Data:- Data is nothing but a known value of row fact which is in the form of letters, digits, or special characters.

Ex:- 1, ravi,… ..... etc

Database:- Database is a collection of organized data which is stored in computer.

Ex:-

1 sone 60

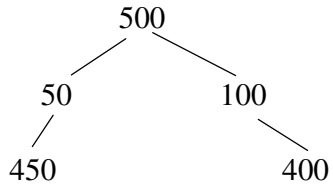
2 stwo 70 student data

DBMS:- DBMS is a software designed for manipulating (storing, retrieving, updating, deleting) the data.

Ex:- oracle, Mysql, DBq, etc

Draw backs of file systems (Advantages of DBMS):-

1. **Data redundancy and inconsistency:-** Redundancy means duplication of data. Some data present in different files in the computer system may cause data redundancy. During updates if we forget the updation of a file, while retrieving the data we may get inconsistent results. This problem can be easily reduced in DBMS.
2. **Difficulty in accessing data:-** In conventional file systems to access the data, we have to write so many number of application programs (written in c, c++, java, etc) for storing retrieving and updating the data effectively.
3. **Data isolation:**-In file systems, the data will be stored in different file formats (.txt, .html, .war, etc). so we have to write application programs to retrieve data from each file format. But in databases the data is stored at one place, so we can access data easily.
4. **Integrity constraints problems:**- Constraint means defining a condition on data. Handling the constraint at application programming level is very difficult. Maintaining integrity constraints in DBMS is very easy.
   Ex:- to define a condition that marks of a student must be less than 100. We have to write application program in file system but in DBMS easily we can implement it.
5. **Atomicity problems:-** During the transactions, there may be the possibility of failing of computer system. In that case the transactions should not be happen, they should be rolled back. Such type of mechanism not there in file systems. This will be effectively done in DBMS.
   Ex:-Suppose an amount of 100rs transferring from account A to B in an application program. If suppose power gone after deducting 100 from A. then that 100rs will not be added to B. so that will cause inconsistency. To avoid this, the transaction should be executed fully or should not be executed at all. i.e. no partial executions of the transactions. This is not maintained strictly in DBMS.
6. **Concurrent-Access problems:-** when multiple users wanted to update data simultaneously, we may get inconsistent data in file systems.
   Ex:- consider an account A=500rs. If two customers want to withdraw 50rs and 100rs simultaneously. Two application programs for two with drawls will give the following.



We got 450rs and 400rs but the actual result is 350rs to maintain constraint results we got for DBMS.

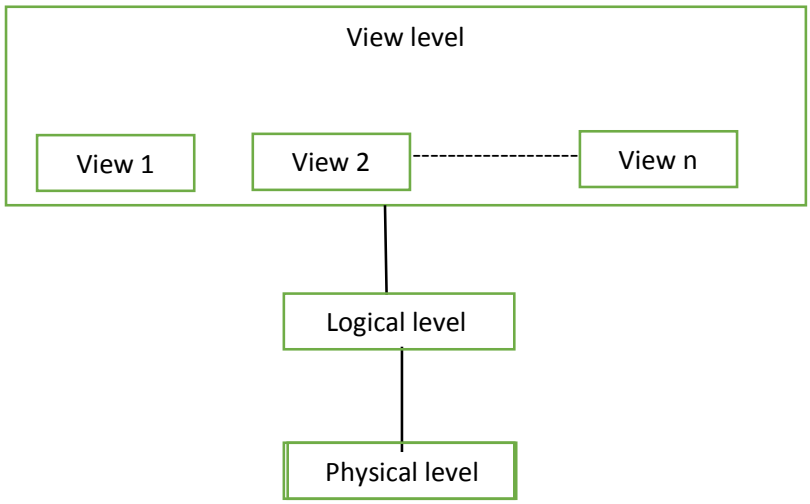
7. **Security problems:-** All the data should not be accessed by all the users. One user must access his permitted data. Maintaining these restrictions in file system is very difficult. We can maintain better security using DBMS.

3-tier architecture of DBMS:-

Data abstraction:- Abstraction is nothing but hiding complex details and exposing the essential things. Database developers hide internal complexity through several levels of abstraction.

There are three levels of abstraction

1. Physical or Internal level
2. Logical or Conceptual level
3. View or external level



Physical level:-This is the lowest level of abstraction. This level describes "How the data is actually stored in the database". This level involves complex structures to get efficient access of data. In this level complexity is high and abstraction is low.

Logical level:- This is the intermediate level of abstraction. This level describes "what data is stored in the database" and "what are the relationships among the data". Database administrator use this level to describe what information is kept in database. This level describes simper structures than physical level. The user of this level need not know the complexities of physical level.

View level:- This is the highest level of abstraction. At this level there may be many views defined for different users. A view for certain group of users describes. "What portion of data is to be made visible" to that particular group.

The main goal of this level is to provide user friendly with the system. The user of this level does not have to know the complexities at the conceptual and physical level.

Data independence:- The ability of DBMS to modify its own schema definition at one level, without affecting the schema definition of next higher level is known as data independence.

There are two levels of data independence

Physical data independence:- It is ability of DBMS to modify the physical schema without causing any changes in the schema at logical level and at the view level. Generally modifications to the physical level are done to improve system performance.

Logical data independence:- It is the ability of DBMS to modify the logical schema without causing any changes in the application programs in the view level. Achieving logical data independency is much more difficult than physical data independence because the application programs highly dependent on the logical structure of data.

Instance:- The collection of information stored in the database at a particular point of time is called instance.

Schema:- The design of database at different levels is called schema.

Database users:-

- For a small personal database one person defines, constructs and manipulates the database and there is no sharing.
- In large organizations many people involved in design, use and maintenance of a large database with hundred of users.
- We have two kinds of database users
    1. **Actors on the scene**
    2. **Workers behind the scene**

Actors on the scene:- These people involves the day-to-day use of a large database.

We have four categories:-

1. **Database administrators (DBA):-** The responsibility of DBA is to administrate the resource of databases. He is responsible for authorizing access the database. He will coordinate and monitor the database and acquire software and hardware resources needed. He is responsible for maintaining database security.
2. **Database designers:**- These are responsible for identifying the data is to be stored in databases and choosing appropriate structures to represent and store this data. These are responsible to communicate database users to know their requirements.
3. **End uses:**- These are the people who need access to database for querying, updating and generating reports.
   The database exists primarily for their use only.
   There are four categories of end uses

a) Casual end user
b) Naive or parametric end user
c) Sophisticated end user
d) Standalone end user

a. **Casual end user:-** These users occasionally access the database, but they may need different information each time. These users use a sophisticated database query language to specify their requests.
Ex: through browsers
b. **Naive or parametric end users:-** These users constantly querying and updating the database.
Ex:- 1. Bank account holders continuously checking the balances
2. Reservation agents continuously checking the tickets availability.
c. **Sophisticated end user:-** These include scientists business analysts who use DBMS to implement their own applications to meet their complex requirements.
d. **Standalone end users:**- These users maintain personal database by using ready-made programming packages that provide easy-to-use and menu based or graphical based interface.
Ex:- user of tax package stores a variety of personal financial data for tax purpose.

4. **System analyst and application programmers:-** system analyst determine the requirements or specifications of end users especially naive or parametric end users. Application programmers implement these specifications as programs. Then they test, debug, document and update these programs.

Workers behind the scene:- These are the people who are not interested in the database content itself.

There are three categories:-

1. **Database system designers and implementers:-** These people will design and implement the DBMS modules and interfaces as a software package. These will implement the modules including the implementation of catalog, query processing, interface processing, buffering data, controlling concurrency, recovery and security.
2. **Tool developers:-** These people will implement the tools that facilitate database modelling and design, database system design, and improved performance. Tools are optional packages that are often purchased separately.
3. **Operators and maintenance personnel:**- These are responsible for actual running and maintenance of the hardware and software environment for the database system.

Applications of DBMS:- Applications where we use database management systems are:-

1. **Banking:-** In banking systems DBMS is used for storing customer information, tracking day-to-day credit and debit transactions and generating bank statements etc……
2. **Online shopping:-**Online websites like amazon, flip kart, use DBMS to store product information, addressing and preferences credit details and provide us for the list of products based on our query.

3. **Airline:-** In airline reservation system, DBMS is used to keep records of flights arrival, departure and delay status.
4. **Manufacturing:-**In manufacturing, DBMS is used to keep track of records of all the details about the products like quantity, bills, purchase, supply management etc
5. **Human resource management:-** In big organizations there are many workers working. Human resource management keep records of each employee's salary, tax and work through DBMS.
6. **Telecom:-**In telecom there is a need of DBMS to keep track of information regarding calls mode, network usage, customer details, etc
7. **Education system:-**Database systems are used frequently in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, fee details, etc
8. **Railways:-** Database is required in railways to keep record of ticket booking, train departure and arrival status and keep track of trains that are getting late etc.
9. **Library management system:-**These are thousands of books in a library and it is very difficult to keep record of all the books information's in a register. Using DBMS we can maintain all the information related issue dates, names of books, author, availability, of the book etc.
10. **Social media:-**Daily thousands, millions of users signed up for the social media sites like face book, twitter, etc. All the information of these users are stored and maintained by using databases only.
11. **Military:-** Military keeps the records of millions of solders information and millions of files using DBMS keeping the data safe and secure.

Data models:-

A data model defines the structure of a database.

Basically there are two types of data models.

1. Object based logical models
   - **a)** Entity-relationship model (E-R model)
   - **b)** Object oriented model
2. Record based logical models
   - **a)** Hierarchical model
   - **b)** Network model
   - **c)** Relational model

Entity relationship model (E-R model):- This model defines the stricture of database as the collection of entities.

Basic notations in E-R model:-

Entity:- An entity is nothing but a 'thing' or object in the real world that is distinguishable from all the other objects.

Ex:- a book, person, etc.

An entity will have a set of properties to represent it, known as attributes.

Ex:- Account entity may have attributes accno, aname, etc.

Entity set:- It is defined as set of same type of entities that share same attributes.

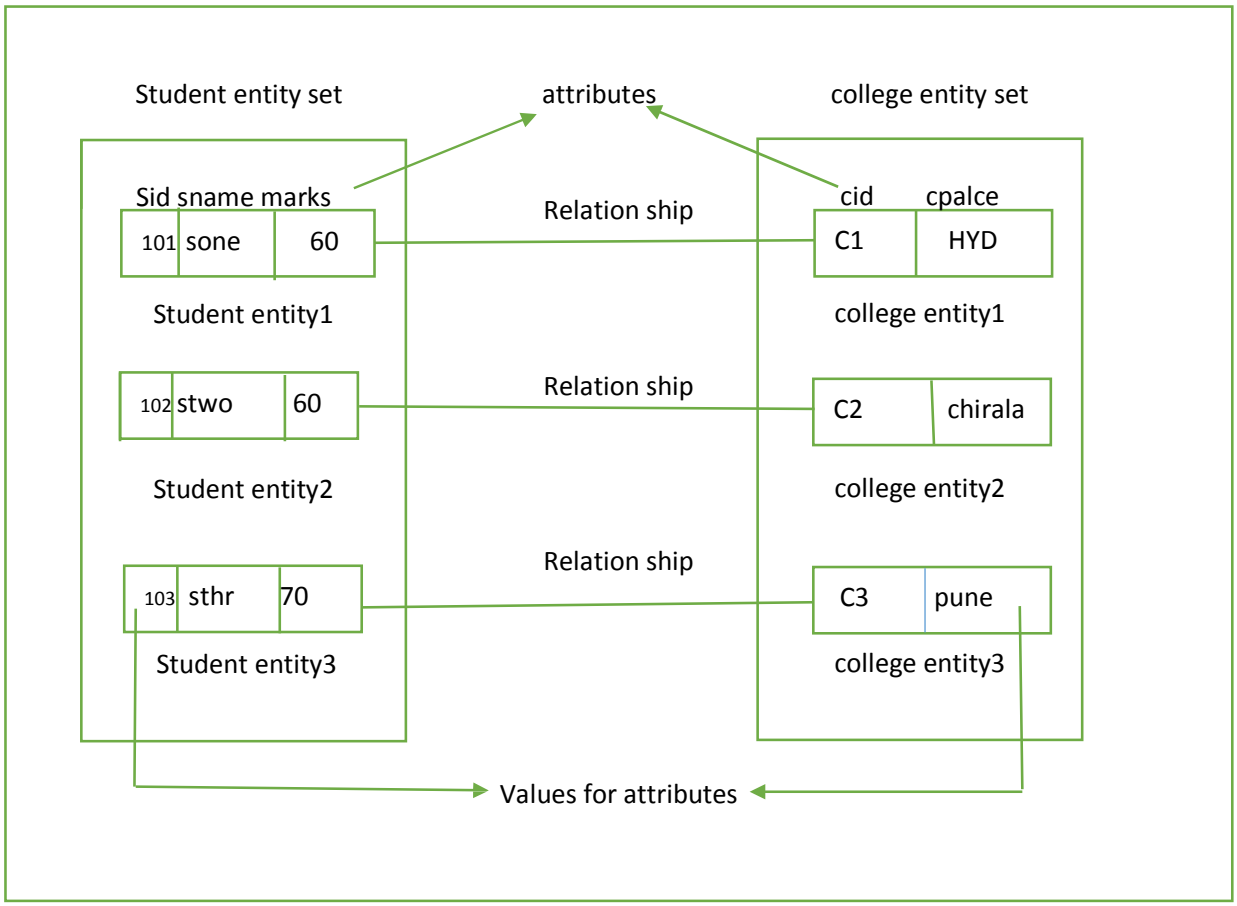
Ex:- set of all customers in a bank.

Set of all students in a college.

Relationship:- It is an association among the entities of two entity set.

Relationship set:- It is defined as set if relationships of same type.

Participation:- It is association among entity set.



Relationship set join={101->c1, 102->c2,103->c3}

Domain:- For each attribute there is a set of permitted values called domain or value set.

In the above diagram for the sid attribute the domain is {101,102,103}

Types of attributes:- There are several types of attributes in E-R model.

1. **Simple and composite attributes:-** A simple attribute cannot be divided into sub attributes where as composite attribute can be divided into sub attributes.
   Ex:- Rollno or regno ➷ simple because cannot ne divided into sub attributes or subparts.
   Name ➷ composite can be divided into sub attributes like first name, middle name, last name.
2. **Single and multivalued attributes:-** A singled value attribute can hold only one value, where as multi valued attribute can hold more than one value,
   Ex:- rollno➷ single valued, can hold only one value.
   Phno ➷ multi valued can hold more than one value.
3. **Derived attributes:-** These are derived from the values of other related attributes.
   Ex:- Age can be derived from the date attribute.

diagram notations:-


Rectangle ➷ represents on entity set.


Diamond ➷ represents relationship set.


Ellipse ➷ represents the attribute.


Double ellipse ➷represents the multivalued attribute.


Double rectangle ➷ weak entity set.

Line ➷ represents association


Double diamond➷ identifying relationship set.

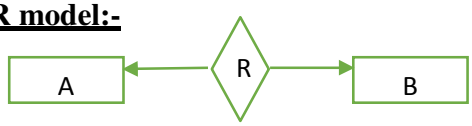

Double line ➷ represents total participation.


Triangle➷ represents ‘is a ‘ relationship.

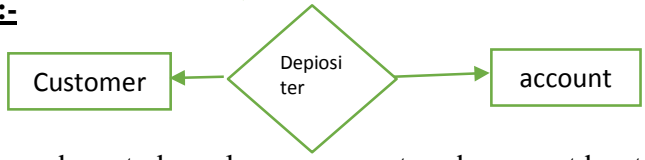
Mapping cardinalities:- For a binary relationship set R between two entity sets A and B, the mapping cardinalities can be one of the following.

1. **One-to-one:-** One entity of A can be associated with at most one entity of B and one more entity of B can be associated with at most one entity of A.

   E-R model:-



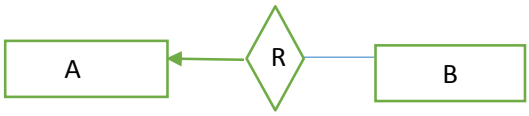
   Ex:-



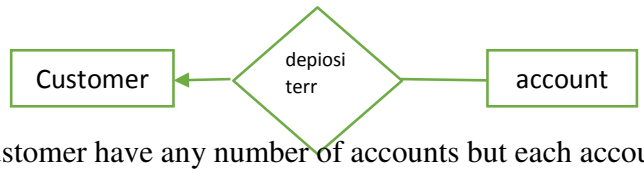
Each customer have to be only one account each account has to be single.

2. **One-to-many:-** one entity of A can be associated with any number of entities of B and one entity of B can be associated with at most one entity of A.

   E-R model:-



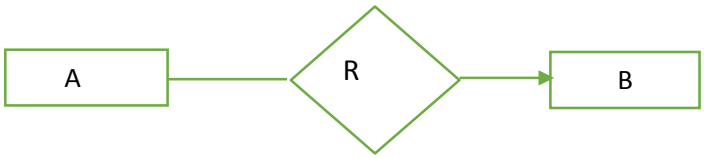
   Ex:-



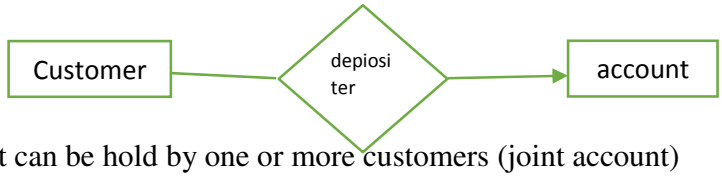
   A customer have any number of accounts but each account has to be single.

3. **Many-to-one:-** One entity of A can be associated with at-most one entity of B and one entity of B can be associated with any number of entities of A.

   E-R model:-



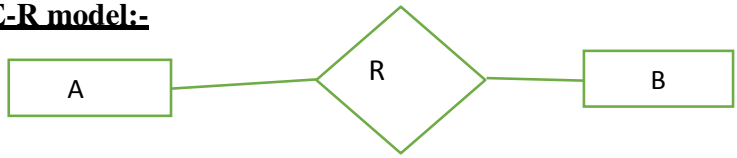
   Ex:-



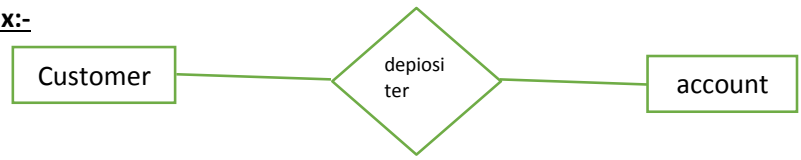
Each account can be hold by one or more customers (joint account)

4. **Many-to-many:-** One entity of A can be associated with any number of entities of B and one entity of B can be associated with any number of entities of A.

   E-R model:-

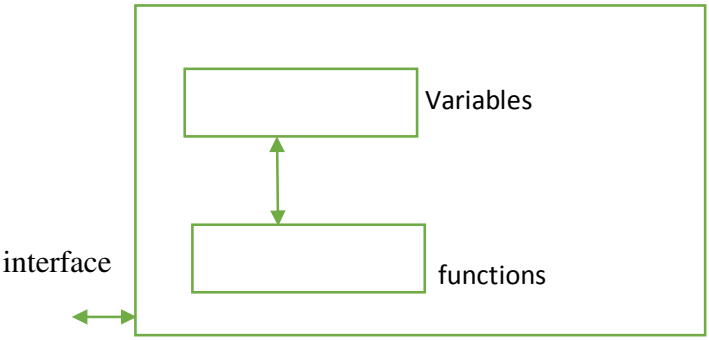


Ex:-



Each customer can have number of accounts and each account can be joint account.

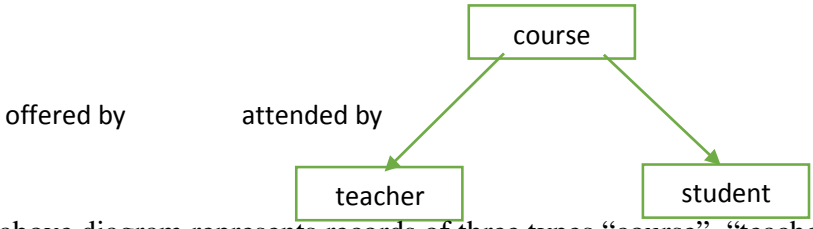
Object-oriented model:- Like E-R model, this model also models a database as a collection of objects. An object body encapsulates data(variables) as well as methods(functions) to manipulate the data. The objects that contain same type of data and same type of functions are generated together a class. So class is a type definition for proposed objects. The only way an object 'A' can access the data items of another object 'B' is by invoking the methods of 'B'. a can reach this by making method calls to 'B' through B's interface.



The structure of object oriented database is method as a set of classes and database will comprise objects belonging to those classes.

Record based logical models:- These models describes data at the logical level. In these models data is stored as collection of records of different types. Each record type can have a fixed number of fields and each field is usually of fixed length.

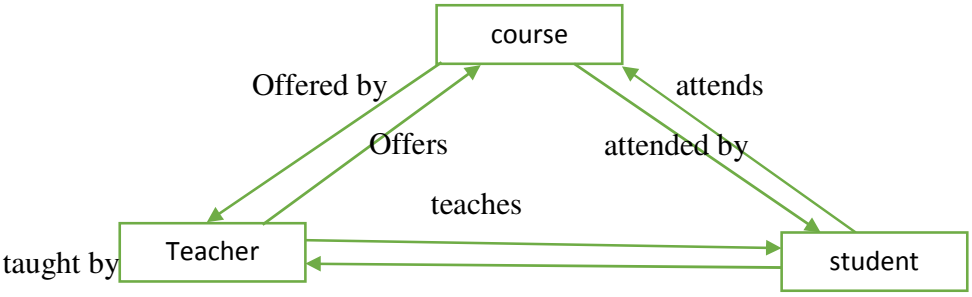
1. **Hierarchical model:-** This is the oldest model. In this model data is represented in the form of records. These records are organized as collection of trees. The relationships among the records are represented by links, which can be viewed as pointers. Tree structure permits that each record can have only one parent record i.e it permits only one-to-many relationship among the records.



above diagram represents records of three types "course", "teacher", and "student". The relationship "offered by" from "course" to "teacher" indicating faculty offering a course and the relationship. "Attended by" from "course" to "student" –indicating the student attending a course. It does not include the relationships "what are different courses being offered by a

faculty", "what are the different course being attended by a student", "who are the students being taught by a faculty" and "who are the faculty teaching a student". This is because of one-to-many representation.

2. **Network model:-**Like the hierarchical model, this model also models a database as a collection of records. These records are organized as a collection of arbitrary graphs (or networks). Thus a record can have any number of parent records and supports many-to-many relationships among the records. The relationships among the records are represented as links(pointers).



so we can represent all possible relationship in network model.

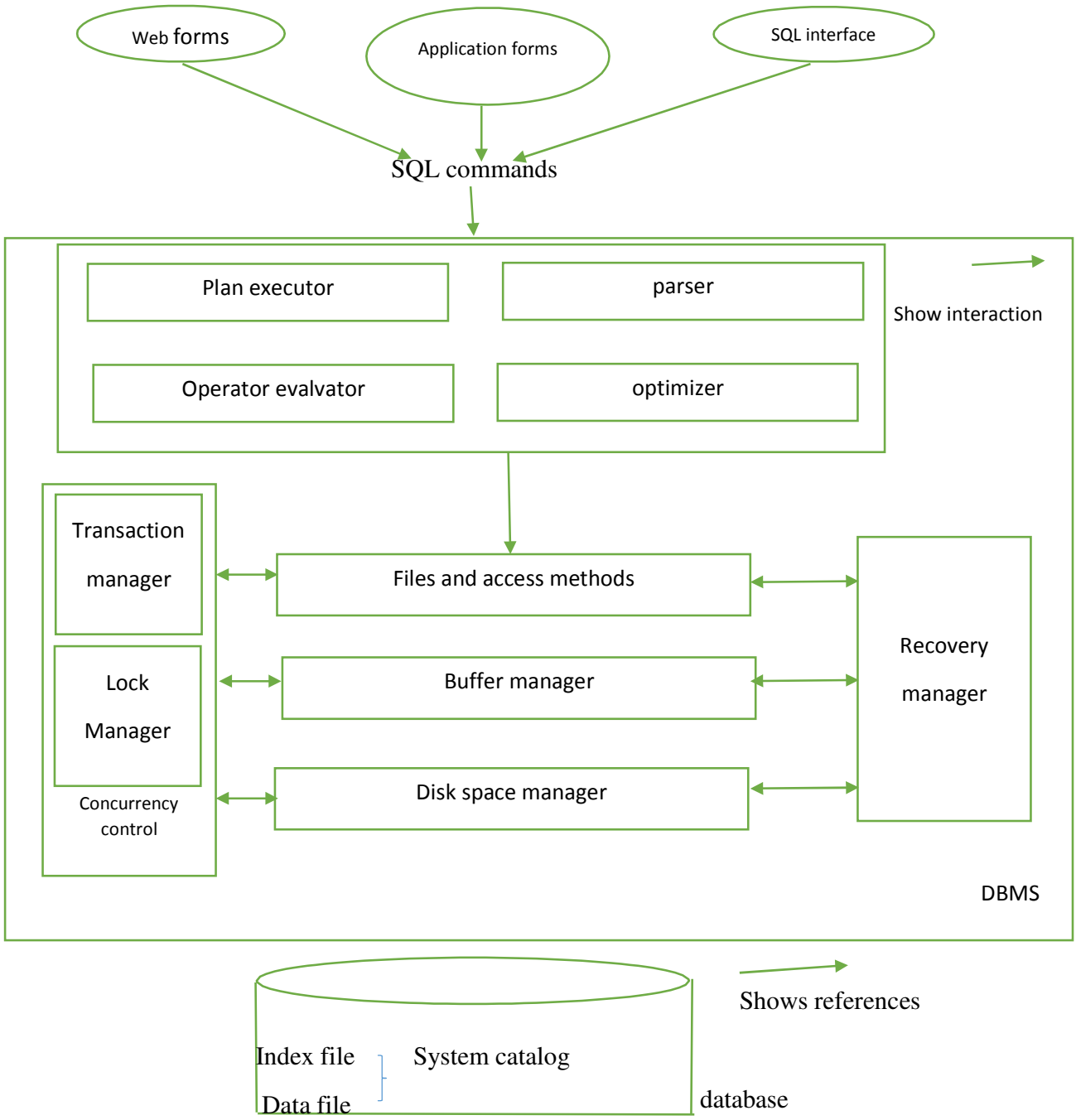
3. **Relational model:-**This is the modern and commonly used model among the record based models. This model models the database as a collection of tables. Each table is called a relation and has a unique name. each relation has a number of columns and unlimited number of rows and each row represents and instance of the relation. Row is also termed as a 'tuple'. The relationships among the tables can be defined by primary key and foreign key relationship.

Studentcolumn rows

sid	Sname	marks
1001	Sone	70
1002	Stwo	60
1003	Sthr	90
1004	Sfour	60

Database system structure:-

Unsophisticated users(customers, travel agents, etc) sophisticated users application programmers, DB admins



- The above diagram shows the structure of DBMS based on the data model.
- The DBMS accepts SQL commands generated from variety of user interfaces and produces query evaluation planes, executes these planes against the database and returns the results.
- When a user issues a query, the parsed query is presented to a query optimizer, which uses the information about how data is stored to produce an efficient execution plan for evaluating the query.

- An execution plan is a blue print for evaluating a query and it is represented as a tree of relational operators. Relational operators some as the building blocks for revaluating queries.
- The code that implements relational operators sits on top of the file and access methods layer. This layer supports the concept of a file, which in a DBMS is a collection of pages or collection of records.
- The files and access methods layer code sits on top of the buffer manager which bring pages from disk to main memory based on the requests.
- The lowest layer of DBMS software deals with the management of space on disk, where the data is stored. The higher layers allocate, deallocate, read and write records through this layer, called disk space manager.
- Transaction manager layer responsible for transactions request and release locks according to a suitable locking protocols and schedules.
- The lock manager layer keeps track of requests for lock and granting locks to eligible transactions.
- Recovery manager is responsible for maintaining logs and restoring the system to a consistent state after a crash.
- The disk manager, buffer manager and file and access methods layer must interact with transaction manager, lock manager and recovery manager.
- As part of system catalog database there are two files: index files, which maintains different types of indexes to the keys to the increase the accessibility and Data files which stores the actual data of databases.

Database administrators (DBA):- The responsibility of DBA is to administrate the resource of databases. He is responsible for authorizing access the database. He will coordinate and monitor the database and acquire software and hardware resources needed. He is responsible for maintaining database security.

Responsibilities of DBA(Data Base Administrator):-

- Creates and maintain all databases required for development and testing.
- Performs ongoing tuning of the database instances.
- Install new versions of the oracle DBMS and its tools and any other tools that access the oracle database.
- Planes and implements backup and recovery the oracle database.
- Controls migrations of programs, database changes, reference data changes through the development life cycle.
- Implement and enforces security for all the oracle databases.
- Performs databases re-organizations to assist performance.
- Evaluates releases of oracle and its tools.
- Provides technical support to application development team.
- Enforces and maintains database constraints to ensure integrity of the database.

- Administrators all database objects, including tables, views, indexes, clusters, packages, and procedures.
- Troubleshoots with problem regarding the databases, application and development tools.
- Create new databases users as required.
- Manage sharing of resources amongst applications.
- DBA has ultimate responsibility for the physical database design.

Centralized database architecture:-

- A centralized database is a database that is located, stored and maintained in a single location. This location is often a central computer or a database system. i.e for ex server CPU or a mainframe computer.
- A centralized database is used by an organization or an institution.
- Users access a centralized database through a computer network through which user can access the central CPU.

Advantages:-

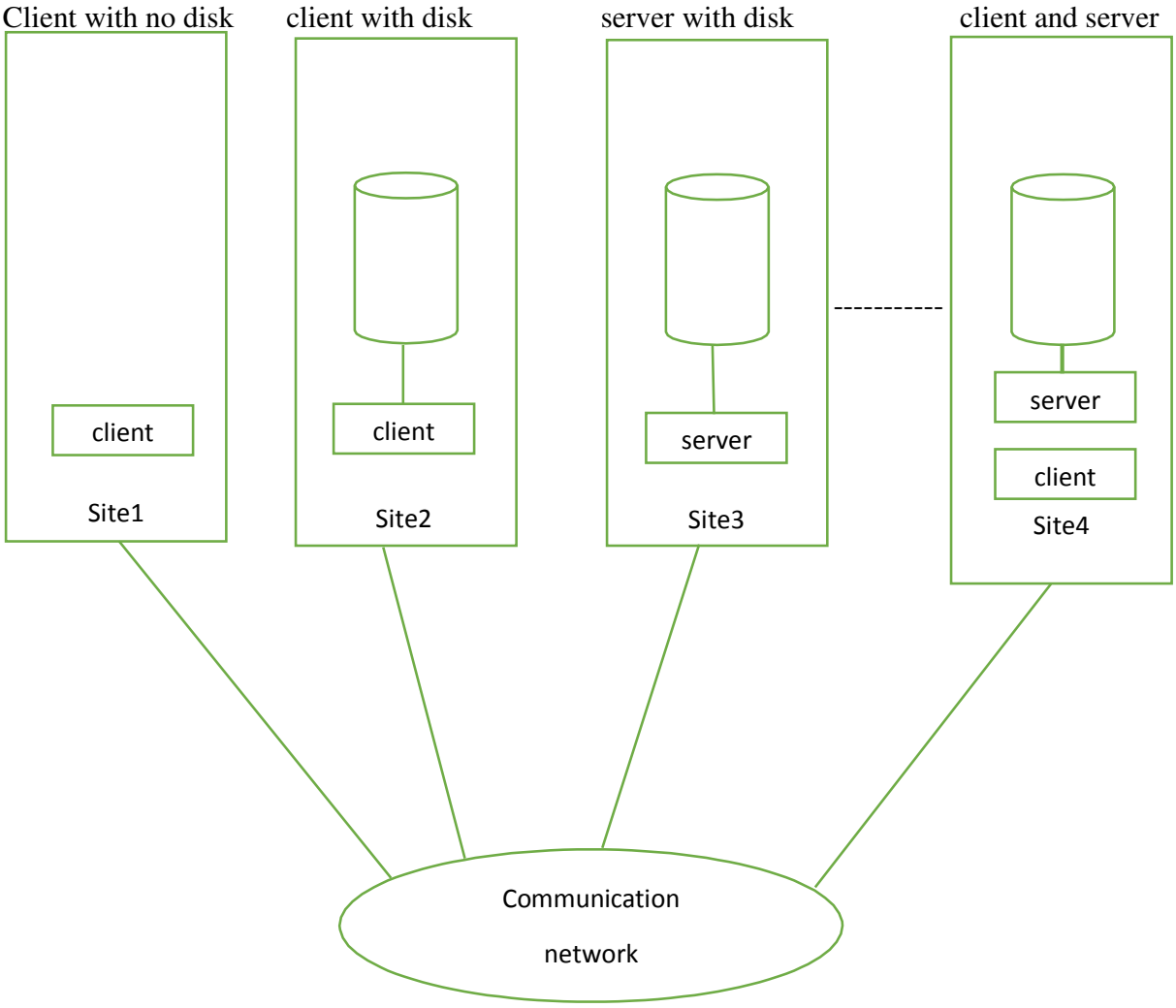
- Data integrity is maximized and redundancy is minimized.
- High data security, because data is stored at one location.
- Easier data portability and data administration.
- Data updates are immediately received by every end user.

Disadvantages:-

- Centralized databases are highly dependent on the network connectivity. If internet connection is low database access time will increase.
- Bottlenecks can occur as a result of high traffic.
- If hardware failure occurs, all the data within the database will be lost.
- Communication costs are expensive.

- If control site DB goes down every system is blocked from using the centralized system until it get recovered.

2-tire client-server architecture:-



- Client server architecture has a two logical components client and server.
- Generally client is a personal computer or work stations. Server is large-work stations or mini range computers or mainframes.
- Applications of DBMS run on one or more clients and the DBMS software resides on the server.

- These server and client computers will be connected through a network.

Advantages:-

1. Less expensive.
2. It is flexible as compared to centralized system.

Disadvantages:-

1. Programming cost is very high.
2. There may be lack of management tools.

Demerits of DBMS:

1. Cost

DBMS requires high initial investment for hardware, software and trained staff. A significant investment based upon size and functionality of organization if required. Also organization has to pay concurrent annual maintenance cost.

2. Complexity

A DBMS fulfill lots of requirement and it solves many problems related to database. But all these functionality has made DBMS extremely complex software. Developer, designer, DBA and End user of database must have complete skills if they want to user it properly. If they don't understand this complex system then it may cause loss of data or database failure.

3. Technical staff requirement

Any organization have many employees working for it and they can perform many others tasks too that are not in their domain but it is not easy for them to work on DBMS. A team of technical staff is required who understand DBMS and company have to pay handsome salary to them too.

4. Database Failure

As we know that in DBMS, all the files are stored in single database so chances of database failure become more. Any accidental failure of component may cause loss of valuable data. This is really a big question mark for big firms.

5. Extra Cost of Hardware

A DBMS requires disk storage for the data and sometimes you need to purchase extra space to store your data. Also sometimes you need to a dedicated machine for better performance of database. These machines and storage space increase extra costs of hardware.

6. **Size**

As DBMS becomes big software due to its functionalities so it requires lots of space and memory to run its application efficiently. It gains bigger size as data is fed in it.

7. **Cost of Data Conversion**

Data conversion may require at any time and organization has to take this step. It is unbelievable that data conversion cost is more than the costs of DBMS hardware and machine combined. Trained staff is needed to convert data to new system. It is a key reason that most of the organizations are still working on their old DBMS due to high cost of data conversion.

8. **Currency Maintenance**

As new threats comes daily, so DBMS requires to updates itself daily. DBMS should be updates according to the current scenario.

9. **Performance**

Traditional files system was very good for small organizations as they give splendid performance. But DBMS gives poor performance for small scale firms as its speed is slow.

Meta data: It is defined as data about the data. In Databases, Metadata defines data elements and attributes (Name, data type, size, etc), data could be registered about structures and records as well (Length, columns and fields). This is **extremely helpful for the reliability** of databases and their efficiency.

Ex:

In a library, for example, the data is the content of the titles, and the Metadata is about the title, the author, and description of the content, the physical location and the date of publication.

Database catalog: The database catalog of a database instance consists of metadata in which definitions of database objects such as base tables, views (virtual tables), synonyms, value ranges, indexes, users, and user groups are stored.