|  |  |  |
| --- | --- | --- |
|  | Index Rebuild | Index Re-organize |
| Space Required | Rebuilding an index requires building a new index before dropping the old index, regardless of the amount of fragmentation present in the old index. This means you need to have enough free space to accommodate the new index. | Reorganizing an index first squishes the index rows together to try to deallocate some index pages, and then shuffles the remaining pages in place to make their physical (allocation) order the same as the logical (key) order. This only requires a single 8-KB page, as a temporary storage for pages being moved around |
| Algorithm Speed | An index rebuild will always build a new index, even if there’s no fragmentation. The length of time the rebuild takes is related to the size of the index, **not the amount of fragmentation in it.** | Reorganizing an index only deals with the fragmentation that exists, and **the more fragmentation there is, the longer a reorganize will take.** |
| Transaction Log Generated | In the FULL recovery mode, an index rebuild is fully logged, so the transaction log will have to accommodate the full size of the index in a single transaction.  In the SIMPLE and BULK\_LOGGED recovery modes, the amount of transaction log generated by an offline index rebuild will be minimal – just the allocations of pages and extents. However, the next log backup performed will also contain all the extents changed by the rebuild, and so the log backup will be roughly the same size as if the rebuild was done in the FULL recovery mode. | In all recovery modes, reorganizing an index is fully logged, but is performed as a series of small transactions so should not cause the transaction log to grow inordinately. |
| Locks Required | **An offline index rebuilds of any index holds a schema-modification.**  An online index rebuild of any index acquires a short-term shared table lock at the start of the operation, holds an intent-shared table lock throughout the operation. **And then acquires a short-term schema-modification table lock at the end of the operation.** | An index reorganize holds an intent-exclusive table lock throughout the operation, which will only block shared, exclusive, and schema-modification table locks |
| Interruptible or Not | An index rebuild operation cannot be interrupted without it rolling back everything it’s done so far – it’s atomic – all or nothing. In SQL Server 2017, however, there is a resumable-online index rebuild feature. | An index reorganize can be interrupted and the worst that will happen is that a single page move operation is rolled back. |
| Progress Reporting or Not | Index rebuilds do not have proper progress reporting. There are other ways to say by looking at the bigintdata1 column in the Progress Report: Online Index Profiler event, which happens to show how many rows of the old index have been scanned. | Index reorganize operations populate the percent\_complete column of sys.dm\_exec\_requests |
| Statistics | An index rebuild will always rebuild the index column statistics with the equivalent of a full scan | manual index statistics maintenance is required. |