

Exercise 18 – Compaction

In this exercise, you will:

- Understand basic Apache Cassandra™ compaction strategies

As memtables fill up, Apache Cassandra™ writes them to disk in the form of SSTables. If this were the end of the story, the number of data files used to contain SSTables would become large and slow the Apache Cassandra™ read performance. Therefore, Apache Cassandra™ must consolidate these files from time to time. This consolidation is called **compaction**.

In this exercise, we observe the effects of compaction.

Steps

- 1) Launch `cqlsh`. Let's recreate the `killrvideo` keyspace and also create the `videos_by_tag` database:

```
CREATE KEYSPACE killrvideo
WITH replication = {'class':'SimpleStrategy', 'replication_factor': 1};

USE killrvideo;

CREATE TABLE videos_by_tag (
  tag TEXT,
  video_id UUID,
  added_date TIMESTAMP,
  title TEXT,
  PRIMARY KEY ((tag), video_id)
);
```

- 2) Now, let's insert a single row into the table:

```
INSERT INTO videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Master');
```

- 3) Exit `cqlsh` and for Apache Cassandra™ to flush the memtable to an SSTable:

```
/home/ubuntu/node/bin/nodetool flush
```

- 4) Let's investigate the SSTable in the node's data directory. Remember the actual name of the directory will be a unique random value under /home/ubuntu/node/data/data/killrvideo:

```
ls -l /home/ubuntu/node/data/data/killrvideo/videos_by_tag-*
```

You will see several files with names that start with. These are the files associated with the first SSTable.

- 5) We can create a second SSTable. Add a second row using the following:

```
INSERT INTO killrvideo.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Genius');
```

- 6) Flush this second memtable to disk:

```
/home/ubuntu/node/bin/nodetool flush
```

- 7) Re-inspect the data directory to see the files associated with the _two_ SSTables:

```
ls -l /home/ubuntu/node/data/data/killrvideo/videos_by_tag-*
```

- 8) Create a third SSTable by inserting the following row:

```
INSERT INTO killrvideo.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Wizard');
```

- 9) Once again, flush the memtable to disk and investigate the data directory:

```
/home/ubuntu/node/bin/nodetool flush
```

```
ls -l /home/ubuntu/node/data/data/killrvideo/videos_by_tag-*
```

NOTE: When Apache Cassandra™ goes to create a fourth SSTable, Apache Cassandra™ will perform compaction.

- 10) Insert a fourth row:

```
INSERT INTO killrvideo.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Ninja');
```

- 11) Flush and investigate the data directory again. Wait a few seconds before executing each command.

```
/home/ubuntu/node/bin/nodetool flush
```

```
ls -l /home/ubuntu/node/data/data/killrvideo/videos_by_tag-*
```

Notice that the three previous SSTable files are gone and a new set has appeared. Also, notice that the file names skipped from mc-3-big to mc-5-big. This new set is the result of the compaction. Apache Cassandra™ created the fourth SSTable and then compacted all four into a fifth SSTable.