UNIT - 1

A Historical Perspective

Data is a collection of facts and figures. The data collection was increasing day to day and they needed to be stored in a device or software which is safer.

Database Management Systems (DBMS) have been around for several decades, and their history can be traced back to the early 1960s. In the early days, computer systems were designed to manage data in a hierarchical or navigational manner, where data was stored in a tree-like structure. This method of storing data was inefficient and difficult to use, as it required a lot of manual effort to access and manage the data.

In the late 1960s, The first general-purpose DBMS, designed by **Charles Bachman**, was called the **Integrated Data Store (IDS)** which was based on network data model for which he was received the **Turing Award** (The most prestigious award which is equivalent to Nobel prize in the field of Computer Science.).

In the late 1970s, **Mr Edgar Codd** proposed a new data representation framework called the **Relational Database Model**. Mr Edgar Codd won the 1981 Turing Award for his seminal work. This model was based on the concept of a table, with rows representing individual records and columns representing individual fields within those records. The relational model allowed for more efficient storage and retrieval of data and was easier to use than the hierarchical or navigational models.

In the late 1980s IBM developed the **Structured Query Language (SQL)** for relational databases, as a part of R project. This system was designed to manage large amounts of data and was used primarily in corporate and government applications. SQL was adopted by the American National Standards Institute (ANSI) and International Organization for Standardization (ISO).

In the 1980s, several new DBMS products were introduced, including Oracle, Sybase, and Microsoft SQL Server. These systems were designed to be more user-friendly and to support more advanced data modeling and query languages.

In the 1990s, **object-oriented DBMS (OODBMS)** emerged, which were designed to store and manage complex data structures, such as multimedia and other types of non-traditional data. These systems were initially popular in research and academic environments, but their adoption was limited in the commercial sector.

In the 1991, **Microsoft ships MS access**, a personal DBMS and that displaces all other personal DBMS products.

In the 1997, **XML** applied to database processing. Many vendors begin to integrate XML into DBMS products.

In the 2000s, web-based applications and cloud computing became more popular, and DBMS systems began to adapt to these new technologies. New DBMS systems were developed to support distributed and web-based applications, including NoSQL databases such as MongoDB and Cassandra.

Today, DBMS systems continue to evolve, with an emphasis on scalability, performance, and support for cloud-based applications. Some of the most popular DBMS systems in use today include Oracle, Microsoft SQL Server, MySQL, PostgreSQL, and MongoDB.

Introduction to Database Management System (DBMS)

A data is anything in a form of Quantity, character or a symbol suitable for use in a computer. Data is distinguished from program. A program is a set of data to perform a task in a computer.

A database is a set of instructions or a collection of related data from which users can be easily retrieve and accessed the desired information.

A database system is designed to managing the data or a large amount of information. A database system is an integrated collection of related files. Management of data involves structures for storage of information and to ensure about the safety of information stored.

A database management system (DBMS) is a system or a program that allows users to access the collection of data which stored in a database. The main objective of DBMS is to provide a convenient and effective technique of storing and retrieving the database information.

The database and database management systems plays an important role in day to day life for storing and retrieving an information. It is very essential for the management of banking, restaurants, government organizations, schools and colleges, etc.

Data, Information and Meta Data:
Data and information are closely related to each other and often used interchangeably. Information is nothing but processed or refined data.

The term data is anything in a form of Quantity, character or a symbol suitable for use in a computer.

Information is a data can be put into meaningful and useful content. Information consists of text, image, audio, video, etc.

Meta data is a data about data. It is a structural component which is describe in the form of tables and their elements.

Database Management System (DBMS):
Database is a collection of interrelated data stored to the server. Databases are organized by fields, records and files. It is designed to managing the data or a large amount of information.

Database Management System (DBMS) is a program or a group of programs which can be work in an operating system to manage, store and process the data. It is a system or a program that allows users to access the collection of data which stored in a database. It acts as an interface between the program and the data stored in a database system.

Database System Applications:
Database systems are widely used around the world. Following are the various applications of database management systems:

1. Banking: It is used to store the customer information such as name, account details, transactions, enquiries, loan details, credits, debits, etc.

2. Universities: Database is widely used in universities to store the information of a students, admission, registrations, data of results, etc.

3. Airlines: Airlines are the first to use databases. It is used to storing information about the seats, reservations, information of passengers.

4. Railways: Database is used widely in railways to storing the information about passengers, reservations, seats availability, etc.

5. Sales: It is used to store the information about customers, products, sales and purchase information. Whenever a customer buys any product it can be store into a database.

6. Manufacturing: Database is used widely in manufacturing to store information about sales, purchase, CAD drawings, information of inventory, raw material availability, etc.

7. Telecommunication: Database is used in telecommunication to store the information about the phones and mobiles call records, incoming and outgoing calls timing, duration, to generate the charges and store the information to the communication network.

8. Human Resources: Databases are used in human resources widely to storing the information such as the employees' records, salaries, taxes, benefits, etc.

Advantages of Database Management System:

1. DBMS provides an efficient access to store and retrieve data.
2. DBMS provides an abstract view of data independence.
3. DBMS provides facilities for recovering the backup data from software or hardware.
4. DBMS provides availability of updated information for all users.
5. It restrict to the unauthorized access of data. It provides a data security.
6. DBMS provides centralized administration of data.

7. DBMS provides centralized administration of data.
8. It represents the complex relationships of data.

Components of DBMS

There are many components available in the DBMS. Each component has a significant task in the DBMS. A database environment is a collection of components that regulates the use of data, management, and a group of data. These components consist of people, the technique of Handel the database, data, hardware, software, etc. there are several components available for the DBMS.

1. Hardware

- Here the hardware means the physical part of the DBMS. Here the hardware includes output devices like a printer, monitor, etc., and storage devices like a hard disk.
- In DBMS, information hardware is the most important visible part. The equipment which is used for the visibility of the data is the printer, computer, scanner, etc. This equipment is used to capture the data and present the output to the user.
- With the help of hardware, the DBMS can access and update the database.
- The server can store a large amount of data, which can be shared with the help of the user's own system.
- The database can be run in any system that ranges from microcomputers to mainframe computers. And this database also provides an interface between the real worlds to the database.
- When we try to run any database software like MySQL, we can type any commands with the help of our keyboards, and RAM, ROM, and processor are part of our computer system.

2. Software

- Software is the main component of the DBMS.
- Software is defined as the collection of programs that are used to instruct the computer about its work. The software consists of a set of procedures, programs, and routines associated with the computer system's operation and performance. Also, we can say that computer software is a set of instructions that is used to instruct the computer hardware for the operation of the computers.

- The software includes so many software like network software and operating software. The database software is used to access the database, and the database application performs the task.
- This software has the ability to understand the database accessing language and then convert these languages to real database commands and then execute the database.
- This is the main component as the total database operation works on a software or application. We can also be called as database software the wrapper of the whole physical database, which provides an easy interface for the user to store, update and delete the data from the database.
- Some examples of DBMS software include MySQL, Oracle, SQL Server, dBase, FileMaker, Clipper, Foxpro, Microsoft Access, etc.

3. Data

- The term data means the collection of any raw fact stored in the database. Here the data are any type of raw material from which meaningful information is generated.
- The database can store any form of data, such as structural data, non-structural data, and logical data.
- The structured data are highly specific in the database and have a structured format. But in the case of non-structural data, it is a collection of different types of data, and these data are stored in their native format.
- We also call the database the structure of the DBMS. With the help of the database, we can create and construct the DBMS. After the creation of the database, we can create, access, and update that database.
- The main reason behind discovering the database is to create and manage the data within the database.
- Data is the most important part of the DBMS. Here the database contains the actual data and metadata. Here metadata means data about data.
- For example, when the user stores the data in a database, some data, such as the size of the data, the name of the data, and some data related to the user, are stored within the database. These data are called metadata.

People

- The people who control and manage the databases and perform different types of operations on the database in the DBMS.
- The people include database administrator, software developer, and End-user.
- Database administrator-database administrator is the one who manages the complete database management system. DBA takes care of the security of the DBMS, its availability, managing the license keys, managing user accounts and access, etc.
- Software developer- theThis user group is involved in developing and designing the parts of DBMS. They can handle massive quantities of data, modify and edit databases, design and develop new databases, and troubleshoot database issues.
- End user - These days, all modern web or mobile applications store user data. How do you think they do it? Yes, applications are programmed in such a way that they collect user data and store the data on a DBMS system running on their server. End users are the ones who store, retrieve, update and delete data.
- The users of the database can be classified into different groups.
    - i. Native Users
    - ii. Online Users
    - iii. Sophisticated Users
    - iv. Specialized Users
    - v. Application Users
    - vi. DBA - Database Administrator

File Systems versus a DBMS

What is a File system?

A file system is a technique of arranging the files in a storage medium like a hard disk, pen drive, DVD, etc. It helps you to organizes the data and allows easy retrieval of files when they are required. It mostly consists of different types of files like mp3, mp4, txt, doc, etc. that are grouped into directories.

A file system enables you to handle the way of reading and writing data to the storage medium. It is directly installed into the computer with the Operating systems such as Windows and Linux.

What is DBMS?

Database Management System (DBMS) is a software for storing and retrieving user’s data while considering appropriate security measures. It consists of a group of programs that manipulate the database. The DBMS accepts the request for data from an application and instructs the DBMS engine to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

Difference between File System and DBMS

FILE SYSTEM	DBMS
Used to manage and organise the files stored in the hard disk of the computer	A software to store and retrieve the user's data
Redundant data is present	No presence of redundant data
Query processing is not so efficient	Query processing is efficient
Data consistency is low	Due to the process of normalisation, the data consistency is high
Less complex, does not support complicated transactions	More complexity in managing the data, easier to implement complicated transactions
Less security	Supports more security mechanisms
Less expensive in comparison to DBMS	Higher cost than the File system
Does not support crash recovery	Crash recovery mechanism is highly supported

Features of a File system

Here are important elements of the file system:

- It helps you to store data in a group of files.
- Files data are dependent on each other.
- C/C++ and COBOL languages were used to design the files.
- Shared File System Support
- Fast File System Recovery.

Features of DBMS

Here, are essential features of DBMS:

- A user-accessible catalog of data
- Transaction support
- Concurrency control with Recovery services
- Authorization services
- The value of data is the same at all places.
- Offers support for data communication
- Independent utility services
- Allows multiple users to share a file at the same time

Data Models in DBMS

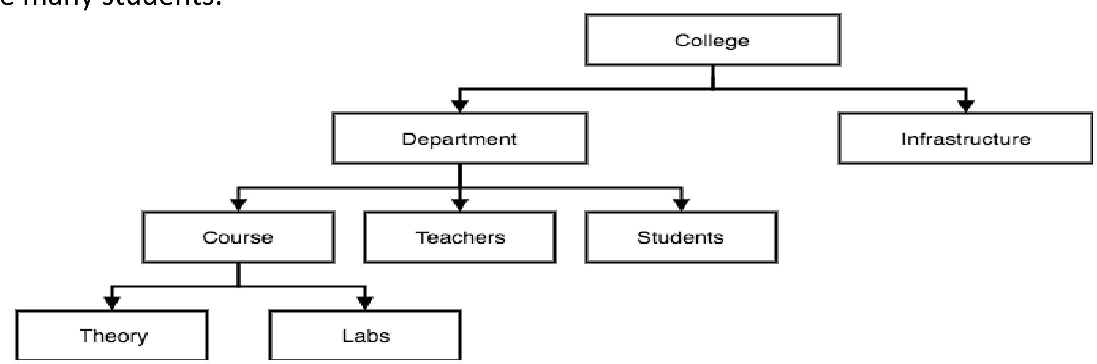
Data models in DBMS help to understand the design at the conceptual, physical, and logical levels as it provides a clear picture of the data making it easier for developers to create a physical database.

Data models are used to describe how the data is stored, accessed, and updated in a DBMS. A set of symbols and text is used to represent them so that all the members of an organization can understand how the data is organized. It provides a set of conceptual tools that are vastly used to represent the description of data.There are many types of data models that are used in the industry.

1. Hierarchical Model

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

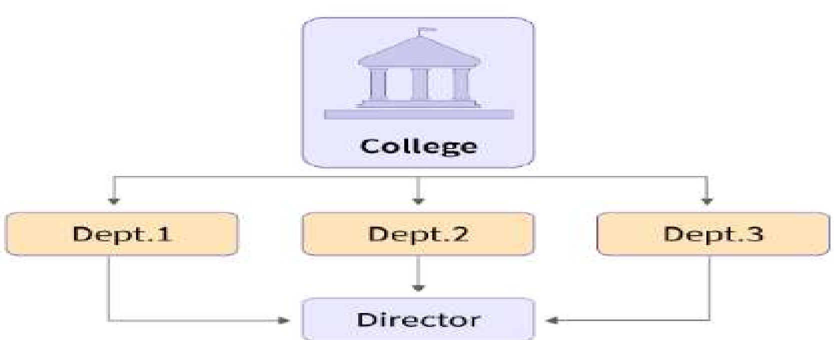
In this model, a child node will only have a single parent node.This model efficiently describes many real-world relationships like index of a book, recipes etc.In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for 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.

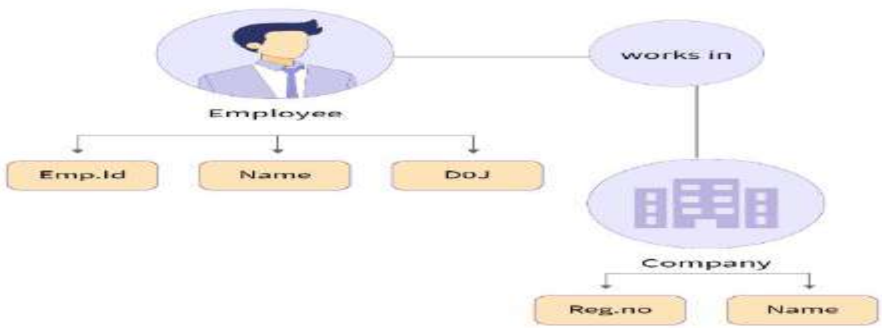


3. Entity-relationship Model

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.

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. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pincode, city etc, and there will be a relationship between them.



4. Relational Model

In this model, data is organised 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, infact, 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.

In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.

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

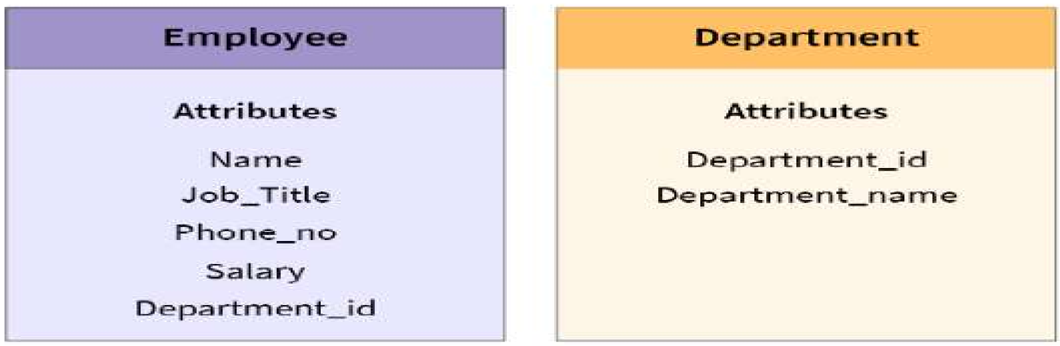
subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

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

5. Object-Oriented Data model

As suggested by its name, the object-oriented data model is a combination of object-oriented programming and relational data model. In this data model, the data and their relationship are represented in a single structure which is known as an object.

Since data is stored as objects we can easily store audio, video, images, etc in the database which was very difficult and inconvenient to do in the relational model. As shown in the image below two objects are connected with each other through links.

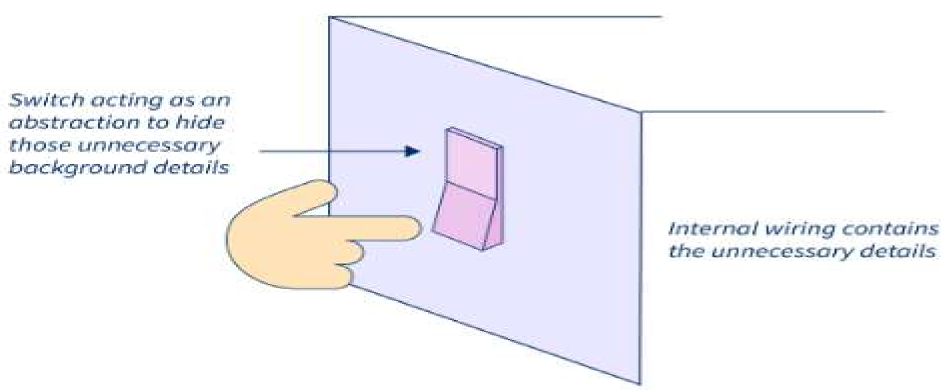


Data Abstraction in DBMS

Data abstraction is the method of hiding the unimportant details that are present in the database from the end users to make the accessing of data easy and secure.

The main purpose of data abstraction is to hide irrelevant data and provide an abstract view of the data. With the help of data abstraction, developers hide irrelevant data from the user and provide them the relevant data. By doing this, users can access the data without any hassle, and the system will also work efficiently.

Let us take a small example. Say, someone, asks you to switch on the fans in a room. All you will need to do is simply walk to the switchboard and turn on the switch for the fan, that's it! Do you need to know where the electricity is coming from, how the poles of the switch are connected, or exactly what the internal working of a fan is? The answer to all this is **NO!** That is what data abstraction is, all these background details are hidden from you inside the switchboard!



Levels of Data Abstractions in DBMS

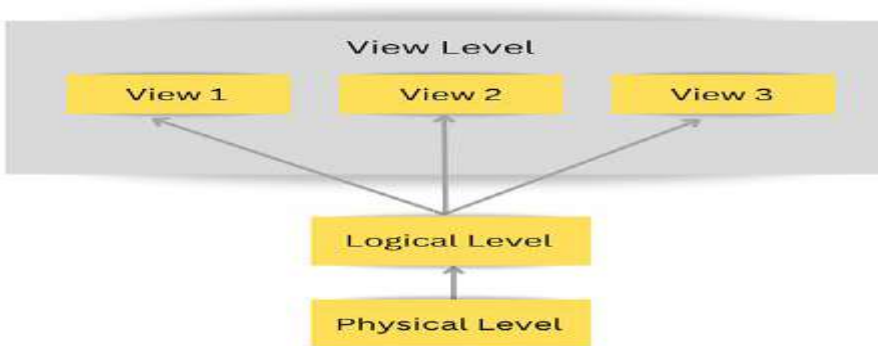
The data abstraction in **DBMS** is implemented in 3 layers:

- Physical or Internal Level
- Logical or Conceptual Level
- View or External Level

1. Physical or Internal Level:

The physical or internal layer is the lowest level of data abstraction in the database management system. It is the layer that defines how data is actually stored in the database. It defines methods to access the data in the database. It defines complex data structures in detail, so it is very complex to understand, which is why it is kept hidden from the end user.

Data Administrators (DBA) decide how to arrange data and where to store data. The Data Administrator (DBA) is the person whose role is to manage the data in the database at the physical or internal level. There is a data center that securely stores the raw data in detail on hard drives at this level.



Levels of Data Abstraction in DBMS

2. Logical or Conceptual Level:

The logical or conceptual level is the intermediate or next level of data abstraction. It explains what data is going to be stored in the database and what the relationship is between them.

It describes the structure of the entire data in the form of tables. The logical level or conceptual level is less complex than the physical level. With the help of the logical level, Data Administrators (DBA) abstract data from raw data present at the physical level.

3. View or External Level:

View or External Level is the highest level of data abstraction. There are different views at this level that define the parts of the overall data of the database. This level is for the end-user interaction; at this level, end users can access the data based on their queries.

Advantages of data abstraction in DBMS

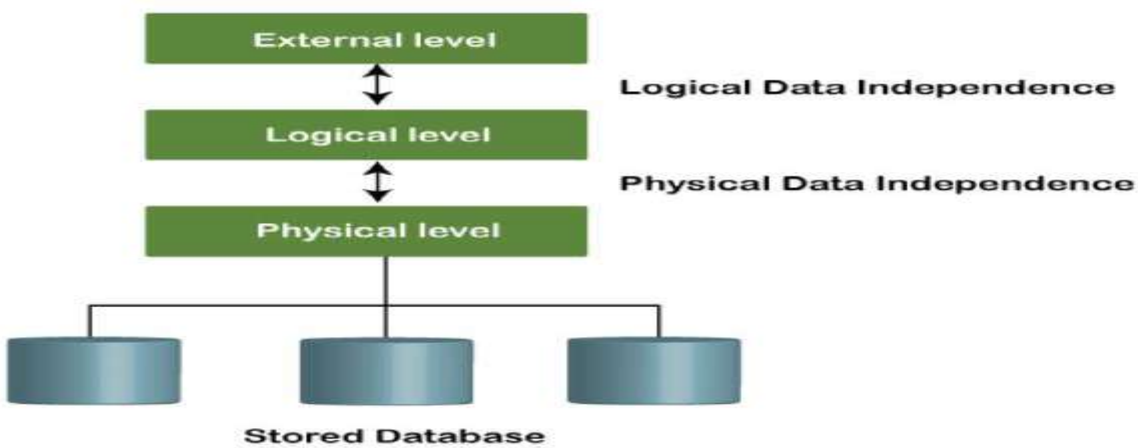
- Users can easily access the data based on their queries.
- It provides security to the data stored in the database.
- Database systems work efficiently because of data abstraction.

Data Independence in DBMS

Data independence is a concept of DBMS which alters the schema of the database at one level of the database system without altering the schema definition of same database at the next higher level. It helps the users by improving the quality of data. By using data independence, there is no need to modify the data structure in the application programs.

In the Database Management System, data independence is of two types:

1. Logical Data Independence
2. Physical Data Independence



Logical Data Independence

Logical Data Independence is a term that changes the schema at the conceptual level without having to modify the external level or application program. The external or user view would not be affected when there is any change in conceptual view data.

This data independence is mainly used for separating the external schema from the schema at the conceptual level. Modifications at the logical level are compulsary when there is any modification in the logical structure of the database.

Following are the examples of logical data independence, which will not affect the external or user view:

1. We can easily break the existing record into more than two records.
2. We can also combine the two records into one.
3. We can easily modify, delete or insert the attribute.

Physical Data Independence

Physical Data Independence is another type of data independence that alters the schema at the internal/ physical level without having to alter the schema at the conceptual level. The conceptual structure of the database systems will not be affected if there is any modification in the storage size of the database.

This data independence is mainly used for separating the internal schema from the schema at the conceptual level. As compared to the above **data independence**, this data independence can be achieved easily.

Following are the examples of physical data independence, which will not affect the conceptual view:

1. We can modify the indexes in the database system.
2. We can easily modify the data structure, which is used for storage.

Structure of DBMS

DBMS (Database Management System) acts as an interface between the user and the database. The user requests the DBMS to perform various operations (insert, delete, update and retrieval) on the database. The components of DBMS perform these requested operations on the database and provide necessary data to the users.

Database Management System (DBMS) is software that allows access to data stored in a database and provides an easy and effective method of –

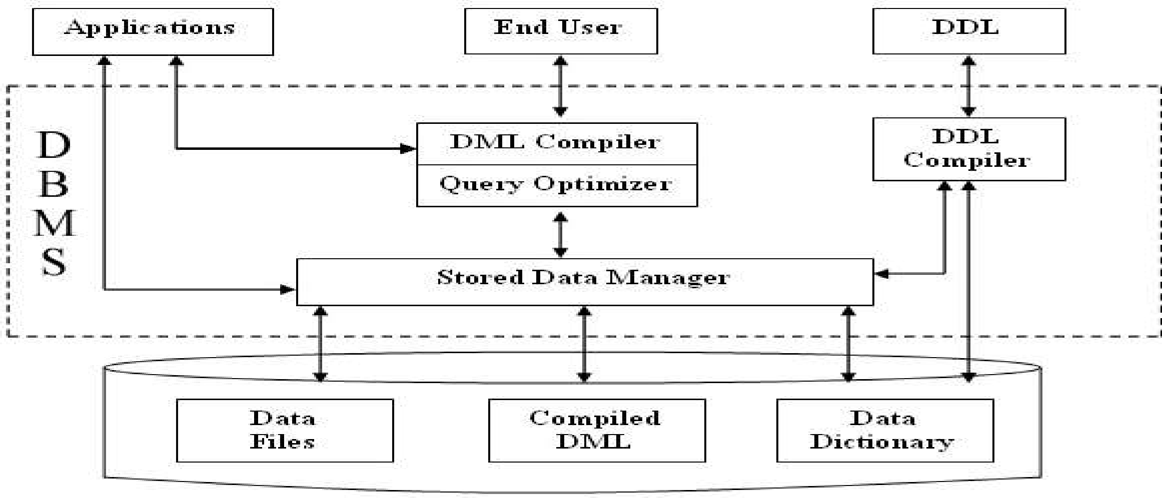
- Defining the information.
- Storing the information.
- Manipulating the information.
- Protecting the information from system crashes or data theft.
- Differentiating access permissions for different users.

Components of DBMS Structure

The structure of DBMS is divided into three components namely "Query Processor", "Storage Manager", and "Disk Storage". Each of these components plays a very crucial role in *Database Management*.

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.

Characteristics of DBMS:

- ✓ **Support for multiple views of data**

A database supports multiple views of data. A view is a subset of the database, which is defined and dedicated for particular users of the system. Multiple users in the system might have different views of the system. Each view might contain only the data of interest to a user or group of users.

- ✓ **Sharing of data and multiuser system**

Current database systems are designed for multiple users. That is, they allow many users to access the same database at the same time. This access is achieved through features called concurrency control strategies. These strategies ensure that the data accessed are always correct and that data integrity is maintained.

The design of modern multiuser database systems is a great improvement from those in the past which restricted usage to one person at a time.

- ✓ **Control of data redundancy**

In the database approach, ideally, each data item is stored in only one place in the database. In some cases, data redundancy still exists to improve system performance, but such redundancy is controlled by application programming and kept to minimum by introducing as little redudancy as possible when designing the database.

- ✓ **Data sharing**

The integration of all the data, for an organization, within a database system has many advantages. First, it allows for data sharing among employees and others who have access to the system. Second, it gives users the ability to generate more information from a given amount of data than would be possible without the integration.

- ✓ **Enforcement of integrity constraints**

Database management systems must provide the ability to define and enforce certain constraints to ensure that users enter valid information and maintain data integrity. A database constraint is a restriction or rule that dictates what can be entered or edited in a table such as a postal code using a certain format or adding a valid city in the City field.

There are many types of database constraints. Data type, for example, determines the sort of data permitted in a field, for example numbers only. Data uniqueness such as the primary key ensures that no duplicates are entered. Constraints can be simple (field based) or complex (programming).

- ✓ **Restriction of unauthorized access**

Not all users of a database system will have the same accessing privileges. For example, one user might have read-only access (i.e., the ability to read a file but not make changes), while another might have read and write privileges, which is the ability to both read and modify a file. For this reason, a database management system should provide a security subsystem to create and control different types of user accounts and restrict unauthorized access.

- ✓ **Data independence**

Another advantage of a database management system is how it allows for data independence. In other words, the system data descriptions or data describing data (metadata) are separated from the application programs. This is possible because changes to the data structure are handled by the database management system and are not embedded in the program itself.

- ✓ **Transaction processing**

A database management system must include concurrency control subsystems. This feature ensures that data remains consistent and valid during transaction processing even if several users update the same information.

- ✓ **Provision for multiple views of data**

By its very nature, a DBMS permits many users to have access to its database either individually or simultaneously. It is not important for users to be aware of how and where the data they access is stored

- ✓ **Backup and recovery facilities**

Backup and recovery are methods that allow you to protect your data from loss. The database system provides a separate process, from that of a network backup, for backing up and recovering data. If a hard drive fails and the database stored on the hard drive is not accessible, the only way to recover the database is from a backup.

If a computer system fails in the middle of a complex update process, the recovery subsystem is responsible for making sure that the database is restored to its original state. These are two more benefits of a database management system.

Database Design in DBMS

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database design in DBMS are to produce logical and physical designs models of the proposed database system.The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

The physical data design model involves translating the logical DB design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

Entity Relationship (ER) Diagram:

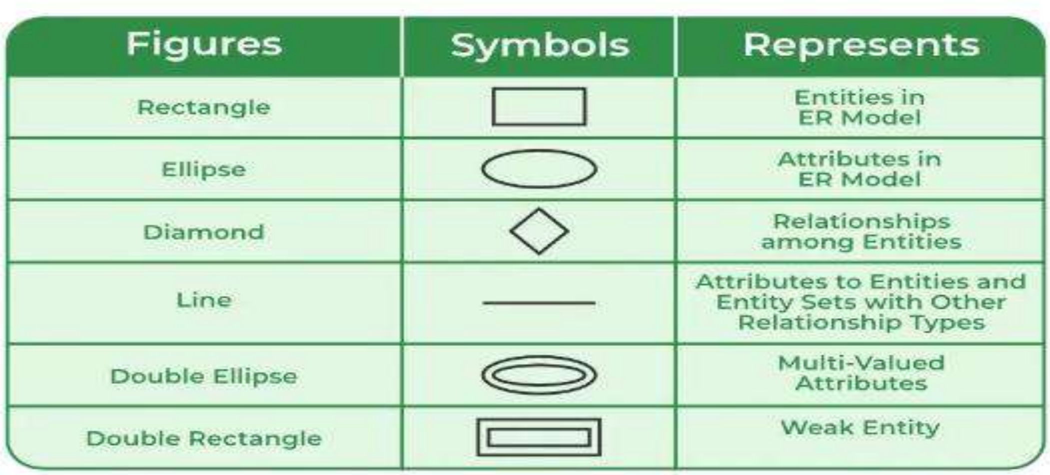
ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

Symbols Used in ER Model

ER Model is used to model the logical view of the system from a data perspective which consists of these symbols:

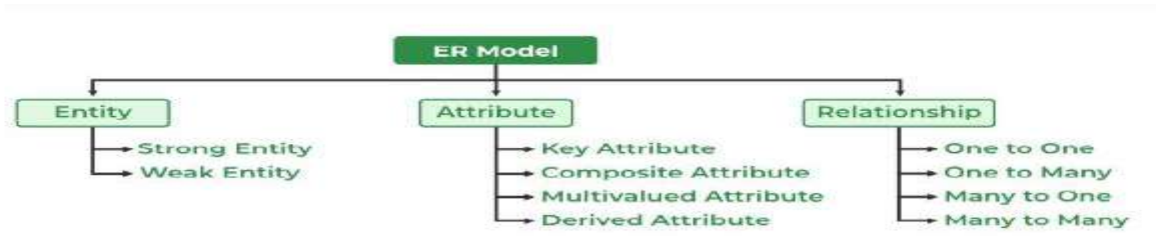
- **Rectangles:** Rectangles represent Entities in ER Model.
- **Ellipses:** Ellipses represent Attributes in ER Model.
- **Diamond:** Diamonds represent Relationships among Entities.
- **Lines:** Lines represent attributes to entities and entity sets with other relationship types.
- **Double Ellipse:** Double Ellipses represent Multi-Valued Attributes.
- **Double Rectangle:** Double Rectangle represents a Weak Entity.



Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

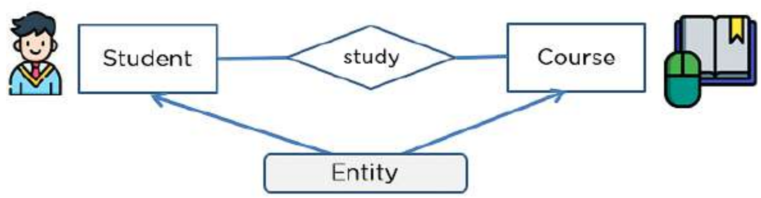
Components of ER Diagram

ER Model consists of Entities, Attributes, and Relationships among Entities in a Database System.



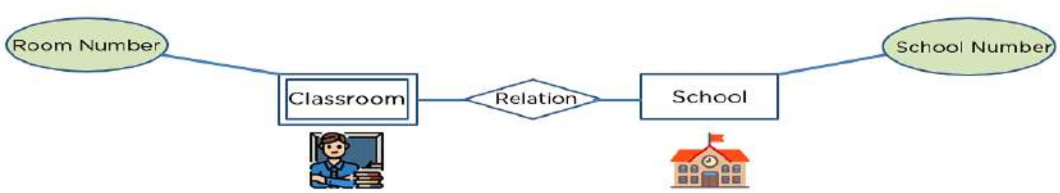
Entities

An entity can be either a living or non-living component. It showcases an entity as a rectangle in an ER diagram. For example, in a student study course, both the student and the course are entities.



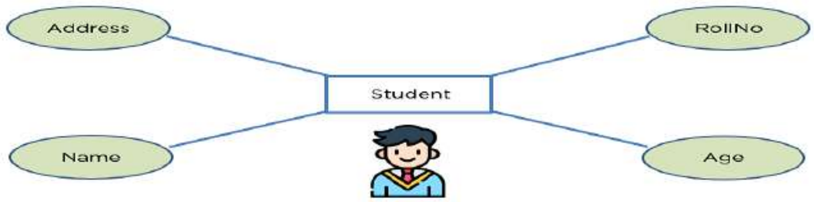
Weak Entity

An entity that makes reliance over another entity is called a weak entity. You showcase the weak entity as a double rectangle in ER Diagram. In the example below, school is a strong entity because it has a primary key attribute - school number. Unlike school, the classroom is a weak entity because it does not have any primary key and the room number here acts only as a discriminator.



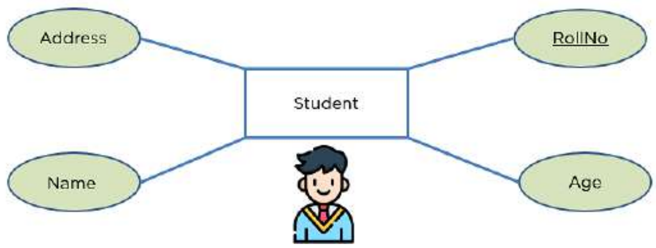
Attribute

An attribute exhibits the properties of an entity. You can illustrate an attribute with an oval shape in an ER diagram.



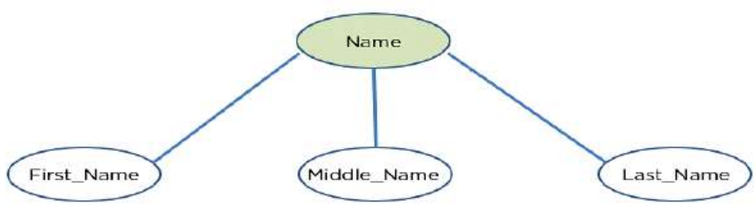
Key Attribute

Key attribute uniquely identifies an entity from an entity set. It underlines the text of a key attribute. For example: For a student entity, the roll number can uniquely identify a student from a set of students.



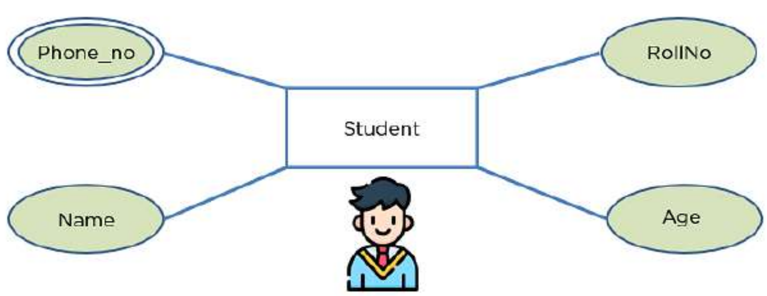
Composite Attribute

An attribute that is composed of several other attributes is known as a composite attribute. An oval showcases the composite attribute, and the composite attribute oval is further connected with other ovals.



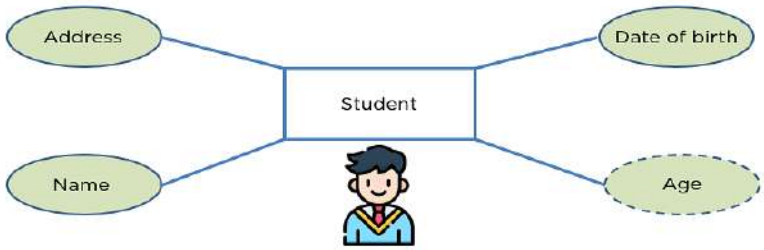
Multivalued Attribute

Some attributes can possess over one value, those attributes are called multivalued attributes. The double oval shape is used to represent a multivalued attribute.



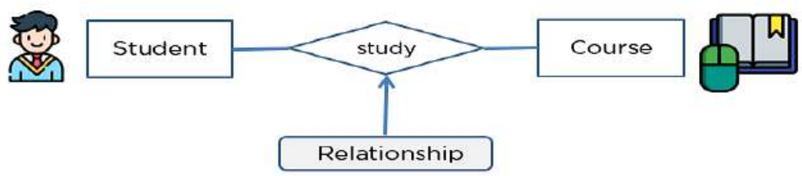
Derived Attribute

An attribute that can be derived from other attributes of the entity is known as a derived attribute. In the ER diagram, the dashed oval represents the derived attribute.



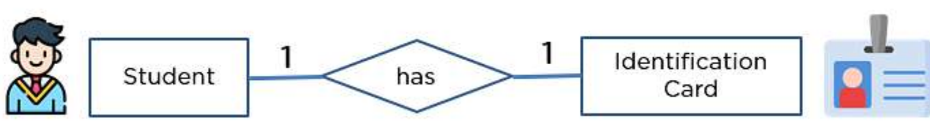
Relationship

The diamond shape showcases a relationship in the ER diagram. It depicts the relationship between two entities. In the example below, both the student and the course are entities, and study is the relationship between them.



One-to-One Relationship

When a single element of an entity is associated with a single element of another entity, it is called a one-to-one relationship. For example, a student has only one identification card and an identification card is given to one person.

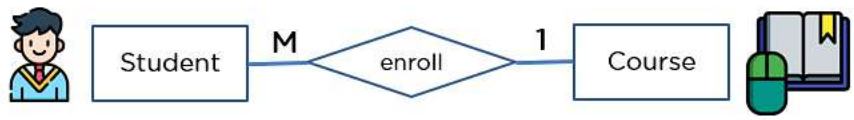


One-to-Many Relationship

When a single element of an entity is associated with more than one element of another entity, it is called a one-to-many relationship. For example, a customer can place many orders, but an order cannot be placed by many customers.

Many-to-One Relationship

When more than one element of an entity is related to a single element of another entity, then it is called a many-to-one relationship. For example, students have to opt for a single course, but a course can have many students.



Many-to-Many Relationship

When more than one element of an entity is associated with more than one element of another entity, this is called a many-to-many relationship. For example, you can assign an employee to many projects and a project can have many employees.

Participation Constraint

Participation Constraint is applied to the entity participating in the relationship set.

1. Total Participation – Each entity in the entity set must participate in the relationship. If each student must enroll in a course, the participation of students will be total. Total participation is shown by a double line in the ER diagram.

2. Partial Participation – The entity in the entity set may or may NOT participate in the relationship. If some courses are not enrolled by any of the students, the participation in the course will be partial.

The diagram depicts the 'Enrolled in' relationship set with Student Entity set having total participation and Course Entity set having partial participation.

Total Participation and Partial Participation

Features of ER

The features of ER Model are as follows −

ER Diagram: ER diagrams are the diagrams that are sketched out to design the database. They are created based on three basic concepts: entities, attributes, and relationships between them. In ER diagram we define the entities, their related attributes, and the relationships between them. This helps in illustrating the logical structure of the databases.

Database Design: The Entity-Relationship model helps the database designers to build the database in a very simple and conceptual manner.

Graphical Representation helps in Better Understanding: ER diagrams are very easy and simple to understand and so the developers can easily use them to communicate with stakeholders.

Easy to build: The ER model is very easy to build.

The extended E-R features: Some of the additional features of ER model are specialization, upper and lower-level entity sets, attribute inheritance, aggregation, and generalization.

Integration of ER model: This model can be integrated into a common dominant relational model and is widely used by database designers for communicating their ideas.

Simplicity and various applications of ER model: It provides a preview of how all your tables should connect, and what fields are going to be on each table, which can be used as a blueprint for implementing data in specific software applications.