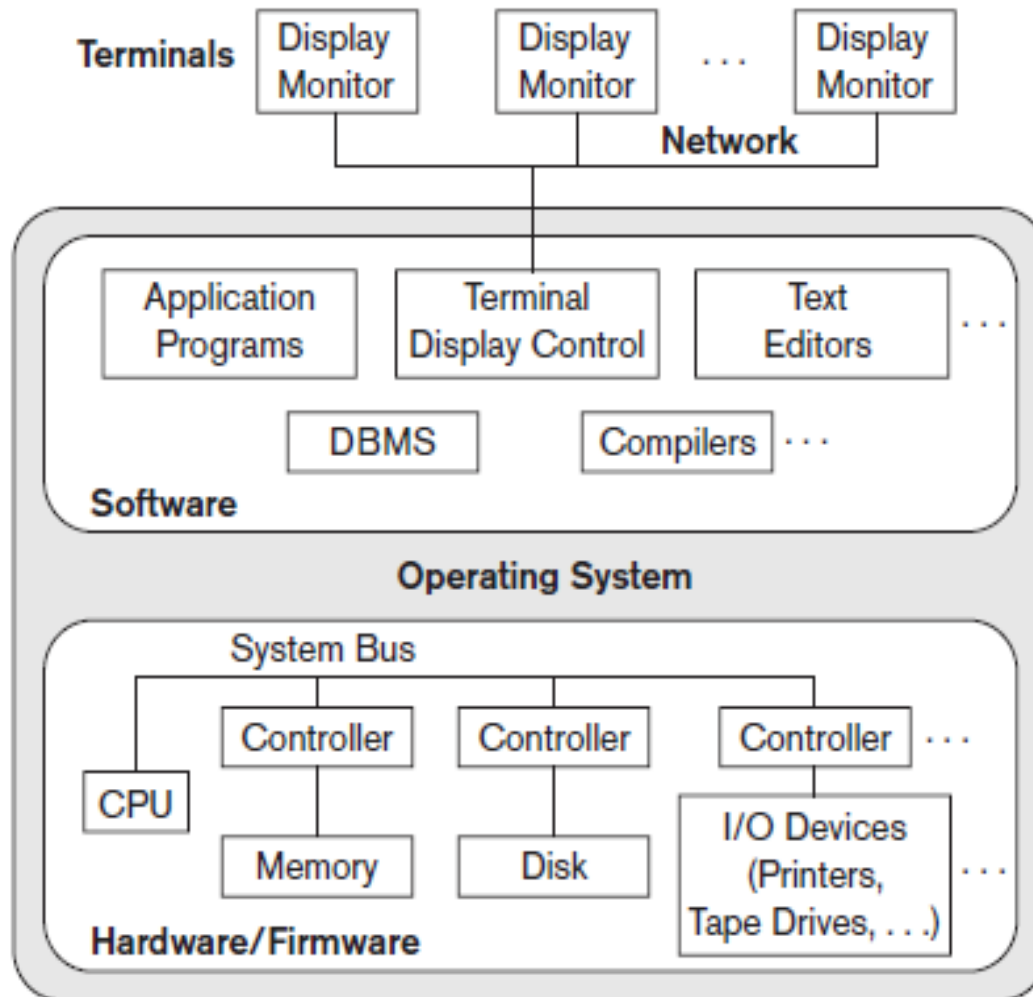
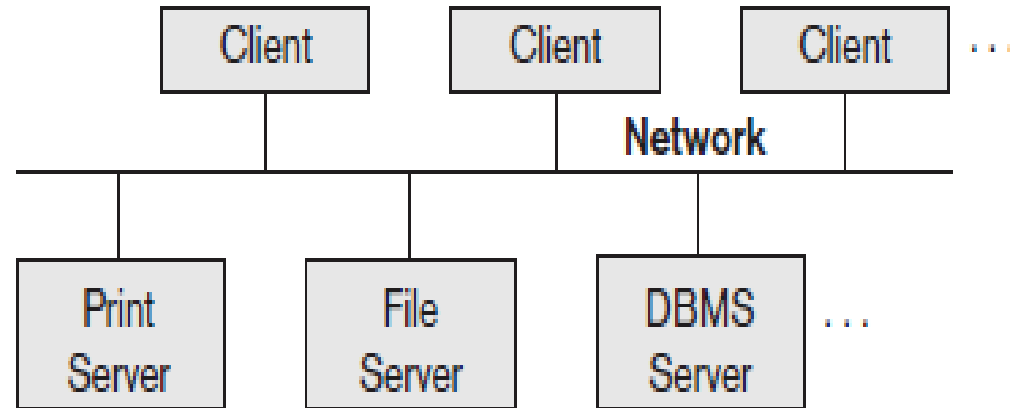


Figure 2.4
A physical centralized architecture.

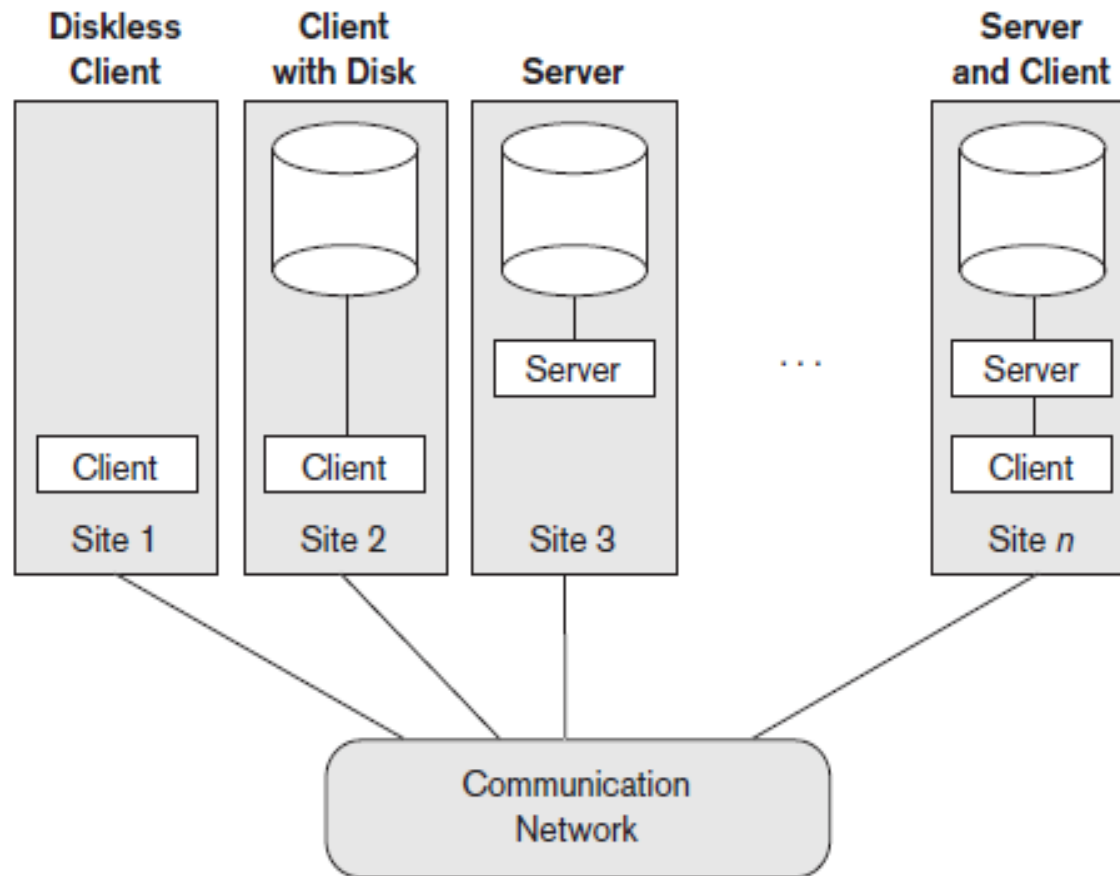
Centralized and Client/Server Architectures for DBMSs

Figure 2.5
Logical two-tier
client/server
architecture.



Centralized and Client/Server Architectures for DBMSs

Figure 2.6
Physical two-tier
client/server architecture.



Centralized and Client/Server Architectures for DBMSs

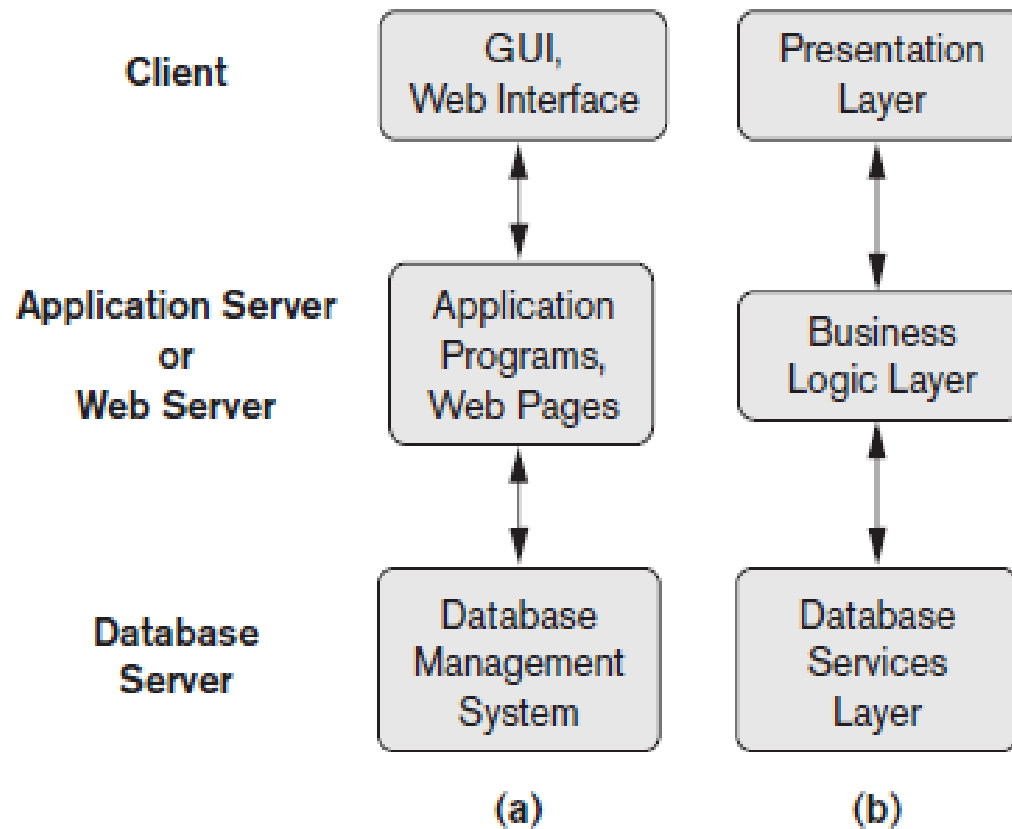


Figure 2.7

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

Classification of DBMS

- Data Model based
 - Relational
 - Object
 - Object-relational
 - Hierarchical
 - Network
 - XML
- No. of users – single vs. multi user
- No. of sites – centralized vs. distributed
- Cost – open source (free), licenses – site/user
- Purpose – general/ special