Ans 1 :- Normalization is a process in DBMS to reduce redundancy and dependency by organizing data in tables. It divides large tables into smaller ones and links them using relationships.

Here are some normal forms :

- 1st normal form (1NF) : Ensure that each columns contains atomic values and each has a unique value.
- 2nd normal form (2NF) : A table is in 2NF, if it is in 1NF and all non-key attributes are fully dependent on the primary key.
- 3rd Normal form (3NF) : A table is in 3NF if it is in 2NF & there is no transitive dependency i.e non-key attribute are not dependent on other non-key attributes.

Ans 2: A primary key is a unique identifier for records in a table. It must contain unique values and cannot contain Null values. A table can have only one primary key, which can be a single column or a combination of columns.

It is important because of the following points.

1) It ensures that each record in a table is unique.
2) It helps in establishing relationship between tables when used with foreign keys.
3) It is essential for indexing, making searches and retrievals faster.

Ans 3 1) Atomicity : Ensures that a transaction is either fully completed or fully rolled back. It prevents partial updates.

2) Consistency : Guarantees that a transaction is either fully will take the database from one valid state to another valid state, maintaining data integrity.

3) Isolation : Ensures that transactions are executed independently of each other. One transaction's changes will not be visible to others until it is completed.

4) Durability : Ensures that once a transaction is committed, its changes are permanent, even in the event of a system failure.

These properties are critical to ensuring the correctness and reliability of transactions in a database.

Ans 4. A deadlock in DBMS occurs when two or more transactions are waiting for each other to release locks, and none of them can proceed. for example, if transactions A hold a lock on Resources 1 and waits for resources 2, while Transaction B holds a lock on Resources 2 and waits for Resources 1, both will be stuck.

Deadlock can be prevented by:

1) Deadlock Prevention: By using protocols that avoid the conditions leading to deadlock, like resource ordering and timeouts.
2) Deadlock Detection: Regularly checking for deadlock cycles in the system and aborting one of the transactions to break the cycle.
3) Deadlock Avoidance: Using algorithms like the Banker's algorithm to check the system state before granting locks, ensuring it doesn't lead to a deadlock.

Ans 5: A distributed database is one in which data is stored across multiple locations, but it appears to user also single database.

Advantages:-

1) Reliability and Availabity:- If one site fails, data base can continue functioning using other sites.
2) Scalability:- It's easier to add new sites or nodes without affecting the whole system.
3) Local control:- Each site can have control over its data, providing flexibility and faster local queries.

Disadvantages:-

1) Complexity:- Managing a distributed system is more complex due ~~to~~ ~~n~~ need for coordination across multiple sites.
2) Security Risks:- Since data is spread across locations, it increases the surface area for potential security threats.
3) Data Integrity:- keeping the data consistent across all sites can be chaddenging especially with concurrent transactions.