

Exercise 4 – Clustering Columns

In this exercise, you will:

- Understand how clustering columns affect the underlying storage mechanism.
- Understand how clustering columns affect queries.

Clustering columns are those columns that are part of the primary key, but are not the partition key. This exercise will help you understand how clustering columns affect what and how you can query.

Steps

- 1) Ensure you are using the killrvideo keyspace by executing the line below:

```
USE killrvideo;
```

- 2) Drop your `videos_by_tag` table that you created in the previous exercise.

```
DROP TABLE videos_by_tag;
```

- 3) Modify the following command to create a new `videos_by_tag` table partitioned based on the tag. The table should also store the rows of each partition so that the newest videos are listed first within the partition.

NOTE: Remember that you must include `WITH CLUSTERING ORDER BY(column1 ASC/DESC, column2 ASC/DESC, etc.)` to do so.

```
CREATE TABLE videos_by_tag (  
    tag text,  
    video_id uuid,  
    added_date timestamp,  
    title text,  
    PRIMARY KEY ());
```

- 4) Import the `videos_by_tag.csv` again via the `COPY` command by executing the code below.

```
COPY videos_by_tag(tag, video_id, added_date, title)  
FROM '/home/ubuntu/labwork/data-files/videos-by-tag.csv'  
WITH HEADER = TRUE;
```

5) Perform a `SELECT *` query on `videos_by_tag`.

NOTE: Notice the rows are still grouped by their partition key value but ordered in descending order of added date.

6) Execute your query again, but list the oldest videos first.

7) Change your query to restrict the partition key value to `'cassandra'`.

8) Change your query to retrieve videos made in 2013 or later.