UNIT - I

DATA

- Data are facts that can be collected through observation & measurement

• For example: amount, roll no, name, telephone no etc.

• The data are organized in the form of characters, fields, record, files and databases.

There are 2 types of data:

1. It is the collection of information needed by the organization.

2. Metadata - is information about the data i.e., data about data.

DATABASE

• A database is a collection of data elements (facts) stored in a computer in a systematic way

DATABASE MANAGEMENT SYSTEM(DBMS)

▪ A Database Management System (DBMS) is a software package to facilitate the creation and maintenance of a computerized database.

▪ Functionalities related to DBMS are:

▪ Creation of database

▪ Insertion of data into DB

▪ Updating the data in DB

▪ Effective retrieval of data from DB

▪ Securing of data stored in DB

▪ Deletion of data from DB

▪ The Database Management System is largely called as DBMS. This DBMS basically contains three important terms

- Database:
  - Storage of data in large quantity
- Management: Database handling activities
  - ▪ Creation of database
  - ▪ Insertion of data into DB
  - ▪ Updating the data in DB
  - ▪ Effective retrieval of data from DB
  - ▪ Securing of data stored in DB
  - ▪ Deletion of data from DB
- System: A complex program or software

Complex software that performs management activities like "creation, addition of data, modifying, removing data and accessing data" on a database is called as Database Management System

Introduction To Database :

▪ Database is a storage that holds large amount of data.
▪ The stored data is used to produce information as and when required.
▪ Generally speaking, the database is nothing but a storage place of data which is of interest to the enterprise.
▪ The database is physically present on the secondary storage or disk. So, the data stored in the database remains permanent so that it can be used later. The quantity of data stored in database depends on the size of the enterprise

The functionalities related to the database

1. **Creation of the database**
   - ω Creation of the database starts from the planning.
   - ω When the idea of preparing to store the data arrives, then a model is prepared.
   - ω On the basis of this model preparation of database is done. Preparing this physically is called as creation of data- base.
   - ω The database created contains only the structure or framework. The created database is ready to hold the data
2. **Insertion of data into the database**
   - ω Once the database is created the next function is to put data into it. Entering data into the database is done through the insertion operation. So, putting data into the database is called insertion of data.
3. **Updating data in the database**
   - ω The data stored once may be the basic facts of the activities of the enterprise. Some data need to be derived from these existing data. Some may require change as per the operations. All these activities come under updating functionality of the database
4. **Effective retrieval of data from the database**
   - ω Once the data is finalized in the database then the functionality which arrives is its retrieval.
   - ω The data stored in the database is retrieved for the purpose of producing the information.
   - ω The retrieval of data must be such that the response time must be as low as possible. Such retrieval is called as effective data retrieval functionality of the database.
5. **Securing of data stored in the database**
   - ω The data stored in the database is used by many people of the enterprise. So, there is every chance losing authenticity of the data.
   - ω The permitted users of the database must access the data. It is necessary that proper security mechanisms must be used to ensure it. This facility of providing security is called securing the data functionality of the database.
6. **Deletion of data from the database**

ω When the data is of no use to the enterprise, that is it has become obsolete, it must be removed from the database. Removing the old data from the database is called as deletion of data functionality of the database

The storage of data largely depends on the basic structure of the database. The basic structures may be of three types.

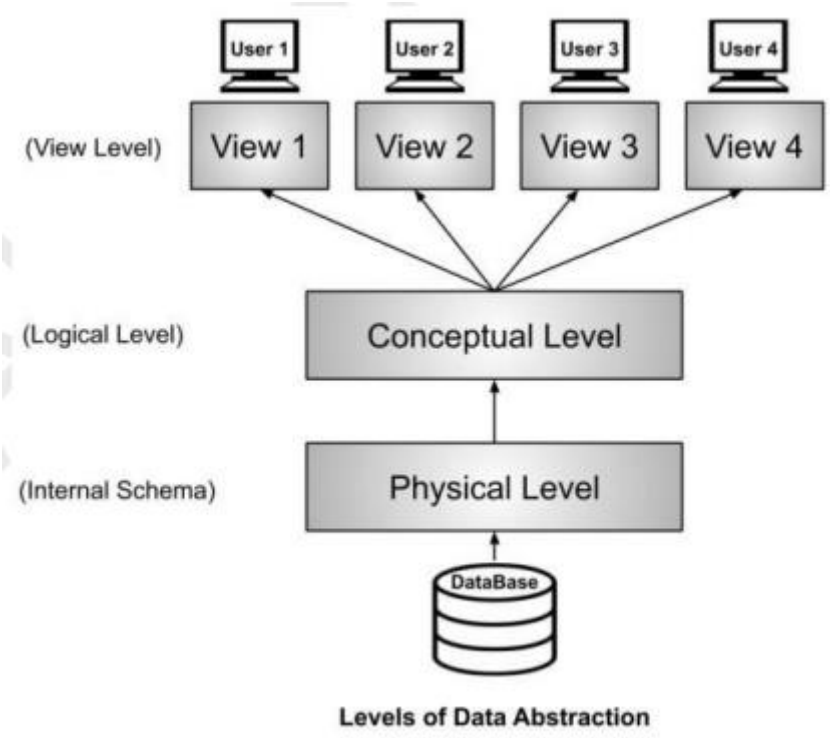
1. **RELATIONAL DATABASE MANAGEMENT SYSTEM"**
   ω If the basic structure is a collection of "flat files" (two-dimensional tables) for basic objects (entities) as well as for the relationships or associations between the objects (entities) then the database is termed as relational database.
   ω The database management system that manages such relational database is called as "Relational Database Management System" (RDBMS).
2. **HIERARCHICAL DATABASE MANAGEMENT SYSTEM**
   ω If the basic structure of the database is "tree structure" then the database is termed as hierarchical database.
   ω The system that manages such database is "Hierarchical Database Management System" (HDBMS)
3. **NETWORK DATABASE MANAGEMENT SYSTEM**
   ω If the basic structure of the database is "graph structure" then the database
   ω is termed as network database. The system that manages such database is "Network Database Management System" (NDBMS).

VIEWS OF DATA

- The provision given by the database management system to view the data by the people or users of enterprise is called as views of data.
- Indirectly the way how the people of the enterprise look for the data in the database is called as views of data.
- The main function of the database management system is to provide users a concrete view of data. This concrete view is called as abstract view. The meaning of the word abstract here is, 'to hide the details

DATA ABSTRACTION

The database management system developers hide the complexity of data in the system. It is done through several levels of abstraction to simplify user interaction with the system

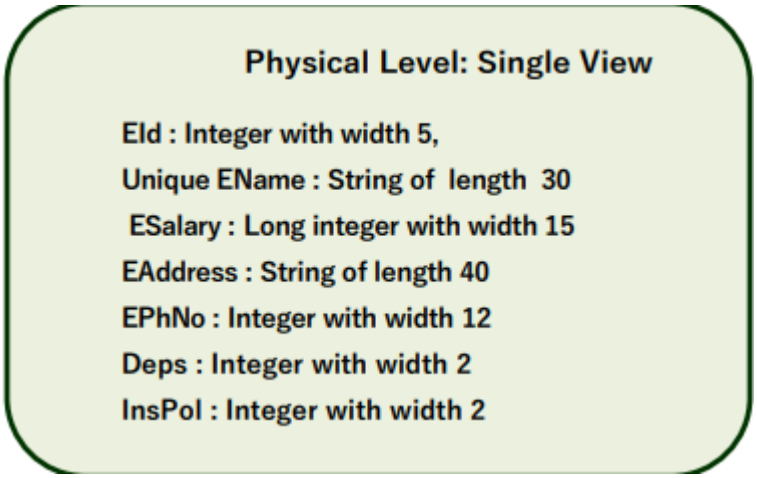


Levels of Data Abstraction

The details of the data are hidden at the THREE levels namely Physical Level, Logical Level and View Level. The View Level is at the highest level of data abstraction whereas the Physical Level is the lowest level. In between comes the Logical Level. Starting from the lowest level the levels of abstraction details are as follows

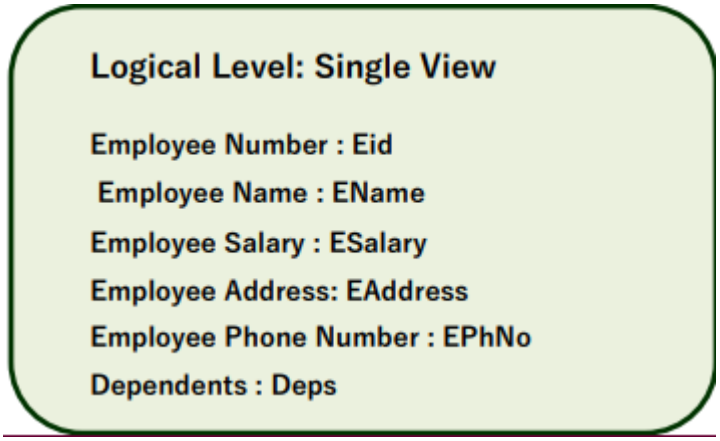
Physical level :

- This is the lowest level of data abstraction.
- It describes the actual storage of data.
- The complex low-level data structures are detailed in this level.
- This level of data view is normally available only to the Database Administrator.
- The details of this level are hidden normally to the Application Programmers and other inexperienced users of DBMS



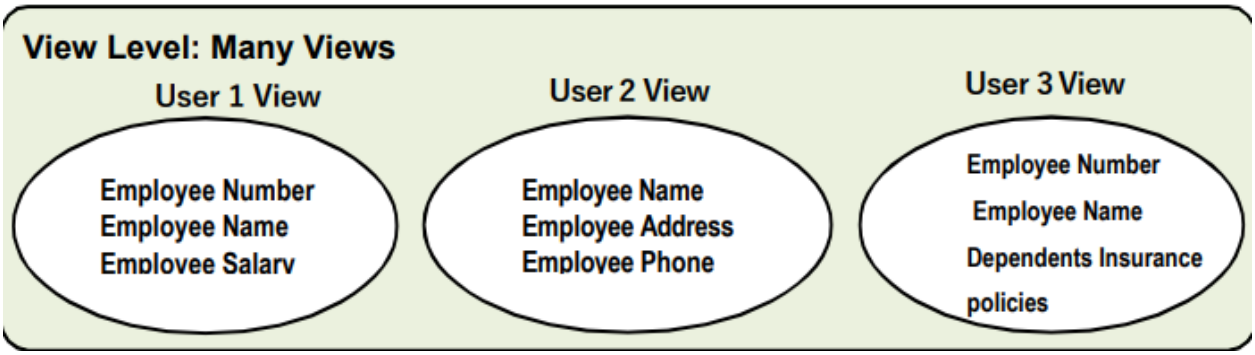
LOGICAL LEVEL

- Next higher level of data abstraction is Logical Level.
- It describes the data stored in the database.
- It also describes the relationships or associations among those data.



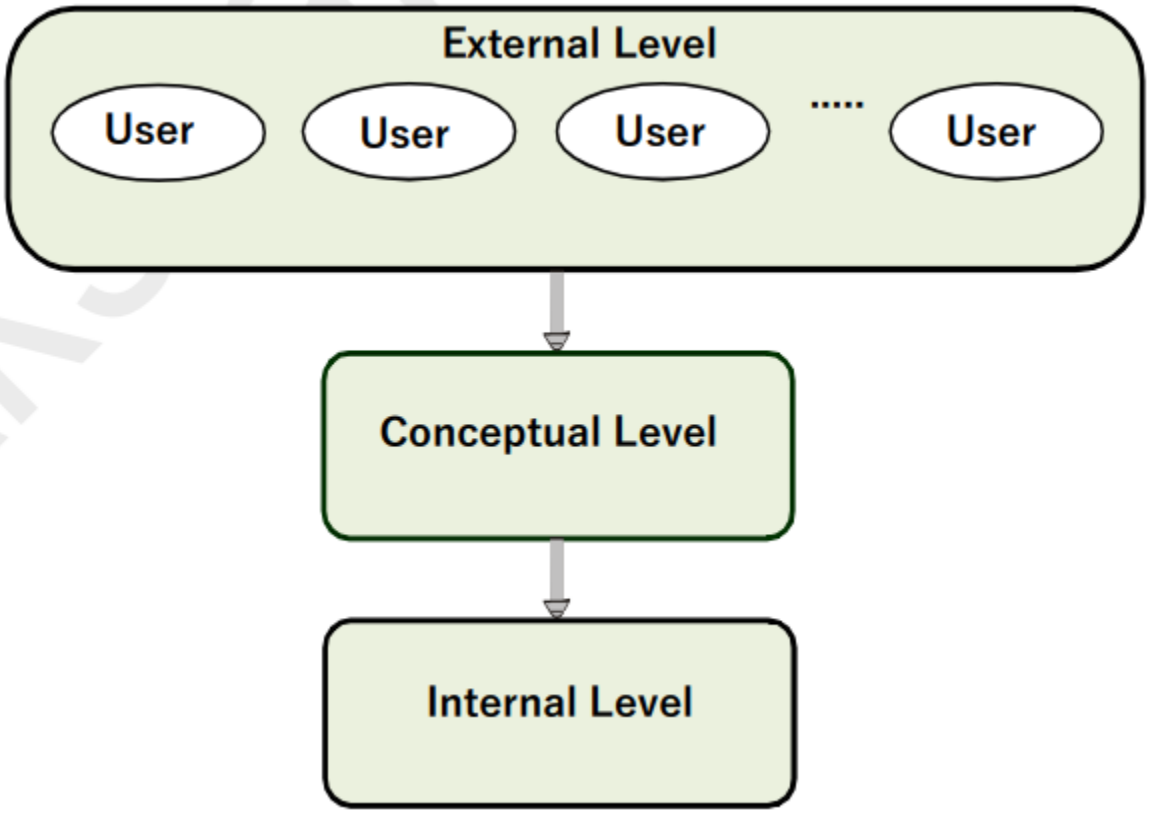
VIEW LEVEL :

- This is the highest level of data abstraction. It provides many views for the same database.
- The users at this level need to access only a part of database. So, their interaction with the system is simplified at this level.



Three-level architecture proposal:

- Using the concept of data abstraction which is at three levels, the architecture of the DBMS can also be proposed at three levels. The levels of the architecture are External Level, Conceptual Level and Internal Level.
- The following figure depicts the 3-tier or 3-level architecture of Database Management System.



1. **External level**

- An external level specifies a view of the data in terms of conceptual level tables.
- Each external level view is used to cater to the needs of a particular category of users.
- For Example, FACULTY of a university is interested in looking course details of students, STUDENTS are interested in looking at all details related to academics, accounts, courses and hostel details as well. So, different views can be generated for different users.
- The main focus of external level is data abstraction. The users at this level think that the database is serving only them. The users are unaware of the other user's requirements. So, even if any change (addition or modification) is made to the database at any next level it is independent.

2. **Conceptual level**

- At conceptual level, data is represented in the form of various database tables.
- For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage.
- Also referred as logical schema, it describes what kind of data is to be stored in the database.

3. **Internal level:**

- At the physical level, the information about the location of database objects in the data store is kept.
- Various users of DBMS are unaware of the locations of these objects.
- In simple terms, physical level of a database describes how the data is being stored in secondary storage devices like disks and tapes and also gives insights on additional storage details

Data independence means a change of data at one level should not affect another level. Two types of data independence are present in this architecture

1. **Physical data independence**:

   Any change in the physical location of tables and indexes should not affect the conceptual level or external view of data. This data independence is easy to achieve and implemented by most of the DBMS.

2. **Conceptual data independence:**

   The data at conceptual level schema and external level schema must be independent. This means a change in conceptual schema should not affect external schema. e.g.; Adding or deleting attributes of a table should not affect the user's view of the table. But this type of independence is difficult to achieve as compared to physical data independence because the changes in conceptual schema are reflected in the user's view

INSTANCES AND SCHEMAS:

- Design of a database is called the schema. Schema is of three types: Physical schema, logical schema and view schema.

- The data stored in database at a particular moment of time is called instance of database.
- For example, lets say we have a single table student in the database, today the table has 100 records, so today the instance of the database has 100 records.

  Lets say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table.

Purpose of database system

1. **Data redundancy**
   - Data redundancy refers to the duplication of data

▪ let's say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed.

▪ Data redundancy often leads to higher storage costs and poor access time.

2. **Data inconsistency**
   - Data redundancy leads to data inconsistency
   - Let's take the same example that we have taken above, a student is enrolled for two courses and we have student address stored twice, now let's say student requests to change his address, if the address is changed at one place and not on all the records then this can lead to data inconsistency
   - Data redundancy refers to the duplication of data
   - let's say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed.
   - Data redundancy often leads to higher storage costs and poor access time.
3. **Atomicity issues**
   - Atomicity of a transaction refers to "All or nothing", which means either all the operations in a transaction executes or none.
   - For example: Let's say Steve transfers 100$ to Negan's account. This transaction consists multiple operations such as debit 100$ from Steve's account, credit 100$ to Negan's account. Like any other device, a computer system can fail let's say it fails after first operation then in that case Steve's account would have been debited by 100$ but the amount was not credited to Negan's account, in such case the rollback of operation should occur to maintain the atomicity of transaction. It is difficult to achieve atomicity in file processing systems
4. **Data security** :
   - Data should be secured from unauthorized access
   - for example, a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.

Advantages of database management system :

1. **Centralized management of data**
   - The database management system (DBMS) helps in placing every data of the enterprise at only one place centrally in the form of a database
2. **Sharing of data**
   - When the data is stored centrally stored, it can be shared among many people of the enterprise or organization or institution depending on their requirements.
   - The Director, the Managers, the Supervisors, the Workers, the Customers and the Investors can access the database for their use but in restricted or controlled manner
3. **Independence of data**
   - The data stored in the database of DBMS is independent at various levels of access.
   - The application programmers or users view the data in different way. It is external to them.
   - Their view of data is independent of how logically the database is prepared by the database designer. It is logical data independence.
   - For the database designer as well as external users again, it is independent the way how it is physically stored on the secondary storage or physical device used. It is physical data independence.

4. **Non Redundancy of Data :**
   - Repetition of the same data at more than one place is called as data redundancy.
   - For the sake to avoid some loss of information, the data can be duplicated by the database designer as per the requirement.
   - The process of removing the redundancy of data is called as non-redundancy of data. It helps in saving the physical storage space, processing time as well as communication cost.

5. **Ease of data access :**
   - As the data is stored at one place it can be accessed easily by all the users of database.

- This access can be granted and controlled by the administrator depending on the user and his level of access.
- Efficient techniques can be imposed for effective retrieval of data very easily. This ease of access helps in providing a good response time as well as increased through put

6. **Data consistency and integrity :**

- Consistency in database systems refers to the requirement that and given database transaction must change affected data only in allowed ways

7. **Imposing proper security :**

- The data in the database can be clearly secured in better way at various levels.
- Both internal as well as external security on database can be easily imposed. The DBMS provides security at database level.

8. **Ease of recovery**
   - DBMS provides techniques to recover the data stored in database when system crashes because of some failures. It is very easy for the administrator to recover the data whenever it is required.
9. **Simultaneous access :**
   - As the data is shared by many people, it can be even accessed simultaneously. Such simultaneous access is called as concurrent access.
   - By means of some controlling measures consistency of the data can be maintained. Such simultaneous access increases the speed of transaction processing.

Disadvantages of dbms :

1. **Implementation cost :**
   - The cost incurred in the development and purchase of software is much higher in comparison to convention file-handling.
   - It is necessary to upgrade the hardware accordingly which is again a costly affair. The cost is also incurred in providing workspace for the execution of complex programs along with storage cost.

2. **Processing overhead to implement security and integrity**
   - The security and the integrity are the major requirements of DBMS that prove the worthiness of its implementation.
   - Implementation of proper security and integrity involves more processing overhead on the DBMS
3. **Degradation of response time and throughput**

   - The database is being shared by many users which is the major advantage of DBMS.
   - The time required to get the result or data from the DBMS is called as response time. When more users attempt for the same database for the data items, the response time is quite large. It is fast in one-to-one access.
   - When the response time is large the throughput is reduced. Number of users getting response in a given unit of time from the DBMS is called as throughput.
4. **Change over cost**
   - The users of conventional file-handling take time in responding to the new system. When DBMS is new to the enterprise, the people of the enterprise spend much time in adopting to it.
5. **Backup requirement**
   - To reduce the duplication of data the complete data of an enterprise is centrally placed as at one place of course using a single database.
   - This forces the organization to prepare a backup of the database keeping the failure of the system in mind.
   - In case of failure the database can be recovered from the backup. In DBMS the recovery and backup are more complex.
6. **Down times and failures**
   - The data in the enterprise is centrally organized and placed at one location as database. So, the enterprise must bear the cost at down times of DBMS and during the failure of DBMS till the recovery. Naturally even the computing system also tends to some failure.

Structure of DBMS

DBMS

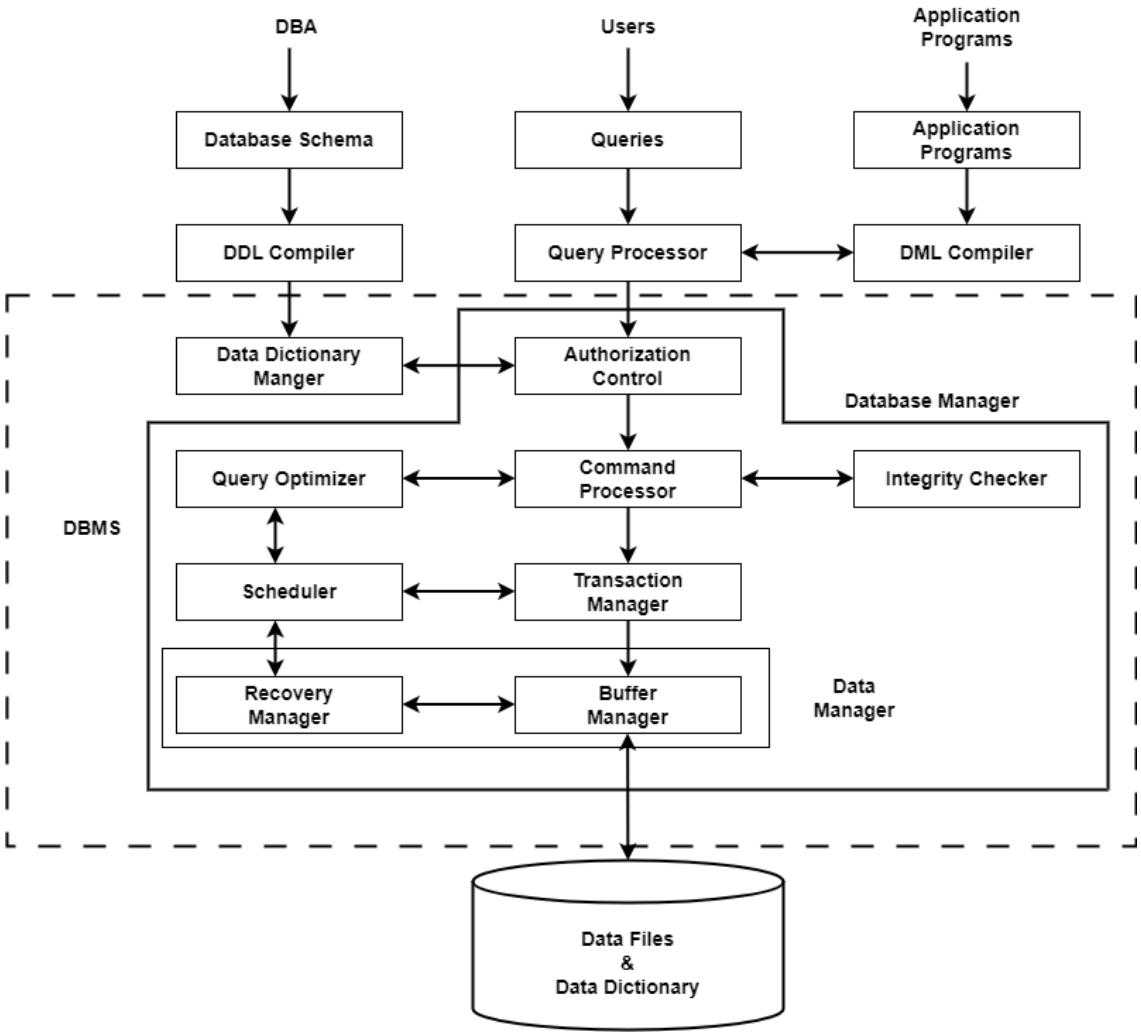
DBMS means Database Management System, which is a tool or software used to create the database or delete or manipulate the database. A software programme created to store, retrieve, query, and manage data is known as a Database Management System (DBMS). Data can be generated, read, updated, and destroyed by authorized entities thanks to user interfaces (UIs).

Because they give programmers, Database Managers, and end users a consolidated view of the data, Database Management Systems are crucial because they relieve applications and end users of the need to comprehend the physical location of the data. Application Programme Interfaces (APIs) manage internet requests and responses for particular sorts of data.

In marketing materials, the phrase "database as a service" (DBaaS) may be used to refer to both relational and non-relational DBMS components that are given via the internet.

Users of DBMSs include application programmers, Database Administrators (DBAs), and end users.

Database Administrators are typically the only people who work directly with a DBMS. Today, end users read and write to databases using front-end interfaces made by programmers, while programmers use cloud APIs to connect with DBMSs.



Three Parts that make up the Database System are:

- Query Processor
- Storage Manager
- Disk Storage

The explanations for these are provided below:

1. Query Processor

The query processing is handled by the query processor, as the name implies. It executes the user's query, to put it simply. In this way, the query processor aids the database system in making data access simple and easy. The query processor's primary duty is to successfully execute the query. The Query Processor transforms

(or interprets) the user's application program-provided requests into instructions that a computer can understand.

Components of the Query Processor

- **DDL Interpreter:**

Data Definition Language is what DDL stands for. As implied by the name, the DDL Interpreter interprets DDL statements like those used in schema definitions (such as create, remove, etc.). This interpretation yields a set of tables that include the meta-data (data of data) that is kept in the data dictionary. Metadata may be stored in a data dictionary. In essence, it is a part of the disc storage that will be covered in a later section of this article.

- **DML Compiler:**

Compiler for DML Data Manipulation Language is what DML stands for. In keeping with its name, the DML Compiler converts DML statements like select, update, and delete into low-level instructions or simply machine-readable object code, to enable execution. The optimization of queries is another function of the DML compiler. Since a single question can typically be translated into a number of evaluation plans. As a result, some optimization is needed to select the evaluation plan with the lowest cost out of all the options. This process, known as query optimization, is exclusively carried out by the DML compiler. Simply put, query optimization determines the most effective technique to carry out a query.

- **Embedded DML Pre-compiler:**

Before the query evaluation, the embedded DML commands in the application program (such as SELECT, FROM, etc., in SQL) must be pre-compiled into standard procedural calls (program instructions that the host language can understand). Therefore, the DML statements which are embedded in an application program must be converted into routine calls by the Embedded DML Pre-compiler.

- **Query Optimizer:**

It starts by taking the evaluation plan for the question, runs it, and then returns the result. Simply said, the query evaluation engine evaluates the SQL commands used to access the database's contents before returning the result of the query. In a nutshell, it is in charge of analyzing the queries and running the object code that the DML Compiler produces. Apache Drill, Presto, and other Query Evaluation Engines are a few examples.

2. Storage Manager:

An application called Storage Manager acts as a conduit between the queries made and the data kept in the database. Another name for it is Database Control System. By applying the restrictions and running the DCL instructions, it keeps the database's consistency and integrity. It is in charge of retrieving, storing, updating, and removing data from the database.

Components of Storage Manager

Following are the components of Storage Manager:

- **Integrity Manager:**

Whenever there is any change in the database, the Integrity manager will manage the integrity constraints.

- **Authorization Manager:**

Authorization manager verifies the user that he is valid and authenticated for the specific query or request.

- **File Manager:**

All the files and data structure of the database are managed by this component.

- **Transaction Manager:**

It is responsible for making the database consistent before and after the transactions. Concurrent processes are generally controlled by this component.

- **Buffer Manager:**

The transfer of data between primary and main memory and managing the cache memory is done by the buffer manager.

3. Disk Storage

A DBMS can use various kinds of Data Structures as a part of physical system implementation in the form of disk storage.

Components of Disk Storage

Following are the components of Disk Manager:

- **Data Dictionary:**

It contains the metadata (data of data), which means each object of the database has some information about its structure. So, it creates a repository which contains the details about the structure of the database object.

- **Data Files:**

This component stores the data in the files.

- **Indices:**

These indices are used to access and retrieve the data in a very fast and efficient way.

Data Model :

- A collection of conceptual or logical tools that is used to describe the following is called as data model.

  • the data
  • the relationships of data
  • the meaning of data
  • the consistency conditions or constraints

  The data models collectively grouped into the major groups like 'Object based conceptual Models' and 'Record based conceptual models.

  Object based conceptual models

  • Object based conceptual models are used to describe the data model at the logical view level and external view levels.

  • These types of models are flexible enough to explain the structuring capabilities.

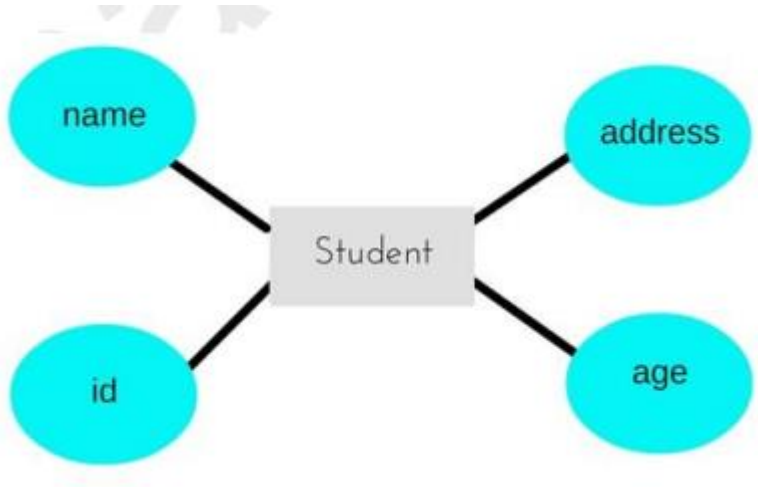
  • The data conditions of consistency are explicitly specified in addition to the structuring of data.

• The object based conceptual models include:

ER (Entity Relationship) Model and Object-Oriented Model.

1. **ER model**

   ▪ In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.
   ▪ Different entities are related using relationships.
   ▪ E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.
   ▪ This model is good to design a database, which can then be turned into tables in relational model
   ▪ Let's take an example, If we have to design a School Database, then Student will be an entity with attributes name, age, address etc.



2. **Object oriented model**

   ▪ The ODBMS which is an abbreviation for object-oriented database management system, is the data model in which data is stored in form of objects, which are instances of classes. These classes and objects together make an object-oriented data model

Record based conceptual models

• This type of data models is also used to describe the data at conceptual (logical) and view levels. The implementation details are provided at physical levels.

• Different types of records in respective format are structured to form a database.

• As the database is a collection of such record formats this data model is called as record-based conceptual model.

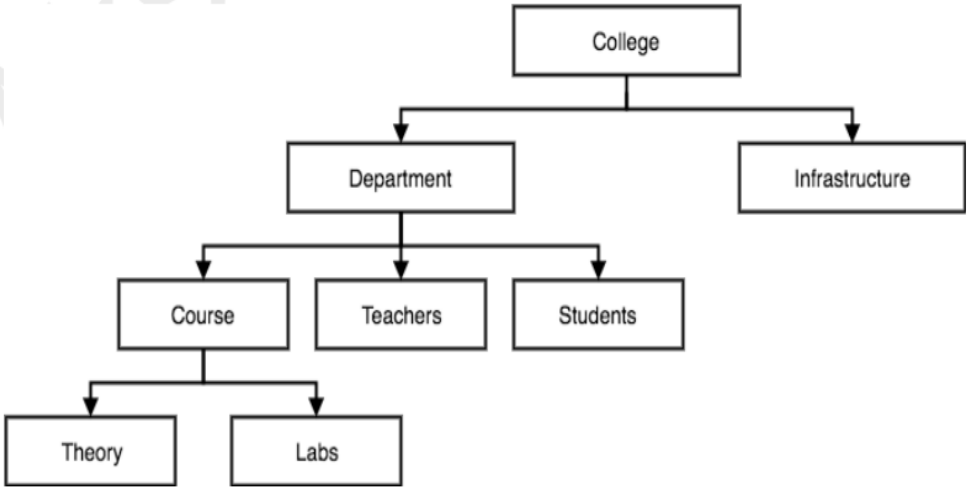
• The models which come under record based conceptual models list are Hierarchical, Network and Relational model. Among these three models the Relational Model is most widely accepted data model.

1. Hierarchical model

▪ This database model organizes data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes.

▪ In this model, a child node will only have a single parent node. ▪ This model efficiently describes many real-world relationships like index of a book, recipes etc.

▪ In hierarchical model, data is organized into tree-like structure with one one-to many relationship between two different types of data, for example, one department can have many courses, many professors and of course many students.

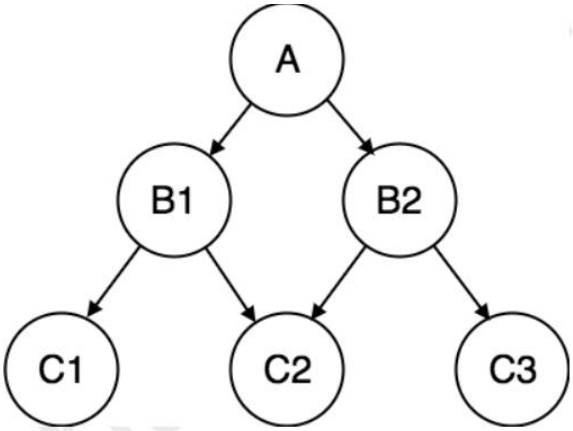


2. NETWORK MODEL :

▪ This is an extension of the Hierarchical model. In this model data is organized more like a graph, and are allowed to have more than one parent node.

▪ In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

▪ This was the most widely used database model, before Relational Model was introduced.

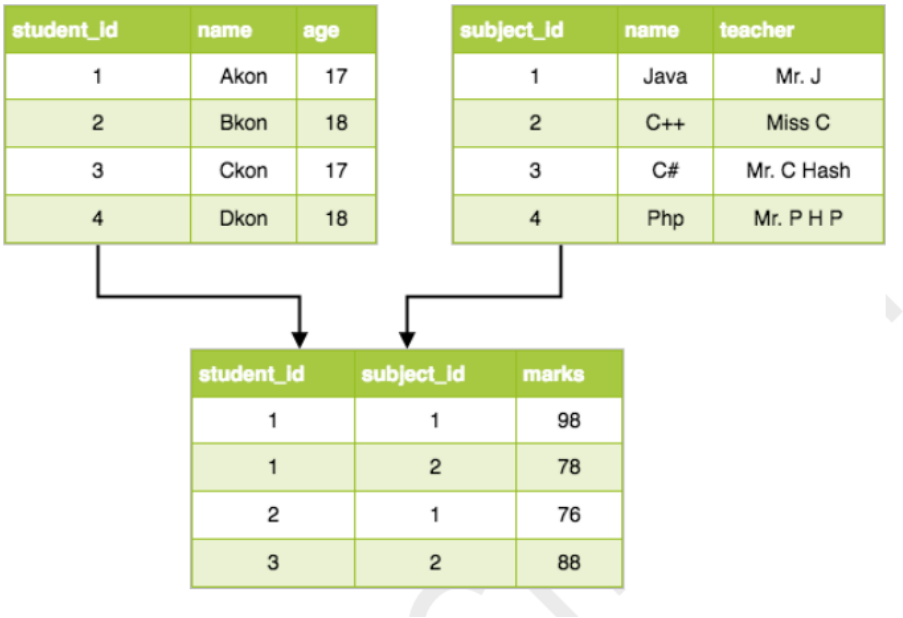


3. RELATIONAL MODEL :

▪ In this model, data is organized in two-dimensional tables and the relationship is maintained by storing a common field.

▪ This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, in fact, we can say the only database model used around the world.

▪ The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table. ▪ Hence, tables are also known as relations in relational model.



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

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

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

Database languages :

1. **Data Definition Language(DDL):**

   - CREATE: Create new database, table, etc.
   - ALTER: Alter existing database, table, etc.
   - DROP: Drop the database
   - RENAME: Set a new name for the table.

2. **Data Manipulation Language (DML)**
   - SELECT: Retrieve data from the database
   - INSERT: Insert data
   - UPDATE: Update data
   - DELETE: Delete all records

3. **Data Control Language (DCL)**
   - GRANT: Give privilege to access the database.
   - REVOKE: Take back the privilege to access the database.

4. **Transaction Control Language**
   - COMMIT: Save the work.
   - SAVEPOINT: Set a point in transaction to rollback later
   - ROLLBACK: Restores since last commit