DBMS

Dr. Media A. Ibrahim
Assist.Lec: Areen Hamad
Software Engineering
ISE Department
Media.ibrahim@epu.edu.iq
Areen.hamad@epu.edu.iq

2024-2025

References

- **Textbooks:**
  - Database Systems Concepts, Silberschatz, Korth, and Sudarshan
  - A First Course in Database Systems, Ullman and Widom
  - An Introduction to Database Systems Eighth Edition by C. J. Date
  - Database management systems second edition by raghu Ramakrishnan and johnnes Gehrke.

- **Software:**
  - Database implementation in Oracle 10g

Introduction to DBMS

Outlines

- Data vs. Information
- What is a Database System?
- Types of Databases
- Three-Levels of Abstraction in a Database System
- What Is a DBMS?
- Architecture of DBMS
- Components of a DBMS
- Functions of a DBMS
- Advantages of DBMS
- Disadvantages of DBMS
- Conclusion

Data vs. Information

Data

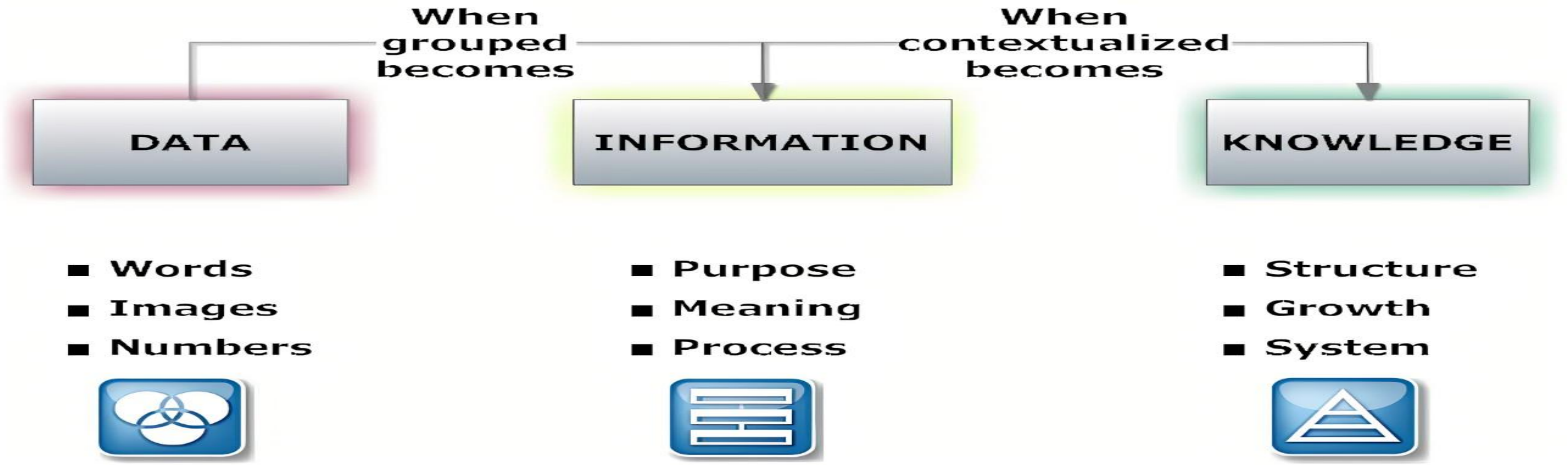
- raw facts
- no context
- just numbers and text

Information

- data with context
- processed data
- value-added to data
  - summarized
  - organized
  - analyzed

DATA/INFORMATION PROCESSING

The process of converting the data (raw facts) into meaningful information is called as data/information processing.



Note: In business processing knowledge is more useful to make decisions for any organization

Why Study Databases?

Databases are useful

- Many computing applications deal with large amounts of information
- Database systems give a set of tools for storing, searching and managing this information

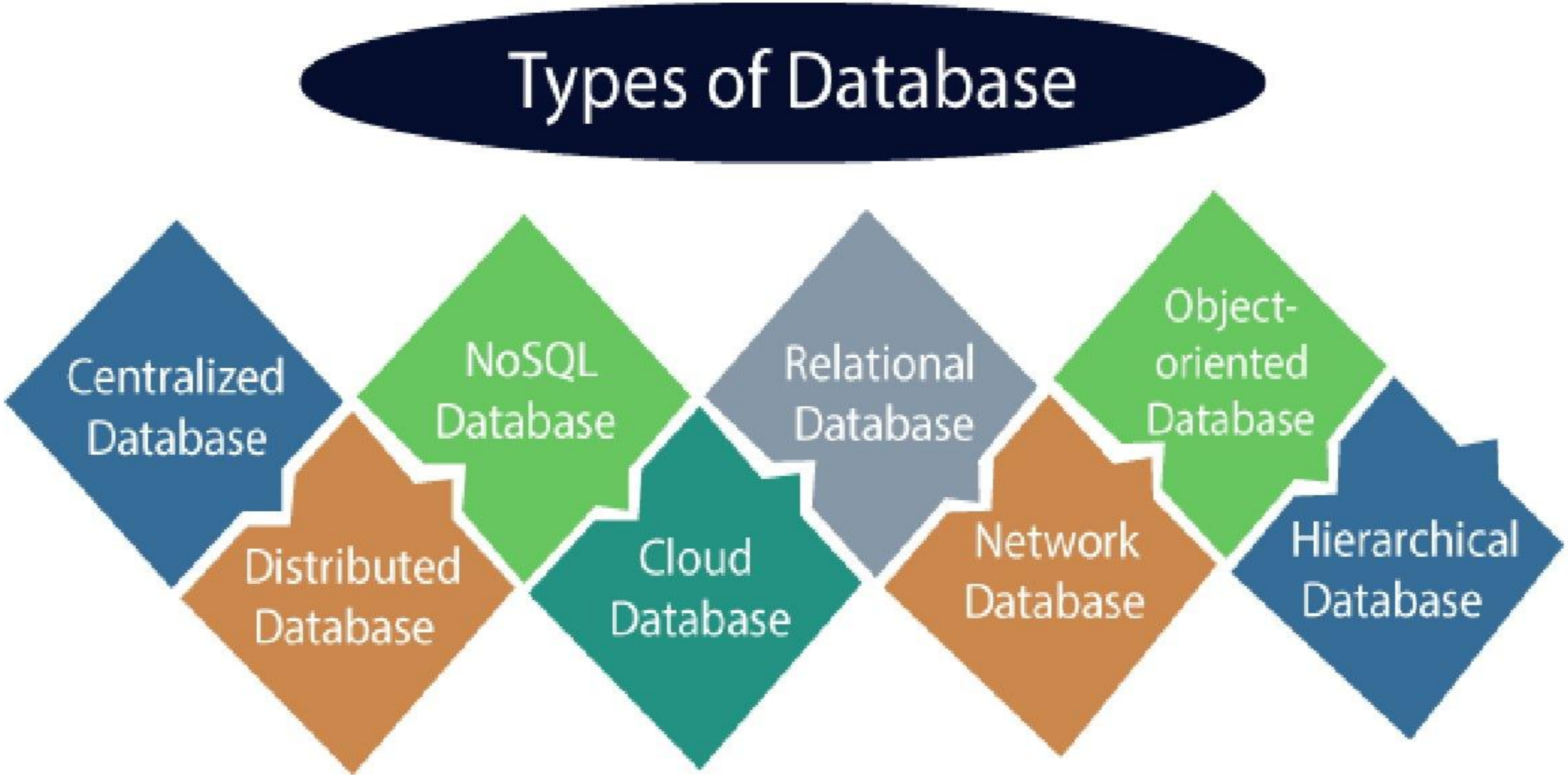
Databases in CS

- Databases are a ‘core topic’ in computer science
- Basic concepts and skills with database systems are part of the skill set you will be assumed to have as a CS graduate

What is a Database System

- A **database** is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, **databases** can be classified according to types of content: bibliographic, full-text, numeric, and images.

Types of Databases



Real-World Example

- Google BigQuery: A cloud-based data warehouse built on a relational DBMS for large-scale analytics.

- MySQL: Widely used open-source relational database for web applications.

- Netflix uses a DBMS to manage user profiles, subscriptions, and viewing histories, ensuring consistent access across devices.

1) Centralized Database

- It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. These applications contain the authentication process to let users access data securely. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

Advantages of Centralized Database

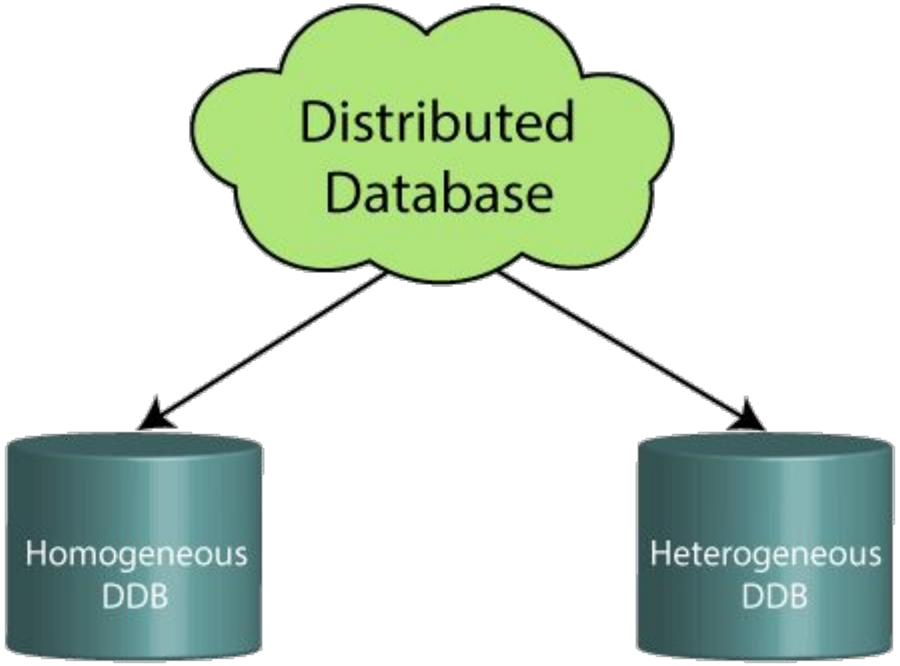
- It has decreased the risk of data management, i.e., manipulation of data will not affect the core data.
- Data consistency is maintained as it manages data in a central repository.
- It provides better data quality, which enables organizations to establish data standards.
- It is less costly because fewer vendors are required to handle the data sets.

Disadvantages of Centralized Database

- The size of the centralized database is large, which increases the response time for fetching the data.
- It is not easy to update such an extensive database system.
- If any server failure occurs, entire data will be lost, which could be a huge loss.

2) Distributed Database

- Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily. **Examples** of the Distributed database are Apache Cassandra, HBase, Ignite, etc



Distributed Database(cont.)

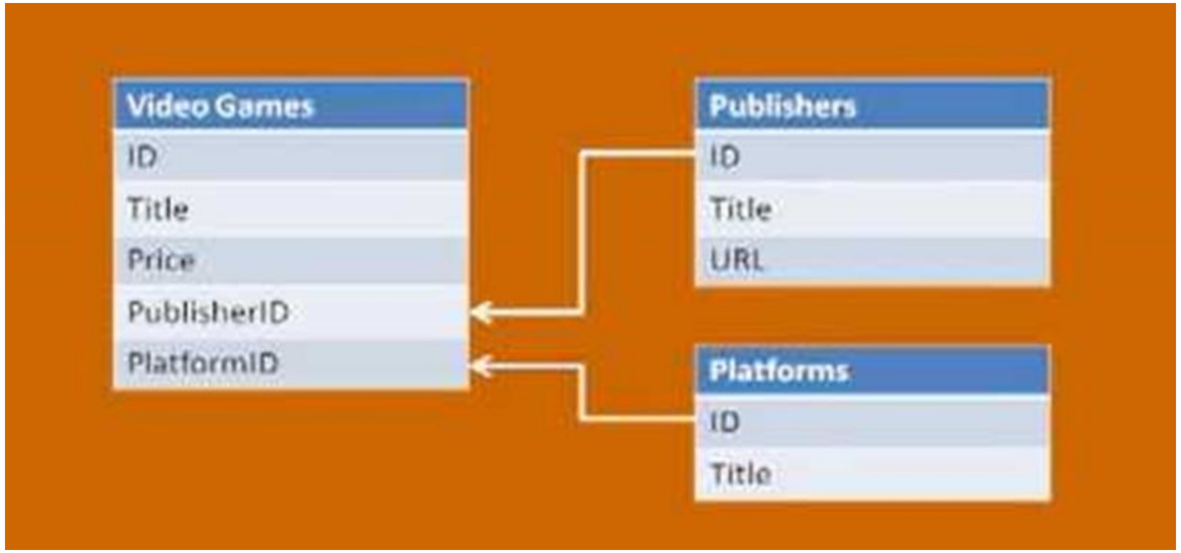
- **Homogeneous DDB:** Those database systems which execute on the same operating system and use the same application process and carry the same hardware devices.
- **Heterogeneous DDB**: Those database systems which execute on different operating systems under different application procedures, and carries different hardware devices

Advantages of Distributed Database

- Modular development is possible in a distributed database, i.e., the system can be expanded by including new computers and connecting them to the distributed system.
- One server failure will not affect the entire data set.

3) Relational database system

- This is the most common of all the different types of databases.
- The data in a relational database is stored in various data tables
- These databases are extensively used in various
- **PayPal** uses a relational DBMS to manage transactions securely and efficiently.



Properties of Relational Database

- There are following four commonly known properties of a relational model known as **ACID** properties, where:
- **A means Atomicity:** This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.
- **C means Consistency:** If we perform any operation over the data, its value before and after the operation should be preserved. For example, the account balance before and after the transaction should be correct, i.e., it should remain conserved.
- **I means Isolation:** There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated. For example, when multiple transactions occur at the same time, one transaction effects should not be visible to the other transactions in the database.
- **D means Durability:** It ensures that once it completes the operation and commits the data, data changes should remain permanent
- **Amazon DynamoDB** (NoSQL): Used for fast, scalable data management for e-commerce.

4) NoSQL Database

- Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways. It came into existence when the demand for building modern applications increased. Thus, NoSQL presented a wide variety of database technologies in response to the demands. We can further divide a NoSQL database into the following four types:

1. **Key-value storage**: It is the simplest type of database storage where it stores every single item as a key (or attribute name) holding its value, together.
2. **Document-oriented Database**: A type of database used to store data as JSON-like document. It helps developers in storing data by using the same document-model format as used in the application code.
3. **Graph Databases**: It is used for storing vast amounts of data in a graph-like structure. Most commonly, social networking websites use the graph database.
4. **Wide-column stores**: It is similar to the data represented in relational databases. Here, data is stored in large columns together, instead of storing in rows.

5) Cloud Database

- A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database. There are numerous cloud platforms, but the best options are:Amazon Web Services(AWS)
- Microsoft Azure
- Kamatera
- PhonixNAP
- ScienceSoft
- Google Cloud SQL, etc

6)Hierarchical Databases

- It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure.

7) Network Databases

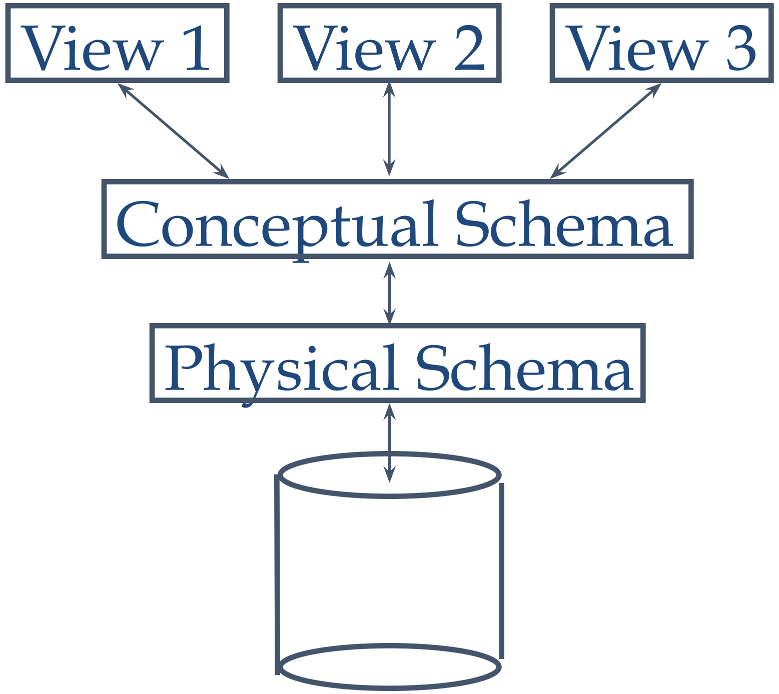
- It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them. Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure

Three-Levels of Abstraction in a Database System

- *views:*
  - Highest level.
  - Views describe how users see the data.
  - Simplifies the interaction with the users.
  - Provide multiple Users.

- *Single conceptual (logical) schema*
  - Intermediate level.
  - Conceptual schema defines logical structure.
  - What data is stored.

- *Single physical schema:*
  - Lowest Level.
  - Physical schema describes the files and indexes used.



What Is a DBMS?

- A **database management system (DBMS)** is **system** software for creating and **managing databases**. The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and **manage** data.
- stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.
  - **RDBMS** stands for *Relational Database Management System.*
  - All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS.
  - It is called Relational Database Management System (RDBMS) because it is based on the relational model

Difference between DBMS and RDBMS

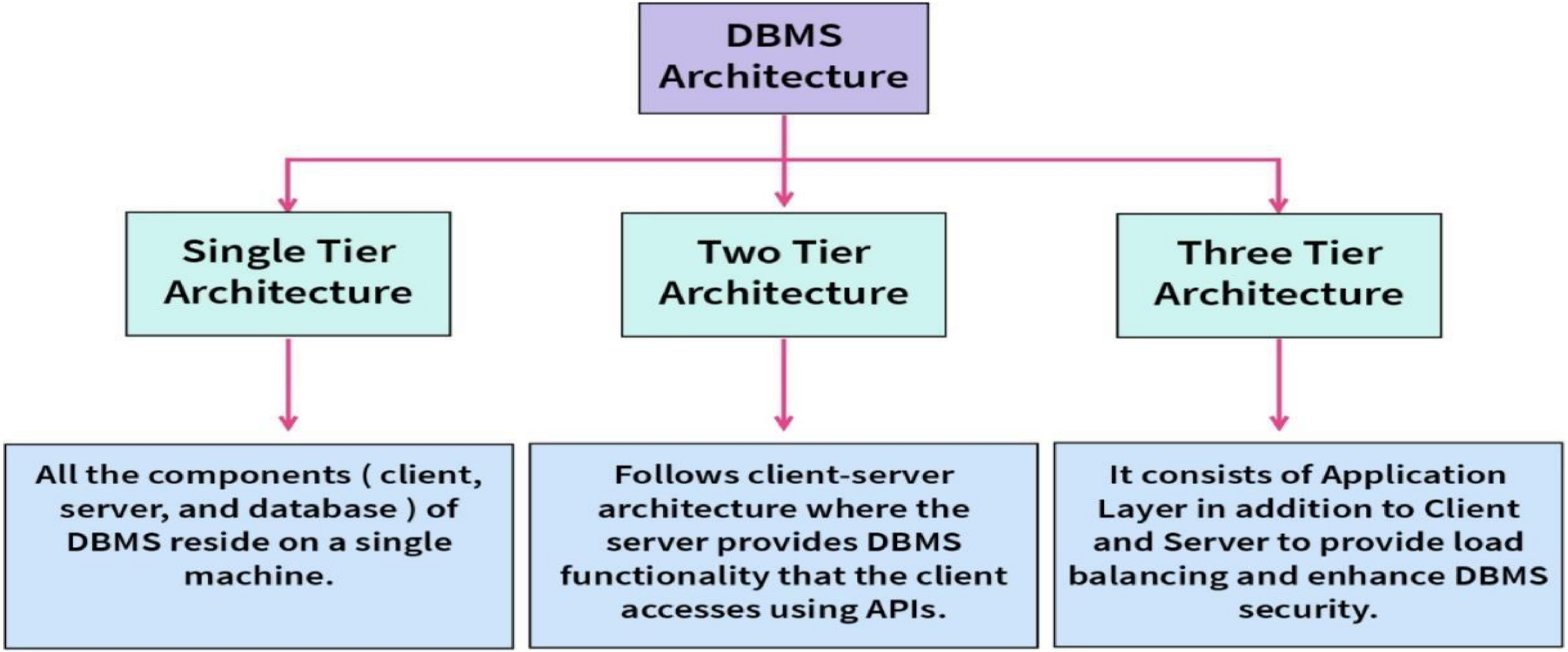
No	DBMS	RDBMS
1)	DBMS applications store **data as file**.	RDBMS applications store **data in a tabular form**.
2)	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables
3)	**Normalization is not** present in DBMS.	**Normalization is** present in RDBMS.
4)	DBMS does **not apply any security** with regards to data manipulation.	RDBMS **defines the integrity constraint** for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
5)	DBMS uses file system to store data, so there will be **no relation between the tables**.	in RDBMS, data values are stored in the form of tables, so a **relationship** between these data values will be stored in the form of a table as well.
6)	DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
7)	DBMS **does not support distributed database**.	RDBMS **supports distributed database**.
8)	DBMS is meant to be for small organization and **deal with small data**. it supports **single user**.	RDBMS is designed to **handle large amount of data**. it supports **multiple users**
9)	Examples of DBMS are file systems, **xml** etc.	Example of RDBMS are **mysql**, **postgre**, **sql server**, **oracle** etc.

- After observing the differences between DBMS and RDBMS, you can say that RDBMS is an extension of DBMS. There are many software products in the market today who are compatible for both DBMS and RDBMS. Means today a RDBMS application is DBMS application and vice-versa

Architecture of a DBMS

- ❖ The DBMS architecture affects the performance of the database as it helps to design, develop, implement, and maintain the database management system. The database management system design depends on the DBMS architecture for its representation. It describes the structure and the way in which the users are connected to a specific database system.
- ❖ The different types of DBMS architectures along with their real-life examples.
- ❖ The most common DBMS architecture:
  - Single Tier Architecture (One-Tier Architecture)
  - Two-Tier Architecture
  - Three-Tier Architecture

Types of DBMS Architecture



DBMS Architecture

Single Tier Architecture

Two Tier Architecture

Three Tier Architecture

All the components (client, server, and database) of DBMS reside on a single machine.

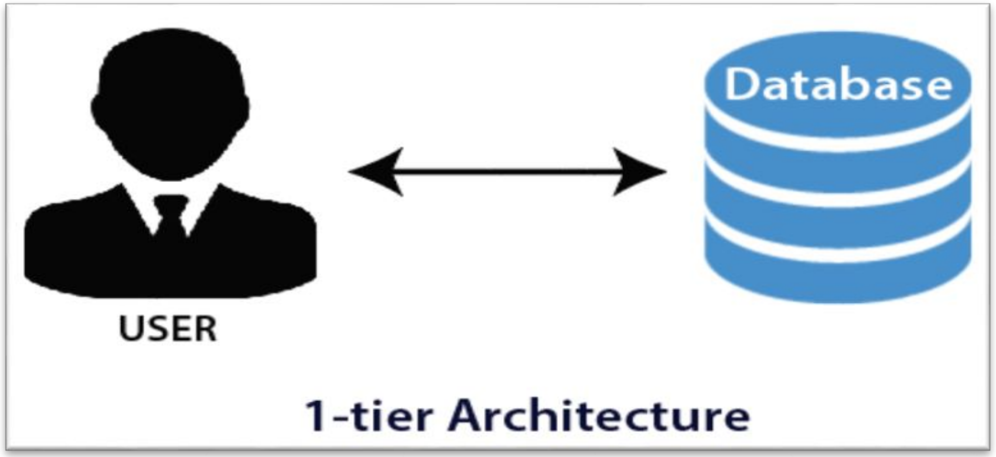
Follows client-server architecture where the server provides DBMS functionality that the client accesses using APIs.

It consists of Application Layer in addition to Client and Server to provide load balancing and enhance DBMS security.

(SaaS, PaaS, IaaS,

Single Tier Architecture

- Single Tier DBMS Architecture is the most straightforward DBMS architecture. All the DBMS components reside on a single server or platform, i.e., the database is directly accessible by the end-user. Because of this direct connection, the DBMS provides a rapid response, due to which programmers widely use this architecture to enhance the local application.
- In this structure, any modifications done by the client are reflected directly in the database, and all the processing is done on a single server. Also, no network connection is required to perform actions on the database. This database management system is also known as the local database system.



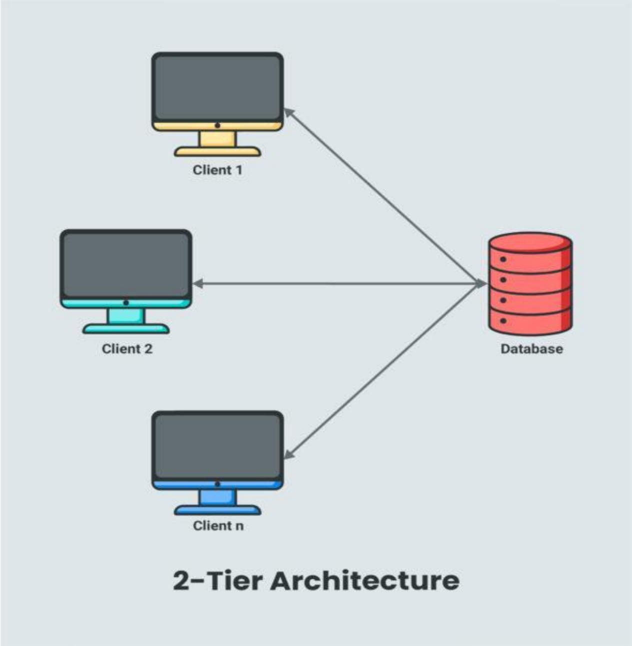
Single Tier DBMS Architecture is used whenever:

- The data isn't changed frequently.
- No multiple users are accessing the database system.
- We need a direct and simple way to modify or access the database for application development.

2. Two Tier Architecture

- Two-Tier DBMS Architecture is similar to the fundamental concept of client-server architecture.
- In a two-tier structure, the server provides the database functionality and it allows the clients to perform operations on the database through a direct internet connection via APIs
- (Application Programming Interface), for example:
  - ODBC: Open Database Connectivity.
  - JDBC: Java Database Connectivity.
- The Two-Tier DBMS Architecture is used when we wish to access the DBMS with the help of an application.

Client-side applications can access the database server directly with the help of API calls, making the application independent of the database in terms of operation, design, and programming.



The main advantages of having a two-tier architecture over a single tier are:

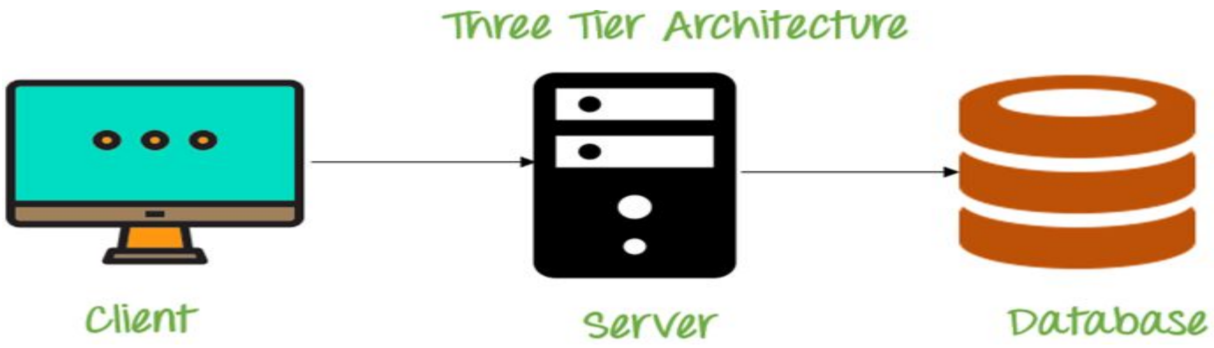
- Multiple users can use it at the same time. Hence, it can be used in an organization.
- It has high processing ability as the database functionality is handled by the server alone.
- Faster access to the database due to the direct connection and improved performance.
- Because of the two independent layers, it's easier to maintain.

The main disadvantages of Two-Tier DBMS Architecture are:

- Scalability - As the number of clients increases, the load on the server increases. Thereby declining the performance of the DBMS and, in turn, the client-side application.
- Security - The Direct connection between the client and server systems makes this architecture vulnerable to attacks

3. Three Tier Architecture

- Three-Tier DBMS Architecture is the most widely used DBMS architecture in which another layer known as Intermediate or Application layer is added between the server (Database Layer) and the client (Presentation Layer) systems to reduce the query processing load of the server. This application or intermediate layer provides the end-user an abstract view of the database.
- Since there is no direct connection between the client and the Server, all the user requests are handled by the Application Layer, i.e., the requests sent by the users are checked and verified by the Intermediate Layer before transferring them to the server.



The main advantages of Three Tier DBMS Architecture are:

- Scalability - Since the database server isn't aware of any users beyond the application layer and the application layer implements load balancing, there can be as many clients as you want.
- Data Integrity - Data corruption and bad requests can be avoided because of the checks performed in the application layer on each client reques.
- Security - The removal of the direct connection between the client and server systems via abstraction reduces unauthorized access to the database

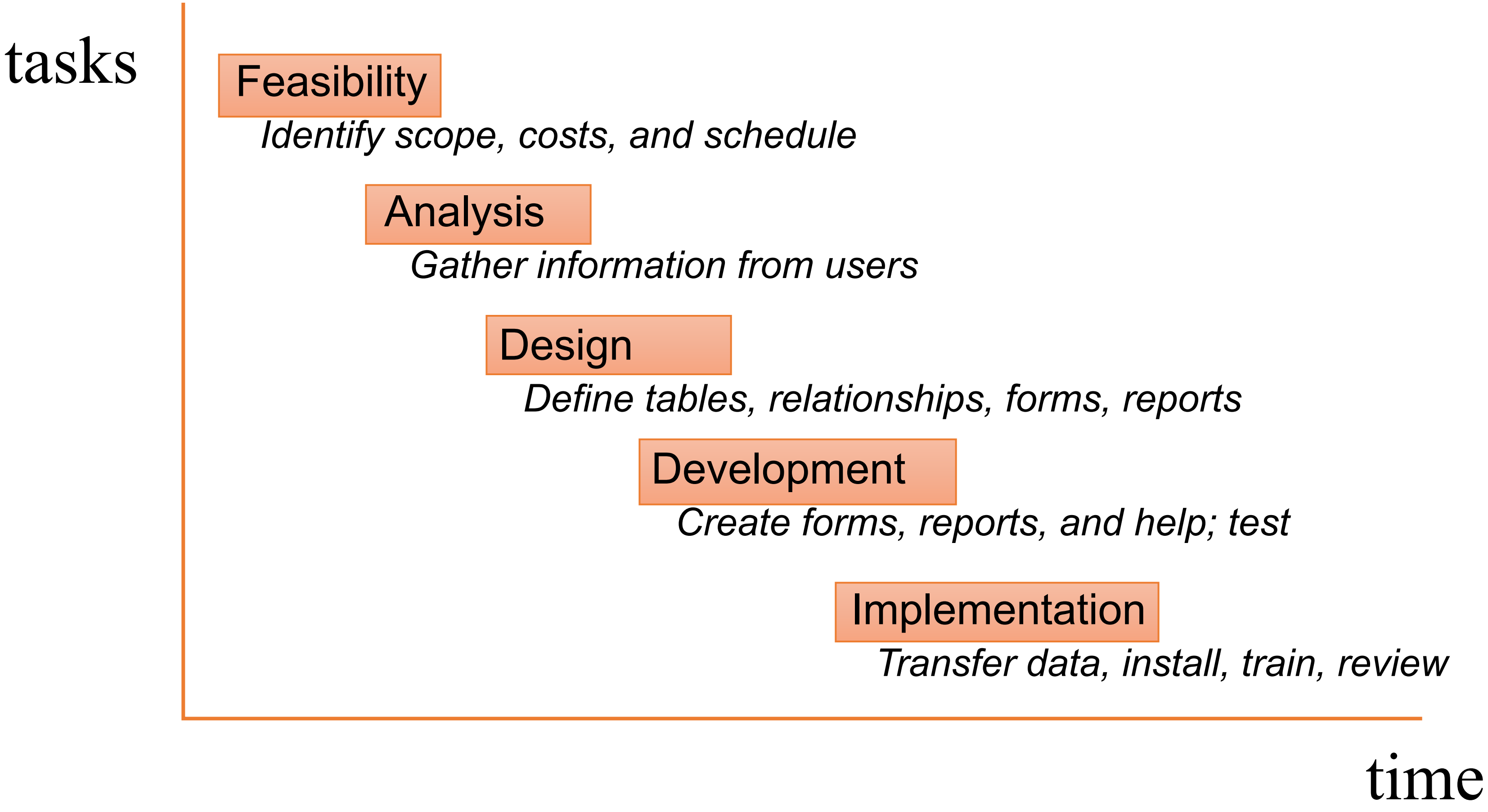
Why use DBMS

- To develop software applications In less time.
- Data independence and efficient use of data.
- For uniform data administration.
- For data integrity and security.
- For concurrent access to data, and data recovery from crashes.
- To use user-friendly declarative query language.

Where is a DBMS being used

- Airlines: reservations, schedules, etc
- Telecom: calls made, customer details, network usage, etc
- Universities: registration, results, grades, etc
- Sales: products, purchases, customers, etc
- Banking: all transactions etc

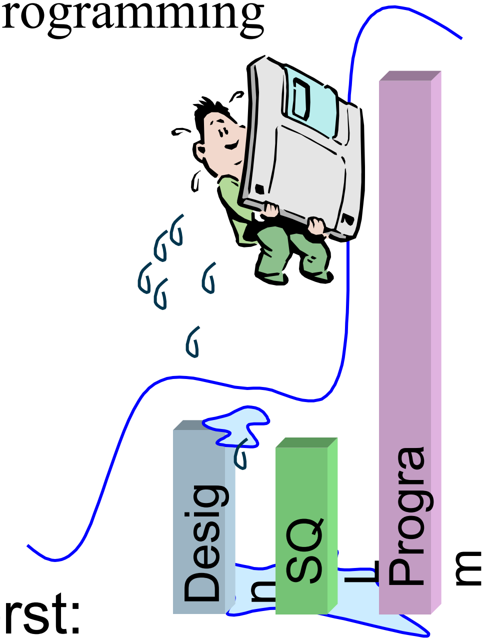
Application Development


tasks
Feasibility
Identify scope, costs, and schedule
Analysis
Gather information from users
Design
Define tables, relationships, forms, reports
Development
Create forms, reports, and help; test
Implementation
Transfer data, install, train, review
time

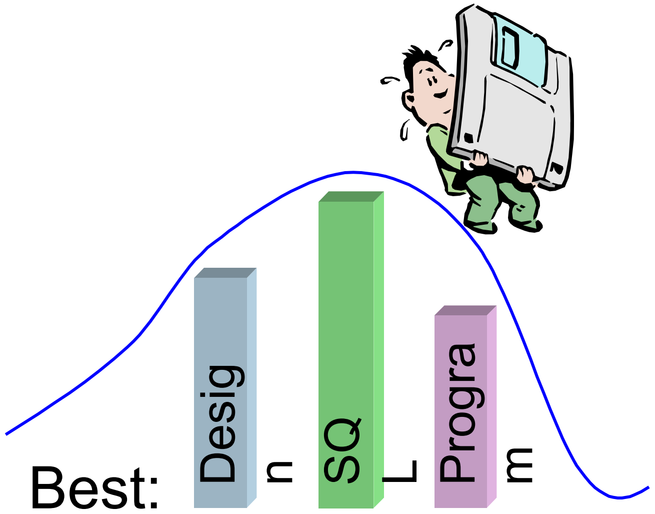
Goal: Build a Business Application

Tools:

- Database Design
- SQL (queries)
- Programming

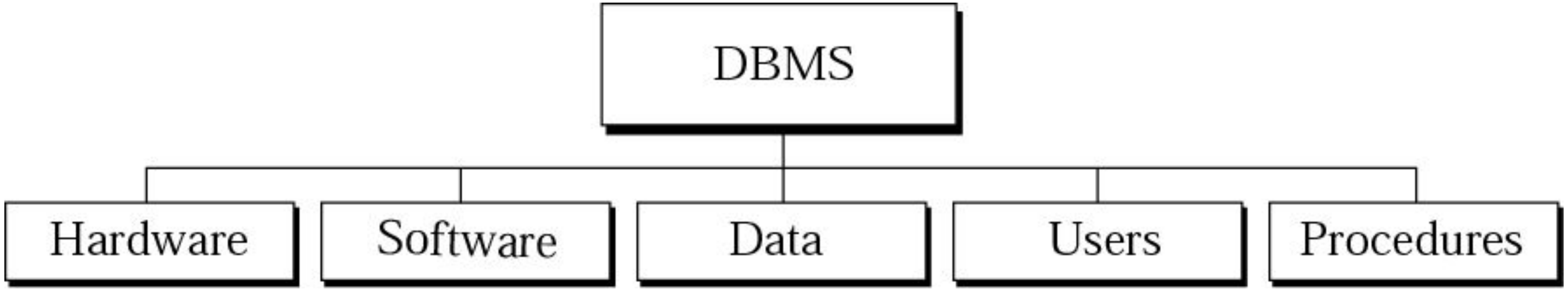


Worst:
Compensate for poor design
and limited SQL with programming.



Best:
Spend your time
on design and SQL.

DBMS components



- Hardware –
  the physical computer system that allows physical access to data.
- Software –
  the actual program that allows users to access, maintain, and update physical data.
- Data – stored physically on the storage devices
- Users –
  - End users - Normal user and DBA (Database Administrator)
  - Application programs
- Procedures –
  a set of rules that should be clearly defined and followed by the users.

DBMS Users

- End users
  - Use the database system to achieve some goal
- Application developers
  - Write software to allow end users to interface with the database system

- Database Administrator (DBA)
  - Designs & manages the database system
    - Database systems programmer
    - Writes the database software itself

Functions of a DBMS

- Views
- Security
- Integrity
- Concurrency
- Backup/Recovery
- Data Independence
- Access Languages and APIs
- Update and retrieve data

Views

- ❖ Custom representations of a database that correspond to the needs of a class of users. Stored SELECT statements.
  - Maintaining a constant user interface
  - Restricting access to specified attributes
  - Specifying user rights

- ❖ Views Permit
  - Views Provide: representations of data for different users to
    - *protect data quality*
    - *insulate users from changes in structure*

CREATE VIEW

VIEWNAME {VIEW ATTRIBUTES}

AS (SELECT ..WHERE ..)

Security

- Components that limit access or actions to limit potential damage to data.
- DBMS is managed by administrators who are determined to secure the database from all the threats in the database environment
- Some DBMS products authorize actions based on specific records and functional descriptions. However, most DBMS's limit actions on tables to one of:
  - Read: view but not change
  - Insert: read and add records
  - Update: read, insert and change records
  - Alter/Delete: read, insert, update and delete records, change table structure

Integrity

- The structure of the Database is designed in such a way to manage the data with proper rules and security.
- Components that preserve the relationship among different related records in the database.
- his feature enables DBMS to follow all the protocols set for the database and manage the data effectively. There are no redundant files in the database as the rules are setup to always keep the updated data in DBMS
- The relationship among records in the database
  - Referential Integrity
  - Non Key Integrity
  - Derived Conditions

Concurrency

- If multiple users are logging into the same database simultaneously, the database does not show any lagging while running the queries and returns the results in the same speed.
- This feature of concurrency helps multiple users to work and get results if it is a large database. As a result, the work is completed faster, and many users benefit from the same..
- Preventing two users from interfering with each other when they use the same information.
- Several algorithms are in place to make different users work on the same database
  - Lockout
    - Restricting access to users who could be misled by partial transactions
  - Versioning
    - Making trial updates on versions of the database and denying one if there is a data conflict

Advantages of DBMS

- Control of data redundancy
- Data consistency
- More information from the same amount of data
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale
- Balance conflicting requirements
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services

Data Independence

❖ Facilities that allow programs to be independent of the structure of the database.

- Addition of a field
- Changing the length of a field
- Creating a new index
- Adding or changing a relationship

Disadvantages of DBMS

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- Cost of conversion
- Performance
- Higher impact of a failure

Access Languages and APIs

- Query language is used to access the data in the database. Writing queries are easy as users need not specify how it should be done as DBMS will take care of the same.
- Structured Query Language or SQL is the most used language in the database as all vendors support this language in their database

Update and Retrieve Data

- Fundamental capability of a DBMS
- Users don’t need to know how data is stored or manipulated
- Users add, change, and delete records during updates

Backup and Recovery

- Provides a mechanism for recovering the database in the event that the database is damaged in any way.
- The process of returning the database to a correct state is called recovery.
- Periodically making a copy of the database is called backup.

Conclusion

- ❖ DBMS is good for large systems where data can be easily managed along with flexibility and data integrity in the system. It has a good backup system as well that helps in securing all the data in the database. In addition, data storage and querying is easy in DBMS when compared to other databases

Exercise:

- Design a database schema for a university registration system. Identify entities (students, courses) and define relationships.

- Compare the advantages of using a relational DBMS vs. a NoSQL DBMS for an e-commerce platform.