Database Management Systems (DBMS) provide structured storage, retrieval, and management of data. One essential concept in relational databases is **normalization**, which aims to reduce redundancy and ensure data integrity. Normalization involves decomposing tables into well-structured forms called *normal forms*.

**1NF (First Normal Form)** requires that table cells contain atomic values and that each record is unique. **2NF** removes partial dependency, meaning no non-key attribute should depend on only part of a composite primary key. **3NF** removes transitive dependency, ensuring non-key attributes depend only on primary keys. Proper normalization prevents anomalies such as update, insertion, and deletion anomalies.

In DBMS, relationships between tables are established through **joins**:

- **INNER JOIN:** Returns only matching rows from both tables.
- **LEFT JOIN:** Returns all rows from the left table and matching rows from the right.
- **RIGHT JOIN:** Opposite of left join.
- **FULL OUTER JOIN:** Returns all rows when there is a match in either table.
- **CROSS JOIN:** Produces a Cartesian product.

Joins enable relational databases to maintain normalized structures while still retrieving meaningful combined data.

Another key concept is **transactions**, which represent a sequence of operations performed as a single logical unit of work. Transactions must satisfy the **ACID properties**:

- **Atomicity:** All steps succeed or none.
- **Consistency:** The database must remain valid before and after the transaction.
- **Isolation:** Concurrent transactions must not interfere with each other.
- **Durability:** Changes persist even if the system crashes.

Database systems use locking, logging, checkpoints, and isolation levels to ensure transaction safety. Understanding normalization, joins, and transactions is critical because these concepts form the backbone of relational data handling.