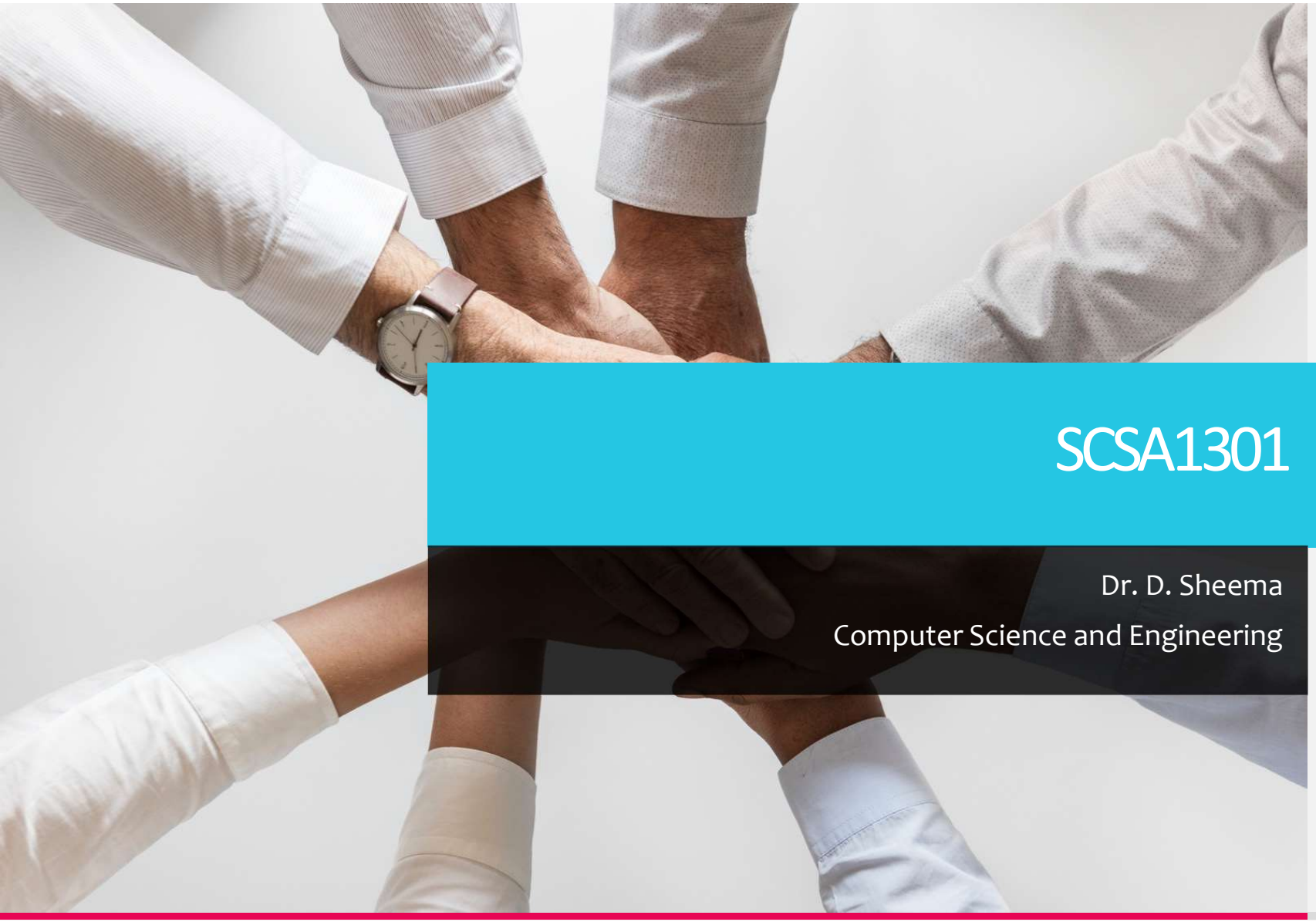


SCSA1301 DBMS

Dr. D. Sheema
Computer Science and Engineering

DBMS

What is Database Security in DBMS?

- Database security in DBMS is a technique for protecting and securing a database from intentional or accidental threats.
- Security considerations will apply not only to the data stored in an organization's database: a breach of security may impact other aspects of the system, which may ultimately affect the database structure.
- As a result, **database security** encompasses hardware parts, software parts, human resources, and data.
- To use the security efficiently, appropriate controls are required, which are separated into a specific goal and purpose for the system.
- The demand for effective security, which was frequently neglected or overlooked in the past, is now being rigorously verified by many businesses.

DBMS

Here, consider database security in the following scenarios:

- Theft and fraudulent.
- Loss of Data privacy.
- Loss of Data integrity.
- Loss of confidentiality or secrecy
- Loss of availability of data.

Why Database Security is Important?

- Security is an important concern in database management because the information stored in a database is a very valuable and, at times, quite sensitive commodity. As a result, data in a database management system must be protected from abuse and illegal access and updates.
- Although most security breaches are caused by hackers, in reality, insiders account for 80% of data loss. The extent to which an incident, such as a data breach, can harm our company is determined by several factors.

DBMS

- **Compromise of intellectual property:** Our intellectual property—trade secrets, inventions, or unique methods—could be essential for our ability to sustain an advantage in our industry. If our intellectual property is stolen or leaked, then we will lose our competitive advantage and it may be difficult to maintain or recover.

- **The reputational harm is done to our brand:** Customers or partners may refuse to buy goods or services from us (or do business with us) if they do not believe they can trust our company to protect their data or their own.

- **The concept of business continuity (or lack of it):** Some businesses are unable to operate until a breach has been resolved.

DBMS

- **Penalties or fines to be paid for failure:** The cost of failing to comply with international regulations such as the Sarbanes-Oxley Act (SAO) or Payment Card Industry Data Security Standard (PCI DSS) specific to industry regulations on data privacy, such as HIPAA, or regional privacy laws like the European Union's General Data Protection Regulation (GDPR) could be significant, with fines exceeding many millions of dollars in the worst-case scenario.
- **Costs of correcting breaches and notifying consumers about them:** Along with notifying customers of a breach, the organization that was breached must fund the investigation and forensic services such as crisis management, triage repairs to the affected systems, and much more.

DBMS

Database Security Threats

- Many software vulnerabilities, **misconfigurations**, or practices of misuse or carelessness could lead to breaches. The following are some of the most well-known causes and types of database security cyber threats.

1) SQL/NoSQL Injection Attacks

Both SQL and NoSQL databases are vulnerable to injection attacks.

- It is a type of attack that occurs when a malicious code is injected into frontend (web) apps and then transmitted to the backend database. SQL injections provide hackers with unrestricted access to any data saved in a database. There are two types of such computer attacks: **SQL injection attacks on traditional databases** and **NoSQL injection attacks on large data databases**.
- NoSQL injection occurs when a query, most commonly delivered by an end-user, is not sanitized, allowing the attacker to include malicious input that executes an unwanted command on the database.

DBMS

2) Malware

- Malware is software designed to corrupt data or harms a database.
- Malware could enter your system via any endpoint device connected to the database's network and exploit vulnerabilities in your system.
- Malware protection is important on any endpoint, but it is particularly necessary on database servers due to their high value and sensitivity.
- **Examples** of common malware include spyware, Trojan viruses, viruses, worms, adware, and ransomware.

3) Lack of Security Expertise and Education

- Databases are breached and leaked due to insufficient level of IT security expertise and education of non-technical employees, who may violate basic database security standards and endanger databases.
- IT security employees may also lack the necessary expertise to create security controls, enforce rules, or execute incident response processes.

DBMS

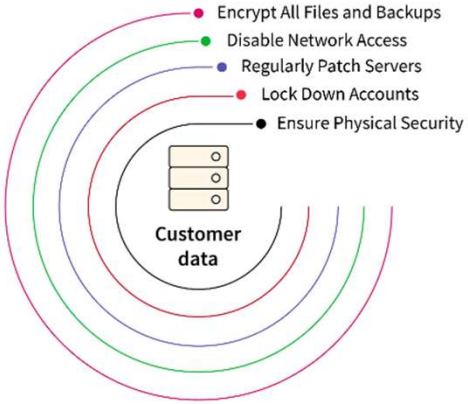
4) Denial of Service (DoS/DDoS) Attacks

- A **Denial-of-Service (DoS) attack** is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.
- In a distributed denial of service **(DDoS)** attack, fake traffic is generated by a large number of computers that are part of an attacker-controlled botnet. This results in extremely high traffic volumes, which are difficult to stop without a highly scalable defensive architecture. Cloud-based DDoS prevention services can dynamically scale up to deal with massive DDoS attacks.

DBMS

5) Exploitation of Database Software Vulnerabilities

- Attackers are continuously attempting to isolate and target software vulnerabilities, and database management software is a particularly desirable target.
- New **vulnerabilities** are identified on daily basis, and security updates are issued regularly by all open-source database management platforms and commercial database software manufacturers. However, if you do not apply these changes immediately, your database may be vulnerable to attack.



6) Excessive Database Privileges

- Database users in DBMS may have varying levels of access. However, users may abuse them, and the three basic categories of privilege abuse are as follows: excessive privilege abuse, legitimate privilege abuse, and unused privilege abuse. **Excessive privileges** always introduce unnecessary risks.

DBMS

- According to statistics, 80% of attacks on company databases are carried out by current or former workers.
- It is recommended that a strict access and privileges control policy be implemented and enforced.

7) Weak Audit Trail

- If a database is not audited, it represents a risk of noncompliance with sensitive data protection rules at the national and international levels. All database events must be automatically logged and registered, and automatic auditing solutions must be used. Failure or **unwillingness** to do so represents a major risk on multiple levels.
- Use automatic auditing solutions that have no impact on database performance.

DBMS

Major Security Vulnerabilities

- Bugs in database software components(eg-buffer overflow)
- Improper security configurations
- Use of default user accounts and passwords
- Use of null passwords
- Excessive privileges
- Lack of network isolation(external or internal)

DBMS

Types of Database Security

- The main purpose of database security is to keep secure sensitive information in a database and maintain the database's confidentiality, integrity, and availability. The types of database security are key techniques that are used to provide database security.
- Database security is important to protect from cyber-attacks which can lead to financial loss, and damage to brand reputation, business continuity, and customer confidence.

The main security **types of databases** are as follows:

- Authentication
- Database Encryption
- Backup Database
- Physical Security

DBMS

- Application Security
- Access Control
- Web Application Firewall
- Use Strong Password
- Database Auditing

1. Authentication

- Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server. In doing this, authentication assures secure systems, secure processes and enterprise information security.

DBMS

2. Database Encryption

- Database encryption is the process of converting data, within a database, in plain text format into meaningless cipher text by means of a suitable algorithm.
- Encryption of a database is costly and requires more storage space than the original data. The steps in encrypting a database are:
  - ✓ Determine the criticality of the need for encryption
  - ✓ Determine what data needs to be encrypted
  - ✓ Determine which algorithms best suit the encryption standard
  - ✓ Determine how the keys will be managed

DBMS

3. Backup Database

A database backup is a **copy of storage that is stored on a server**. Backup is used to prevent unexpected data loss. If original data gets lost, then with the help of a backup, it is easy to gain access to the data again. There are two types of database backup.

- Physical backup
- Logical backup

4. Physical security

It strictly limits access to the physical server and hardware components. Many organizations with **on-premises databases use locked rooms with restricted access** to the database server hardware and networking devices. It's also important to limit access to backup media by storing it at a secure offsite location.

DBMS

5. Application security

- An **application security policy** is a list of application security requirements and rules that regulate user access to database objects.
- Considerations for Using Application-Based Security An application security implementation should consider both application and database users and whether to enforce security in the application or in the database.

6. Access Control

- The security mechanism of DBMS must include some provisions for restricting access to the database by unauthorized users.
- Access control is done by creating user accounts and controlling login process by the DBMS.

DBMS

- So, that database access of sensitive data is possible only to those people (database users) who are allowed to access such data and to restrict access to unauthorized persons.
- The database system must also keep track of all operations performed by certain users throughout the entire login time.

7. Web Application Firewall

- A WAF or web application firewall helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet.
- It typically protects web applications from attacks such as cross-site forgery, cross-site-scripting (XSS), file inclusion, and SQL injection, among others.

DBMS

8. Use Strong Password

- Dual passwords, to enable clients to connect using either a primary or secondary password.
- Password strength assessment, to require strong passwords. Random password generation, as an alternative to requiring explicit administrator-specified literal passwords.

9. Database Auditing

- Auditing in DBMS refers to the examination of access to data stored in databases[1].
- It helps the database administrator (DBA) to keep track of the database resources and authority from the DBMS by maintaining the history of transactions stored in the database[2].
- Auditing is used to measure, monitor and have proof of access to the data stored in databases[1].

DBMS

A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security of portions of a database against unauthorized access.

It is now customary to refer to two types of database security mechanisms:

- Discretionary security mechanisms. These are used to grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).

DAC attributes include:

- User may transfer object ownership to another user(s).
- User may determine the access type of other users.
- After several attempts, authorization failures restrict user access.

DBMS

- Unauthorized users are blind to object characteristics, such as file size, file name and directory path.
- Object access is determined during access control list (ACL) authorization and based on user identification and/or group membership.
- DAC is easy to implement and intuitive but has certain disadvantages, including:
- Inherent vulnerabilities (Trojan horse)
- ACL maintenance or capability
- Grant and revoke permissions maintenance
- Limited negative authorization power

DBMS

- Mandatory security mechanisms. These are used to enforce multilevel security by classifying the data and users into various security classes and then implementing the appropriate security policy of the organization.
- For example, a typical security policy is to permit users at a certain classification (or clearance) level to see only the data items classified at the user's own (or lower) classification level. An extension of this is rolebased security, which enforces policies and privileges based on the concept of organizational roles.

  - Top secret (TS), Secret (S), Confidential (C), unclassified (U),
  - TS is the highest level and U the lowest.
  - The commonly used model for multilevel security, known as the Bell-LaPadula model.

Security Classifications

- There are three layers of database security: the database level, the access level, and the perimeter level.
- Security at the database level occurs within the database itself, where the data live.
- Access layer security focuses on controlling who can access certain data or systems containing it.
- Security policy at the perimeter level determines who can and cannot get into databases.
- Each level requires unique security solutions.

Security Level	Database Security Solutions
Database Level	• Masking • Tokenization • Encryption
Access Level	• Access Control Lists • Permissions
Perimeter Level	• Firewalls • Virtual Private Networks

DBMS

Role-Based Access Control

- RBAC-managing and enforcing security in large-scale enterprise wide systems
- Roles can be created using the CREATE ROLE and DESTROY ROLE CREATE ROLE role_name [WITH ADMIN {CURRENT_USER | CURRENT_ROLE}]
- Role hierarchy in RBAC is a natural way of organizing roles to reflect the organization's lines of authority and responsibility.
- Many DBMSs have allowed the concept of roles, where privileges can be assigned to roles.

DBMS

Distributed DBMS

- A distributed database is a set of interconnected databases that is distributed over the computer network or internet.
- A Distributed Database Management System (DDBMS) manages the distributed database and provides mechanisms so as to make the databases transparent to the users. In these systems, data is intentionally distributed among multiple nodes so that all computing resources of the organization can be optimally used.
- A **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.
- A distributed database management system (DDBMS) is a centralized software system that manages a distributed database in a manner as if it were all stored in a single location.

DBMS

Features

- Databases in the collection are logically interrelated with each other. Often they represent a single logical database.
- Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites.
- The processors in the sites are connected via a network. They do not have any multiprocessor configuration.
- A distributed database is not a loosely connected file system.
- A distributed database incorporates transaction processing, but it is not synonymous with a transaction processing system.
- It is used to create, retrieve, update and delete distributed databases.
- It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users.

DBMS

- It ensures that the data modified at any site is universally updated.
- It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously.
- It is designed for heterogeneous database platforms.
- It maintains confidentiality and data integrity of the databases.

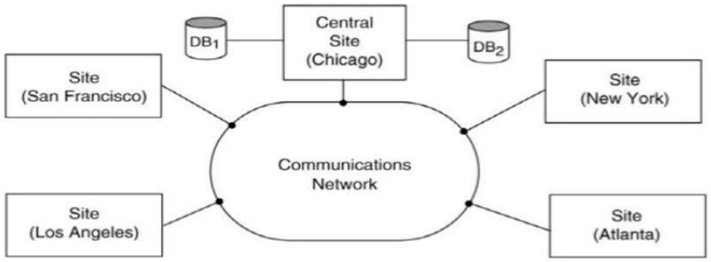
Distributed Database Vs Centralized Database

Centralized DBMS	*Distributed DBMS*
In Centralized DBMS the database are stored in a only one site	In Distributed DBMS the database are stored in different site and help of network it can access it
If the data is stored at a single computer site,which can be used by multiple users	Database and DBMS software distributed over many sites,connected by a computer network
Database is maintained at one site	Database is maintained at a number of different sites

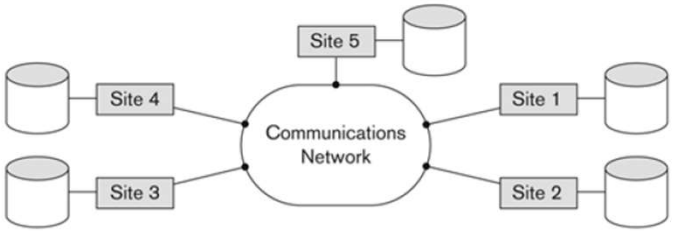
Centralized DBMS	*Distributed DBMS*
If centralized system fails,entire system is halted	If one system fails,system continues work with other site
It is a less reliable	It is a more reliable

DBMS

Centralized database



Distributed database



DBMS

Data Fragmentation in Distributed Database Design

- Fragments

  Logical units of the database

- Horizontal fragmentation (sharding)
- Horizontal fragment or shard of a relation is a subset of the tuples in that relation can be specified by condition on one or more attributes or by some other method
- Groups rows to create subsets of tuples

  Each subset has a certain logical meaning

- Vertical fragmentation
- Divides a relation vertically by columns
- Keeps only certain attributes of the relation

Complete horizontal fragmentation

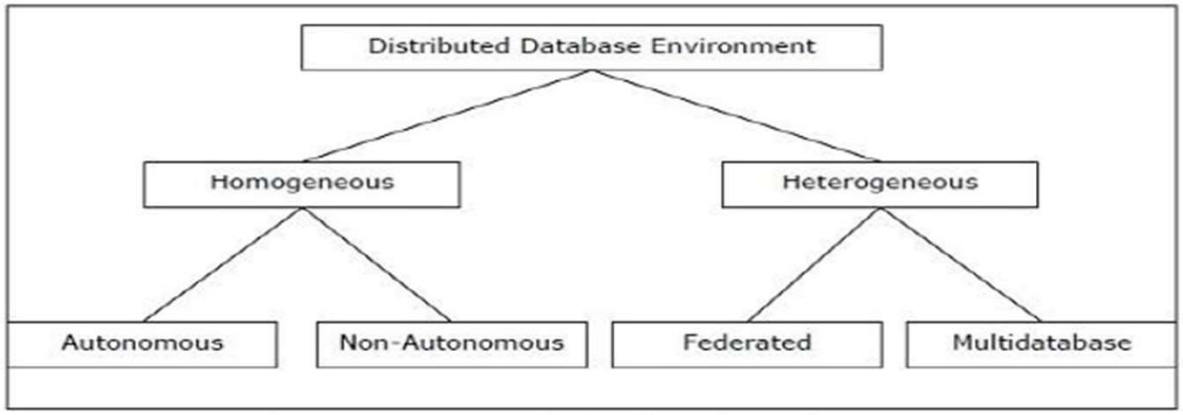
Apply UNION operation to the fragments to reconstruct relation

Complete vertical fragmentation

Apply OUTER UNION or FULL OUTER JOIN operation to reconstruct relation.

DBMS

Types of Distributed Databases



DBMS

Distributed databases can be broadly classified into homogeneous and heterogeneous distributed database environments

Homogeneous Distributed Databases

In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are −

- The sites use very similar software.
- The sites use identical DBMS or DBMS from the same vendor.
- Each site is aware of all other sites and cooperates with other sites to process user requests.
- The database is accessed through a single interface as if it is a single database.

DBMS

Types of Homogeneous Distributed Database

There are two types of homogeneous distributed database −

Autonomous − Each database is independent that functions on its own. They are integrated by a controlling application and use message passing to share data updates.

Non-autonomous − Data is distributed across the homogeneous nodes and a central or master DBMS co-ordinates data updates across the sites.

Heterogeneous Distributed Databases

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models. Its properties are −

- Different sites use dissimilar schemas and software.
- The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.

DBMS

Query processing is complex due to dissimilar schemas. Transaction processing is complex due to dissimilar software.

• A site may not be aware of other sites and so there is limited co-operation in processing user requests.

Types of Heterogeneous Distributed Databases

Federated − The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.

Un-federated − The database systems employ a central coordinating module through which the databases are accessed.

DBMS

DDBMS architectures are generally developed depending on three parameters −

• Distribution − It states the physical distribution of data across the different sites.

Autonomy − It indicates the distribution of control of the database system and the degree to which each constituent DBMS can operate independently.

• Heterogeneity − It refers to the uniformity or dissimilarity of the data models, system components and databases.

DBMS

Client - Server Architecture for DDBMS

- This is a two-level architecture where the functionality is divided into servers and clients.
- The server functions primarily encompass data management, query processing, optimization and transaction management. Client functions include mainly user interface. However, they have some functions like consistency checking and transaction management.
- Distinguish the functionality and divide these functions into two classes, server functions and client functions.
- Server does most of the data management work
  - ❑ – query processing
  - ❑ – data management
  - ❑ – Optimization
  - ❑ – Transaction management etc

DBMS

Client performs

- – Application
- – User interface
- – DBMS Client model

The two different client - server architecture are −

- Single Server Multiple Client
- Multiple Server accessed by multiple clients

- A client-server architecture has a number of clients and a few servers connected to a network.
- A client sends a query to one of the servers. The earliest available server solves it and replies.
- A Client-server architecture is simple to implement and execute due to a centralized server system.

DBMS

An Overview of Railway Reservation System

RAILWAY RESEVATION SYSTEM

PROBLEM STATEMENT

A software has to be developed for automating the manual railway reservation system. The system should have distributed functionalities as described below:-

1. RESERVE SEAT:- A passenger should be able to reserve a seat in the train specified by him if available. For this he has to fill a reservation form with the details about his journey. The clerk checks for the availability of the seat in the train and if the seat is available then he makes entries regarding train name, train number, date of journey, boarding station, destination. The passenger is the asked to pay the fair .After making payment the passenger can collect the ticket from the clerk.

2. CANCEL RESERVATION:-There may arise a case when the passenger wants to cancel his reservation .For this he has to fill a cancellation form providing all the details about the ticket reserved by him. The clerk then checks for the entries from the database and cancels the reservation finally returning the ticket amount with some deduction

3. UPDATE TRAIN INFORMATION & REPORT GENERATION :- Only the Administrator has the right to make changes in train details(train name, train no. etc.).The system should also be able to generate report when needed in the form of reservation charts , train schedule charts etc.

4. LOGIN: Only the user with specified login id & password can get access to the system. This provides security from unauthorized access.

5. VIEW RESERVATION STATUS & TRAIN SCHEDULE: All the users should be able to see the information about the reservation status & train schedule, train name, train number etc.

DFD - Railway Reservation System

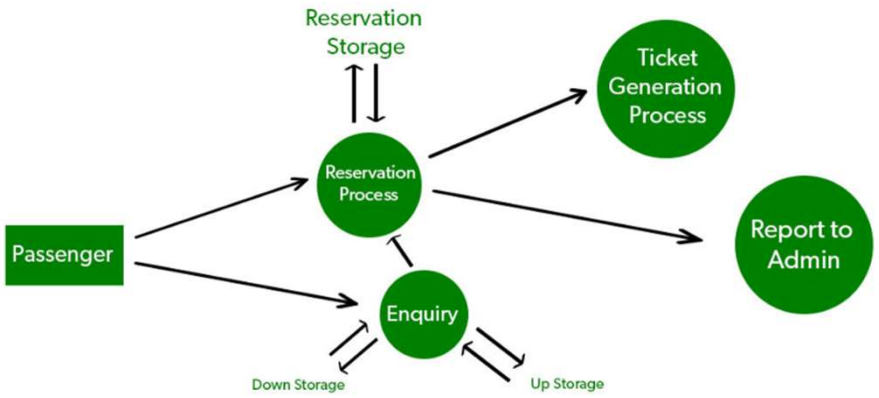
Data Flow Diagrams (DFD) are graphical representations of a system that illustrate the flow of data within the system. DFDs can be divided into different levels, which provide varying degrees of detail about the system. The following are the four levels of DFDs:

1.Level 0 DFD: This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.

0-LEVEL DFD

DFD - Railway Reservation System

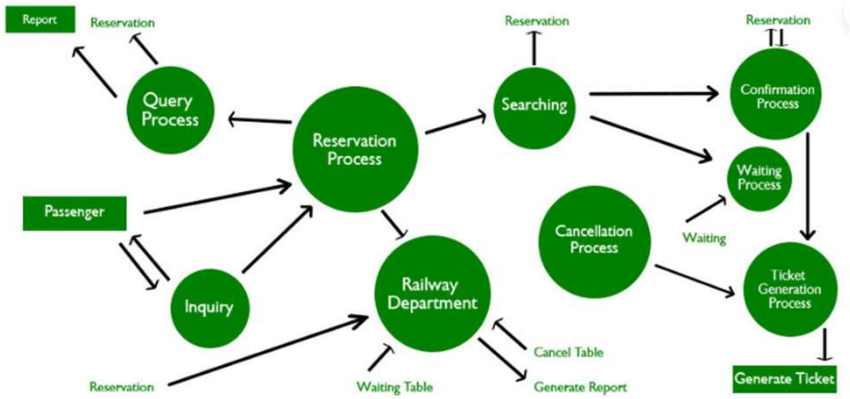
1.Level 1 DFD: This level provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown.



1-LEVEL DFD

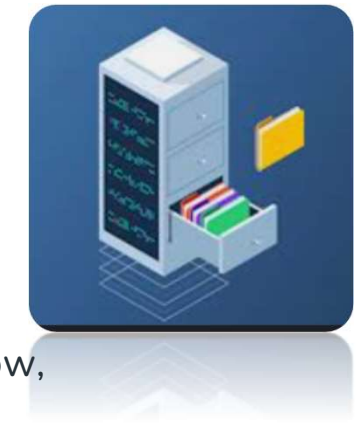
DFD - Railway Reservation System

Level 2 DFD: This level provides an even more detailed view of the system by breaking down the sub-processes identified in the level 1 DFD into further sub-processes. Each sub-process is depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process are also shown.



2-LEVEL DFD

DBMS



Advantages of using Data Flow Diagrams (DFD) include:

1.Easy to understand: DFDs are graphical representations that are easy to understand and communicate, making them useful for non-technical stakeholders and team members.

2.Improves system analysis: DFDs are useful for analyzing a system's processes and data flow, which can help identify inefficiencies, redundancies, and other problems that may exist in the system.

3.Supports system design: DFDs can be used to design a system's architecture and structure, which can help ensure that the system is designed to meet the requirements of the stakeholders.

4.Enables testing and verification: DFDs can be used to identify the inputs and outputs of a system, which can help in the testing and verification of the system's functionality.

5.Facilitates documentation: DFDs provide a visual representation of a system, making it easier to document and maintain the system over time.

DBMS

ENTITY RELATIONSHIP DIAGRAM - Railway Reservation System