Index Fragmentation and Lab for Index

Index Fragmentation

- SQL Server index fragmentation is a common source of database performance degradation.
- Fragmentation occurs when there is a lot of empty space on a data page (internal fragmentation)
- When the logical order of pages in the index doesn’t match the physical order of pages in the data file (external fragmentation).
- Fragmentation-related performance issues are most often observed when executing queries that perform index scans.
- Queries that perform index seeks may not be affected by high index fragmentation.
- The index fragmentation is the index performance value in percentage, which can be fetched by SQL Server DMV.

Internal index fragmentation

- Internal fragmentation occurs when data pages have too much free space.
- This extra space is introduced through a few different avenues:
- SQL Server stores data on 8KB pages. So when you insert less than 8KB of data into a table, you're left with blank space on the page.
- if you insert more data than the page has space for, the excess is sent to another page. It's unlikely that the additional data will perfectly fill the subsequent pages, so you are, again, left with blank space on a page.
- Blank space on a data page also occurs when data is deleted from a table.
- Internal fragmentation primarily causes performance issues when SQL Server does an index scan.
- Performance slows when SQL Server has to scan many partially filled pages to find the data it's looking for.

External index fragmentation

- External fragmentation is a result of data pages being out of order.
- This is caused by inserting or updating data to full leaf pages. When data is added to a full page, SQL Server creates a page split to accommodate the extra data, and the new page is separated from the original page.
- External fragmentation causes performance issues by increasing random I/O.
- When pages are not sequential, SQL Server has to read data from multiple locations, which is more time-consuming than reading in order.

Fix SQL Server Index fragmentation

- Preventing fragmentation 100 percent of the time is impossible, it’s important to know how to fix SQL Server index fragmentation if performance is suffering.
- Before you can decide how to approach your SQL Server index fragmentation problem, you first have to determine how extensive a problem you are dealing with.
- The best place to start is using the sys.dm_db_index_physical_stats DMF to analyze the fragmentation level of your indexes.
- Once you know how extensive the index fragmentation is, you can plot your plan of attack with one of three solutions: rebuild the index, reorganize the index, or do nothing.
- **Rebuild:** Rebuild indexes when fragmentation reaches greater than 30 percent.
- **Reorganize:** Reorganize indexes with between 11-30 percent fragmentation.
- **Ignore:** Fragmentation levels of 10 percent or less should not pose a performance problem, so you don’t need to do anything.