MS SQL Server INDEX

Why We need INDEX and what is Table Scan

- Consider the example of a book
- Imagine it has 10 chapters about human anatomy , but with NO INDEX at the end
- To find about HEART and its functions you have to read the whole book to find which chapter have information.
- Similarly if we have a table with no INDEX, Its Called a HEAP.
- A heap is a table that is stored without any underlying order.
- When rows are inserted into a heap, there is no way to ensure where the pages will be written nor are those pages guaranteed to remain in the same order as the table is written to or when maintenance is performed against it.
- To find a row the SQL Server engine has to do TABLE SCAN
- A TABLE SCAN is a pretty straightforward process. When your query engine performs a table scan it starts from the physical beginning of the table and goes through every row in the table. If a row matches the criterion then it includes that into the result set.

What is an INDEX

- The primary reason indexes are built is to provide faster data access to the specific data your query is trying to retrieve.
- This could be either a clustered or non-clustered index.
- Without having an index SQL Server would need to read through all of the data in order to find the rows that satisfy the query.
- Indexes are created on columns in tables or views.
- The index provides a fast way to look up data based on the values within those columns.
- For example, if you create an index on the primary key and then search for a row of data based on one of the primary key values, SQL Server first finds that value in the index, and then uses the index to quickly locate the entire row of data.
- You can create indexes on most columns in a table or a view.
- The exceptions are primarily those columns configured with large object (LOB) data types, such as image, text, and varchar(max).

Structure of an INDEX

- An index is made up of a set of pages (index nodes) that are organized in a B-tree structure.
- This structure is hierarchical in nature.
- Root node at the top of the hierarchy and the leaf nodes at the bottom.
- When a query is issued against an indexed column, the query engine starts at the root node and navigates down through the intermediate nodes, with each layer of the intermediate level more granular than the one above.
- The query engine continues down through the index nodes until it reaches the leaf node.