

# Databases and Database Users

Part 1

# Introduction

- A database is a collection of related data.
- Implicit properties:
  - A database represents some aspect of the real world, sometimes called the **miniworld** or the **universe of discourse (UoD)**. Changes to the miniworld are reflected in the database.
  - A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
  - A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested

- A database can be of any size and complexity.
  - For example, the list of names and addresses may consist of only a few hundred records, each with a simple structure.
  - On the other hand, the computerized catalog of a large library may contain half a million entries organized under different categories—by primary author's last name, by subject, by book title—with each category organized alphabetically.
  - A database of even greater size and complexity is maintained by the Internal Revenue Service (IRS) to monitor tax forms filed by U.S. taxpayers.
  - An example of a large commercial database is Amazon.com.

- A database may be generated and maintained manually or it may be computerized.

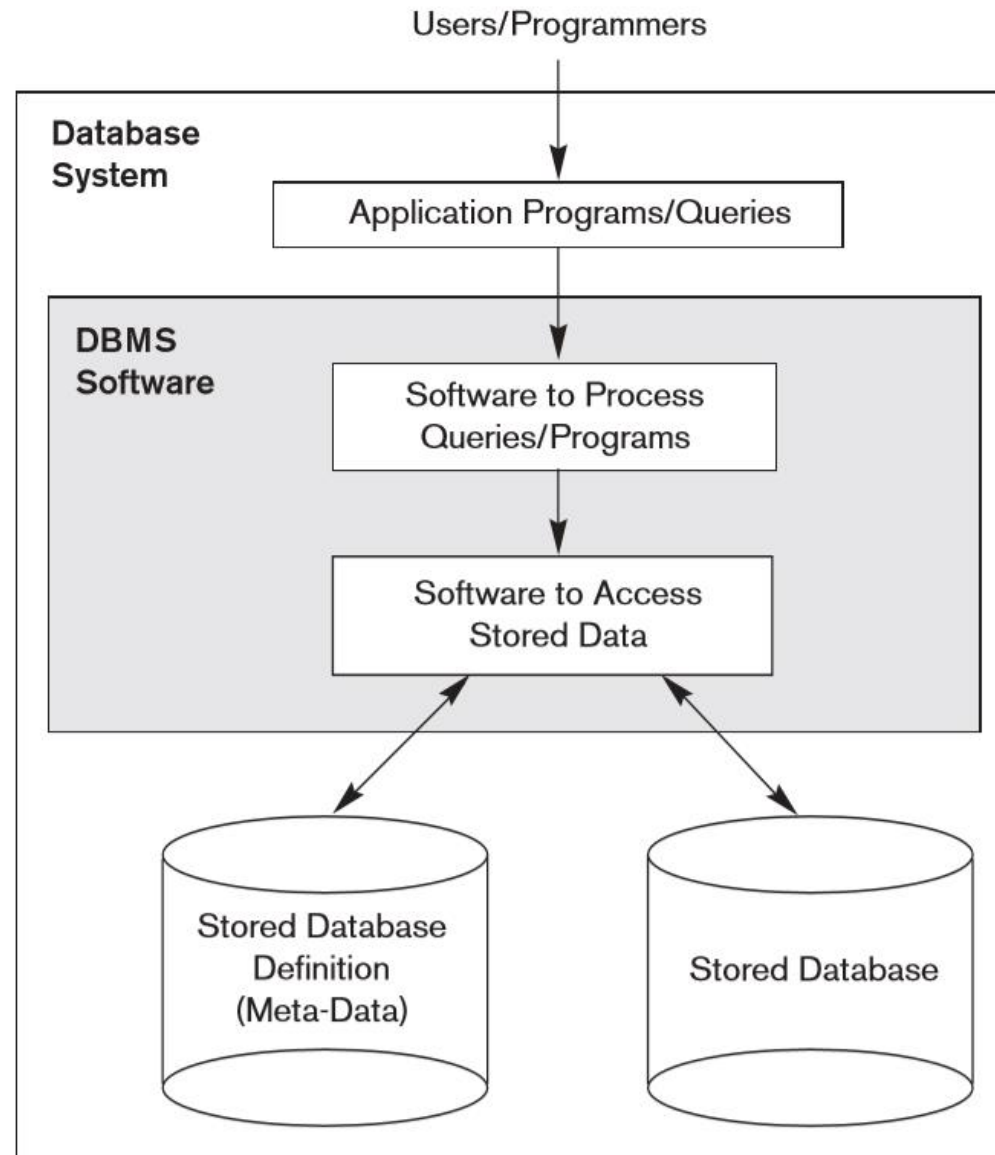
- A **database management system (DBMS)** is a collection of programs that enables users to create and maintain a database.
- The DBMS is a *general-purpose software system* that facilitates the processes of *defining, constructing, manipulating, and sharing* databases among various users and applications

- 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** 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 **maintain** the database system by allowing the system to evolve as requirements change over time.



- We will call the database and DBMS software together a **database system**.



An Example

**STUDENT**

| Name  | Student_number | Class | Major |
|-------|----------------|-------|-------|
| Smith | 17             | 1     | CS    |
| Brown | 8              | 2     | CS    |

**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     | 07   | King       |
| 92                 | CS1310        | Fall     | 07   | Anderson   |
| 102                | CS3320        | Spring   | 08   | Knuth      |
| 112                | MATH2410      | Fall     | 08   | Chang      |
| 119                | CS1310        | Fall     | 08   | Anderson   |
| 135                | CS3380        | Fall     | 08   | 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              |

# Characteristics of the Database Approach

# Self-Describing Nature of a Database System

- A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints.
- This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data.
- The information stored in the catalog is called **meta-data**.

- The catalog is used by the DBMS software and also by database users who need information about the database structure.
- In traditional file processing, data definition is typically part of the application programs themselves.

## 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        |

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

# Insulation between Programs and Data, and Data Abstraction

- The structure of data files is stored in the DBMS catalog separately from the access programs.
- We call this property **program-data independence**.



- In some types of database systems, such as object-oriented and object-relational systems, users can define operations on data as part of the database definitions.
- An **operation**(also called a *function* or *method*) is specified in two parts.
  - The *interface* (or *signature*) of an operation includes the operation name and the data types of its arguments (or parameters).
  - The *implementation* (or *method*) of the operation is specified separately and can be changed without affecting the interface.
- User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented.
- This may be termed **program-operation independence**.

- The characteristic that allows program-data independence and program-operation independence is called **data abstraction**.
- The **data model** *hides* storage and implementation details that are not of interest to most database users.

| Data Item Name | Starting Position in Record | Length in Characters (bytes) |
|----------------|-----------------------------|------------------------------|
| Name           | 1                           | 30                           |
| Student_number | 31                          | 4                            |
| Class          | 35                          | 1                            |
| Major          | 36                          | 4                            |

# Support of Multiple Views of the Data

- A database typically has many users, each of whom may require a different perspective or **view** of the database.
- A view may be a subset of the database or it may contain **virtual data** that is derived from the database files but is not explicitly stored.

## TRANSCRIPT

| Student_name | Student_transcript |       |          |      |            |
|--------------|--------------------|-------|----------|------|------------|
|              | Course_number      | Grade | Semester | Year | Section_id |
| Smith        | CS1310             | C     | Fall     | 08   | 119        |
|              | MATH2410           | B     | Fall     | 08   | 112        |
| Brown        | MATH2410           | A     | Fall     | 07   | 85         |
|              | CS1310             | A     | Fall     | 07   | 92         |
|              | CS3320             | B     | Spring   | 08   | 102        |
|              | CS3380             | A     | Fall     | 08   | 135        |

(a)

## COURSE\_PREREQUISITES

| Course_name     | Course_number | Prerequisites |
|-----------------|---------------|---------------|
| Database        | CS3380        | CS3320        |
|                 |               | MATH2410      |
| Data Structures | CS3320        | CS1310        |

(b)

# Sharing of Data and Multiuser Transaction Processing

- A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly and efficiently.
- These types of applications are generally called **online transaction processing (OLTP)** applications.
- The DBMS must include **concurrency control** software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.

- Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions.
- The DBMS must enforce several transaction properties.
- The **isolation** property ensures that each transaction appears to execute in isolation from other transactions, even though hundreds of transactions may be executing concurrently.
- The **atomicity** property ensures that either all the database operations in a transaction are executed or none are.