

204222 - Fundamentals of Database Systems

Chapter 2

Database System Concepts and Architecture

Adapted for 204222

by Areerat Trongratsameethong

Chapter 2 Outline

- Data Models, Schemas, and Instances
- Three-Schema Architecture and Data Independence
- Database Languages and Interfaces
- The Database System Environment
- Centralized and Client/Server Architectures for DBMSs
- Classification of Database Management Systems

Data Models, Schemas, and Instances

- **Data abstraction**

- Suppression of details of data organization and storage: ไม่ได้ให้รายละเอียดขององค์ประกอบข้อมูล และรายละเอียดของที่ใช้จัดเก็บข้อมูล
- Highlighting of the essential (สำคัญ หรือ จำเป็น) features for an improved understanding of data: เน้นเฉพาะ features ที่สำคัญ หรือจำเป็น ที่ทำให้เพิ่มความเข้าใจข้อมูลมากยิ่งขึ้น

Data Models, Schemas, and Instances (cont'd.)

- **Data model**

- **Data Model:** แบบจำลองข้อมูลเพื่อให้เข้าใจแนวคิด และใช้อธิบายโครงสร้างของฐานข้อมูลในรูปแบบแผนภาพ
- ทำให้เข้าใจความหมายของข้อมูลในระดับแนวคิด
- **Basic operations:** มี operation พื้นฐาน สำหรับดึงข้อมูล และปรับปรุงข้อมูล
 - Specify retrievals and updates on the database
- **Dynamic aspect or behavior** of a database application
 - Database designer สามารถกำหนดได้ว่า operations ใดที่อนุญาตให้ดำเนินการกับข้อมูลในฐานข้อมูล

Categories of Data Models

- **High-level or conceptual data models**
 - Close to the way many users perceive data: อยู่ระดับที่ใกล้เคียงกับมุมมองที่ผู้ใช้เข้าใจ
- **Low-level or physical data models**
 - Describe the details of how data is stored on computer storage media: อธิบายรายละเอียดของข้อมูลว่าจัดเก็บอย่างไรใน physical storage
- **Representational data models: อยู่ตรงกลางระหว่าง Low-Level กับ High-Level**
 - นำเสนอในลักษณะของ concepts ที่ผู้ใช้เข้าใจง่าย และใกล้เคียงกับมุมมองที่ข้อมูลถูกจัดเก็บใน computer storage

Categories of Data Models (cont'd.)

- **Entity**

- นำเสนอสิ่งต่างๆที่เกิดขึ้น หรือนำเสนอแนวคิด (concept)

- **Attribute**

- นำเสนอคุณสมบัติของสิ่งที่สนใจ และช่วยอธิบายความหมายของแต่ละ entity

- **Relationship among two or more entities:**

ความสัมพันธ์ระหว่างสอง **entity** หรือมากกว่า

- นำเสนอความสัมพันธ์ และความเกี่ยวข้องกัน ของข้อมูลระหว่าง **entity**

- นำเสนอในรูปแบบของ **Entity-Relationship model**

Categories of Data Models (cont'd.)

- **Relational data model**

- ถูกใช้ส่วนใหญ่ใน DBMS แบบดั้งเดิมที่ใช้ในวงการธุรกิจ

- **Object data model**

- เป็น data model ใหม่ที่มีลักษณะใกล้เคียงกับ conceptual data model ซึ่งเหมาะกับข้อมูลที่นำเสนอในรูปแบบของ Object Oriented Concept

Categories of Data Models (cont'd.)

- **Physical data models**

- อธิบายลักษณะการจัดเก็บข้อมูลระดับ **physical** ในเครื่องคอมพิวเตอร์

- **Access path**

- โครงสร้างที่ทำให้การค้นหาข้อมูลระดับ **records** ในฐานข้อมูลมีประสิทธิภาพ

- **Index**

- การสร้าง **index** เป็นตัวอย่างหนึ่งของ **access path** ที่ทำให้การสืบค้นข้อมูล หรือ การเข้าถึงข้อมูลทำได้โดยตรง ซึ่งทำให้การเข้าถึงทำได้รวดเร็วขึ้น

Schemas, Instances, and Database State

- **Database schema**

- คำนิยาม หรือคำอธิบายของฐานข้อมูล ซึ่งประกอบด้วยโครงสร้างฐานข้อมูล และชนิดข้อมูล

- **Schema diagram**

- ใช้แสดง หรือนำเสนอมุมมองของ schema

- **Schema construct**

- สิ่งที่อยู่ใน schema เช่น Student และ Course เป็นต้น

- **Database state or snapshot**

- สถานะ (หรือค่า) ของข้อมูลที่จัดเก็บในฐานข้อมูล ณ เวลาใดเวลาหนึ่ง

Schemas, Instances, and Database State (cont'd.)

Figure 2.1

Schema diagram for the database in Figure 1.2.

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

⁶Schema changes are usually needed as the requirements of the database applications change. Newer database systems include operations for allowing schema changes, although the schema change process is more involved than simple database updates.

⁷It is customary in database parlance to use *schemas* as the plural for *schema*, even though *schemata* is the proper plural form. The word *scheme* is also sometimes used to refer to a schema.

Schemas, Instances, and Database State (cont'd.)

- **Define a new database**

- ในการสร้างฐานข้อมูลใหม่ ต้องกำหนด **database schema** โดยใช้ **DBMS**

- **Initial state**

- ค่าข้อมูลเริ่มต้น คือค่าข้อมูลเริ่มต้นหลังจากที่สร้างฐานข้อมูล (เมื่อเวลาผ่านไป ค่าหรือสถานะของข้อมูล บางตัวอาจมีการเปลี่ยนแปลง)

- **Valid state**

- ข้อมูลจะเรียกว่าอยู่ในสถานะที่ถูกต้อง ก็ต่อเมื่อ มีโครงสร้างถูกต้อง และมีค่าข้อมูลตรงกับกฎข้อบังคับที่กำหนดไว้ใน **database schema**

- **Schema evolution**

- **Schema** อาจมีการเปลี่ยนแปลงได้ตามความต้องการที่เปลี่ยนแปลงไปของระบบ

Three-Schema Architecture and Data Independence

- **Internal level**

- อธิบายถึงวิธีการจัดเก็บข้อมูลใน **physical storage** และ **access path** ของ **database**

- **Conceptual level**

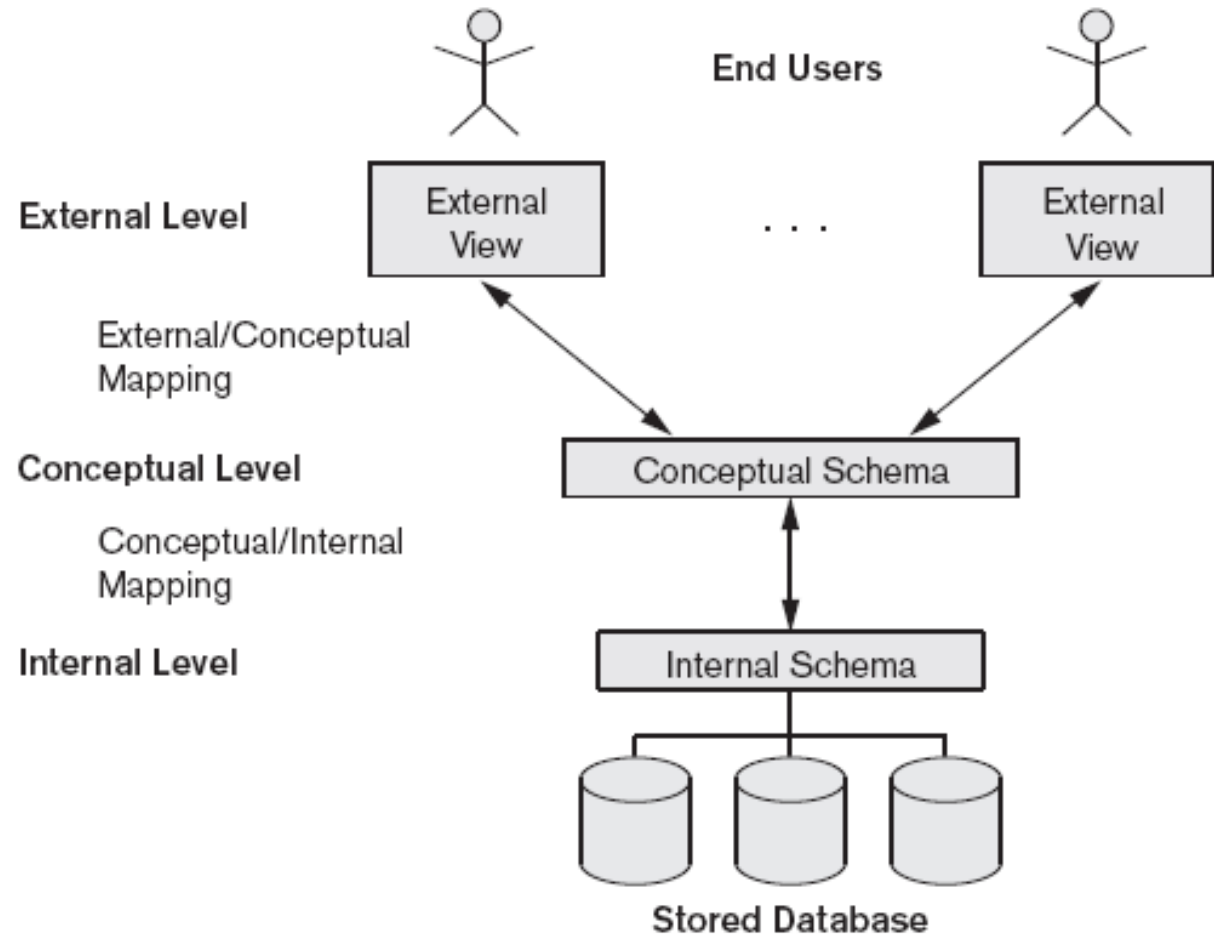
- อธิบายโครงสร้างของฐานข้อมูลในระดับแนวคิด (**concept**) สำหรับผู้ใช้ ฐานข้อมูลร่วมกัน ซึ่งอยู่ในรูปของ **Representational data models**

- **External or view level**

- อธิบายฐานข้อมูลในมุมมองของผู้ใช้เฉพาะแต่ละกลุ่มที่สนใจข้อมูลต่างมุมมอง จะแสดงเฉพาะสิ่งที่ผู้ใช้สนใจเท่านั้น ส่วนที่ไม่สนใจจะถูกซ่อนไว้ เช่น ผู้ใช้แผนกขาย จะเห็นข้อมูลต่างมุมมองกับผู้ใช้แผนกบัญชี เป็นต้น

Three-Schema Architecture and Data Independence (cont'd.)

Figure 2.2
The three-schema architecture.



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
 - การแบ่งข้อมูลออกเป็น 3 Level ทำให้การเปลี่ยนแปลงของ schema ใน Level หนึ่ง ไม่มีผลกระทบกับ Level ที่อยู่สูงกว่า (ไม่ต้องแก้ Schema ที่อยู่ใน Level ที่อยู่สูงกว่า)
- **Types:** ความเป็นอิสระของข้อมูล แบ่งออกเป็น 2 ประเภท คือ
 - **Logical:** แก้ไขที่ conceptual schema ไม่มีผลกระทบกับ external schemas หรือ application programs เช่น แก้ไข constraint ไม่มีผลกระทบกับ end user
 - **Physical:** แก้ไขที่ internal schema ไม่มีผลกระทบกับ conceptual schema และ external schema เช่น การสร้าง index เพื่อให้การสืบค้นข้อมูลเร็วขึ้น ไม่มีผลกระทบกับ level อื่นๆ

DBMS Languages

- **Data definition language (DDL)**
 - Defines both schemas (Conceptual + Internal)
- **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

DBMS Languages (cont'd.)

- **High-level or nonprocedural DML:** e.g. SQL
 - Can be used on its own to specify complex database operations concisely
 - **Set-at-a-time or set-oriented:** ประมวลผลได้ครั้งละหลายๆ record
- **Low-level or procedural DML**
 - Must be embedded in a general-purpose programming language (e.g. PHP)
 - **Record-at-a-time:** ประมวลผลได้ครั้งละ 1 record โดยใช้คำสั่ง **Loop** ช่วย

DBMS Interfaces

- Menu-based interfaces for Web clients or browsing
- Forms-based interfaces เช่น Microsoft Access
- Graphical user interfaces
- Natural language interfaces
- Speech input and output
- Interfaces for parametric users: เช่น มี Interface เฉพาะสำหรับพนักงาน Bank
- Interfaces for the DBA

The Database System Environment

- DBMS component modules
 - **Buffer management:** มี Buffer ไว้เก็บข้อมูลที่กำลัง execute เพื่อลด Overhead จากการอ่านหรือเขียนข้อมูลลง Disk
 - **Stored data manager:** ควบคุมการเข้าถึงข้อมูลสารสนเทศของ DBMS ที่จัดเก็บอยู่ใน disk หรือ ข้อมูลที่เก็บอยู่ใน system catalog
 - **DDL compiler**
 - **Interactive query interface**
 - Query compiler
 - Query optimizer: ช่วยทำให้การจัดการข้อมูลผ่าน Query มีประสิทธิภาพขึ้น
 - **Precompiler:** แยกคำสั่ง DML ออกจาก Application Program

The Database System Environment (cont'd.)

- DBMS component modules
 - **Runtime database processor:** execute privileged command, query execution plan
 - **System catalog**
 - **Concurrency control system**
 - **Backup and recovery system**

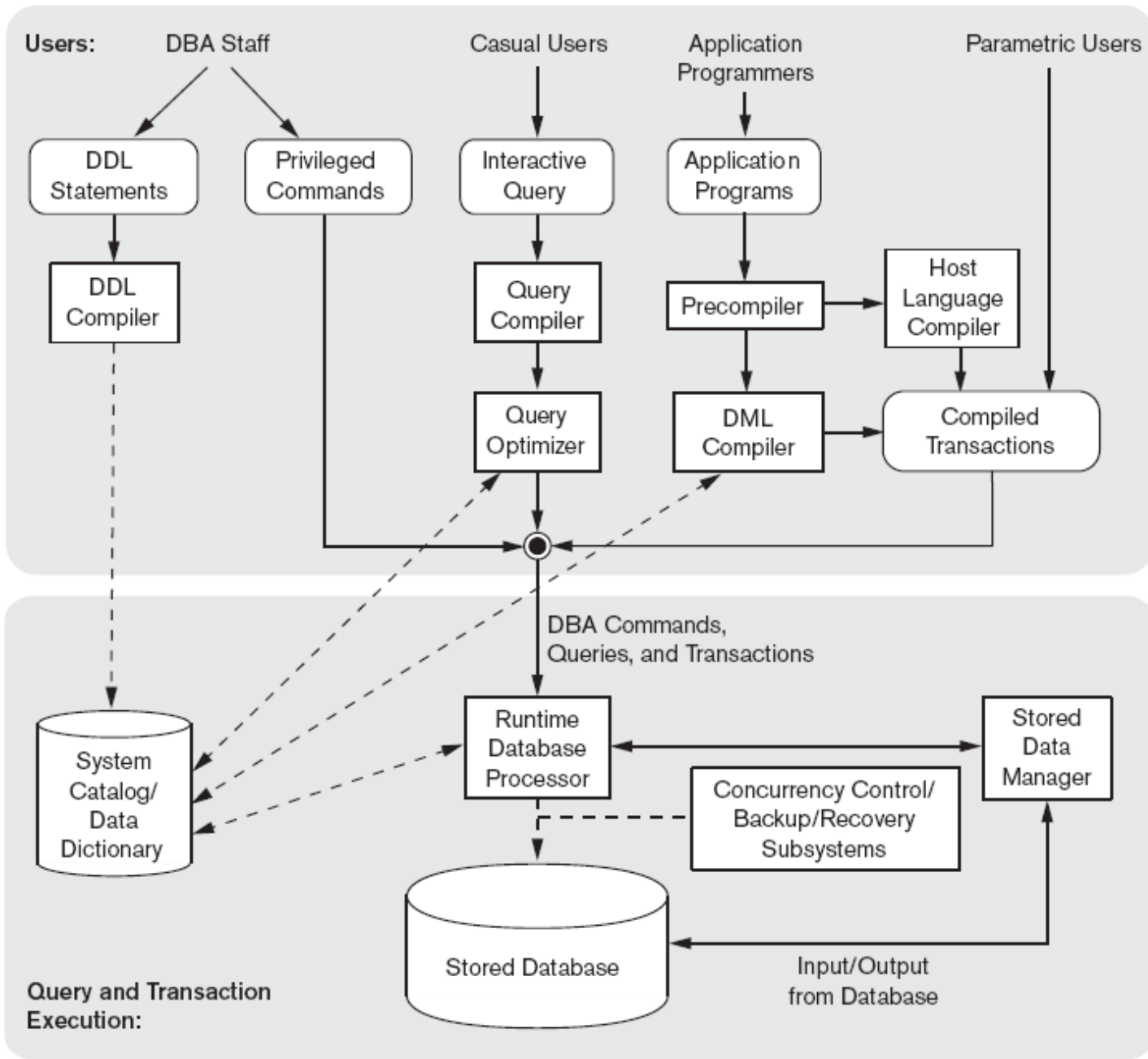


Figure 2.3
Component modules of a DBMS and their interactions.

Database System 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, e.g. create new access path to improve performance
- **Performance monitoring**
 - Monitors database usage and provides statistics to the DBA: สำหรับ monitor ดูการใช้งาน database และให้ข้อมูลสถิติต่างๆ กับ DBA

Tools, Application Environments, and Communications Facilities

- CASE Tools: ใช้ตอน Design Phase
- **Data dictionary (data repository) system:** คล้ายกับ DBMS Catalog แต่มีข้อมูลที่จัดเก็บเพิ่มเติม ดังต่อไปนี้
 - Stores design decisions, usage standards, application program descriptions, and user information
- **Application development environments:** เครื่องมือที่ช่วยในการพัฒนาโปรแกรม
- **Communications software:** ทำหน้าที่ติดต่อฐานข้อมูลผ่านระบบเครือข่าย

Centralized and Client/Server Architectures for DBMSs

- **Centralized DBMSs Architecture**

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

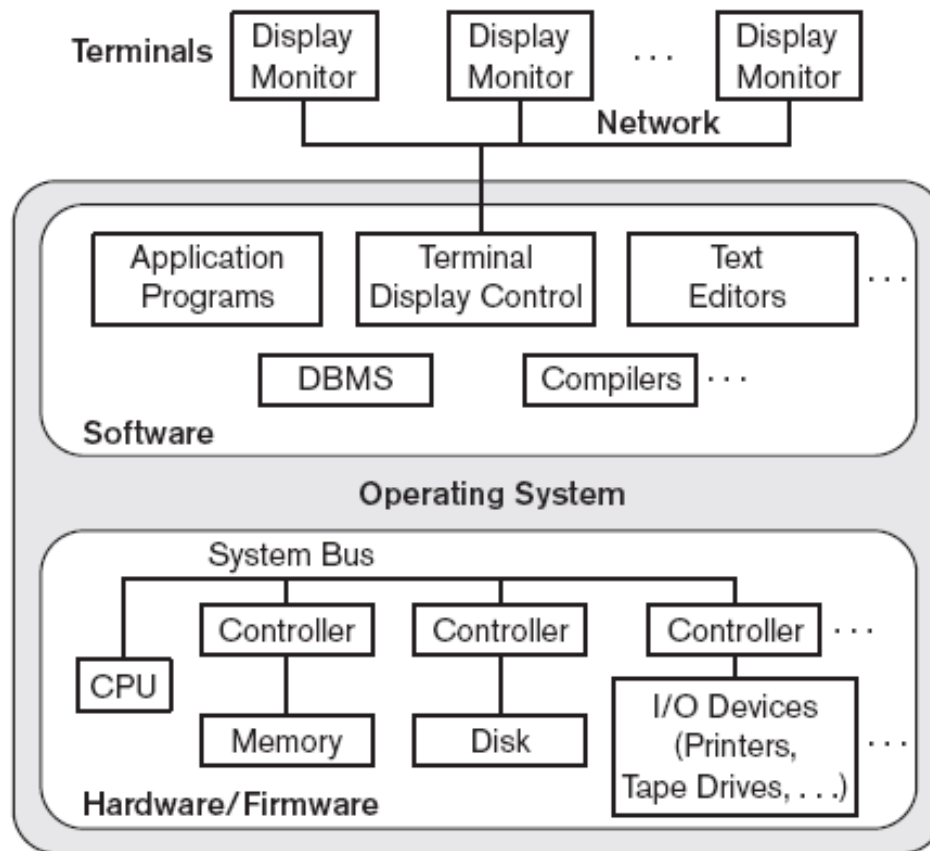


Figure 2.4
A physical centralized architecture.

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 utilize these servers
 - Local processing power to run local applications

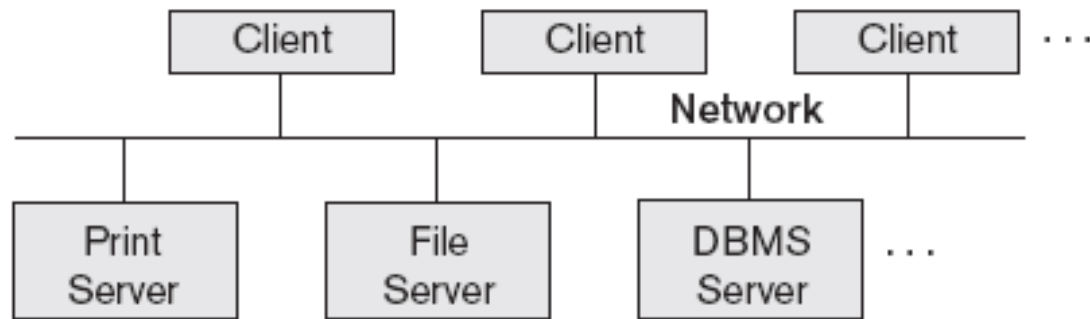


Figure 2.5
Logical two-tier
client/server
architecture.

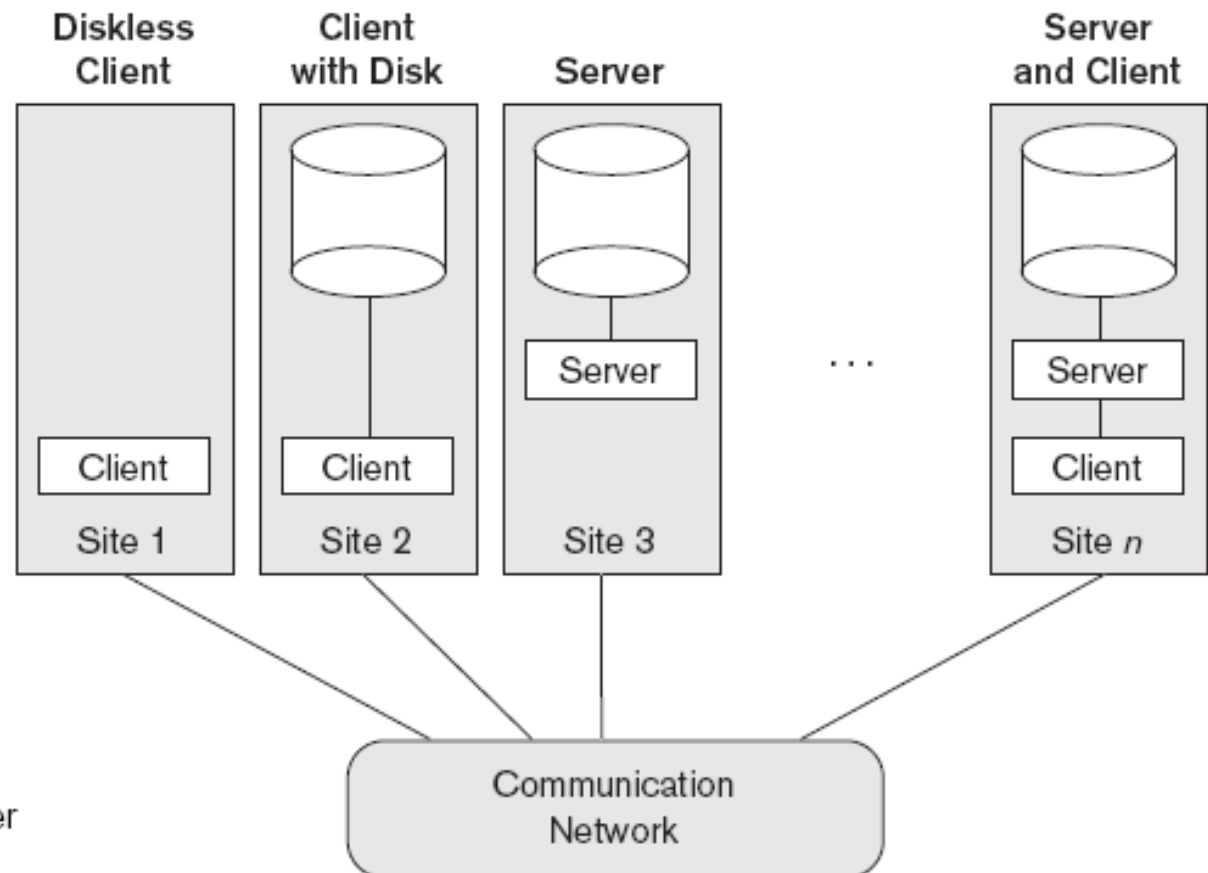


Figure 2.6
Physical two-tier client/server
architecture.

Basic Client/Server Architectures (cont'd.)

- **Client**

- User machine that provides user interface capabilities and local processing

- **Server**

- System containing both hardware and software
- Provides services to the client machines
 - Such as file access, printing, archiving, or database access

Two-Tier Client/Server Architectures for DBMSs

- Server handles
 - Query and transaction functionality related to SQL processing
- Client handles
 - User interface programs and application programs
- Open Database Connectivity (ODBC)
 - Provides application programming interface (API)
 - 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

Three-Tier and n-Tier Architectures for Web Applications

- **Application server or Web server**
 - Adds intermediate layer between client and the database server
 - Runs application programs and stores business rules
- **N-tier**
 - Divide the layers between the user and the stored data further into finer components

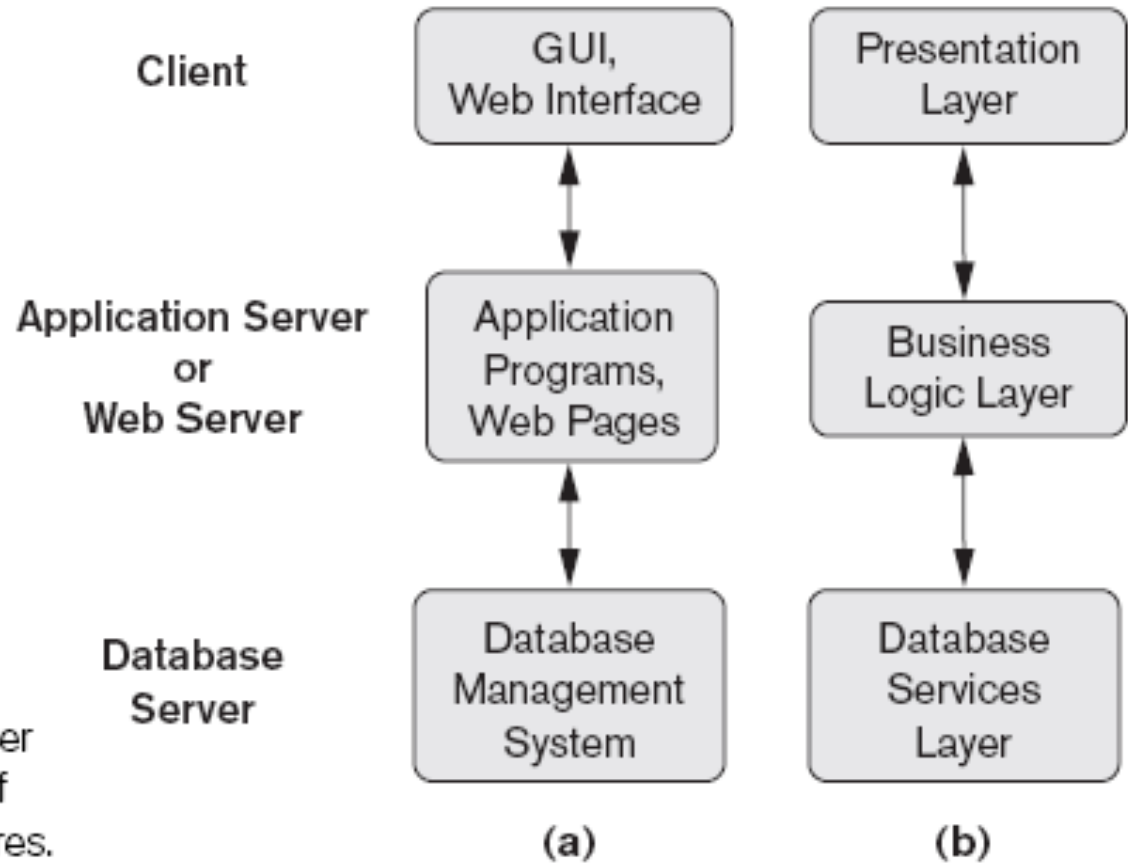
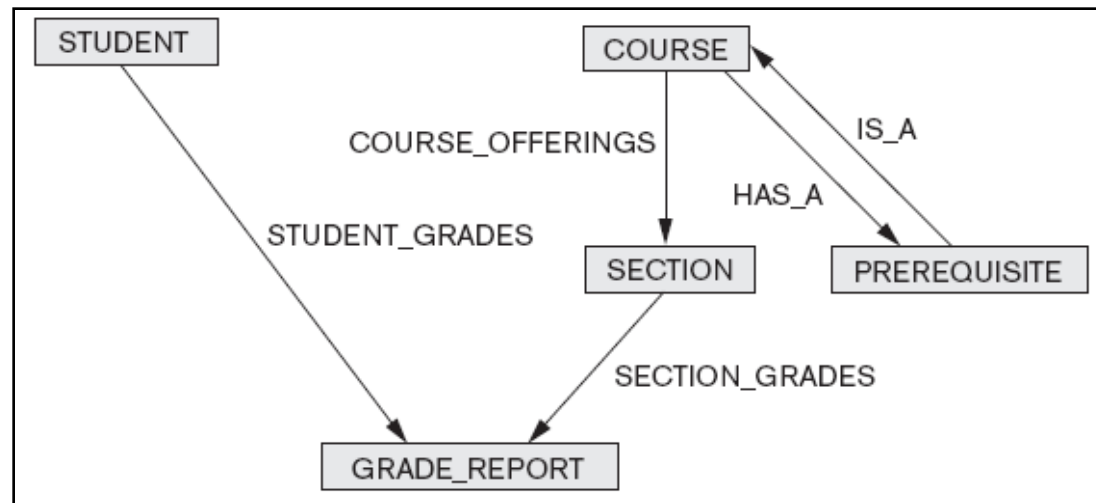


Figure 2.7
 Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

Classification of Database Management Systems

- **Data model**
 - Relational
 - Object
 - Hierarchical and network (legacy)
 - Native XML DBMS
- **Number of users**
 - Single-user
 - Multiuser



Example of Network Model

Classification of Database Management Systems (cont'd.)

- **Number of sites**
 - **Centralized**
 - **Distributed**
 - **Homogeneous:** use the same DBMS software at all the sites
 - **Heterogeneous:** can use different DBMS software at each site
- **Cost**
 - Open source
 - Different types of licensing

Summary

- Concepts used in database systems
- Main categories of data models
- Types of languages supported by DMBSs
- Interfaces provided by the DBMS
- DBMS classification criteria:
 - Data model, number of users, number of sties, access paths, cost