

SI No	Terminology	Types	Definition
1	Normalization		It is a process of efficiently organizing data in a database. It is a set of rules / guidelines / statements that we follow while storing data. There are 2 reasons for the Normalization process: 1) Eliminating Redundant / Duplicate Data, for example, storing the same data in more than one tables 2) Ensuring Data Dependencies makes sense
2		First Normal Form (1NF)	1) Define the data items. This means looking at the data to be stored, organizing the data into columns, defining what type of data each column contains, and finally putting related columns into their own table. 2) Ensure that there are no repeating groups of data 3) Ensure that there is a primary key
3		Second Normal Form (2NF)	1) It should meet all the rules for 1NF 2) There must be no partial dependences of any of the columns on the primary key
4		Third Normal Form (3NF)	1) It should meet all the rules for 2NF 2) Tables should have relationship.
5	Constraints		Constraints are the rules enforced on data columns on table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.
6		Primary Key	Uniquely identifies each row/record in a database table. Specify it as NOT NULL while declaring a column as primary key after table creation. Not required at the time of table creation
7		Unique	Ensures all values in a column are different
8		Not Null	Ensures a column cannot have null values
9		Default	Provides a default value to a column when none is specified
10		Foreign Key	Uniquely identifies a row/record in any other database table
11	Joins	Check	Ensures that all values in a column satisfy certain conditions
12			Joins are required when we need to <b>retrieve</b> data from more than one table with at least one common field
13		Inner Join	Returns rows when there is a match in both tables
14		Left Join	Returns all rows from the left table even when there is no match in the right table
15		Right Join	Returns all rows from the right table even when there is no match in the left table
16		Full Join	Returns rows when there is a match in one of the tables
17		Cartesian or Cross Join	Returns all records from the right table for each record in the left table
18	Indexes	Self Join	Joins a table to itself, as if the table were two tables, temporarily renaming at least one table in the SQL statement. There is nothing known as SELF JOIN
19		INDEX	Indexes are special lookup tables that the database search engine can use to speed up data retrieval. It speeds up data retrieval but slows down data manipulation like Insert, Delete etc.
20		Composite Index	Creates index for more than 1 column name
21		Explicit Index	Index created by user
22		Implicit Index	Index created automatically by SQL, i.e while creating Primary Key or Unique constraint on a table
23		Clustered Index	Data will be shuffled and information in the column will be clustered where all similar items will be kept together. SQL will automatically create Clustered Index for Primary Key
24	SQL Views	Non Clustered Index	Ordinary index
25		Views	Views are the <u>virtual tables</u> , i.e. Tables not having rows and columns. A view is nothing more than a SQL statement that is stored in the database with an associated ID. A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.
26	SQL Rules	Rules	While defining the VIEW, SELECT clause should not contain DISTINCT, SUMMARY, SET, ORDER BY FROM clause should not contain multiple tables WHERE clause should not contain sub-queries Query should not contain GROUP BY or HAVING Calculated columns created in the view should not be updated All columns with NOT NULL constraint in the base table must be included in the view for INSERT to function