DBMS/RDBMS

Relevel
by Unacademy

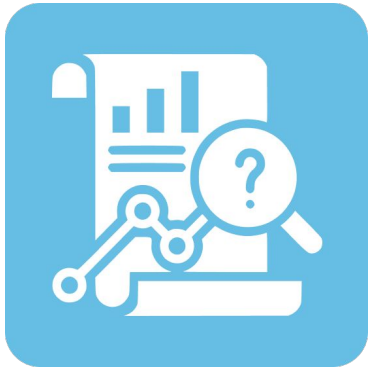
Overview

Database management system is a software which is used to manage the database. For example: MySQL, Oracle, and PostgreSQL.

DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, and creating a table in the database. Additionally, it provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

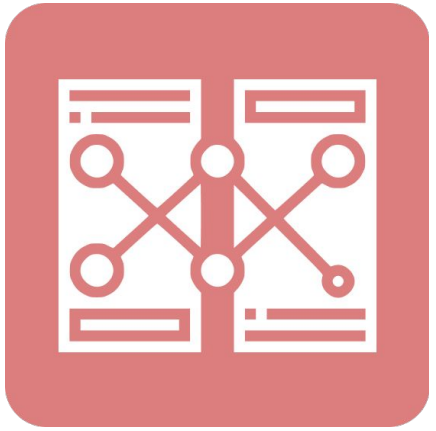
What is Data?

Word 'Data' is originated from the word 'datum' that means 'single piece of information.' Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media or bytes. It can be stored in pieces of paper or electronic memory.



What is Database?

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. We can organize data into tables, rows, columns, and index it to make it easier to find relevant information.
For example: The college Database organizes the data about the admin, staff, students and faculty.



The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data.
SQL or **Structured Query Language** is used to operate on the data stored in a database.

DBMS tasks

Data Definition: It is used for creation, modification, and removal of definition that defines the organization of data in the database.

Data Updation: It is used for the insertion, modification, and deletion of the actual data in the database.

Data Retrieval: It is used to retrieve the data from the database which can be used by applications for various purposes.

User Administration: It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

DBMS Attributes

- It can provide a clear and logical view of the process that manipulates data.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.

Advantages of DBMS

- Controls database redundancy
- Data sharing
- Easily Maintenance
- Reduce time

Disadvantages of DBMS

- Cost of Hardware and Software
- Large Size and memory
- Complexity
- Higher impact of failure

Evolution of Databases

The database has completed more than 50 years of journey of its evolution.

File-Based - introduced in 1968, data was maintained in a flat file.

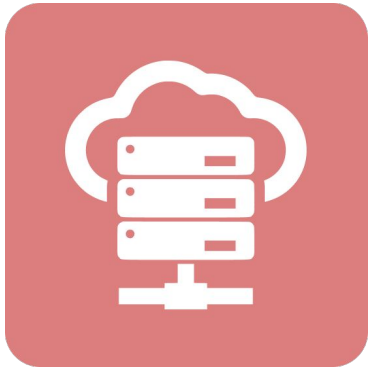
Hierarchical Data Model - 1968-1980 was the era of the Hierarchical Database, where files were related in a parent/child manner.

Network Data Model - Developed in 1960s, In this model, files are related as owners and members.

Relational Database - Introduced in 1970, used till date, it consists of instance(table) and schema.

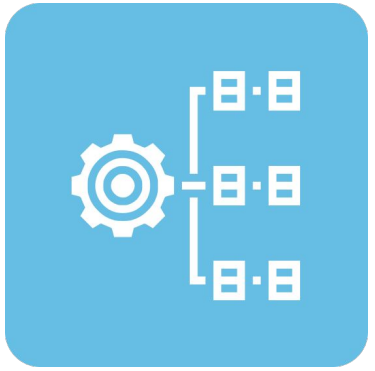
Types of Databases

During the era of the relational database, many more models are introduced like object-oriented model, Cloud based and NoSQL.



Cloud database: Cloud database facilitates you to store, manage, and retrieve their structured, unstructured data via a cloud platform. This data is accessible over the Internet. Example - AWS

Object-Oriented Databases: The object-oriented databases contain data in the form of object(real-world entity) and classes.



OOPS Properties:

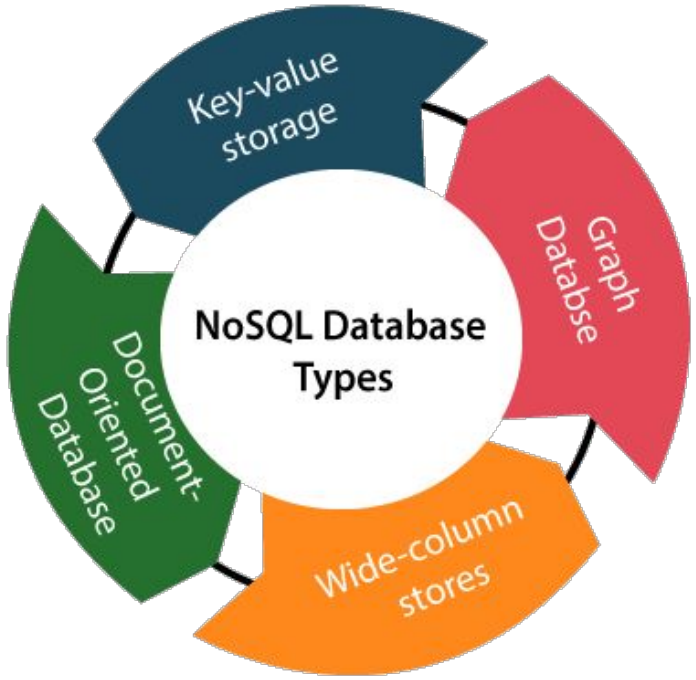
- Objects
- Classes
- Inheritance
- Polymorphism
- Encapsulation

Types of Databases

NoSQL: Stands for "not only SQL.

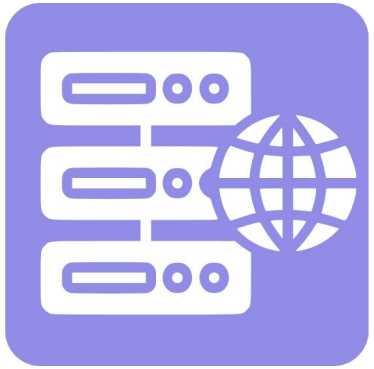
Example:

- MongoDB, CouchDB, Cloudant **(Document-based)**
- Memcached, Redis, Coherence **(key-value store)**
- HBase, Big Table, Accumulo **(Tabular)**
- **(Graph- based)**



Centralized and Distributed Database

Centralized Database - It is the type of database that stores data at a centralized database system. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.



Advantages:

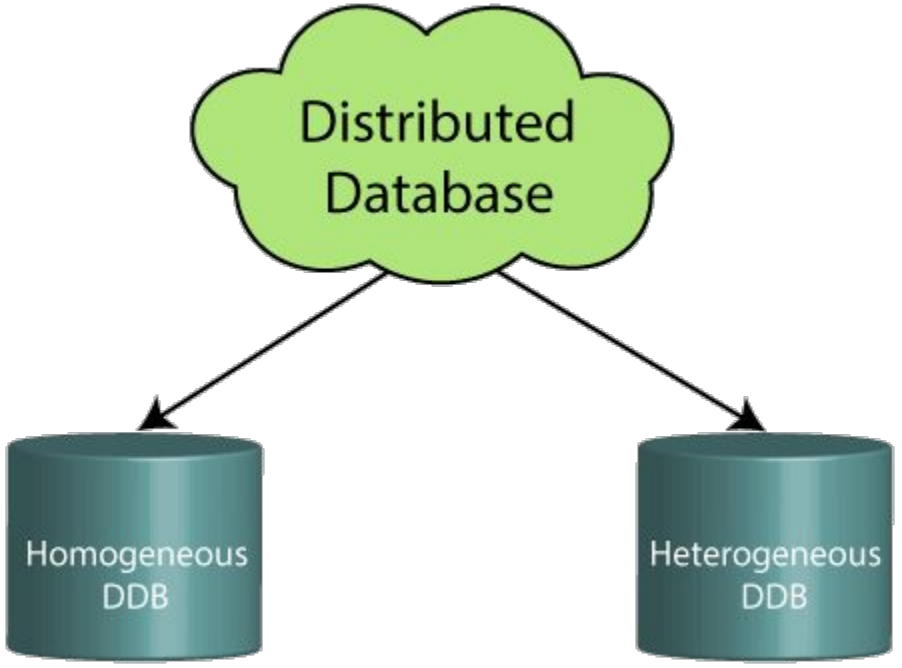
- It has decreased the risk of data management.
- Data consistency is maintained.
- It establishes data standards.
- It is less costly

Disadvantages:

- Large Size
- Difficult to update

Centralized and Distributed Database

Distributed Database - In distributed systems, data is distributed among different database systems of an organization. Examples of the Distributed database are Apache Cassandra, HBase, Ignite, etc.



Centralized and Distributed Database

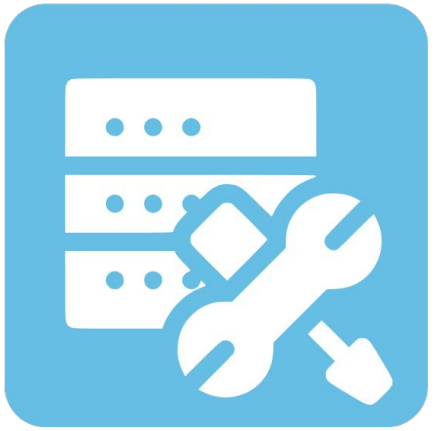
Distributed Database -

Advantages:

- Modular development is possible.
- One server failure will not affect the entire data set.

Disadvantages:

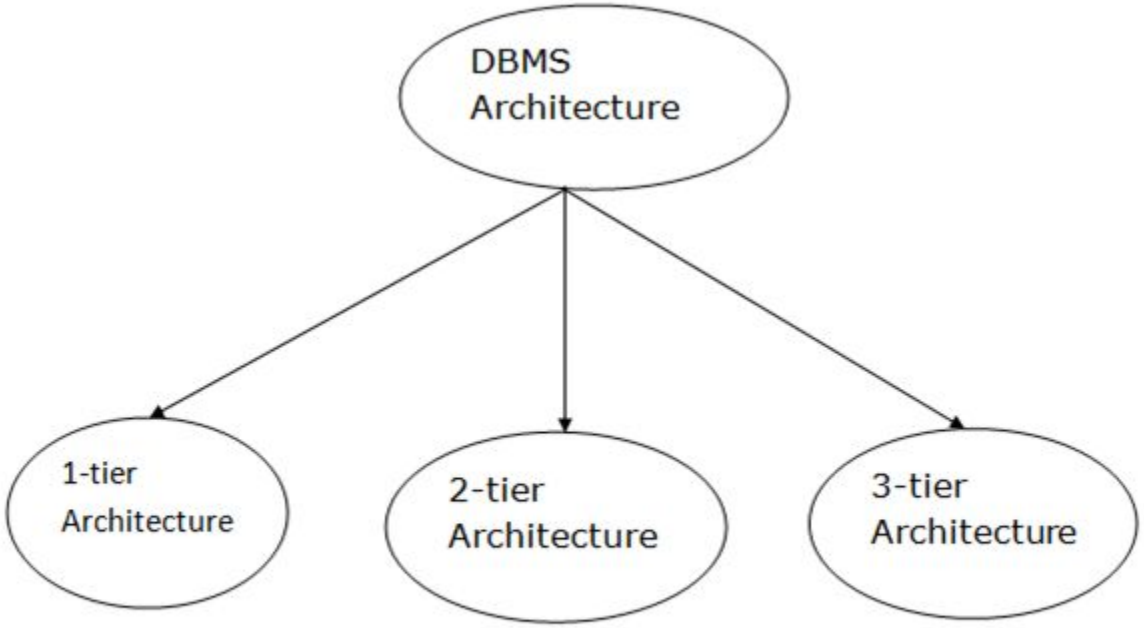
- Complex nature



DBMS Architecture

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture



Types of DBMS Architecture

Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

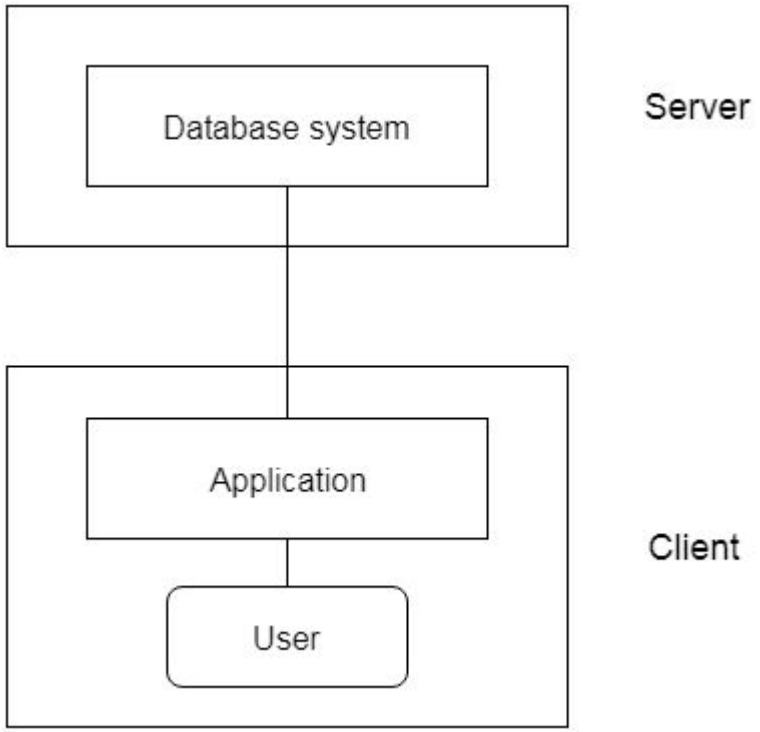
1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

Types of DBMS Architecture

2-Tier Architecture

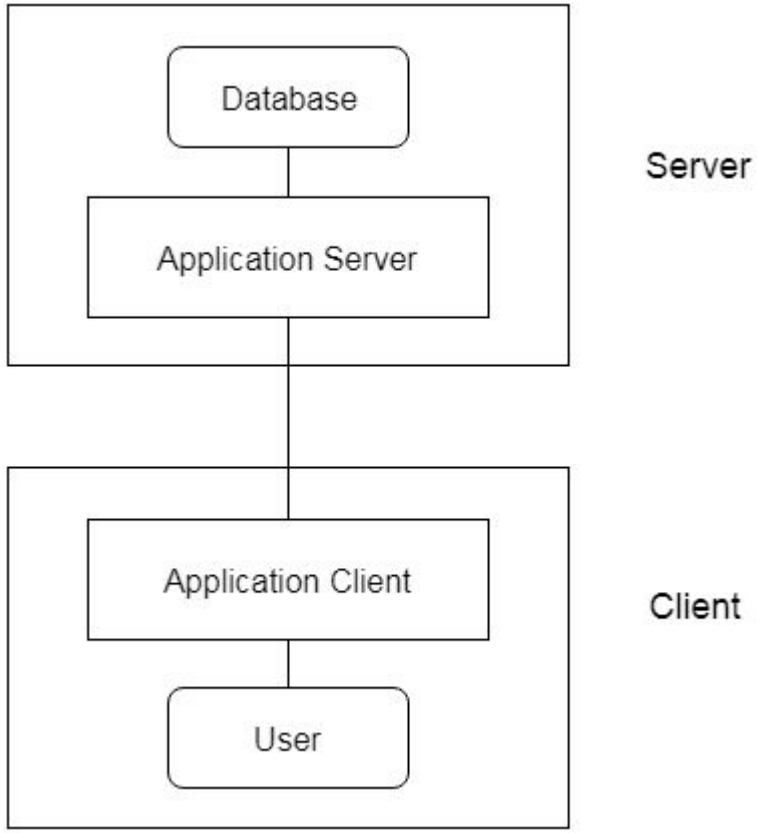
- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.



Types of DBMS Architecture

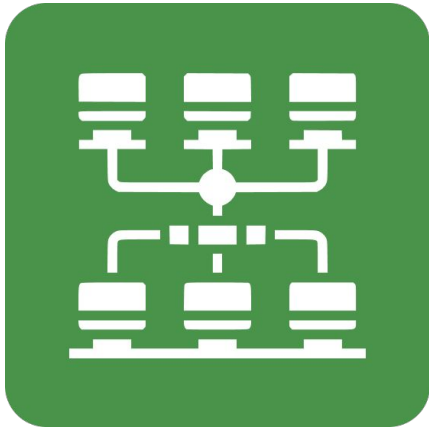
3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.

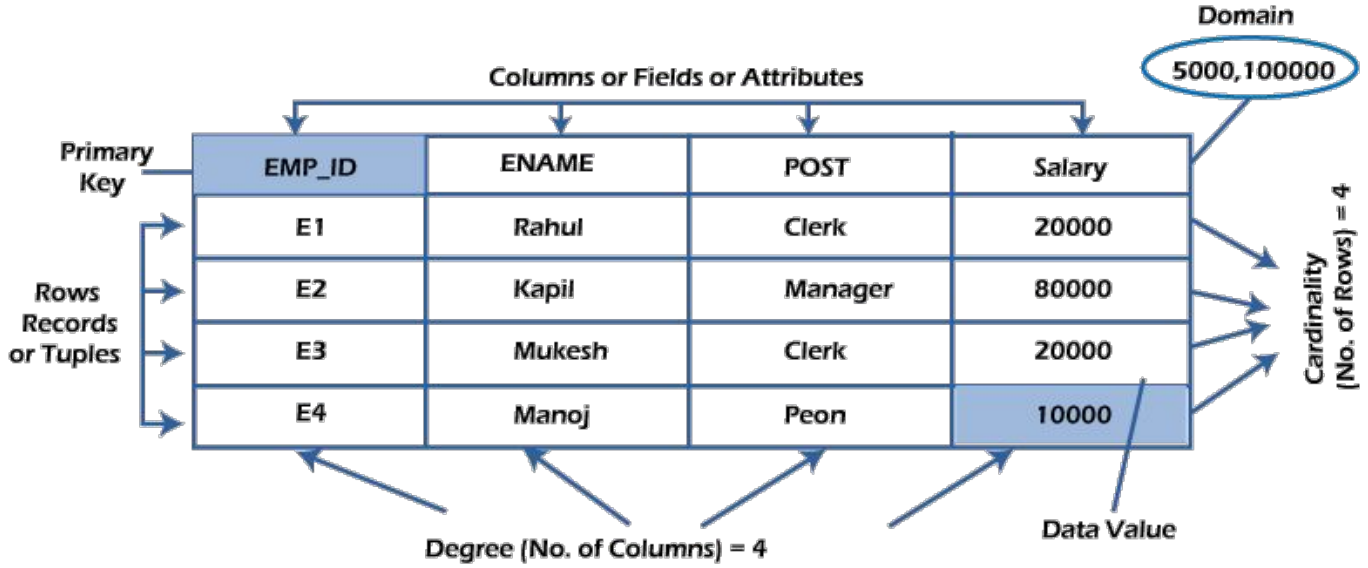


What is RDBMS?

- RDBMS stands for **Relational Database Management System**.
- All modern database management systems like **SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access** are based on RDBMS.
- Additionally, it is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd(1970-1972)
- A relational database is the most commonly used database. It contains several tables, and each table has its primary key
- Data is represented in terms of tuples (rows) in RDBMS



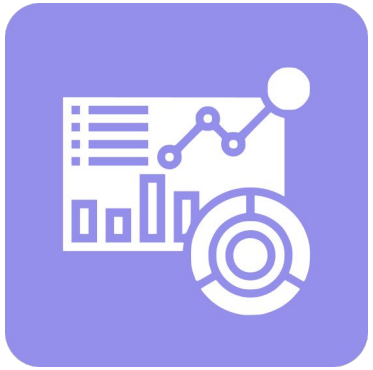
RDBMS Terminology



RDBMS Terminology

1. Table/Relation

A table is the simplest example of data stored in RDBMS.
Everything in a relational database is stored in the form of relations. The RDBMS database uses tables to store data. A table is a collection of related data entries and contains rows and columns to store data. Each table represents some real-world objects such as person, place, or event about which information is collected. The organized collection of data into a relational table is known as the logical view of the database.



Properties of a Relation:

- Each relation has a unique name by which it is identified in the database.
- Relation does not contain duplicate tuples.
- The tuples of a relation have no specific order.
- All attributes in a relation are atomic, i.e., each cell of a relation contains exactly one value.

RDBMS Terminology

1. Table/Relation

Student Table Example:

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech
2	Aryan	20	C.A
3	Mahesh	21	BCA
4	Ratan	22	MCA
5	Vimal	26	BSC

RDBMS Terminology

2. Row/Record

A row of a table is also called a record or tuple. It contains the specific information of each entry in the table. It is a horizontal entity in the table. For example, The above table contains 5 records.

Properties of a row:

- No two tuples are identical to each other in all their entries.
- All tuples of the relation have the same format and the same number of entries.
- The order of the tuple is irrelevant. They are identified by their content, not by their position.

One record/row Example:

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech

RDBMS Terminology

3. Column/attribute

A column is a vertical entity in the table which contains all information associated with a specific field in a table. For example, "name" is a column in the above table which contains all information about a student's name.

Properties of an Attribute:

- Every attribute of a relation must have a name.
- Null values are permitted for the attributes.
- Default values can be specified for an attribute automatically inserted if no other value is specified for an attribute.
- Attributes that uniquely identify each tuple of a relation are the primary key.

One Column/attribute Example:

Name
Ajeet
Aryan
Mahesh
Ratan
Vimal

RDBMS Terminology

4. Data item/Cell

The smallest unit of data in the table is the individual data item. It is stored at the intersection of tuples and attributes.

Properties of data items:

- Data items are atomic.
- The data items for an attribute should be drawn from the same domain.

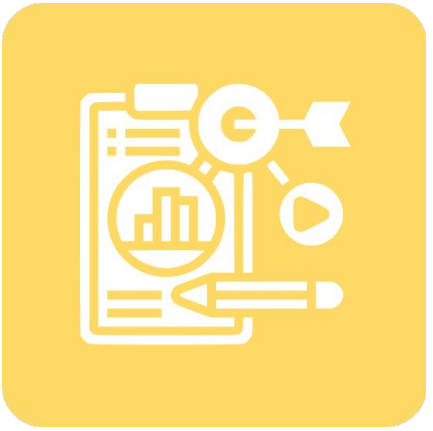
Data item example in the student table consisting of Ajeet, 24 and Btech

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech

RDBMS Terminology

Some other important terminologies of RDBMS include:

1. Degree
2. Cardinality
3. Domain
4. Null values



RDBMS Terminology

1. Degree:

The total number of attributes that comprise a relation is known as the degree of the table.

For example, the student table has 4 attributes, and its degree is 4.

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech
2	Aryan	20	C.A
3	Mahesh	21	BCA
4	Ratan	22	MCA
5	Vimal	26	BSC

RDBMS Terminology

2. Cardinality

The total number of tuples at any one time in a relation is known as the table's cardinality. The relation whose cardinality is 0 is called an empty table.

For example, the student table has 5 rows, and its cardinality is 5.

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech
2	Aryan	20	C.A
3	Mahesh	21	BCA
4	Ratan	22	MCA
5	Vimal	26	BSC

RDBMS Terminology

3. Domain

The domain refers to the possible values each attribute can contain. It can be specified using standard data types such as integers, floating numbers, etc.
For example, An attribute entitled Marital_Status may be limited to married or unmarried values.

4. Null values

The NULL value of the table specifies that the field has been left blank during record creation. It is different from the value filled with zero or a field that contains space.

ID	Name	AGE	COURSE
1	Ajeet	NA	B.Tech

Data Integrity

There are the following categories of data integrity exist with each RDBMS:

Entity integrity: It specifies that there should be no duplicate rows in a table.

Domain integrity: It enforces valid entries for a given column by restricting the type, the format, or the range of values.

Referential integrity specifies that rows cannot be deleted, which are used by other records.

User-defined integrity: It enforces some specific business rules defined by users. These rules are different from the entity, domain, or referential integrity.

DBMS and RDBMS difference

RDBMS is an extension of DBMS, with software products available compatible for both DBMS and RDBMS.

No.	DBMS	RDBMS
1.	DBMS applications store data as file.	RDBMS applications store data in a tabular form.
2.	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3.	Normalization is not present in DBMS.	Normalization is present in RDBMS.
4.	DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrity constraint for the purpose of ACID (Atomicity, Consistency, Isolation and Durability) property.
5.	DBMS uses file system to store data, so there will be no relation between the tables.	in RDBMS, data values are stored in the form of tables, so a relationship between these data values will be stored in the form of a table as well.
6.	DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
7.	DBMS does not support distributed database.	RDBMS supports distributed database.
8.	DBMS is meant to be for small organization and deal with small data. it supports single user.	RDBMS is designed to handle large amount of data. it supports multiple users.
9.	Examples of DBMS are file systems, xml etc.	Example of RDBMS are mysql, postgre, sql server, oracle.

Assignment

Difficulty level - Easy(E), medium(M), Hard(H)

1. Object oriented model came into existence in relational DBMS era. Elaborate OOPS properties. **(M)**
2. For the below mentioned student table, what will be its degree and cardinality? **(M)**

ID	Name	AGE	COURSE
1	Ajeet	24	B.Tech
2	Aryan	20	C.A
3	Mahesh	21	BCA
4	Ratan	22	MCA
5	Vimal	26	BSC

3. _____ & _____ are the other two synonyms of column in RDBMS. **(M)**
4. In RDBMS, a row or set of rows is also known _____ **(E)**
5. if every row in a table represents relationships for a unique entity, the table should have one column or a set of columns that provides a unique identifier for the rows of the table. This is ensured by which type of integrity? **(M)**

Thank You