

Exercise 03.04: Finalizing Physical Data Modeling

In this exercise, you will:

- Add data types to the physical data model
- Run the CQL `CREATE TABLE` statements for each table in physical model
- Load data and run some queries to test the physical data model

Background

In this exercise, you will build on your logical data model and finalize your physical data model. Not all your tables have data types for columns.

You will also need to build and run the `CREATE TABLE` statements for each of the tables, and run some queries to verify that everything is working correctly.

Steps

1. Use a text editor to open and review the *killrvideo.cql* file in the *labwork/final/* directory.
2. There are several tables with columns marked with **CQL Type**. Fill in the appropriate data type for the columns in 'users_by_email,' 'users,' 'videos_by_user,' and 'comments_by_user.'
3. In the *labwork/final/* directory, startup `cqlsh`.
4. Run the `SOURCE` command on the *killrvideo.cql* file to execute the `CREATE TABLE` statements.

```
CREATE TABLE users_by_email (  
    email TEXT,  
    password TEXT,  
    user_id UUID,  
    PRIMARY KEY ((email))  
);  
  
CREATE TABLE users (  
    user_id UUID,  
    email TEXT,  
    first_name TEXT,  
    last_name TEXT,  
    registration_date TIMESTAMP,
```

```

    PRIMARY KEY((user_id))
);

CREATE TABLE videos_by_user (
  user_id UUID,
  video_id TIMEUUID,
  title TEXT,
  type TEXT,
  tags SET<TEXT>,
  preview_thumbnails MAP<INT,BLOB>,
  PRIMARY KEY ((user_id), video_id))
WITH CLUSTERING ORDER BY (video_id DESC);

CREATE TABLE comments_by_user (
  user_id UUID,
  posted_timestamp TIMESTAMP,
  video_id TIMEUUID,
  comment TEXT,
  title TEXT,
  type TEXT,
  tags SET<TEXT>,
  preview_thumbnails MAP<INT,BLOB>,
  PRIMARY KEY ((user_id), posted_timestamp, video_id))
WITH CLUSTERING ORDER BY (posted_timestamp DESC, video_id ASC);

```

If the SOURCE command was successful, you should now have a new keyspace called 'killr_video.'

5. Run the DESCRIBE KEYSPACE command on the 'killr_video' keyspace to review the keyspace and table schema.
6. Set the default keyspace to 'killr_video' with the USE command.

```

SOURCE 'killrvideo.cql';

DESCRIBE KEYSPACE killr_video;

USE killr_video;

```

7. Load video data into your new keyspace. Use the COPY command to load data from the files into the following tables:

```

COPY videos FROM 'videos.csv' WITH HEADER=true; COPY latest_videos FROM
'latest_videos.csv' WITH HEADER=true; COPY trailers_by_video FROM
'trailers_by_video.csv' WITH HEADER=true; COPY actors_by_video FROM
'actors_by_video.csv' WITH HEADER=true;

```

If everything is successful, you should now have video data available for reading.

8. Query the 'latest_videos' table to find the most recent 50 videos that were uploaded.

- Is there a video uploaded for the movie *Gone Girl*? What is the 'video_id' for that particular movie?
9. Let's find out some more information about this movie. Query the 'videos' table using the previously found 'video_id.'
- When was this movie released? What are the genres for this movie?
10. We can also find the actors that were in the movie and the characters they played. Go ahead and query the 'actors_by_video' table using the 'video_id' for *Gone Girl*.
- Who was the actor that played the character Desi Collings?
11. At this point, an interested user might want to watch a trailer for this movie. Query the 'trailers_by_video' table to check if there are any trailers available for this movie.
12. If there is a trailer available, make note of the 'trailer_id' and then query the 'videos' table again using the 'trailer_id' value as the equality condition for the 'video_id' column.
- What is the URL for the trailer?
13. Copy the URL into a web browser and enjoy!

END OF EXERCISE