- File System:

The file system is typically described as various files and numbers of different application programs are written to extract records from and add records to the appropriate file.

- Each program defines and manges it's own data.

-Limitations of file-Based approach.

1. **separation and isolation of data.**
   - Each program maintains its own set of data.
   - Users of one program may be unaware of potentially useful data held by other programs.
2. **Duplication of data**
   - Some data is held by different programs.
   - Wasted space and potentially different formats of the same item.
3. **Data dependency**
   - File structureis defined in the program code.
   - Definition of data was embedded in application programs, rather than being stored separately and independently.
4. **Incompatible file formats**
   - Programs are written in different languages, and so conn't easily access each other's files.

Database approach:

Main terms:

-Data:known facts that can be recorded and have an implicit meaning. There are different types of data:

- Numbers, string, date, time, Text, picture, audio, video, graphics…

-Information: Information is created when data is presented in a way that has meaning to the recipient.

-Data versus Information

Data itself has no meaning without meta data which describe data.

Ex: 0780000000 is a data, doesn't give us any information, but knowing the meta data that explains data: phone number.

Data is what you store in the database, while **information** is what you retrieve from the database.

- Database:

A database is one of the essential components for many applications and is used for storing a series of data in a single set. In other words, it is a group/package of information that is put in order so that it can be easily accessed, manage, and update.

-There are different types of databases. They are:

- full-text
- numeric
- images

In a database, even the smallest portion of information becomes the data. For example, a Student is a data, a roll number is a data, and the address is data, height, weight, marks everything is data. In brief, all the living and non-living objects in this world are data. In this chapter of the database, you will learn about the fundamental terminologies that are used in DBMS.

-Database Advantages:

- Minimum data redundancy.
- Improved data security.
- Increased consistency.
- Lower updating errors.
- Reduced costs of data entry, data storage, and data retrieval.
- Improved data access using host and query languages.
- Higher data integrity from application programs.

-Database Disadvantages:

- Database systems are complex, difficult, and time-consuming to design.
- Substantial hardware and software start-up costs.
- Damage to database affects virtually all applications programs.
- Extensive conversion costs in moving form a file-based system to a database system.
- Initial training required for all programmers and users.

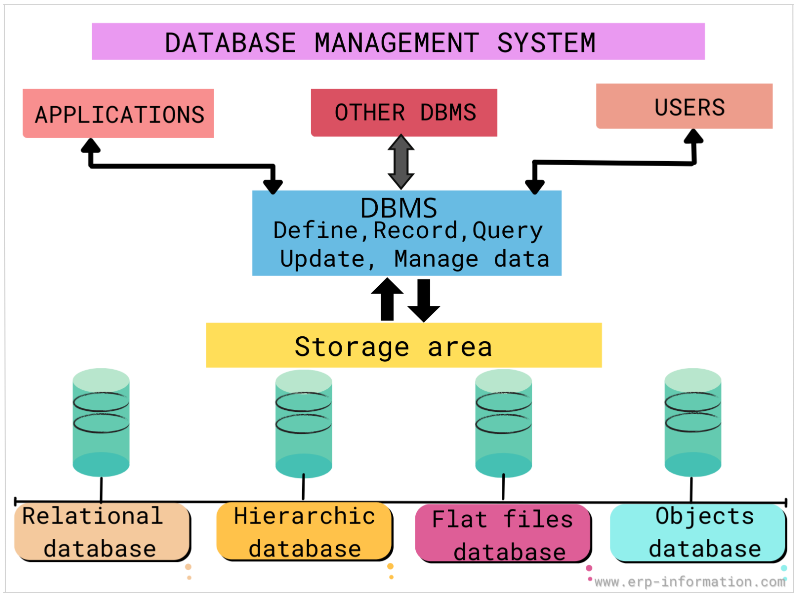
-Database Environment:

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 the database, and the data also.

- The hardware in a database environment means the computers and computer peripherals that are being used to manage a database,
- The software means the whole thing right from the operating system (OS) to the application programs that include database management software like M.S. Access, SQL Server, MySQL, Oracle..etc.
- The people in a database environment include those people who administrate and use the system.
- The techniques are the rules, concepts, and instructions given to both the people and the software along with the data with the group of facts and information positioned within the database environment.

-Database Mangement System (DBMS):

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. The DBMS essentially serves as an interface between databases and users or application programs, ensuring that data is consistently organized and remains easily accessible.



Architecture of database:

-Data Abstraction

One fundamental characteristic of the database approach is that it provides some level of data abstraction.

The major purpose of DB system is to provide users with an abstract view of the data ,that is the system hides certain details of how the data is stored and maintained. However, in order for the system to be usable ,data must be retrieved efficiently. This concern has led to the design of complex data structure for the representation of data in the DB since many DBS users are not computer trained ,the complexity is hidden from them through several levels of abstraction in order to simplify their interaction with the system.

- Database levels(Three-level Architecture) :

The goal of the three schema architecture,is to separate the user applications from the physical database. In the following are the three levels:

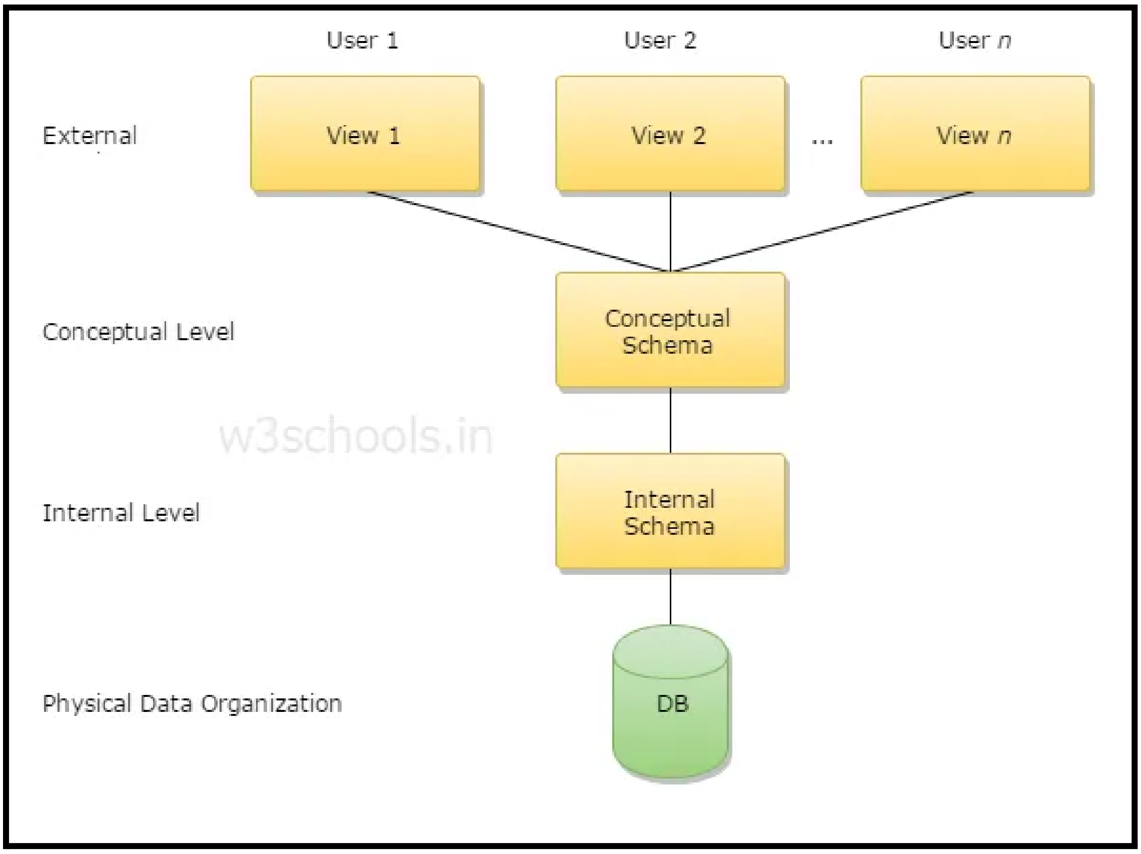
1-External level (view level)-User:

The highest level of abstraction,it's the way users recognize the data. It describes the portion of a database relevant to a particular user or a group of users while hiding the rest of database.

2- Conceptual Level(Logical level) -database administrator:

It describes the structure of the whole database, it describes **what** data are actually stored in the database. This illustrates the entities, attributes with their data types and constraints, user operations and relationships.

3- Internal level(Physical level):The lowest level of abstraction describes **how** the data are actually stored in details, it’s the way the DBMS and the operating system distinguish the data. It contains the definition of stored records, the methods of the representation, the data fields and indexes.



DB levels

-Schema(intention): Is a description of a particular collection of data, using the given data model. For example the following database where the (ID,Name and grade are the description of data ,while (291,Noor,20,321,Ali,22) are the data :

ID	Name	Grade
291	Noor	20
321	Ali	22

The schema is rarely changes, when we define a new database, we specify it's schema.the structure, data types, and constraints that describes that database.

-Instance(data state,extension): The actual data in the database at any point of time. It changes rapidly.

- When we initially load data into the database, it is said to move into the initial state of the database.

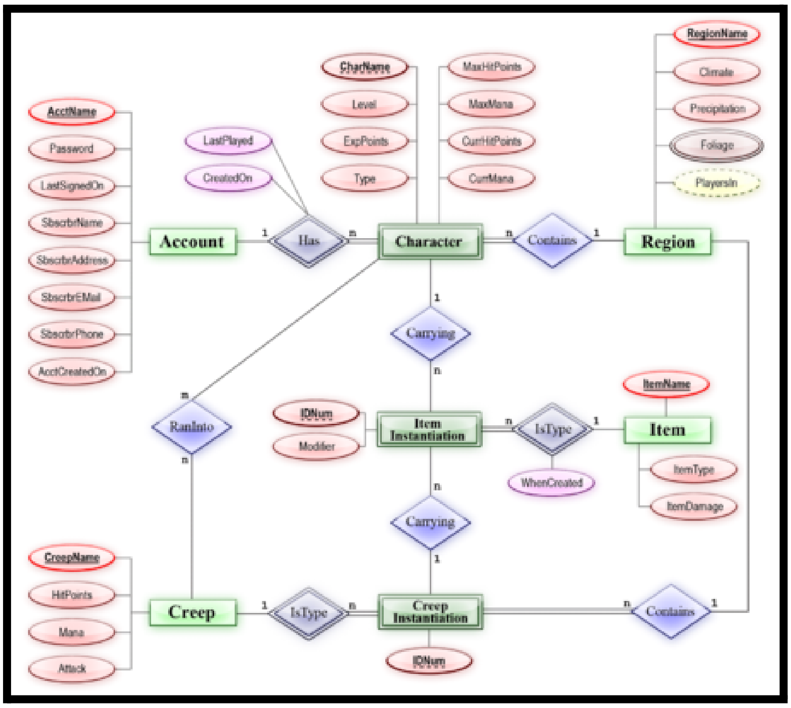
- Each write operation(insert, delete, modify) changes the current state in the database to its new state.

Data Models:

in Database Management System (DBMS), is the concept of tools that are developed to summarize the description of the database.There are three types of data models:

1- **High level or conceptual data model**

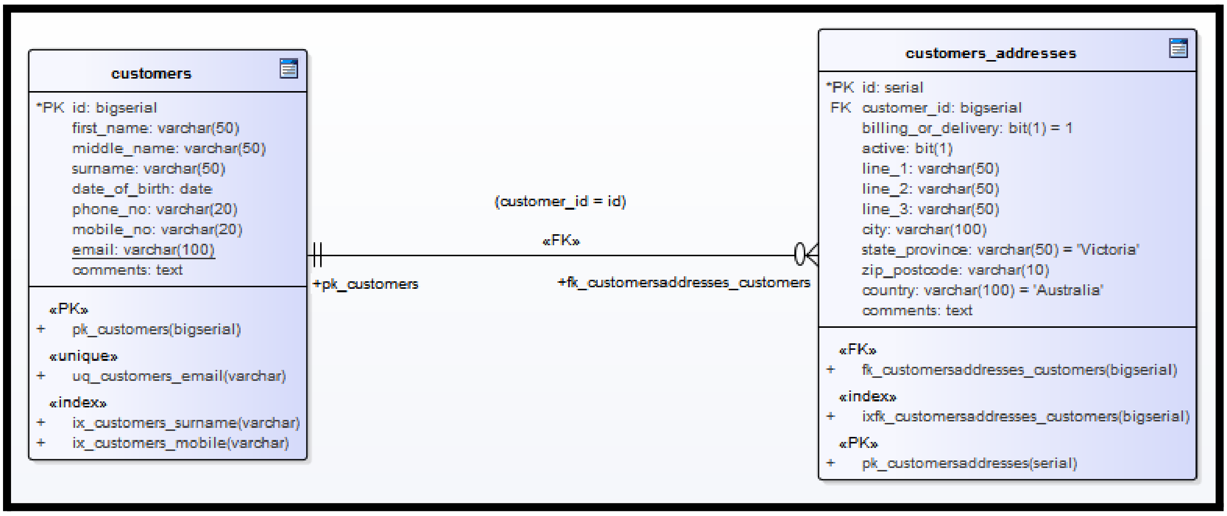
High-level conceptual data models provide concepts for presenting data in ways that are close to the way people perceive data.This model describe data in general (Tables,attributes,,relationships). A typical example is the entity relationship model, which uses main concepts like entities, attributes and relationships. An entity represents a real-world object such as an employee or a project. The entity has attributes that represent properties such as an employee's name, address and birthdate. A relationship represents an association among entities; for example, an employee works on many projects. A relationship exists between the employee and each project. This model describe data in general.



Entity relationship model

2- Low level or physical data model

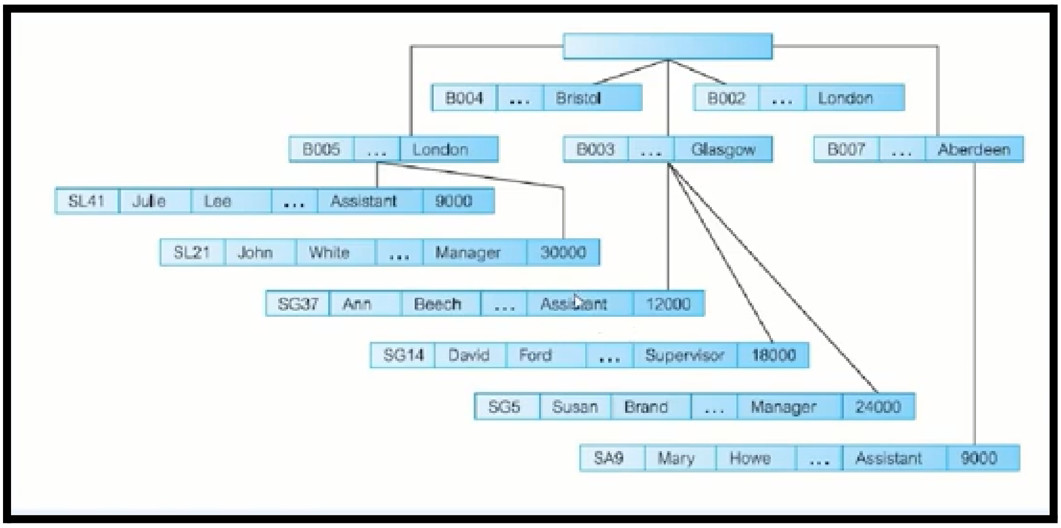
provides concepts that describe the details of how data is stored in the computer model. Low-level data model is only for Computer specialists not for end-user. This type describes accurate details about the database such as how data is stored on disk, the operations of the database, indexing, and relationships.


customers
*PK id: bigserial
first_name: varchar(50)
middle_name: varchar(50)
surname: varchar(50)
date_of_birth: date
phone_no: varchar(20)
mobile_no: varchar(20)
email: varchar(100)
comments: text
«PK»
+ pk_customers(bigserial)
«unique»
+ uq_customers_email(varchar)
«index»
+ ix_customers_surname(varchar)
+ ix_customers_mobile(varchar)
(customer_id = id)
«FK»
+pk_customers
+fk_customersaddresses_customers
customers_addresses
*PK id: serial
FK customer_id: bigserial
billing_or_delivery: bit(1) = 1
active: bit(1)
line_1: varchar(50)
line_2: varchar(50)
line_3: varchar(50)
city: varchar(100)
state_province: varchar(50) = 'Victoria'
zip_postcode: varchar(10)
country: varchar(100) = 'Australia'
comments: text
«FK»
+ fk_customersaddresses_customers(bigserial)
«index»
+ ixfk_customersaddresses_customers(bigserial)
«PK»
+ pk_customersaddresses(serial)

Physical Model

3- Representational data model

This type of data model is used to represent only the logical part of the database and does not represent the physical structure of the databases. The representational data model allows us to focus primarily, on the design part of the database. A popular representational model is Relational model.



B004 ... Bristol

B002 ... London

B005 ... London

B003 ... Glasgow

B007 ... Aberdeen

SL41 Julie Lee ... Assistant 9000

SL21 John White ... Manager 30000

SG37 Ann Beech ... Assistant 12000

SG14 David Ford ... Supervisor 18000

SG5 Susan Brand ... Manager 24000

SA9 Mary Howe ... Assistant 9000

Relational model

Distributed database

Distributed database overview

- ➔ A distributed database is, in brief, an integrated database which is built on top of a computer network rather than on a single computer .
- ➔ In a distributed database system, the database is stored on several computer, the computers communicate with one another through various communication media, such as high-speed buses or telephone lines.
- ➔ Databases in the collection are logically interrelated with each other. Often they represent a single logical database.
- ➔ The processers distributed system may vary in size and function. They may include small microcomputers, workstations, minicomputers, and large general purpose computer systems.
- ➔ These processors are referred to by a number of different names, such as sites, nodes, and computers, depending on the context in which they are mentioned.We mainly use the term site, in order to emphasize the physical distribution of these systems.
- ➔ A distributed database system consists of a collection of sites, each of which may participate in the execution of transaction which access data at one site, or several sites.
- ➔ The main difference between centralized and distributed database systems is that, in the former, the data resides in one single location, while in the latter, the data resides in several locations.
- ➔ Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites.

Structure Of Distributed Database:

A distributed database system haves of a collection of sites, each of which keeps a local databases system. Every site is able to process local transactions, the transactions that access data only in that one site. In addition, a site may participate in the implementation of global transactions, those transactions that access data at various sites.

The execution of global transactions on the distributed architecture needs communication between the sites. Figure below illustrates a representative distributed database system architecture having transactions.

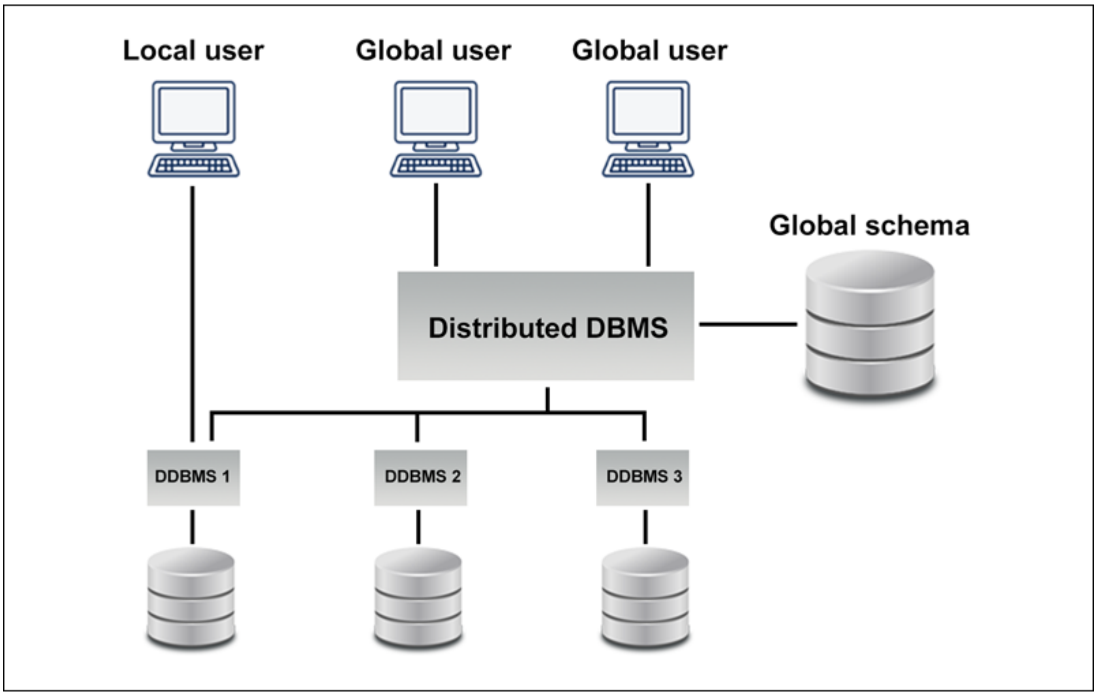
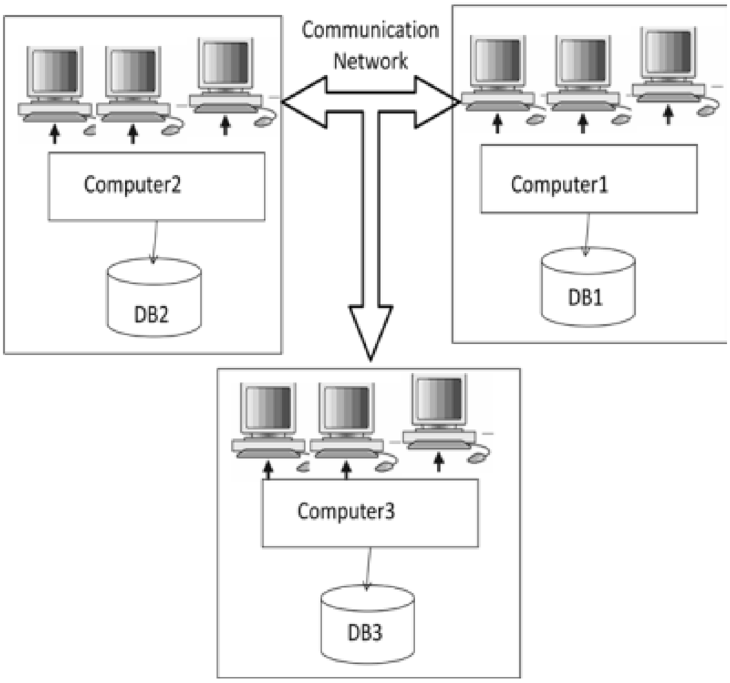


Figure: Distributed Database Schema and Transactions

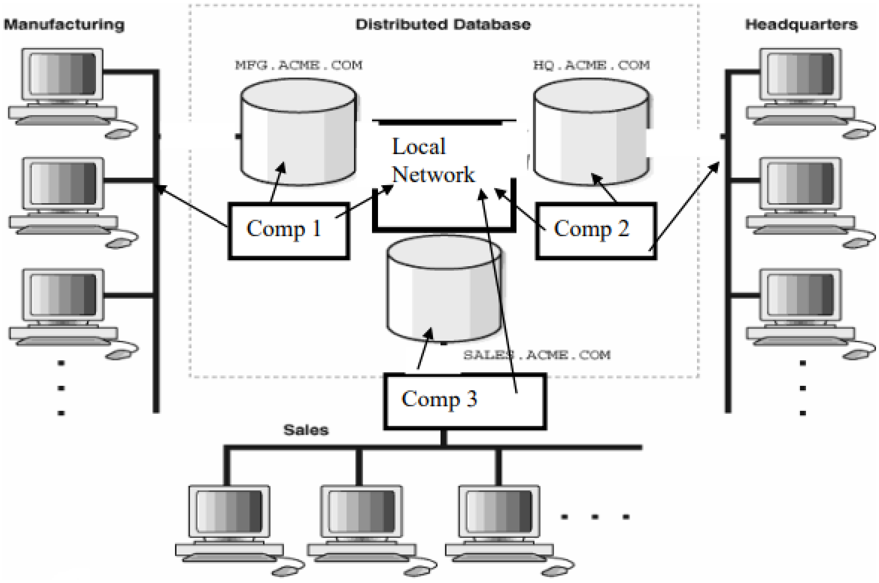
There are many examples of the structure of distributed database depending on local and global distribution of data and transactions, let us consider a few examples:

Example 1: Consider a bank that has three branches at different locations. At each branch, a computer controls the teller terminals of the branch and the account database at one branch. Computer are connected by a communication network (see figure below).



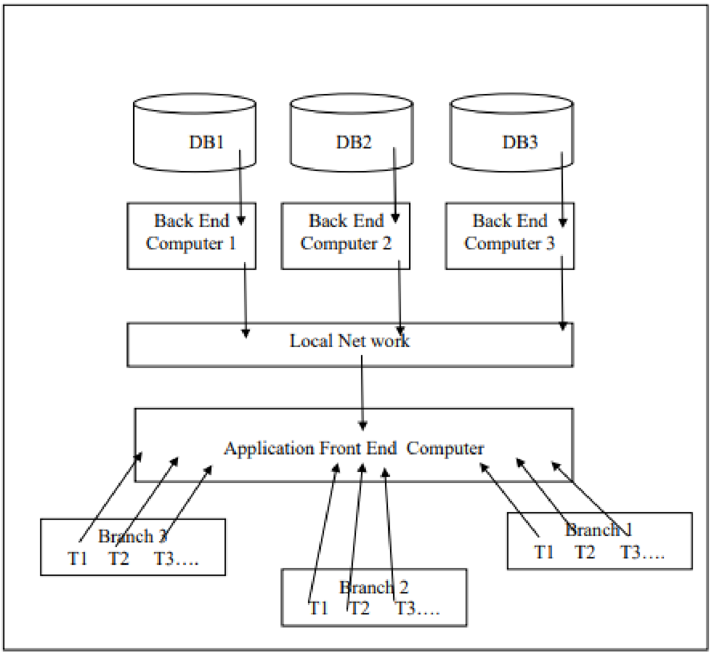
Distributed Database for global application

Example 2: Consider the same bank of previous example, with a system configuration as shown in figure below . The same processors with their databases have been moved away from the branches to a common building and are now connected with a high-band width local network.



Distributed Database on Local Network LAN

Example 3: Consider the same bank of the previous example but with the system configuration shown in figure below . The data of the different branches are distributed on three "backend" computers, which perform the database management functions. The application programs are executed by a different computer, which requests database access services from back ends when necessary.



Multi processor system

NOTE:

- The back end which is manages access structures, query evolution and optimization, concurrency control and recovery.
- The front end of DB consist of tools such as : SQL user interface, form interface, report generation tools and data mining and analysis tools.

Advantages of Distributed Database

1-Improved availability: Availability is defined as the probability that the system is continuously available during a time interval. A failure at one site of a DDBS, or a failure of a communication link making some sites unreachable, does not make the entire system inoperable. Distributed DBMSs are designed to carry on the function despite such failures.

2-More Reliable : Reliability is basically defined as the probability that a system is running at a certain time.In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.

3- Easier Expansion: If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning. However, in distributed databases, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions.

4- Improved Performance: With the concept of parallelism of distributed DBMSs,we can achieve interquery and intraquery parallelism by executing multiple queries at different sites by breaking up a query into a number of subqueries that basically executes in parallel which basically leads to improvement in performance.

- **Inter-query Parallelism** It is a form of parallelism where many different Queries or Transactions are executed in parallel with one another on many processors.
- **Intra-Query Parallelism** It is the form of parallelism where Single Query is executed in parallel on many processors.

5- Better Response: If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. On the other hand, in centralized systems, all queries have to pass through the central computer for processing, which increases the response time.

7- Data Sharing and distributed control:

If a number of different sites are connected , then a user at one site may be able to access data available at another site. For example, in the distributed banking system it is possible for a user in one branch to access data in another branch. Without this capability, a user wishing to transfer funds from one branch to another would have to resort to some external mechanism that would, in effect, be a single centralized database.

The primary advantage of sharing data by means of data distribution is that each site is able to retain a degree of control over data stored locally. In a centralized system, the database administrator of the central site controls the database. In a distributed system, there is a global database administrator responsible for the full system. A part of these responsibilities is delegated the local database administrator for each site. Depending upon the design of the distributed database system, each administrator may have a different degree of local autonomy. The possibility of local autonomy is often a major advantage of distributed databases.

Disadvantages of DDB

- Need for complex and expensive software –Distributed Databases are a network of many computers present at different locations and they provide an outstanding level of performance, availability, and of course reliability. DDBMS demands complex and often expensive software to provide data transparency and coordination across several sites.Therefore, the Distributed DBMS is comparatively more complex than a centralized DBMS.

- Processing overhead – Even simple operations may require a large number of communications and additional calculations to provide uniformity in data across the sites.

- Data integrity – In a vast Distributed database system, maintaining data consistency is important. All changes made to data at one site must be reflected on all the sites, so the need for updating data in multiple sites pose problems of data integrity.
- Overheads for improper data distribution – Responsiveness of queries is largely dependent upon proper data distribution. Improper data distribution often leads to very slow response to user requests.
- Security issues-In a Distributed Database, the security of data as well as a network is a prime concern. A network can be easily attacked for data theft and misuse.

Fundamental principles of distributed DB

The user in the distributed system should be able to behave exactly as if the system were not distributed because distributed systems have:

1- Local autonomy.
2- No reliance on a central site.
3- Continuous operation.
4- Location independence.
5- Fragmentation independence.
6- Replication independence.
7- Distributed query processing.
8- Distributed transaction management.
9- Hardware independence.
10- Network independence.
11- DBMS independence

1- local autonomy:

- The sites in the distributed system should be autonomous since Local autonomy means that all operations at a given site are controlled by that site. No site depends on other sites for its successful operation.
- Local autonomy also implies that local data is locally owned and managed with local accountability. All data belongs to some local database even if it is accessible from other sites. Such matters as integrity, security, and physical storage representation of local data remain under the control of the local site.

2- No reliance on a center site:-

local autonomy implies that all sites must be treated as equals ,so there must not be any reliance on central site. Reliance on a central site would be undesirable for at least the following reasons:

1- Central site may be a bottleneck.

2- The system would be vulnerable if the central site went down.

3- Continuous operation.

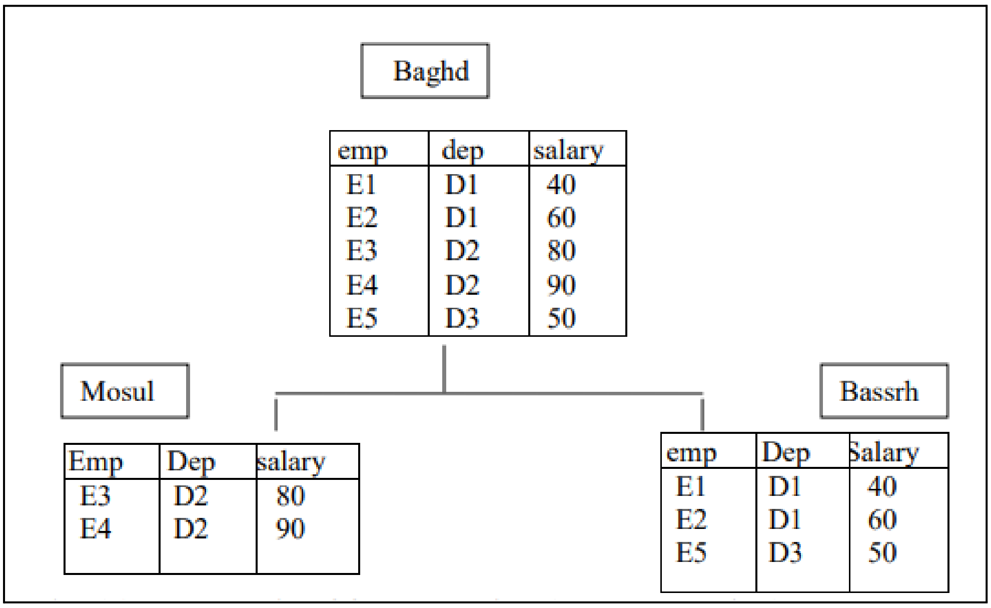
4- Location independence.

It is also known as location transparency the idea of transparency is: The users should not have to know where data is physically stored. But rather should be able to behave at least from a logical standpoint as if the data were all stored at their own local site. Location independence is desirable because it simplifies application programs and end-user activities, such as it allow data to migrate from site to site.

And that allows data to be moved around the network in response to changing performance requirements.

5- Fragmentation independence.

The data can be divided into pieces or fragments for physical storage purposes and distinct fragments can be stored at different sites. The fragment is important for performance reasons. Data can be stored at the location where it is most frequently used. So that most operations are local and network traffic is reduced. Look at the following example:



Baghd

emp	dep	salary
E1	D1	40
E2	D1	60
E3	D2	80
E4	D2	90
E5	D3	50

Mosul

Emp	Dep	salary
E3	D2	80
E4	D2	90

Bassrh

emp	Dep	Salary
E1	D1	40
E2	D1	60
E5	D3	50

There are basically three kinds of fragmentation:

- Horizontal fragmentation.
- Vertical fragmentation.
- Mixed fragmentation.

Fragmentation independence also known as fragmentation transparency and it like location transparency is desirable because it simplifies application programs and end user activities, such as it allows the data to be defragmentation at any time.

Fragmentation transparency implies that users will be presented with a view of the data in which the fragments are logically recombined by means of suitable joins and unions. It is the responsibility of the optimizer to determine which fragments need to be physically accessed in order to satisfy any given user request.

6- Replication independence

It means a given fragment can be represented in storage by many distinct copies or replicas stored at many distinct sites. Replication is desirable because :

1- it can mean better performance applications can operate on local copies instead of having to communicate with remote sites.

2- It can also mean better availability of the processing at least for retrieval as long as at least one copy remains available.

- The disadvantage of replication is that when a given replicated object is updated, all copies of that object must be updated which call update propagation.

7- Distributed query processing Distributed databases answer queries in a distributed environment that manages data at multiple sites. High-level queries are transformed into a query execution plan for simpler management.

8- Distributed Transaction Management –

The basic function of DDBMS is its ability to devise execution strategies for queries and transactions that access data from more than one site and to synchronize the access to distributed data and basically to maintain the integrity of the complete database.

9- Network independence:

- All, real-world computer installation typically involve a multiplicity of different machines like IBM machines, fujitsu machines, HP machines, PCs and workstations of various kinds, and so on .
- And there is a real need to be able to integrate the data on all of those systems and present the user with a “single – system image”. Thus , it is desirable to be able to run the same DBMS on different hardware platforms, and furthermore to have those different machines all participate as equal partners in a distributed system.

10-Operating system independence

This objective is partly a corollary of the previous one, and also does not really require much discussion here. it is obviously desirable, not only to be able to run the same DBMS on different hardware platforms, but also to be able to run it on different operating system platforms as well including different operating systems on the same hardware and have an OS/390 version and a UNIX version and a Windows version all participate in the same distributed system.

11-Network independence Once again there is not much to say ; if the system is to be able to support many disparate sites, with disparate hardware and disparate operating system , it is obviously desirable to be able to support a variety of disparate communication networks also.

12- DBMS independence

Under this heading, All that is really needed is that the DBMS instances at different sites all support the same interface __ they do not necessarily all have to be copies of the same DBMS software. the fact is, real-world computer installations typically run not only many different machines and many different operating systems, they very often run different DBMSs as well; and it would be nice if those different DBMSs could all participate somehow in a distributed system, in other words, the ideal distributed system should provide DBMS independence.

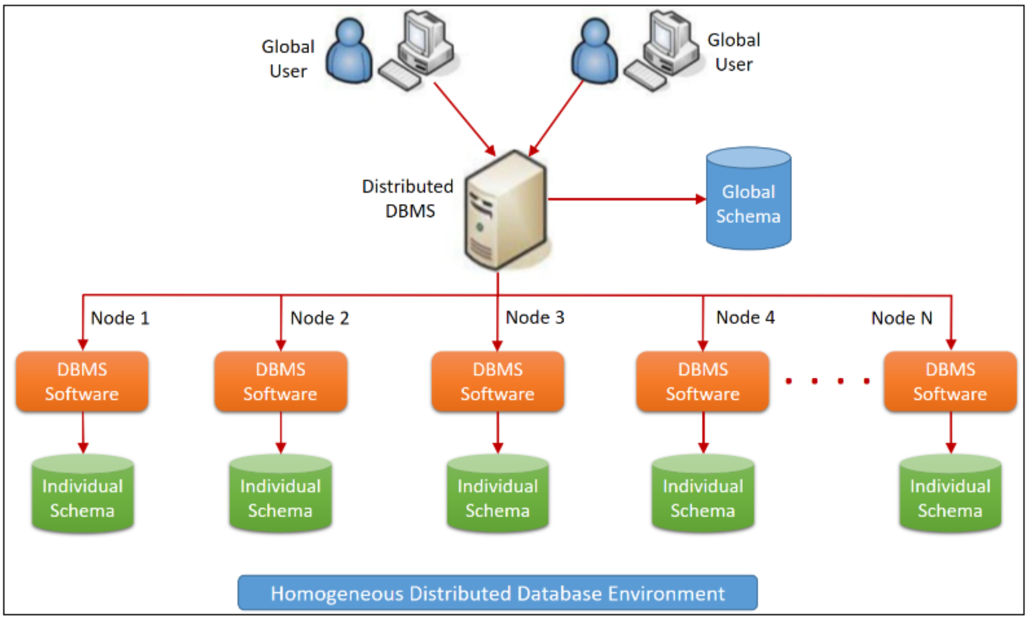
Types of Distributed Databases:

There are two types of distributed databases:

1-Homogeneous فديوات عن الشرح

- A homogenous distributed database is a network of identical databases stored on multiple sites. The sites have the same operating system, DDBMS, and data structure, making them easily manageable.
- 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.

The following diagram shows an example of a homogeneous database:

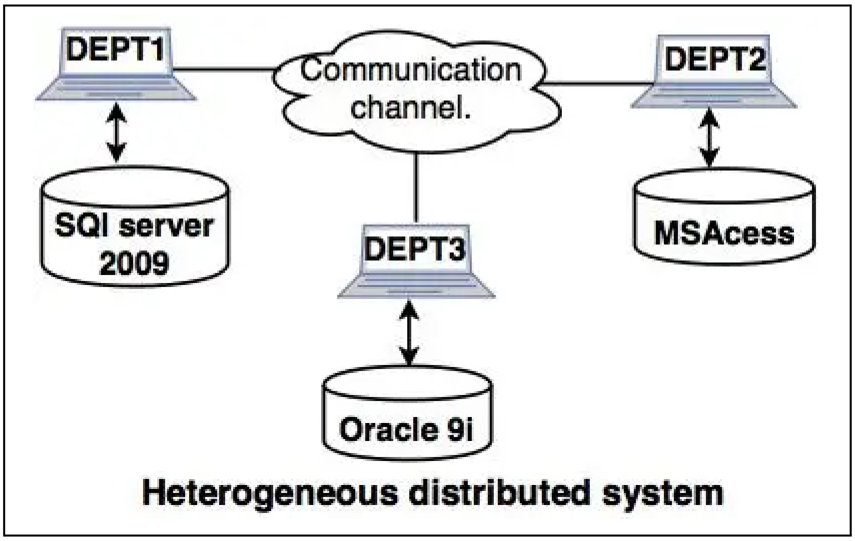


2- Heterogeneous

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.
- 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.

The following diagram shows the Heterogenous database:



TIP:

There many types of DBMS:

1. Hierarchical databases
2. Network databases
3. Relational databases
4. Object-oriented databases
5. Graph databases
6. ER model databases
7. Document databases
8. NoSQL databases

Distributed DBMS Architectures

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 DBMS can operate independently.
- **Heterogeneity** − It refers to the uniformity or dissimilarity of the data models, system components and databases.

Architectural Models

Some of the common architectural models are

1- Client - Server Architecture for DDBMS.

2- Peer - to - Peer Architecture for DDBMS.

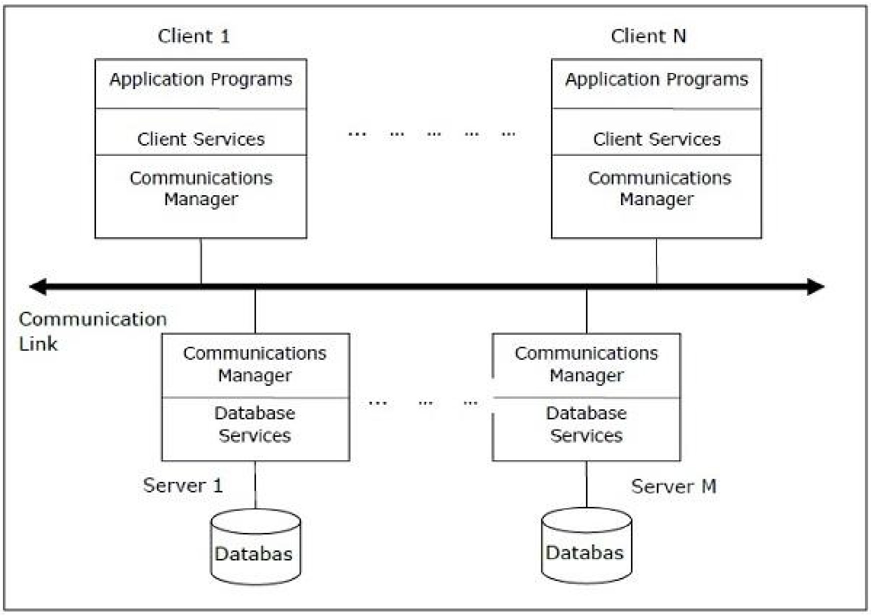
3- Multi - DBMS Architecture.

1- 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.

The two different client - server architecture are :

- Single Server Multiple Client
- Multiple Server Multiple Client (shown in the following diagram)

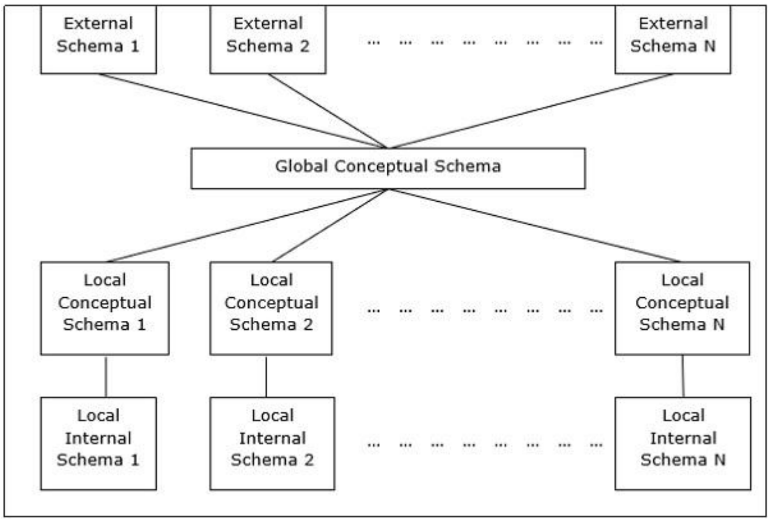


2- Peer- to-Peer Architecture for DDBMS

In these systems, each peer acts both as a client and a server for imparting database services. The peers share their resource with other peers and co-ordinate their activities.

This architecture generally has four levels of schemas −

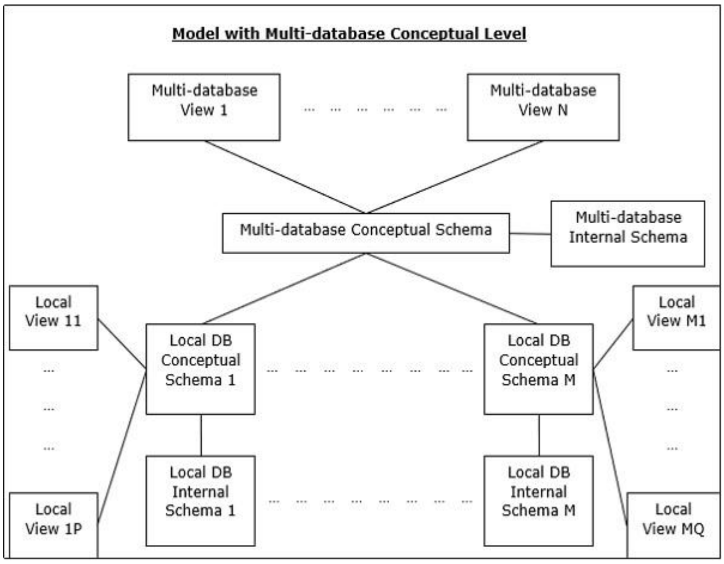
- Global Conceptual Schema − Describes the global logical view of data.
- Local Conceptual Schema − Describes logical data organization at each site.
- Local Internal Schema − Describes physical data organization at each site.
- External Schema − Describes user view of data.

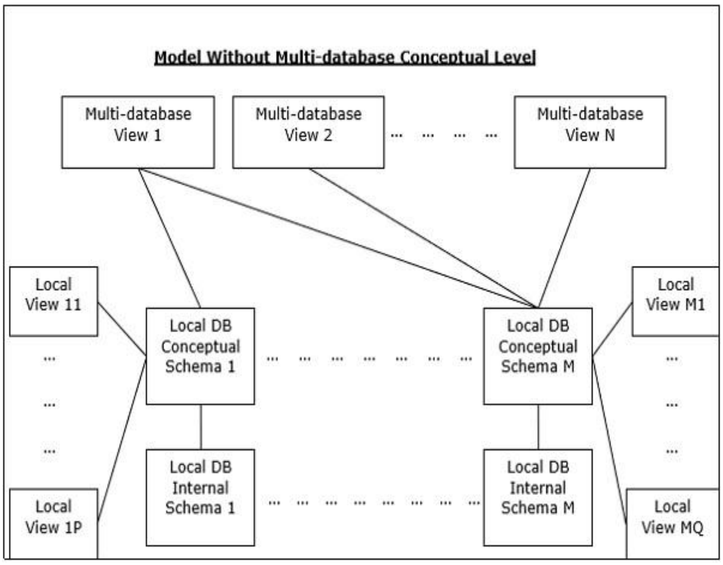


3- Multi - DBMS Architectures

This is an integrated DBS consist of a collection of two or more autonomous database systems.

- There are two design alternatives for multi-DBMS −
  - Model with multi-database conceptual level.
  - Model without multi-database conceptual level.
- Multi-DBMS can be expressed through six levels of schemas
  - Multi-database View Level − Describes multiple user views comprising of subsets of the integrated distributed database.
  - Multi-database Conceptual Level − Describes integrated multi-database that comprises of global logical multi-database structure definitions.
  - Multi-database Internal Level − Describes the data distribution across different sites and multi-database to local data mapping.
  - Local database View Level − Describes public view of local data.
  - Local database Conceptual Level − Describes local data organization at each site.
  - Local database Internal Level − Describes physical data organization at each site.


Model with Multi-database Conceptual Level
Multi-database View 1
Multi-database View N
Multi-database Conceptual Schema
Multi-database Internal Schema
Local View 11
Local View 1P
Local DB Conceptual Schema 1
Local DB Internal Schema 1
Local DB Conceptual Schema M
Local DB Internal Schema M
Local View M1
Local View MQ

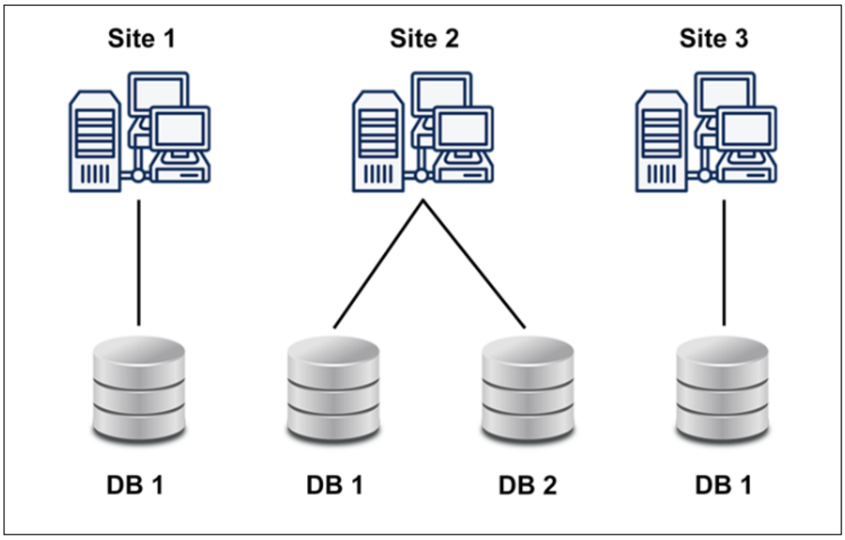

Model Without Multi-database Conceptual Level
Multi-database View 1
Multi-database View 2
Multi-database View N
Local View 11
Local View 1P
Local DB Conceptual Schema 1
Local DB Internal Schema 1
Local DB Conceptual Schema M
Local DB Internal Schema M
Local View M1
Local View MQ

Distributed Database Storage

Distributed database storage is managed in two ways:

1-Replication

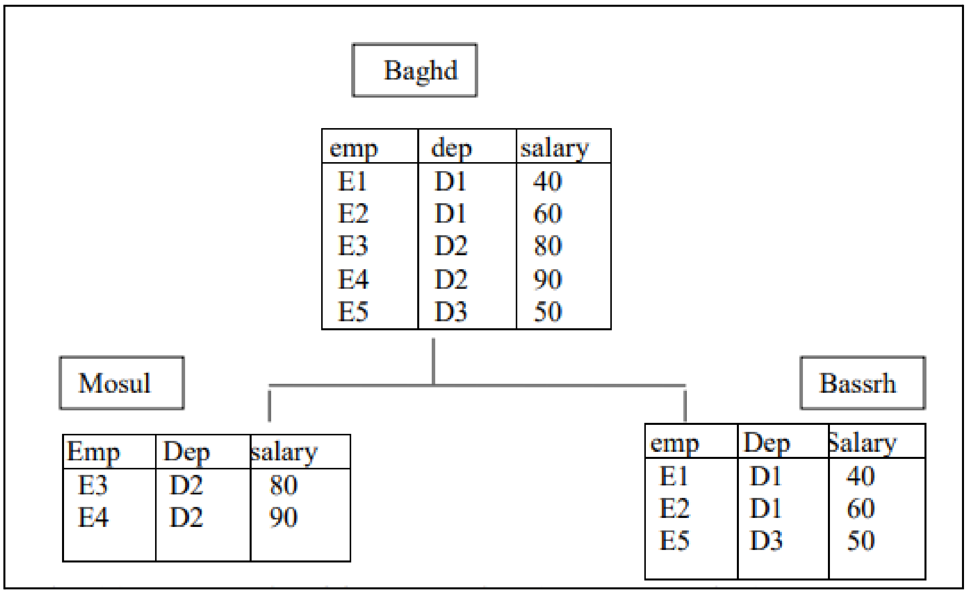
- In database replication, the systems store copies of data on different sites. If an entire database is available on multiple sites, it is a fully redundant database.
- The advantage of database replication is that it increases data availability on different sites and allows for parallel query requests to be processed.
- However, database replication means that data requires constant updates and synchronization with other sites to maintain an exact database copy. Any changes made on one site must be recorded on other sites, or else conflict will occur.
- Constant updates cause a lot of server overhead and complicate concurrency control, as a lot of concurrent queries must be checked in all available sites.



2- Fragmentation

- Fragmentation is a process of dividing the whole or full database into various subtables or sub relations so that data can be stored in different systems.
- The small pieces of sub relations or subtables are called *fragments*. These fragments are called logical data units and are stored at various sites.
- The precondirtion for fragmentation is to make sure that the fragments can later be reconstructed into the original relation without losing data.
- The advantage of fragmentation :
  - There are no data copies, which prevents data inconsistency.
  - The fragment is important for performance reasons,data can be stored at the location where it is most frequently used so that most operations are local and network traffic is reduced .

- Look at the following example:



Baghd

emp	dep	salary
E1	D1	40
E2	D1	60
E3	D2	80
E4	D2	90
E5	D3	50

Mosul

Emp	Dep	salary
E3	D2	80
E4	D2	90

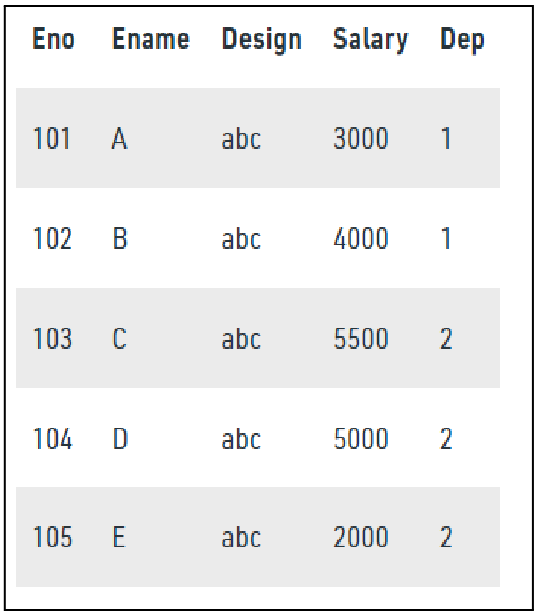
Bassrh

emp	Dep	Salary
E1	D1	40
E2	D1	60
E5	D3	50

There are basically three kinds of fragmentation:

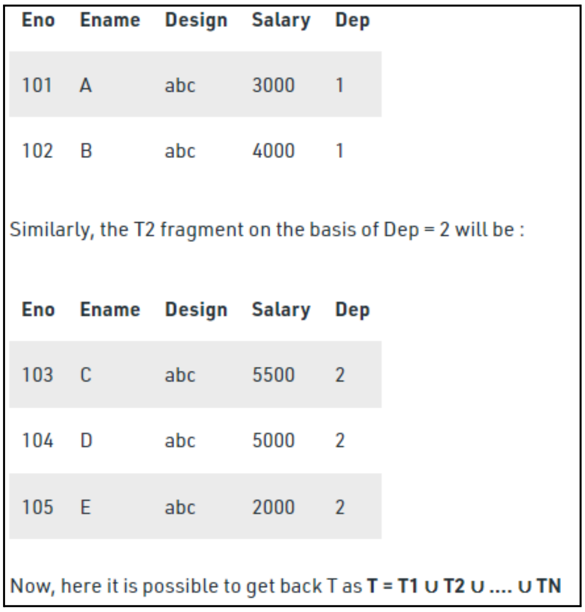
1. **Horizontal fragmentation**

   refers to the process of dividing a table horizontally by assigning each row or (a group of rows) of relation to one or more fragments.These fragments are then be assigned to different sides in the distributed system. Some of the rows or tuples of the table are placed in one system and the rest are placed in other systems. The rows that belong to the horizontal fragments are specified by a condition on one or more attributes of the relation. If we have EMPLOYEE table **T** , this Employee table can be divided into different fragments like (T1 and T2):



Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

⇒



Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1

Similarly, the T2 fragment on the basis of Dep = 2 will be :

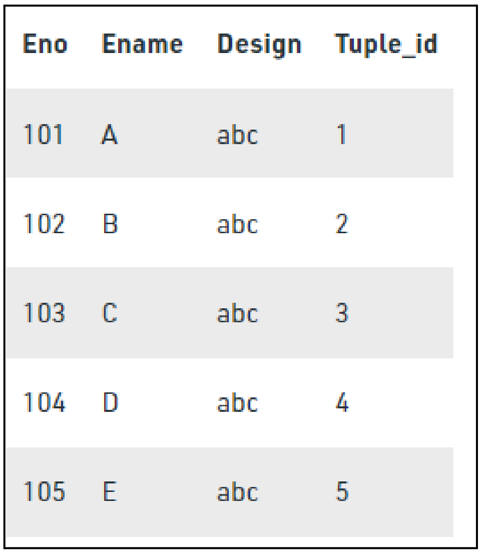
Eno	Ename	Design	Salary	Dep
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

Now, here it is possible to get back T as **T = T1 ∪ T2 ∪ .... ∪ TN**

Now, to get back T as
T = T1 ∪ **T2** ∪ **.... ∪ TN**
Using **UNION** operation

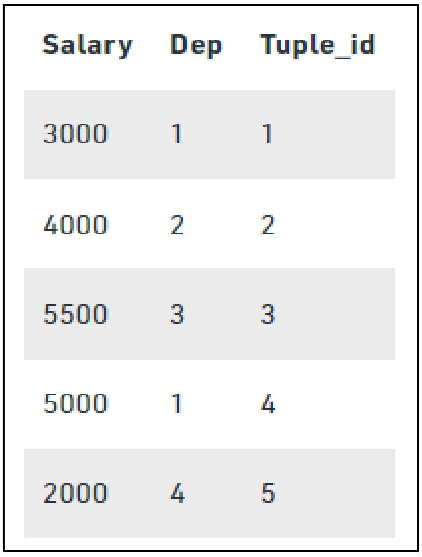
2. Vertical Fragmentation

refers to the process of dividing a table vertically by attributes are columns. In this fragmentation, some of the attributes are stored in one system and the rest are stored in other systems. This is because each site may not need all columns of a table. In order to take care of restoration, each fragment must contain the primary key field(s) in a table. The fragmentation should be in such a manner that we can rebuild a table from the fragment by taking the natural JOIN operation and to make it possible we need to include a special attribute called ***Tuple-id*** to the schema. Using theEmployee table, it can be fragmented vertically into T1 and T2 as follow:



Eno	Ename	Design	Tuple_id
101	A	abc	1
102	B	abc	2
103	C	abc	3
104	D	abc	4
105	E	abc	5

⇛



Salary	Dep	Tuple_id
3000	1	1
4000	2	2
5500	3	3
5000	1	4
2000	4	5

To get back to the original T, we join these two fragments T1 and T2 as **πEMPLOYEE (T1 ⋈ T2)** using **JOIN** operation

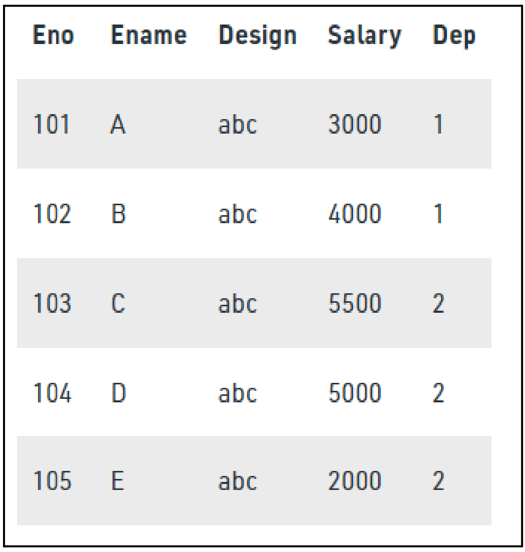
3. Mixed Fragmentation

The combination of vertical fragmentation of a table followed by further horizontal fragmentation of some fragments is called mixed or hybrid fragmentation. In some situations, the horizontal and the vertical fragmentation isn't enough to distribute data for some applications and in that conditions, we need a fragmentation called a mixed fragmentation.

Mixed fragmentation can be done in two different ways:

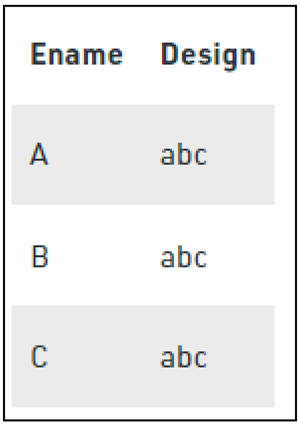
1. The first method is to first create a set or group of horizontal fragments and then create vertical fragments from one or more of the horizontal fragments.
2. The second method is to first create a set or group of vertical fragments and then create horizontal fragments from one or more of the vertical fragments.

For example, we implement Mixed defragmation on the Employee table:



Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

⇛



Ename	Design
A	abc
B	abc
C	abc

The original relation can be obtained by the combination of **JOIN** and **UNION** operations

-ISO (International Standared Organization)

Surely you have noticed that many products have this mark. This symbol is given to the company that applies a set of quality standards in a specific field. The institution was founded in 1947 in Geneva and now includes hundreds of institutions specialized in providing quality certificates.

Why ISO certification is required:

- If your product is ISO marked or your company has an ISO mark, then it is proved that you are providing standard services to your customers or users.
- If the product has an ISO mark, then the customers will have confidence that the product they are using is good and beneficial for them.

Two main features of ISO:

1. Internal quality audit at predetermined intervals.
2. Continuous monitoring of the quality of the system.

Standards of ISO: There are many standards of ISO available for every type of product, business, and industry.

1. ISO 9000: ISO 9000 is a family of standards deals with quality management systems.

2. ISO 9001 This standard is a quality management standard. The standard is useful for merchant exporters, manufacturers, training centres, and service providers.

3. ISO 22000: This standard is for food safety. This is a very important standard that is useful for food manufacturing units, food supplier units, and restaurants.

4. ISO 14001: This is for environmental protection. If an organization or company has this standard, it means the organization is following all rules and regulations of the environment.

5. ISO 27000: This standard is for information technology. This standard is useful for the IT industry. Guidelines for improving the performance of the IT industry.

6. ISO / TS 16949: This standard is for the automobile industry. Safety major, it for the automobile industry, and the training process of workers.

7. ISO 50001: This is an energy standard. The company can achieve this standard by saving energy and using energy in the best way.

8. ISO 13485: This standard is for medical equipment and is useful for medical equipment manufacturing companies.

9. ISO 31000: This standard is for risk management. The standard is useful for minimizing manufacturer risk, accidental risk.

10. ISO 26000: This standard is very important for employees and labour. It includes improving labour policies, ethical behaviour, human rights & consumer issues, etc.

11. ISO 14000: It is a set of rules that help companies to reduce industrial waste and environmental damage. It is used for environmental management. The revised version of this standard is 14001.

—---

-OSI (Open System Interconnection)

- The International Standard Organization has a well-defined model for Communication Systems known as Open System Interconnection, or the OSI Model.

- This layered model (OSI) is a conceptualized view of how one system should transfer information and communicate with the other, using various protocols defined in each layer.
There are the seven OSI layers. Each layer has different functions:

7-Application Layer

Sometimes called the application interface, because it is the stage in which the user deals with the interface of programs such as the browser, file transfer and mail. This layer defines the type of protocol used (HTTP, FTP, SMTP,.....).

6- Presentation Layer

The functions of this layer are to prepare the data that will be sent over the network to take the appropriate extension. This data includes images, videos, and texts. After the previous preparation process, it will be compressed (when sending) or decompressed (when receiving), and the data will also be encrypted or decrypted.

5-Session Layer

This layer is responsible for opening communication channels between the user and the other party and managing and closing the session. When you open more than one application on your device, the role of this layer appears here, as it opens a communication channel for each application separately. This layer also determines the type of connection if it is (single), (half duplex) or (full duplex).

4-Transport Layer

One of the functions of this layer is to segment the data received from the session layer into segments. It is also responsible for determining the type of protocol used:

UDP: It is a protocol that transmits data without making sure that it has arrived, for example, video files or television and radio stations on the Internet.

TCP: It is a protocol that sends data and ensures the integrity of its arrival.

3-Network Layer

After the data stream is transferred from the Transport layer, an IP will be added to it to become Packet, and then the best path to transfer this data is determined, and this is called routing. The device that deals with this layer is the router.

2-Data-Link Layer

-A header and trailer are added to the packet coming from the Network layer, to be called a frame.

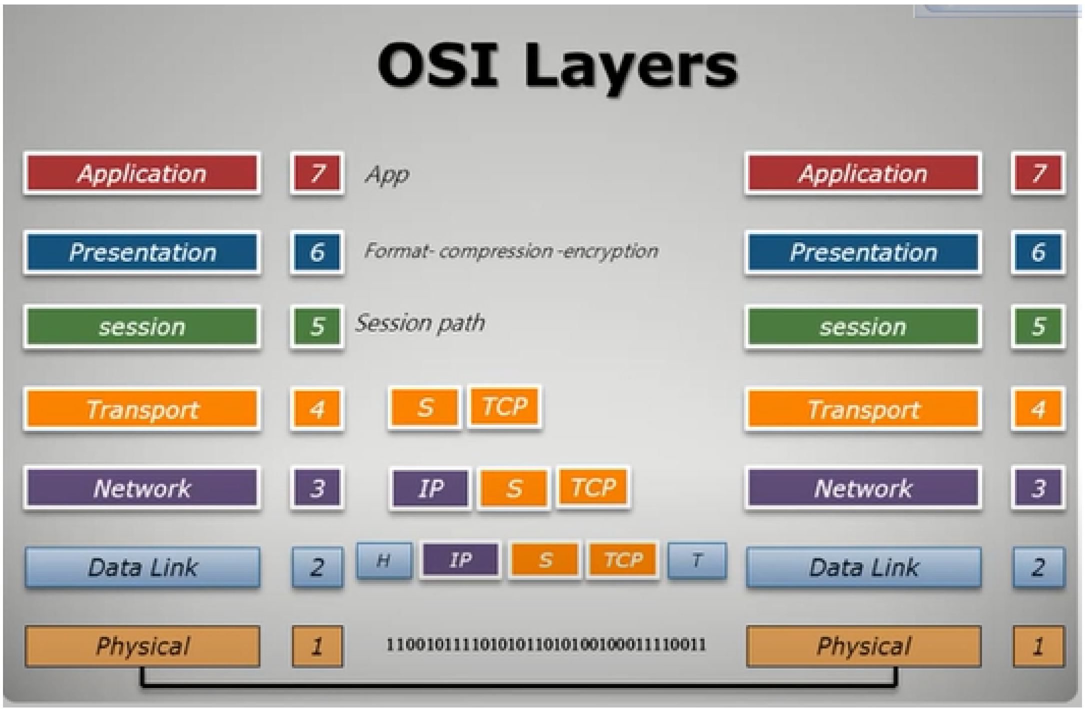
-Adding the Mac address of the devices to the header, and the error detection to the trailer.

-Determine the best time to send data over the cable, by making sure that the cable is free of any data before transmission.

-The device or hardware that deals with this layer is the switch or bridge.

1-Physical Layer

In this layer, the received data is converted as a frame from the previous layer and converted into bits, that is, into electrical signals to pass through the cable.the devices that deals with this layer are NIC ,hub, cables,..etc.



Sender Receiver

Types of Computer Networks

A computer network is a cluster of computers over a shared communication path that works for the purpose of sharing resources from one computer to another, provided by or located on the network nodes. The following are the types of networks:

1. Personal Area Network (PAN)

This network is used to connect personal devices such as your phone and computer using a USB cable.

2. Local Area Network (LAN)

The devices in the network are in one place such as computers in a building are connected togather and operated under single administrative system.

3. Wide Area Network (WAN)

It is called by this name when the network connects distant places such as cities or countries, this type of network connect more than LAN by internet.

4. Wireless Local Area Network (WLAN)

WLAN is a type of computer network that acts as a local area network but makes use of wireless network technology like Wi-Fi.

5. Metropolitan Area Network (MAN)

In this type, a group of LAN networks is linked in the same city or the same country, such as linking branches of an institution or company.

6. Campus Area Network (CAN)

CAN is bigger than a LAN but smaller than a MAN. This is a type of computer network which is usually used in places like a school or college. This network covers a limited geographical area that is..

7. Storage Area Network (SAN)

SAN is a type of computer network that is high-speed and connects groups of storage devices to several servers. This network does not depend on LAN or WAN.. Instead, a SAN moves the storage resources. Examples of SAN are a network of disks accessed by a network of servers.

8. System-Area Network (SAN)

A SAN is a type of computer network that connects a cluster of high-performance computers. It is a high-bandwidth network. A SAN is a type of LAN that handles high amounts of information in large requests. This network is useful for processing applications that require high network performance. Microsoft SQL Server 2005.

9. Passive Optical Local Area Network (POLAN)

A POLAN is a type of computer network which is an alternative to a LAN. POLAN uses optical splitters to split an optical signal from a single strand of single-mode optical fiber to multiple signals to distribute users and devices.

10. Enterprise Private Network (EPN)

EPN is a type of computer network mostly used by businesses that want a secure connection over various locations to share computer resources.

11. Virtual Private Network (VPN)

A VPN is a type of computer network that extends a private network across the internet and lets the user send and receive data as if they were connected to a private network even though they are not. VPN protects you from malicious sources by operating as a medium that gives you a protected network connection.

12. Home Area Network (HAN)

Many of the houses might have more than a computer. To interconnect those computers and with other peripheral devices, a network should be established similar to the local area network (LAN) within that home. Such a type of network that allows a user to interconnect multiple computers and other digital devices within the home is referred to as Home Area Network (HAN). HAN encourages sharing of resources, files, and programs within the network. It supports both wired and wireless communication.

—---

-Network protocols

A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. The following are some of the protocols:

1. **Transmission Control Protocol (TCP)**: One of the important communication protocols that is used to transfer data over the network, and it works with the (IP) protocol as a twin known as (TCP / IP). It divides any message into a series of packets that are sent from source to destination and there it gets reassembled at the destination.

2. **Internet Protocol (IP):** IP is designed as an addressing protocol. It is responsible for the addressing process and gives each device a unique address. The IP addresses in packets help in routing them through different nodes in a network until it reaches the destination system. TCP/IP is the most popular protocol connecting networks.

3. **User Datagram Protocol (UDP):** It used to send short messages are called datagrams, but it is distinguished by not requiring an initial connection between the two parties, that is, it is very similar to the (TCP) protocol, but the difference between them is the way of communication between devices and the speed of data delivery. In addition, unlike the (TCP) protocol, it does not wait for a notification to confirm the completion of the transfer, but is only concerned with completing the required task without caring about its success or not.

4. **Post office Protocol (POP)**: POP3 is designed for receiving incoming E-mails.

5. **Simple mail transport Protocol (SMTP)**: SMTP is designed to send and distribute outgoing E-Mail.

6. **File Transfer Protocol (FTP):** FTP allows users to transfer files from one machine to another. Types of files may include program files, multimedia files, text files, and documents, etc.

7. **Hyper Text Transfer Protocol (HTTP)**: HTTP is designed for transferring a hypertext among two or more systems. HTML tags are used for creating links. These links may be in any form like text or images. HTTP is designed on Client-server principles which allow a client system for establishing a connection with the server machine for making a request.

8. **Hyper Text Transfer Protocol Secure (HTTPS)**: same in case of HTTP except that the transferring of data is done in an encrypted format. So it can be said that https thwart hackers from interpretation or modification of data throughout the transfer of packets.

Types of Computer Networks

A computer network is a cluster of computers over a shared communication path that works for the purpose of sharing resources from one computer to another, provided by or located on the network nodes. The following are the types of networks:

1. **Personal Area Network (PAN)**

This network is used to connect personal devices such as your phone and computer using a USB cable.

2. **Local Area Network (LAN)**

The devices in the network are in one place such as computers in a building are connected togather and operated under single administrative system.

3. **Wide Area Network (WAN)**

It is called by this name when the network connects distant places such as cities or countries, this type of network connect more than LAN by internet.

4. **Wireless Local Area Network (WLAN)**

WLAN is a type of computer network that acts as a local area network but makes use of wireless network technology like Wi-Fi.

5. **Metropolitan Area Network (MAN)**

In this type, a group of LAN networks is linked in the same city or the same country, such as linking branches of an institution or company.

6. **Campus Area Network (CAN)**

CAN is bigger than a LAN but smaller than a MAN. This is a type of computer network which is usually used in places like a school or college. This network covers a limited geographical area that is..

7. **Storage Area Network (SAN)**

SAN is a type of computer network that is high-speed and connects groups of storage devices to several servers. This network does not depend on LAN or WAN.. Instead, a SAN moves the storage resources. Examples of SAN are a network of disks accessed by a network of servers.

8. **System-Area Network (SAN)**

A SAN is a type of computer network that connects a cluster of high-performance computers. It is a high-bandwidth network. A SAN is a type of LAN that handles high amounts of information in large requests. This network is useful for processing applications that require high network performance. Microsoft SQL Server 2005.

9. Passive Optical Local Area Network (POLAN)

A POLAN is a type of computer network which is an alternative to a LAN. POLAN uses optical splitters to split an optical signal from a single strand of single-mode optical fiber to multiple signals to distribute users and devices.

10. Enterprise Private Network (EPN)

EPN is a type of computer network mostly used by businesses that want a secure connection over various locations to share computer resources.

11. Virtual Private Network (VPN)

A VPN is a type of computer network that extends a private network across the internet and lets the user send and receive data as if they were connected to a private network even though they are not. VPN protects you from malicious sources by operating as a medium that gives you a protected network connection.

12. Home Area Network (HAN)

Many of the houses might have more than a computer. To interconnect those computers and with other peripheral devices, a network should be established similar to the local area network (LAN) within that home. Such a type of network that allows a user to interconnect multiple computers and other digital devices within the home is referred to as Home Area Network (HAN). HAN encourages sharing of resources, files, and programs within the network. It supports both wired and wireless communication.

—---

-Network protocols

A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. The following are some of the protocols:

1. **Transmission Control Protocol (TCP)**: One of the important communication protocols that is used to transfer data over the network, and it works with the (IP) protocol as a twin known as (TCP / IP). It divides any message into a series of packets that are sent from source to destination and there it gets reassembled at the destination.

2. **Internet Protocol (IP):** IP is designed as an addressing protocol. It is responsible for the addressing process and gives each device a unique address. The IP addresses in packets help in routing them through different nodes in a network until it reaches the destination system. TCP/IP is the most popular protocol connecting networks.

3. **User Datagram Protocol (UDP):** It used to send short messages are called datagrams, but it is distinguished by not requiring an initial connection between the two parties, that is, it is very similar to the (TCP) protocol, but the difference between them is the way of communication between devices and the speed of data delivery. In addition, unlike the (TCP) protocol, it does not wait for a notification to confirm the completion of the transfer, but is only concerned with completing the required task without caring about its success or not.

4. **Post office Protocol (POP)**: POP3 is designed for receiving incoming E-mails.

5. **Simple mail transport Protocol (SMTP)**: SMTP is designed to send and distribute outgoing E-Mail.

6. **File Transfer Protocol (FTP):** FTP allows users to transfer files from one machine to another. Types of files may include program files, multimedia files, text files, and documents, etc.

7. **Hyper Text Transfer Protocol (HTTP)**: HTTP is designed for transferring a hypertext among two or more systems. HTML tags are used for creating links. These links may be in any form like text or images. HTTP is designed on Client-server principles which allow a client system for establishing a connection with the server machine for making a request.

8. **Hyper Text Transfer Protocol Secure (HTTPS)**: same in case of HTTP except that the transferring of data is done in an encrypted format. So it can be said that https thwart hackers from interpretation or modification of data throughout the transfer of packets.

—---

Data integrity

What is data integrity?

Data integrity is the overall accuracy, completeness, and consistency of data. It is maintained by a collection of processes, rules, and standards implemented during the design phase. When the integrity of data is secure, the information stored in a database will remain complete, accurate, and reliable no matter how long it's stored or how often it's accessed.

Types of Integrity Constraint: There are two types of integrity:

A- Physcial integrity

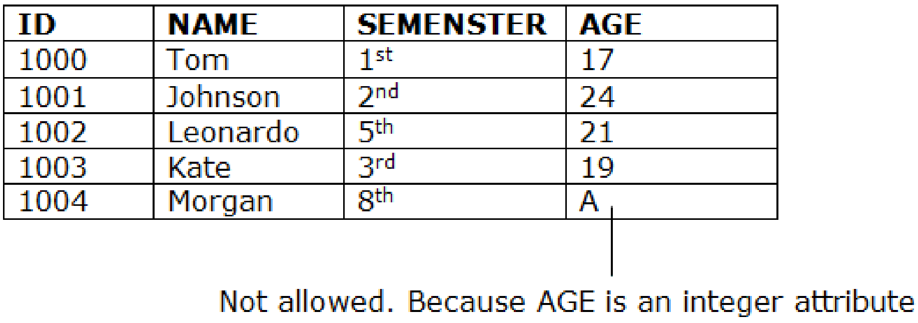
Protecting data against external factors, such as natural calamities, power outages, or hackers, falls under the domain of physical integrity. Moreover, human faults, storage attrition, and several other problems can make data operators unable to obtain information from a database.

B- Logical integrity

It concerns the rationality of data present within the relational database. Logical integrity constraints can be categorized into four types:

1- Domain Integrity

- A domain is defined as a set of suitable values that a column is permitted to include.
- The data type of domain includes string, character, integer, time, date, currency, etc.
- Domain integrity includes rules and other processes restricting the format, type, and volume of data recorded in a database. It ensures that every column in a relational database is in a defined domain.
- See the following example:

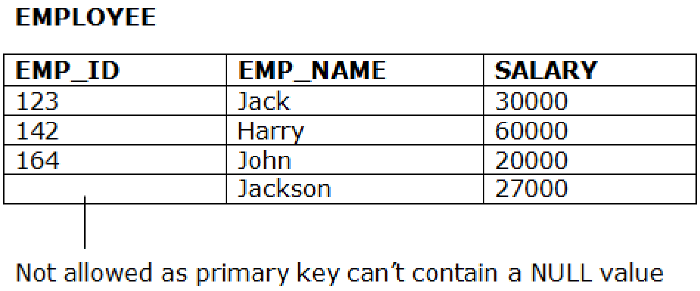


ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

Entity integrity relies on the creation of primary keys — the unique values that identify pieces of data — to ensure that data isn't listed more than once and that no field in a table is null. It's a feature of relational systems which store data in tables that can be linked and used in a variety of ways.



EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

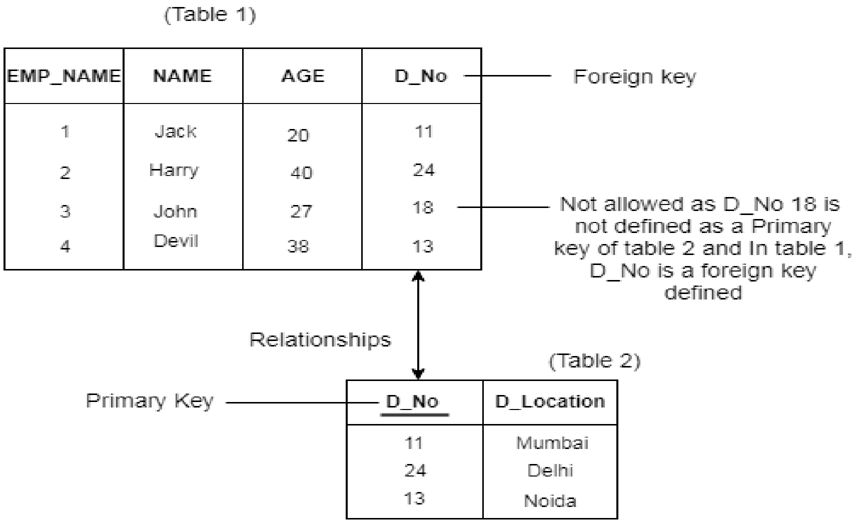
Not allowed as primary key can't contain a NULL value

3- Referential integrity

- Referential integrity refers to the series of processes that make sure data is stored and used uniformly. Rules embedded into the database's structure about how foreign keys are used ensure that only appropriate changes, additions, or deletions of data occur.

- A referential integrity constraint is specified between two tables.

- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.



(Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

Foreign key

Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined

Relationships

(Table 2)

Primary Key

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

4- User-defined integrity

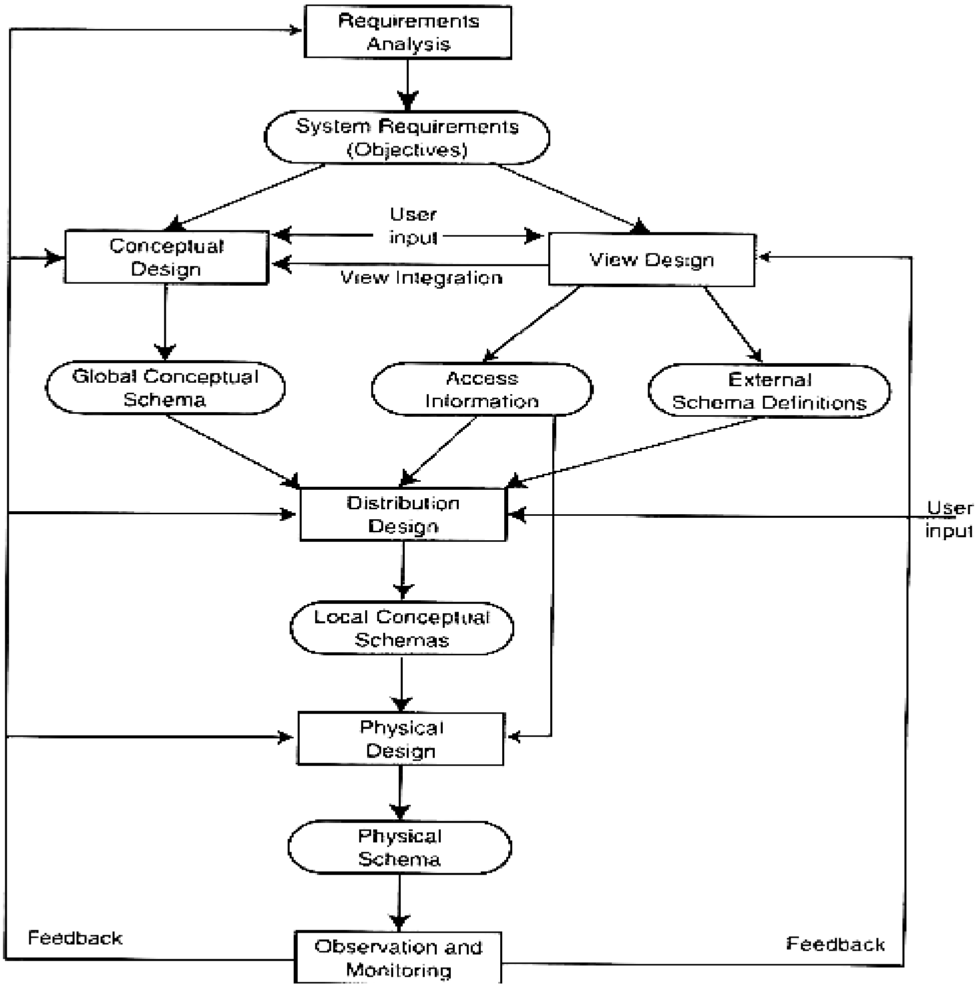
User-defined integrity involves the rules and constraints created by the user to fit their particular needs. Sometimes entity, referential, and domain integrity aren't enough to safeguard data. Often, specificbusiness rules must be considered and integrated into data integrity processes to meet enterprise standards.

Database design is the process of creating a design that will support the organization's mission statement and mission objectives(goals) for the required database system. There are many aspects we talked about that to be taken into account in designing the database including Network design, Data storage, Distributed database types, Architectural models...etc.

DataBase Design approaches: Two main approaches to the design of a database are followed. These are:

1- Top-down approach

- This method is used to build new database.
- The top-down method starts from the general and moves to the specific. Basically, you start with a general idea of what is needed for the system and then ask the end-users what data they need to store.
- Using the top-down method requires that the analyst has a detailed understanding of the system.
- The top-down method also can have shortcomings. In some cases, top-down design can lead to unsatisfactory results because the analyst and end-users can miss something that is important and is necessary for the system.
- The requirements study also specifies that the final system is expected to achieve the objectives of distributed database(performance, availability and reliability, economics, and expandability(flexibility).
- The requirements is to two parallel activities: view design and conceptual design. The view design deals with defining the interfaces of end users and by determine the fundamental functions, also it's called functional analysis. The conceptual design is the process by which the enterprise(organization) is examined to determine entity types, entity attributes and relationships among these entities, it is also called Entity analysis. The results of these two steps need to be cross-referenced to get a better understanding of which function deal with which entities.



Top-Down Approach

2-Bottom-up approach

- The bottom-up approach begins with the specific details of the project and moves up to the general (starts with the attributes then works up by identifying the entities to which the attributes need to be linked, in turn, the relationships among the entities).To begin a bottom-up design, the system analyst will check all the interfaces that the system has, checking reports, screens, and forms. The analyst will work backward through the system to determine what data should be stored in the database based on the analysis of the front end.

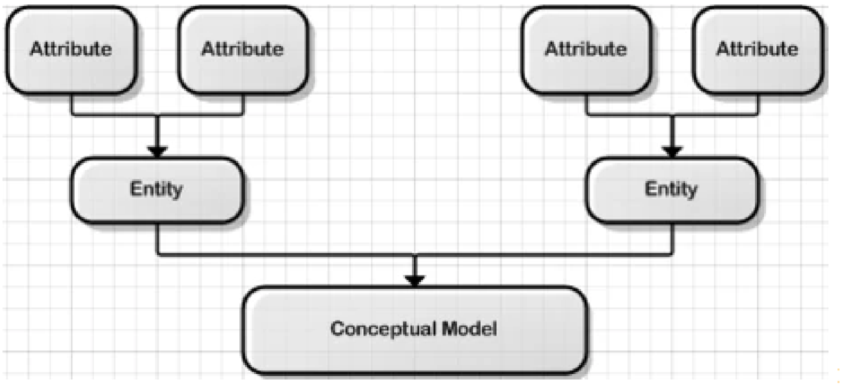
- This type used when:

  - ☐ we try to migrate data from one system to another. For example when we want to migrate from Oracle to SQL: server usually requires some changes as the column data types are not completely compatible.

  - ☐ You would create tables based on existing tables.

  - ☐ Some times you try to near exact copy, to minimize the application coding changes.

  - ☐ you change the table structure, usually to normalize further or to group columns together in a more logical way.



Bottom-up approach

To understand the differences between these approaches, let's consider some jobs that are bottom-up in nature. In statistical analysis, analysts are taught to take a sample from a small population and then conclude the results to the overall population. Physicians are also trained in the bottom-up approach. Doctors examine specific symptoms and then conclude the general disease that causes the symptoms.

An example of jobs that require the top-down approach include project management and engineering tasks where the overall requirements must be specified before the detail can be understood. For example, an automobile manufacturer must follow a top-down approach to meet the overall specifications for the car. If a car has the requirement that it costs less than 15,000 dollars, gets 25 miles per gallon, and seating five people. In order to meet these requirements the designers must start by creating a specification document and then drilling down to meet these requirements.

Analysts task:

The analyst will have no choice but to talk and work with the users to determine what is important to the users and as a result, determines what data should be stored in the database. What the analyst usually does is create some prototype reports, screens, and forms to help the users visualize what the system will look like and how the system will work.

What is Concerency control in a DBMS?

It is the method of managing concurrent transactions(operations) on the database without getting any obstruction with one another.

Why we need Concurrency control?

- To enforce isolation among transactions.
- To preserve database consistency through consistency preserving execution of transactions.
- To resolve read-write and write-read conflicts.

Example: In the concurrent execution environment if T1 conflicts with T2 over a data item A, then the existing concurrency control desides if T1 or T2 should get the A and if the other transaction is rolled-back or waits.

Concurrency Control Protocols

1- Lock Based Concurrency Control Protocol.

2-Time Stamp Concurrency Control Protocol.

3- Validation Based Concurrency Control Protocol.

1- Lock Based Concurrency Control Protocol In this type of protocol, any transaction cannot read or write data until it acquires an appropriate lock on it. There are two modes of lock:

1. Shared lock/ Lock-s(Read lock)

- It is also known as a Read-only lock. In a shared lock, the data item can only read by the transaction.
- More than one transaction can apply share lock on X for reading it's value, but no write lock can be applied on X by any other transaction.

2. Exclusive lock/ lock-x(Write lock)

- In the exclusive lock, only one write lock on X can exist at any time and no shared lock can be applied by any other transactions on X.

- **Lock conversion :** Converting the mode of lock from one to another, there are two states for conversion:

1- Lock upgrade: Existing read lock to write lock

```
If Ti has a read-lock(X) and Tj has no read-lock(X) (i≠j) then
   convert  read-lock() to write-lock(X)
else
force Ti to wait until Tj unlocks X
```

2-Lock downgrade: Existing write lock to read Lock.

Ti has a write-lock(X) ("no transition can have any lock on x")

convert write-lock(X) to read-lock(X)

The locked protocol have two phases(So, it's called two-phase locking :

1- Locking(Growing)phase :

A transaction applies lock (Read or Write) on desired data items one at a time.

2- Unlocking(Shrinking)phase:

A transaction unlocks its data items one at a time.

For a transaction, these two phases mustbe mutually exclusively , that is during locking phase:Unlockung phase must not start, during Unlocking phase: Locking phase must not begin.

DeadLock :

Deadlock is a state of a database system having two or more transactions, when each transaction is waiting for a data item that is being locked by some other transaction.

Dealing with DeadLock by concurrency control:

There are two approach of 2-phase policy gererates two locking algorithms:

1-Basic :

Transtion locks data items incremently. This may cause deadlock which is dealt with.

2-Conservative:

Prevents deadlock by locking all data items before transaction begin execution.

3-Strict:

A more stricter version of Basic algorithm where unlocking is performed after a transaction terminates(commit or aborts and rolled-back). This is the most commonly used.

"In the distributed DBMS , the site where the transaction enters is designated as the controlling site. The controlling site sends messages to the sites where the data items are located to lock the items. Then it waits for confirmation. When all the sites have confirmed that they have locked the data items, the transaction starts. If any site or communication link fails, the transaction has to wait until they have been repaired."

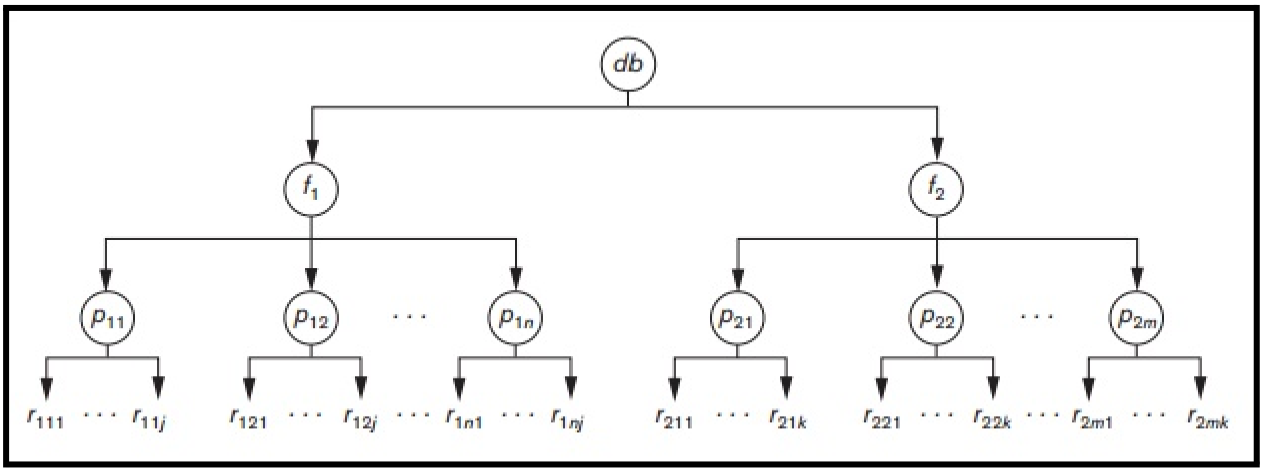
Granularity:

It is the size of data item allowed to lock.

Example of data item granularity:

- A field of a database record (an attribute of a tuple)
- A database record(atuple or relation)
- An entrie file.
- The entrie database.
-

- Alocktable unit of data defines it’s granularity. Granularity can be coarse(entire database) or it can be fine (a tupleor an attribute).
- Data item granularity significantly affects concurrency control performance. Thus, the degree of concurrency is low for coarse granularity and high for fine granularity.


db
f1
f2
p11
p12
. . .
p1n
p21
p22
. . .
p2m
r111 . . . r11j
r121 . . . r12j . . . r1n1 . . . r1nj
r211 . . . r21k
r221 . . . r22k . . . r2m1 . . . r2mk

Granularity of data item diagram

What is Data Recovery?

It is the method of restoring the database to its correct state in the event of a failure at the time of the transaction or after the end of a process.

Failure types:

Failures are generally classified as transaction, system, and media failures. There are several possible reasons for transaction to fail in the middle of execution:

1. **A computer failure (system crash).** A hardware, software, or network error occures in the computer system during transction execution. hardware crashes are usually media failure- for example, main memory failure.
2. **A transction or system error.** Some operation in transition may cause it to fail, such as integer overflow or division by zero. Transaction failure may also occur because of error. A additionally, the user may interrupt the transaction during it's execution.
3. **Local errors or exception conditions detected by the transactio**n. During transaction execution, certain conditions may occur that necessitate cancelation of the transaction. For example, data for transactions may not be found. An exception condition, such as insufficient account balance in a banking database, may cause a transaction, such as a fund withdrawal, to be canceled.
4. **Concurrency control enforcement**. The concurrency method may decide to abort one or more transactions to resolve a state of deadlock among several transactions.
5. **Disk failure**. Some disk blocks may lose their data because of a read or write malfunction or because of a disk read/write head crash. This may happen during a read or a write operation of the transaction.
6. **Physical problems and catastrophes.** This refer to an endless list of problems that includes power or air-conditioning failure, fire, theft....etc.

Recovery techniques:

The data recovery techniques in DBMS make sure, that the state of data is preserved to protect the atomic property and the data is always recoverable to protect the durability property. The following techniques are used to recover data in a DBMS:

1- Log-based recovery

- Recovery through Deferred Update
- Recovery through Immediate Update

2- Checkpoints

3- Shadow paging

4- Buffer mangement

1. **Log-based recovery.**

Any DBMS has its own system logs that have the records for all the activity that has occurred in the system. Databases handle different log files for activities like errors, queries, and other changes in the database. The log is stored in the files in the following formats:

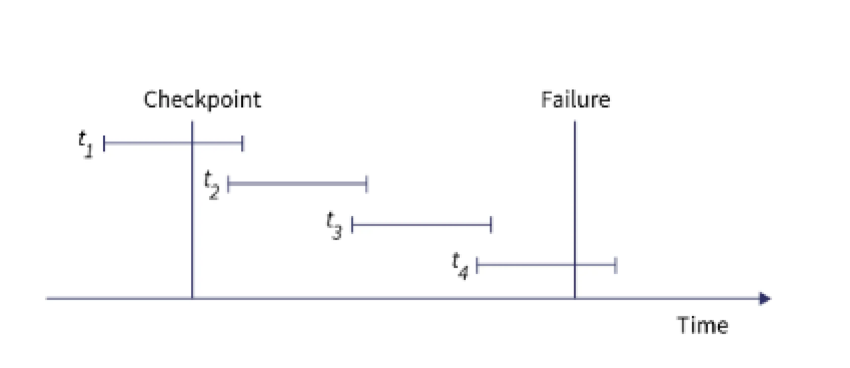
- The structure [start_transaction, T] denotes the start of execution of transaction T.
- [write_item, T, X, old_value, new_value] shows that the value of the variable, X is changed from old_value to new_value by the transaction T.
- [read_item, T, X] represents that the value of X is read by the transaction T.
- [commit, T] indicates the changes in the data are stored in the database through a commit and can't be further modified by the transaction. There will be no error after a commit has been made to the database.
- [abort, T] is used to show that the transaction, T is aborted.

- We can use these logs to see the change in the state of the data during a transaction and can recover the data to the previous state or new state.

An undo operation can be used to examine the [write_item, T, X, old_value, new_value] operation and retrieve the state of data to old_data.

A redo operation can be done to convert the old state of data to the new state that was lost due to system failure and is only possible if the [commit, T] operation is performed.

Ex: Consider multiple transactions named t1,t2,t3, and t4 as shown in the image below. A checkpoint at a time during the first transaction and the system fails during the fourth transaction, but it is possible to recovery the data to the state of the checkpoint made during t1.



A checkpoint is made after all the records of a transaction are written to logs to transfer all the logs from the local storage to the permanent storage for future use.

- **Recovery through Deferred Update**

It is a technique for the maintenance of the transaction log files of the DBMS. It is also called NO-UNDO/REDO technique. It is used for the recovery of transaction failures that occur due to power, memory, or OS failures. Whenever any transaction is executed, the updates are not made immediately to the database. They are first recorded on the log file and then those changes are applied once the commit is done. This is called the “Re-doing” process. Once the rollback is done none of the changes are applied to the database and the changes in the log file are also discarded. If the commit is done before crashing the system, then after restarting the system the changes that have been recorded in the log file are thus applied to the database.

- **Recovery through Immediate Update**

Immediate Update: It is a technique for the maintenance of the transaction log files of the DBMS. It is also called UNDO/REDO technique. It is used for the recovery of transaction failures that occur due to power, memory, or OS failures. Whenever any transaction is executed, the updates are made directly to the database and the log file is also maintained which contains both old and new values. Once the commit is done, all the changes get stored permanently in the database, and records in the log file are thus discarded. Once rollback is done the old values get restored in the database and all the changes made to the database are also discarded. This is called the “Un-doing” process. If the commit is done before crashing the system, then after restarting the system the changes are stored permanently in the database.