Databases and DBMS

In general...

Database - collection of related data with implicit meaning.

Data - known facts about the real world.

More specific...

Implict properties of database

- designed for a specific purpose, specific users, and specific applications in mind to utilize to access that database.
- Logically conherent collection of data with an inherent meaning.
- Represents an aspect of the real world (miniworld) universe of disourse.

Database Management System (DBMS)

- Example: MYSQL, MSSQL
- Definition: Computerized system to help us create and maintain a database.

4 main functions (DCMS)

- Defining - data types, structures, constraints (stored as metadata) in the database catalog.
- Constructing - storing the data on some storage medium.
- Manipulating - querying data, performing updates, etc...
- Sharing - multiple users or programs access the data at the same time.

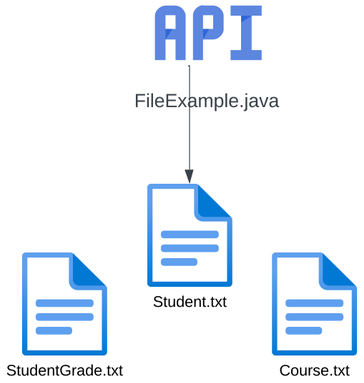
Additional functions...

Protecting- specify security policies (RBAC) via. security modules (supplied by the DBMS).

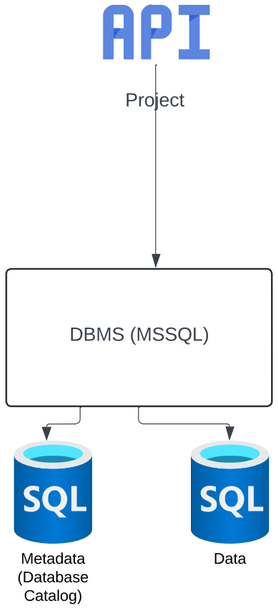
Maintaining - helping with the evolution of our databases. Can include both software and hardware.

DBMS vs. File Approach

FIle Approach



DBMS Approach



Redundancy

- Leads to inconsistencies.
- Duplication of effort.
- Duplicating storage space.

Supports Transaction(s)

- Isolation - it appears as the end user as if I'm the only one accessing the data.
- Atomicity - either all operations are successful or none are.

Database Design Process

High Level

- Requirements specification and analysis.
- Conceptual Design (ER Model).
- Logical Design (Relational Schema).
- Physical Design.

Low Level

Abstraction (*Data Abstraction*)

- Program-data independence.
- Program-operation independence.

Database Users

Actors on the scene

Database Administrators (Primary database, DBMS)

- Authorizing Access, Coordinating and Monitoring it's use.
- Assisted by a larger team.

Database Designers (You!)

- Identify the data needing to be stored.
- Choosing the appropriate way to represent and store the data.

Need to ensure the designed database fulfills the requirements of the *end users*.

End users

- querying, updating, etc.. the database.
- Casual - occasionally accessing the database (not necessarily any requirements around what they need). Know how to access the data via GUI or SQL.
- Naive - known access to various fields in your database controlled via canned transactions (Stored Procedures).
- .... (to be continued next time)