

Exercise 01.02: CQL Fundamentals

- Create a Keyspace for KillrVideo
- Create a table to store video metadata
- Load the data for the video table from a CSV file

Background

Welcome to the KillrVideo company! KillrVideo hired you to build the latest and greatest video sharing application on the Internet. Your task is to ramp up the domain and become acquainted with the DataStax Distributions of Apache Cassandra™. To start, you decided to look into creating a table schema and to load some video data.

The video metadata is made up of:

Column Name	Data Type
video_id	timeuuid
added_date	timestamp
description	text
title	text
user_id	uuid

Steps

1. If necessary, SSH to cloud environment provided by your instructor.
2. Open the file `/home/ubuntu/labwork/cql/videos.csv` with a text editor and review the contents of the file.

IMPORTANT: Notice the order of the columns matches the order shown above.

3. Navigate to the `/home/ubuntu/labwork/cql` directory within the IDE.

```
cd /home/ubuntu/labwork/cql
```

4. Within the terminal prompt, start `cqlsh`.
5. In `cqlsh`, create a keyspace called `killrvideo` and switch to that keyspace.
 - a. Use the IF NOT EXISTS option to ensure this is a clean keyspace.

- b. Use the option *SimpleStrategy* for the replication class with a replication factor of one (1).
- c. Remember the `USE` command switches keyspaces.

NOTE: You can press the tab key within the CREATE KEYSPACE command to have `cqlsh` autocomplete the replication parameters.

6. Create a single table schema called *videos* with the same structure as shown in table above. The primary key is *video_id*.
7. Load the newly created table with the file *videos.csv* using the COPY command.

```
cqlsh:killrvideo> COPY videos  
(video_id,added_date,description,title,user_id) FROM 'videos.csv' WITH  
HEADER=true;
```

8. Use SELECT to verify the data loaded correctly. Include LIMIT to retrieve only the first 10 rows.
9. Use the command SELECT to COUNT(*) the number of imported rows. It should match the number of rows COPY reported as imported.
10. Use SELECT to find a row where the `video_id = 6c4cffb9-0dc4-1d59-af24-c960b5fc3652`.

Next, we will explore some other CQL commands that will come in handy. The command TRUNCATE will be explored in a later exercise, but here we will show you how to add/remove (non-primary key) columns.

11. Remove the data from our table using TRUNCATE. Do not delete or truncate the keyspace, but the 'videos' table only.

```
cqlsh:killrvideo> truncate videos;
```

12. Confirm the table 'videos' is now empty. What would the syntax be here?
13. Exit `cqlsh`.

END OF EXERCISE