DATASTAX ACADEMY

Exercise 3 – Partitions

In this exercise, you will:

- Experiment with partitions

Steps

NOTE: Be sure Apache Cassandra™ is running before doing these exercises. You can check by running dsetool on the command line:

```
/home/ubuntu/node/bin/dsetool status
```

1) Start the CQL command shell at the command line:

```
/home/ubuntu/node/resources/cassandra/bin/cqlsh
```

2) Switch to the killrvideo keyspace via the USE command:

```
USE killrvideo;
```

3) Execute the following command to view the metadata for the `videos` table you created earlier.

```
DESCRIBE TABLE videos;
```

- What is the partition key?

```
video_id
```

- How many partitions are in this table?

One for each unique primary key value.

4) Execute the following query to view the partitioner token value for each video id.

```
SELECT token(video_id), video_id
FROM videos;
```

5) Exit `cqlsh` and use the following command to inspect the file named `/home/ubuntu/labwork/data-files/videos-by-tag.csv`:

```
cat /home/ubuntu/labwork/data-files/videos-by-tag.csv
```

NOTE: Notice this CSV file categorizes the videos by one of two tags: `cassandra` or `datastax`.

6) Restart `cqlsh` and switch to the `killrvideo` keyspace.

7) Your mission, should you choose to accept it, is to write a `CREATE TABLE` statement that will store this data partitioned by tags. With this given data set, there should be two partitions, one for each tag. Call your table `videos_by_tag`.

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

8) Execute the following COPY command to import the videos-by-tag.csv data.

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

9) Verify CQL imported your data correctly by writing a `SELECT *` command.

```
SELECT *
FROM videos_by_tag;
```

- Note that if the table only contains 2 records, the primary key may be wrong - containing only the tag field - be sure to include the `video_id` as a cluster column.

10) Write a SELECT statement to retrieve all rows tagged with `cassandra`.

```
SELECT *
FROM videos_by_tag
WHERE tag = 'cassandra';
```

11) Now, find all videos tagged with `datastax` (similar to the previous query).

```
SELECT *
FROM videos_by_tag
WHERE tag = 'datastax';
```

12) Finally, write a query to retrieve the video having a title of `Cassandra Intro`.

```
SELECT *
FROM videos_by_tag
WHERE title = 'Cassandra Intro';
```

NOTE: Notice your query errors out. Apache Cassandra™ only allows queries on the partition key (and clustering columns shown in the next section). Since `title` is not the partition key, Apache Cassandra™ fails the query. If Apache Cassandra™ allowed querying on non-partition key columns, Apache Cassandra™ would have to scan all partitions on all nodes to produce a result set (which goes against the reason you would use Apache Cassandra™ in the first place).