

## DBMS Notes 1:

In our daily life we use multiple applications and websites which showcase so many contents. But where are these contents stored?

Let us take an example, open the Amazon mobile application and add a cute dress to the shopping cart. Now close the app and open the Amazon website either through mobile or laptop and login to your account same as that in the mobile application. Check the shopping cart and you will find that the cute dress that you added from the application is also visible here. How does that happen?

This scenario and many other scenarios like this are possible because of the existence of a database.

A database is nothing but a repository or storage area where data are stored and can be later retrieved from, updated and deleted. Now the storage of data would not be random. It would be stored in such a way that retrieving it or performing any other operation on it would become much easier.

**Definition:** A database is a structured collection of data that is organized in a way that makes it easy to access and manage. Databases are used to store all sorts of data, including customer information, product information, financial data, and more.

**OR**

A database can be defined as a collection of interrelated data stored together to serve multiple applications. The data is stored in such a manner that it is independent of programs which use the data and has a common and controlled to add and modify data.

Before we move further into the database concepts, we first need to know why we use database and what are their needs.

The file system in your computer is also a tool to save data. So if we already have a tool that helps us to save our data why do we need databases?

There are a few reasons:

1. Data Sharing: Databases make data sharing between applications much easier.
2. Data Integrity: Databases helps to make sure data is consistent and accurate.
3. Data Security: Databases can be used to secure the stored data from unauthorized access.

In a typical file processing system, there are many limitations including **data redundancy** or duplication of data, data inconsistency etc.

To solve these problems while using a database we have a centralized software that manages the data and minimize these problems. This software is known as a **Database Management System** or **DBMS**.

~~✓~~ **Definition:** A **DBMS** refers to a software that is responsible for storing, maintaining and utilizing database.

Think of the Database as a cycle and the DBMS as the pedals of the cycle, without the pedals the cycle would be there and it might run also, but the question is "Kab tak?".

The DBMS provides various functionalities to the user for operating on the database which we would discuss as we move further into the syllabus.

There are two main types of databases:

1. Relational Database
2. Non-Relational Database

→ MongoDB, Firebase

For our discussion we would limit ourselves to Relational databases as understanding relational database concepts would give a distinct idea how databases work.

**NOTE:** Non-Relational databases are gaining popularity now and are in trend even though most of the web is built upon Relational databases.

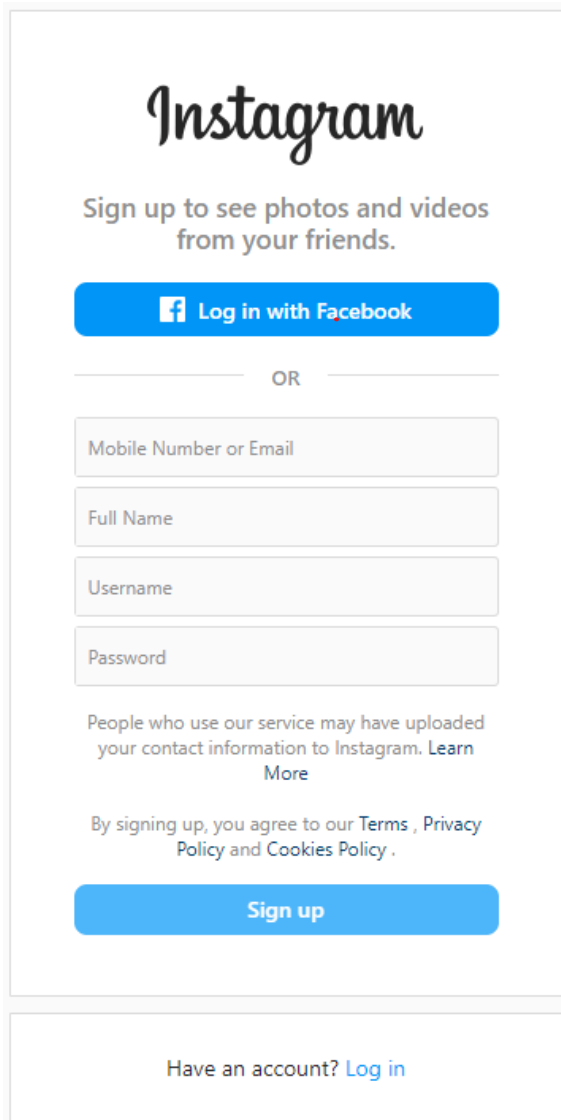
Relational Databases:

In a relational database, the data is organized into tables, rows and columns. These tables are called relations.

A row in the table represents the relationship among a set of values.


Let us take an example.

Suppose while registering into Instagram you fill out a registration form, something like this:



Instagram

Sign up to see photos and videos from your friends.

 Log in with Facebook

OR

Mobile Number or Email

Full Name

Username

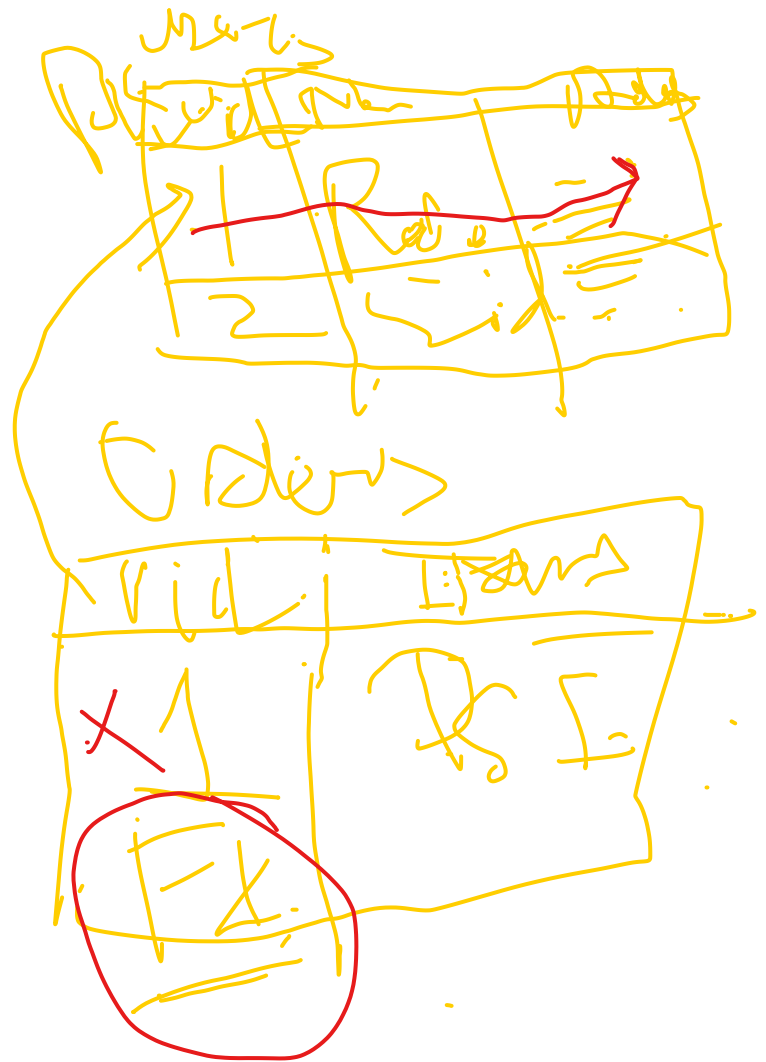
Password

People who use our service may have uploaded your contact information to Instagram. [Learn More](#)

By signing up, you agree to our [Terms](#), [Privacy Policy](#) and [Cookies Policy](#).

Sign up

Have an account? [Log in](#)



Now the details you enter in this form is stored in the database in the form of tables. For example, we have a table with a name **"Users"** which stores these data. Then the table would look something like this:

Mobile Number	Name	Username	Password
999999999	Rodosee Roy	Rodosee@1231	1234

Now when another person fills up the form using his/her device then the information of that person is also stored in the same **"Users"** table. It would look like this:

Mobile Number	Name	Username	Password
999999999	Rodosee Roy	Rodosee@1231	1234
888888888	Ram Sharma	Ram@000	0000

Your information is not lost or changed but a new information is added into the same table. This is how a relational table works.

Some Keywords in relational database:

1. **Tuple:** A tuple is a row in a table. It represents a single record. For example, a tuple in the Users table example, represents each user which might be you or any other person.
2. **Attributes:** An attribute is a column in a table. It represents a single feature or attribute of the entire table. For example, in the Users table the Name is attribute that each entry has.

3. **Domain:**

Suppose you are applying for a driving license for which you have to be 18 years and above. Now when you fill the form the age you write should be 18 or more than 18. So, any age less than 18 would not be accepted. So that is the domain of the Age attribute in that particular relation.

It is the set of all possible values for an attribute.

4. **Degree:** The number of attributes in a relation.

5. **Cardinality:** The number of rows in a relation.

6. **Key:**

A key is one or more attributes in a relation that uniquely identify a particular tuple.

For example, in the Users table, you would notice that the username is unique for every user, if search for information for the username "Rodosee@1231" then I would be able to get all your information and the other person's information would be left out.

This signifies that the username attribute in the Users table uniquely identifies each and every row.

There are various types of Keys in the relational database:

- a. **Primary key:** A primary key is a set of one or more attributes that can uniquely identify tuples within the relation.
- b. **Candidate key:** All attribute combinations inside a relation that can serve as primary key are Candidate keys as they are candidates for the primary key position.

- c. Alternate Key: A candidate key that is not the primary key is called an Alternate key.
- d. Foreign key: A non-key attribute whose values are derived from the primary key of some other relation, is known as a foreign key.

Relational Algebra: Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries.

There are a few operations in relational algebra:

1. Selection ( $\sigma$ ): Selection in relational algebra returns those tuples (records) in a relation that fulfil a condition.

Example:  $\sigma_{\text{Name} = \text{"Rodosee"}}(\text{Users})$ , this selects tuples from Users table where Name = Rodosee.

2. Projection ( $\Pi$ ): Projection in relational algebra returns those columns in a relation that given in the attribute list.

Example:  $\Pi_{\text{Name}, \text{Username}}(\text{Users})$ , this shows the columns named as Name and Username.

3. Union ( $\cup$ ): The union operator is used to combine two or more tables. In the union operation, duplicate records will be automatically removed from the resultant table.

4. Cartesian Product ( $\times$ ): Cartesian product returns a number of rows equal to number of rows in the first table multiply by number of rows in the second table. At the same time, number of columns equal to number of columns in the first table added by number of columns in the second table