Functional Dependency

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

Functional dependency helps you to maintain the quality of data in the database. A functional dependency is denoted by an arrow →. The functional dependency of X on Y is represented by X → Y. Functional Dependency plays a vital role to find the difference between good and bad database design.

Example:

Empno	EmpName	Salary	City
1	SMITH	5000	Ranchi
2	SCOTT	6000	Bokaro
3	CLARKE	7000	Tata

In this example, if we know the value of Empno, we can obtain EmpName, city, salary, etc. By this, we can say that the city, EmpName, and salary are functionally depended on EmpNo.

To get the dependencies to all the properties X, Y, and Z are set of attributes in a relation R. So it can representing in several ways according to rules of functional dependency like:

1. Reflexive Rule

In the reflexive rule, if Y is a subset of X, then X determines Y.

X → Y

Example:

X = {a, b, c, d, e}

Y = {a, b, c}

2. Augmentation Rule

The augmentation is also called as a partial dependency. In augmentation, if X determines Y, then XZ determines YZ for any Z.

If X → Y then XZ → YZ

Example:

For R(ABCD), **if** A → B then AC → BC

3. Transitive Rule

In the transitive rule, if X determines Y and Y determine Z, then X must also determine Z.

If X → Y and Y → Z then X → Z

4. Union Rule

Union rule says, if X determines Y and X determines Z, then X must also determine Y and Z.

If X → Y and X → Z then X → YZ

5. Decomposition Rule

Decomposition rule is also known as project rule. It is the reverse of union rule.

This Rule says, if X determines Y and Z, then X determines Y and X determines Z separately.

If X → YZ then X → Y and X → Z