# DBMS.

__/__/____

Database Management System in short refers to the technology of storing and retrieving user data with utmost efficiency along with appropriate security measures.

## Characteristics of DBMS

1) It can provide a clear and logical view of the process that manipulates data.

2) DBMS contains automatic backup and recovery procedures.

3) It can reduce the complex relationship between data.

4) It is used to support manipulation and storing of data.

5) It is used to provide security of data.

6) It contains ACID properties.

* ACID Properties

1) Atomicity - The entire transaction takes place at once or doesn't happen at all.

2) Consistency - The database must be consistent before and after the transaction.

3) Isolation - Multiple transactions offer independently without interference.

4) Durability - The changes of a successful transaction occurs even if the system failure occurs.

* Advantages of DBMS

- Controls database redundancy
- Data sharing.
- Easy ~~Maintainance~~ Maintenance
- Reduce time.
- Backup
- Multiple User Interface.

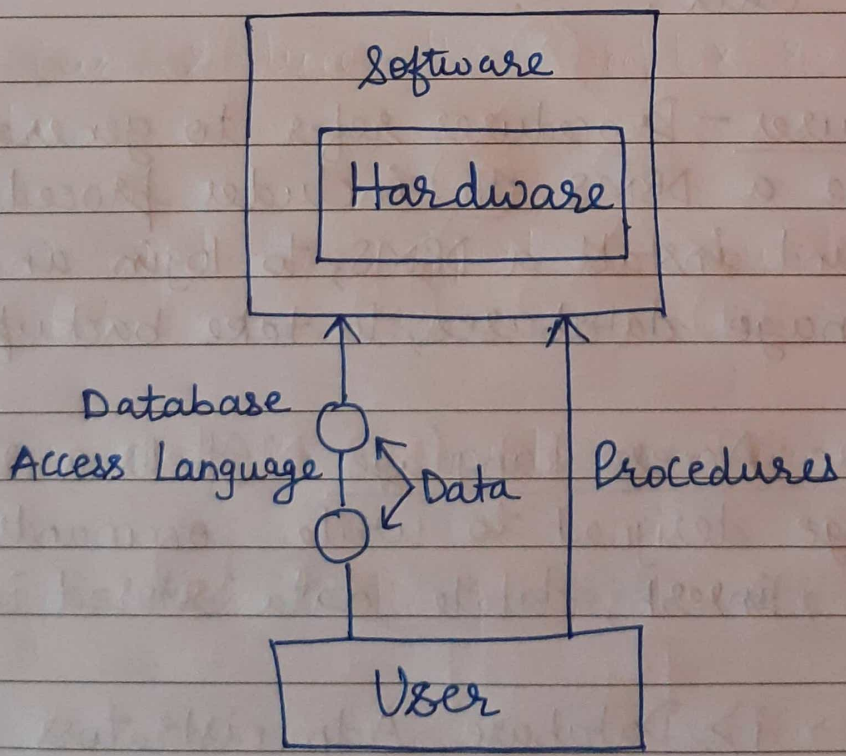
__/__/____

## * Disadvantages of DBMS

- Cost of hardware and software
- Size requirement is high
- Complexity.
- Higher impact of failure.

## * Components of DBMS

⇉




1) Hardware - Hard disks, computer, I/O channels for data, or any other physical component.

2) Software - The DBMS Software is more like a wrapper around the physical database, which provides us an easy-to-use interface to store, access and update the data.

It is capable of understanding the Database Access Language and interpret into actual database commands.

3) Data - Data is the resource.
In typical database, the user saved data is present and meta data is stored. Metadata is data about the data.

4) Procedures - Procedures refer to general instructions to use a DBMS. It includes procedures to setup and install a DBMS, to login and logout, to manage databases, to take backups, etc.

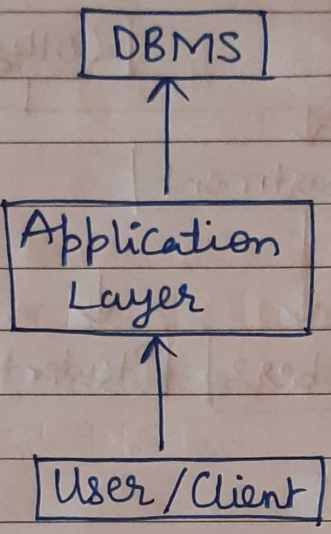
5) Database Access Language - It is a simple language designed to write commands to access, updata, insert, delete data stored in database.

6) Users → i) Database Administrators
ii) Application Programmer or Software Developer
iii) End User.

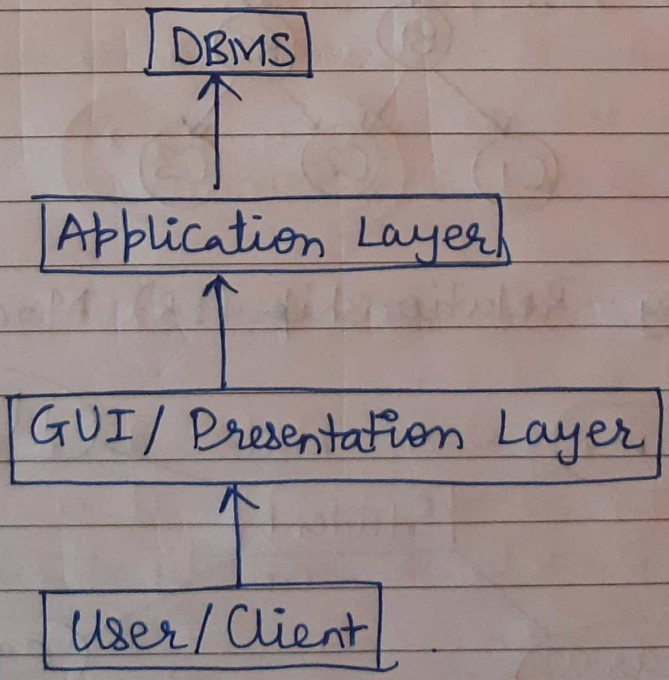
__/__/____

## * DBMS Architecture

1) 2-tier Architecture



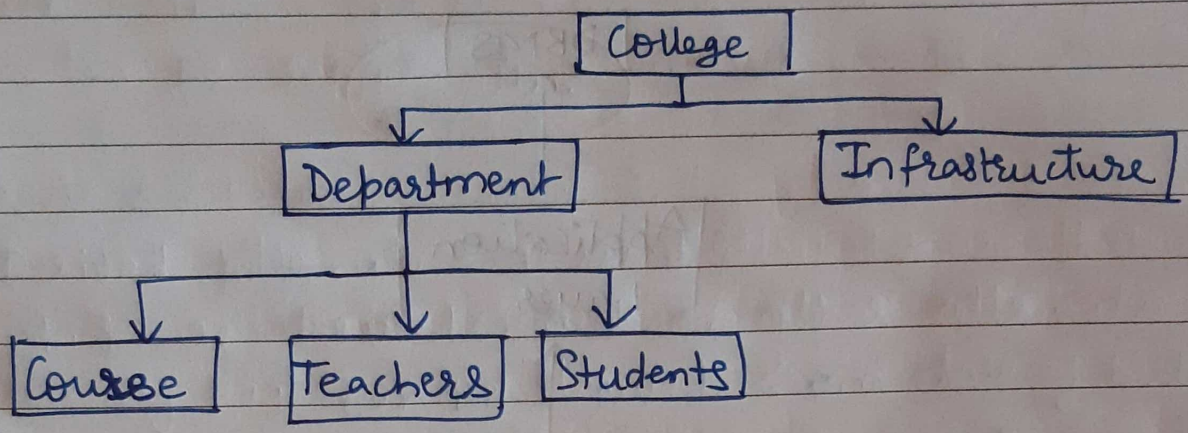

2) 3-tier Architecture



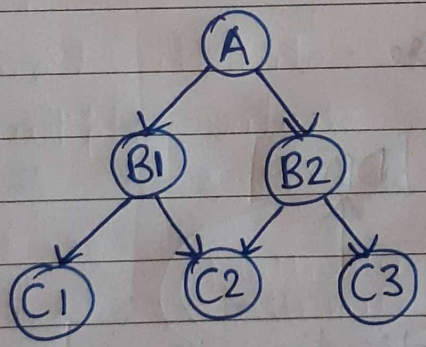
__/__/____

# * DBMS Database Models.

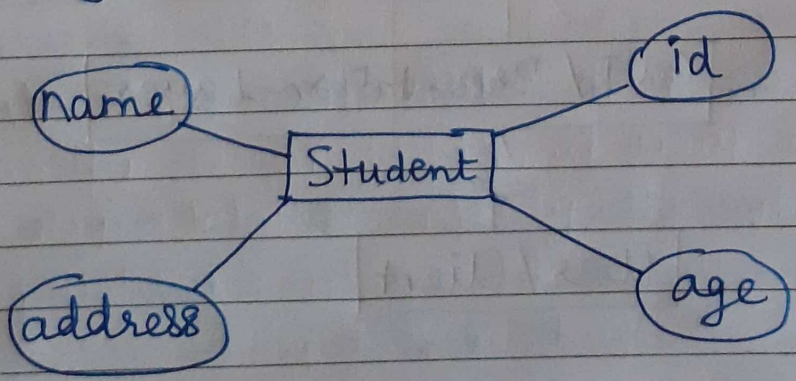
## 1> Hierarchical Model



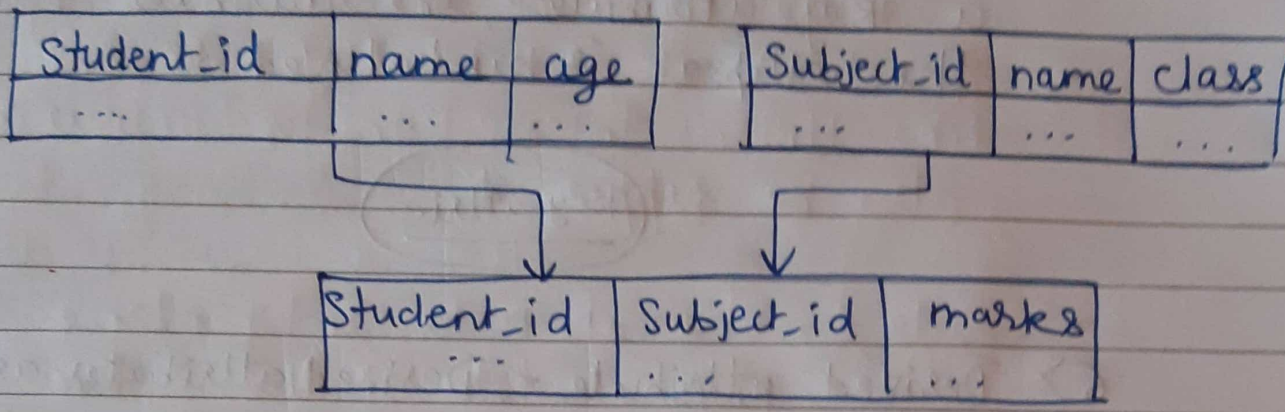

## 2> Network Model




## 3> Entity - Relationship (ER) Model

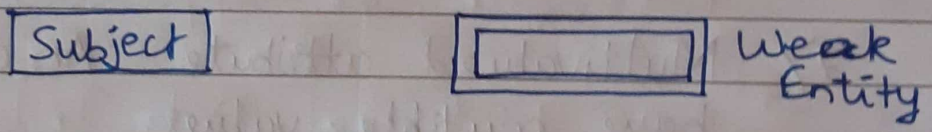


4) Relational Model.

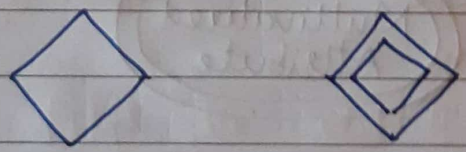



* Components of ER diagram

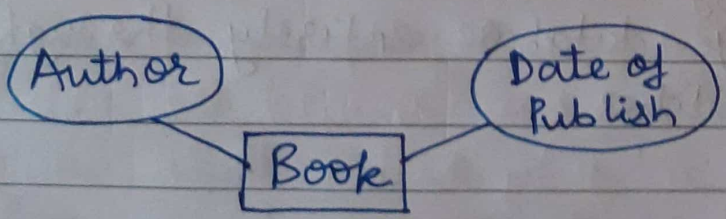
1) Entity - Simple rectangular box



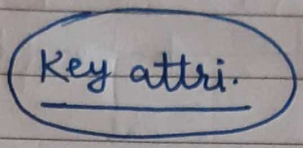

2) Relationship between entities - Rhombus.



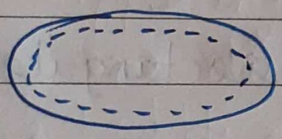

4) Attributes for any Entity - Ellipse.



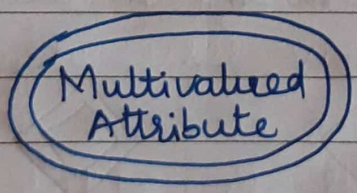
5> Key attribute for any entity - To represent a key attribute, the attribute name inside the ellipse is underlined.




6> Derived attribute - Derived attributes are those which are derived based on other attributes For eg. age can be derived from date of birth.



7> Multivalued attribute - An attribute which can have multiple values.




* Codd's Rule for ~~RDMS~~ RDBMS.

-> Rule Zero - It states that for a system to qualify as an RDMS, it must be able to manage database entirely through the relational capabilities.

__/__/____

→ Rule 1 : Information Rule.
All information is to be represented as stored data in cells of tables.

→ Rule 2 : ~~Gur~~ Guaranteed Access
Each unique piece of data (atomic value) should be accessible by : Table Name + Primary Key (Row) + Attribute (column).

→ Rule 3 : Systematic Treatment of NULL.
NULL can mean missing data, not applicable or no value. It should be handled consistently. Primary key must not be null. Expression on Null must give NULL.

→ Rule 4 : Active Online Catalog.
Database dictionary (catalog) is the structure description of the complete Database and it must be stored online.

→ Rule 5 : Powerfull and Well-Structured Language.
One well structured language must be there to provide all manners of access to the data stored in the database.

→ Rule 6 : View Updation Rule.
All the view that are theoretically uptable should be updatable by the system as well.

→ Rule 7 : Relational Level Operation.
There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

→ Rule 8 : Physical Data Independence.
~~The physical data of the system~~
The physical storage of data should not matter to the system.
Eg. If some supporting file is renamed or moved from one disk to another, it should not effect the application.

→ Rule 9 : Logical Data Independence.
If there is change in the logical structure of database, the user view of data should not change.
Eg. If a table is split into two, a new view should give result as join of the two tables.
This rule is most difficult to satisfy.

→ Rule 10 : Integrity Independence
The database should be able to enforce its own integrity rather than using other programs.
This also makes RDBMS independent of front-end.

__/__/____

→ ~~Rule~~ Rule 11 : Distribution Independence.
A database should work properly regardless of its distribution across a network. Even if the database is geographically distributed, in pieces, the end user should get an impression that it is stored at the same place.

→ Rule 12 : Non Subversion Rule.
If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change the data. This can be achieved by some sort of locking or encryption.

* Database Keys

⇒ A key can be a single attribute or group of attributes, where the combination may act as a key.
They are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.

1) Super Key
Super key is defined as a set of attributes within a table that can uniquely identify each record within a table. ~~SuperSet~~

Super Key is a superset of Candidate Key.

2) Candidate Key.

Candidate Keys are defined as the minimal set of fields which can uniquely identify each record in a table. It can act as a Primary Key for a table to uniquely identify each record in that table. There can be more than one candidate key.

It can never be NULL or empty.

3) Primary Key.

Primary key is candidate key that is most appropriate to become the main key for any table.

4) Composite Key.

Key that consists of two or more attributes that uniquely identify any record in a table is called Composite Key.

5) Secondary Or Alternative Key

The candidate key which are not selected as primary key are secondary / alternative keys.

6) Non- Prime Attributes.
The attributes other than Primary Key attributes.

* Normalization
It is a systematic approach of decomposing tables to eliminate data redundancy.

1) First Normal Form (1NF)
The Table should have single valued attributes.

| | | |
|---|---|---|
| 66 | Sanjay | P1,P2 |
| 73 | Jay | P3 |

| | | |
|---|---|---|
| 66 | Sanjay | P1 |
| 66 | Sanjay | P2 |
| 73 | Jay | P3 |

Normalized to INF.

2) Second Normal Form (2NF)
- It must be in 1NF
- It should be free from partial dependencies

| Student_id | Subject_id | marks | Teacher |
|---|---|---|---|
| 191 | 13A | 70 | Java |
| 191 | 15D | 75 | C++ |
| 191 | 13A | 80 | Java. |

Primary key is the combination of student_id and subject_id columns.
But the column teacher entirely depends on

Subject_id column and independent of student_id.

| Student_id | Subject_id | Marks |
|---|---|---|
| 191 | 13A | 70 |
| 191 | 15D | 75 |
| 191 | 13A | 80 |

| Subject_id | Teacher |
|---|---|
| 13A | Java |
| 15D | C++ |

3) Third Normal Form (3NF)

- It should be in 2NF.
- There is no transitive dependency.

Transitive Dependency

When non-prime attribute depends on another non-prime attribute

If a table has transitive dependency, decompose it into 2 tables.

4) Boyce Codd Normal Form (BCNF)

- It should be in 3NF..
- For every functional dependency, A → B, A must be a superkey i.e. in any functional dependency, L.H.S. attribute must be a SuperKey.

5) Fourth Normal Form (4NF)

- It should be in BCNF.
- It is free from multivalued dependency.

Multivalued Dependency

A dependency $A \rightarrow B$ for a single value of $\in A$ there exist multiple values of B, then ~~search~~ such dependencies are called as multivalued dependency.

| S_ID | Course | Interest |
|---|---|---|
| 66 | Math | Drums |
| 66 | Mining | Ayurveda |
| 73 | Java | Drums |

Table with multivalued dependency.

↓

| S_ID | Course |
|---|---|
| 66 | Math |
| 66 | Mining |
| 73 | Java |

| S_ID | Interest |
|---|---|
| 66 | Drums |
| 66 | Ayurveda |
| 73 | Drums |

6) Fifth Normal Form (5NF)

- It should be in 4NF.
- It should not have join dependency.

It is also called as Project Join Normal Form-PJNF

If the table is having join dependency, then it should be broken into smaller pieces.

| Supplier | Product | Customer |
|---|---|---|
| S1 | P1 | C1 |
| S1 | P2 | C2 |
| S2 | P2 | C1 |

↓

| Supplier | Product |
|---|---|
| S1 | P1 |
| S1 | P2 |
| S2 | P2 |

| Supplier | Customer |
|---|---|
| S1 | C1 |
| S1 | C2 |
| S2 | C1 |

| Product | Customer |
|---|---|
| P1 | C1 |
| P2 | C2 |
| P1 | C2 |

## * Transactions

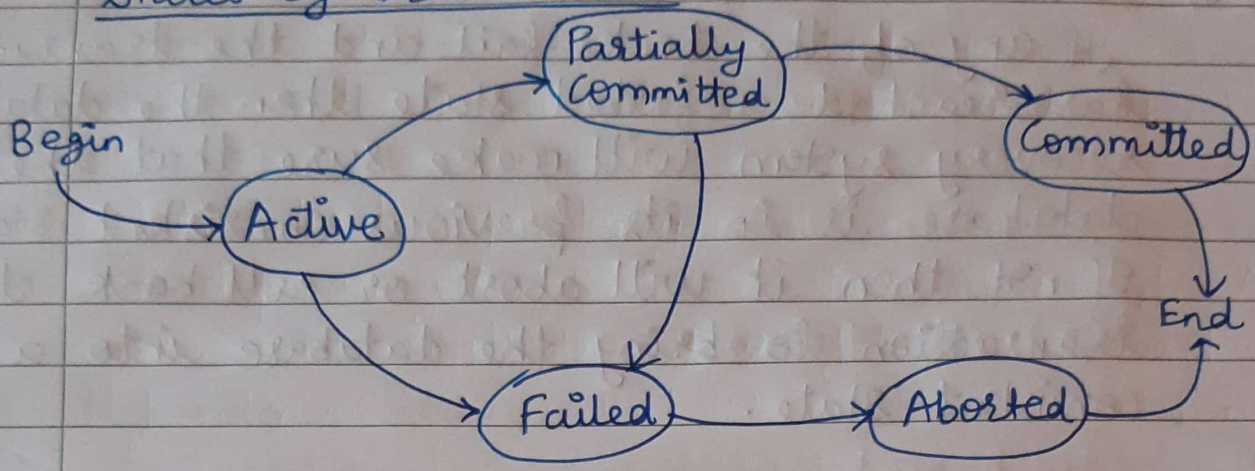
A transaction is a set of logically related operation. It contains a group of tasks.

### - Properties of Transactions

The properties are ACID Properties
(Explained Earlier)

__/__/____

## - States of Transaction




1> Active State

It is the first state of every transaction. In this state, the transaction is being executed.

2> Partially Committed

In the partially ~~comitted~~ committed state, a transaction executes its final operation, but the data is still not saved to the database.

3> Committed.

A transaction is said to be in a committed state if it executes all its operations successfully. In this state, all the effects are now permanently saved on the database system.

4> Failed state

If any of the checks made by the database recovery system fails, then the transaction is said to be in the failed state.

5) Aborted.

If any of the checks fail and the transaction has reached a failed state then the database recovery system will make sure that the database is in its previous consistent state. If not then it will abort or roll back the transaction to bring the database into a consistent state.

If the transaction fails in the middle of the transaction then before executing the transaction, all the executed transactions are rolled back to its consistent ~~re~~ state.

After aborting the transaction, the database recovery module will select one of the two operations :- 1) Restart the transaction.
2) Kill the transaction.

* Schedule

A series of operation from one transaction to another transaction is known as schedule. It is used to preserve the order of the operation in each of the individual transaction.

__/__/____

1> Serial Schedule.

It is a type of schedule where one transaction is executed completely before starting another transaction. In serial schedule, the first transaction completes its cycle, then the next transaction is executed.

2> Non-serial Schedule

If interleaving of operations is allowed, then there ~~is~~ will be non-serial schedule.

It contains many possible orders in which the system can execute the individual operations of the transaction.

3> Serializable Schedule

The serializability of schedules is used to find non-serial schedules that allows the transaction to execute concurrently without interfering with one another.

It identifies which schedules are correct when executions of the transaction have interleaving of their operations.

A non-serial schedule will be serializable if its result is equal to the result of its transactions executed serially.

* Failure Classification.

1> Transaction failure.

2> System Crash.

3> Disk Failure.

* Checkpoint

- The checkpoint is a type of mechanism where all the previous logs are removed from the system and permanently stored in the storage disk.

- The checkpoint is like a bookmark. While the execution of the transaction, such checkpoints are marked, ~~the~~ and the transaction is executed then using the steps of the transaction, the log files will be created.

- When it reaches to the checkpoint, then the transaction will be updated into the database, and till that point, the entire log file will be removed from the file. Then the log file is updated with the ~~step~~ new step of transaction till next checkpoint and so on.

__/__/____

## * Data Abstraction

It is a process of hiding irrelevant details from users. Because database systems are made of complex data structures so, it makes accessible the user interaction with the database.

-3 levels of Data Abstraction :-

1) Physical Level (Lowest level)
It describes how data are stored.

2) Logical Level (Higher Level)
It describes what data are stored in the database and what the relationship among those data is.

3) View Level (Highest Level)
It describes the entire database.

## * Two integrity rules in DBMS

1) Entity Integrity :- It specifies "Primary key cannot have a NULL value.".

2) Referential Integrity :- It specifies "Foreign Key can be either a NULL value or should be the Primary Key value of other relation."

* **Extension**

It is the number of tuples present in a table at any instance. It changes as the tuples are created, updated and destroyed.

So, the data in the database at a particular moment in time is known as extension. It is time dependent.

* **Intension**

It is also known as Data Schema and defined as the description of the database, which is specified during database design and is expected to remain unchanged.

The Intension is a constant value that gives the DB name, structure of tables and the constraints laid on it.