**CHANGE DATA CAPTURE**

When a configurable size-on-disk for the CDC log is reached, Change Data Capture (CDC) offers a technique to refuse writes to certain tables and flag them for archiving. By setting the table property cdc=true (either when creating the table or modifying it), an operator can enable CDC on a table. A hard-link to the CommitLogSegment is made in the directory designated by cassandra.yaml at the time of the segment's creation. A <segment\_name>\_cdc.idx file is also created with the integer offset of how much data from the original segment is persisted to disk when a segment fsyncs to disk and contains CDC data.

The human-readable term "COMPLETED" will be appended to the \_cdc.idx file on the second line following final segment flush, signifying that Cassandra has finished processing the file.   
  
Since data can reflect in a kernel buffer that is not yet persisted to disk, we employ an index file instead of just pushing clients to parse the log in real-time off a memory mapped handle. To make sure you only parse durable CDC data, only parse up to the offset specified in the \_cdc.idx file.

Please be aware that while updating is accomplished by first truncating the file and then writing to it, there is a potential that the consumer will read an empty value from the \_cdc.idx file in unusual circumstances, such as a sluggish disk. The customer should attempt reading the index file again in such a situation.

A consumer must parse and remove files from the designated cdc\_raw directory before newly allocated CommitLogSegments will enable CDC data, according to a threshold of total disk space allowed that is specified in the yaml file.

OPERATING CDC :

The cdc table property can be used to enable or disable CDC.

For example:

CREATE TABLE foo (a int, b text, PRIMARY KEY(a)) WITH cdc=true;

ALTER TABLE foo WITH cdc=true;

ALTER TABLE foo WITH cdc=false;

**COMPACTION OVERVIEW :**

Memtables are used in Cassandra to create data. The data is written to an immutable disk file called an SSTable once a memory threshold is reached in order to free up memory once more.

Since SSTables are immutable, previous data is not overwritten by new inserts or updates or eliminated from the SSTable when data is modified or removed. Rather, the old SSTable is marked for destruction and a new one is produced with the changed data and a new timestamp. The erased data point is referred to as a tombstone.

Cassandra may write multiple copies of a row in various SSTables over time. It's possible that every version has a distinct set of columns that are timestamped differently.

The distribution of data may involve accessing an increasing number of SSTables in order to extract a complete row as SSTables grow.

Every now and then Cassandra merges SSTables and removes outdated data to maintain the database's health. Compaction is the term for this procedure.

**ESSENTIALS OF COMPACTION :**

Because read operations consult SSTables, it's critical to maintain a minimal amount of SSTables. Compaction is required because write operations will result in an increase in the number of SSTables. In addition to the tombstone problem, there are other reasons why data gets erased, as when some data's Time-To-Live (TTL) expires. Compaction can occur when data is updated, deleted, or expires.

**WORKING :**

Compaction processes a group of SSTables. Compaction uses the most recent version (by timestamp) of each row's columns to create a single, full row from these SSTables by gathering all versions of each unique row. Because each SSTable's rows are sorted by partition key and no random I/O is used during the merge process, it is performant. Each row's updated version is written to a fresh SSTable. When pending reads are finished, the previous versions are removed from the old SSTables along with any rows that are ready to be destroyed.

Compaction achieves two major goals: disk space reclamation and performance enhancement. Read operations in SSTables with redundant data that has to be read are slower. Read operations are faster when duplicates and tombstones are eliminated. SSTables consume disk space; compaction allows you to reduce the size of SSTables and free up storage space.S