1. **What is database normalization?**

Normalization is a process of organizing the columns and tables of a relational database to minimize data redundancy and improve data integrity.

1. **Why is normalization important?**

Normalization is crucial for several reasons that contribute to a healthy and efficient database;

1. **Reduce data redundancy**: By eliminating duplicate data, its saves storage space and prevents inconsistencies that can arise when the same data is stored in multiple places.
2. **Improve data integrity**: Ensures that data is consistent and accurate across the database. When data changes, it only needs to be updated in one place.
3. **Enhances Data Consistency**: Prevents update, insertion, and deletion anomalies, which are problems that occur when data is not properly organized.
4. **Simplifies Queries**: While it might seem counterintuitive due to more joins, a normalized schema often makes it easier to write clear and specific queries because each piece of information has a designated, logical home.
5. **Better Database Design**: Promotes a cleaner, more logical and more scalable database structure.

Imagine a school database. If you store student details (name, address, and phone) directly in a Courses table for every course a student takes, you'll repeat the student's name and address multiple times. If the student moves, you'd have to update their address in many places, risking errors. Normalization would put student details in a separate Students table, and the Courses table would just have a StudentID to link to it.

**Advantages of Normalization**

1. Removes redundancy
2. Improves data integrity
3. Easy to update & maintain
4. Saves storage
5. Better organization

**Disadvantages of Normalization**

1. More complex quires
2. Slower reads
3. Hard for beginners
4. Difficult for reporting

**What is De-normalization?**

**De-normalization** is the process of combining normalized tables into a single table (or fewer tables) to improve read/query performance, even though it may increase redundancy and risk of consistencies.

**Difference between Normalization and De-normalization**

Normalization

Purpose - Reduce redundancy, ensure consistency => Improve read/query performance

Data Storage – Less storage, compact => More storage, duplicate data

Joins - More joins needed => Fewer joins

Redundancy – Redundancy remove => Redundancy introduced

Data Integrity - High data integrity => Lower integrity (more chances of error)

**What is Redundancy?**

Redundancy means storing the same piece of data in multiple places unnecessarily.