**What is normalization in sql**

Database normalization is an important process used to organize and structure relational databases.

This process ensures that data is stored in a way that minimizes redundancy, simplifies querying, and improves data integrity.

Normalization is the process of organizing data within a database to eliminate redundancy.

Normalization involves breaking down a large, complex table into smaller and simpler tables while maintaining data relationships.

Normalization is commonly used when dealing with large datasets.

A major problem with redundant data is that it occupies unnecessary storage space. If we store the same product details in every order record, it leads to duplication. With normalization, we can eliminate redundancy by splitting data into separate tables.

**Why is Normalization in SQL Important?**

Normalization basically plays very important role in database design.

Reasons why normalization is essential:

1. **It Reduces redundancy:** when the same information is stored multiple times, it creates duplicates and a good way of avoiding this is to split the data into smaller tables.
2. **Improves performance:**You can perform faster query execution on smaller tables that have undergone normalization.
3. **It Minimizes update anomalies:** We can easily update data without affecting other records.
4. **Enhanced data integrity:**  Normalization ensures that data remains consistent and accurate.

**Types of Database Normalization**

**First Normal Form(1NF)**

This normalization level ensures that each column in your data contains only atomic values. Atomic values in this context means that each entry in a column is indivisible.

It is like saying that each cell in a spreadsheet should hold just one piece of information. 1NF ensures atomicity of data, with each column cell containing only a single value and each column having unique names.

**Second Normal Form (2NF)**

It eliminates partial dependencies by ensuring that non-key attributes depend only on the primary key.

There should be a direct relationship between each column and the primary key column and not between other columns.

**Third Normal Form (3NF)**

It removes transitive dependencies by ensuring that non-key attributes depend only on the primary key. This level of normalization builds on 2NF.

**Boyce-Codd Normal Form (BCNF)**

This is a strict version of 3NF that addresses additional anomalies. At this normalization level, every determinant is a candidate key.

**Fourth Normal Form (4NF)**

This is a normalization level that builds on BCNF by dealing with multi-valued dependencies.

**Fifth Normal Form (5NF)**

5NF is the highest normalization level that addresses join dependencies. It is used in specific scenarios to further minimize redundancy by breaking a table into smaller tables.