# **IST769 Homework Submission**

Name: **Sathish Kumar Rajendiran**   
SUID: **666555028**  
Email: srajendi@syr.edu  
Due Date: **09/07/2021**

Task: **Cassandra and CQL**

Homework #:**9**

**Exercise(s):**

1. Design your own scenario for which a Cassandra table would be a good solution. Make sure to explain the scenario and the specific characteristics of the scenario which would make Cassandra a good fit. Make sure to follow a query first approach and justify how the partition and cluster keys should be setup

**Solution:**

-- Objective

Explore Cassandra’s data modeling concepts with an example.

-- Scenario: Analyzing YouTube Playlist.

I have 3 personal playlists created as private in YouTube. In this exercise, lets create a keyspace, table and materialized views to explore/query contents. Table to store information such as playlist id, track\_url, base\_url, category, language, title, year of release and composed by. So, facts like, number of videos posted by category, year released, available contents can be retrieved. In addition, several other attributes can also be added to perform additional analytics.

Keyspace: Youtube\_analytics

Table: private\_playlist

**Evidence:**

**Graphical user interface, application

Description automatically generated**

Graphical user interface, text

Description automatically generated

1. Create your Cassandra table in CQL based on your scenario from the previous exercise. You should define the columns and data types to suit your scenario in addition to configuring the partition and cluster keys.

**Solution:**

-- CQL to create keyspace

CREATE KEYSPACE youtube\_analytics with replication = {‘class’: ’SimpleStrategy’, ‘replication\_factor’:’3};

-- Choose keyspace

use youtube\_analytics;

-- CQL to create table

CREATE TABLE private\_playlist (playlist\_id int, category text, language text, base\_url text, track\_url text, year int, title text, composed\_by text, PRIMARY KEY (playlist\_id, track\_url));

-- CQL to select data from table

SELECT \* FROM private\_playlist

**Evidence:**

Text

Description automatically generated

**Text

Description automatically generated**

1. Write CQL statements to add data to your table. Add at least 9 records consisting of 3 different partition and cluster keys

**Solution:**

-- CQL to Insert data

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (1, ‘J2Bh68GTUOU’, ‘PLzHz-2xygVFmqbXQiSXfXEGzT0Hnexbh6’, ’bollywood’, ’pritam’, ’hindi’, ’crazy kiya re’, 2006);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (1, ‘Jbn39j-xa-k’, ‘PLzHz-2xygVFmqbXQiSXfXEGzT0Hnexbh6’, ’bollywood’, ‘devdas’, ’hindi’, ‘dola re dola’, 2002);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (1, ‘C8kSrkz8Hz8’, ‘PLzHz-2xygVFmqbXQiSXfXEGzT0Hnexbh6’, ’bollywood’, ‘pritam, ’hindi’, ‘kamli’, 2013);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (1, ‘4dsFQFCvVGU’, ‘PLzHz-2xygVFmqbXQiSXfXEGzT0Hnexbh6’, ’bollywood’, ‘shankar’, ’hindi’, ‘kajra re’, 2005);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (2, ‘bXrvHkbByik’, ‘PLnVYEpTNGNtXTmmcpa60hHL-LT4Hynoss’, ‘statistics’, ‘mathtutordvd’, ‘english’, ‘intro to statistics’, 2016);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (2, ‘A8fBm7-tPXo’, ‘PLnVYEpTNGNtXTmmcpa60hHL-LT4Hynoss’, ‘statistics’, ‘mathtutordvd’, ‘english’, ‘what is a population in statistics’, 2016);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (2, ‘OqoWtOvD8w0’, ‘PLnVYEpTNGNtXTmmcpa60hHL-LT4Hynoss’, ‘statistics’, ‘mathtutordvd’, ‘english’, ‘descriptive vs inferential statistics’, 2016);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (2, ‘uRfRH01UyjU’, ‘PLnVYEpTNGNtXTmmcpa60hHL-LT4Hynoss’, ‘statistics’, ‘mathtutordvd’, ‘english’, ‘statistics definitions’, 2016);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (3, ‘y03oYWDLu0Q’, ‘PLTd6ceoshpreIS\_2qHSfKe-iBwW-6eltD’, ‘database normalization’, ‘computer science’, ‘english’, ‘introduction’, 2020);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (3, ‘jgUeOjImOOw’, ‘PLTd6ceoshpreIS\_2qHSfKe-iBwW-6eltD’, ‘database normalization’, ‘computer science’, ‘english’, ‘first normal form’, 2020);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (3, ‘9L10Q1nAfyg’, ‘PLTd6ceoshpreIS\_2qHSfKe-iBwW-6eltD’, ‘database normalization’, ‘computer science’, ‘english’, ‘second normal form’, 2020);

INSERT INTO private\_playlist (playlist\_id, track\_url, base\_url, category, composed\_by, language, title, year) VALUES (3, ‘\_K7fcFQowy8’, ‘PLTd6ceoshpreIS\_2qHSfKe-iBwW-6eltD’, ‘database normalization’, ‘computer science’, ‘english’, ‘third normal form’, 2020);

**Evidence:**

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

1. Write a CQL statement to create an index or materialized view on your table so that you can set a different partition key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.

**Solution:**

-- create an index on title and verify ALLOW FILTERING behavior

CREATE INDEX ix\_private\_playlist on youtube\_analytics.private\_playlist(title);

-- retrieve data from table

select \* from private\_playlist where playlist\_id = 1 and title =’kamli’ ALLOW FILTERING’;

select \* from private\_playlist where playlist\_id = 1 and title =’kamli’;

select \* from private\_playlist where title =’kamli’;

**Evidence:**

Text

Description automatically generated

**Text

Description automatically generated**

1. Write a CQL statement to create an index or materialized view on your table so that you can set a different cluster key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.

**Solution:**

-- create a materialized view and verify ALLOW FILTERING behavior

DROP MATERIALIZED VIEW IF EXISTS private\_playlist\_by\_category;

CREATE MATERIALIZED VIEW private\_playlist\_by\_category as SELECT \* FROM private\_playlist WHERE playlist\_id IS NOT NULL and track\_url IS NOT NULL and category IS NOT NULL PRIMARY KEY (playlist\_id, track\_url, category);

-- retrieve values

select \* from private\_playlist\_by\_category;

select \* from private\_playlist\_by\_category where playlist\_id =2;

select \* from private\_playlist\_by\_category where playlist\_id =2 and track\_url = ‘uRfRH01UyjU’;

**Evidence:**

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

# **Appendix**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Text

Description automatically generated**

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, Teams

Description automatically generated

Graphical user interface, application

Description automatically generated

Rectangle

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated