03_ClusteringColumns_Solution

READY

READY

```
%sh
STATUS="$(service cassandra status)"

if [[ $STATUS == *"is running"* ]]; then
    echo "Cassandra is running"
else
    echo " Cassandra not running .... Starting"
    service cassandra restart > /dev/null 2>&1 &
    echo " Started"
fi
```

FINISHED

Exercise 3 – 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.

Took 0 sec. Last updated by anonymous at July 13 2020, 9:11:28 PM. (outdated)

READY

Steps

1. Ensure you are using the YooToob keyspace by executing the line below:

READY

```
%cassandra

USE YooToob;
```

READY

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

```
DROP TABLE videos_by_tag;
```

READY

03_ClusteringColumns_Solution

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.

READY

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

READY

```
CREATE TABLE videos_by_tag (
tag text,
video_id uuid,
added_date timestamp,
title text,
PRIMARY KEY ((tag), added_date, video_id)
) WITH CLUSTERING ORDER BY(added date DESC);
```

READY

4. Insert the videos_by_tag.csv again via the insert commands by executing the code below.

FINISHED

Took 0 sec. Last updated by anonymous at July 13 2020, 2:38:18 PM.

```
INSERT INTO videos_by_tag(tag, video_id, added_date, title)
VALUES ('cassandra', 1645ea59-14bd-11e5-a993-8138354b7e31, '2014-01-29', 'Cassandra History');

INSERT INTO videos_by_tag(tag, video_id, added_date, title)
VALUES ('cassandra',245e8024-14bd-11e5-9743-8238356b7e32, '2012-04-03', 'Cassandra & SSDs');

INSERT INTO videos_by_tag(tag, video_id, added_date, title)
VALUES ('cassandra',3452f7de-14bd-11e5-855e-8738355b7e3a, '2013-03-17', 'Cassandra Intro');

INSERT INTO videos_by_tag(tag, video_id, added_date, title)
VALUES ('fenago',4845ed97-14bd-11e5-8a40-8338255b7e33, '2013-10-16', 'Apache Cassandra');

INSERT INTO videos_by_tag(tag, video_id, added_date, title)
VALUES ('fenago',5645f8bd-14bd-11e5-af1a-8638355b8e3a, '2013-04-16', 'What is Apache Cassandra?'
```

READY

5. Perform a SELECT * query on videos_by_tag.

READY

```
SELECT *
FROM videos_by_tag;
```

READY

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

READY

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

READY

03_ClusteringColumns_Solution

Note: Your query will fail. When specifying ORDER BY, you must restrict the partition key.

```
SELECT *
FROM videos_by_tag
ORDER BY added_date ASC;
```

READY

7. Change your query to restrict the partition key value to ‘cassandra’.

READY

Note: Clustering columns allow range queries (<, <=, >=).

```
SELECT *
FROM videos_by_tag
WHERE tag = 'cassandra'
ORDER BY added_date ASC;
```

READY

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

READY

```
SELECT *
FROM videos_by_tag
WHERE tag = 'cassandra' and added_date >= '2013-1-1'
ORDER BY added_date ASC;
```

READY