CHAPTER 1

Databases and Database Users

Content

- Introduction
- An Example
- Characteristics of the database Approach
- Actors on the scene
- Workers behind the scene
- Advantage of using the DBMS aaproach

DataBase

- A database is a collection of related data.
- By data, we mean known facts that can be recorded and that have implicit meaning.

- For example,

consider the names, telephone numbers, and addresses of the people you   know.

Nowadays, this data is typically stored in mobile phones, which have their own simple database software.

- This data can also be recorded in an indexed address book or stored on a hard drive, using a personal computer and software such as Microsoft Access or Excel.
- This collection of related data with an implicit meaning is a database.

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

What is data, database, DBMS

- <mark>Data</mark>: Known facts that can be recorded and have an implicit meaning; raw
- <mark>Database</mark>: a highly organized, interrelated, and structured set of data about a particular enterprise
    - Controlled by a database management system (DBMS)
- <mark>DBMS</mark>
    - Set of programs to access the data
    - An environment that is both *convenient* and *efficient* to use
- Database systems are used to manage collections of data that are:
    - Highly valuable
    - Relatively large
    - Accessed by multiple users and applications, often at the same time.
- <mark>A modern database</mark> system is a complex software system whose task is to manage a large, complex collection of data.
- Databases touch all aspects of our lives

Types of Databases and Database Applications

- Traditional applications:
  - Numeric and textual databases
- More recent applications:
  - Multimedia databases(image, audio,clip, and video stream digitally)
  - Geographic Information Systems (GIS)
     (store and analyze maps, weather data, and satellite images)
  - Data warehouses مخزنand online analytical processing(OLAP)
     (extract and analyze useful business information from every large database and support decision making)
  - Mobile databases

Recent Developments

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

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.

DBMS

- Is a collection of programs that enables users to create and maintain a database.

- The DBMS is a general purpose software system that facilitate the process of defining, constructing , manipulating and sharing database among various users and applications.

-

- Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data.

- Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS.

- Manipulating a database includes functions such as querying the database to retrieve specific data, updating the data-base to reflect changes in the miniworld, and generating reports from the data.

- Sharing a database allows multiple users and programs to access the database simultaneously

- An application program accesses the database by sending queries or requests for data to the DBMS.
- A query typically causes some data to be retrieved;
- A transaction عملية may cause some data to be read and some data to be written into the database.
- Other important functions provided by the DBMS include protecting the database and maintaining it over a long period of time. Protection includes system protection against hardware or software malfunction (or crashes) and security protection against unauthorized or malicious access

- A typical large database may have a life cycle of many years, so the DBMS must be able to <mark>maintain</mark> the database system by allowing the system to evolve as requirements change over time

Example

- UNIVERSITY database

    information concerning students , courses and grades in a university environment.

- **Data records:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - Grade_report
  - prerequisites

• **Example , cont.,**

- Examples of queries:
  - Retrieve the transcript
  - Lists names of students who took the section of the 'database' course offered in fall 2008 and their grades in the section
  - Lists prerequisites of the database course

Examples of updates:

- Change the class of 'Smith to sophomore

- - create a new section for the database course for this semester

- - enter a grade of ' A' in the Smith in the 'database'

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.

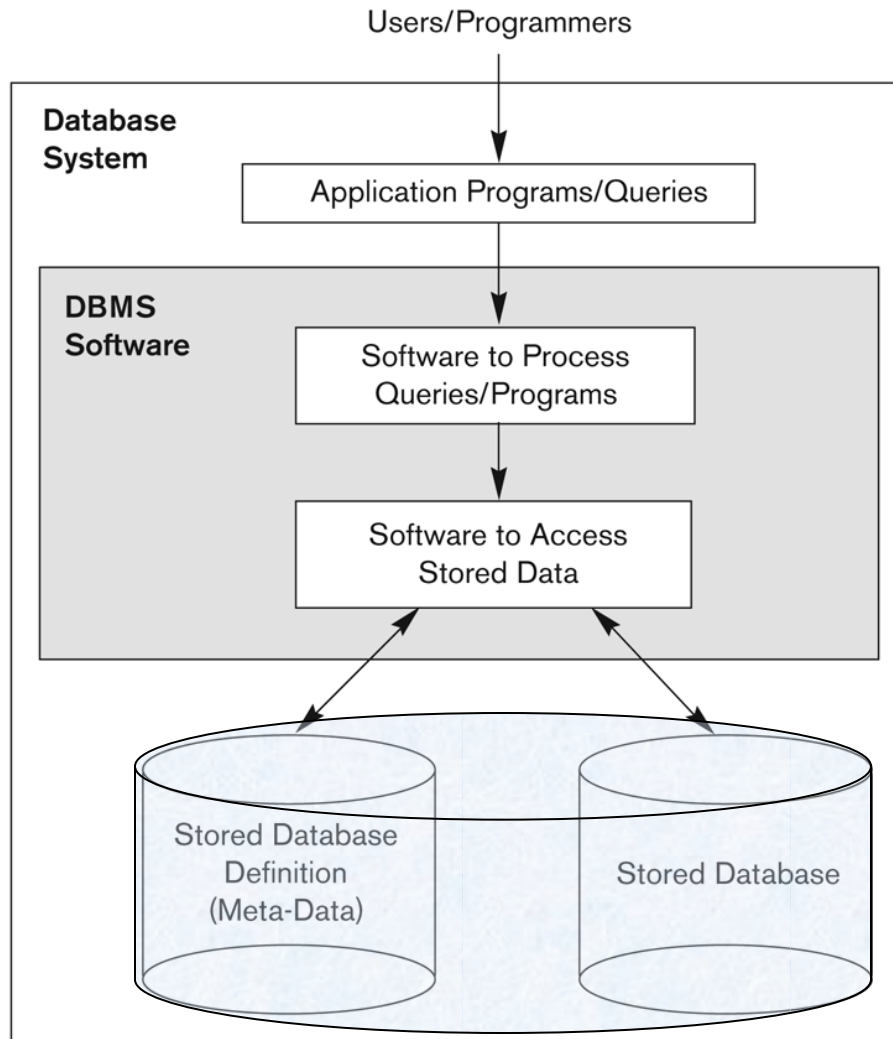# A simplified architecture for a database system

Figure 1.1
A simplified database system environment.

Characteristics of the Database Approach

- In traditional file processing, each user defines and implements the files needed for a specific software application as part of programming the application.

- For example,
- one user, the grade reporting office, may keep files on students and their grades.
- Programs to print a student's transcript and to enter new grades are implemented
- as part of the application.
- A second user, the accounting office, may keep track of students' fees and their payments.
- Although both users are interested in data about students, each user maintains separate files—and programs to manipulate these files—because each requires some data not available from the other user's files.
- This redundancy in defining and storing data results in wasted storage space and
- in redundant efforts to maintain common up-to-date data.
- In the database approach, a single repository maintains data that is defined once
- and then accessed by various users repeatedly through queries, transactions, and
- application programs

Disadvantages of File Processing

- Program-Data Dependence
  - All programs maintain metadata for each file they use
- Duplication of Data
  - Different systems/programs have separate copies of the same data.
- Limited Data Sharing
  - No centralized control of data
- Length Development Times
  - Programmers must design their own files formats
- Excessive Program Maintenance
  - 0% of information systems budget

Solution :Database Approach

- Central repositor of shared data
- Data is managed by a controlling agent
- Stored in a standardized , convenient form

Characteristics of the Database Approach cont.,

- Self-describing nature of a database system

- Insulation between programs and data, and data abstraction

- Support of multiple views of the data

- Sharing of data and multiuser transaction processing

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

# 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.

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
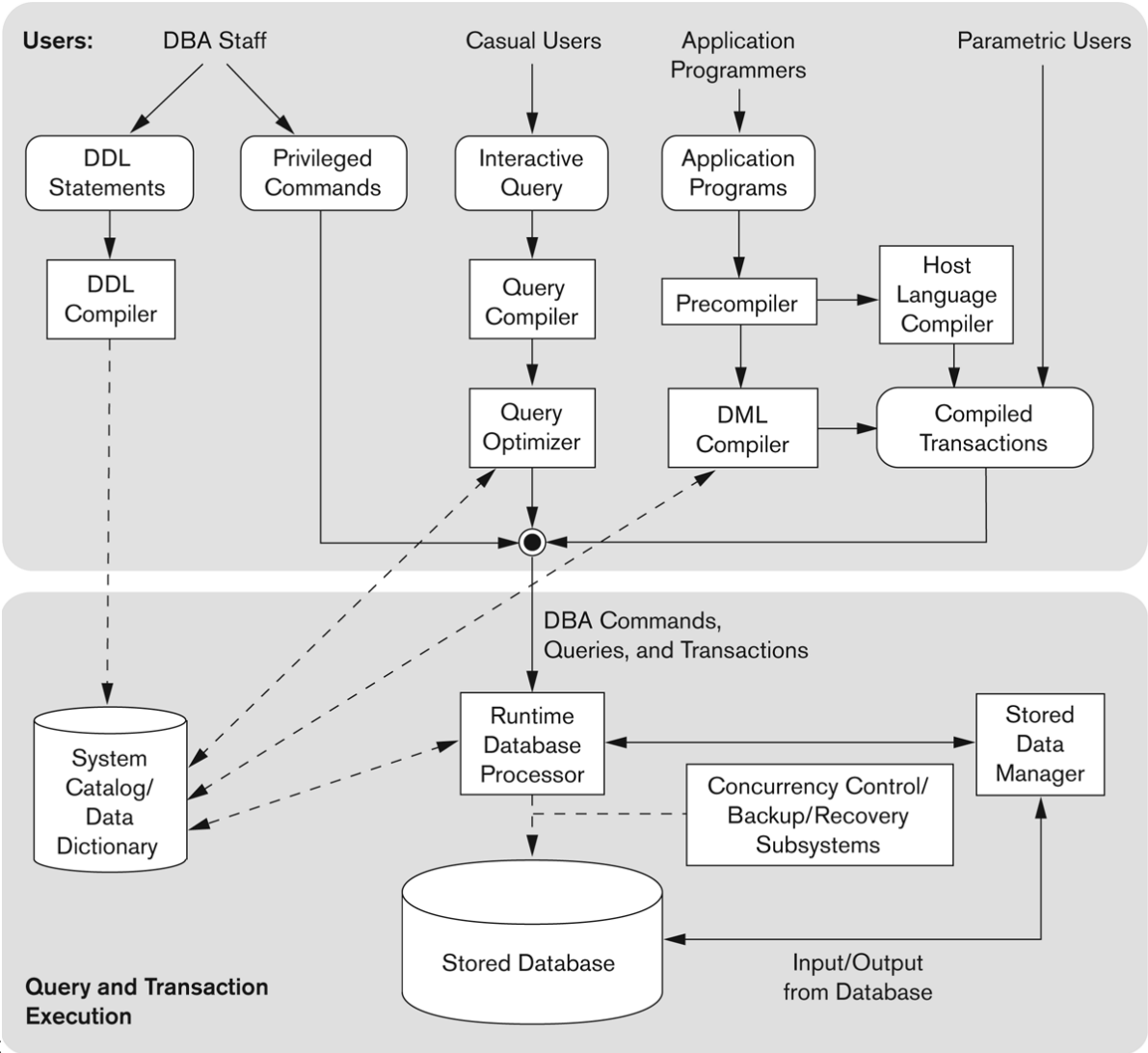
A simplified architecture for a database system

View level: what application programs see; views can also hide information (such as an instructor's salary) for security purposes.
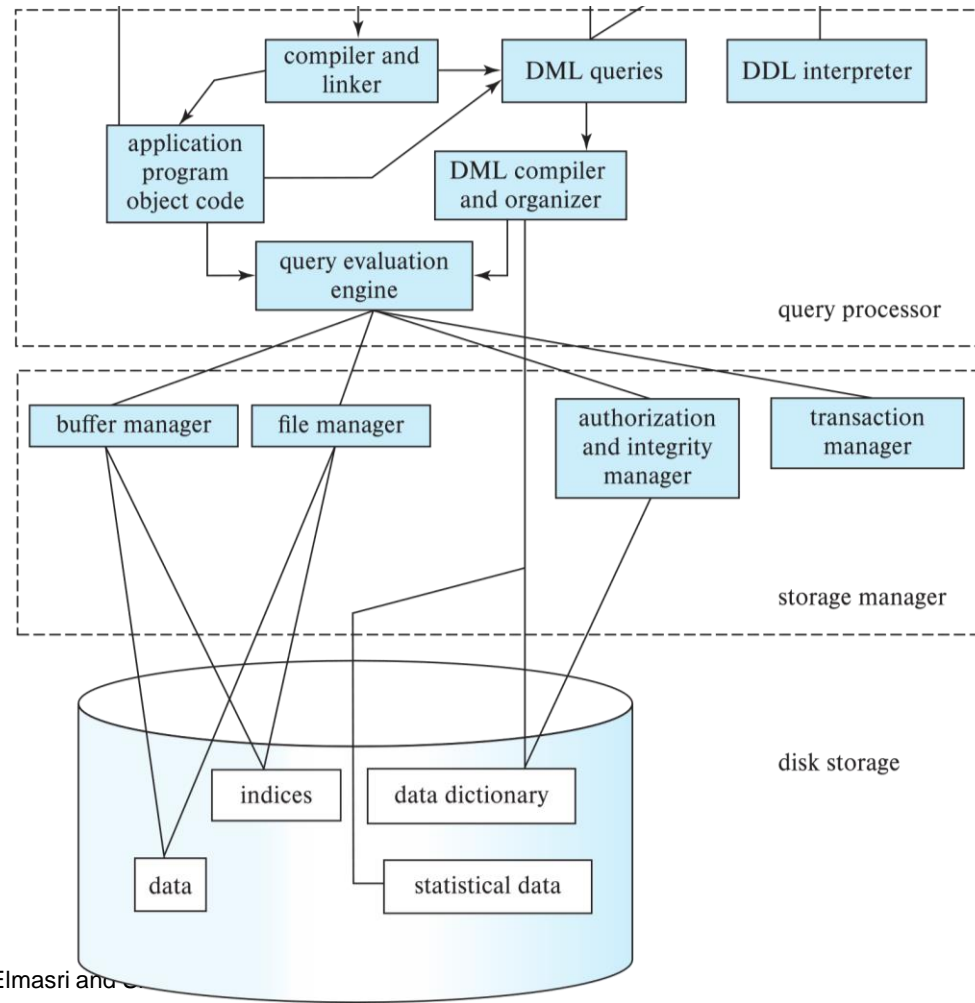
view level

| view 1 | view 2 | · · · | view n |
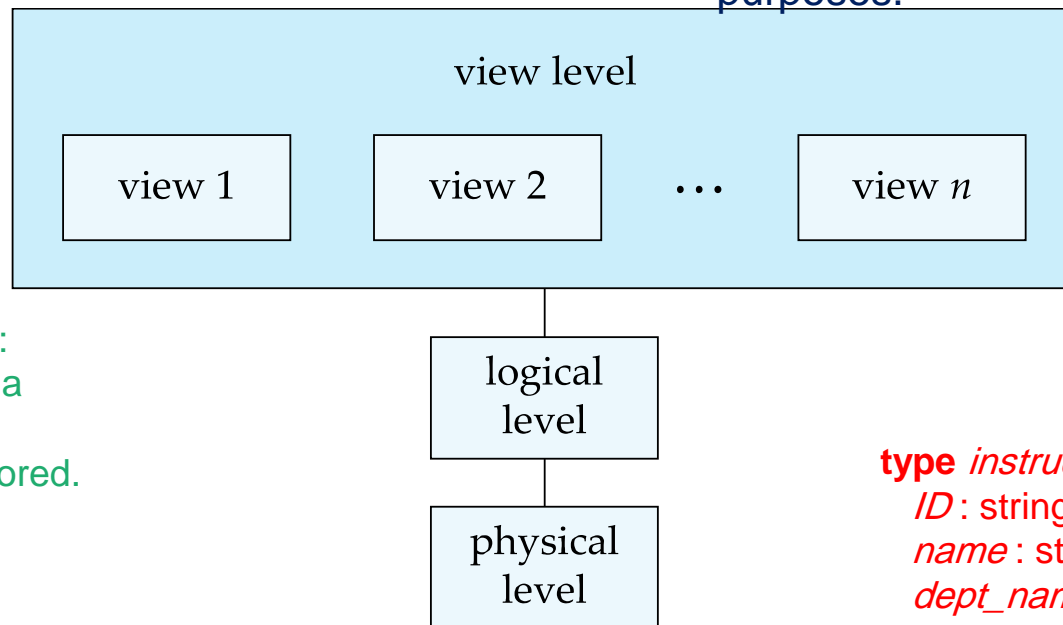
Physical level: describes how a record (e.g., instructor) is stored.

logical level

physical level

type *instructor* = **record**
  ID : string;
  name : string;
  dept_name : string;
  salary : integer;
end;

A simplified architecture for a database system

Copyright

A simplified architecture for a database system

What a DBMS Facilitates خدمات

- *Define* a particular database in terms of its <mark>data types</mark>, <mark>structures</mark>, and <mark>constraints</mark>

- *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

Application Programs and DBMS

- Applications interact with a database by generating
  - Queries: that access different parts of data and formulate the result of a request
  - Transactions: that may read some data and "update" certain values or generate new data and store that in the database

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

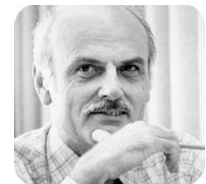# Example of a Database (with a Conceptual Data Model)

- **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 (ER) data or UML class model (see Chapters 3, 4)

The relational model

E.F. "Ted" Codd

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)

- **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; 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
  - 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.

■Slide 1-39

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:
    - **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.

Additional Implications of Using the Database Approach (continued)

- Flexibility to change data structures:
  - Database structure may evolve as new requirements are defined.

- Availability of current information:
  - Extremely important for on-line transaction systems such as shopping, airline, hotel, car reservations.

- Economies of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

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)

Historical Development of Database Technology (continued)

- Data on the Web and e-commerce applications:
  - Web contains data in HTML (Hypertext markup language) with links among pages
  - Has given rise to a new set of applications and E-commerce is using new standards like XML (eXtended Markup Language) (see Ch. 13).
  - Script programming languages such as PHP and JavaScript allow generation of dynamic Web pages that are partially generated from a database (see Ch. 11).
    - Also allow database updates through Web pages

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