UNIT 1:

FUNDAMENTALS OF DBMS:

1. DEFINE FOLLOWING TERMS.

1. **Data**
   - Data is a collection of facts, such as text, numbers, words, images, sounds etc.
   - For example:
     - Your name, age, height, weight, Photo of student, contact details etc.
2. **Information:**
   - Information means processed data.
   - Data is raw so that we need to do process like calculations, comparisons, analysis, summarization etc.
3. **Database:**
   - Database is an organized collection of meaningful data that designed for a specific purpose.
   - It is a place where data is stored & retrieve data in a way that we want.
   - Some real word example of database are:
     - Telephone directory
     - Dictionary
     - Student attendance register
     - Employee register
4. **DBMS**
   - DBMS stands for Database Management System.
   - It is collection of software that store, retrieve, manage & maintain database.
   - Some popular DBMS software are:
     - DBASE , DBASE II, FOXPRO, FOXBASE, SQL SERVER, ORACLE, MYSQL
5. **RDBMS:**

- RDBMS stands for relational database management system, which is a program that creates, updates, and manages relational databases.
- Relational databases are a collection of data that organized in predefined relationships, or logical connections, between tables.

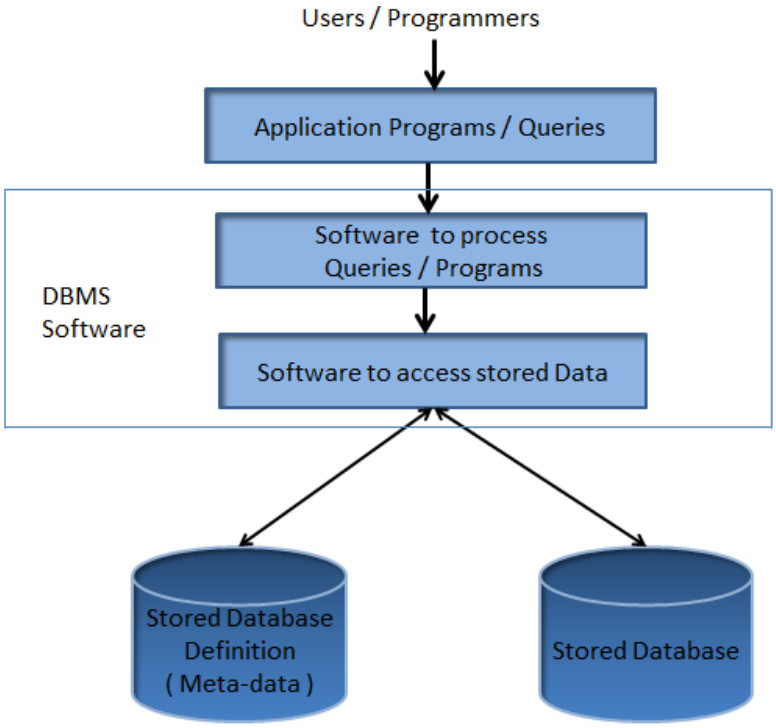
6. **Field:**
   - A field is a character or group of characters that have a specific meaning.
   - A field is a single piece of information that contain similar types of data.
7. **Record /tuple:**
   - A record is a collection of logically related fields.
8. **Meta data:**
   - Data about data is call Meta data.
   - Meta data stores information about tables, fields of tables, data type & the relationship between tables.
9. **Database designer:**
   - A database designer is in charge of designing, developing, executing and preserving a company's data management systems.
10. **Database Administrator (DBA):**
    - DBA stands for "Database Administrator."
    - DBA is a person that controls, protects, and defends the system's data so that users can run their businesses safely and privately.

2. WHAT IS DBMS? EXPLAIN DBMS ENVIRONMENT.

- DBMS stands for Database Management System.
- DBMS a collection of software that store, retrieve, manage & maintain database.
- DBMS are pieces of software that securely store and retrieve user data. It consists of many software programs that use the database.

DATABASE ENVRIONMENT:

- A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data, which consist of software, hardware, people, techniques of handling database and data.



Database System Environment

- Design of a new application for an existing database or design of a brand new database starts with a phase called requirements specification and analysis.
- These requirements are documented in detail and transformed into a conceptual design that can be represented and manipulated using some computerized tools so that it can be easily maintained, modified, and transformed into a database implementation.
- The design translated to a logical design that can expressed in a data model implemented in a commercial DBMS.
- The final stage is physical design, during which further specifications provided for storing and accessing the database.

3. WRITE DOWN THE DIFFERENCE BETWEEN DBMS AND RDBMS.

DBMS	RDBMS
DBMS stands for Database Management System.	RDBMS stands for Relational Database Management System.
DBMS is a set of related records that used for accessing, storing and managing a data.	RDBMS is a set of software programs that used for creating, maintaining, modifying & determining relational database.
A General-purpose software provides the users with the process of defining, constructing & manipulating the database for various applications.	It can used to create the application that a user will require for interacting with the data that stored within the database.
The performance of DBMS operation is not good	The RDBMS operation gives better performance
It is not based on the concept of relationships	It is based on the concept of relationships
The speed of DBMS operation is very slow	The speed of RDBMS operation is very faster
Hardware and Software requirements are less	Hardware and Software requirements are high
Facilities and Utilities are limited.	Provide High Facilities and Utilities.
It uses the concept of file	It uses the concept of table
DBMS uses a 3GL (Third Graphical Language)	RDBMS uses a 4GL (Fourth Graphical Language)
Provides less security as compared to RDBMS	Provides high security as compared to DBMS.
Normalization is not present	Normalization is present
It deals with small quantity of data.	It can work with large amount of data.

It support single user.	Multiple user can access data.
Data redundancy is the issue.	Data redundancy is reduced.
Examples of DBMS are : DBASE DBASE-II FOXBASE FOXPRO	Examples of RDBMS are: ORACLE SQL SERVER MYSQL INGERS

4. EXPLAIN CHARACTERISTICS / FEATURES OF DBMS.

1. **Real Word Entity:**
    - DBMS allows you to store data like real-world entities & describe by its properties.
    - For example
        - We can store data of any student where student that is stored like real-world entity with properties like his/her name, gender, age, roll number, etc.
2. **Self describing nature:**
    - Database system not only stores data but also contains Meta data.
    - Metadata maintains automatically.
3. **Concurrent Use of Database:**
    - Multiple users allowed to access data simultaneously.
    - Concurrent access to centralized data can allowed under some supervision.
    - This results in better performance of system and faster response.
4. **ACID Properties:**
    - ACID stands for Atomicity , Completeness, Isolation, and Durability
    - DBMS supports ACID properties.
    - In DBMS that the real purpose of data should not be lost while performing transactions like delete, insert, and update.
5. **Insulation Between Data & Program:**
    - In DBMS, structure of data files is not stored in the program but it is stored in system catalogue.
    - So that any changes in the data do not have any effect on application software.

6. **Security:**
    - DBMS provides security to the data stored in it because
    - It prevents unauthorized access to data
    - Some of the users can access the whole database while other can access a small part of database.

7. **Data Integrity:**
    - Data in database must be correct and consistent.
    - DBMS provides different ways to implement such type of constraints (rules).
    - This improves data integrity in a database.

8. **Relational data base:**
    - Relational databases organize data into tables consisting of rows and columns.
    - Each row represents a single record while each column holds a value associated with a specific field & linked together.

9. **Query Language:**
    - Queries used to retrieve and manipulate data.
    - It allows users to access & manipulate data they want by applying different sets of queries.

10. **Backup & recovery:**
    - There are many chances of failure of the whole database.
    - DBMS provides a solution is to take backup of database and whenever it needed, it can be stored back.

5. EXPLAIN ADVANTAGES OF DBMS.

- Database management system have following advantages as compare to file system:
    1. Remove Data Redundancy
    2. Data Sharing
    3. Data Consistency
    4. Data Access
    5. Data Integrity
    6. Data Security
    7. Concurrent Access Of Data
    8. Guaranteed Atomicity
    9. Backup And Recovery

1. Remove Data Redundancy (Duplication):

- Due to centralized database, it is possible to avoid duplication of information.
- This leads to reduce data redundancy.

2. Data Sharing:

- All authorized user and application program can share database easily.

3. Data Consistency:

- Data inconsistency occurs due to data redundancy.
- With reduced data redundancy, such type of data inconsistency can removed.
- This results in improved data consistency.

4. Data Access:

- DBMS utilizes a variety of techniques to retrieve data.
- Required data can retrieved by providing appropriate query to the DBMS.

- Thus, data can accessed in convenient and efficient manner.

5. Data Integrity:

- Data in database must be correct and consistent.
- Therefore, data stored in database must satisfy certain types of constraints (rules).
- DBMS provides different ways to implement such type of constraints (rules) to improve data integrity.

6. Data Security:

- DBMS provides way to control the access to data for different user according to their requirement.
- It prevents unauthorized access to data.

7. Concurrent Access Of Data:

- Multiple users allowed to access data simultaneously.
- Concurrent access to centralized data allowed under some supervision.
- This results in better performance of system and faster response.

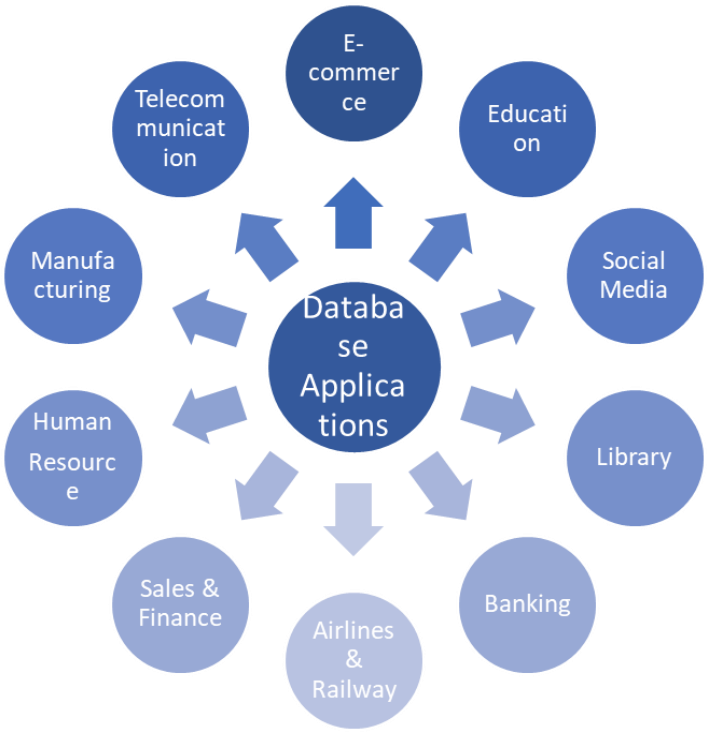
8. Guaranteed Atomicity:

- Any operation on database must be atomic.
- This means, operation must be executed either 100% or 0%.
- This type of atomicity guaranteed in DBMS.

9. Backup & recovery:

- There are many chances of failure of the whole database.
- DBMS provides a solution is to take backup of database and whenever it needed, it can be stored back.

6. LIST AND EXPLAIN THE APPLICATIONS OF DBMS.



1. **Ecommerce:**
   - Ecommerce platform use database to maintain Inventory Information, Purchases, sales records, Invoices, billing Etc.
2. **Education:**
   - Schools and colleges use databases for student's information, course registration, result, and other information.
3. **Social media:**
   - Social media platform uses database to store User Data, log in information, Preferences, Etc.
4. **Library:**
   - Library uses database for keeping records of Books, Issue Date, Return Date & check availability of books etc.
5. **Banking:**
   - Banks use databases for customer inquiry, accounts, loans, and other transactions.

6. **Airlines & railways:**
    - Airlines & railways uses database for Flight or rail Information, Reservation Information, Tickets, Schedules, Etc.
7. **Telecommunication:**
    - Telecommunication departments use databases to store information about the communication network, telephone numbers, record of calls, for generating monthly bills, etc.
8. **Sales & finance:**
    - Sales & finance uses database for Store Product Information, Purchases Of stocks and bonds, sales, customer and transaction details etc.
9. **Human resource:**
    - Human resource department uses database for Employee Information, Salary, Pay cheques, taxes Etc.
10. **Manufacturing:**
    - Manufacturing departments uses database for Supplier Information, Bills, Inventory, Etc.

7. WRITE A NOTE ON DATABASE USERS

- There are several types of database users.

❖ **Actors on the Scene:**

- Database Administrator
- Database Designers
- End Users
  - Casual End User
  - Native or Parameter End User
  - Sophistical End User
  - Standalone User
- System Analysts and Application Programmers (Software Engineers)

Database Administrator:-

- Administrating these resources is the responsibility of database administrator (DBA).
- The DBA is responsible for authorizing the access to the database coordinating and monitoring its use and acquiring software and hardware resources, as they required.
- The DBA is responsible for handling the problems such as breach of security or poor response time.

Database Designer:-

- The database designers are responsible for identify the data to be stored in the database and for choosing the appropriate structure to represent and store these data.

- Database designers communicate with potential users to understand their needs and create a design that meets their demands.

End User:-

- The end user are the persons whose jobs required access to the database for querying, updating and for generating the reports.
- Following are the various categories of End User.

1. Casual End Users:-

- This type of user occasionally accesses the database but they may require different information at each time.

2. Naive or Parametric End Users:-

- This type of user makes a sizeable portion of the database and the user.
- Their main job function revolves around consistently query and updating the database.
- The common examples of such type of user are bank clerks and reservation clerks.

3. Sophisticated End Users:-

- This category of end users include engineers, scientists, businessmen, and others who possess extensive knowledge of DBMS to effectively implement application that answer their intricate needs.

4. Standalone User:-

- This user maintains a personal database using a ready-made program with a simple menu or graphics interface.
- An example is the user of accounting package that stores variety of personal information and financial information for the tax planning purpose.

System Analysts and Application Programmers (Software Engineers):

- System analysts evaluate the needs of end users, especially naive and parametric ones, and provide canned transaction specifications.
- Application programmers convert these specifications into code, test, debug, document, and maintain scripted transactions.
- To do their jobs, analysts and programmers should know the DBMS's entire capabilities.

Workers behind the Scene:

1. DBMS system designers and implementers:

- They design and implement the DBMS modules and interfaces as a software package.
- The DBMS must interface with other system software, such as the operating system and compilers for various programming languages.

2. Tool developers:

- They design and implement tools—the software packages that facilitate database modeling and design, database system design, and improved performance.
- Tools are optional packages that often purchased separately.

3. Operators and maintenance personnel (system administration personnel):

- They are responsible for the actual running and maintenance of the hardware and software environment for the database system.

8. WHAT IS SCHEMA & INSTANCE?

Schema

- The overall design or descriptions of a database called schema.
- It represents the logical view of the entire database.
- A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.
- Other aspects are not specified in the schema diagram;
- Schemas changed **rarely.**
- For example, the bellow Figure shows neither the data type of each data item nor the relationships among the various files.

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

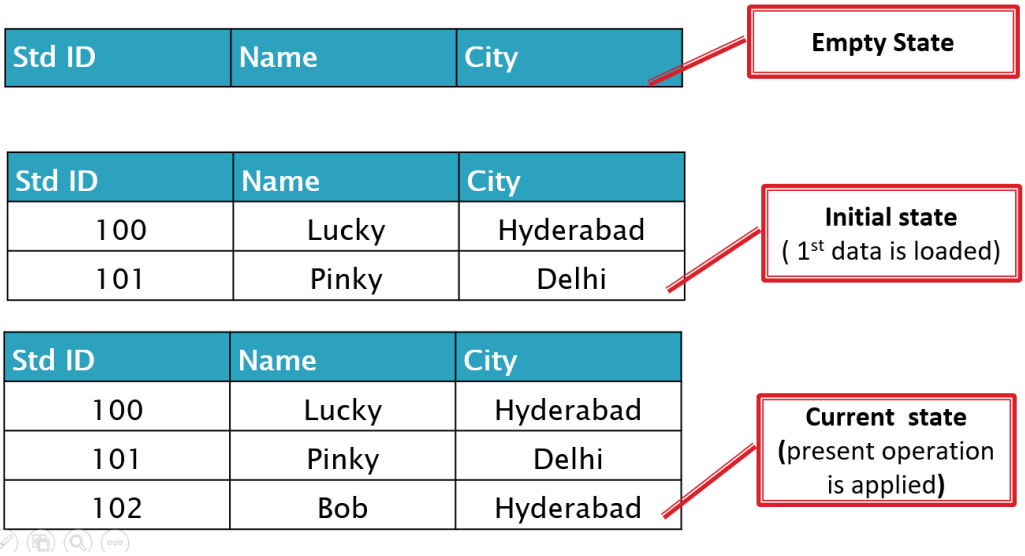
GRADE_REPORT

Student_number	Section_identifier	Grade

Instance

- Instances are the collection of information stored at a particular moment.
- It is also known as database state
- Instances changed **frequently**.

- There are three types of instance:
  - Empty State
  - Initial State
  - Current State



Empty state:

- When a new database is defined, it specify only database schema to the DBMS
- At this point, the corresponding database state is the empty state with no data.

Initial State:

- When the database is first loaded or populated then it database is in initial state.

Current State:

- An update operation is applied to the database.
- At any point in time, the database has a current state.

9. DIFFERENTIATE: SCHEMA AND INSTANCE

Schema	Instance
The overall design of database is called schema	Instances are the collection of information stored at a particular moment.
The schema remains same for whole database	Data in instance will change during the updatation, deletion or addition of data.
Schema changed rarely	Instance changed frequently
Define the basic structure of database. i.e. how the data will be stored in the database.	It is a set of information stored in particular time.

10. EXPLAIN DATA INDEPENDENCE.

- Data independence is ability to modify a schema definition in one level without affecting a schema definition in the next higher level.
- It helps you to improve the quality of the data.
- There are two levels of data independence:
  i. Physical Data Independence
  ii. Logical Data Independence

i. Physical Data Independence:

- Physical data independence is the capacity to change the internal schema without having to change the conceptual schema.
- Hence, the external schemas need not be changed as well.
- Changes to the internal schema may be needed because some physical files were reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update.
- If the same data as before remains in the database, we should not have to change the conceptual schema.

ii. Logical Data Independence

- Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
- We may change the conceptual schema to expand the database to change constraints, or to reduce the database
- For example, the external schema of UNIVERSITY database should not be affected by changing the GRADE_REPORT file (or record type) shown in Figure 1 into the one shown in figure 2

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

GRADE_REPORT

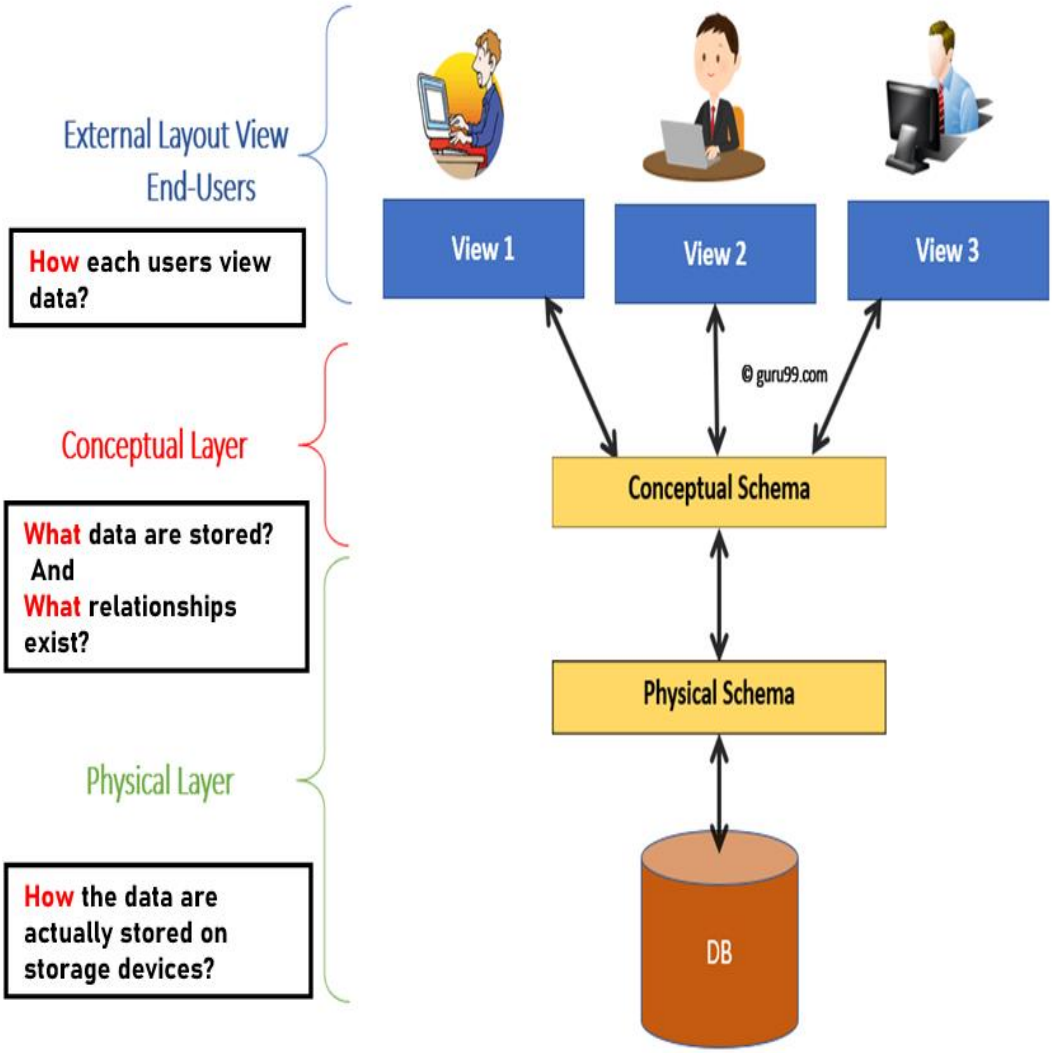
Student_number	Student_name	Section_identifier	Course_number	Grade
17	Smith	112	MATH2410	B
17	Smith	119	CS1310	C
8	Brown	85	MATH2410	A
8	Brown	92	CS1310	A
8	Brown	102	CS3320	B
8	Brown	135	CS3380	A

11. EXPLAIN THREE LEVELS ANSI SPARC DATABASE SYSTEM

OR

EXPLAIN THREE LEVEL DATA ABSTRACTION.

- Data Abstraction is a process of hiding unwanted or irrelevant details from the end user.
- There are 3 levels of data abstraction:
  - View level (end user level)
  - Conceptual level (logical level)
  - Physical level (internal level)



- **View level (external level)**
  - The external or user view is at the highest level of database system where only those portions of the database, which are of the concern to a user or application program is included.
  - It describes only part of the entire database that used by end user.
  - It is a highest level, which specify some part of database.
  - It Specify how data will represents to each users.
  - Each user can show a different view for specific data.
- **Conceptual level (logical level)**
  - It is a middle level of data abstractions.
  - It describes what data are stored in the database and what are relationships exist among those data.
  - It describes each entity with their relationship.
  - It specify table or entity along with its attributes & logical relations between entities.
  - This is the view of the application programmer who creates the software for interaction with the user, and database administrator who administers and maintains the database.
- **Physical level (internal level)**
  - It is a lowest level of data abstractions.
  - DBA works at this level.
  - It indicates how the data will be stored and describes the data structures, the file structures, the special storage architecture and the access methods to be used by the database.
  - It describe how the data is stored on the storage device.
  - It provides internal view of physical storage of data.
  - At this level, we can know where the data is stored in file, size of the file, memory as well as location of the file.

12. WHAT IS A DATABASE MODELS? EXPLAIN DIFFERENT TYPES OF DATA MODELS IN BRIEF.

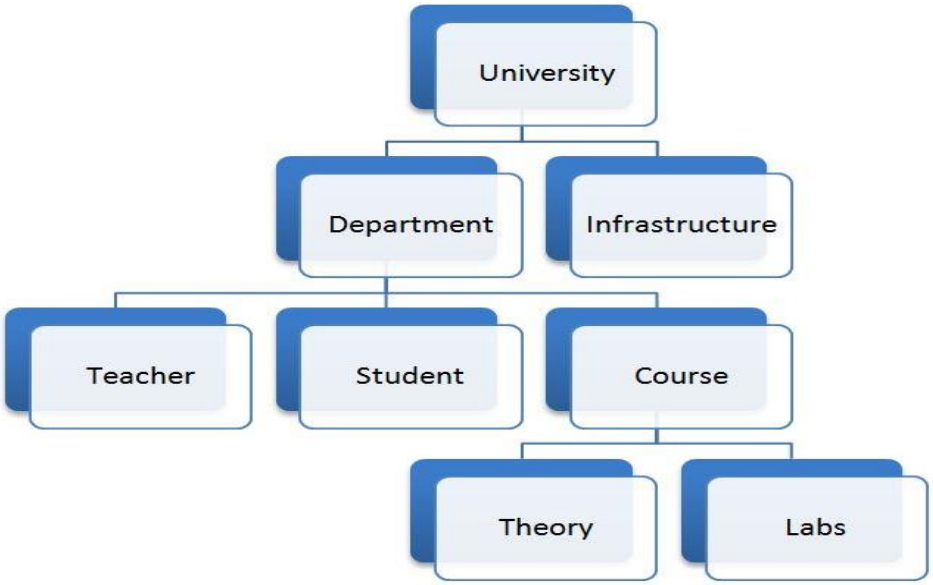
- A data model decides the method of storing data in database.
- Defines the logical structure of a database.
- When the data is stored in the database it needs to be stored in a particular format.
- Therefore, data model decide the structure to store data.
- Data modelling makes it easier for developers, data architects, business analysts, and other stakeholders to view and understand relationships among the data in a database or data warehouse.

Type of Database Models are:

- Hierarchical Model
- Network Model
- Entity-relationship Model
- Relational Model

13. SHORT NOTE: HIERARCHICAL MODEL.

- IBM developed it, in the 1960s.
- It uses **tree structure** to represent data.
- The hierarchy starts from the Root node and expands like a tree, adding child nodes to the parent nodes.
- A parent node contain one or more child node.
- Data are viewed as a collection of tables.
- Support one to one or one to many relationship.



Advantages:

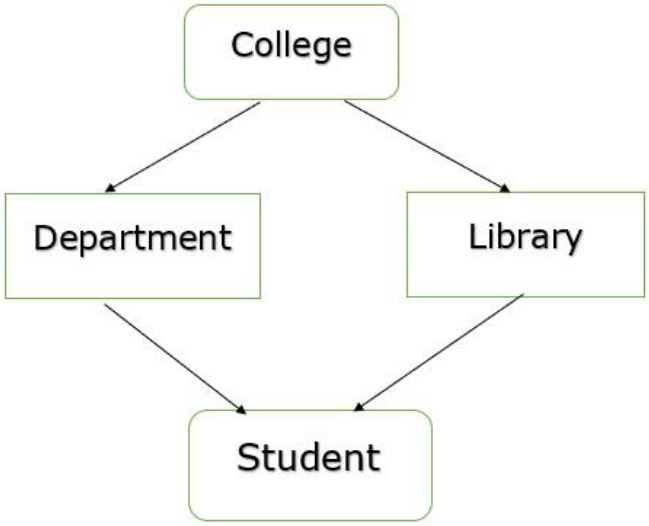
- Simple & Easy to understand.
- Data security.
- Maintain data integrity.

Disadvantages:

- Top to down traversal approach.
- Complex model.
- One parent per child allowed in hierarchical model.
- Does not support many too many relationships.

14. EXPLAIN NETWORK MODEL WITH ADVANTAGES & DISADVANTAGES.

- This is an extension of the Hierarchical model.
- In this model, data organized more like a graph.
- It allowed having more than one parent node.
- It is design to represent objects and their relationships flexibly.
- Each set contains one owner or parent record as well as one or more child or member records.



Advantages:

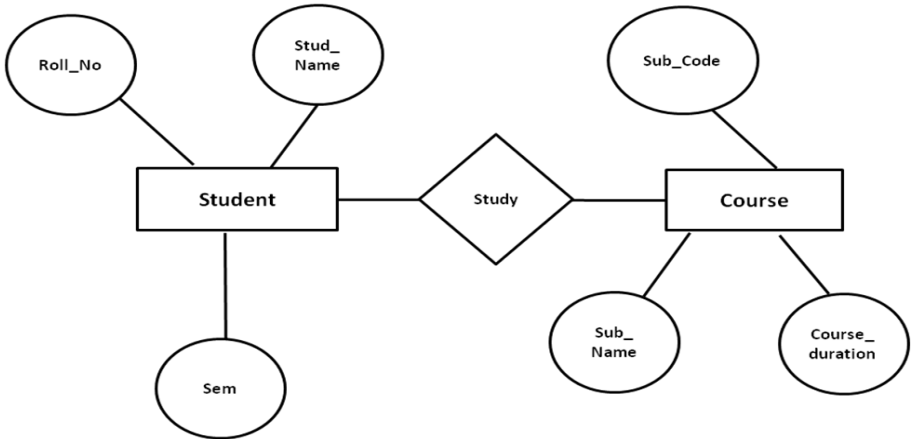
- Conceptually simple and easy to design.
- Data security
- Maintain data integrity
- Support many too many relationships.
- Data access is easier and flexible

Disadvantages:

- System complexity
- Lack of structural independence

15. WRITE A DETAIL NOTE ON ENTITY-RELATIONSHIP MODEL.

- Peter Chen firstly develops ER diagram in 1976.
- It is based on real-world entities & their relationships.
- Used to design a conceptual view of database.
- In terms of DBMS, an entity is a table or attribute of a table in database.



Advantages:

- Easy to understand
- Simple
- More specific to relational database modelling
- Good DBMS support
- Visual representation

Disadvantages:

- Limited expressiveness
- Can be ambiguous
- No standards: many versions that can be confusing
- Mostly for relational database only
- Limited constraint representation

16. EXPLAIN IN DETAIL: RELATIONAL MODEL

- This data model introduced in 1970's.
- Currently, it is consider as the most widely used data model.
- It represents the database as a collection of relations.
- A relation is nothing but a table of values.
- Data in the form of table
- Each table represents one entity.
- Each row of table represents instances of that entity.

Student ID	Name	Age
1	Aman	18
2	John	18
3	Karthik	17
4	Ravi	18

Subject ID	Name	Teacher
1	C	Mr. Kelvin
2	C++	Mrs. Kumar
3	Java	Mrs. Khan
4	Python	Mr. Sam

Student ID	Subject ID	Marks
1	1	86
1	2	92
2	1	81
2	2	86

Advantages:

- Structural independence
- Conceptual simplicity
- Design , implementation , maintenance and usage ease
- Query capability
- Limits redundancy
- Flexible
- Offers better data integrity

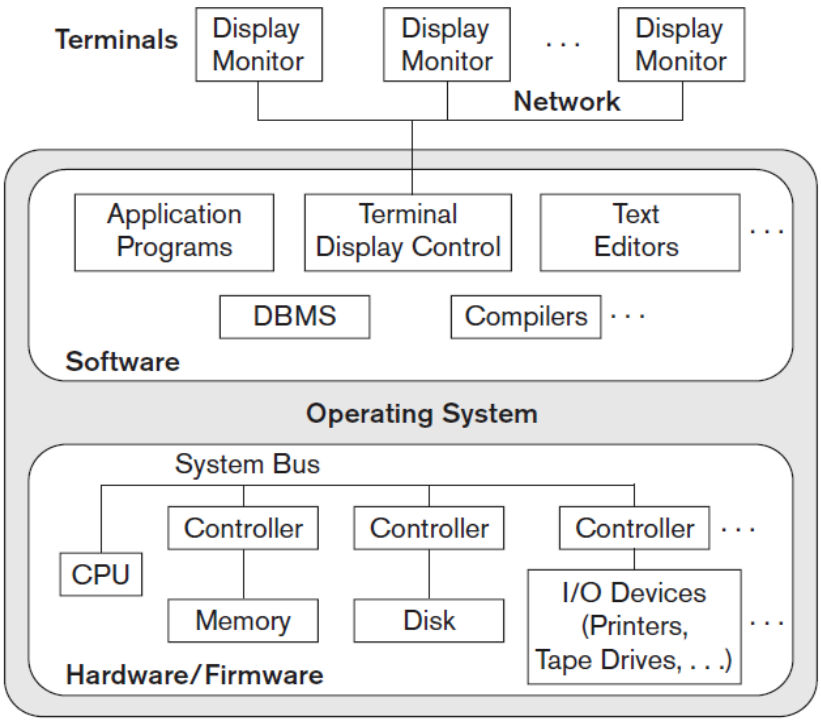
Disadvantages:

- Hardware Overhead
- Performance Issue
- Complex to understand when there is more number of tables.

17. EXPLAIN CENTRALIZED DATABASE ARCHITECTURE

Centralized DBMSs Architecture

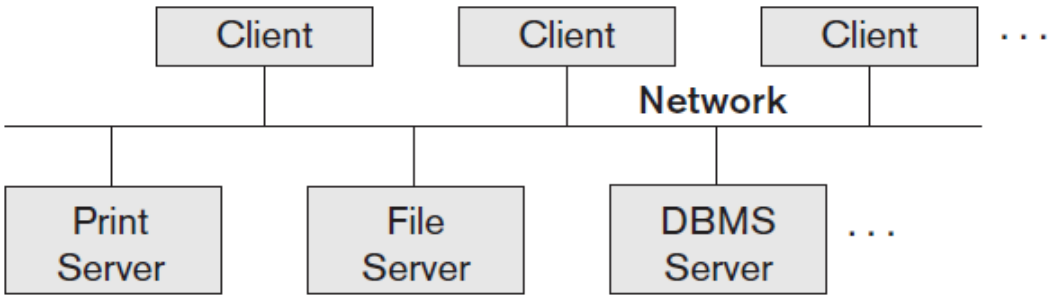
- Database systems first used computers similarly to display terminals, resulting in a centralized DBMS with full functionality, application program execution, and user interface processing on one machine.
- The following figure illustrates the physical components in a centralized architecture.
- Gradually, DBMS systems started to exploit the available processing power at the user side, which led to client/server DBMS architectures.



Centralized DBMS Architecture

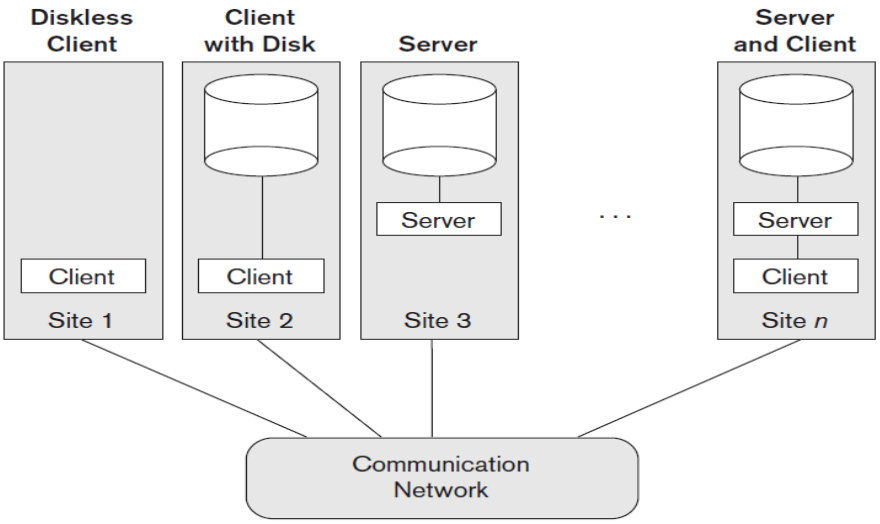
Basic Client/Server Architectures:

- The client/server architecture designed for networks with many PCs, workstations, file servers, printers, database servers, Web servers, and email servers.
- The idea is to define specialized servers with specific functionalities.
- For instance, multiple PCs or small workstations can be connected to a file server to store their files.
- Another machine can be designated as a printer server by being connected to various printers; all print requests by the clients are forwarded to this machine. Web servers or e-mail servers also fall into the specialized server category.
- The resources provided by specialized servers can be accessed by many client machines.
- The client machines provide the user with the appropriate interfaces to utilize these servers, as well as with local processing power to run local applications.



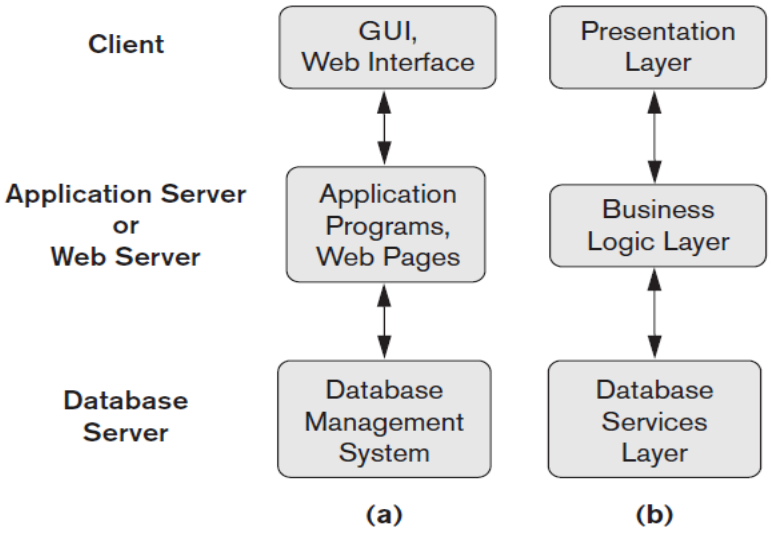
Two-Tier Client/Server Architectures for DBMSs

- Many relational database management systems (RDBMSs) started as centralised systems, with the user interface and application applications transferred to the client side initially.
- SQL established a logical divide between client and server by standardizing RDBMS languages.
- These designs are two-tier because software components are distributed over client and server platforms.
- This architecture is simple and compatible with existing systems.



Three-Tier and n-Tier Architectures for Web Applications

- Many Web applications use an architecture called the three-tier architecture, which adds an intermediate layer between the client and the database server, as shown in the following figure:



- The middle tier is called the application server or Web server, depending on the application.
- Application program and business rules (procedures or restrictions) run on this server to access database server data.
- It can also secure databases by validating client credentials before sending requests to the database server.
- Clients have browsers and UIs.