BACKGROUND

Welcome to the KillrVideo project! On KillrVideo project, you will build the latest and greatest video sharing application on the Internet.
Your task is to ramp up on the domain and become acquainted with 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 '/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. cd /home/Ubuntu/labwork/cql
4. At the prompt, start 'cqlsh'.
5. In 'cqlsh', create a keyspace called 'killrvideo' and switch to that keyspace. Use 'SimpleStrategy' for the replication class with a replication factor of one. 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 called 'videos' with the same structure as shown in table above. 'video_id' is the primary key.
7. Load the newly created table with the 'videos.csv' file using the 'COPY' command.
COPY videos FROM 'videos.csv' WITH HEADER=true;
NOTE: Notice COPY does not require column names when the target table schema and source CSV file columns match respectively. We will see more on table schema later.
8. Use SELECT to verify the data loaded correctly. Include LIMIT to retrieve only the first 10 rows.
9. Use 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, like TRUNCATE in a later exercise, we will show you how to add/remove (non-primary key) columns.
11. Let's remove the data from our table using TRUNCATE .truncate videos;
12. Create a second table in the 'killrvideo' keyspace called 'videos_by_title_year' with the structure shown in above table. Be sure users can query this table on both 'title' and 'added_year' by combining them into the partition key.
13. Load the data from the 'videos_by_title_year.csv' file using the `COPY` command.
14. COPY videos_by_title_year FROM 'videos_by_title_year.csv' WITH HEADER=true;
15. Try running queries on the 'videos_by_title_year' table to query on a specific 'title' and 'added_year'.
16. What error does Cassandra return when you try to query on just title or just year? Why?
17. Create a table with the columns above to facilitate querying for videos by tag within a given year range returning the results in descending order by year.
18. We wrote most of the CREATE TABLE for you. Fill in the PRIMARY KEY and CLUSTERING ORDER BY.

```
CREATE TABLE videos_by_tag_year (
tag text,
added_year int,
video_id timeuuid,
added_date timestamp,
description text,
title text,
```

```
user_id uuid,
PRIMARY KEY ( )
) WITH CLUSTERING ORDER BY ( );
```

19. Load the data from the 'videos_by_tag_year.csv' file in the provided 'exercise=4' directory using the COPY command.
COPY videos_by_tag_year FROM 'videos_by_tag_year.csv' WITH HEADER=true;
20. Check the number of rows in the 'videos_by_tag_year' table.
NOTE: The number of rows should match the number of rows imported by the COPY command.
If not, you had upserts again and will need to adjust your PRIMARY KEY. Ask your instructor for help if necessary.
21. Try running queries on the 'videos_by_tag_year' table to query on a specific tag and added year.

Example queries:

Tag	Added_year
trailer	2015
cql	2014
spark	2014

22. Try querying for all videos with tag "cql" added before the year 2015. Notice you can do range queries on clustering columns.
23. Try querying for all videos added before 2015. The query will fail. What error message does cqlsh report? Why did the query fail whereas the previous query worked?
24. At the prompt, navigate to '/home/ubuntu/labwork/udts'. Launch 'cqlsh' and switch to the 'killrvideo' keyspace.
25. Run the TRUNCATE command to erase the data from the 'videos' table.
26. Alter the 'videos' table to add a 'tags' column.
27. Load the data from the 'videos.csv' file using the COPY command. COPY videos FROM 'videos.csv' WITH HEADER=true;
Remember, we do not need to create the user defined type called 'video_encoding' because we did so in the previous exercise. However, take a look at the code below as a refresher. Do not run it again or you will get an error!

```
CREATE TYPE video_encoding (
bit_rates SET<TEXT>,
encoding TEXT,
height INT,
width INT,
);
```

28. Alter your table to add an 'encoding' column of the 'video_encoding' type.
29. Load the data from the 'videos_encoding.csv' file using the COPY command.
COPY videos (video_id, encoding) FROM 'videos_encoding.csv' WITH HEADER=true;
30. Run a query to retrieve the first 10 rows of the 'videos' table. Notice the altered table contains data for the new 'tags' and 'encoding' column.
31. Exit cqlsh