**MANNAR THIRUMALAI NAICKER COLLEGE(AUTONOMOUS)**

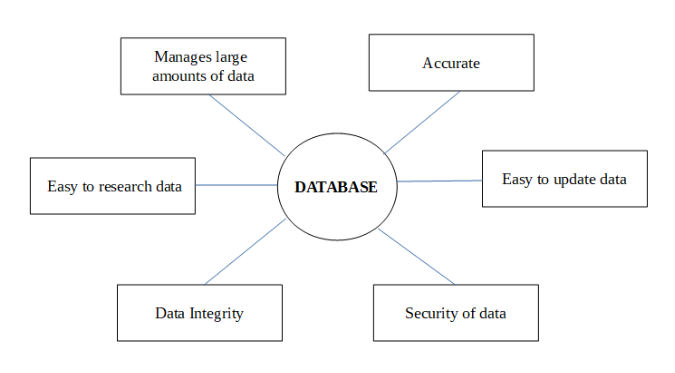
**DEPARTMENT OF INFORMATION TECHNOLOGY**

**RELATIONAL DATABASE MANAGEMENT SYSTEM**

**(21UITC31)**

A database is a collection of data, usually stored in electronic form. A database is typically designed so that it is easy to store and access information.

A good database is crucial to any company or organisation. This is because the database stores all the pertinent details about  the company such as employee records, transactional records, salary details etc.



The various reasons a database is important are −

**Manages large amounts of data**

A database stores and manages a large amount of data on a daily basis. This would not be possible using any other tool such as a spreadsheet as they would simply not work.

**Accurate**

A database is pretty accurate as it has all sorts of build in constraints, checks etc. This means that the information available in a database is guaranteed to be correct in most cases.

**Easy to update data**

In a database, it is easy to update data using various Data Manipulation languages (DML) available. One of these languages is SQL.

**Security of data**

Databases have various methods to ensure security of data. There are user logins required before accessing a database and various access specifiers. These allow only authorised users to access the database.

**Data integrity**

This is ensured in databases by using various constraints for data. Data integrity in databases makes sure that the data is accurate and consistent in a database.

**Easy to research data**

It is very easy to access and research data in a database. This is done using Data Query Languages (DQL) which allow searching of any data in the database and performing computations on it.

## What is Database?

A database is a collection of related data or a database is a collection of information that is organized so that it can be easily accessed, managed, and updated.

## Characteristics of Data in a Database:

The data in a database should have the following features:

### **1. Shared:**

Data should be sharable among different users and applications.

### **2. Persistence:**

Data should exist permanently in the database. Changes in the database must not be lost because of any failure.

### **3. Validity/Integrity/Correctness:**

It should maintain integrity so that there is always correct data in the database.

### **4. Security:**

Data should be protected from unauthorized access.

### **5. Non-redundancy:**

Data should not be repeated.

### **6. Consistency:**

A consistent state of the database satisfies all the constraints specified in the database. Data in a database is consistent if any changes in the database take the database from one consistent state to another.

### **7. Independence:**

The three levels in the schema should be independent of each other so that the changes in the schema at one level should not affect the other levels.

**DATABASE MANAGEMENT SYSTEM**

DBMS is a collection of programs used for managing data and simultaneously it supports different types of users to create, manage, retrieve, update and store information.

A Database Management System (DBMS) creates and manages databases. With the help of DBMS, programmers can easily retrieve, create, update, and manage data. There are various functions performed by DBMS, which gives it an upper hand over the traditional file system. Users can also create their personalized database as per requirement.

## Need of DBMS

DBMS is useful in the following ways:

***1.    Ease of Accessing Data***

In the file system, different files are created for each user containing which data they can access. Also, in the file system, for the user to extract data, there is a need for code or application. DBMS removes redundancy by granting access to users and decides which and how many parts of data is accessible to them from the database. Users can get easy access to data and can also specify the type of data they want to extract. In DBMS, users through queries can get easy access to data.

***2.    Storage and Management of Data***

Data cannot be stored in the form of objects in the file system. The data in the practical world is generally stored in the form of objects and not files. So, an application is required to map the data into objects for further usage. In DBMS, the data can be directly stored in the form of objects. In DBMS, user can query the database whereas in file system level code is written for handling, saving, and scanning of data.

***3.    Easy and Efficient File Management***

In the file system, the entire database runs for every query operation as files are indexed. It takes a lot of time compared to DBMS, where objects are indexed based on the attribute of data. The complex management of memory becomes easy to handle. With this, retrieval of data is faster than the traditional file system.

***4.    Avoiding duplicates and Redundancy***

Redundancy means repetition of the same data. In the file system, the storage of data might take place multiple times. Like, if a student is pursuing two courses in the same institution say English and Science, then his general information might get stored in both the English dept as well as Science dept. It results in prolonged hours of accessing and storing data. It further results in the inconsistency of data in both departments. Data normalization is used in DBMS to avoid duplicate data.

***5.    Concurrent Data Accessing***

Users can access data simultaneously through different applications. In the file system, this simultaneous access leads to inconsistency. Let’s take a simple example of depositing money in a bank account. Suppose two depositions of A and B of amounts 100 and 200 are made in an account X which initially contains 1000. Now since these depositions are taking place simultaneously, different depositions update account differently. A reads 1000, credits 100, updates the account to 1100. B also reads 1000, credits 200, and updates the account to 1200. Both cases have wrong information regarding the amount in account X. The result is data inconsistency. In DBMS, users can access data concurrently, and a mechanism is provided to deal with this kind of inconsistency. DBMS uses the ACID approach. ACID stands for atomicity, durability, isolation, and consistency, which ensure efficient transactions without any corruption of data.

***6.    Database Integrity***

Any data stored in the database needs to satisfy integrity constraints. For example, a database contains designations of various employees at a company say HR, account specialist, engineer, analyst, project manager, etc. Then we have a schema of employees working under these designations. A database management system ensures that the employee comes under only one out of the listed job profiles in the database. It helps in the preservation of database integrity.

These were the main features of a database management system. It has various other advantages, including:

* The time taken for developing any application gets reduced.
* Users can change the database schema.
* It also improves the security of data as all the users can’t access all the data. Security permissions are needed to use some restricted data.
* It has multiple user interfaces.
* In DBMS, if the user changes the schema of the database, all the schemas get updated accordingly.

Therefore, using DBMS is not only efficient and consistent but also saves a lot of time and energy otherwise spent on the file system. It also provides easy accessibility to each user of data they require and restricts the confidential data with security codes and permissions. Also, DBMS saves using of applications for carrying out mapping of data into objects which the file system uses.

**Database Schema**

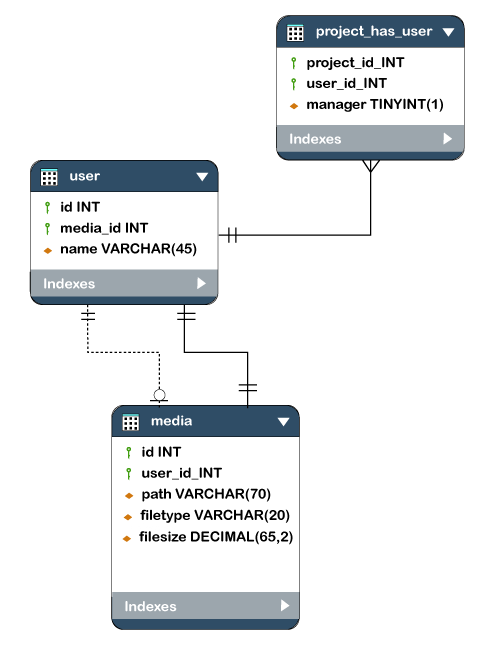
**A database schema is a structure that represents the logical storage of the data in a database**. It represents the organization of data and provides information about the relationships between the tables in a given database. In this topic, we will understand more about database schema and its types. Before understanding database schema, lets first understand what a Database is.

**What is Database?**

A [database](https://www.javatpoint.com/what-is-database) is a place to store information. It can store the simplest data, such as a list of people as well as the most complex data. The database stores the information in a well-structured format.

**What is Database Schema?**

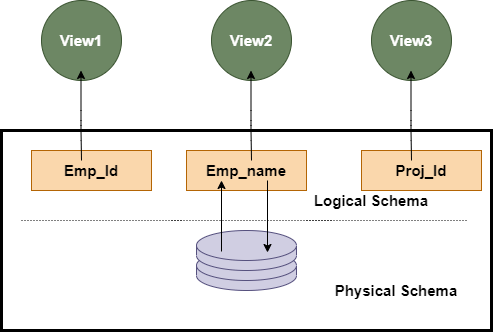
* A database schema is the logical representation of a database, which shows how the data is stored logically in the entire database. It contains list of attributes and instruction that informs the database engine that how the data is organized and how the elements are related to each other.
* A database schema contains schema objects that may include **tables, fields, packages, views, relationships, primary key, foreign key,**
* In actual, the data is physically stored in files that may be in unstructured form, but to retrieve it and use it, we need to put it in a structured form. To do this, a database schema is used. It provides knowledge about how the data is organized in a database and how it is associated with other data.
* **The schema does not physically contain the data itself; instead, it gives information about the shape of data and how it can be related to other tables or models.**
* A database schema object includes the following:
  + Consistent formatting for all data entries.
  + Database objects and unique keys for all data entries.
  + Tables with multiple columns, and each column contains its name and datatype.
* The complexity & the size of the schema vary as per the size of the project. It helps developers to easily manage and structure the database before coding it.
* The given diagram is an example of a database schema. It contains three tables, their data types. This also represents the relationships between the tables and primary keys as well as foreign keys.



**Types of Database Schema**

The database schema is divided into three types, which are:

1. **Logical Schema**
2. **Physical Schema**
3. **View Schema**



### **1. Physical Database Schema**

A physical database schema specifies how the data is stored physically on a storage system or disk storage in the form of Files and Indices. Designing a database at the physical level is called a **physical schema**.

2. Logical Database Schema

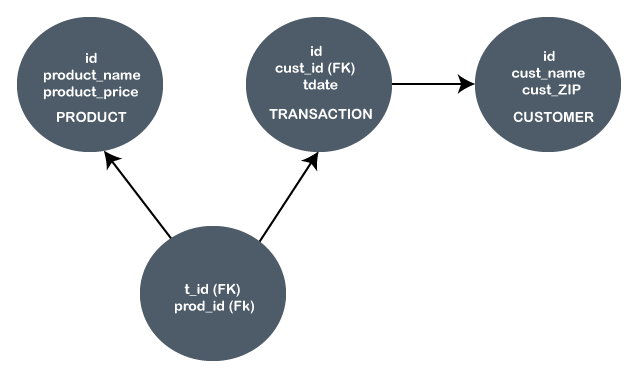
The Logical database schema specifies all the logical constraints that need to be applied to the stored data. It defines the views, integrity constraints, and table. Here the term **integrity constraints** define the set of rules that are used by [DBMS (Database Management System)](https://www.javatpoint.com/dbms-tutorial) to maintain the quality for insertion & update the data. The logical schema represents how the data is stored in the form of tables and how the attributes of a table are linked together.

At this level, programmers and administrators work, and the implementation of the data structure is hidden at this level.

Various tools are used to create a logical database schema, and these tools demonstrate the relationships between the component of your data; this process is called **ER modelling**.

The ER modelling stands for entity-relationship modelling, which specifies the relationships between different entities.

We can understand it with an example of a basic commerce application. Below is the schema diagram, the simple ER model representing the logical flow of transaction in a commerce application.



In the given example, the Ids are given in each circle, and these Ids are primary key & foreign keys.

The **primary key is** used to uniquely identify the entry in a document or record. The Ids of the upper three circles are the primary keys.

The **Foreign key** is used as the primary key for other tables. The FK represent the foreign key in the diagram. It relates one table to another table.

3. View Schema

The view level design of a database is known as **view schema**. This schema generally describes the end-user interaction with the database systems.

**Difference between the Physical and Logical Database Schema**

|  |  |
| --- | --- |
| **Physical database schema** | **Logical Database schema** |
| It does not include the attributes. | It includes the attributes. |
| It contains both primary & secondary Keys. | It also contains both primary & secondary keys. |
| It contains the table name. | It contains the names of the tables. |
| It contains the column names and their data types. | It does not contain any column name or datatype. |

Types of DBMS

**TYPES OF DBMS**

The types of DBMS based on data model are as follows −

* Relational database.
* Object oriented database.
* Hierarchical database.
* Network database.

**Relational Database**

A relational database management system (RDBMS) is a system where data is organized in two-dimensional tables using rows and columns.

* This is one of the most popular data models which is used in industries. It is based on SQL.
* Every table in a database has a key field which uniquely identifies each record.
* This type of system is the most widely used DBMS.

Relational database management system software is available for personal computers, workstation and large mainframe systems.

For example − Oracle Database, MySQL, Microsoft SQL Server etc.

| **Std ID** | **Name** | **City** |
| --- | --- | --- |
| 201 | Bob | Hyderabad |
| 204 | Lucky | Chennai |
| 205 | Pinky | Bangalore |

In the above student table Std ID, Name and city are called as attributes and their values. Std ID is a primary key attribute which uniquely identifies each record in the student table.

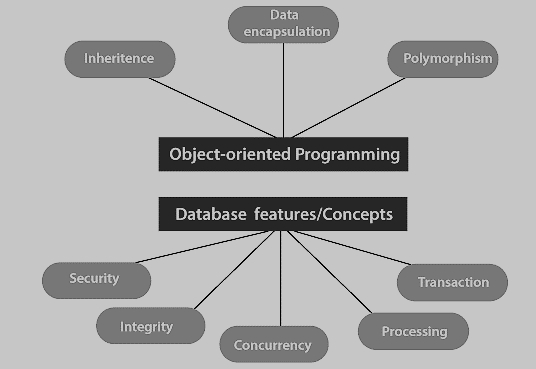
**Object Oriented Database**

It is a system where information or data is represented in the form of objects which is used in object-oriented programming.

* It is a combination of relational database concepts and object-oriented principles.
* Relational database concepts are concurrency control, transactions, etc.
* OOPs principles are data encapsulation, inheritance, and polymorphism.
* It requires less code and is easy to maintain.

For example − Object DB software.

The object oriented database is represented in diagram format below −

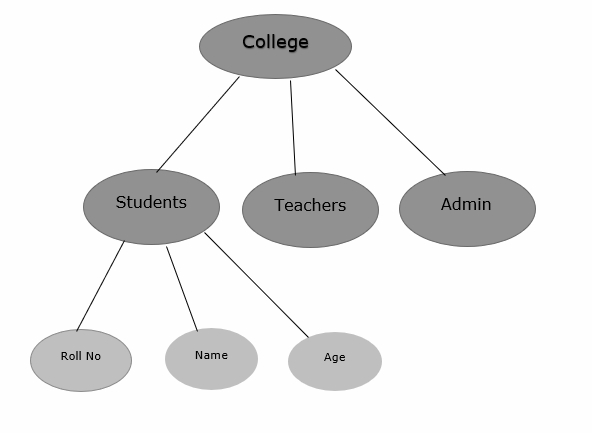


**Hierarchical Database**

It is a system where the data elements have a one to many relationship (1: N). Here data is organized like a tree which is similar to a folder structure in your computer system.

The hierarchy starts from the root node, connecting all the child nodes to the parent node.

* It is used in industry on mainframe platforms.
* For example− IMS(IBM), Windows registry (Microsoft).
* An example of a hierarchical database is given below −



**Network database**

A Network database management system is a system where the data elements maintain one to one relationship (1: 1) or many to many relationship (N: N).

It also has a hierarchical structure, but the data is organized like a graph and it is allowed to have more than one parent for one child record.

Example

Teachers can teach in multiple departments. This is shown below −

