Name:Prathamesh Arvind Jadhav

# DBMS (DataBase Management System)

- **Introduction:**

## ➢ What is Data?

In computer science or DBMS, data is an efficient model of information, which is stored in a format that is highly efficient for movement over various networks.

It can be internal networks of the computer itself(i.e local movement in storage disk of a computer) or global movement via the internet.

**Data is generally a raw format of any information**. It may be text information or any rich formatted information.

*Consider you're visiting a website, your computer downloads all the HTML, CSS and various java-script for that website and renders the page. All this is datas download.*

**Information**
Information is nothing but that data being viewed in a structured format.

Example, when you visited that website. The browser stores your cookies and the website based on your IP address can find the country you're accessing the website from. Thus, your IP Address which is datas becomes information when a meaning full context is applied to it.

Name:Prathamesh Arvind Jadhav

## Units of data storage

***Bit:***

The smallest/fundamental unit of measurement used for measuring data is a single bit, it contains a binary value such as true/false(1 and 0's)

***Byte***

- 
    - A byte corresponds to 8 bits of data which is the fundamental unit of measurement of data
    - A byte can Store $2^n$ bits i.e 1 byte = $2^8$ bits = 256 different values

***Kilobyte (KB)***

Files required thousands of bytes to store, textual data files are often measured in ***kilobytes(KB).***

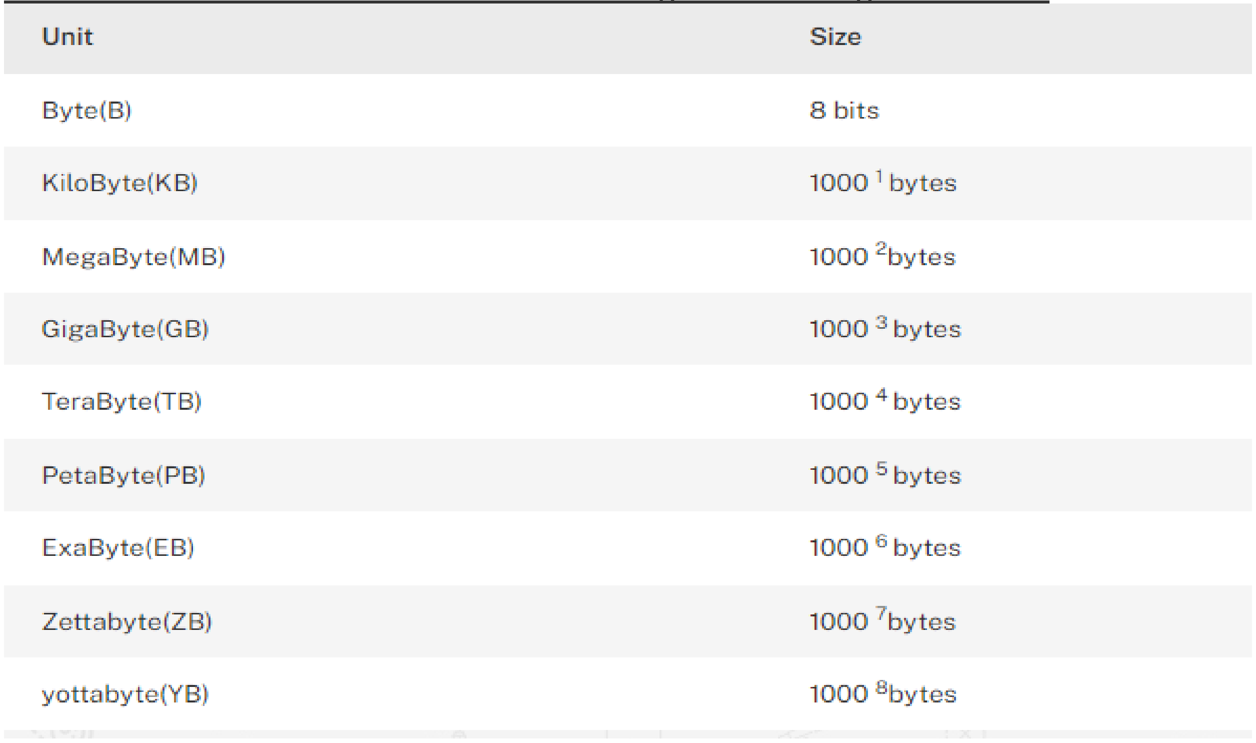
***Megabytes(MB):***

Larger files such as images, audio files, videos contain millions of bytes hence visual and audio datas stored in **MB**.

***Gigabytes(GB):***

Storage devices store thousands of files in a wide range of formats varying from a simple text file to high-end graphical videos audios images, etc to store data in the form of **gigabytes** or even ***terabytes (TB).***

**Below is a list of standard measure of data storage in increasing order of size**



| Unit | Size |
|---|---|
| Byte(B) | 8 bits |
| KiloByte(KB) | $1000^{1}$ bytes |
| MegaByte(MB) | $1000^{2}$bytes |
| GigaByte(GB) | $1000^{3}$ bytes |
| TeraByte(TB) | $1000^{4}$ bytes |
| PetaByte(PB) | $1000^{5}$ bytes |
| ExaByte(EB) | $1000^{6}$ bytes |
| Zettabyte(ZB) | $1000^{7}$bytes |
| yottabyte(YB) | $1000^{8}$bytes |

## ➢ What is Database?

**The database is a collection of related data, organized in a structured and meaningful way.** For computers, having the data stored in a way that is allowed faster –

- Storage(entry)
- Access
- Update
- Manipulation

Is also important than just having meaningfully sorted and stored. Consider this, there is a phonebook directory, with 1 million + business contacts in your city and the entry is done randomly.

To view this large data, you need to have the name of the business, phone numbers, addresses arranged together in an alphabetical format. Then only it will be called a databases of numbers right.

The database is a container of multiple tables, views, procedures, functions, triggers, and other database objects. The data is stored in the form of tables and can be queried, updated, and manipulated using SQL (Structured Query Language) commands. The DBMS is responsible for managing and maintaining the integrity, security, and consistency of the data in the database.

This is why Database is a most efficient and structured way of storing information together so various operations like entry, addition, updation and manipulation are very easy, along with the view format as well.

***Databases in early systems***

The databases in early systems were stored on tapes, which were read only format, which means once you store data. It is impossible to delete or update or even add any new data on it easily.

Which is why the father of database system, Founder of Oracle, **Larry Ellison** found the need to a fully managed database system and launched oracle in the market.

## ➢ What is Database Management System?

**The database management system is software that controls all the different manipulation of stored or to be stored data in a database.** It allows the creation, update, manipulation, definition of a database.

As clear from the name it manages the whole database end to end for whatever operation that may be required. Some examples of DBMS are –

- SQL
- mySQL
- noSQL
- Oracle
- IBM DB2
- PostgreSQL
- MongoDB

A database management system perform the following –

- Database Definition
- Data Updation
- Data Retrieval
- Administration
- Security

## ***Database Definition***

This essentially defines how data is defined in the database. For example we have to create a database for students are PrepInsta users.

We will define the data as follows –

- Unique ID of signed up user
- Unique email ID of signed up user
- First Name
- Last Name
- Passwords etc

## ***Data Updation***

This majorly allows various operations like insertion of the new data, updating new data, deletion of data in the various database tables.

- Insertion – Inserting information of new user signed up on PrepInsta website
- Updation – Updating the password of the user who requested to change it
- Deletion – Deleting the complete or partial data of the user who has requested account deletion on PrepInsta.

***Data Retrieval***

The data is stored in various database servers and stored globally. For different instances like requesting login authentication details for a website. You need to communicate and retrieve a lot of data from database tables.

***Administration***

Database administration is important, a for a new employee who is working on database system for your company. You don’t want to give him deleting capabilities. So you create different roles like –

- Database Administrator – Who call perform all actions like modification, deletion, updation and creation etc.
- Database Manager – Who can define, update, modify but can’t delete
- Database Editor – Who can just retrieve and insert in the database

**Note** – The above are just for examples. Companies create their custom roles based upon exact requirements.

***Security***

The security of the system is also important. You don’t want hackers to steal your confidential database or maybe update your database with false data. Thus for any access to database node. There is an authentication that happens along.

## ➢ File System Vs DBMS

File System is an old method of storing data in set of files. It helps us to organize the data and allows easy retrieval of files when they are required whereas DBMS System is the new method of storing data in a tabular format. It consists of a group of programs that manipulate the database.

## Difference Comparison

- **File System** **–** Earlier the data were stored in a file format where the **data is stored in one file or set of files**. These files have no logical relation with one another and data is just thrown in them. With the user needing to remember the logical relationships between them on his own.

Imagine, having multiple files, one for usernames, one for passwords, one for email ID, and maybe 1000’s of such individual files. The user will have to logically remember the relationship amongst them on his own.

- **DBMS System** – This is new-age **software that maintains the data stored in a tabular format** with various different entities having logical relationships amongst one another. The software gives the programmer easy methods to do various operations like creation, definition, updating, deletion, access modification rights, etc.

Imagine, having one single table named as a user with columns being email ID, userID, Username, Firstname, LastName, Passwords, etc. Each row contains information for one user of the website and one more table containing product information. Which has a common column userID(Same that of user table) and further columns containing information like – date of purchase, productID, validity, etc.

## Advantage Comparison

***Data concurrency –***

Imagine a banking transaction, where your bank balance is 1000Rs and you try to take out 900Rs. cash from two different ATMs at the same time. In such cases the ATM will check that your current balance is sufficient and allows transaction. But, you're able to withdraw 1800 cash right as both transactions were done at the same time. This called concurrency.

There is a need of a locking mechanism which makes sure that these situations don't arrive. In File systems we don't have such system to manage such situations. However, modern DBMS systems by default provide inbuilt concurrency handling systems.

***Data Searching –***

It is very easy in DBMS systems, which have most of the highly efficient search operations inbuilt. User only needs to write a one line query to get the results. However, in file systems, you need to write your own long code for various search operations.

***Data Integrity –***

When a data is being inserted in the database there maybe cases when, we need only a particular type of data being entered. Like RollNo should only accept int values. Such constraints are there in DBMS systems but not in File systems.

***Data Sharing –***

Data sharing and access rights are easy and inbuilt in the DBMS system. But in file systems its too complex.

***Data Redundancy and Inconsistency –***

Imagine Amazon.com having two different tables for users and product purchase, with both having email ID as one data. Now, the user may change his/her email ID. Now, this may only reflect in user table, but not in the product table. Which may cause data inconsistency in file sharing systems. This is handled really well in DBMS system because of its inter-dependent relationship in nature.

***Tabular Format***

Data stored in DBMS systems mostly is in tabular format . However, in file systems its generally not the case.

| ID | Name | Age | Salary |
|---|---|---|---|
| 1 | Adam | 34 | 13000 |
| 2 | Alex | 28 | 15000 |
| 3 | Stuart | 20 | 18000 |
| 4 | Ross | 42 | 19020 |

## ➢ TUPLE

Each tuple represents a complete record of the specific data item, each tuple stores different data with the same structure.

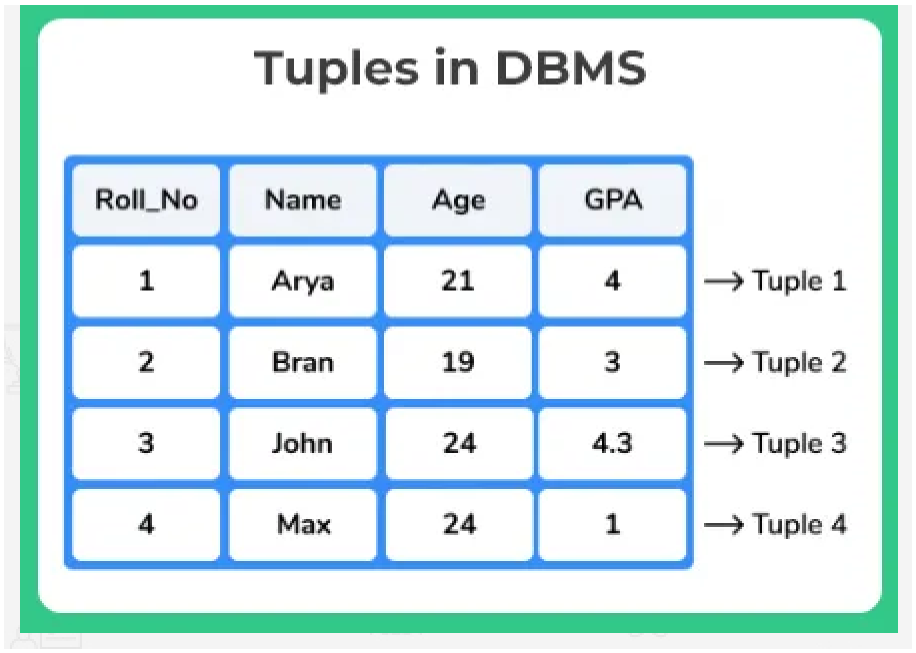
Example : For example in an employee table, we ***need to store the details of 10 employees then the database table will consist of 10 tuples*** every each tuple contains details of that particular employee i.e 1 tuple for storing the record of one employee

**Consider a sample table '*emp'* .**

| ENAME | JOB | SAL | HIREDATE | HIREDATE+2 |
|---|---|---|---|---|
| KING | PRESIDENT | 5000 | 17-NOV-81 | 19-NOV-81 |
| BLAKE | MANAGER | 2850 | 01-MAY-81 | 03-MAY-81 |
| CLARK | MANAGER | 2450 | 09-JUN-81 | 11-JUN-81 |
| JONES | MANAGER | 2975 | 02-APR-81 | 04-APR-81 |
| SCOTT | ANALYST | 3000 | 19-APR-87 | 21-APR-87 |
| FORD | ANALYST | 3000 | 03-DEC-81 | 05-DEC-81 |
| SMITH | CLERK | 800 | 17-DEC-80 | 19-DEC-80 |

Table contents seven rows i.e it stores the details of all the 7 employees in 7 separate tuples

Name:Prathamesh Arvind Jadhav



Tuples in DBMS

| Roll_No | Name | Age | GPA | |
|---|---|---|---|---|
| 1 | Arya | 21 | 4 | → Tuple 1 |
| 2 | Bran | 19 | 3 | → Tuple 2 |
| 3 | John | 24 | 4.3 | → Tuple 3 |
| 4 | Max | 24 | 1 | → Tuple 4 |

***<u>SQL Query to retrieve particular tuples</u>***

```
SELECT * FROM EMP
WHERE DEPTNO=20;
Output:
4 rows selected.
```

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | – | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | 500 | 20 |

Only those tuples corresponding to the details of Department number 20 are displayed

***<u>ROW NUM in SQL</u>***

- The ***rownum*** clause in SQL is used to sp***ecify the number of tuples present in a database table***

Name:Prathamesh Arvind Jadhav

- Each tuple will be added a number starting from 1 to n so that the user can easily identify the number of tuple present in the table

***Example***

```
select rownum,empno,ename,job,sal,deptno
FROM EMP;
Output:
7 rows selected.
```

| ROWNUM | EMPNO | ENAME | JOB | SAL | DEPTNO |
|---|---|---|---|---|---|
| 1 | 7839 | KING | PRESIDENT | 5000 | 10 |
| 2 | 7698 | BLAKE | MANAGER | 2850 | 30 |
| 3 | 7782 | CLARK | MANAGER | 2450 | 10 |
| 4 | 7566 | JONES | MANAGER | 2975 | 20 |
| 5 | 7788 | SCOTT | ANALYST | 3000 | 20 |
| 6 | 7902 | FORD | ANALYST | 3000 | 20 |
| 7 | 7369 | SMITH | CLERK | 800 | 20 |

.

## ***Displaying top N tuples***

Suppose if you want to ***display the top three tuples present in emp table*** then you can use this ***rownum*** along with the relational operator

***Example***

```
select rownum,empno,ename,job,sal,deptno
FROM EMP
where rownum<=3;
Output:
3 rows selected.
```

| ROWNUM | EMPNO | ENAME | JOB | SAL | DEPTNO |
|---|---|---|---|---|---|
| 1 | 7839 | KING | PRESIDENT | 5000 | 10 |
| 2 | 7698 | BLAKE | MANAGER | 2850 | 30 |
| 3 | 7782 | CLARK | MANAGER | 2450 | 10 |

Only the first 3 tuples in a table or displayed

Name:Prathamesh Arvind Jadhav

## ➢ RECORD

Each record represents complete information of the specific data item, each record stores different data with the same structure

Example : For example in an employee table, we ***need to store the details of 10 employees then the database table will consist of 10 records*** every each record contains details of that particular employee i.e 1 record for storing the information of one employee

**Consider a sample table '*emp*'**

| ENAME | JOB | SAL | HIREDATE | HIREDATE+2 |
|---|---|---|---|---|
| KING | PRESIDENT | 5000 | 17-NOV-81 | 19-NOV-81 |
| BLAKE | MANAGER | 2850 | 01-MAY-81 | 03-MAY-81 |
| CLARK | MANAGER | 2450 | 09-JUN-81 | 11-JUN-81 |
| JONES | MANAGER | 2975 | 02-APR-81 | 04-APR-81 |
| SCOTT | ANALYST | 3000 | 19-APR-87 | 21-APR-87 |
| FORD | ANALYST | 3000 | 03-DEC-81 | 05-DEC-81 |
| SMITH | CLERK | 800 | 17-DEC-80 | 19-DEC-80 |

Table contents seven rows i.e it stores the details of all the 7 employees in 7 separate records

***SQL Query to retrieve particular records***

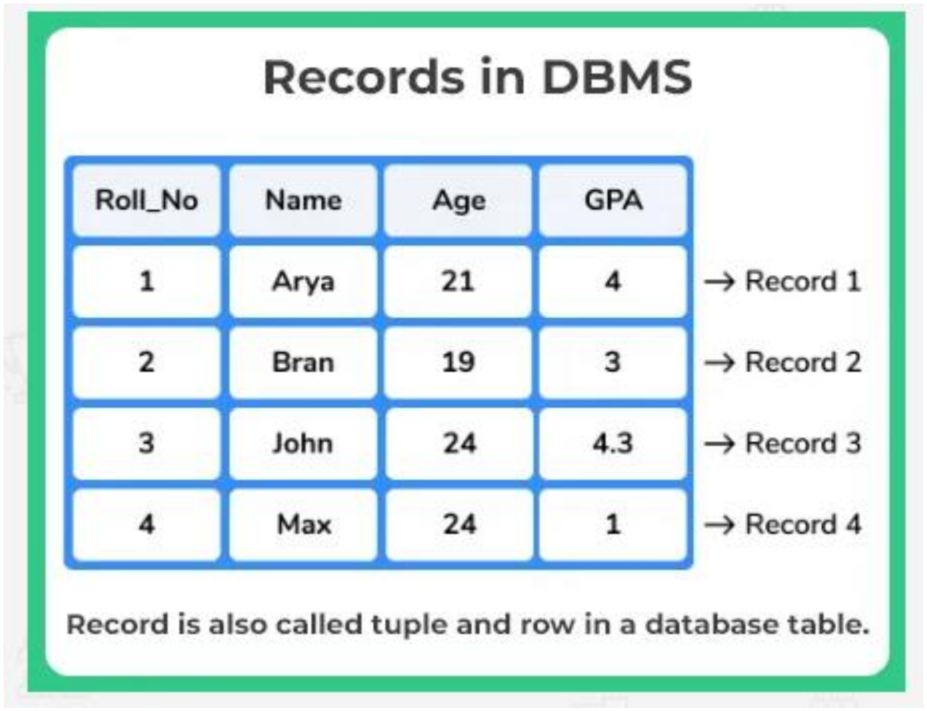
```
➢ SELECT * FROM EMP
➢ WHERE DEPTNO=20;
➢ Output:
➢ 4 rows selected.
```

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | – | 20 |

Name:Prathamesh Arvind Jadhav

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | 500 | 20 |

Only those records corresponding to the details of Department number 20 are displayed.



| Roll_No | Name | Age | GPA |
|---|---|---|---|
| 1 | Arya | 21 | 4 |
| 2 | Bran | 19 | 3 |
| 3 | John | 24 | 4.3 |
| 4 | Max | 24 | 1 |

***ROW NUM in SQL***

- The ***rownum*** clause in SQL is used to sp***ecify the number of rcords present in a database table.***
- Each record will be added a number starting from 1 to n so that the user can easily identify the number of records present in the table.

***Example***

```
select rownum,empno,ename,job,sal,deptno
FROM EMP;
```

Name:Prathamesh Arvind Jadhav

```
Output:
7 rows selected.
```

| ROWNUM | EMPNO | ENAME | JOB | SAL | DEPTNO |
|---|---|---|---|---|---|
| 1 | 7839 | KING | PRESIDENT | 5000 | 10 |
| 2 | 7698 | BLAKE | MANAGER | 2850 | 30 |
| 3 | 7782 | CLARK | MANAGER | 2450 | 10 |
| 4 | 7566 | JONES | MANAGER | 2975 | 20 |
| 5 | 7788 | SCOTT | ANALYST | 3000 | 20 |
| 6 | 7902 | FORD | ANALYST | 3000 | 20 |
| 7 | 7369 | SMITH | CLERK | 800 | 20 |

## ***Displaying top N records***

Suppose if you want to ***display the top three records present in emp table*** then you can use this ***rownum*** along with the relational operator.

## ***Example***

```
select rownum,empno,ename,job,sal,deptno
FROM EMP
where rownum<=3;
Output:
3 rows selected.
```

| ROWNUM | EMPNO | ENAME | JOB | SAL | DEPTNO |
|---|---|---|---|---|---|
| 1 | 7839 | KING | PRESIDENT | 5000 | 10 |
| 2 | 7698 | BLAKE | MANAGER | 2850 | 30 |
| 3 | 7782 | CLARK | MANAGER | 2450 | 10 |

Only the first 3 records in a table or displayed

Name:Prathamesh Arvind Jadhav

# Architecture and Models

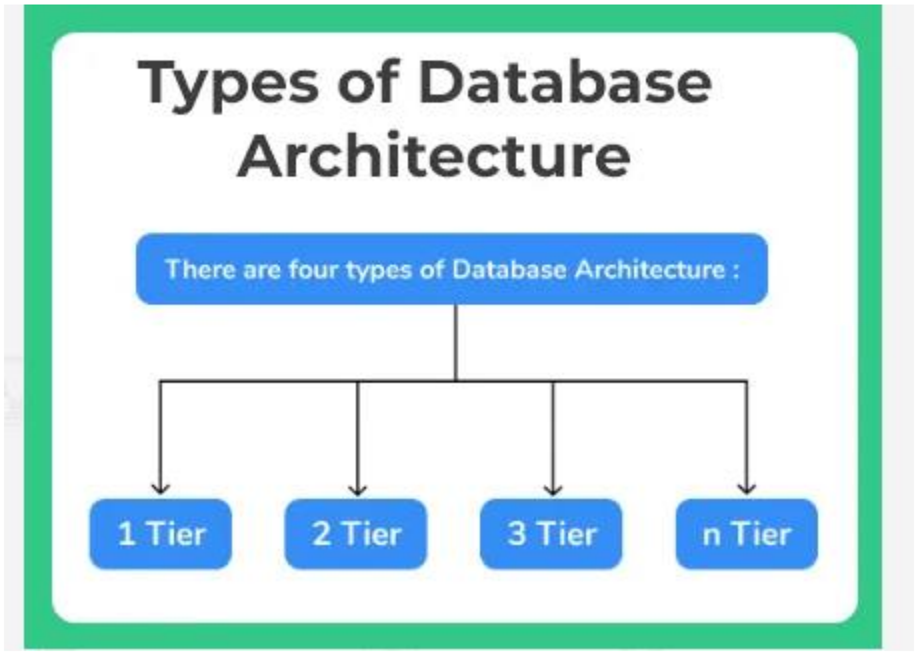
## ➢ DBMS Architecture

**A DBMS has a three-level architecture, consisting of the external level, the conceptual level, and the internal level. This architecture allows for the separation of data and the logic that accesses the data, which enables the database to be updated and optimized without affecting the end-users or the administrator.**

**DBMS Architecture (What is Architecture in DBMS)**

The following are different types of database architectures, based upon the needs and requirements of the system. One chooses the correct suitable architecture for them –

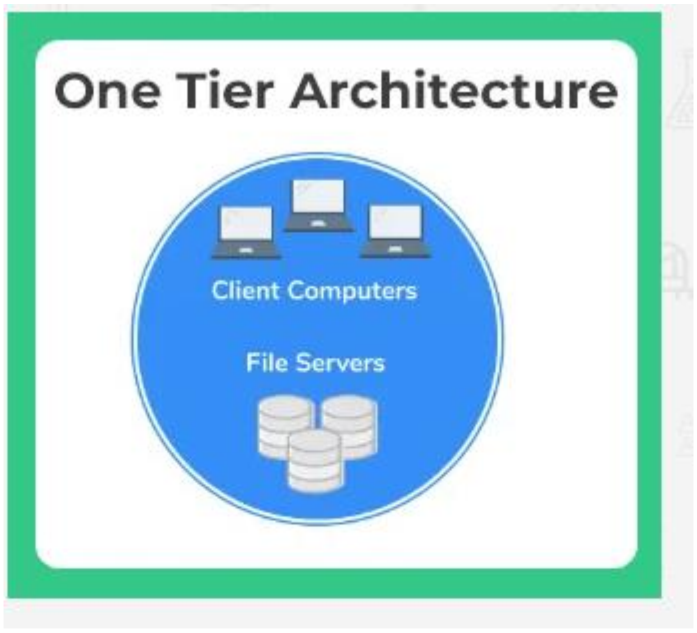
- 1 Tier Architecture
- 2 Tier Architecture
- 3 Tier Architecture
- n Tier Architecture


Types of Database Architecture
There are four types of Database Architecture :
1 Tier
2 Tier
3 Tier
n Tier

**Single Tier Architecture**

In such systems all the required components like – the interface, Middleware, and back-end data, all on one server or platform.

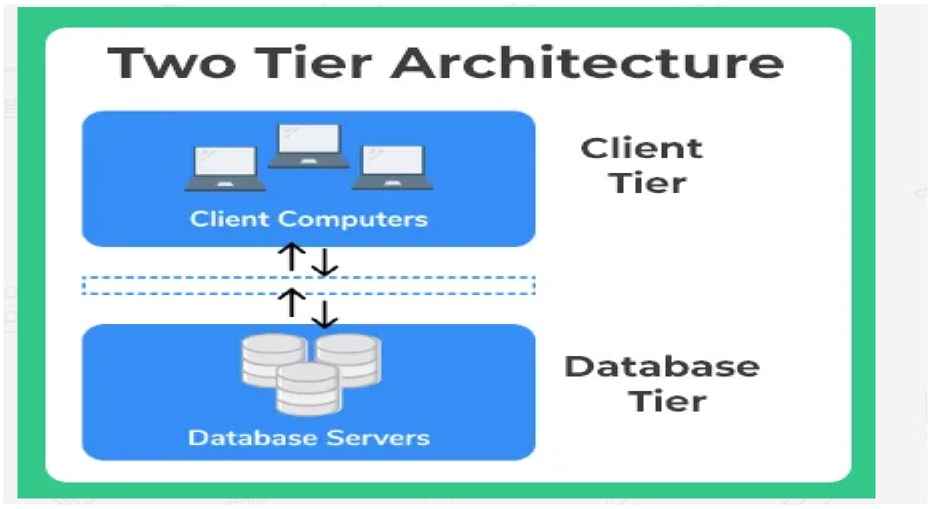
Most of the developers choose such systems as it's easy and simple. But, it doesn't provide many tools for the end-users. Thus, it is not suitable for data delivery platforms like websites globally but, may be useful for storing large data.



**Two Tier Architecture**

In such systems, the communication is between the client and server. The client is where the data needs to be delivered to and the server is where data is stored, updated and processed and communication between the two happens.

The application tier is there to provide communication between the two and is independent in terms of the code level logic happening at server or client.

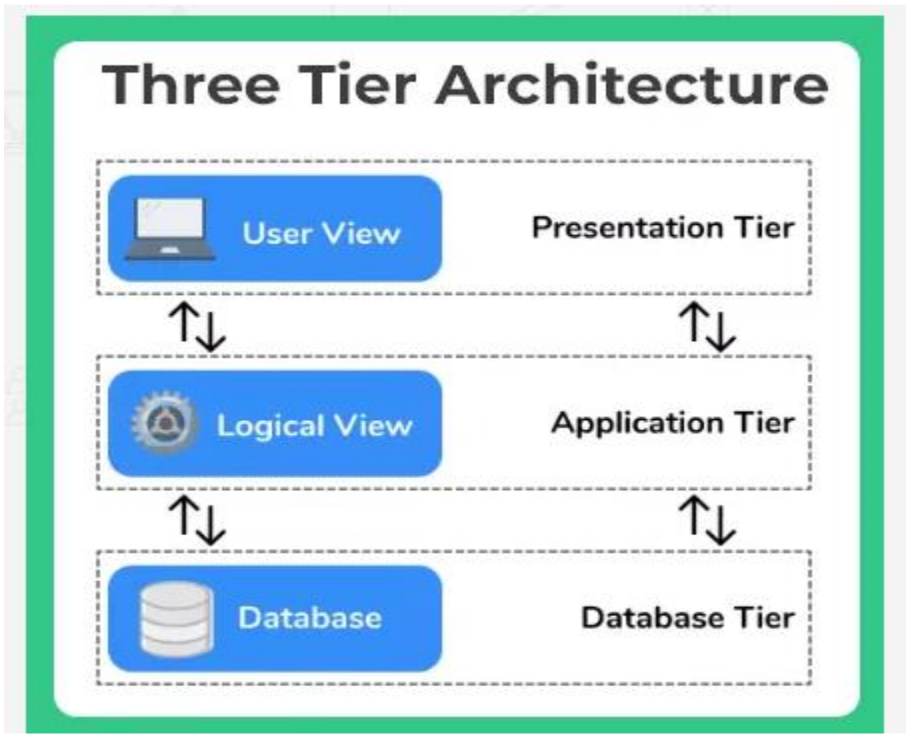


Name:Prathamesh Arvind Jadhav

## Three Tier Architecture

This is the most widely used architecture these days and all the tiers are separated from one another. The following are the tiers –

- **Database Tier –** The raw data is present at this tier along with its DBMS system language and its queries that allow different operation on the DB. Example – All the database of PrepInsta website's user resides on this tier
- **Application Tier –** This acts as an intermediary between the database tier and the user. The code level program that can access the database for various operation sits here. For example – if one wants to write a code for updating email ID in the database. In this case the user is not interacting with the DB directly. But, is interacting with application tier, which does its code level processing and then interacts with the database.
- **Presentation Tier –** This also known as user tier. Here the user is able to view all the database results. Best for his viewable format, like – if user is accessing its profile page. All the details were sent from DB to application tier and then sent to presentation tier which then displays the same in a view so that user can understand and read it. HTML/CSS is also used to modify how this data looks to the end user.

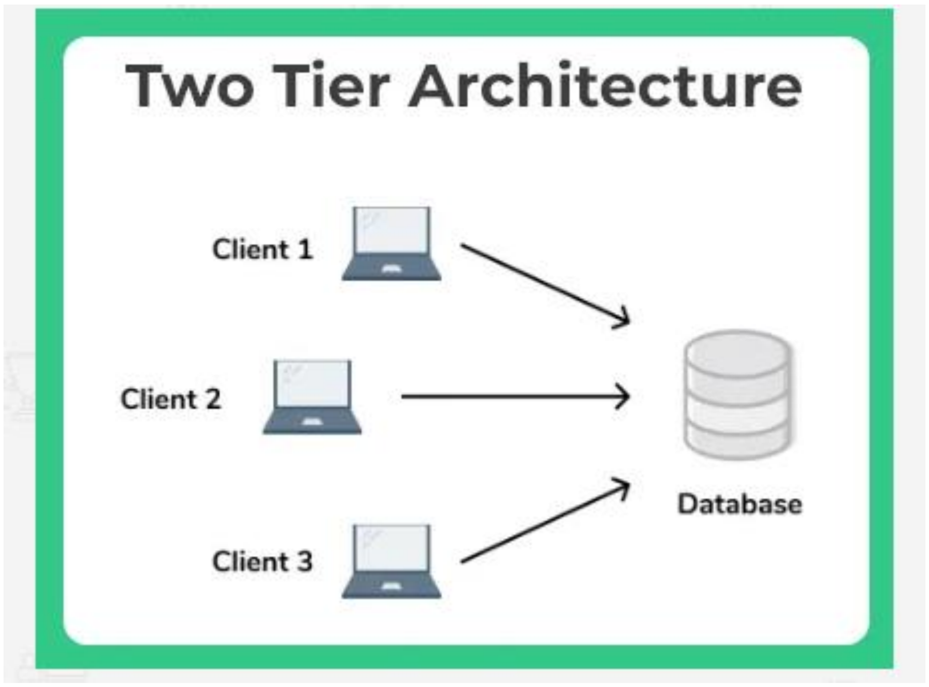


**n Tier Architecture**
The n tier is more customised format for corporation. Where there are n different divisions of the whole format which are related but independent of one another.

## 2 Tier Architecture in DBMS

**2 tier architecture in DBMS**
On this page, we will learn about 2 Tier Architecture in DBMS.

- DBMS ***architecture plays a key role in the design, development, implementation, and maintenance*** of the database management system of the company
- The ***proper selection of database architecture will solve many design problems*** initially and also helps in quick and secured data access
- Any database management system uses any of the following to 2 architectures
    - 2 tier architecture (two-level)
    - 3 tier architecture (three-level)



***<u>Two Tier architecture</u>***
The two-tier architecture is similar to the basic Foundation model i.e ***client-server model.***

**Components of two tier architecture**
It consists of two tiers

1. Client-side application:

- 
    - ***User interfaces and application programs*** run generally on the client-side
    - It is generally ***a presentation layer that runs on a client*** (PC, Mobile, Tablet, etc)
    - This client application ***establishes a connection by sending a request to the server-side*** up an application in order to establish communication between the database

2. Server side application

- 
    - The server side is ***responsible for query processing and transaction management***
    - In this, client-side directly communicates with the database base that is present on the server-side and the server sends a response to the request received from the client
    - Simply ***server-side represented data is stored on a Server***.
    - Generally, APIS like ***ODBC, JDBC is used for this interaction*** i.e for sending requests from the client-side application.

***<u>Real-time examples for two-tier architecture</u>***

- Desktop applications excel sheets, word document, desktop games
- The ***contact management system that is created using MS Access*** is the most observed example is for this two-tier architecture.

***<u>Advantages of two-tier architecture</u>***

- As it is simple and easily understandable ***provides direct and faster communication***
- ***Compatible*** with existing systems

- Provides some ***secretary to the DBMS*** as it is not directly e exposed to the end-user range.

***Limitation of this two-tier architecture***

It ***cannot be used for dynamic web applications*** where there are a large number of users and generally used for static desktop and small web applications.

## 3 Tier Architecture in DBMS

### 3 Tier Architecture in DBMS

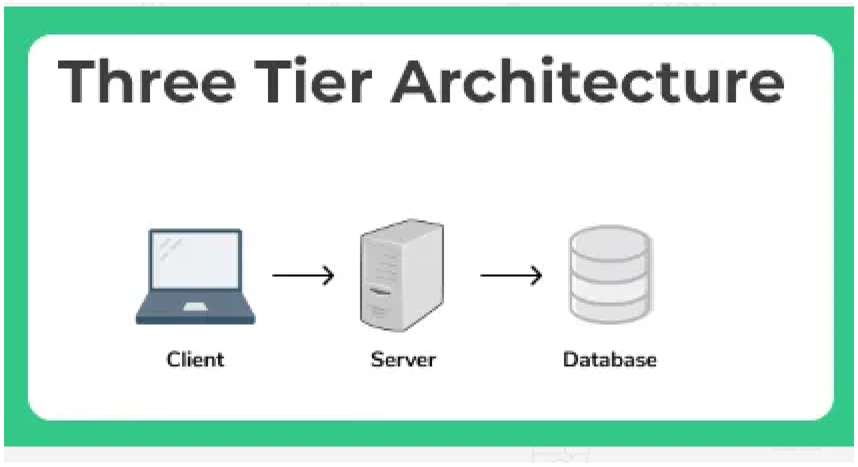
In a 3-tier architecture, the DBMS is split into three parts: the client, the application server, and the database server. On this page, we will learn about 3 Tier Architecture in DBMS.

- DBMS architecture ***plays a key role in the design, development, implementation, and maintenance*** of the database management system of the company
- All the dynamic web application over the internet uses this three-tier architecture
- The proper selection of database architecture will solve many design problems initially and also helps in quick and secured data access
- Any database management system uses any of the following 2 architectures
  - 2 tier architecture (two-level)
  - 3 tier architecture (three-level)

## 3 tier architecture

- 3 tier architecture is the most popularly used DBMS architecture
- As the name suggests three tiers consists of three components in addition to the client-side and server-side application of two-tier architecture, it consists application server layer as an intermediate tier between these two.

Name:Prathamesh Arvind Jadhav



**Components of 3 tier architecture**

1. Client-side :

- 
    - It is nothing but the ***presentation layer*** (your PC, Tablet, Mobile, etc.)
    - It ***sends a request to the server-side*** via the application server layer

2. Application server :

- 
    - It is present as an ***intermediate tier between the client and server tires***
    - Unlike two-tier architecture, in this requests are not directly sent from the client to the server-side, ***when a client sends a request first it is transferred to the application server then this application server transfers the request to the server-side***, followed by query processing and transaction management
    - This intermediate layer ***also acts as a senses medium*** for the exchange of partially processed data between server and client

3. Server-side:

- 
    - Nothing but the ***database of the server-side application***
    - It ***sends responses to the requests received from the client-side to the application server*** and the application server, in turn, transfer them to the client-side.

***Examples of three-tier architecture***

- All ***large dynamic web applications present over the internet*** or examples of this 3 tier architecture
- A large ***website like prepinsta.com*** over the internet is the best example, isn't it??

***Purpose of three-tier architecture***

- ***Integrity***: because of the middle layer between the client and server-side data corruption can be removed
- ***Security***: as there is no direct interaction between client and server we can restrict unauthorized access if any
- To provide a clear ***separation between the user application and physical database***
- Program ***data independence and multiple views of data***
- Because of the above features, a large number of users can access a database easily

***Demerits***

- There might be ***complexity in implementation and communication***
- Sometimes it becomes ***difficult for interaction*** due to the presence of middle layers and ***responses may be delayed*** but they can be easily overcome by efficient server administration.

## Need for DBMS

### Need for DBMS

The need for a DBMS arises from the fact that data is a vital component of any organization and managing it efficiently and effectively is crucial to the success of the organization. A DBMS allows for the centralization of data, data integrity, data security, data sharing and data consistency, making it an essential tool for any organization. On this page, we will learn about the need for DBMS.

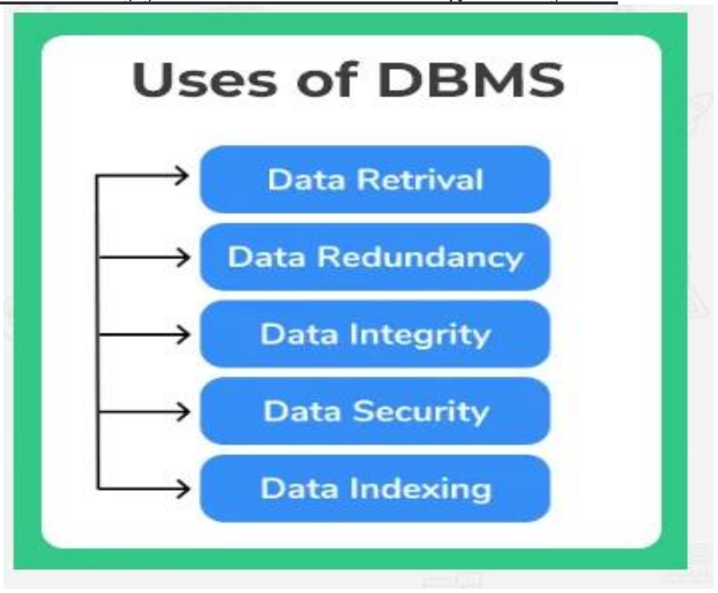
Name:Prathamesh Arvind Jadhav

## Uses of DBMS

- The file management system is a traditional mechanism to store data permanently into secondary devices but in today digital word TBs of data need to be stored in an organized place which needs security as well as retrieval in seconds of time
- Traditional file systems have the very lowest level of storage and processing capabilities which created problems regarding security, integrity, memory storage and duplicate data, data inconsistency and so on

For these problems, DBMS is a solution

***Let's discuss the top 5 reasons why you need a database management system***



1. Data retrieval
If you want to retrieve data from the flat file then we must develop application programs in a high-level language, so that ***data can be stored and retrieved fastly and securely within the time bound***

Ex: SQL – structured query language

2. Data redundancy
In any storage, we need to make copies of data for backup but in traditional file management systems once we update data in one location sometimes it fails to get updated in the copy of the data, so that it may create problems of inconsistency this rate is ***called duplicate data or redundant data***

- 
    - The database automatically maintains consistent data through a transaction using certain rules and procedures
    - Each transaction internally follows four properties known as acid properties(atomicity, consistency, durability, isolation)
    - The database is capable of eliminating all problems of insertion-deletion updation of data through levels of the normalization process.

3. Data integrity
Data integrity ensure  that only  required  data is stored in the database.e data is validated before entered into the database using integrity constraints such as primary key, foreign key, etc.

4. Data security
In traditional file management, there is no authentication mechanism at high-end whereas DBMS provides  levels of security authentication which can be done at user level admin level, etc

5. Data indexing

- If you want to retrieve data very fastly from the database we are using indexing mechanism whereas Flat files don't support indexing and solely  depend upon secondary storage devices
- Indexing is a mechanism where data is uniquely identified and stored using some computational techniques so that data is retrieved very fastly.

# Data Abstraction and Data Independence in DBMS

## Data Abstraction and Data Independence in DBMS

### Data Abstraction:

**Definition:** Data abstraction refers to the hiding of details from users at certain levels, for authentication and security purpose

**Any DBMS architecture mainly consists of three levels of**

1.

   1. ***Conceptual level***
   2. ***Internal  level***
   3. ***External level***

***Conceptual level***

- 
   - It describes the ***logical structure of the database***
   - Specifies what type of data can be stored in the database by defining the data type and data type signs and constraints like (primary key foreign key )etc
   - It also specifies the relationship between tables

**Example** : Create table emp(id num(5) primary key ,name varchar(10));

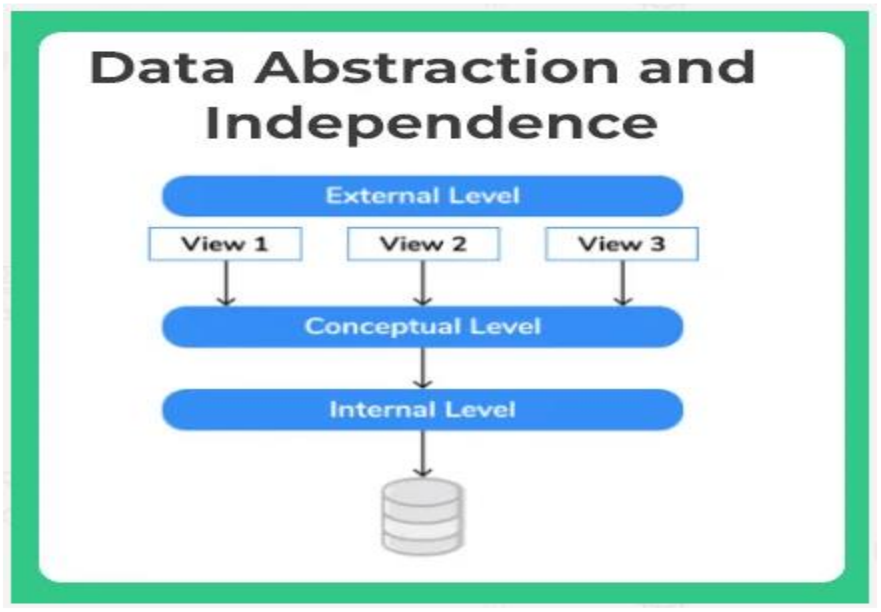
***External level***

- 
   - External level ***describe users view of the database***
   - It provides security mechanism i.e some users can access only a certain portion of data, it depends upon the database administrator which users can access the conceptual level and at what extent

***Internal level***

- 
   - This is the ***lowest level of abstraction describes how physical data is stored***
   - It provides details about the ***complex data structures that are used for storage of data***
   - Internal level provides indexes and  clusters  to control and manage the physically  stored data in hard disk

Name:Prathamesh Arvind Jadhav



## Data Independence

These levels of abstraction provide data independence ***i.e is all the transactions or changes made at one level are unaffected to other levels***

DBMS architecture provides two types of data independence

1.
    1. ***Logical data independence***
    2. ***Physical data independence***

***<u>Logical data independence</u>***

Logical data Independence states that ***external level is completely unaffected are free from any changes that are made at the conceptual level and vice-versa***
ex: Adding a new entity in the conceptual level should not affect the external level

***<u>Physical data independence</u>***

Physical data Independence states that ***conceptual level is completely unaffected are free from any changes that are made at the internal level and vice-versa***
ex: Adding a new entity in the internal level should not affect the conceptual level

## DBMS Database Models

Name:Prathamesh Arvind Jadhav

## DBMS Database Models

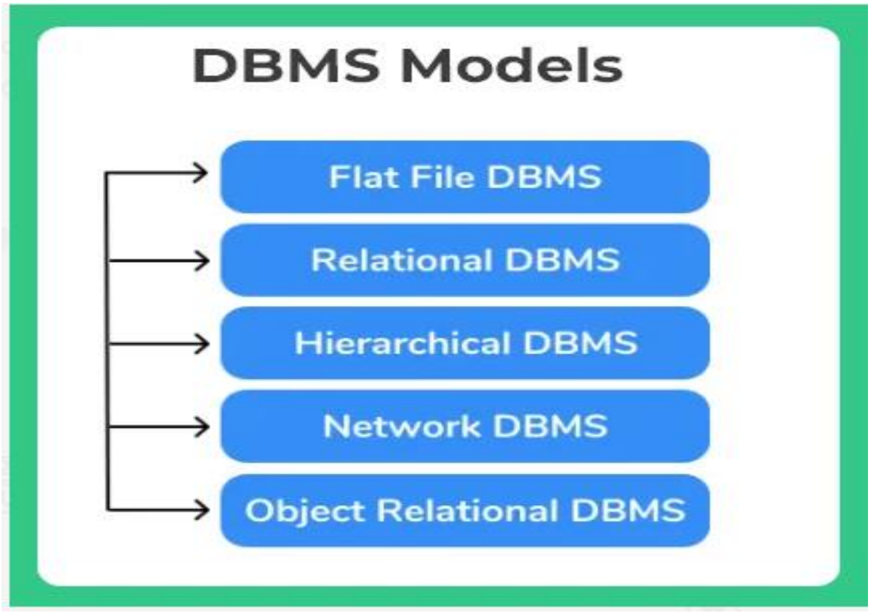
In the history of database design, 5 database models have been used.

***Flat file DBMS***

- The file management system was the first method used to store data in the computer is a database
- Each ***data item is stored on a disc sequentially in one large file***
- If you want to locate a particular item the search start from the beginning and each item is checked one by one sequentially till the match is found.

  **Drawbacks**

  Data duplication, very poor security, retrieving of data is very slow, no support for data, types dynamic memory allocation, concurrent data sharing is not allowed, no security mechanism through passwords, etc.



***Relational DBMS***

- It was first outlined by EF.codd 1970

The 2 components of the relational model are:

1.

1. collection of objects or relations that stores the data
2. operations that can act on relations to produce other relations

**This is the most frequent and commonly database today**

Data integrity, logical representation in the form of table rows and columns, no data redundancy, logical links between relations are maintained, null, values integrity constraint, provides high security, unlimited size, security, range of data types, multiple users can work at a time

***<u>Hierarchical DBMS</u>***

- organize data like a ***tree structure***, record types are the same as a relational data model
- ***The hierarchical data model is implemented based on one too many relations between parent and child records***
- Because of a parent, child records in one-many relation many child records have single parent records as a result of repetition of child records occur

***<u>Network DBMS</u>***

- in 1970s CODASY(conference on data system language) committee introduced a network data model
- in this data, the model will maintain less duplicate data because their ***parent-child records are implemented based on many-many relation***
- In this model also data is represented in the format of records and also record type is the same as a table in relation format

***<u>Object-relational DBMS</u>***

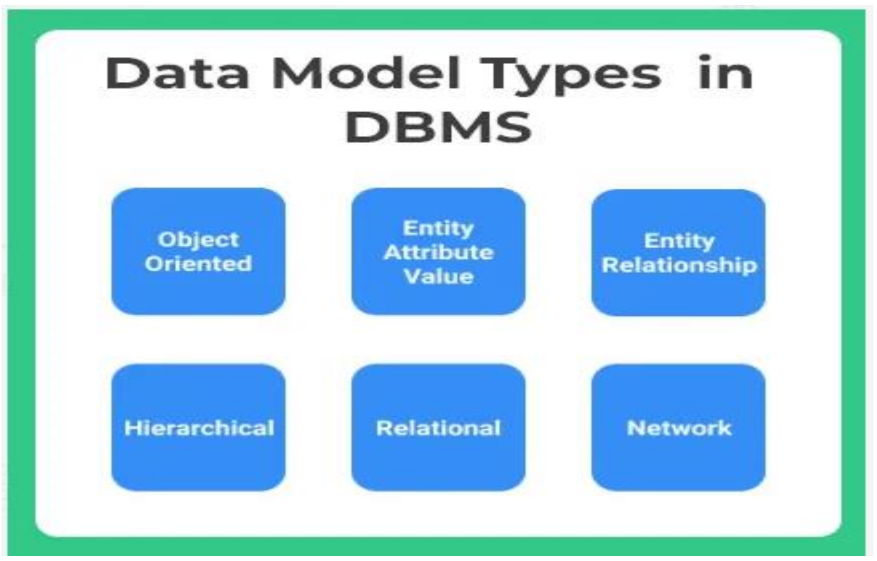
- In this model relational database but within object-oriented database design
- ***Object, class, inheritance is directly supported In database schema and the query language***
- It provides a middle ground between the relational database and object-oriented database
- are OOPS is added to the relational database design productivity, security boosts a lot

Name:Prathamesh Arvind Jadhav

# Data Models in DBMS

## Types of Data Models in DBMS

- Hierarchical database model
- Relational model
- Network model
- Entity-relationship model
- Object-oriented database model
- Document model
- Entity-attribute-value model
- Star schema

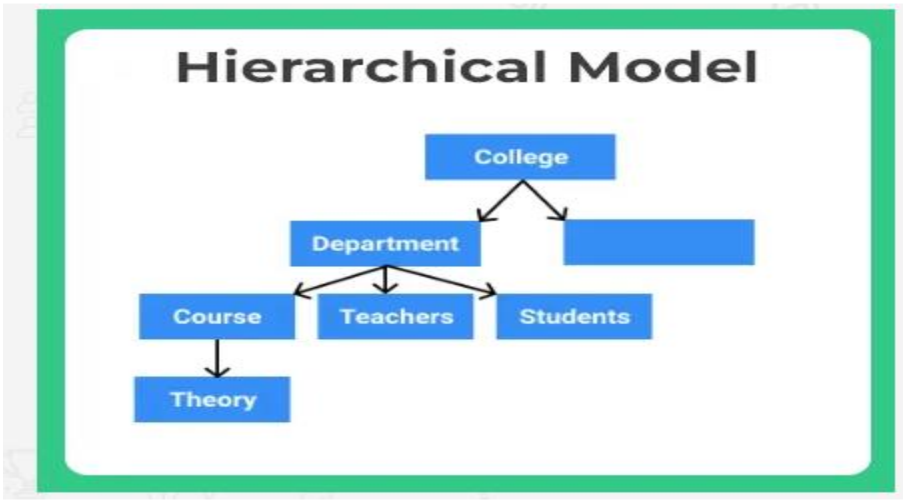


***<u>Hierarchical database model</u>***

The Hierarchical model is inspired from tree based data structure format. Where in there is a single root node and other data is linked to the same and expands like a tree.

1.
    1. A child node data may only have a single parent node
    2. For any parent node one to many relationship must be maintained.

Name:Prathamesh Arvind Jadhav



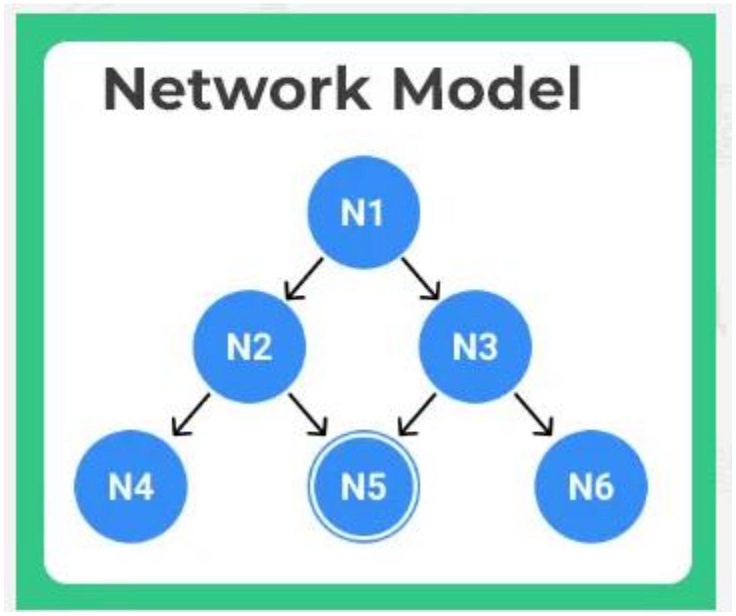
***Network model***

Network model was made to overcome the drawbacks fo hierarchical model. This works more like a graph rather than a tree.

As the name suggests there is network based looped relationship between various data linked to one another.

- 
  - A child node may have more than one parent. ( As for Node – N6)
  - There can be many to many relationships between data.

This was very much used before there was relational data model in practise. Since, there are various networks relationships. It was a very fast way to access any data.



Name:Prathamesh Arvind Jadhav

***Entity-relationship model***

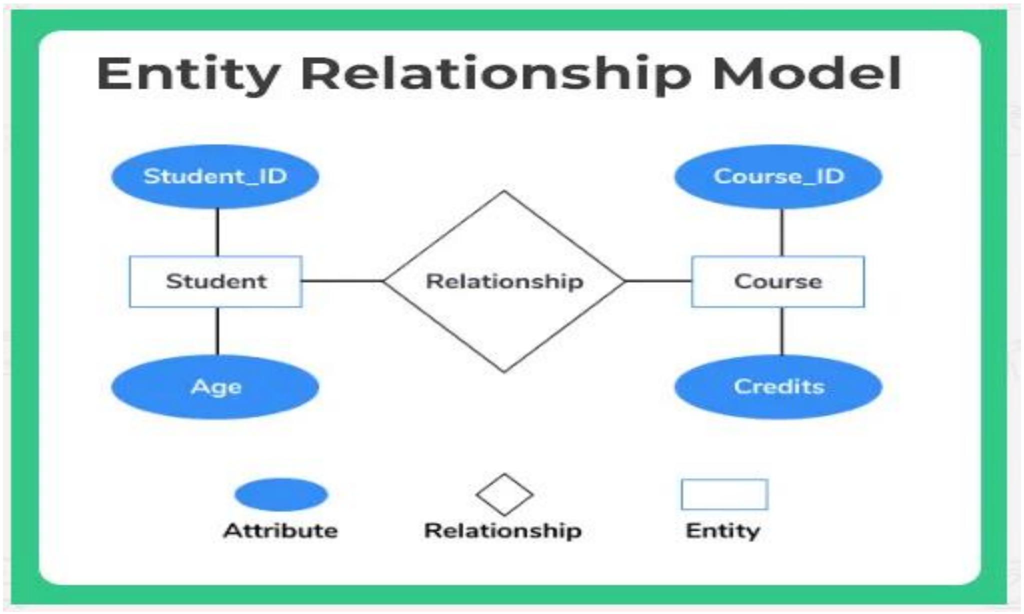
This is modeled about break and related model, i.e. we break data into different components based on characteristics, in terms of entity and attributes.

Imagine an entity student, its various attributes maybe studentID, age, dateOfBirth etc etc. It may have a relationship with courses offered to the student, which further may have attributes like CourseID and credits.

We will learn more about ER Diagrams in detail in further posts of ours.

In short Entity Relationship model is based on –

- 
    - **Entity –** These are representation of a real world entities in code.
    - **Attributes –** These are the characteristic properties of entities
    - **Relationship –** Logical relationship between various entities involved in the Database creation.
        - One to one
        - One to many
        - Many to one
        - Many to many



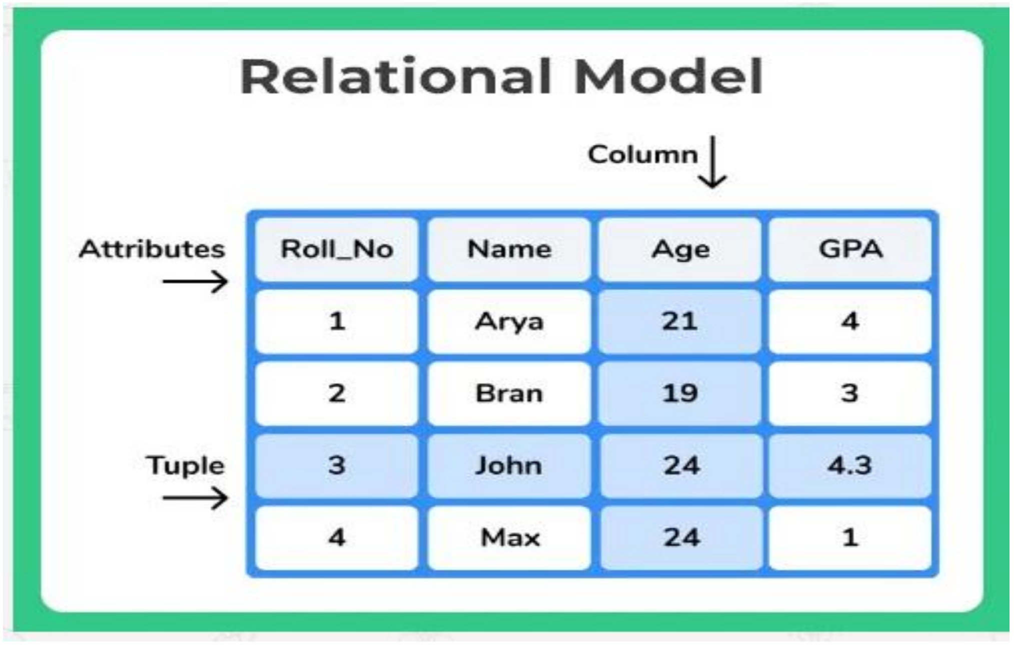
Name:Prathamesh Arvind Jadhav

***Relational model***

This is the most used model and even with DBMS that we will studying in future posts it will be based upon the same. Relational model is a tabular 2 dimensional relationship between various related tables having atleast one common field.

This works around

- 
    - Rows
    - Entities
    - Tuples
    - Columns
    - Table



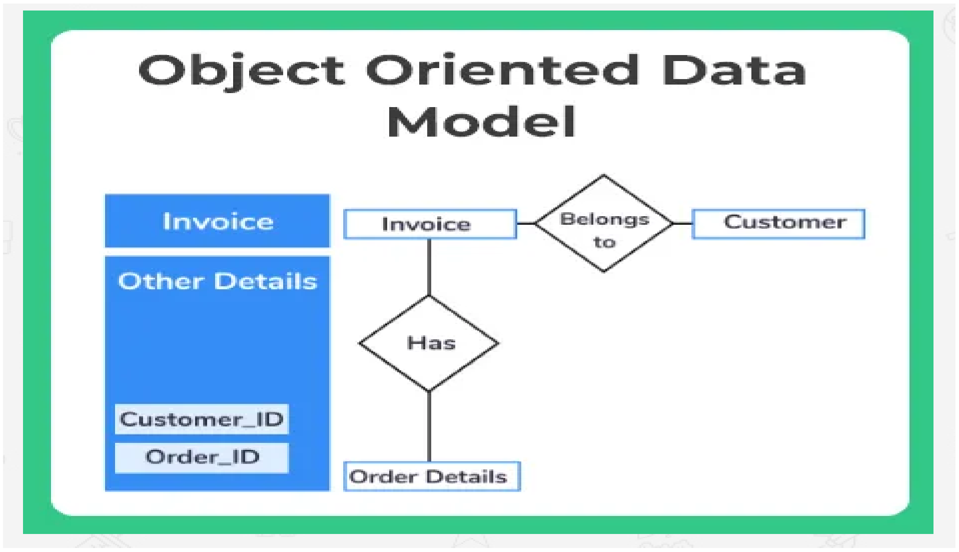
| Roll_No | Name | Age | GPA |
|---|---|---|---|
| 1 | Arya | 21 | 4 |
| 2 | Bran | 19 | 3 |
| 3 | John | 24 | 4.3 |
| 4 | Max | 24 | 1 |

***Object-oriented database model***

The increasing complexity of the code in the modern age, where we try to create real time scenerios. Specially in Artificial Intelligence, Machine learning or Image processing.

We need a database that can also represent the real work scenerios very clearly and hence it is solved by having an object oriented database model.

- 
    - An object is an abstraction of a real-world entity. More or less an object is a real world instance of an entity in Entity Relationship Model
    - Attributes describe the properties of an object, i.e customerID or orderID maybe called as properties of the object or attributes as in entity relationship model
    - As we use classes and inheritance in OOPs concept. The similar can be used here. Classes are a superior set of objects with similar characteristics right?
    - Inheritance is highly used in such, thus also using network or hierarchical model as default inspirations



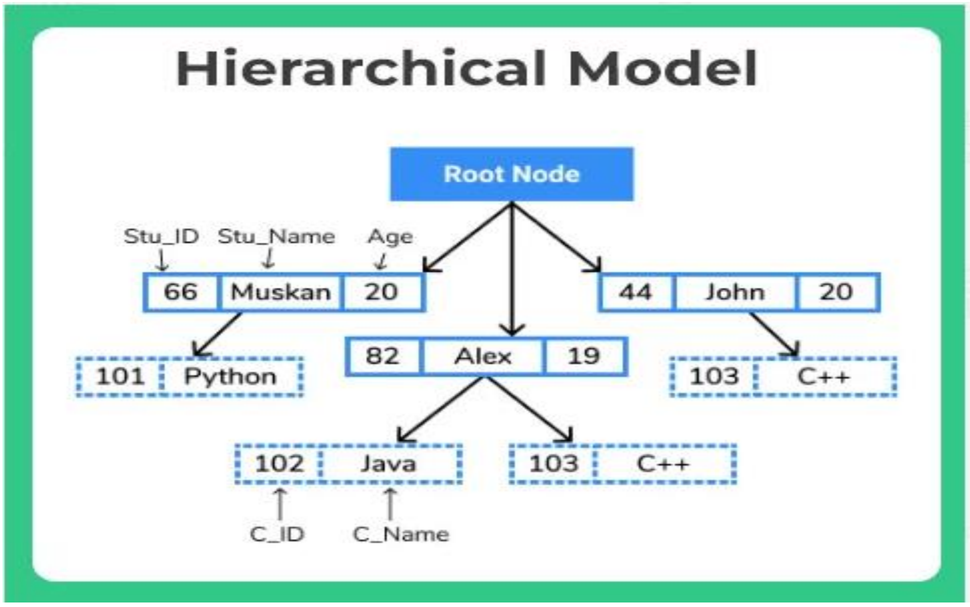
# Hierarchical Model in DBMS

## Hierarchical model in DBMS

- Hierarchical data model is being used from the 1960s onwards where data is organized like a tree structure
- In 1966 IBM introduced an information management system(IMS product) ***which is based on this hierarchical data model but now it is rarely used.***

## Hierarchical model:

The hierarchical model organizes the data into ***a tree structure which consist of a single root node where each record is having a parent record and many child records and expands like a tree***



***Features of the hierarchical model***

- A child node will have only ***one parent node***
- One to many relationship: Hierarchical model is implemented ***based on one to many relationship***.Based on this a***ny parent node should have more than one child nodes i.e one parent for many child.***

***Example for hierarchical model***

- Example of students and courses where a course can be assigned to a single student  where as student can take a number of courses hence ***one to many relationship is observed***
- Here the ***hierarchical data can be represented as relation tables considered as student table*** and one as courts table and the link between these two tables is maintained using a foreign key.

Name:Prathamesh Arvind Jadhav

***Drawbacks of the hierarchical model***

As we have used one to many relationship in data organization, ***child records may be repeated and sometimes we need to maintain empty child records which is a memory wastage.***

# Network Model in DBMS

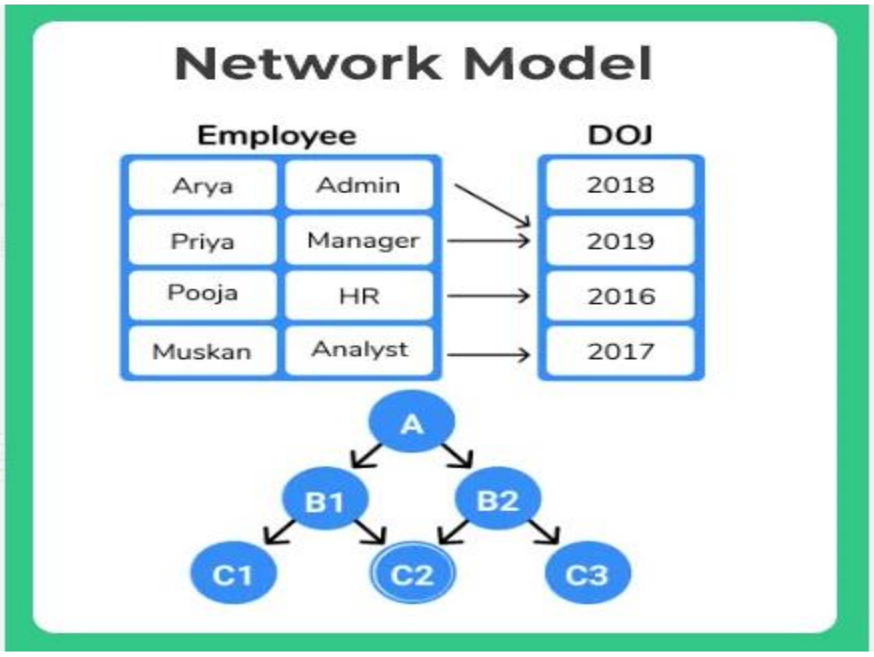
## Network model in DBMS

- In 1970s CODASYL (conference on data system language) introduced a network model which is an extension to the hierarchical model but the difference is that the network model uses many to many rather than one to many.

- In 1970 IBM introduced **IDMS(information data management system)**, a product based on the network data model.

### Network data model

- Data Organisation in the network model is just like a graph rather than a tree, a child record can have any number of parent records
- This was the most widely used model ***before the relational database model was introduced***



***Features of the network data model***

- It was introduced to overcome the drawbacks of the hierarchical model because the ***hierarchical model uses one to many relationship as a result only a few relationships can be established.***
- Many to many relationships: But here in network model It uses many to many relationship where ***the child only is allowed to have more than one parent node as a result large number of relationships can be applied with different data tables and data access becomes easier and faster .***

***Example for or network data model***

- Consider two tables "employee" table and "date of joining" table with the help of this network model ***more than one employee record can be linked to 2 more than and one record in the "date of joining" table***
- For example, as shown in the figure both admin and HR uses the same data row present in the date of joining table i.e [2014] as a result data duplicate is reduced .

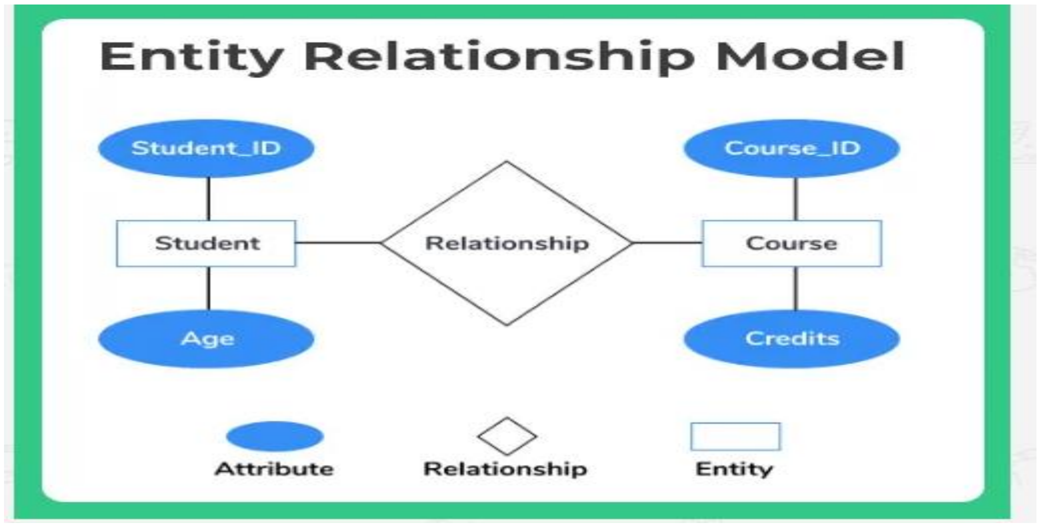
***Advantages of the network model***
As ***more number of relationships or established between data tables it reduces data duplicity saves memory*** improve access time of data.

# Entity Relationship Model (ER model) in DBMS

## Entity-Relationship model (ER model)

- An effective system where data is organized into discrete categories called entities(different database tables)
- ERmodel is an illustration of various entities in business and the relationship between them
- It is built during the ***analysis phase of the system development life cycle***
- The ER model separates the information required in the business i.e it establishes a clear distinction between data that is useful and not useful for business operations

Name:Prathamesh Arvind Jadhav



***Key components of the ER model***

Entity :

It is a set of related information i.e ***nothing but a database table.***

Attributes :

- 
  - It defines ***what type of data an entity is storing*** i.e different columns in a table
  - For example, ***a student entity consists of student name, ID, marks as attributes***
  - An entity may contain any numbers of attribute but ***it must contain at least one attribute .***

Relationships :

- 
  - It is defined as ***an association between entities links between different database tables for data accessing***
  - **Several relationships may exist between the same entity** basically 4 relationships are identified
    - One to one
    - One to many

    - Many to one
    - Many to many
- 
  - These relationships are nothing but ***the links between in a table to another table or set of tables*** so that data can be transferred and used from more than one table .

***<u>ER model benefits</u>***

- Provide a clear picture of the ***scope of information requirement***
- Provides an easily understandable ***pictorial map for database design***
- It offers an effective ***Framework for integrating multiple applications.***

# Relational Model in DBMS

## Relational model in DBMS

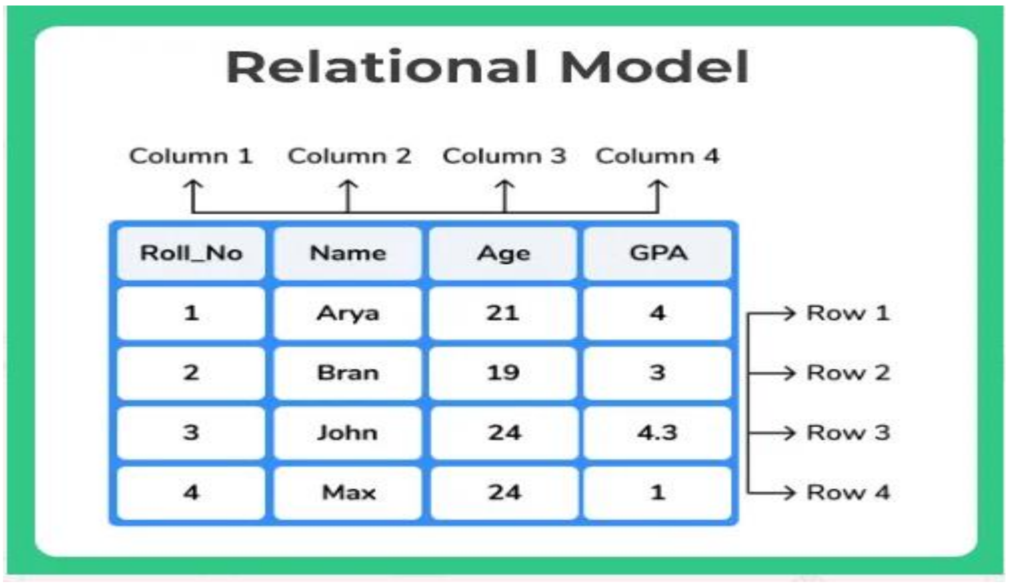
In this article, we will learn about Relational Model in DBMS.

- Relational model was first outlined by EF Codd in 1970 and since then it was the most widely used data model and in fact, the only used database management system today in the world
- The basic idea between this relational data model is simple two-dimensional tables, also called relations which consist of rows and columns.

***<u>Main components of the relational model</u>***

- ***Rows and columns*** of tables
- ***Operations*** on these rows and columns ***manipulating required data***

Name:Prathamesh Arvind Jadhav



| Roll_No | Name | Age | GPA |
|---|---|---|---|
| 1 | Arya | 21 | 4 |
| 2 | Bran | 19 | 3 |
| 3 | John | 24 | 4.3 |
| 4 | Max | 24 | 1 |

***Example:***

Student table consists of different types of data like ***student name, marks, and age, etc*** and all these details are stored in separate columns where each individual student details is stored in a separate row in the table.

***Advantages of the relational model***

- ***Data integrity*** for accuracy and consistency
- ***No data redundancy***
- ***Access control and integrity*** in the form of constraints which enables validation before entering and accessing the data
- Provides ***high security***
- Supports to ***store any type any data types***(numbers, characters, date, images, audios, text files )
- Data can be ***managed and used by several users*** at a time.
- Data can be ***shared across several platforms.***

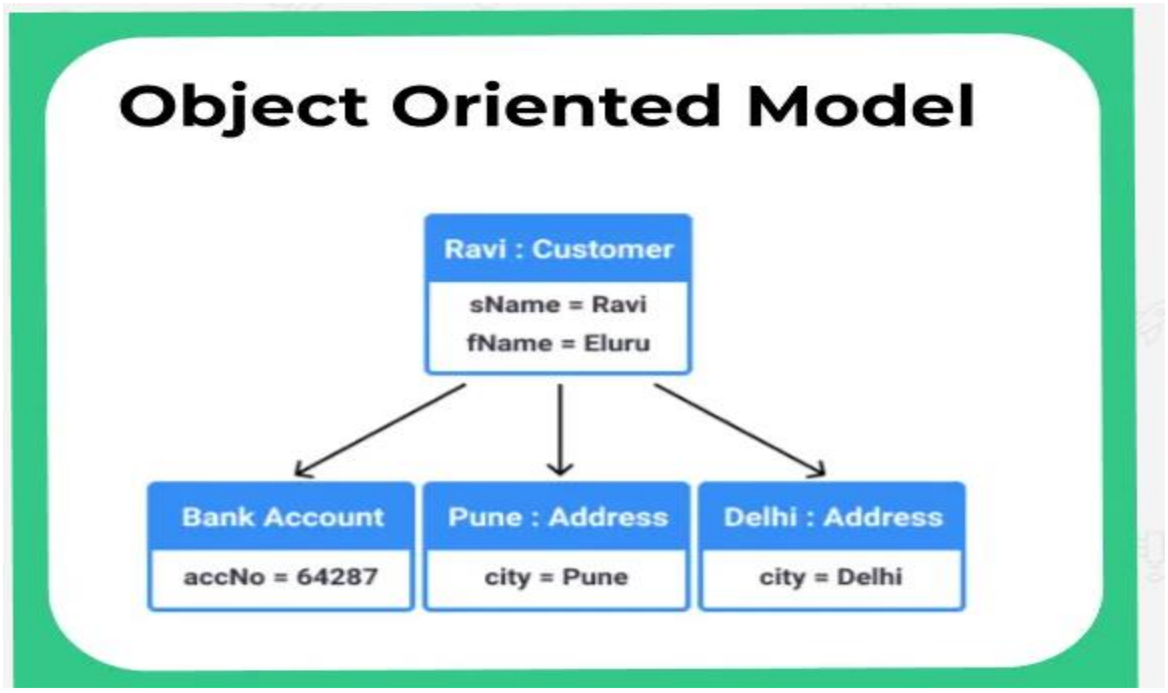
**Note:**Oracle 10g a most popularly used RDBMS is proven to be the fastest database for transaction processing including data warehousing and third-party application on several servers of all sizes.

## Object Oriented Database Model

### Object-oriented database model

In this article, we will learn about the Object-Oriented database model.

- In the increasing complexity of the application speed in the modern age, where we try to create real-time scenarios. Especially in Artificial Intelligence, Machine learning or Image processing, we need a database that can also represent the real work scenarios very clearly and hence it is solved by having an object-oriented database model.
- In contrast to relational database management systems (RDBMSs), where data is stored in tables with rows and columns, an object-oriented database stores complex data and relationships between data directly, without mapping any links to relational rows and columns.



***<u>What object-orientation actually?</u>***

- The main idea in design is to ***maintain separate sets of memories that mean separate memory spaces for each and every individual row*** instead of collectively storing as all the rows in a single space
- As a result data ***independence is achieved*** so that all operations and transactions done in one data are independent and unaffected with other data today's minimum as possible .

***<u>Examples</u>***

Some OODBMSs are designed to collborate with other programming languages (such as Java, Python, Perl, Delphi, Ruby, C#, Visual Basic .NET, C++, etc). Others have their own language of choice

- 
    - Versant Object Database by Actian
    - Objectivity/DB by Objectivity
    - ObjectStore by Ignite Technologies
    - Caché by Intersystems
    - ZODB by Zope

***Advantages of OO design***

- Data Hiding: Giving a***ccess to only required data*** at different levels of users
- Inheritance: data ***reusability***
- Data security and integrity: Providing ***different levels of access to different levels of users*** using access modifiers, validations, authentication
- Polymorphism: Using the ***same object in different ways***
- Encapsulation: Combining different data from ***different units into a consolidated one***

Some ***DBMS are a hybrid of OODBMS and RDBMS***, and are therefore referred to as ***object-relational databases (ORD)***