Introduction of DBMS (Database Management System)

A **database** is a collection of interrelated data that helps in the efficient retrieval, insertion, and deletion of data from the database and organizes the data in the form of tables, views, schemas, reports, etc. **For Example**, a university database organizes the data about students, faculty, admin staff, etc. which helps in the efficient retrieval, insertion, and deletion of data from it.

What is DBMS?

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database. DBMS provides an environment to store and retrieve data in convenient and efficient manner.

Key Features of DBMS

- **Data modeling:** A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
- **Data storage and retrieval:** A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
- **Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
- **Data integrity and security:** A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
- **Backup and recovery:** A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
- **DBMS can be classified into two types:** Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL)
- **RDBMS:** Data is organized in the form of tables and each table has a set of rows and columns. The data are related to each other through primary and foreign keys.
- **NoSQL:** Data is organized in the form of key-value pairs, documents, graphs, or column-based. These are designed to handle large-scale, high-performance scenarios**.**

Types of DBMS

1. **Relational Database Management System (RDBMS):** Data is organized into tables (relations) with rows and columns, and the relationships between the data are managed through primary and foreign keys. SQL (Structured Query Language) is used to query and manipulate the data.
2. **NoSQL DBMS:** Designed for high-performance scenarios and large-scale data, NoSQL databases store data in various non-relational formats such as key-value pairs, documents, graphs, or columns.
3. **Object-Oriented DBMS (OODBMS):** Stores data as objects, similar to those used in object-oriented programming, allowing for complex data representations and relationships

Database Languages

- **Data Definition Language**
- **Data Manipulation Language**
- **Data Control Language**
- **Transactional Control Language**

Data Definition Language (DDL)

DDL is the short name for Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- **CREATE:** to create a database and its objects like (table, index, views, store procedure, function, and triggers)
- **ALTER:** alters the structure of the existing database

- **DROP:** delete objects from the database
- **TRUNCATE:** remove all records from a table, including all spaces allocated for the records are removed
- **COMMENT:** add comments to the data dictionary
- **RENAME:** rename an object

Data Manipulation Language (DML)

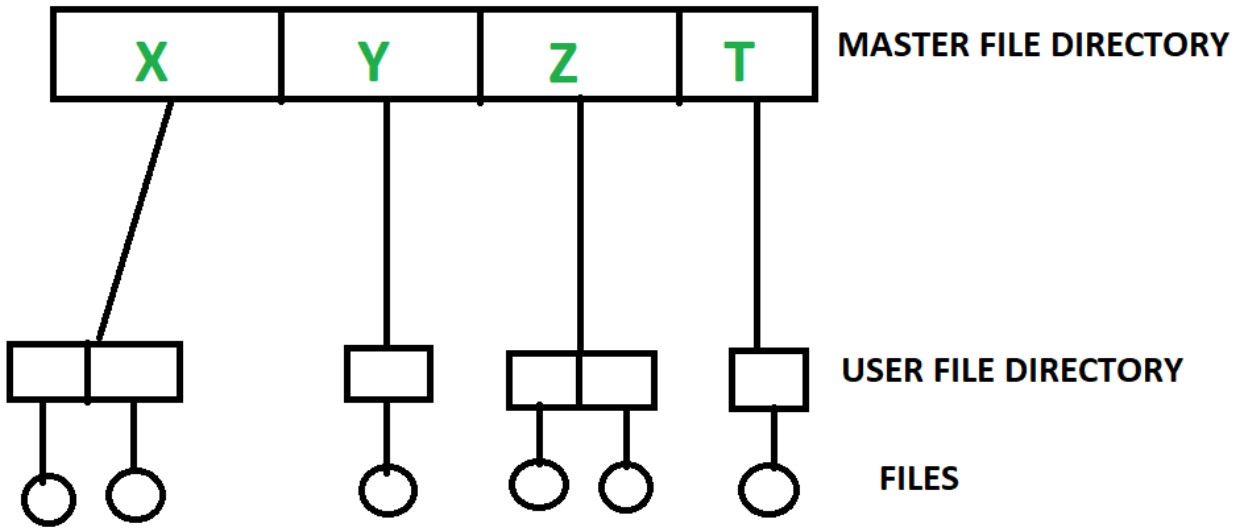
DML is the short name for Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database. **Data query language(DQL)** is the subset of "Data Manipulation Language". The most common command of DQL is **SELECT** statement. SELECT statement help on retrieving the data from the table without changing anything in the table.

- **SELECT:** retrieve data from a database
- **INSERT:** insert data into a table
- **UPDATE:** updates existing data within a table
- **DELETE:** Delete all records from a database table
- **MERGE:** UPSERT operation (insert or update)
- **CALL:** call a PL/SQL or Java subprogram
- **EXPLAIN PLAN:** interpretation of the data access path
- **LOCK TABLE:** concurrency Control

File System

The **file system** is basically a way of arranging the files in a storage medium like a hard disk. The file system organizes the files and helps in the retrieval of files when they are required. File systems consist of different files which are grouped into directories. The directories further contain other folders and files. The file system performs basic operations like management, file naming, giving access rules, etc.

Example: NTFS(New Technology File System) , EXT(Extended File System).



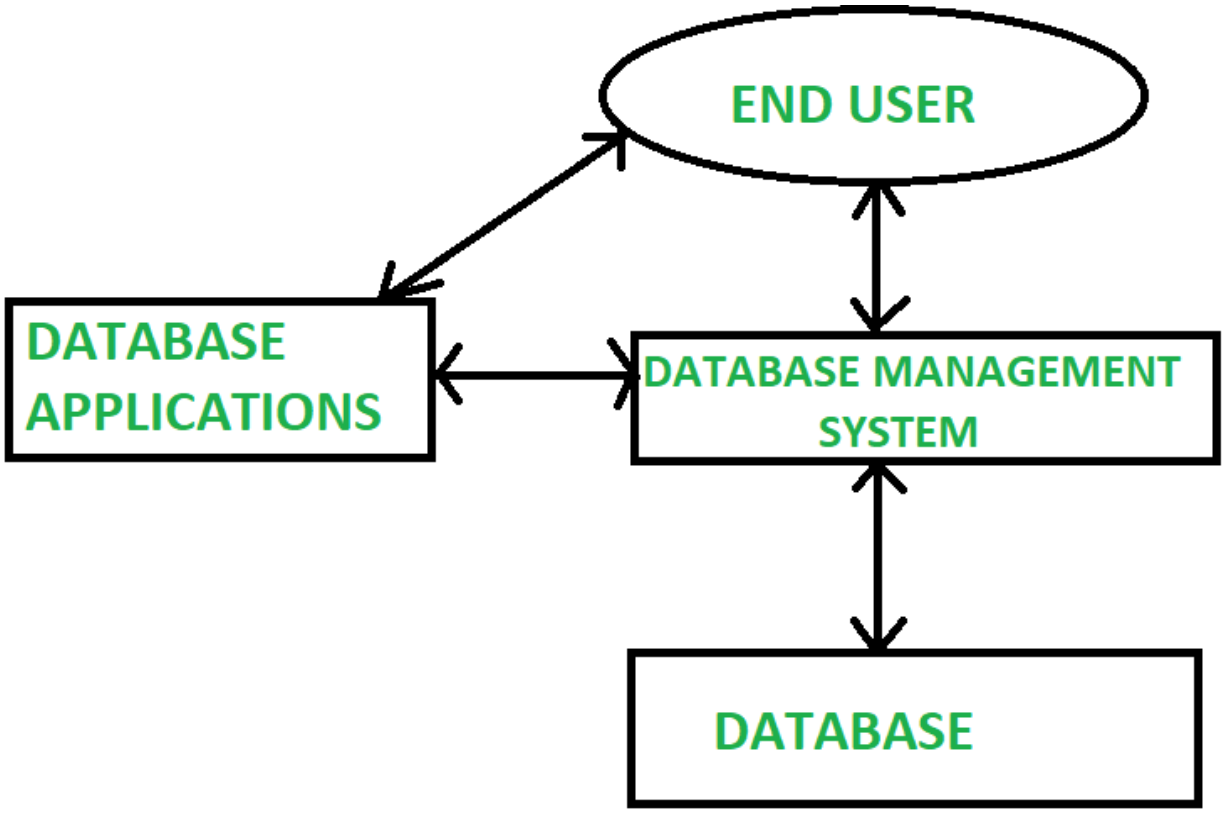
DBMS (Database Management System)

Database Management System is basically software that manages the collection of related data. It is used for storing data and retrieving the data effectively when it is needed. It also provides proper security

measures for protecting the data from unauthorized access. In Database Management System the data can be fetched by SQL queries and relational algebra. It also provides mechanisms for data recovery and data backup.

Example:

Oracle, MySQL, MS SQL server.



Advantages of Database Management System

The advantages of database management systems are:

1. **Data Security:** The more accessible and usable the database, the more it is prone to security issues. As the number of users increases, the data transferring or data sharing rate also increases thus increasing the risk of data security. It is widely used in the corporate world where companies invest large amounts of money, time, and effort to ensure data is secure and used properly. A Database Management System (DBMS) provides a better platform for data privacy and security policies thus, helping companies to improve Data Security.
2. **Data integration:** Due to the Database Management System we have access to well-managed and synchronized forms of data thus it makes data handling very easy and gives an integrated view of how a particular organization is working and also helps to keep track of how one segment of the company affects another segment.
3. **Data abstraction:** The major purpose of a database system is to provide users with an abstract view of the data. Since many complex algorithms are used by the developers to increase the efficiency of databases that are being hidden by the users through various data abstraction levels to allow users to easily interact with the system.

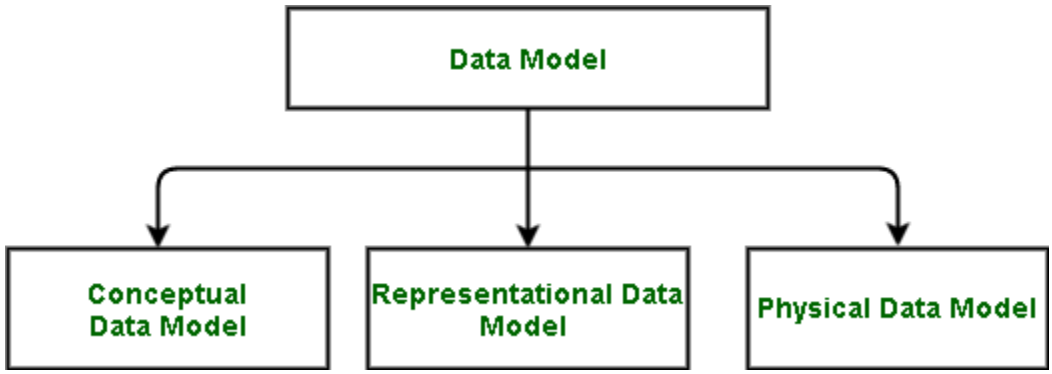
4. **Reduction in data Redundancy:** When working with a structured database, DBMS provides the feature to prevent the input of duplicate items in the database. for e.g. – If there are two same students in different rows, then one of the duplicate data will be deleted.
5. **Data sharing:** A DBMS provides a platform for sharing data across multiple applications and users, which can increase productivity and collaboration.
6. **Data consistency and accuracy:** DBMS ensures that data is consistent and accurate by enforcing data integrity constraints and preventing data duplication. This helps to eliminate data discrepancies and errors that can occur when data is stored and managed manually.
7. **Data organization:** A DBMS provides a systematic approach to organizing data in a structured way, which makes it easier to retrieve and manage data efficiently.
8. **Efficient data access and retrieval:** DBMS allows for efficient data access and retrieval by providing indexing and query optimization techniques that speed up data retrieval. This reduces the time required to process large volumes of data and increases the overall performance of the system.
9. **Concurrency and maintained Atomicity** : That means, if some operation is performed on one particular table of the database, then the change must be reflected for the entire database. The DBMS allows concurrent access to multiple users by using the synchronization technique.
10. **Scalability and flexibility:** DBMS is highly scalable and can easily accommodate changes in data volumes and user requirements. DBMS can easily handle large volumes of data, and can scale up or down depending on the needs of the organization. It provides flexibility in data storage, retrieval, and manipulation, allowing users to easily modify the structure and content of the database as needed.

Data Models in DBMS

A Data Model in Database Management System (DBMS) is the concept of tools that are developed to summarize the description of the database. Data Models provide us with a transparent picture of data which helps us in creating an actual database. It shows us from the design of the data to its proper implementation of data.

Types of Relational Models

1. Conceptual Data Model
2. Representational Data Model
3. Physical Data Model



1. Conceptual Data Model

The conceptual data model describes the database at a very high level and is useful to understand the needs or requirements of the database. It is this model, that is used in the requirement-gathering process i.e. before the Database Designers start making a particular database.

2. 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 database. The representational data model allows us to focus primarily, on the design part of the database. A popular representational model is a Relational model.

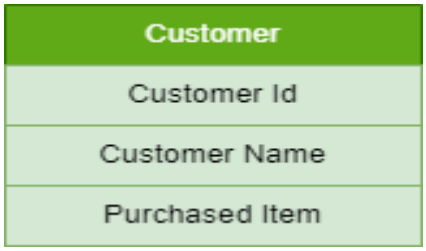
3. Physical Data Model

The physical Data Model is used to practically implement Relational Data Model. Ultimately, all data in a database is stored physically on a secondary storage device such as discs and tapes. This is stored in the form of files, records, and certain other data structures.

Database Schemas

What is Schema?

- The Skeleton of the database is created by the attributes and this skeleton is named Schema.
- Schema mentions the logical constraints like table, primary key, etc.
- The schema does not represent the data type of the attributes.



Database Schema

- A database schema is a **logical representation of data** that shows how the data in a database should be stored logically. It shows how the data is organized and the relationship between the tables.
- Database schema contains table, field, views and relation between different keys like primary key, foreign key.
- Data are stored in the form of files which is unstructured in nature which makes accessing the data difficult. Thus to resolve the issue the data are organized in structured way with the help of database schema.
- Database schema provides the organization of data and the relationship between the stored data.
- Database schema defines a set of guidelines that control the database along with that it provides information about the way of accessing and modifying the data

Types of Database Schemas

There are 3 types of database schema:

Physical Database Schema

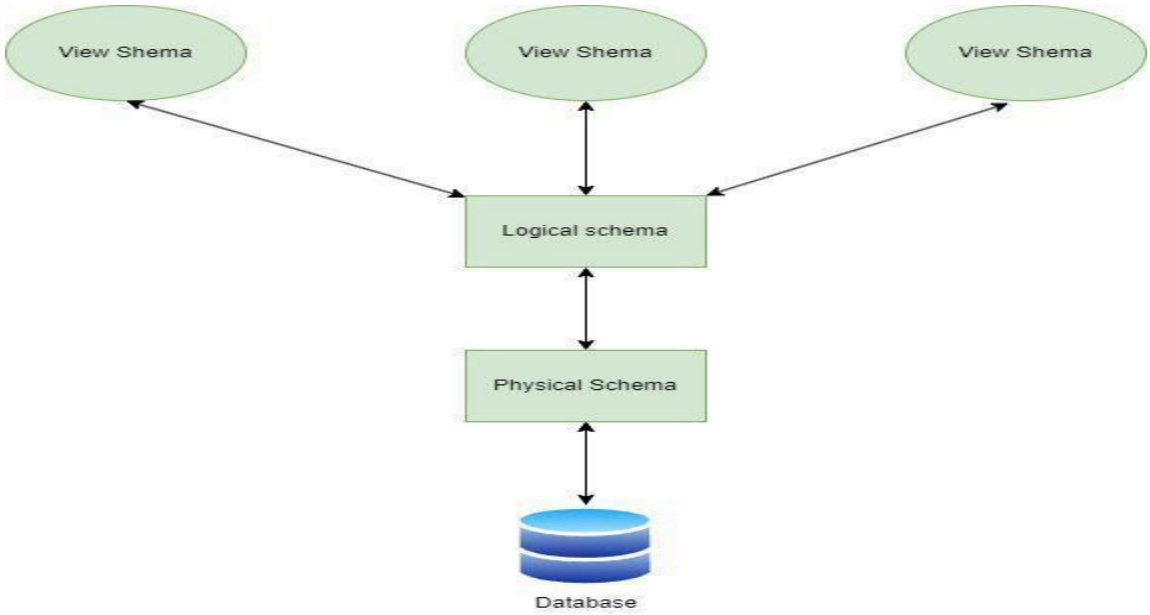
- A Physical schema defines, how the data or information is stored physically in the storage systems in the form of files & indices. This is the actual code or syntax needed to create the structure of a database, we can say that when we design a database at a physical level, it's called physical schema.
- The Database administrator chooses where and how to store the data in the different blocks of storage

Logical Database Schema

- A logical database schema defines all the logical constraints that need to be applied to the stored data, and also describes tables, views, entity relationships, and integrity constraints.
- The Logical schema describes how the data is stored in the form of tables & how the attributes of a table are connected.

View Database Schema

- It is a view level design which is able to define the interaction between end-user and database.
- User is able to interact with the database with the help of the interface without knowing much about the stored mechanism of data in database



Why Use ER Diagrams In DBMS?

- ER diagrams represent the E-R model in a database, making them easy to convert into relations (tables).
- ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful.
- ER diagrams require no technical knowledge and no hardware support.
- These diagrams are very easy to understand and easy to create even for a naive user.
- It gives a standard solution for visualizing the data logically.

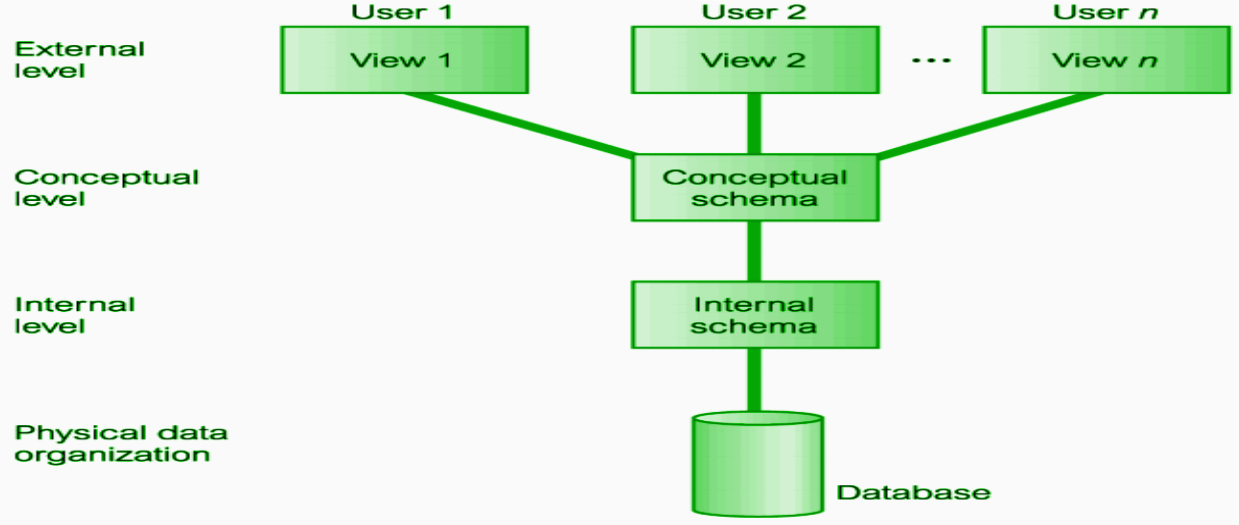
Symbols Used in ER Model

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 the ER Model.
- **Ellipses:** Ellipses represent Attributes in the 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.

What is Data Independence in DBMS?

In the context of a database management system, data independence is the feature that allows the schema of one layer of the database system to be changed without any impact on the schema of the next higher level of the database system. ” Through data independence, we can build an environment in which data is independent of all programs, and through the three schema architectures, data independence will be more understandable. Data via two card stencils along with centralized DBMS data is a form of transparency that has value for someone.



Types of Data Independence

There are two types of data independence.

- logical data independence
- Physical data independence

Logical Data Independence

- Changing the logical schema (conceptual level) without changing the external schema (view level) is called logical data independence.
- It is used to keep the external schema separate from the logical schema.
- If we make any changes at the conceptual level of data, it does not affect the view level.
- This happens at the user interface level.

Physical Data Independence

- Making changes to the physical schema without changing the logical schema is called physical data independence.
- If we change the storage size of the database system server, it will not affect the conceptual structure of the database.
- It is used to keep the conceptual level separate from the internal level

Using the ER model for bigger data creates a lot of complexity while designing a database model, So in order to minimize the complexity Generalization, Specialization, and Aggregation were introduced in the ER model and these were used for data abstraction in which an abstraction mechanism is used to hide details of a set of objects. Some of the terms were added to the Enhanced ER Model, where some new concepts were added. These new concepts are:

- Generalization
- Specialization
- Aggregation

Generalization

Generalization is the process of extracting common properties from a set of entities and creating a generalized entity from it. It is a bottom-up approach in which two or more entities can be generalized to a higher-level entity if they have some attributes in common.

Specialization

In specialization, an entity is divided into sub-entities based on its characteristics. It is a top-down approach where the higher-level entity is specialized into two or more lower-level entities.

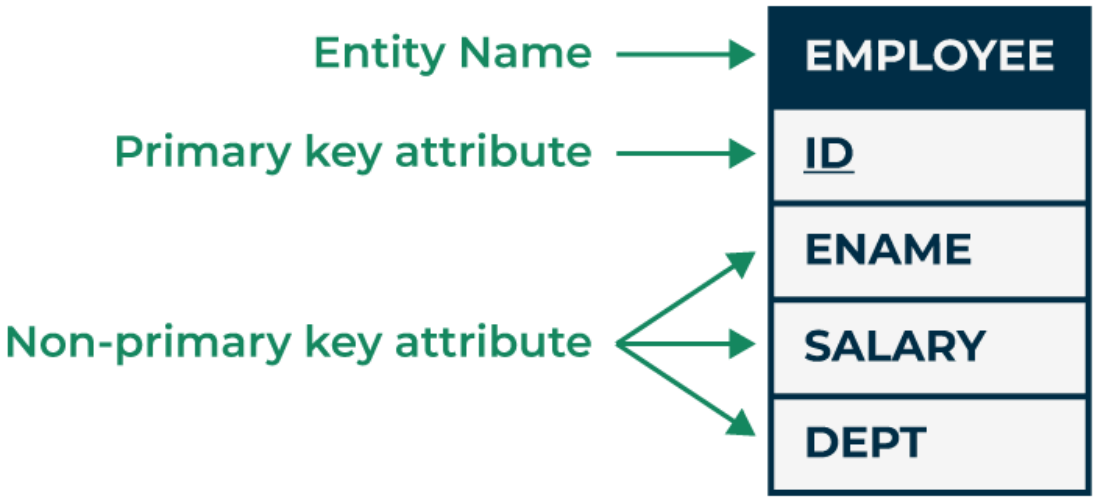
Aggregation

An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity. Aggregation is an abstraction through which we can represent relationships as higher-level entity sets.

Entity

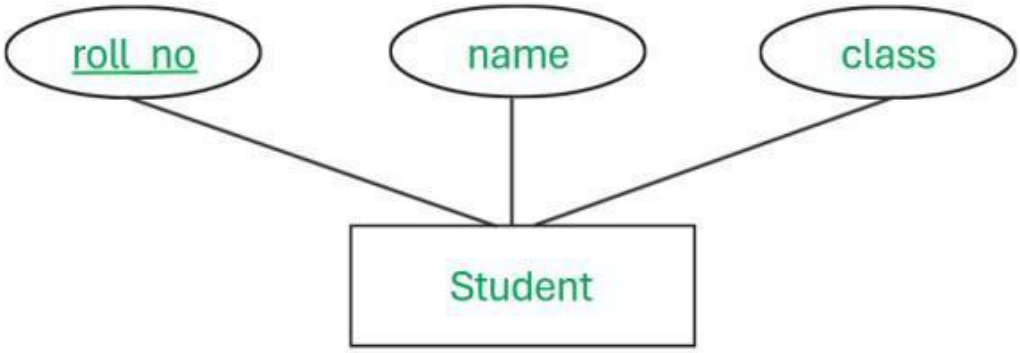
An entity is a "**thing**" or "**object**" in the real world. An entity contains attributes, which describe that entity. So anything about which we store information is called an entity. Entities are recorded in the database and must be distinguishable, i.e., easily recognized from the group.

For example: A student, An employee, or bank a/c, etc. all are entities.



Attributes

Attributes are properties or characteristics of an entity. Attributes are used to describe the entity. The attribute is nothing but a piece of data that gives more information about the entity. Attributes are used to distinguish one entity from the other entity.



Types Of Attribute

- **Simple Attribute.**
- **Composite Attribute.**
- **Single Valued Attribute.**
- **Multivalued Attribute.**
- **Key Attribute.**
- **Derived Attribute.**
- **Stored Attribute.**
- **Complex Attribute**

Difference Between RDBMS and DBMS

Database Management System (DBMS) is a software that is used to define, create, and maintain a database and provides controlled access to the data.

Why is DBMS Required?

Database management system, as the name suggests, is a management system that is used to manage the entire flow of data, i.e, the insertion of data or the retrieval of data, how the data is inserted into the database, or how fast the data should be retrieved, so DBMS takes care of all these features, as it maintains the uniformity of the database as well does the faster insertions as well as retrievals.

Why is RDBMS Required?

RDBMS on the other hand is a type of DBMS, as the name suggests it deals with relations as well as various key constraints. So here we have tables which are called schema and we have rows which are called tuples. It also aids in the reduction of data redundancy and the preservation of database integrity.

Relational Database Management System is an **advanced** version of a DBMS.