

# Book Chapter 1 - Databases and Database Users

Intro Traditional database systems: most data is textual or numeric  
 Bigdata, NoSQL  $\Rightarrow$  New databases for large storage applications  
 Multimedia databases, geographic information systems (GIS)

Database: collection of related data; by data, we mean known facts that can be recorded and have implicit meaning

Implicit properties of a database: (1) represents some aspect of the real world, sometimes called the miniworld or universe of discourse (UoD). (2) Logically coherent collection of data with some meaning. (3) Designed, built, and populated with data for a specific purpose

$\rightarrow$  Can be any size or complexity, may be generated and maintained manually or by a computer.

Database Management System (DBMS): Computerized system that enables users to create and maintain a database.

$\rightarrow$  Software that enables defining, constructing, manipulating, and sharing databases among users.

Metadata: Place where database definition or descriptive information is stored

$\rightarrow$  An application accesses database by sending queries. Queries are statements that retrieve data.

$\rightarrow$  To define a database, we must specify the structure of the records by specifying the different types of data elements.

Characteristics of the Database Approach

$\rightarrow$  Database approach differs from the older approach of simply storing information in files

$\rightarrow$  In traditional file processing, each user defines and implements the files needed for a specific software application

Summary: Databases are collections of related data that represent parts of the real world. Databases have some specific purpose, and are defined, constructed, manipulated and shared.

## Book Chapter 1 - Databases and Database Users

- Files are generally separate because the programs are separate; this leads to redundancy and wasted storage space
- In the database approach, a single repo maintains data that is defined once and then accessed by various users and programs

\* Database approach has following benefits: Self-describing nature, insulation between programs and data, supports multiple views  
 Self-describing nature: DB approach means not only database itself, but also structure and constraints

- Stored in a catalog, typically called metadata
- NoSQL does not require metadata; data is self-describing
- File-processing software can only access specific databases; DBMS software can access diverse databases

Insulation between programs / data / data abstraction: structure of data files is embedded in the application programs, so changing file structure requires changing all programs. However, DBMS programs do not require changes since structure of data files is stored in the catalog separate from the access programs.

\* This is program - data independence.

- Users can define operations (functions or methods)
  - ↳ Interface: Includes operation name and the data types of its arguments
  - ↳ Implementation: Actual function contents

\* Program-operation independence: Applications can work with data through operations regardless of how they are implemented.

Summary: Databases are better than file based approaches because they offer a separation between the data itself and the structure of the data, often stored in metadata. DBs also offer program-data and program-operation independence.

# Book Chapter 1 - Databases and Database Users

- \* Data abstraction: Allows for program-data and program-operation independence  
→ DBMS hides/abstracts details of how things work from the user.
- \* Data model: Type of conceptual representation of data.  
Application => conceptual representation => actual data on disk
- \* Views: Subset of database, or virtual data derived from the actual tables, but not explicitly stored.  
→ Multi-user databases must implement concurrency control to ensure multiple users working with data do not conflict
- \* Transaction: executing program or process that includes DB access.
- \* Isolation: Transactions occur in isolation from other transactions
- \* Atomicity: Either all operations in a transaction happen, or none do.

## Database actors

- DBA: responsible for authorizing access
- Designers: responsible for identifying the data to be stored, and choosing appropriate structures to represent and store the data.
- End users: people for whom the DB is intended. Examples: Casual end users, Naive/parametric end users, sophisticated end users, standalone users.

## Advantages of using DBMS approach

- \* → Controlling redundancy: data normalization ensures consistency and saves storage space.
- Denormalization: when data is placed together for performance
- DBMS should provide security and authorization subsystem.

## Providing Persistent Storage

Persistent data survives the termination of the program or object for later retrieval.

Summary: DBMS systems provide abstractions of data through data abstractions and data models so that the database actors don't have to worry about how the physical data is stored. DBMS systems offer data normalization and persistence.