**INDEX:**

• An index creates an entry for each value, and it will be faster to retrieve data.

• A SQL Server Index is used on a database table for faster data access.

• To facilitate quick retrieval of data from a database.

• An index contains information that allows us to find specific data without scanning through the entire table.

• Indexes are created on tables and views.

• Index on a table or view, is very similar to an index that we find in a book.

• In fact, the existence of the right indexes, can improve the performance of the query. If there is no index to help the query, then query engine checks every row in the table from the beginning to end. This is called as table scan. Table scan is very bad for performance.

• Indexes are mostly performed on large tables.

• Create indexes on columns that will be frequently searched against.

• An index is a pointer to data in a table.

• An index helps to speed up SELECT queries and WHERE clauses, but it slows down data input, with the update and insert statements. Indexes can be created or dropped with no effect on the data.

• Indexes are automatically created when PRIMARY KEY and UNIQUE constraints are defined on table.

• SP\_HELPINDEX system store procedure used to find indexes on a table.

**Single Column Indexes:**

• A single column indexes are created based on only one table column.

**Implicit Indexes:**

• Implicit indexes are indexes that are automatically created by the database server when an object is created. Indexes are automatically created for primary key constraints and unique constraints.

**When should indexes be avoided:**

• Indexes should not be used on small tables.

• Tables that have frequent, large batch updates or insert operations.

• Indexes should not be used on columns that contain a high number of NULL values.

• Columns that are frequently manipulated should not be indexed.

**Clustered Index:**

• A clustered index causes records to be physically stored in a sorted or sequential order.

• A clustered index determines the actual order in which data is stored in the database. Hence, we can create only one clustered index in a table.

• Uniqueness of value in a clustered index is maintained explicitly using the UNIQUE keyword or implicitly using an internal unique identifier.

• Clustered index is same as dictionary where the data is arranged by alphabetical order.

• We can have only one clustered index in one table, but we can have one clustered index on multiple columns, and that type of index is called **composite index**.

**Non-Clustered Index:**

• A non-clustered is same as an index of a book.

• The data is stored in one place, and index is stored in another place.

• Since the non-clustered index is stored separately from the actual data, a table can have more than one non-clustered index.

**Unique and Non-Unique indexes in SQL:**

• A unique index can be created on a column that does not have any duplicate values.

• Once a unique index is created, duplicate values will not be accepted in the column.

• Thus, unique indexes should be created only on columns where uniqueness of values is a key characteristic.

• A unique index ensures entity integrity in a table.

• If a table definition has a PRIMARY KEY or a column with a unique constraint, SQL Server automatically creates a unique index when we execute the create table statement.

• UNIQUENESS is a property of an index, and both clustered and non-clustered indexes can be unique.

• There is no big difference between a unique constraint and a unique index.