# CSE 5330

## DATABASE SYS & FILE STRUCTURES
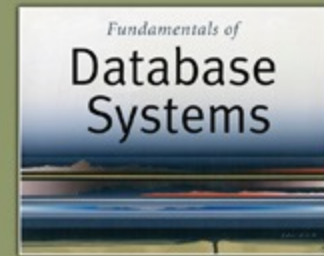
Fundamentals of
Database Systems
5th Edition
Elmasri / Navathe

PEARSON
Addison Wesley

# Nadra Guizani

- nadra.guizani@uta.edu

- Office Hours:W – 11am – 1pm, 2pm – 4 pm. Or by appointment.
- Material is posted and updated on **Canvas**

- Announcements on **Teams**

- **Textbook:** *Fundamentals of Database Systems, Seventh Edition* by Elmasri/Navathe

# GTA -- TBD

- **GTA –** TBA

- **Grader Email:**
- **GTA Office Hours:**

# Grading Breakdown

| Exam 1 | 20% | March 10th |
|--------|-----|------------|
| Exam 2 | 20% | May 3rd |
| Project 1 | 30% | |
| Project 2 | 30% | |

- <u>Projects will require</u>: Python programming **or** JAVA programming using JDBC, **or** C/C++/C# programming with ODBC **or** other programming languages upon approval, and the use of database systems such as MariaDB **or** MySQL **or** PostGRES or Oracle.

FUNDAMENTALS OF
DATABASE SYSTEMS

7TH Edition

ELMASRI • NAVATHE
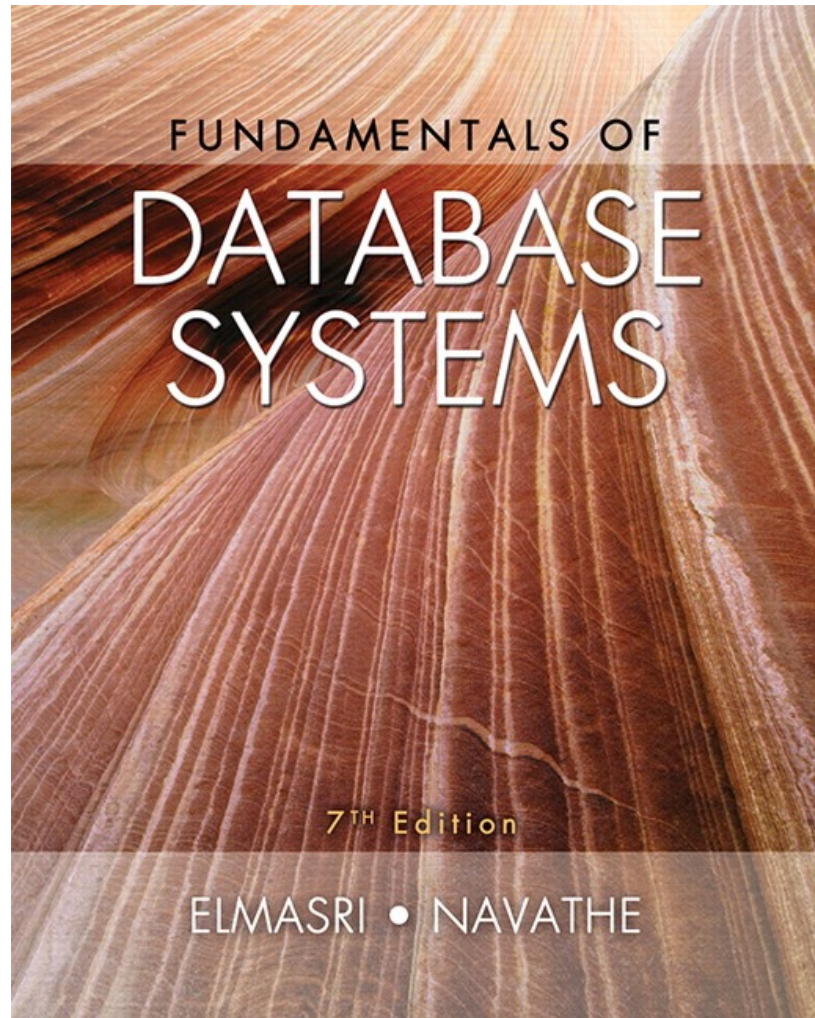
# CHAPTER 1

# Databases and Database Users

# OUTLINE

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- Historical Development of Database Technology
- Extending Database Capabilities
- When Not to Use Databases

# Types of Databases and Database Applications

- Traditional Applications:
    - Numeric and Textual Databases
- More Recent Applications:
    - Multimedia Databases
    - Geographic Information Systems (GIS)
    - Biological and Genome Databases
    - Data Warehouses
    - Mobile databases
    - Real-time and Active Databases
- First part of book focuses on traditional applications
- *A number of recent applications are described later in the book (for example, Chapters 24,25,26,27,28,29)*

# Recent Developments (1)

- Social Networks started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in systems such as:

- Facebook

- Twitter

- Linked-In

- All of the above constitutes data

- Search Engines, Google, Bing, Yahoo: collect their own repository of web pages for searching purposes

# Recent Developments (2)

- New Technologies are emerging from the so-called non-database software vendors to manage vast amounts of data generated on the web:

- **Big Data** storage systems involving large clusters of distributed computers (Chapter 25)

- **NOSQL** (Non-SQL, Not Only SQL) systems (Chapter 24)

- A large amount of data now resides on the "cloud" which means it is in huge data centers using thousands of machines.
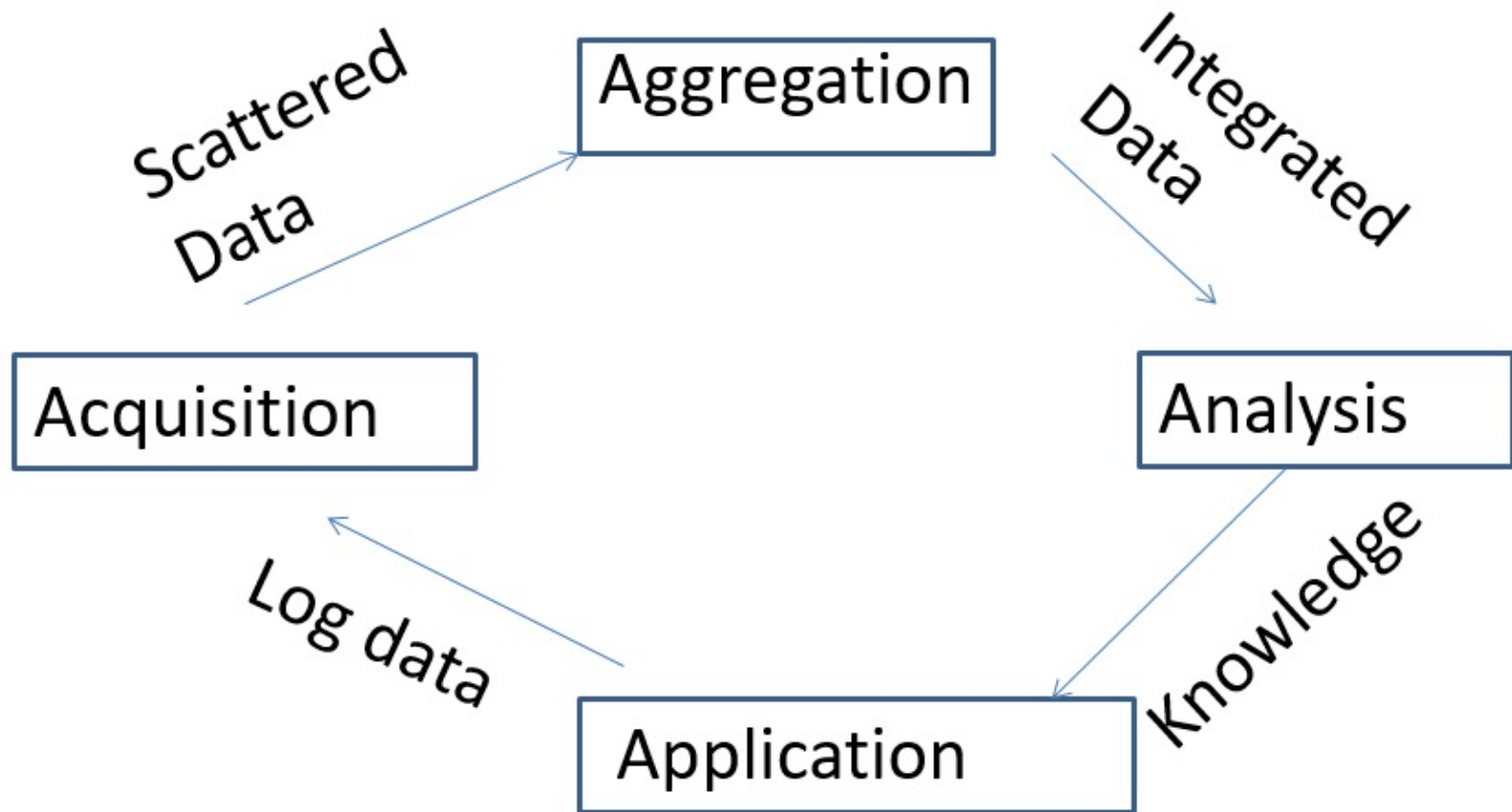
# What is "big data"?

- "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization"  (Gartner 2012)

- Bottom line: Any data that exceeds our current capability of processing can be regarded as "big"

  - Complicated (intelligent) analysis of data may make a small data "appear" to be "big"
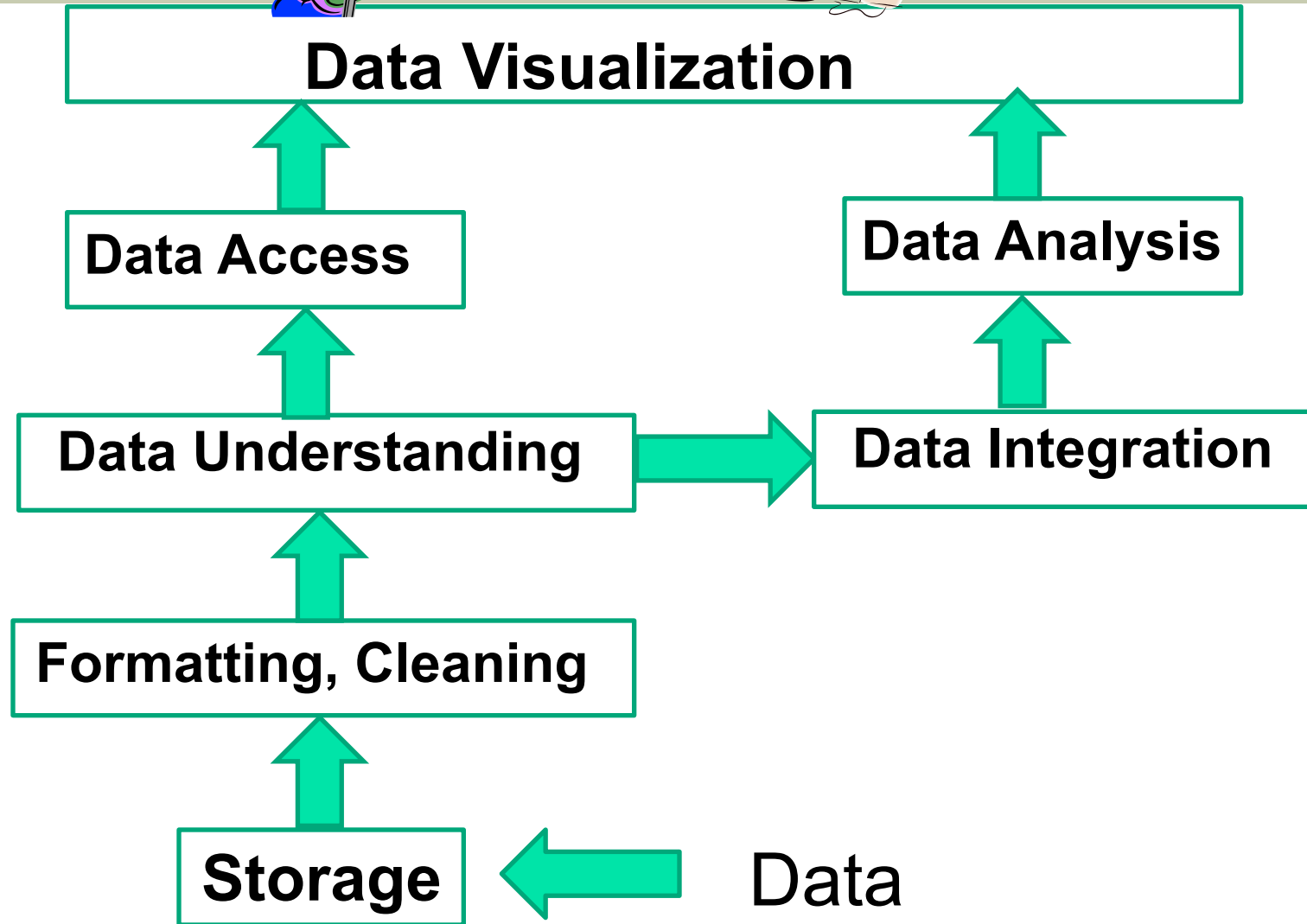
# Why is "big data" a "big deal"?

- **Government**
- **Private Sector**
  - **Walmart handles more than 1 million customer transactions every hour, which is imported into databases estimated to contain more than 2.5 petabytes of data**
  - **Facebook handles 40 billion photos from its user base**
  - Falcon Credit Card Fraud Detection System protects 2.1 billion active accounts world-wide
- **Science**
  - Large Synoptic Survey Telescope will generate 140 Terabyte of data every 5 days
  - Biomedical computation like decoding human Genome and personalized medicine

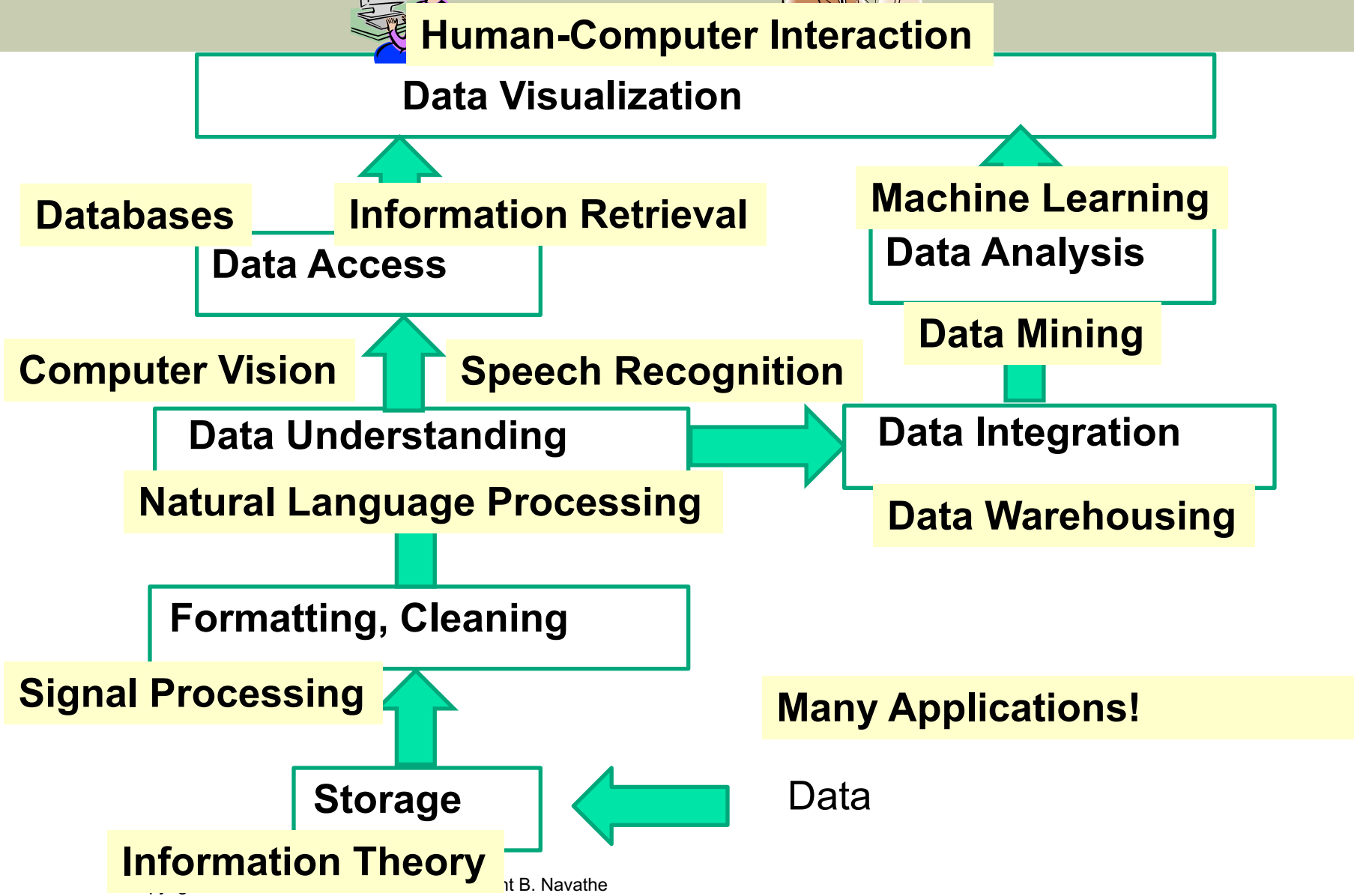# Lifecycle of Data: 4 "A"s

# Computational View of  Big Data



**Data Visualization**

**Data Access**    **Data Analysis**

**Data Understanding** → **Data Integration**

**Formatting, Cleaning**

**Storage** ← Data

# Big Data & Related Disciplines

**Human-Computer Interaction**

Data Visualization

**Databases**       **Information Retrieval**

Data Access

**Computer Vision**       **Speech Recognition**

Data Understanding

**Natural Language Processing**

Formatting, Cleaning

**Signal Processing**

Storage

**Information Theory**

**Machine Learning**

Data Analysis

**Data Mining**

Data Integration

**Data Warehousing**

**Many Applications!**

Data

# Basic Definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts that can be recorded and have an implicit meaning.
- **Mini-world:**
  - Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software package/system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
  - The DBMS software together with the data itself. Sometimes, the applications are also included.

# Impact of Databases and Database Technology

- **Businesses:** Banking, Insurance, Retail, Transportation, Healthcare, Manufacturing

- **Service industries:** Financial, Real-estate, Legal, Electronic Commerce, Small businesses

- **Education :** Resources for content and Delivery

- **More recently:** Social Networks, Environmental and Scientific Applications, Medicine and Genetics

- **Personalized applications:** based on smart mobile devices

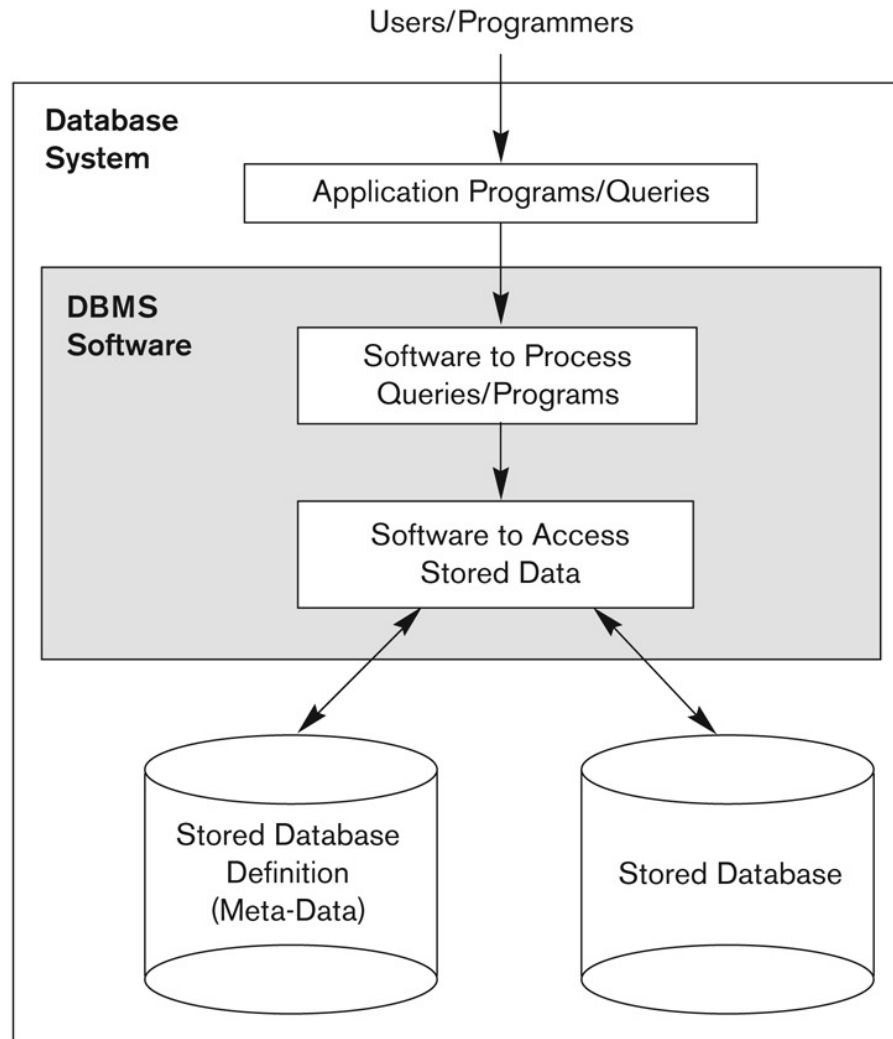# Simplified database system environment



**Figure 1.1**
A simplified database system environment.

# What a DBMS Facilitates

- *Define* a particular database in terms of its data types, structures, and constraints
- *Construct* or load the initial database contents on a secondary storage medium
- *Manipulating* the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing the database through Web applications
- *Processing* and *sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

# Other DBMS Functionalities

- DBMS may additionally provide:
    - Protection or Security measures to prevent unauthorized access
    - "Active" processing to take internal actions on data
    - Presentation and visualization of data
    - Maintenance of the database and associated programs over the lifetime of the database application

# Example of a Database
# (with a Conceptual Data Model)

- **Mini-world for the example:**
    - Part of a UNIVERSITY environment
- **Some mini-world *entities*:**
    - STUDENTs
    - COURSEs
    - SECTIONs (of COURSEs)
    - (Academic) DEPARTMENTs
    - INSTRUCTORs

# Example of a Database (with a Conceptual Data Model)

- **Some mini-world *relationships*:**
    - SECTIONs *are of specific* COURSEs
    - STUDENTs *take* SECTIONs
    - COURSEs *have  prerequisite* COURSEs
    - INSTRUCTORs *teach*  SECTIONs
    - COURSEs *are offered by*  DEPARTMENTs
    - STUDENTs *major in*  DEPARTMENTs

- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP data model (see Chapters 3, 4)

# Example of a Simple Database

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 04 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

**Figure 1.2**
A database that stores student and course information.

# Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
    - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
    - The description is called **meta-data***.
    - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
    - Called **program-data independence**.
    - Allows changing data structures and storage organization without having to change the DBMS access programs
        - E.g., ADTs

# Example of a Simplified Database Catalog

**RELATIONS**

| Relation_name | No_of_columns |
|---|---|
| STUDENT | 4 |
| COURSE | 4 |
| SECTION | 5 |
| GRADE_REPORT | 3 |
| PREREQUISITE | 2 |

**COLUMNS**

| Column_name | Data_type | Belongs_to_relation |
|---|---|---|
| Name | Character (30) | STUDENT |
| Student_number | Character (4) | STUDENT |
| Class | Integer (1) | STUDENT |
| Major | Major_type | STUDENT |
| Course_name | Character (10) | COURSE |
| Course_number | XXXXNNNN | COURSE |
| …. | …. | ….. |
| …. | …. | ….. |
| …. | …. | ….. |
| Prerequisite_number | XXXXNNNN | PREREQUISITE |

**Figure 1.3**
An example of a database catalog for the database in Figure 1.2.

*Note*: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

# Main Characteristics of the Database Approach (continued)

- **Data Abstraction:**

  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.

  - Programs refer to the data model constructs rather than data storage details

- **Support of multiple views of the data:**

  - Each user may see a different view of the database, which describes **only** the data of interest to that user.

# Main Characteristics of the Database Approach (continued)

- **Sharing of data and multi-user transaction processing:**
    - Allowing a set of **concurrent users** to retrieve from and to update the database.
    - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
    - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
    - **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

# Database Users

- Users may be divided into
  - Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene"), and
  - Those who design and develop the DBMS software and related tools, and the computer systems operators (called "Workers Behind the Scene").

# Database Users – Actors on the Scene

- ## Actors on the scene
  - ### Database administrators:
    - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
  - ### Database designers:
    - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

# Database End Users

- Actors on the scene (continued)
  - **End-users:** They use the data for queries, reports and some of them update the database content. End-users can be categorized into:
    - **Casual**: access database occasionally when needed
    - **Naïve** or Parametric: they make up a large section of the end-user population.
      - They use previously well-defined functions in the form of "canned transactions" against the database.
      - Users of Mobile Apps mostly fall in this category
      - Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations.
      - Social Media Users post and read information from websites

# Database End Users (continued)

- **Sophisticated:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.

- **Stand-alone:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is the user of a tax program that creates its own internal database.
  - Another example is a user that maintains a database of personal photos and videos.

# Database Users – Actors on the Scene (continued)

- **System Analysts and Application Developers**

  This category currently accounts for a very large proportion of the IT work force.

  - **System Analysts**: They understand the user requirements of naïve and sophisticated users and design applications including canned transactions to meet those requirements.

  - **Application Programmers:** Implement the specifications developed by analysts and test and debug them before deployment.

  - **Business Analysts**: There is an increasing need for such people who can analyze vast amounts of business data and real-time data ("Big Data") for better decision making related to planning, advertising, marketing etc.

# Database Users – Actors behind the Scene

- **System Designers and Implementors:** Design and implement DBMS packages in the form of modules and interfaces and test and debug them. The DBMS must interface with applications, language compilers, operating system components, etc.

- **Tool Developers**: Design and implement software systems called tools for modeling and designing databases, performance monitoring, prototyping, test data generation, user interface creation, simulation etc. that facilitate building of applications and allow using database effectively.

- **Operators and Maintenance Personnel:** They manage the actual running and maintenance of the database system hardware and software environment.

# Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data. Only the DBA staff uses privileged commands and facilities.
- Providing persistent storage for program Objects
  - E.g., Object-oriented DBMSs make program objects persistent– see Chapter 12.
- Providing storage structures (e.g. indexes) for efficient query processing – see Chapter 17.

# Advantages of Using the Database Approach (continued)

- Providing optimization of queries for efficient processing
- Providing backup and recovery services
- Providing multiple interfaces to different classes of users
- Representing complex relationships among data
- Enforcing integrity constraints on the database
- Drawing inferences and actions from the stored data using deductive and active rules and triggers

# Additional Implications of Using the Database Approach

- Potential for enforcing standards:
  - This is very crucial for the success of database applications in large organizations. **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - Incremental time to add each new application is reduced.

# Historical Development of Database Technology

- Early database applications:
  - The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.
  - A bulk of the worldwide database processing still occurs using these models, particularly, the hierarchical model using IBM's IMS system.
- Relational model based systems:
  - Relational model was originally introduced in 1970, was heavily researched and experimented within IBM Research and several universities.
  - Relational DBMS Products emerged in the early 1980s.

# Historical Development of Database Technology (continued)

- Object-oriented and emerging applications:
  - Object-Oriented Database Management Systems (OODBMSs) were introduced in late 1980s and early 1990s to cater to the need of complex data processing in CAD and other applications.
    - Their use has not taken off much
  - Many relational DBMSs have incorporated object database concepts, leading to a new category called *object-relational* DBMSs (ORDBMSs)
  - *Extended relational* systems add further capabilities (e.g. for multimedia data, text, XML, and other data types)

# Extending Database Capabilities (1)

- New functionality is being added to DBMSs in the following areas:
    - Scientific Applications – Physics, Chemistry, Biology - Genetics
    - Spatial: weather, earth and atmospheric sciences and astronomy
    - XML (eXtensible Markup Language)
    - Image storage and management
    - Audio and video data management
    - Time series and historical data management
- The above gives rise to *new research and development* in incorporating new data types, complex data structures, new operations and storage and indexing schemes in database systems.

# When not to use a DBMS

- Main inhibitors (costs) of using a DBMS:
    - High initial investment and possible need for additional hardware
    - Overhead for providing generality, security, concurrency control, recovery, and integrity functions
- When a DBMS may be unnecessary:
    - If the database and applications are simple, well defined, and not expected to change
    - If access to data by multiple users is not required
- When a DBMS may be infeasible
    - In embedded systems where a general purpose DBMS may not fit in available storage

# When not to use a DBMS

- When no DBMS may suffice:

  - If there are stringent real-time requirements that may not be met because of DBMS overhead (e.g., telephone switching systems)

  - If the database system is not able to handle the complexity of data because of modeling limitations (e.g., in complex genome and protein databases)

  - If the database users need special operations not supported by the DBMS (e.g., GIS and location based services).

# Chapter Summary

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- Historical Development of Database Technology
- Extending Database Capabilities
- When Not to Use Databases