

01 INTRODUCTION

CS 340: INTRODUCTION TO DATABASE SYSTEMS

Adapted from: Dr. Omar Alomeir 2023

Course instructor:
Dr. Maram Alajlan.

LEARNING GOALS

CLO1: Describe basic database concepts, architecture, and DBMS principles. Demonstrate knowledge of emerging topics in databases. (Knowledge & Understanding)

Chapter objectives:

- Define a database, and database management system
- Explain why we need databases

DEFINITIONS

Database: An organized collection of related data, stored on disk, sometimes accessed by multiple people.

Examples: University database contains data on courses, students, faculty, and other resources

A movies database contains data on movies, genres, actors, and directors

A database management system manages databases. It allows: accessing, modifying, backing up and recovering the database.

Examples: Oracle, PostgreSQL, MySQL, and MongoDB

WHY USE A DATABASE MANAGEMENT SYSTEM

- Databases store all the data that a system would need.
- They do so reliably, in a way that is secure, can be shared, and with efficiency.
- However, they need to be designed carefully.
- Why “design” a database and use a management system? Why not just throw my data into a file and be done with it?

FILE BASED SYSTEM

- Prior to DBMS, file system provided by OS was used to store information.
- In a file-based system, we have collection of application programs that perform services for the end users. Each program defines and manages its own data.
- How do the following things work in this system:
  - Security (unauthorized access).
  - Concurrent access (multiple people at once).
  - Recovery (Hard drive failure, power goes off, etc).
  - Performance (Can you guarantee access at certain speeds? Optimized look up?).
  - Redundancy (Data about a student and their courses, repeat the student name?).

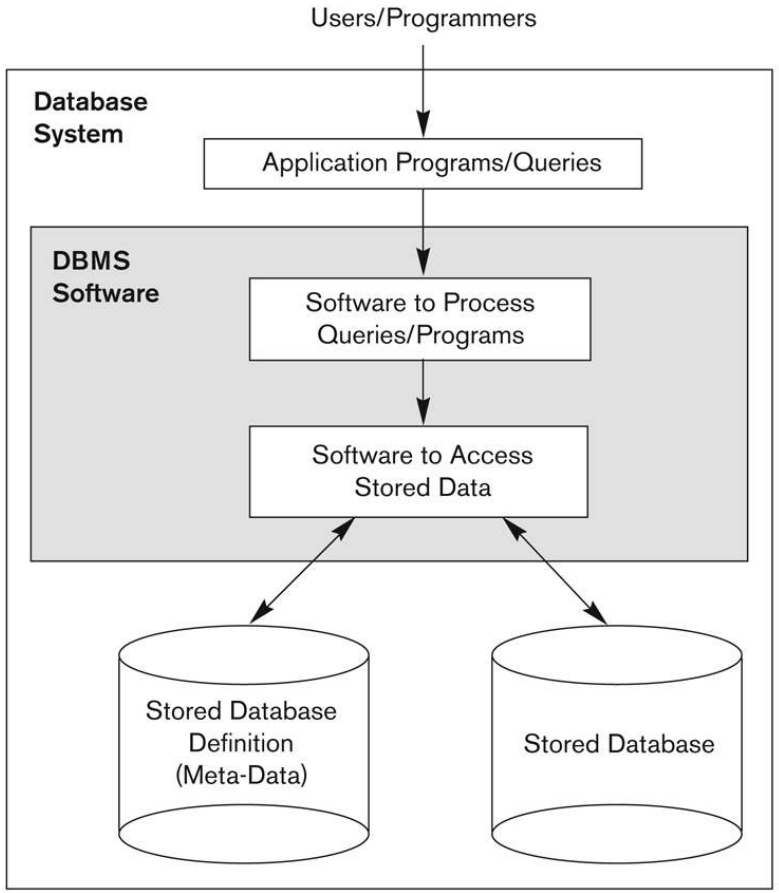
DBMS INSTEAD OF FILE SYSTEM

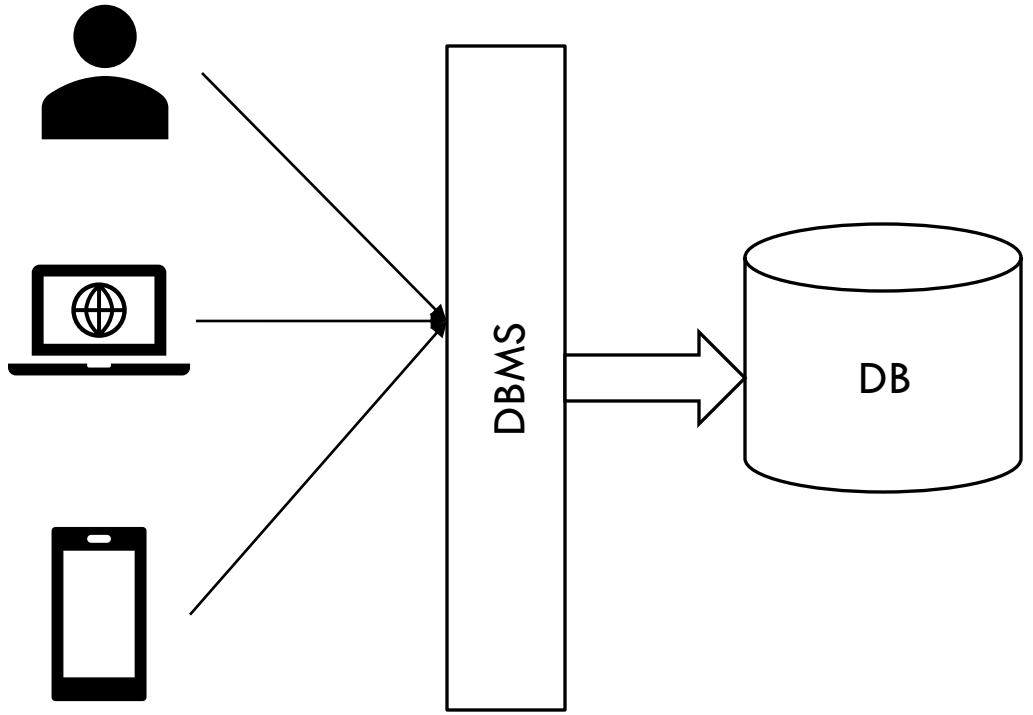
A database management system would do all of that for you, and more.

A database is the data, the database management system is what manages that data.

Data is entities (student, course) and relationships (Student takes course)

SIMPLIFIED DBMS





OTHER USEFUL FUNCTIONS OF A DBMS

- **Define a database**: Structure, data types, and constraints of a database.
- **Manipulate the database**:
  - Retrieve (query) data
  - Modify: Insert, update, or delete data
  - Manage access through applications
- Handle concurrent access by multiple users
- Handle security measures
- Handle maintenance of the database over its lifetime
- Applications issue two things to the database:
  - Queries: requests that generate a result based on the stored data
  - Transactions: Collections of requests

OBJECTIVES OF DBMS

- **Data Availability:** Data are made available to wide variety of users in a meaningful format at reasonable cost so that the users can easily access the data.
- **Data Integrity:** Data available in the database is a reliable/correct data.
- **Data Security:** Only authorized users can access the data. Data security can be enforced by passwords.
- **Data Independence:** DBMS allows the user to store, update, and retrieve data in an efficient manner. It hides certain details of how the data are stored and maintained.

MAIN CHARACTERISTICS OF THE DATABASE APPROACH

- **Self-describing nature of a database system:**
  - A database system stores not only the database but also a complete definition or description of a particular database known as System Catalog (e.g. data structures, types, storage format of each data and constraints on the data).
  - The description stored in the System Catalog is called **meta-data***.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - The structure of data files is stored in the DBMS catalog separately from the access programs. DBMS allows changing data structures and storage organization without having to change the DBMS access programs.

HOW DO I DESIGN AND IMPLEMENT A DATABASE?

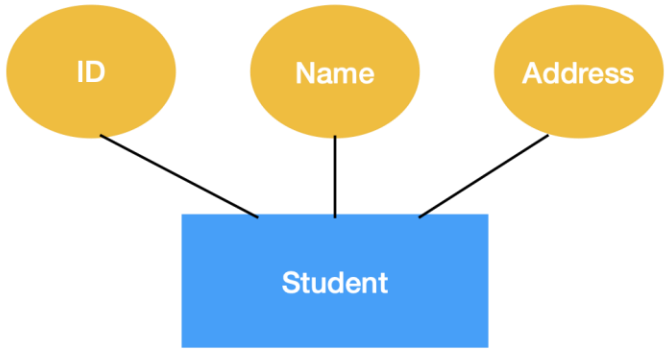
There are multiple steps and they align with the contents of this course:

1. Conceptual modeling
2. Logical modeling
3. Optimize the design
4. Decide how to query
5. Other database applications (Data warehouse, OLAP(Online analytical processing), Data Mining)
6. Create your own application

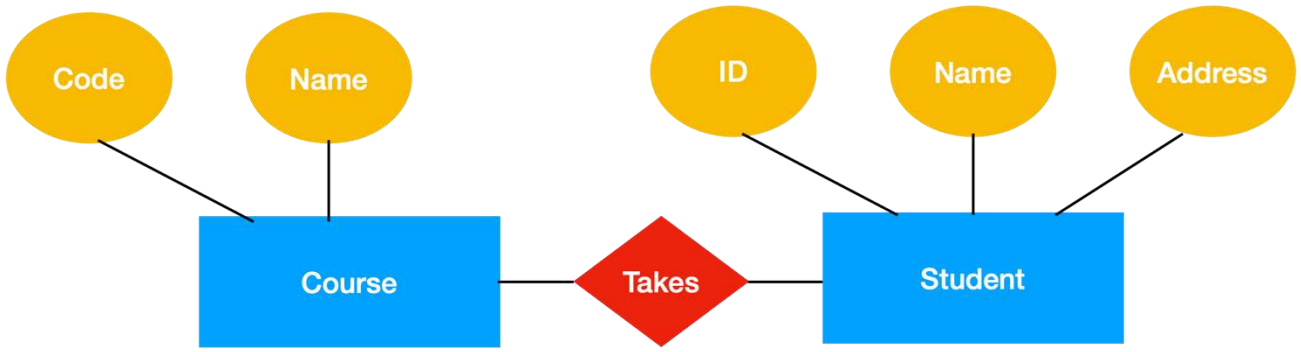
1. CONCEPTUAL MODELING

Entity relationship diagrams:

Entity: A real world object, a noun. Described by attributes.



A relationship: An association between two or more entities. A verb.



2. LOGICAL MODELING

- Data model: The tools the describe: Data, relationships, and constraints
- We use the relational model
- The most used model in the world today

Example: Student data

Student	ID	Major
Ahmad	123	CS
Ali	124	SE

Logical view is separate from physical view of data (how data is stored)

Applications refer to logical model tables rather than where data is stored

3. OPTIMIZATION

We need to carefully design our databases so that:

- Capture all the information we need
- Take up less space
- It is easy to maintain

We will look mainly at normalization, and different “Normal forms”

4. QUERYING THE DATA

We will use **Structured Query Language** (SQL) to query the data.

Example:

Plain English: Find all students who are majoring in CS

SQL:

```
SELECT s.name
FROM Student s
WHERE s.major = "CS"
```

SQL is a declarative language

> Declarative language means statements do not describe computations directly, but instead describe the desired result of some computation

We will also look at relational algebra, the original language for SQL

4. QUERYING THE DATA (CONTD)

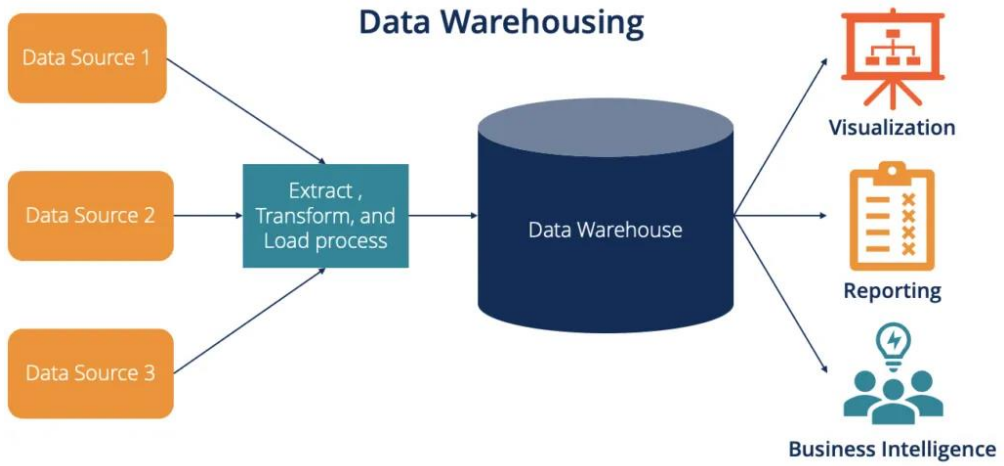
Users of the database can be divided into:

- Database administrator (DBA): Handle authorizing access for users, monitor db use, control and manage efficiency of database.
- Database designers: Define the content, structure, and constraints.
- End-users: Can be application developers or application users. They issue queries, ask for reports, they can even update the content of the db.

5. OTHER DATABASE APPLICATIONS

Data Warehousing, online analytical processing (OLAP)

- Organizations collect and organize current and historical data to identify patterns.
- Emphasis on analysis of complex queries on large datasets.



5. OTHER DATABASE APPLICATIONS (CONTD)

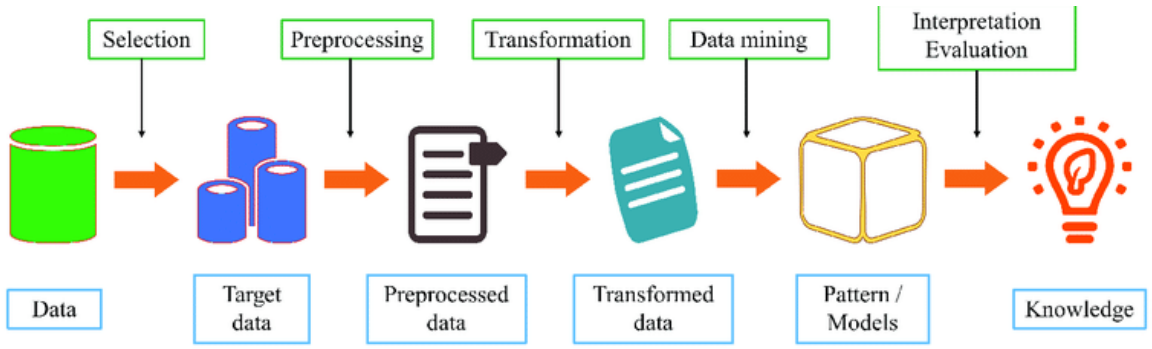
Online transaction processing (OLTP)

- Transactions are groups or collection of requests to be executed as a single unit
- DBMS guarantees the ACID property:
  - A = Atomicity, every transaction is a single unit
  - C = Consistency, the state before and after a transaction is consistent (Data integrity constrains must be followed - Data is valid before and after, e.g. the amount can not be negative)
  - I = Isolation, Transactions executed concurrently don't affect each other
  - D = Durability, Once committed a transaction is committed unaffected by failure (power off, crash, etc)

5. OTHER DATABASE APPLICATIONS (CONTD)

Data mining and knowledge discovery

- Exploration and analysis of large data to find knowledge in the form of patterns in data.
- The challenge is related to the fields of: statistics, databases, machine learning, optimization, among others.

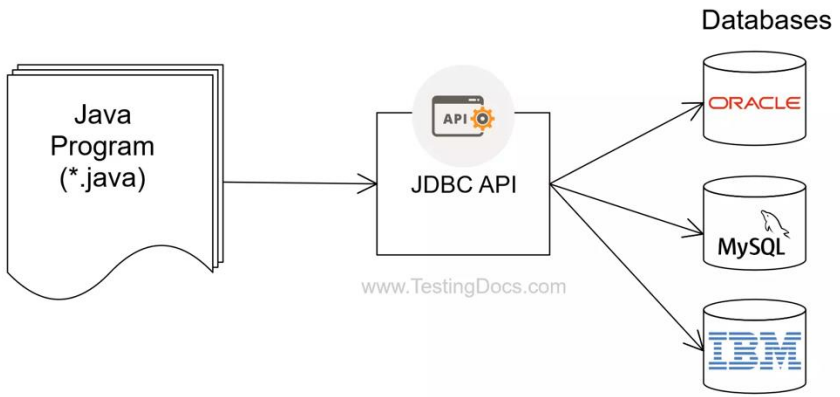


6. CREATE YOUR OWN APPLICATION

Your data is completely up to you.

We will focus on using Java and Java Database Connectivity (JDBC). It is up to you to use something different (Python, NodeJS, etc).

We will also aim to create a full application with a GUI.



MAIN CHARACTERISTICS OF THE DATABASE APPROACH

- **Self-describing nature of a database system:**
  - A database system stores not only the database but also a complete definition or description of a particular database known as System Catalog (e.g. data structures, types, storage format of each data and constraints on the data).
  - The description stored in the System Catalog is called **meta-data***.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - The structure of data files is stored in the DBMS catalog separately from the access programs. DBMS allows changing data structures and storage organization without having to change the DBMS access programs.

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

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

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

EXAMPLE OF A SIMPLIFIED DATABASE CATALOG

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

Figure 1.4
Internal storage format for a STUDENT record, based on the database catalog in Figure 1.3.

MAIN CHARACTERISTICS OF THE DATABASE APPROACH (CONTINUED)

- **Data Abstraction:**
  - The characteristic that allows program-data independence is called data abstraction.
  - A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented. A **data model** is a type of data abstraction that 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

MAIN CHARACTERISTICS OF THE DATABASE APPROACH (CONTINUED)

- **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.

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)

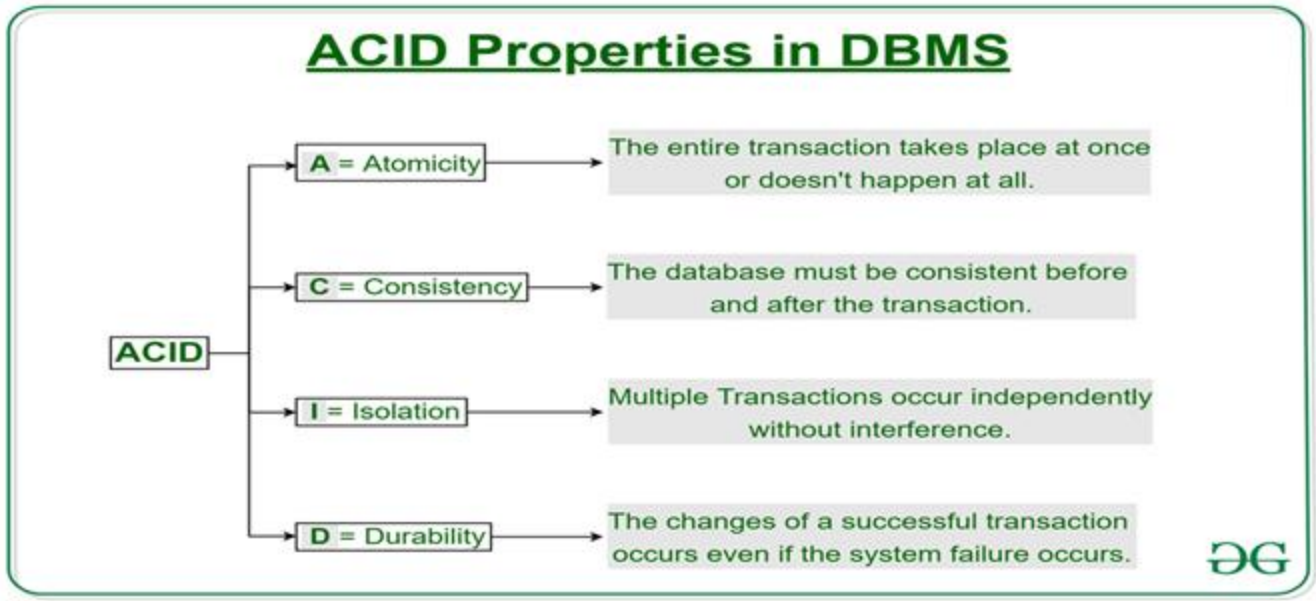
Figure 1.5
Two views derived from the database in Figure 1.2. (a) The TRANSCRIPT view. (b) The COURSE_PREREQUISITES view.

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.

MAIN CHARACTERISTICS OF THE DATABASE APPROACH (CONTINUED)

- Provides security and removes redundancy.
- Any DBMS is able to support the ACID concept (Atomicity, Consistency, Isolation, and Durability). It is made sure in every DBMS that the real purpose of data should not be lost while performing transactions like delete, insert and update.



DO I ALWAYS NEED A DATABASE?

Short answer: No.

Why? Designing a DBMS is a long process that incurs a lot of overhead.

When not to use a database:

- If your data is static, never changing.
- If you're working with a lot of hardware limitations (embedded systems).
- No multiple user access.

SUMMARY

In this chapter we looked at the database as a concept as well as the database management system and how they interact.

We explored the benefits of DBMS.

We also looked briefly at the process of designing a database application.

We will continue to look at all of those steps in detail starting next lecture with conceptual modeling, insha' Allah.