Introduction to DBMS

Literature
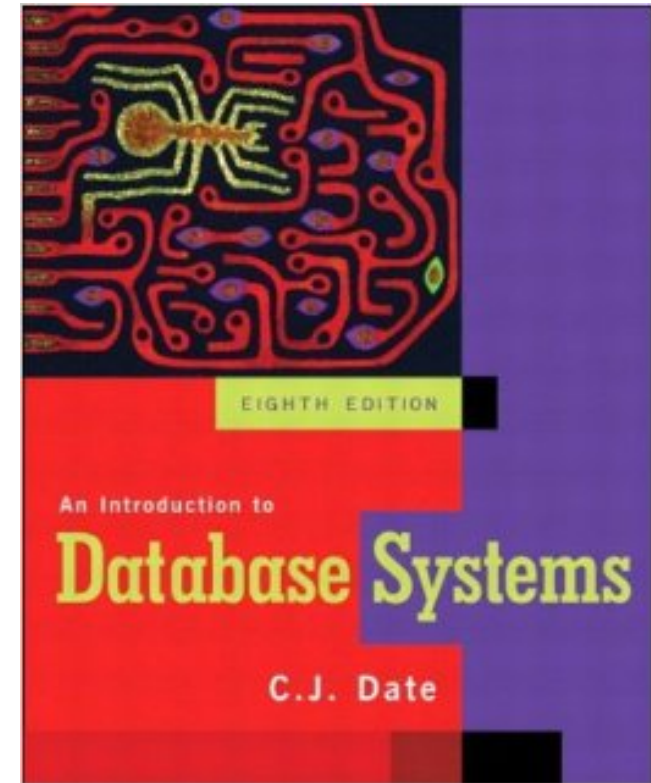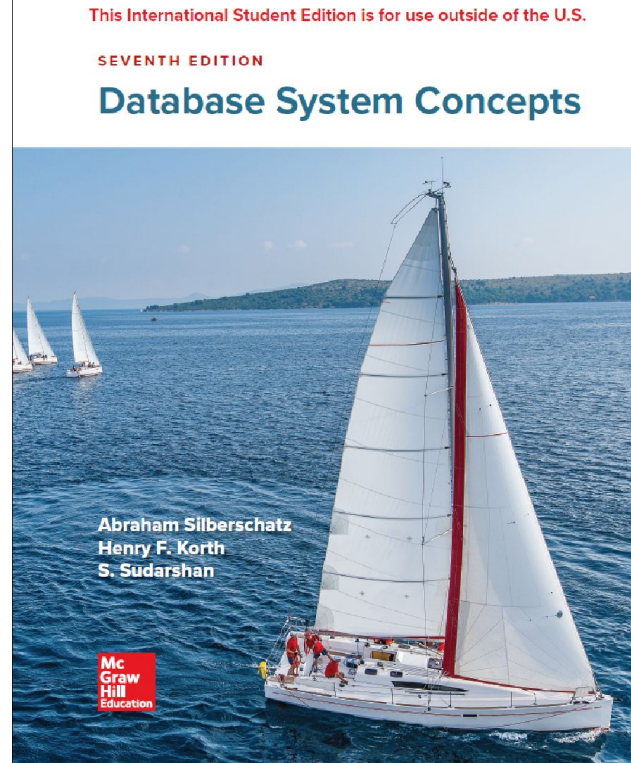
Literature
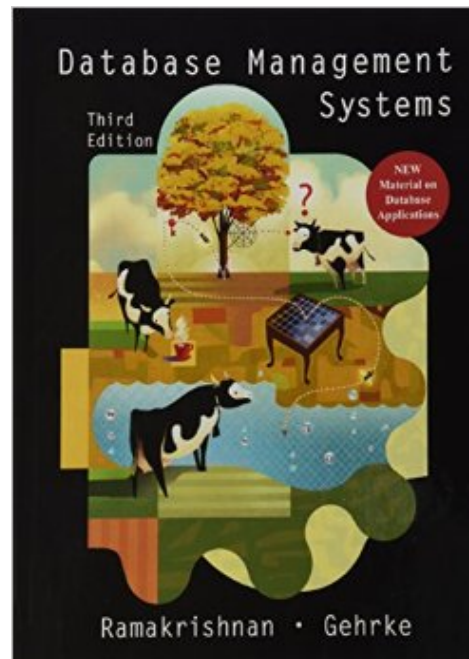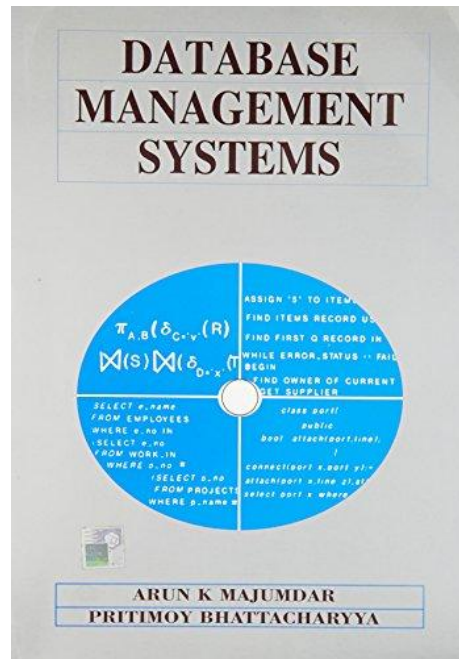
Data, Database, DBMS

❑ **Data:** Known facts that can be recorded and have an implicit meaning; raw

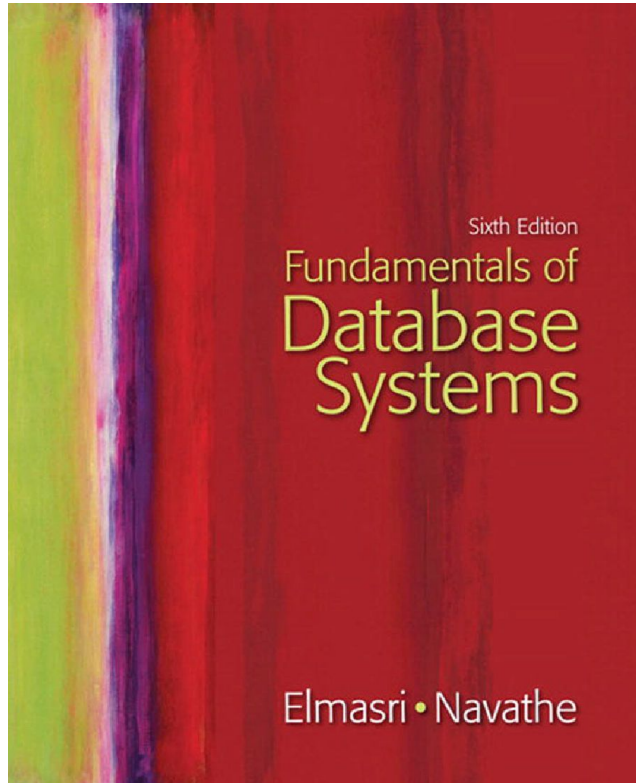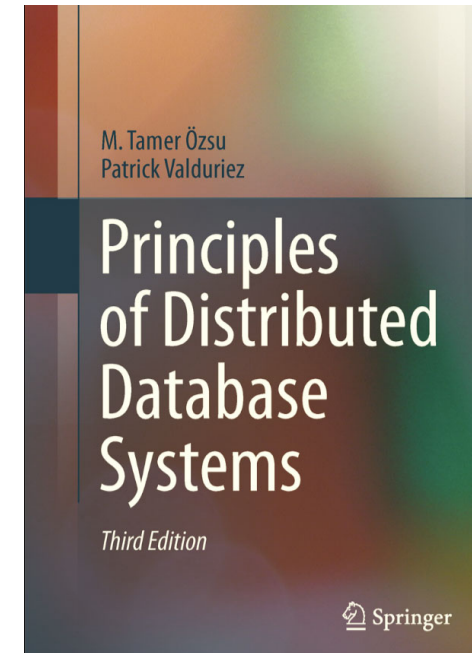❑ **Database:** A highly organized, interrelated, and structured set of data about a particular enterprise
- Controlled by a database management system (DBMS)

❑ **DBMS:**
- A collection of programs that enables users to create and maintain a database
- Set of programs to access the data
- An environment that is both convenient and efficient to use
- A software package/system to facilitate the creation and maintenance of a computerized database

❏ **Mini-world:**

- Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.

❏ Database systems are used to manage collections of data that are:

- Highly valuable
- Relatively large
- Accessed by multiple users and applications, often at the same time.

❏ A modern database system is a complex software system whose task is to manage a large, complex collection of data.

Types of Databases and Database Applications

❑ Traditional applications:

- Numeric and textual databases

❑ More recent applications:

- Multimedia databases

- Geographic Information Systems (GIS)

- Data warehouses

- Mobile databases

- Real-time and active databases

Recent Developments

❑ Social Networks started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in systems such as:

- Facebook

- Twitter

- Linked-In

❑ All of the above constitutes data

❑ Search Engines, Google, Bing, Yahoo: collect their own repository of web pages for searching purposes

❑ New technologies are emerging from the so-called non-SQL, non-database software vendors to manage vast amounts of data generated on the web:

▪ **Big data** storage systems involving large clusters of distributed computers

▪ **NOSQL** (Non-SQL, Not Only SQL) systems are non-tabular databases

❑ A large amount of data now resides on the "cloud" which means it is in huge data centers using thousands of machines. Cloud means over the internet.

What a DBMS Facilitates

❑ DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

❑ *Define* a particular database in terms of its data types, structures, and constraints

❑ *Construct* or load the initial database contents

❑ *Manipulating* the database:

- Retrieval: Querying, generating reports
- Modification: Insertions, deletions and updates to its content
- Accessing the database through Web applications (e.g., GLA Module)

❑ *Processing* and *sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

DBMS vs File System

File System

❏ File based systems were an early attempt to computerize the manual system.

❏ It is also called a traditional based approach in which a decentralized approach was taken where each department stored and controlled its own data with the help of a data processing specialist.

❏ The main role of a data processing specialist was to create the necessary computer file structures, and also manage the data within structures and design some application programs that create reports based on file data.

File System

❏ Consider an example of a student's file system
The student file will contain information regarding
the student (i.e. roll no, student name, course etc.)

❏ Similarly, we have a subject file that contains
information about the subject and the
result file which contains the information
regarding the result.

Users	Users	Users
STUDENT FILE	SUBJECT FILE	RESULT FILE
Roll_no, St_Name Course Phone_no, Address, Room, Sub,Name Marks etc.	Sub_Id Sub_Name Course Room etc.	Roll_no, St_Name, Course Sub_Name Marks etc.
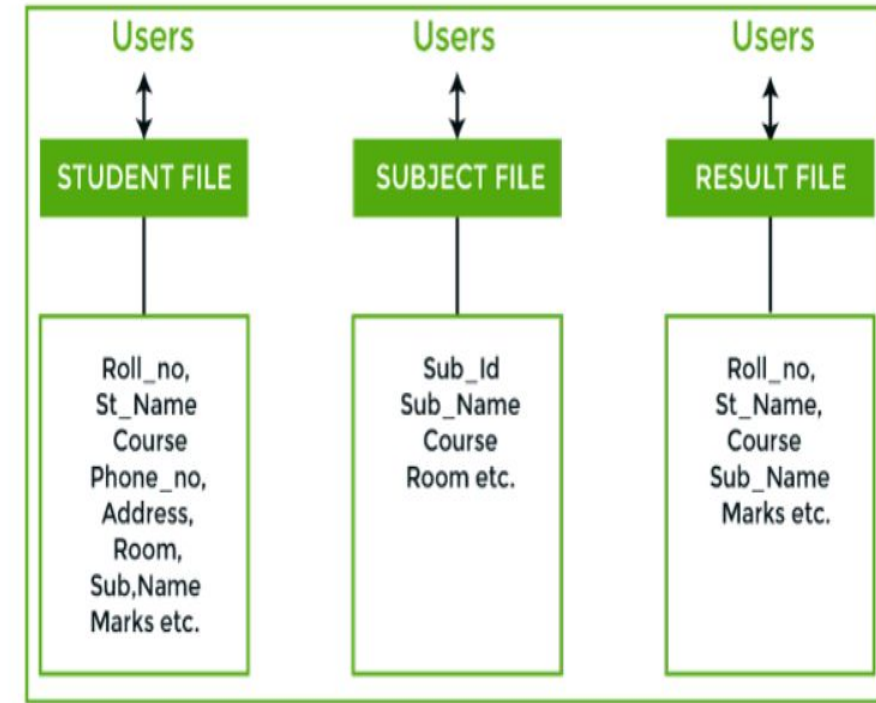
❏ Some fields are duplicated in more than one file, which leads to **data redundancy**. So to overcome this problem, we need to create a centralized system, i.e. DBMS approach

DBMS File System

DBMS

❑ A database approach is a well-organized collection of data that are related in a meaningful way which can be accessed by different users but stored only once in a system.

❑ The various operations performed by the DBMS system are: Insertion, deletion, selection, sorting etc.

❑ In the above figure, duplication of data is reduced due to centralization of data.
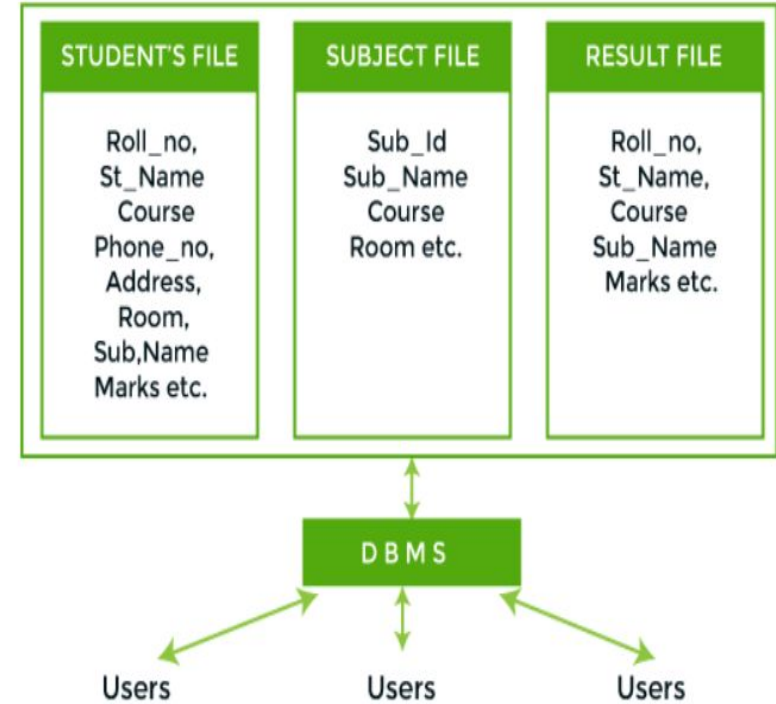
Difference between File system & DBMS

File System	Database Management System
1. File system is a software that manages and organizes the data files in a computer system.	1. DBMS is a software that is used for accessing , creating and managing the databases.
2. The File system provides the details of data representation and storage of data.	2. DBMS gives an abstract view of data that hides the details.
3. Redundant data can be present in a file system.	3. In DBMS there is no redundant data.
4.There is no efficient query processing in File system.	4. DBMS contains efficient query processing (SQL).
5.Backup and recovery of data is not efficient, because it is not possible to recover the lost data.	5. DBMS provides backup and recovery of data even if it is lost.

Difference between File system & DBMS

File System	Database Management System
6. Provide less security.	6. DBMS has more security mechanism as compared to file system.
7. Less complex	7. Complex
8. The file system does not have a crash recovery mechanism. i.e. if the system crashes while entering some data, than the content of the file will lost.	8. DBMS provides a crash recovery mechanism i.e. DBMS protects the user from the system failure
9.In the file system, concurrent access has many problems.	9. DBMS takes care of concurrent access of data using some form of locking mechanism.
10. File system is appropriate to handle data of a small scale organization or individual user.	10. DBMS is suitable for medium to large organization or multiple users.

Schema and State

1. Database Schema

❑ **Database Schema**

  ▪ The **description** of a database.

  ▪ Specified during database design and is not expected to change frequently.

  ▪ Includes descriptions of the database structure, data types, and the constraints on the database.

❑ **Schema Diagram**

  ▪ An **illustrative** display of (most aspects of) a database schema.

STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

2. Database State

❑ The actual data stored in a database at a **particular moment in time**. This includes the collection of all the data in the database.

❑ Also called database instance (or occurrence or snapshot).

❑ Refers to the **content** of a database at a moment in time.

Example of Database State

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Database Schema vs. Database State

❏ The database schema changes very infrequently.

❏ The database state changes every time the database is updated.

❏ Schema is also called **intension**.

❏ State is also called **extension**.

Database Users

❑ Database administrators

- In any organization where many people use the same resources, there is a need for a chief administrator to oversee and manage these resources.

- In a database environment, the primary resource is the database itself, and the secondary resource is the DBMS and related software.

- Database administrator (DBA) is responsible for authorizing access to the database, coordinating and monitoring its use, acquiring software and hardware resources as needed, controlling its use and monitoring efficiency of operations.

- The DBA is accountable for problems such as security breaches and poor system response time.

❑ Database designers

- Database designers are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data.

- Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Database Users

❏ End-users

- End users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use.

- They use the data for queries, reports and some of them update the database content.
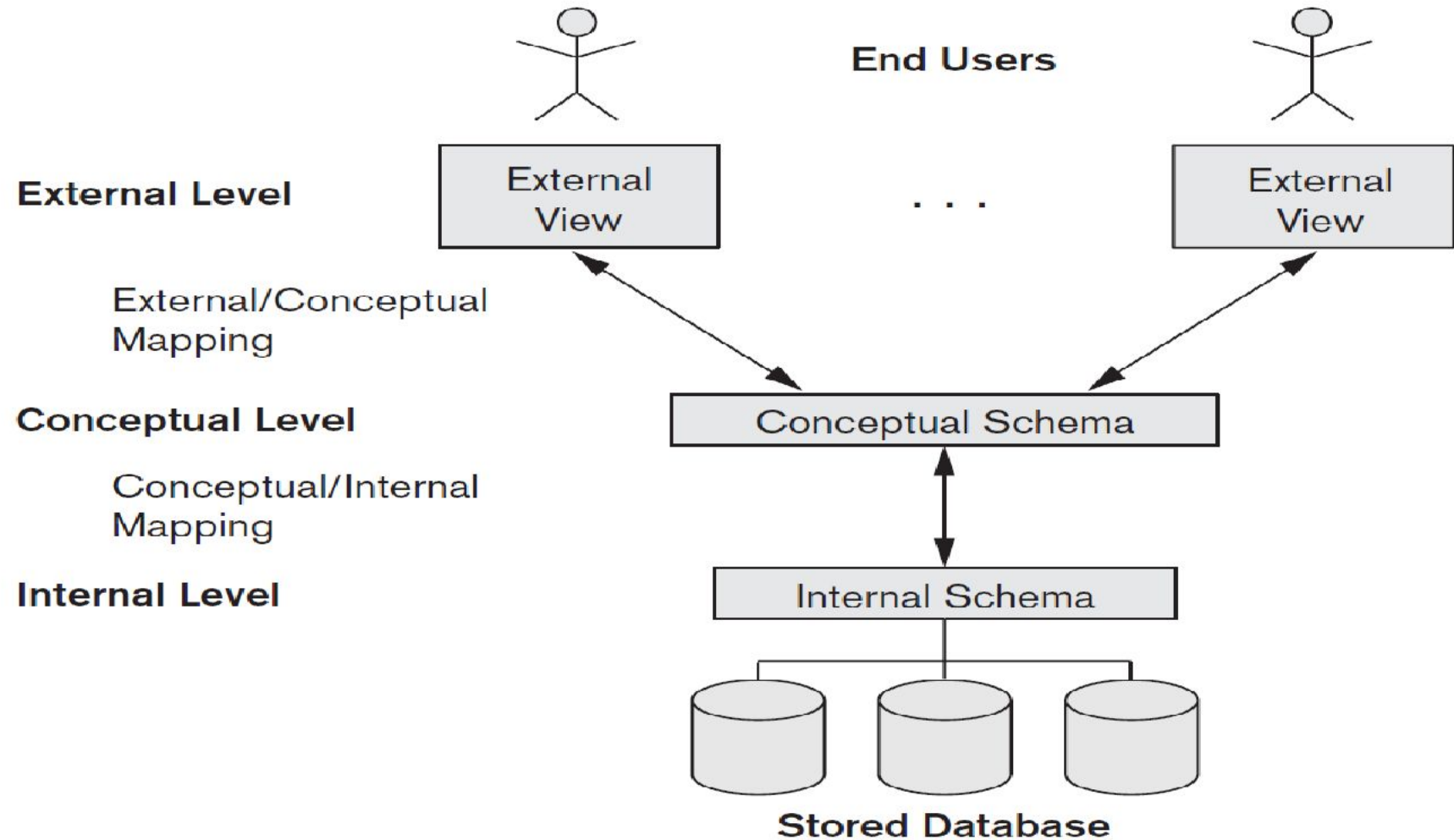
Three-Tier Architecture

Three-Schema Architecture

❏ The goal of the three-schema architecture is to separate user applications from the physical database.

❏ Defines DBMS schemas at **three** levels:

- **Internal schema(or Physical Schema):** it describes physical storage structures and access paths (e.g. indexes).

- Also Known as Low-level.

- In simple terms, it tells "How data physically stored in DB".

Three-Schema Architecture

- **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
  - hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
  - In simple terms, it describes "<u>What data exists in the Database</u>".

- **External schemas** at the external level to describe the various user views
  - Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
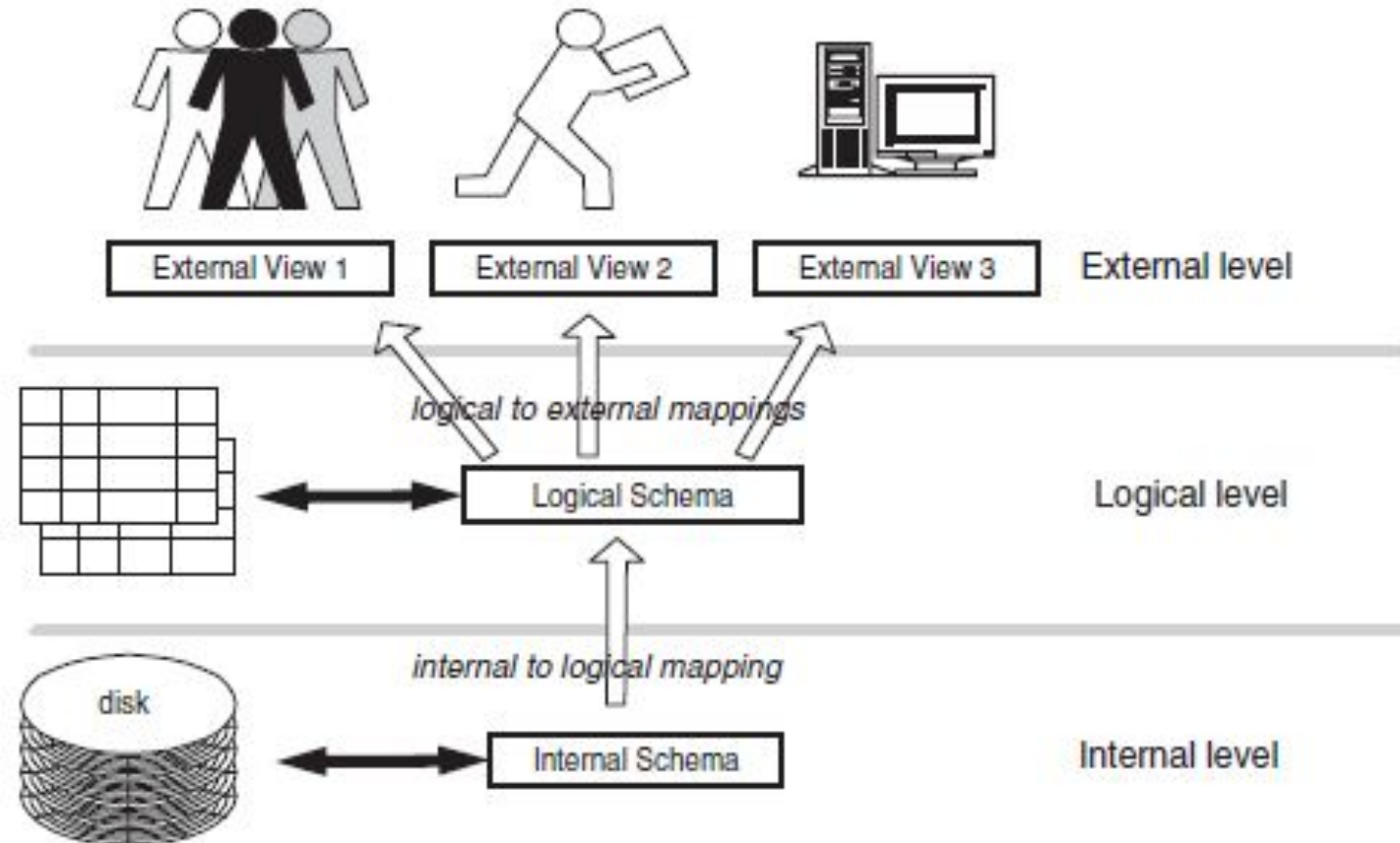
Three-Schema Architecture

❑ The processes of transforming requests and results between levels are called **mappings**.

Three-Schema Architecture

Three-Schema Architecture

External view 1

sNo	fName	lName	age	salary

External view 2

staffNo	lName	branchNo

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo

Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOf Birth;
      float salary;
      struct STAFF *next;          /* pointer to next Staff record */
};
index staffNo; index branchNo;     /* define indexes for staff */
```

Three-Schema Architecture

An example of the three levels

Customer_Loan
    Cust_ID   : 101
    Loan_No   : 1011
    Amount_in_Dollars : 8755.00
	External

CREATE TABLE Customer_Loan (
    Cust_ID   NUMBER(4)
    Loan_No   NUMBER(4)
    Amount_in_Dollars NUMBER(7,2))
	Conceptual

Cust_ID TYPE = BYTE (4), OFFSET = 0 Loan_No TYPE = BYTE (4), OFFSET = 4 Amount_in_Dollars TYPE = BYTE (7), OFFSET = 8	Internal

Data Independence

Data Independence

❑It is defined as **a property** of a DBMS that helps to change the schema at one level of a database system without having to change the schema at the next higher level.

❑**Types of Data Independence**
   • Physical data independence
   • Logical data independence

Data Independence

❏ 1. <u>Physical data independence</u>

- ▪ With Physical data independence, you can easily change the physical storage structures or devices without an effect on the conceptual schema.

- ▪ Any change done would be absorbed by the mapping between the conceptual and internal levels.

Data Independence

- **Examples of changes under Physical Data Independence:** Due to Physical independence, any of the below change will not affect the conceptual layer.

    o Using a new storage device like Hard Drive or Magnetic Tapes

    o Modifying the file organization technique in the Database

    o Switching to different data structures.

    o Changing the access method.

    o Modifying indexes.

    o Changes to compression techniques or hashing algorithms.

    o Change of Location of Database from say C drive to D Drive

Data Independence

❑ **2.** Logical data independence

- Logical Data Independence is the ability to change the conceptual scheme without changing
  - External views
  - External API or programs
- We may change the conceptual schema to expand the database (by adding records), to change the constraints, or to reduce the database.
- For example: for user1 application program is different, for user2 application program is different, and so on.

❑ Logical data independence is achieved by implementing **VIEWS**. Views are virtual table.

Data Independence

- **Examples of changes under Logical Data Independence:** Due to Logical independence, any of the below change will not affect the external layer.
  - Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
  - Merging two records into one
  - Breaking an existing record into two or more records

Difference between Physical and Logical Data Independence

Logica Data Independence	Physical Data Independence
Logical Data Independence is mainly concerned with the structure or changing the data definition.	Mainly concerned with the storage of the data.
Compared to Logic Physical independence it is difficult to achieve logical data independence.	Compared to Logical Independence it is easy to achieve physical data independence.
Concerned with conceptual schema	Concerned with internal schema
Example: Add/Modify/Delete a new attribute	Example: change in compression techniques, hashing algorithms, storage devices, etc

Classification of DBMSs

Classification of DBMSs

❏ Based on the data model used

- Legacy: Hierarchical, Network.
- Currently Used: Relational, Object-oriented, Object-relational
- Recent Technologies: Key-value storage systems, NOSQL systems: document based, graph-based DBMSs etc.

❏ Based on Architecture

- Single-user (typically used with personal computers) vs. multi-user (most DBMSs).
- Centralized (uses a single computer with one database) vs. distributed (multiple computers, multiple DBs)

1. Based on Data Models

- A set of rules and standards that defines how the database organizes/store data is called database model.

- It also defines how users views the organization of data.

Hierarchical Model

❑ Hierarchical Database model is one of the oldest database models.

❑ The hierarchical model represents data as hierarchical tree structures. Each hierarchy represents a number of related records.

❑ One to many relationship.
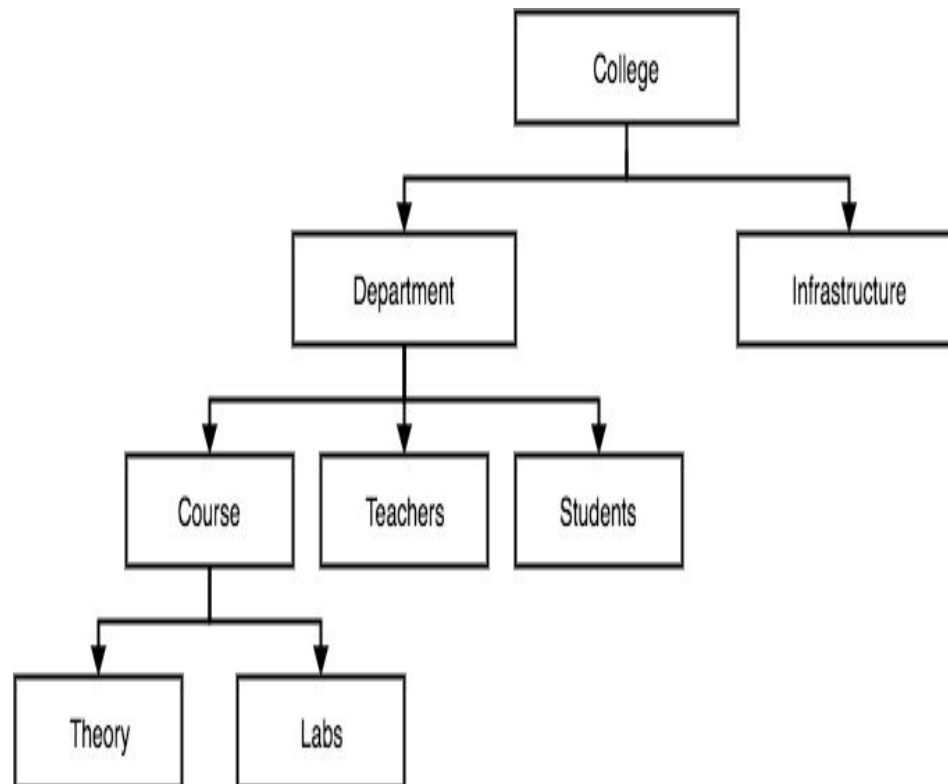
Network Model

❑ It is the extension of Hierarchical Model.

❑ The Network model represents data as a graph.

❑ The main difference of the network model from the hierarchical model, is its ability to handle many to many (M:N) relations.
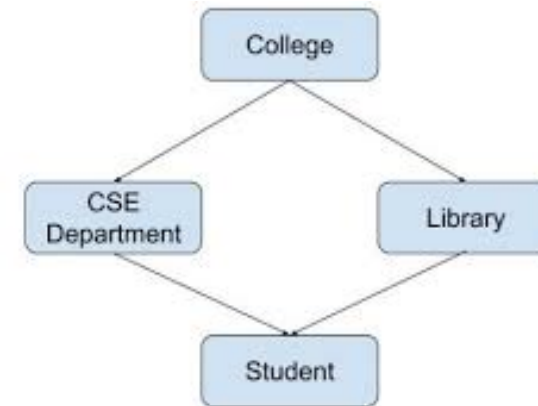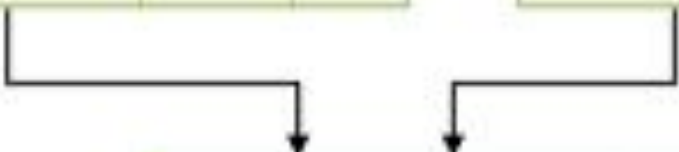

Network Model

Relational Model

❏ Relational model stores data in the form of Tables.

❏ Tables consist of rows and columns.

❏ Each column has specific data types and constraints.

❏ Tables are connected with other tables.

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Object-oriented Data Models

❑ An Object oriented data model defines a database in terms of objects, their properties, and their operations.

❑ All the data and relationships of each object are combined as a single unit.

❑ Objects with the same structure and behavior belong to a **class**.

❑ The operations of each class are specified in terms of predefined procedures called **methods**.

Employee	Department
Attributes	**Attributes**
Name Job_Title Phone_no Salary Dept_id	Dept_id Dept_name
Methods	**Methods**
Get Hired Change Number	Change Department
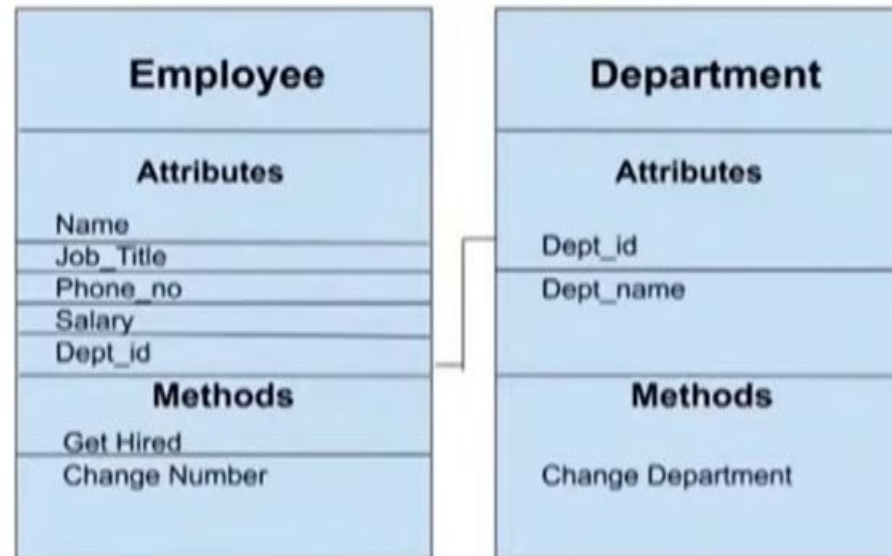
Object_Oriented_Model

Object-Relational Models

❑ Relational DBMSs have been extending their models to incorporate object database concepts.

❑ These systems are referred to as object-relational or **extended relational systems.**

❑ An Object relational model is a combination of a Object oriented database model and a Relational database model. So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.

❏ One of the major goals of Object relational data model is to close the gap between relational databases and the object oriented practises frequently used in many programming languages such as C++, C#, Java etc.

❏ The problem with this model is that this can get complex and difficult to handle. So, proper understanding of this model is required.

❏ Relational model = ER diagrams
❏ Object-Relational = EER diagrams

XML model

❏ Extensible Markup language (XML) is widely used for representing the semi-structured data.

❏ Semi structured model is an evolved form of the relational model.

❏ The XML model has emerged as a standard for exchanging data over the Web.

❏ XML uses hierarchical tree structures.

❏ It combines database concepts with concepts from document representation models.

❏ Data is represented as elements with the use of tags, data can be nested to create complex hierarchical structures.

XML

- <student 1>
  <Roll No.>...................</Roll No.>
  <Name>...................</Name>
  <Class>...................</Class>
  <Age>...................</Age>
</student 1>

<student 2>
  <Name>...................</Name>
  <Class>...................</Class>
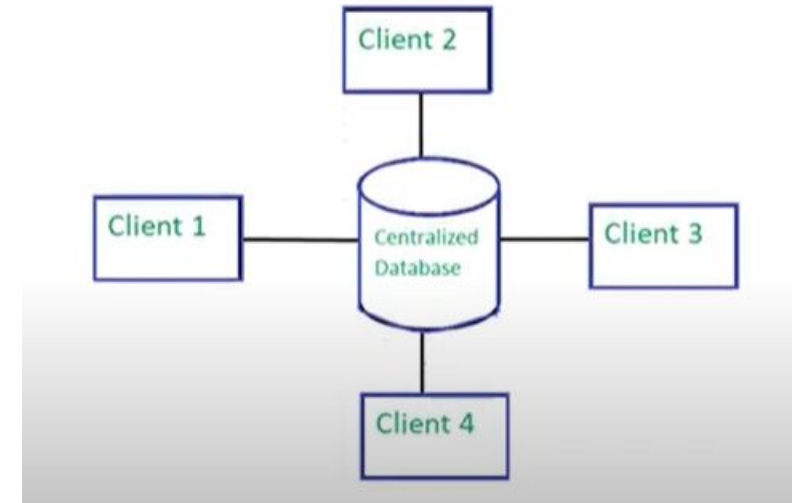 <Age>...................</Age>
</student 2>

2. Based on Architecture

1. Centralised Database

- A database that is stored, located and maintained at a single location this centralized database is accessed via an internet connection.

  Advantages

- it is less costly.

- it is easy to manage and update.

- very minimal data redundancy because all the

data is stored at a single place.

Centralised Database

Disadvantages

- Performance issues(multiple users access the database at the same time).
- Due to single failure data will be lost and user cant access the database
- Highly dependent on network.

2. Distributed Database

• It is an integrated collection of database that is physically distributed across sites in a computer network.

• that appears to users as a single database.

Advantage

• If one database fails then users can access the other database files.

• This databases is more secure as compared to centralised database.
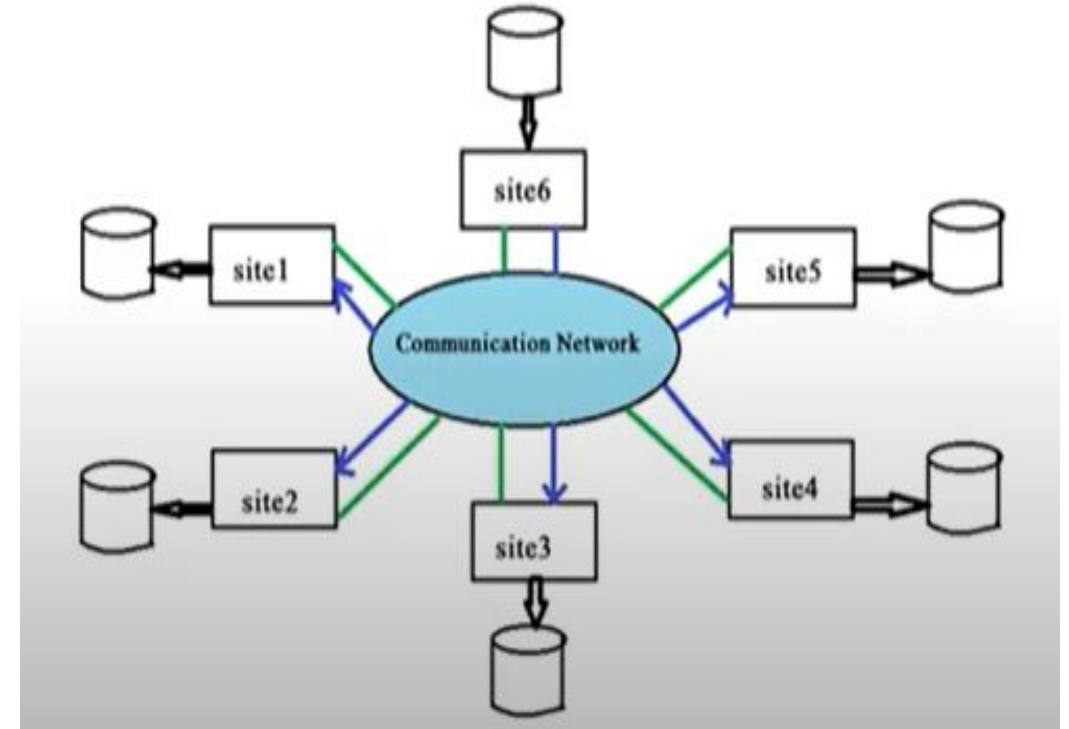
Distributed Database

- More reliable -Due to single failure entire data will not lost.

- Data traffic is less.

- Performance- No bottlenecks as the load is spread over multiple server.

Disadvantages

- Data integrity concerns- need to update data in multiple sites.

- Time for synchronization between sites.

- This database is costly.

- It is difficult to maintain and update due to its complexity.

Difference Between Two-Tier And Three-Tier Database Architecture

S.NO	Two-Tier Database Architecture	Three-Tier Database Architecture
1	It is a Client-Server Architecture.	It is a Web-based application.
2	In two-tier, the application logic is either buried inside the user interface on the client or within the database on the server (or both).	In three-tier, the application logic or process resides in the middle-tier, it is separated from the data and the user interface.
3	Two-tier architecture consists of two layers : Client Tier and Database (Data Tier).	Three-tier architecture consists of three layers : Client Layer, Business Layer and Data Layer.
4	It is easy to build and maintain.	It is complex to build and maintain.
5	Two-tier architecture runs slower.	Three-tier architecture runs faster.
6	It is less secured as client can communicate with database directly.	It is secured as client is not allowed to communicate with database directly.
7	It results in performance loss whenever the users increase rapidly.	It results in performance loss whenever the system is run on Internet but gives more performance than two-tier architecture.
8	Example – Contact Management System created using MS-Access or Railway Reservation System, etc.	Example – Designing registration form which contains text box, label, button or a large website on the Internet, etc.