Part2: OLTP Queries

Group Members:

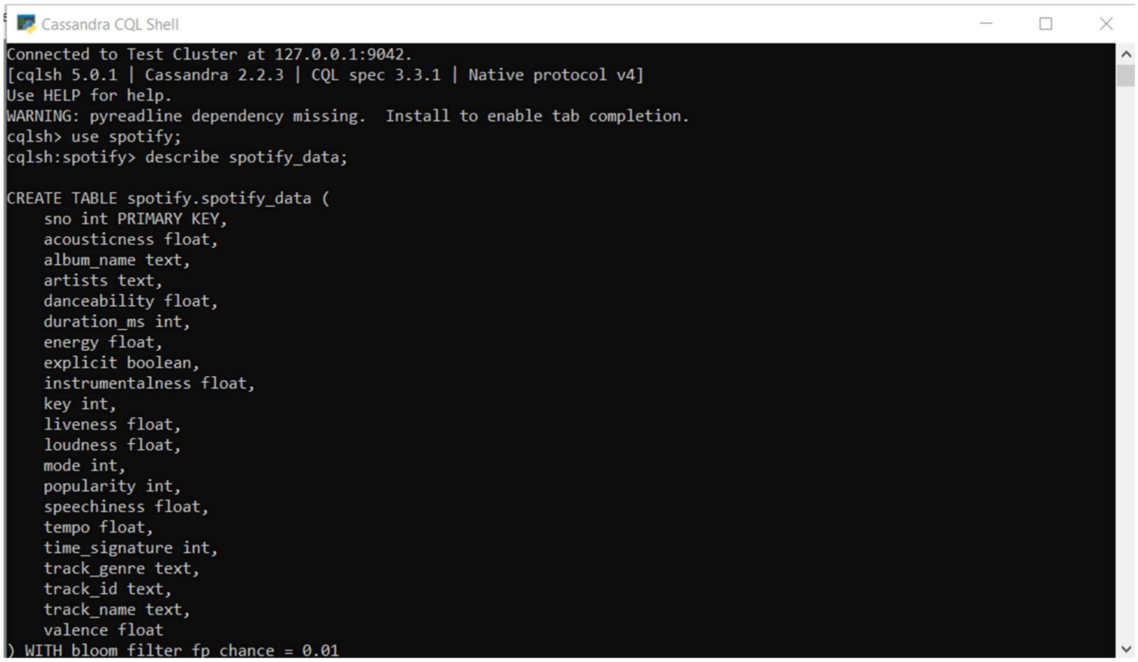
Nirdosh Mishra: 2022OG04021

Rishabh Mishra: 2022OG04039

Loom Video link:

https://www.loom.com/share/90b3a3e65d90481a9fa3a0c861b0d311?sid=315c34b5-e39c-463c-a590-fcea729d5be4

Database used: Cassandra cqlsh



```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.2.3 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> use spotify;
cqlsh:spotify> describe spotify_data;

CREATE TABLE spotify.spotify_data (
    sno int PRIMARY KEY,
    acousticness float,
    album_name text,
    artists text,
    danceability float,
    duration_ms int,
    energy float,
    explicit boolean,
    instrumentalness float,
    key int,
    liveness float,
    loudness float,
    mode int,
    popularity int,
    speechiness float,
    tempo float,
    time_signature int,
    track_genre text,
    track_id text,
    track_name text,
    valence float
) WITH bloom_filter_fp_chance = 0.01
```

# Created keyspace: spotify

create keyspace spotify with replication = {'class': 'SimpleStrategy', 'replication_factor': 1 };

select * from system_schema.keyspaces;

use spotify;

# Created table structure

create table spotify_data (sno int PRIMARY KEY,track_id text ,artists text,album_name text,track_name text,popularity int,duration_ms int,explicit boolean,danceability float,energy float,key int,loudness float,mode int,speechiness float,acousticness float,instrumentalness float,liveness float,valence float,tempo float,time_signature int,track_genre text);

create index pop on spotify_data(popularity);

# Import train data as mentioned in the Spotify assignment.

```
copy spotify_data (sno,track_id,artists,album_name,track_name,popularity,duration_ms,explicit,danceability,energy,key,loudness,mode,speechiness,acousticness,instrumentalness,liveness,valence,tempo,time_signature,track_genre) from 'D:\M.Tech BITS Pilani\Semester 3\Big Data Systems (S1-23_DSEOGZG522)\Assignment\Spotify\train.csv' with header = TRUE;
```

# Created Index to retrieve records from secondary columns.

```
CREATE INDEX IF NOT EXISTS popularity_index ON spotify_data (popularity);
```

# Find records based on specific popularity

```
SELECT track_id,artists,popularity FROM spotify.spotify_data where popularity > 90;
```

# Insert record into spotify_data

```
insert into spotify_data(sno,track_id,artists,album_name,track_name,popularity,duration_ms,explicit,danceability,energy,key,loudness,mode,speechiness,acousticness,instrumentalness,liveness,valence,tempo,time_signature,track_genre) values (114001,'test111','test111','test111','test111',90,32131,false,0.324,0548,111,4.548,1,0.568,0.235,0,07854,0.561,57.5,5,'test111');
```

```
select * from spotify_data where sno = 114001;
```

# Modify record based on spefic criteria

```
select album_name, popularity from spotify_data where album_name = 'Mega Hits Autumn/Fall 2022'
```

```
update spotify_data set popularity = 90 where album_name = 'Mega Hits Autumn/Fall 2022';
```

# Delete record based on specific record

# Deleted the record:

```
delete from spotify_data where sno = 114000;
```

# Verify if deleted.

```
select * from  spotify_data where sno = 114000;
```