An Introduction to the Database Management Systems

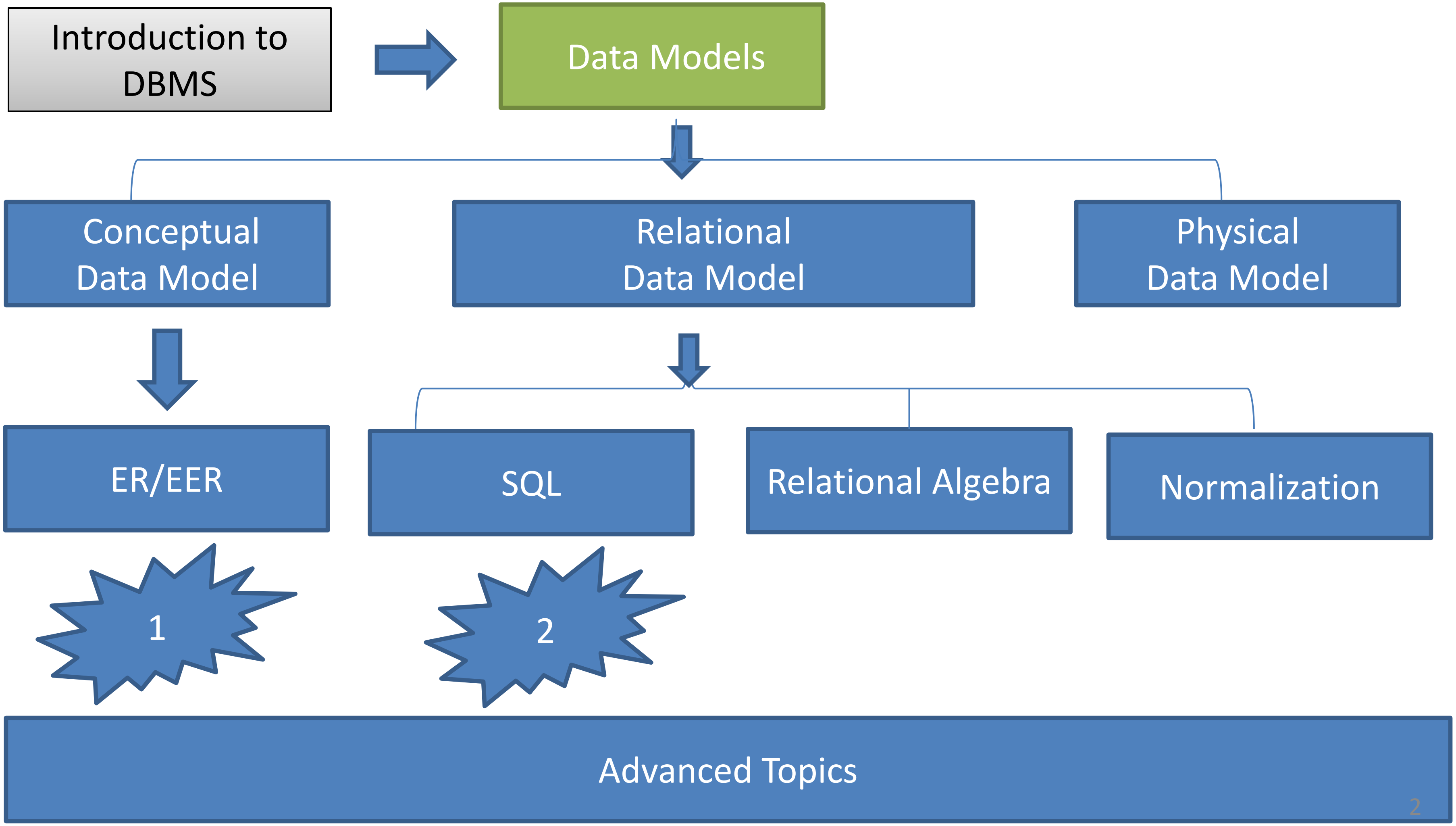
By

Hossein Rahmani

Slides originally by Book(s) Resources

Road Map

(Might change!)



Introduction to DBMS

Data Models

Conceptual Data Model

Relational Data Model

Physical Data Model

ER/EER

SQL

Relational Algebra

Normalization

1

2

Advanced Topics

Database System Concepts And Architecture

- 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

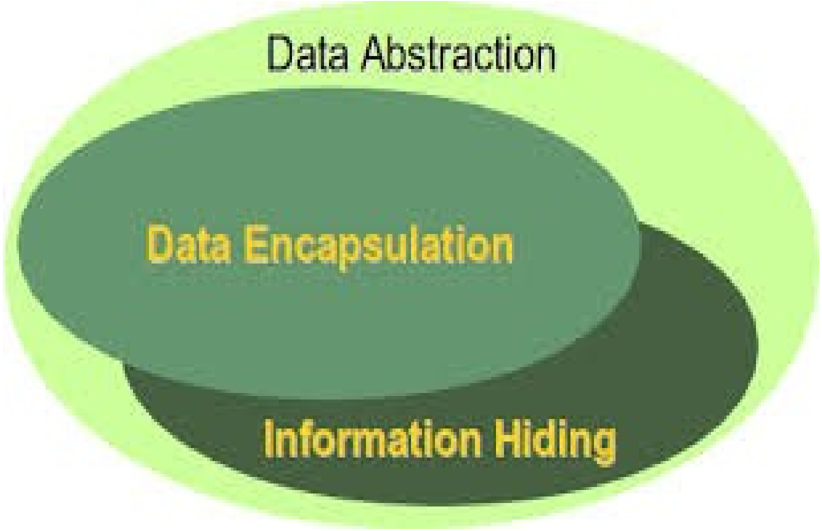
Data Models, Schemas, and Instances

- **Data abstraction**
  - Suppression of details of data organization and storage
  - Highlighting of the essential features for an improved understanding of data

OR

  - Reduction of a particular body of data to a simplified representation of the whole

Data Abstraction



The main thing is How to drive a car…….

How the car is moving and how the engine is working this information is hidden.

(Encapsulation)

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

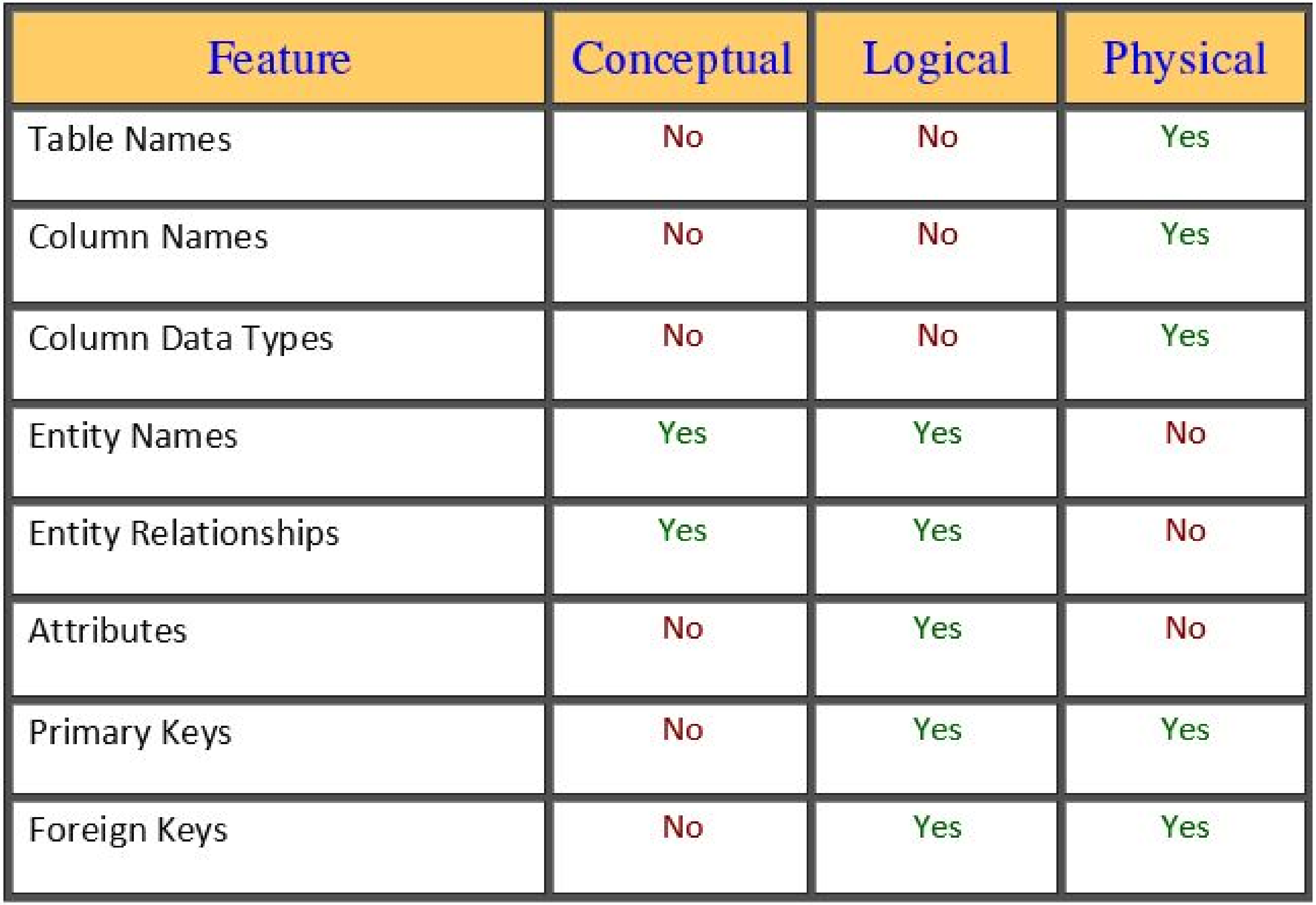
- **Data model**
  - Collection of concepts that describe the structure of a database
  - Provides means to achieve data abstraction
  - **Can Contain:**
    - **Basic operations** (Update, Delete)
    - **Dynamic aspect** or **behavior** of a database

      (user defined operations)

Data models

- **Conceptual** (high-level)
  - for end users / analists

    versus
- **Physical** (low-level)
  - for programmers
- In between: **Representational** (implementation) data model
  - Easily understood by end users
  - Also similar to how data organized in computer storage
  - Widely used Relational model



Feature	Conceptual	Logical	Physical
Table Names	No	No	Yes
Column Names	No	No	Yes
Column Data Types	No	No	Yes
Entity Names	Yes	Yes	No
Entity Relationships	Yes	Yes	No
Attributes	No	Yes	No
Primary Keys	No	Yes	Yes
Foreign Keys	No	Yes	Yes

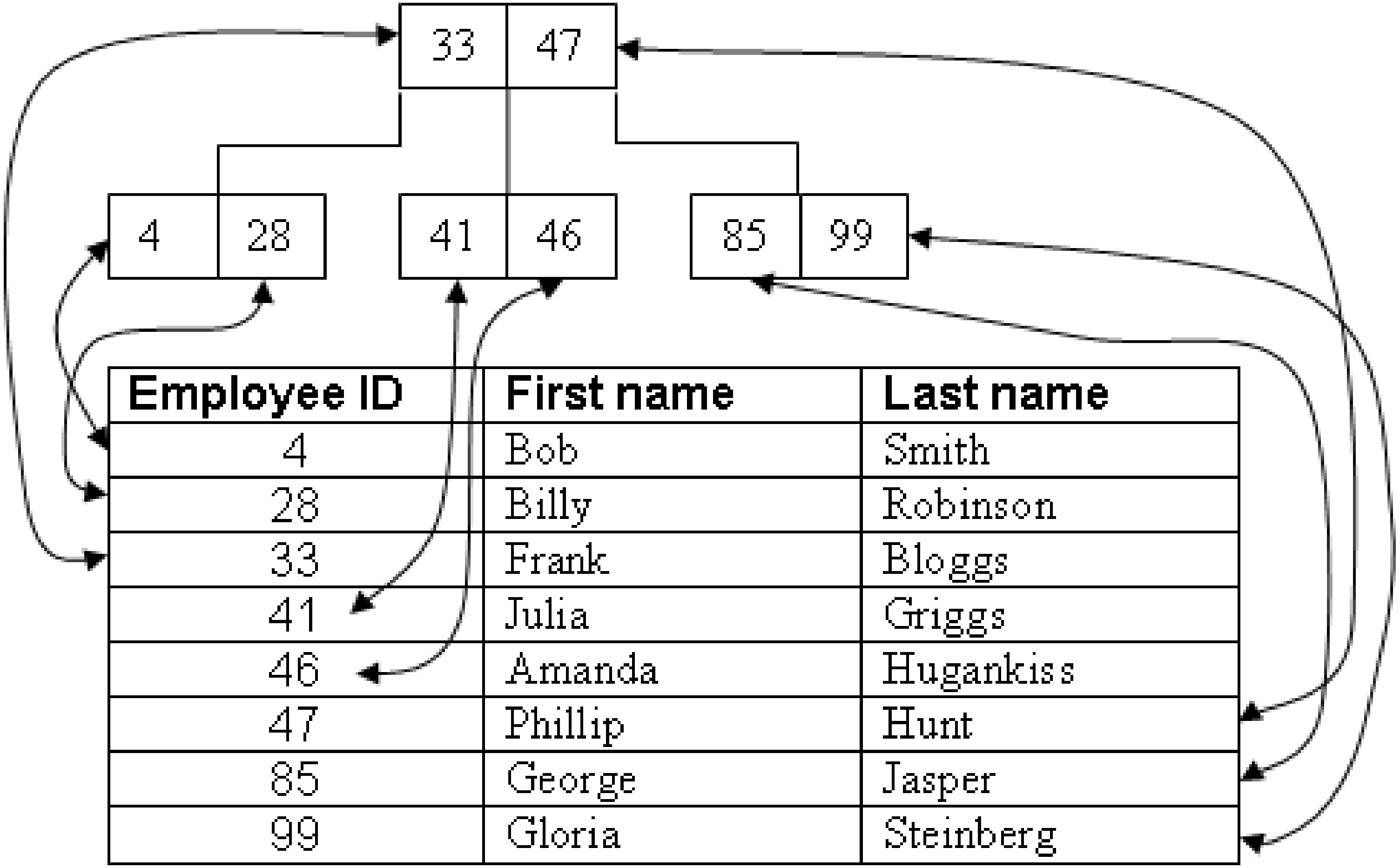
Representational data model

- Relational data model ✓
- Network model
- Hierarchical model
- Object model
- …

Physical data model

- Describe how data is stored as files in the computer
- **Access path**
  - Structure that makes the search for particular database records efficient
- **Index**
  - Example of an access path
  - Allows direct access to data using an index term or a keyword

B-Tree Index

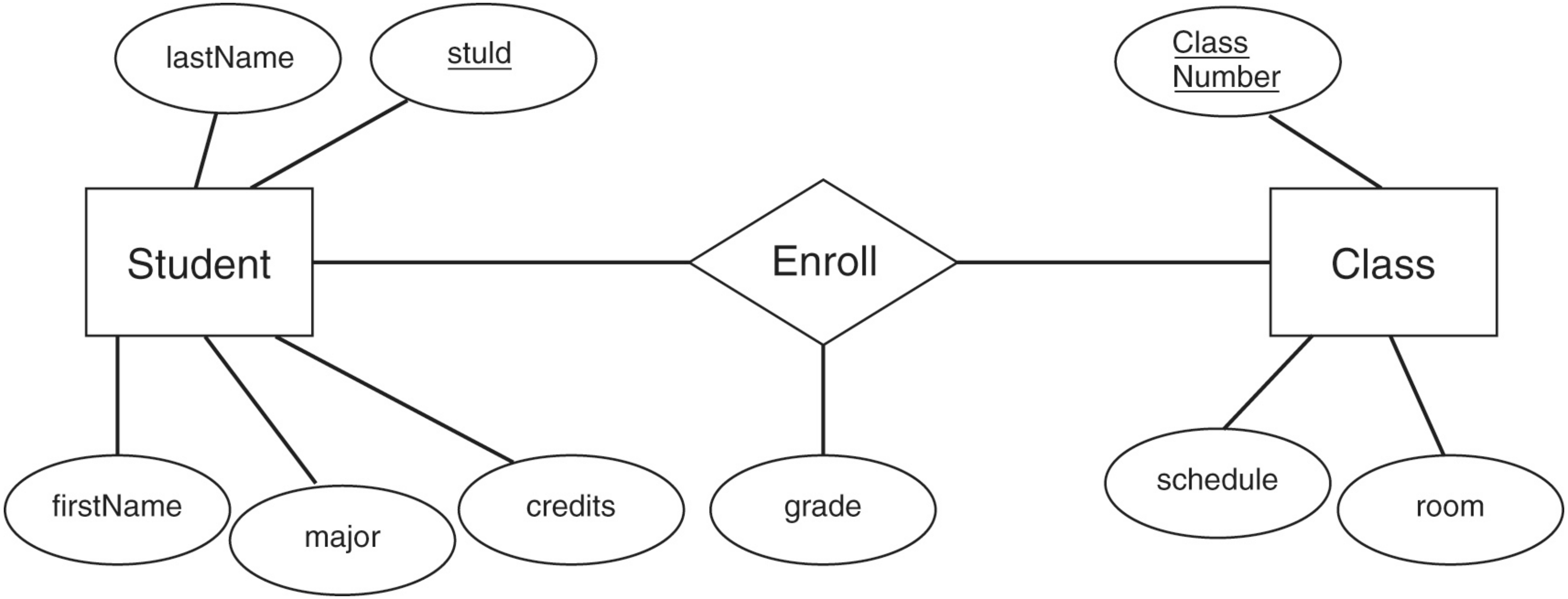


Employee ID	First name	Last name
4	Bob	Smith
28	Billy	Robinson
33	Frank	Bloggs
41	Julia	Griggs
46	Amanda	Hugankiss
47	Phillip	Hunt
85	George	Jasper
99	Gloria	Steinberg

Conceptual data model
Built around…

- **Entity**
  - Represents a real-world object or concept
- **Attribute**
  - Represents some property of interest
  - Further describes an entity
- **Relationship** among two or more entities
  - Represents an association among the entities
  - **Entity-Relationship model**

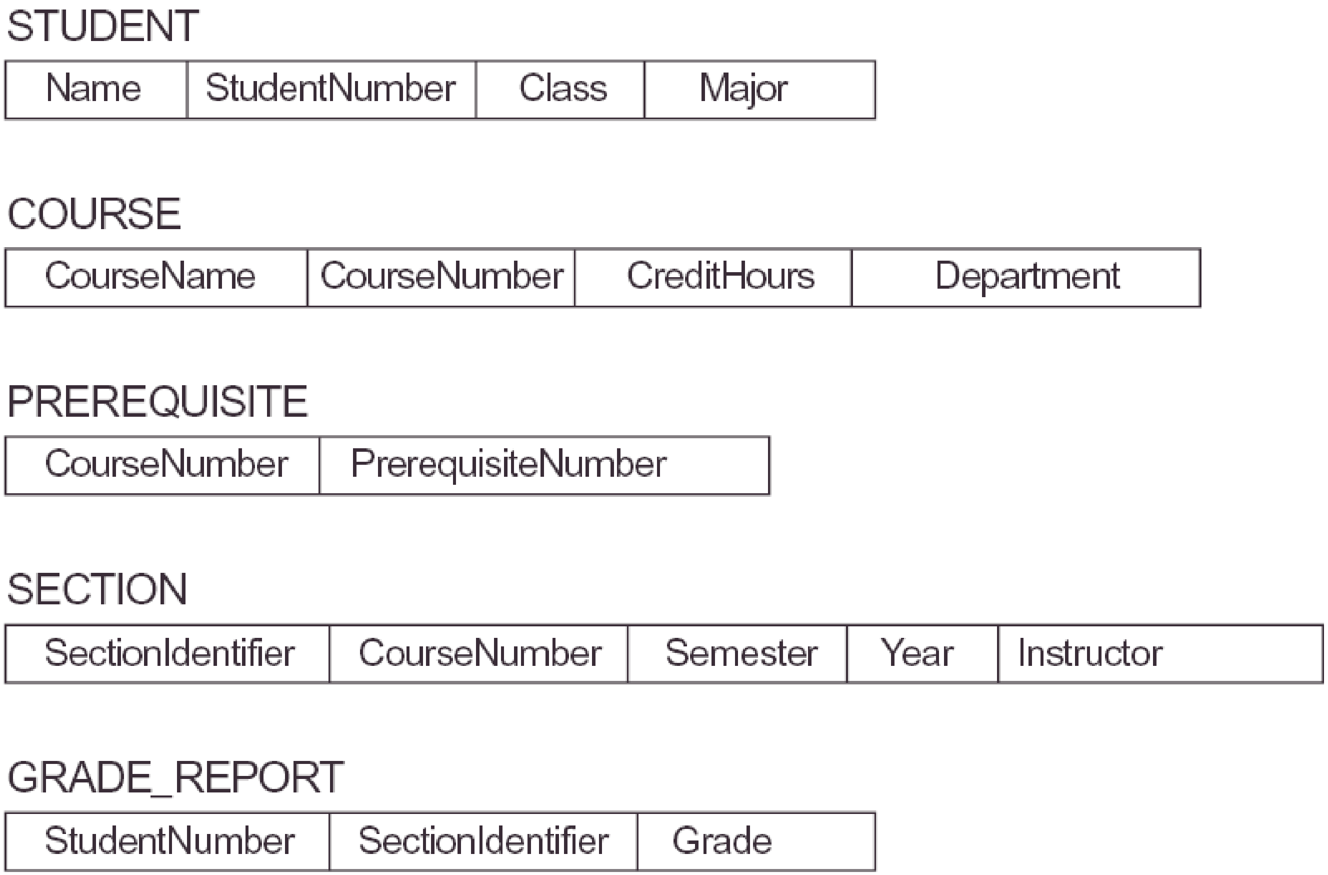
Sample ER Model



Schemas, Instances, and Database State

- **Database schema**
  - Description of a database
- **Schema diagram**
  - Displays selected aspects of schema
- **Schema construct**
  - Each object in the schema
- **Database state** or **snapshot**
  - Data in database at a particular moment in time
- These are stored in a **catalogue**

Database schema diagram



STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

PREREQUISITE

CourseNumber	PrerequisiteNumber

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

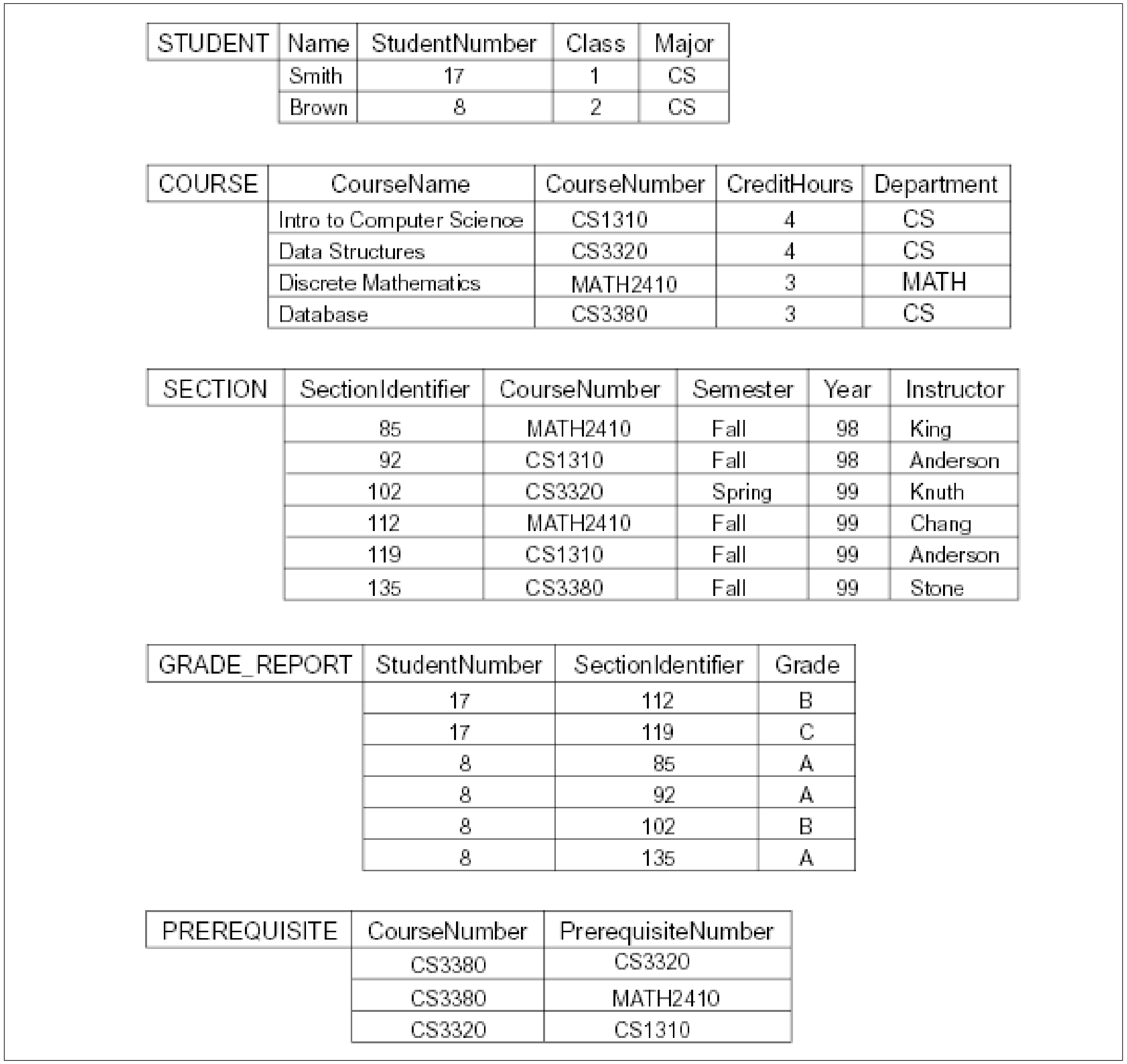
GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

Schemas, Instances, and Database State

- A database schema is (relatively) fixed, but the **content** of a database varies:
  - Database state or *snapshot*
  - Collection of current instances

Database state



STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

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

- **Define** a new database
  - Specify database schema to the DBMS
- **Initial state**
  - **Populated** or **loaded** with the initial data
- **Valid state**
  - Satisfies the structure and constraints specified in the schema

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

- **Schema evolution**
  - Changes applied to schema as application requirements change

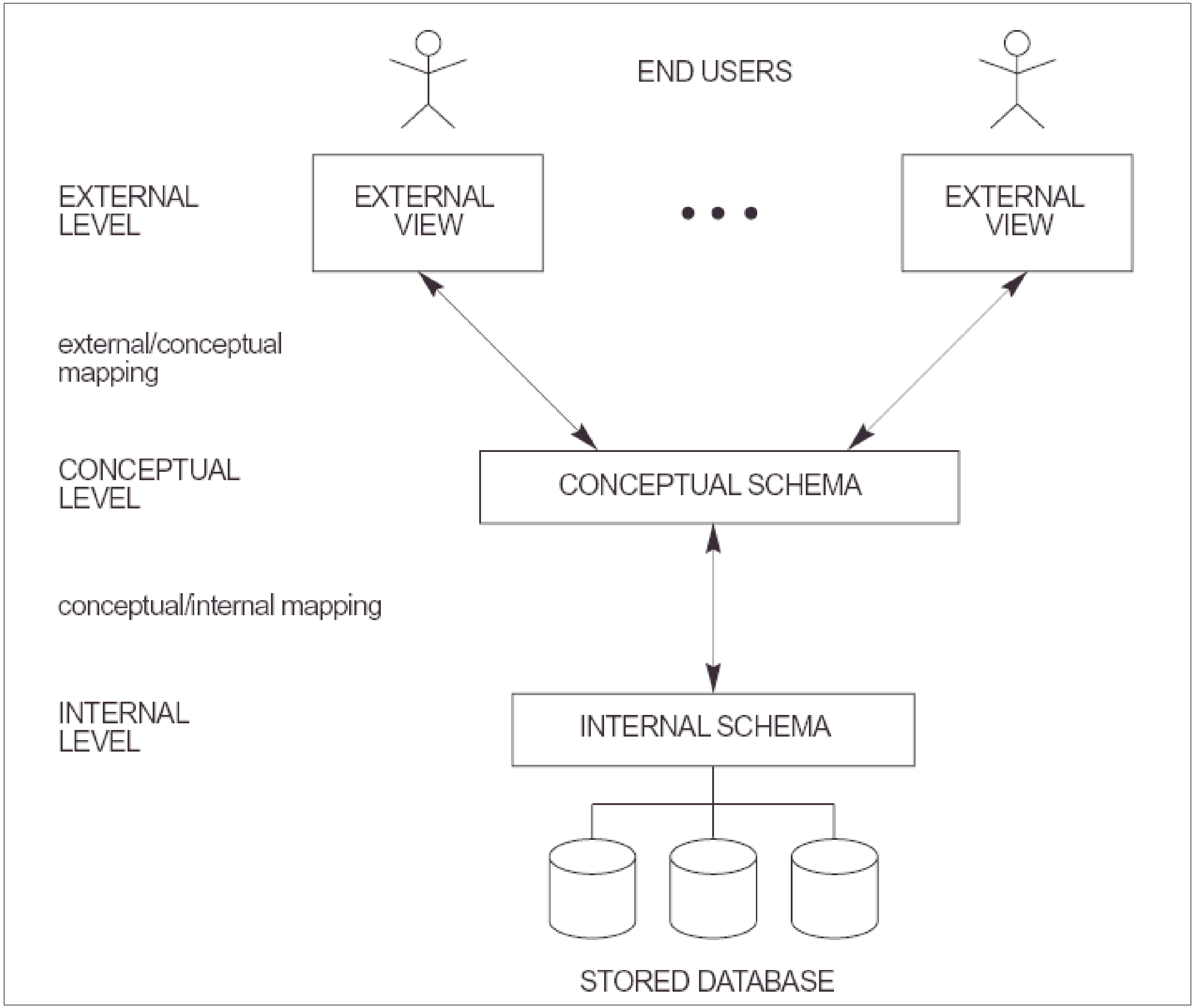
Database System Concepts And Architecture

- 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

DBMS architecture

- Three-schema-architecture or three-layer-architecture:
  - **Internal** layer: internal schema (physical)
  - **Conceptual** layer: conceptual schema (ER)
  - **External** layer: user views

- Predefined **mappings** between the layers


END USERS
EXTERNAL LEVEL
EXTERNAL VIEW
• • •
EXTERNAL VIEW
external/conceptual mapping
CONCEPTUAL LEVEL
CONCEPTUAL SCHEMA
conceptual/internal mapping
INTERNAL LEVEL
INTERNAL SCHEMA
STORED DATABASE

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
- Types:
  - **Logical** data independence
    - Change conceptual schema without changing external views
  - **Physical** data independence
    - Change physical layer without changing conceptual schema

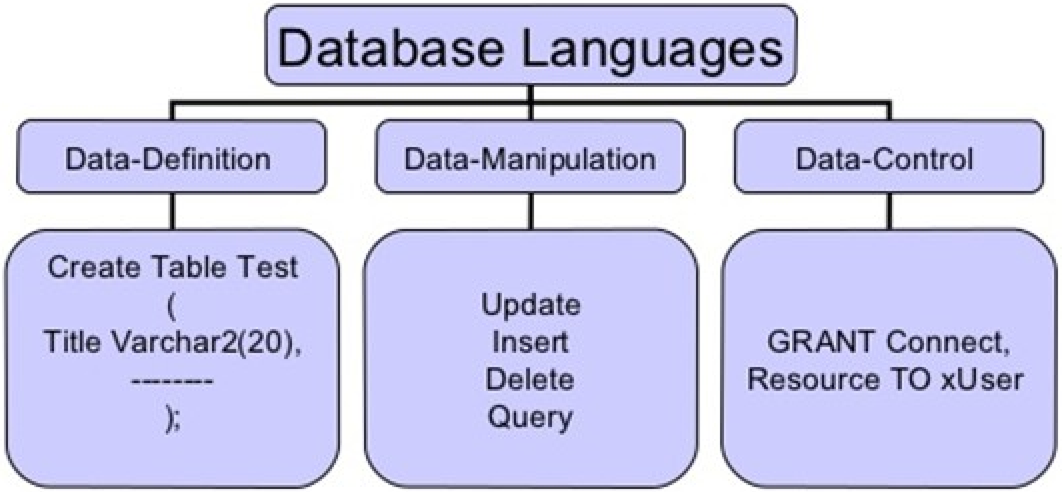
Database System Concepts And Architecture

- 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

Database languages

- Data-definition language (DDL)
  - Specify conceptual schema
- Storage-definition language (SDL)
  - Specify internal schema
- View-definition language (VDL)
  - Specify user views
- Data-manipulation language (DML)
  - Create, Retrieve, Update and Delete operations

Database Languages



Database Languages

Data-Definition

Data-Manipulation

Data-Control

Create Table Test
(
Title Varchar2(20),

);

Update
Insert
Delete
Query

GRANT Connect,
Resource TO xUser

How the Programmer Sees the DBMS

- Start with DDL to *create tables*:

```
CREATE TABLE Students (
            Name CHAR(30)
            SSN CHAR(9) PRIMARY KEY NOT NULL,
            Category CHAR(20)
)   . . .
```

- Continue with DML to *populate tables:*

```
INSERT INTO Students
VALUES(‘Charles’, ‘123456789’, ‘undergraduate’)
.  .  .  .
```

How the Programmer Sees the DBMS

- Tables:

Students:

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	…	…

Takes:

SSN	CID
123-45-6789	CSE444
123-45-6789	CSE444
234-56-7890	CSE142
	…

Courses:

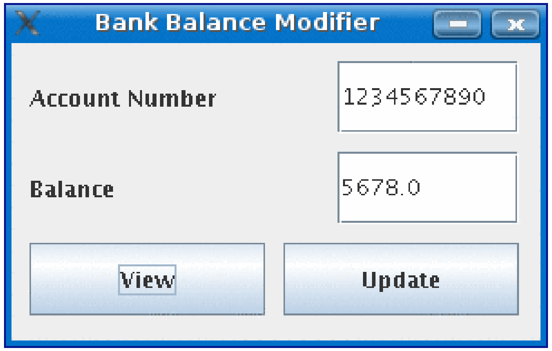
CID	Name	Quarter
CSE444	Databases	fall
CSE541	Operating systems	winter

- Still implemented as files, but behind the scenes can be quite complex

“*data independence*” = separate *logical* view from *physical implementation*

Database interfaces

- Menu-based (browsing)
- Form-based (form spec language)
- GUI
- Natural language interface
  - "Free Form" textual query
- Specialized interfaces for parametric users
  - Small frequent operations
- Interfaces for the administrator
  - Privileged commands by DBA, Security commands etc



Account Number 1234567890

Balance 5678.0

Database System Concepts And Architecture

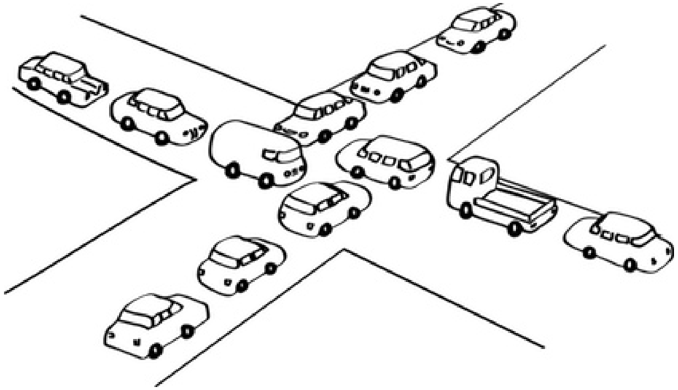
- Data Models, Schemas, and Instances
- Three-Schema Architecture and Data Independence
- Database Languages and Interfaces
- The Database System Environment
- Centralized and Client/Server Architectures for DBMSs
- Classification of Database Management Systems

The Database System Environment

- DBMS component modules
  - Buffer management
  - Stored data manager
  - DDL compiler
  - Interactive query interface
    - Query compiler
    - Query optimizer

The Database System Environment (cont'd.)

- DBMS component modules
  - Runtime database processor
  - System catalog
  - Concurrency control system
  - Backup and recovery system



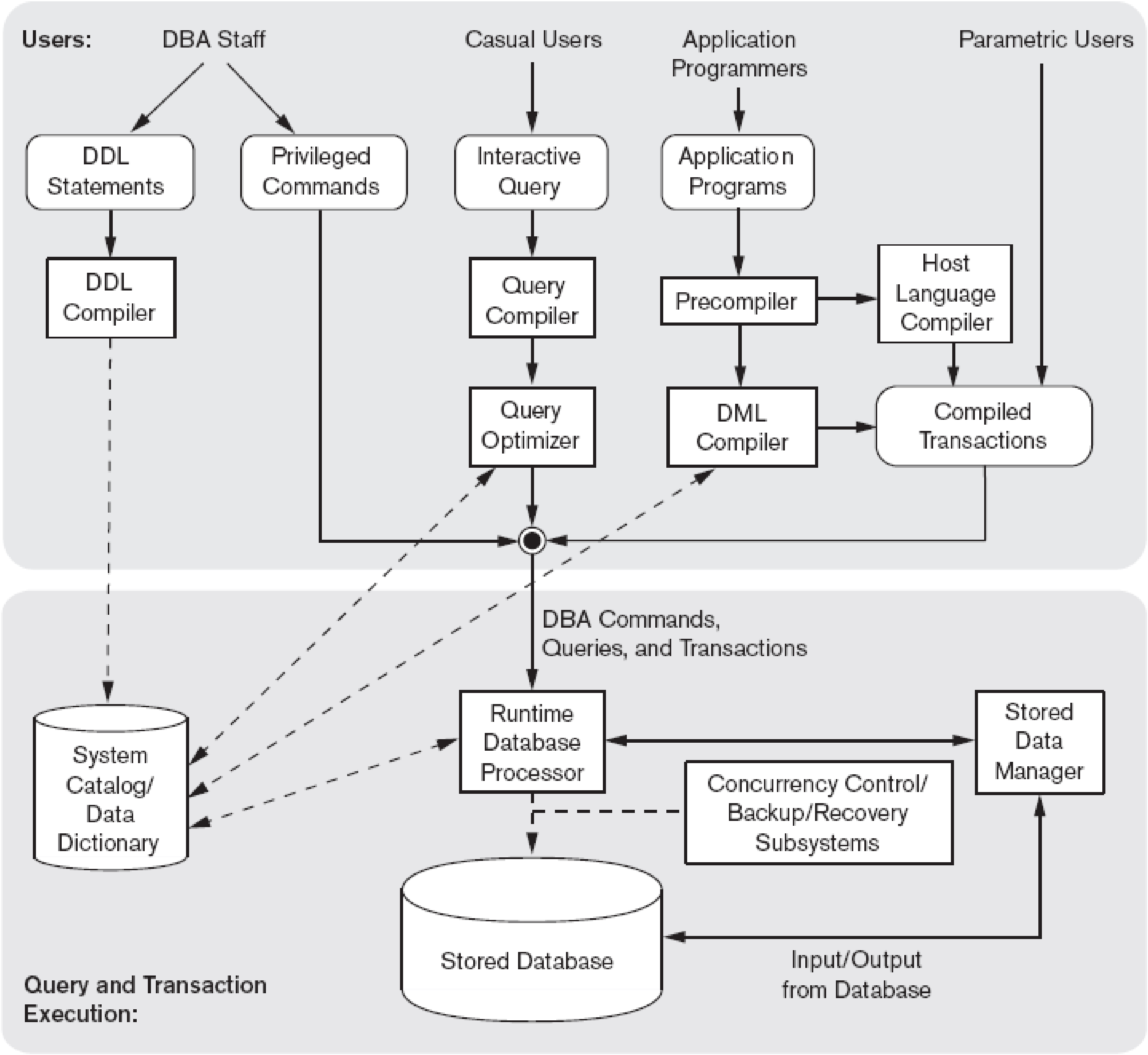


Figure 2.3
Component modules of a DBMS and their interactions.

Database system utilities

- Loading data
  - Loading existing data files
  - Transferring data from another DBMS
  - Data conversion is needed
- Backup
- Reorganisation
  - Improve performance
- Performance monitoring
  - Provide statics to DBA
- Other (sorting files, data compression, user-access monitoring, network interfacing, …)

Other tools

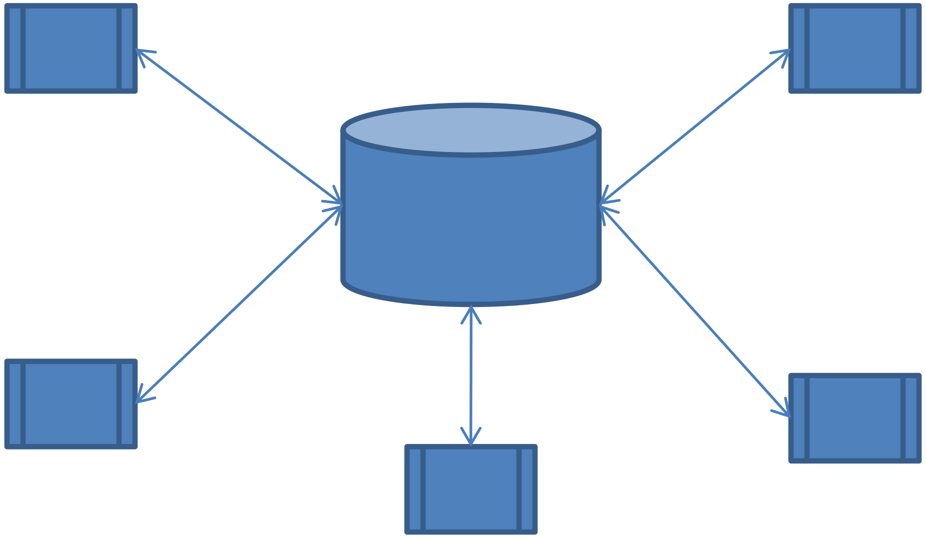
- CASE tools (for designing databases)
- Data dictionary system
  - Store design decisions, usage standards, etc
  - Used mainly by users and not by DBMS software
- (Rapid) Application development environment
  - Powerbuilder (Sybase) / JBuilder (Borland) / Visual Basic
- Communication software (DB/DC)
  - Allow users to remotely access data

Database System Concepts And Architecture

- 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

Centralized architecture

- Centralized architecture
  - DBMS runs on a central computer (mainframe) to which (dumb) terminals are connected



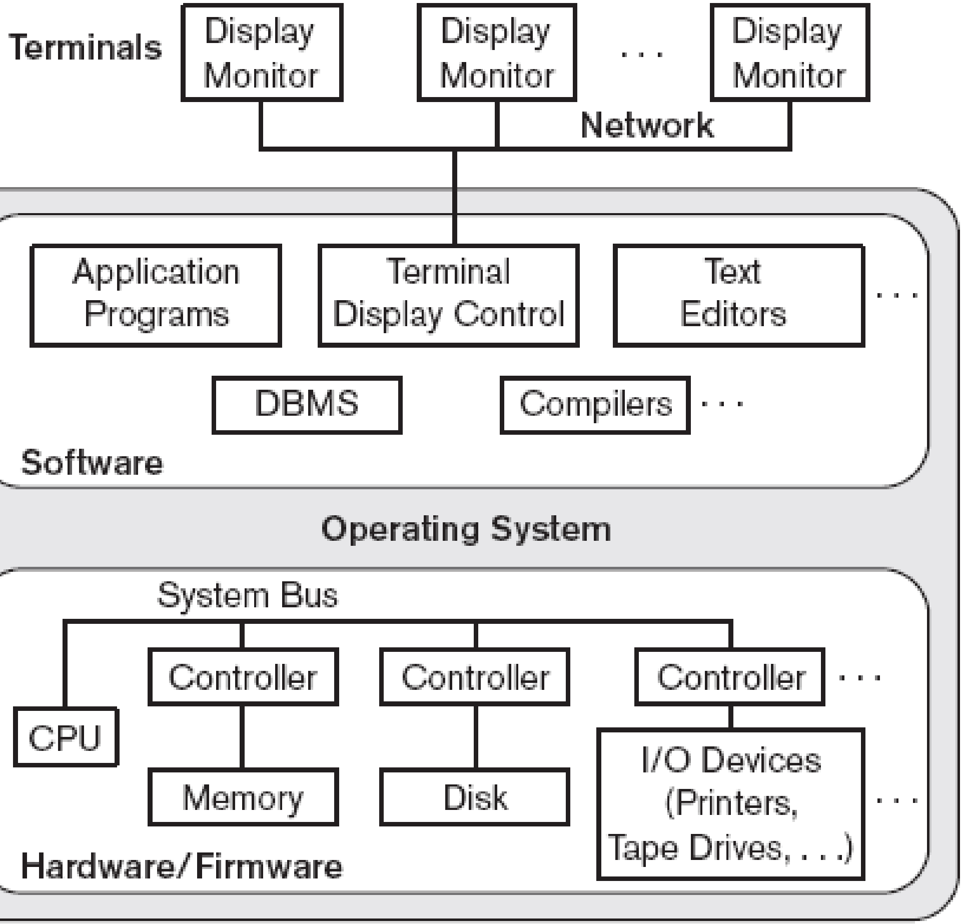


Figure 2.4
A physical centralized architecture.

Introduction of LAN and PC’s

Terminals became smart PC’s and were able to take over certain tasks of the DBMS

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**

Basic Client/Server Architectures (cont'd.)

- **Client machines**
  - Provide user with:
    - Appropriate interfaces to utilize these servers
    - Local processing power to run local applications

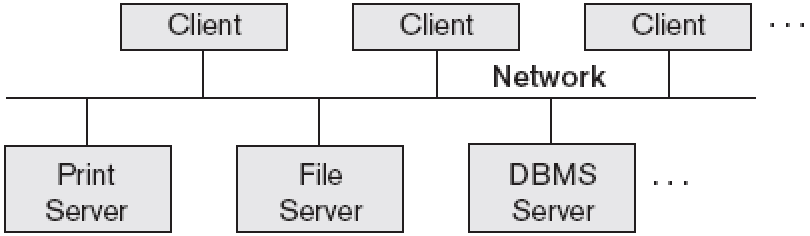


Figure 2.5
Logical two-tier client/server architecture.

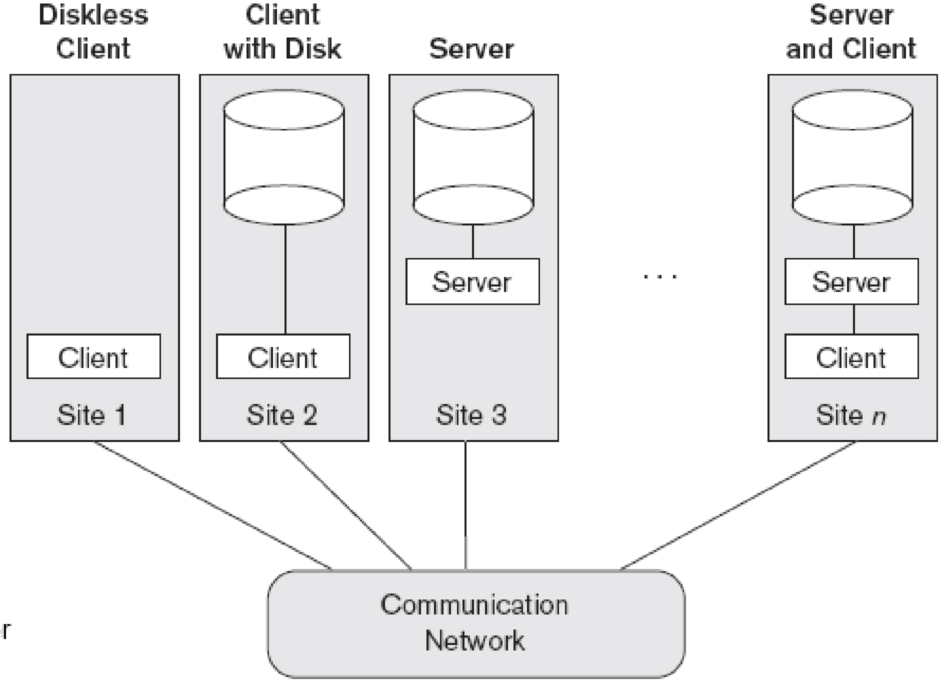


Figure 2.6
Physical two-tier client/server architecture.

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 architecture for DBMSs

- **Server** machine runs DBMS (Oracle, SQLserver, MySQL)
  - Contains all data and meta-data
  - Runs queries and updates
- **Client** machine runs application software and communicates with Server
  - Via ODBC, JDBC, .NET, SQLnet, etc.

Three-tier architecture

- Database **Servers** running DBMS
- **Clients** run Web-browser (*thin* client)
- In between are ***Application Servers*** that run the database applications (process management, business logic)

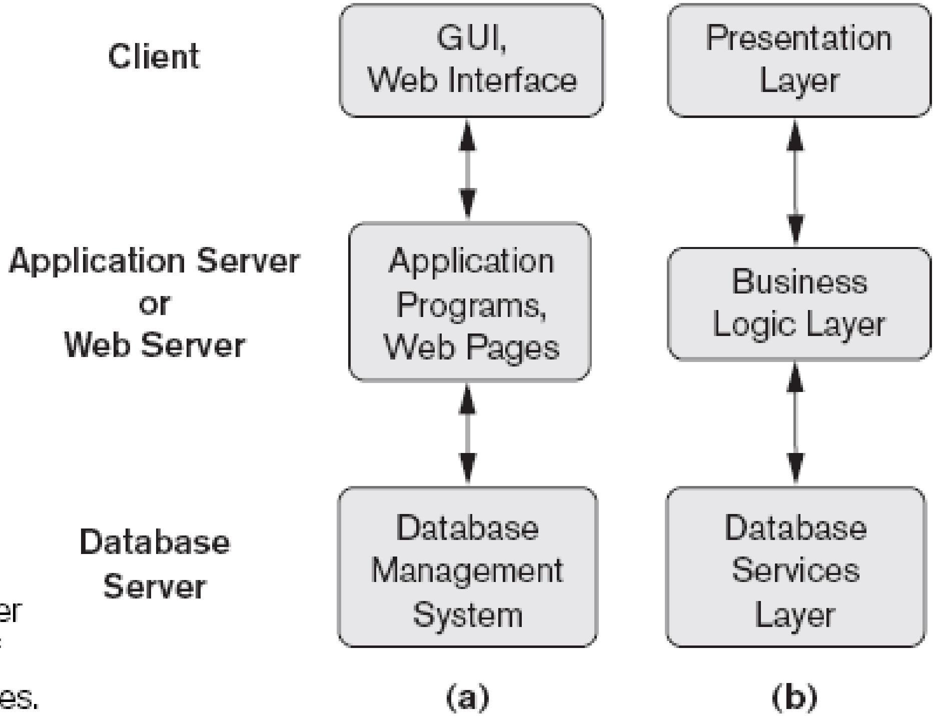


Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

Database System Concepts And Architecture

- 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

DBMS Classification

- Criteria
  - Data model (relational, network, object)
  - Number of users (1, more, many)
  - Number of places (centralized, distributed)
  - Costs (open source, commercial)
  - General / specialized
  - etc

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, cost

Quiz

- If you were designing a Web-based system to make airline reservations and sell airline tickets, which DBMS architecture would you choose? Why?