MODULE 4

FUNCTIONAL DEENDANCY:

- In any relation, a functional dependency α → β holds if two tupleshaving same value of attribute α also have same value for attribute β.
- If β is an attribute and α → β, Then β is said to befunctionallydetermined by α.
  - There are two types of functional dependencies
    - Trivial Functional Dependencies
    - Non-trivial Functional Dependencies
- Trivial Functional Dependencies
  - A functional dependency X → Y is said to be trivial if and only if Y

$\subseteq$ X.

- Thus, if RHS of a functional dependency is a subset of LHS, thenit is called as a trivial functional dependency.

- $AB \rightarrow A$
- $AB \rightarrow B$
- $AB \rightarrow AB$
- Non Trivial Functional Dependencies
  - A functional dependency $X \rightarrow Y$ is said to be non-trivial if and onlyif $Y \not\subset X$.
  - Thus, if there exists at least one attribute in the RHS of a

    functional dependency that is not a part of LHS, then it is calledas a non-trivial functional dependency.

Rules of Functional Dependency are called Armstrong's axioms.

- **Reflexivity**
  - If B is a subset of A, then A → B always holds.
- **Transitivity**
  - If A → B and B → C, then A → C always holds.
- **Augmentation**
  - If A → B, then AC → BC always holds.
- **Decomposition**
  - If A → BC, then A → B and A → C always holds.
- **Composition**
  - If A → B and C → D, then AC → BD always holds.

- **Additive**
  - If A → B and A → C, then A → BC always holds.
- The set of all those attributes which can be functionally determinedfrom an attribute set is called as a closure of that attribute set.
- Closure of attribute set {X} is denoted as $\{X\}^+$.
- Consider a relation R (A , B , C , D , E , F , G) with the functionaldependencies-

  A → BC

  BC→DE

  D → F

  CF→

  G

Find the closure of A and D

$A \rightarrow BC$

$BC \rightarrow DE$

$D \rightarrow F$

CF->G

$A \rightarrow BC$ $BC \rightarrow DE$

$D \rightarrow F$

CF->G

- If the closure of an attribute set gives all the attributes on a relation, then the attribute set is the super key of that relation.
- If there exists no subset of an attribute set whose closure contains all the attributes of the relation, then that attribute set is called as a candidate key of that relation.

ANOMALY

- Anomalies are caused when there is too much redundancy in thedatabase's information.
- There are different types of anomaly
  - Insertion Anomaly
  - Deletion Anomaly
  - Update Anomaly

emp_id	**emp_name**	**emp_address**	**emp_dept**	**emp_dept_name**
101	Rajeev	Delhi	D001	IT
101	Rajeev	Delhi	D002	Finance

123	Maggie	Agra	D890	Cleaning
166	Tom	Chennai	D900	Insurance
166	Tom	Chennai	D004	Admin

UPDATEANOMALY

- In the above table, we have two rows for employee Rajeev as hebelongs to two departments of the company.
- If we want to update the address of Rajeev then we have to updatethe same in two rows otherwise the data will become inconsistent.
- This situation is known as update anomalies

INSERTANOMALY

- Suppose a new employee joins the company, who is under training and currently not assigned to any department then we would not beable to insert the data into the table if the e_dept field doesn’t allownulls.

- Such a situation is called Insert Anomalies

DELETEANOMALY

- Suppose, if at a point of time the company closes the department D890 then deleting the rows that are having e_dept as D890 would also delete the information of employee Maggie since she is assigned only to this department.
- Such situation is called delete anomalies.

NORMALIZATION

- In DBMS, database normalization is a process of making thedatabase consistent by
  - Reducing the redundancies
  - Ensuring the integrity of data through lossless decomposition
- Normalization is done through normal forms.

- The standard normal forms used are
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)

FIRST NORMAL FORM

- A given relation is called in First Normal Form (1NF) if each cell ofthe table contains only an atomic value.
- A given relation is called in First Normal Form (1NF) if the attributeof every tuple is either single valued or a null value.
- By default, every relation is in 1NF.

- This is because formal definition of a relation states that value of allthe attributes must be atomic.

Student_id	**Name**	**Mobile Number**
100	Akshay	123456, 2547689

Student_id	**Name**	**Mobile Number**
100	Akshay	123456
100	Akshay	2547689

SECOND NORMAL FORM

- A given relation is called in Second Normal Form (2NF) if and only if
  - Relation already exists in 1NF.
  - No partial dependency exists in the relation.
- A partial dependency is a dependency where few attributes of thecandidate key determines non-prime attribute(s).
- A → B is called a partial dependency if and only if
  - A is a subset of some candidate key
  - B is a non-prime attribute.

- If any one condition fails, then it will not be a partial dependency.
- To avoid partial dependency, incomplete candidate key must notdetermine any non-prime attribute.
- However, incomplete candidate key can determine prime attributes.

- Consider a relation- R (V , W , X , Y , Z) with functional dependencies: VW → XY, Y → V, WX → YZ. The possible candidate keys for this relation are VW , WX , WY.
- Prime attributes = { V , W , X , Y }
- Non-prime attributes = { Z }
- There is no partial dependency.
- This is because there exists no dependency where incompletecandidate key determines any non-prime attribute.

THIRD NORMAL FORM

- A given relation is called in Third Normal Form (3NF) if and only if
  - Relation already exists in 2NF.

  - No transitive dependency exists for non-prime attributes.
- A → B is called a transitive dependency if and only if
  - A is not a super key.
  - B is a non-prime attribute.
- If any one condition fails, then it is not a transitive dependency.
- A relation is called in Third Normal Form (3NF) if and only if
  - Any one condition holds for each non-trivial functionaldependency A → B
    - A is a super key
    - B is a prime attribute
- Consider a relation- R (A , B , C , D , E) with functional dependencies A → BC, CD → E, B → D, E → A. The possible

candidate keys for this relation are A , E , CD , BC.

- From here,
  - Prime attributes = { A , B , C , D , E }
  - There are no non-prime attributes
- It is clear that there are no non-prime attributes in the relation.
- In other words, all the attributes of relation are prime attributes.
- Thus, all the attributes on RHS of each functional dependency areprime attributes.

BOYCE CODD NORMAL FORM

- A given relation is called in BCNF if and only if
  - Relation already exists in 3NF.

- For each non-trivial functional dependency A → B, A is a superkey of the relation.

- Consider a relation- R (A , B , C) with the functional dependencies-A → B, B → C, C → A
- The possible candidate keys for this relation are A,B,C
- Now, we can observe that RHS of each given functionaldependency is a candidate key.
- Thus, we conclude that the given relation is in BCNF.

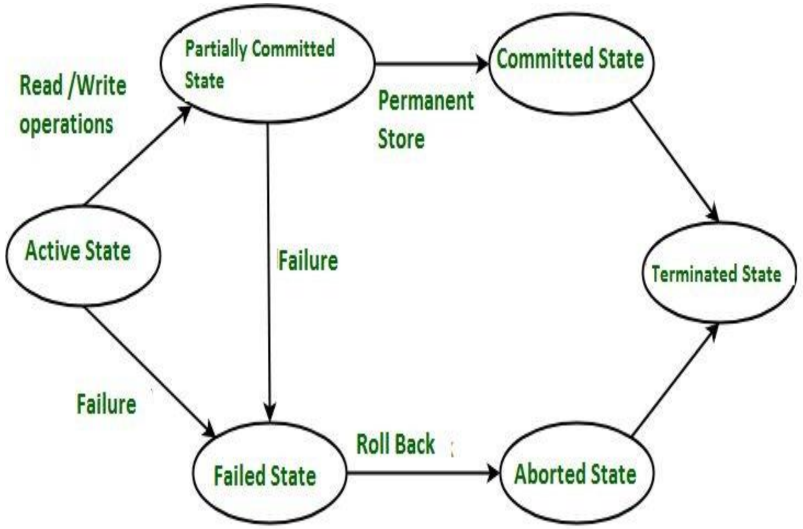
TRANSACTION

- Transaction is a set of operations which are all logically related.
- The main operations in a transaction are
    - Read Operation
    - Write Operation
- Read Operation
    - Read operation reads the data from the database and thenstores it in the buffer in main memory.
    - For example- **Read(A)** instruction will read the value of A fromthe database and will store it in the buffer in main memory.

- Write Operation
  - Write operation writes the updated data value back to thedatabase from the buffer.
  - For example- **Write(A)** will write the updated value of A fromthe buffer to the database.

TRANSACTIONSTATES

- Transaction states are as follows
  - Active state
  - Partially committed state
  - Committed state
  - Failed state
  - Aborted state
  - Terminated state



Transaction States in DBMS

TRANSACTIONSTATES

- **Active state**
    - This is the first state in the life cycle of a transaction.
    - A transaction is called in an **active state** as long as itsinstructions are getting executed.
    - All the changes made by the transaction now are stored inthe buffer in main memory.

TRANSACTION STATES

- **Partially committed state**
    - After the last instruction of transaction has executed, it enters into a **partially committed state**.
    - After entering this state, the transaction is considered to be partially committed.
    - It is not considered fully committed because all the changes made by the transaction are still stored in the buffer in main memory.

TRANSACTION STATES

- Committed state
  - After all the changes made by the transaction have been successfully stored into the database, it enters into a**committed state**.
  - Now, the transaction is considered to be fully committed.
- Failed state
  - When a transaction is getting executed in the active state orpartially committed state and some failure occurs due to which it becomes impossible to continue the execution, it enters into a **failed state**.
- Aborted state
  - After the transaction has failed and entered into a failedstate, all the changes made by it have to be undone.

    - To undo the changes made by the transaction, it becomesnecessary to roll back the transaction.
    - After the transaction has rolled back completely, it entersinto an **aborted state**.

- Terminated state
    - This is the last state in the life cycle of a transaction.
    - After entering the committed state or aborted state, thetransaction finally enters into a **terminated state** whereits life cycle finally comes to an end.

- After a transaction has entered the committed state, it is not possible to roll back the transaction.
- In other words, it is not possible to undo the changes that has beenmade by the transaction.
- This is because the system is updated into a new consistent state.
- The only way to undo the changes is by carrying out anothertransaction called as **compensating transaction** that performs thereverse operations.

ACID PROPERTIES

- It is important to ensure that the database remains consistentbefore and after the transaction.
- To ensure the consistency of database, certain properties arefollowed by all the transactions occurring in the system.
- These properties are called as **ACID Properties** of a transaction.

- It is important to ensure that the database remains consistentbefore and after the transaction.
- To ensure the consistency of database, certain properties arefollowed by all the transactions occurring in the system.
- These properties are called as **ACID Properties** of a transaction.

ATOMICITY

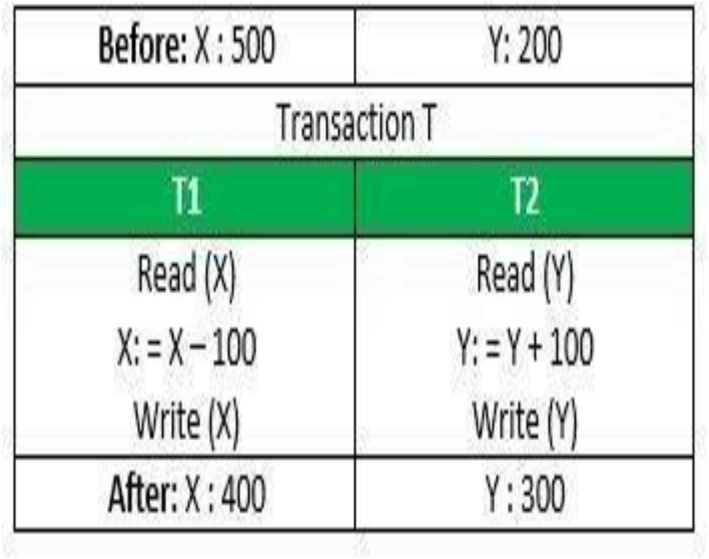
- This property ensures that either the transaction occurs completelyor it does not occur at all.
- In other words, it ensures that no transaction occurs partially.
- That is why, it is also referred to as "**All or nothing rule**".
- It is the responsibility of Transaction Control Manager to ensureatomicity of the transactions.

- It involves the following two operations.
- **Abort**: If a transaction aborts, changes made to database are notvisible.
- **Commit**: If a transaction commits, changes made are visible.

CONSISTENCY

- This property ensures that integrity constraints are maintained.
- In other words, it ensures that the database remains consistentbefore and after the transaction.
- It is the responsibility of DBMS and application programmer toensure consistency of the database.

- The total amount before and after thetransaction must be maintained.Total **before T** occurs = **500 + 200 = 700**. Total **after T occurs** = **400 + 300 = 700**.



Before: X : 500	Y: 200
Transaction T	
T1	T2
Read (X) X: = X − 100 Write (X)	Read (Y) Y: = Y + 100 Write (Y)
After: X : 400	Y : 300

ISOLATION

- This property ensures that multiple transactions can occursimultaneously without causing any inconsistency.
- During execution, each transaction feels as if it is getting executedalone in the system.
- A transaction does not realize that there are other transactions aswell getting executed parallely.
- Changes made by a transaction becomes visible to othertransactions onl
- The resultant state of the system after executing all the transactionsis same as the state that would be achieved if the transactions wereexecuted serially one after the other.
- It is the responsibility of concurrency control manager to ensure isolation for all the transactions.

DURABILITY

- This property ensures that all the changes made by a transactionafter its successful execution are written successfully to the disk.

- It also ensures that these changes exist permanently and are neverlost even if there occurs a failure of any kind.
- It is the responsibility of recovery manager to ensure durability inthe database

CONCURRENCY .

- The ability of a database system which handles simultaneously or a number of transactions by interleaving parts of the actions or the overlapping this is called concurrency of the system.
- Advantages
    - Decreasing waiting time.
    - Decreasing response time.
    - Increases resource utilization.
    - Increases efficiency.
    - Increased throughput

CONCURRENCY PROBLEM

- When multiple transactions execute concurrently in an uncontrolledor unrestricted manner, then it might lead to several problems.
- Such problems are called as **concurrency problems**.

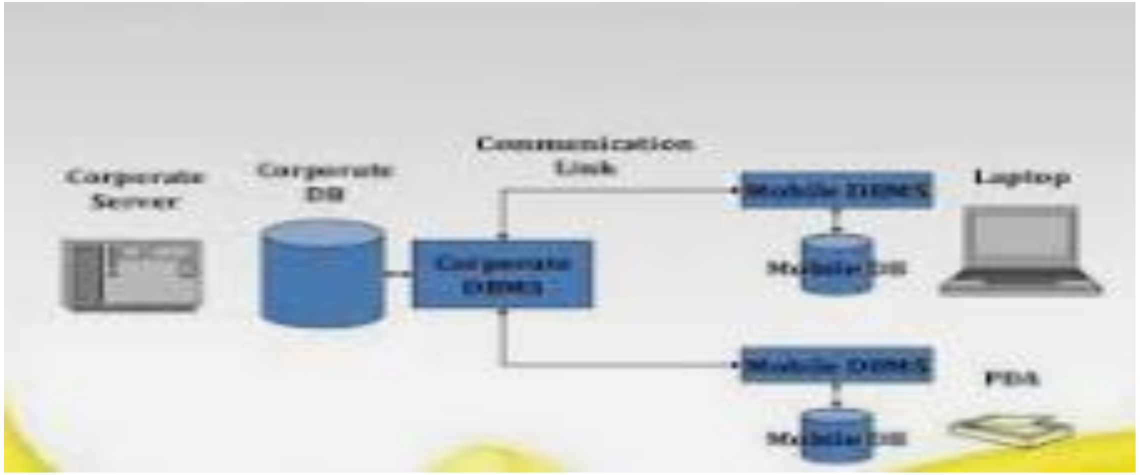
MOBILE DATABASE:

Mobile databases are separate from the main database and can easily be transported to various places. Even though they are not connected to the main database, they can still communicate with the database to share and exchange data.

The mobile database includes the following components −

- The main system database that stores all the data and is linked to the mobile database.
- The mobile database that allows users to view information even while on the move. It shares information with the main database.

- The device that uses the mobile database to access data. This device can be a mobile phone, laptop etc.
- A communication link that allows the transfer of data between the mobile database and the main database.



The components of a mobile database environment include:

- Corporate database server and DBMS that deals with and stores the corporate data and provides corporate applications
- Remote database and DBMS usually manages and stores the mobile data and provides mobile applications
- mobile database platform that includes a laptop, PDA, or other Internet access devices
- Two-way communication links between corporate and mobile DBMS.

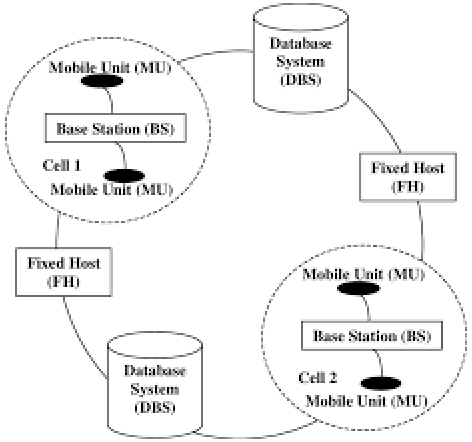
Advantages of Mobile Databases

- Some advantages of mobile databases are −
- The data in a database can be accessed from anywhere using a mobile database. It provides wireless database access.
- The database systems are synchronized using mobile databases and multiple users can access the data with seamless delivery process.
- Mobile databases require very little support and maintenance.
- The mobile database can be synchronized with multiple devices such as mobiles, computer devices, laptops etc.

Disadvantages of Mobile Databases

- It has Limited wireless bandwidth.
- In the mobile database, Wireless communication speed.
- It required Unlimited battery power to access.
- It is Less secured.
- It is Hard to make theft-proof.

Mobile Database typically involves three parties :



Fixed Hosts –

- It performs the transactions and data management functions with the help of database servers.

Mobiles Units –

- These are portable computers that move around a geographical region that includes the cellular network that these units use to communicate to base stations.

Base Stations –

- These are two-way radios installation in fixed locations, that pass communication with the mobile units to and from the fixed hosts.