Music App Database Management System Development

CPS 510-011

[redacted]

TABLE OF CONTENTS

ONLINE MUSIC APPLICATION DBMS (SPOTIFY)

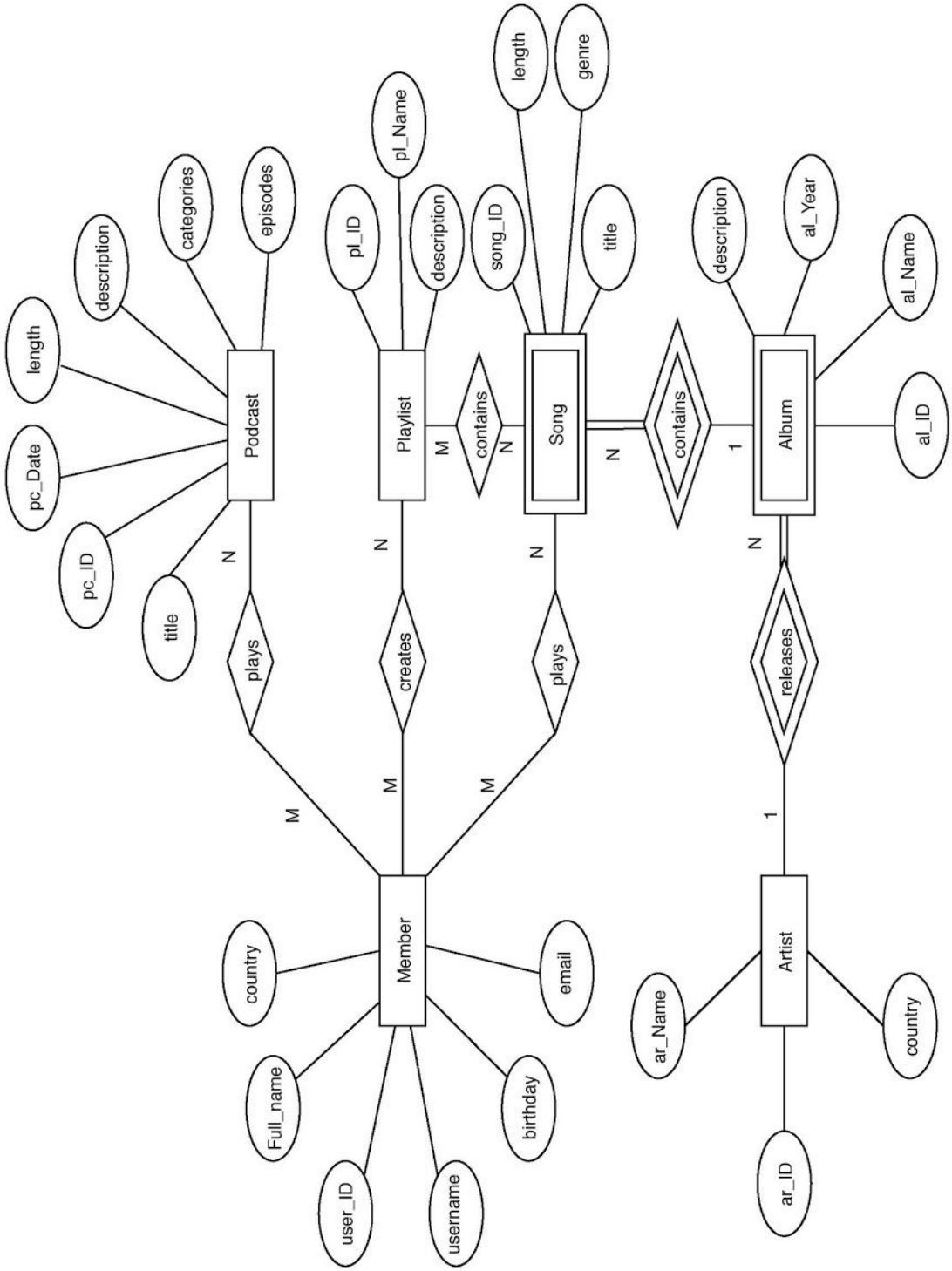
Spotify is a popular audio streaming platform used globally by people of all ages. The application that launched in 2008, provides account users with access to over 50 million music tracks, as well as podcasts, for free. However, Spotify users can also upgrade their account by switching to the paid subscription option to help further improve their streaming quality (i.e. no ads). The audio streaming platform's availability and accessibility over a wide range of devices has only added to its popularity, resulting in 232 million monthly active users as of July 2019.

Spotify allows its users to be able to listen to the kind of music they want to hear, when they want to hear it. For easy access, music tracks are grouped in certain categories, such as artist, genre, and mood. Users may also create their own playlists of songs, and even choose to share them with the public, or keep them private. Spotify also tries to provide the users with song recommendations and curated playlists based on the user's listening history. Premium (or paying) users of Spotify acquire special access throughout the application, like downloading music to listen off-line, skipping unlimited songs, and subscribing to their favorite artists.

To access Spotify, the user must first enter a unique username and password to create an account. They must then enter their name, email and country to ensure their identity, as well as receive an email verifying their login credentials. The users must also choose whether they would like to access Spotify using the Premium (paying) plan or the basic plan (free).These information can be seen by the user under their account profile. Once the account has been created, the user can search for songs to listen or add to their playlist, by searching through Spotify's database either by name, album, artist, genre or mood. Each artist has a name and monthly listener recorded as a part of their entry. Likewise, each album has a title, year, genre, and description, and each song has a title and song length recorded. That being said, each user playlist also has a name, and status (public/private).

The Spotify DBMS consists of the following entities: User, Playlist, Songs, Podcasts, Artist and Album. The aforementioned entities have the following relationships: Users play songs/podcasts and create playlists, playlists and albums contain songs while the artists release albums.

ER DIAGRAM


Podcast
length
pc_Date
pc_ID
description
title
categories
episodes
plays
N
M
Playlist
pl_ID
pl_Name
description
creates
N
M
contains
M
N
Song
song_ID
length
genre
title
plays
N
M
contains
N
1
Album
description
al_Year
al_Name
al_ID
releases
N
1
Member
country
Full_name
user_ID
username
birthday
email
Artist
ar_Name
ar_ID
country

SCHEMA DESIGN

```
DROP TABLE creates;
DROP TABLE releases;
DROP TABLE plays;
DROP TABLE listens_to;
DROP TABLE contains;
DROP TABLE has;

DROP TABLE members;
DROP TABLE artist;
DROP TABLE podcast;
DROP TABLE playlist;
DROP TABLE song;
DROP TABLE album;

CREATE TABLE members (
   User_id NUMBER PRIMARY KEY,
   Username VARCHAR2(20),
   Country VARCHAR2(30),
   Email VARCHAR2(30),
   Full_name VARCHAR(25),
   Birthday DATE
);

CREATE TABLE artist (
   Artist_id NUMBER PRIMARY KEY,
   Artist_name VARCHAR2(20) NOT NULL,
   Country VARCHAR2(30),
   Album_num NUMBER
);

CREATE TABLE podcast (
   Podcast_id NUMBER PRIMARY KEY,
   Title VARCHAR2(35),
   Podcast_date DATE,
   Podcast_length VARCHAR2(5),
   Categories VARCHAR2(25),
   Episodes VARCHAR2(50)
);
```

```
CREATE TABLE playlist (
   Playlist_id NUMBER PRIMARY KEY,
   Playlist_name VARCHAR2(25),
   Total_songs NUMBER
);

CREATE TABLE song (
   Song_id NUMBER PRIMARY KEY,
   Song_length VARCHAR2(5),
   Genre VARCHAR2(15),
   Title VARCHAR2(35)
);

CREATE TABLE album (
   Album_id NUMBER PRIMARY KEY,
   Album_name VARCHAR2(40) NOT NULL,
   Album_year DATE,
   Song_number NUMBER
);

INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(1232, '[redacted]', 'Indonesia', '[redacted]@gmail.com', '[redacted]', to_date('1999-09-08',
'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(2344, '[redacted]', 'India', '[redacted]@gmail.com', '[redacted]', to_date('1999-11-28',
'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(3459, '[redacted]', 'Canada', '[redacted]@gmail.com', '[redacted]',
to_date('1999-07-19', 'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(6923, 'army_1306', 'Canada', 'bts_ftw13@gmail.com', 'Bangtan Bangtan', to_date('2013-06-13',
'yyyy-mm-dd'));

INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (7852, 'Ariana
Grande', 'America', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (8916, 'BTS',
'Korea', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (3404, 'Halsey',
'America', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (0734, 'Shawn
Mendes', 'Canada', 1);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (7488, 'Troye
Sivan', 'Australia', 1);
```

```
INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories, Episodes) VALUES (3025, 'TED Talks Daily',  to_date('2019-09-13', 'yyyy-mm-dd'), '13:49', 'Environment', 'Can Seaweed Help Global Warming?');
INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories, Episodes) VALUES (8934, 'MISFITS', to_date('2019-08-15', 'yyyy-mm-dd'), '53:38', 'Comedy', 'The Poland Story');
INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories, Episodes) VALUES (2436, 'Uncover', to_date('2019-05-28', 'yyyy-mm-dd'), '29:45', 'True Crime', 'Behind Bars');

INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (7892, 'Party Mode', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (5409, 'Everyday Favs', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (8943, 'Chill Vibes', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (6781, 'All Time BTS', 6);

INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (4523, '2:59', 'RnB', '7 Rings');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (1459, '3.27', 'Pop', 'thank u, next');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (1431, '3:32', 'RnB', 'Imagine');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (9304, '3:17', 'Pop', 'One Last Time');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7584, '3:19', 'RnB', 'Bang Bang');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3542, '3:35', 'Pop', 'Break Free');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (0833, '3:50', 'Kpop', 'Boy With Luv');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3894, '3:44', 'Kpop', 'Mikrokosmos');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8905, '3:47', 'Kpop', 'Jamais Vu');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (6554, '3:57', 'Kpop', 'RUN');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3854, '3:59', 'Kpop', 'Butterfly');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (0983, '4:17', 'Kpop', 'Ma City');
```

```
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8123, '4:09', 'Electro Pop',
'Colors');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (2894, '3:04', 'Indie', 'New
Americana');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3409, '4:38', 'Electro Pop',
'Castle');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7895, '3:52', 'Metal',
'Nightmare');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3098, '3:22', 'Pop', 'Without
Me');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8767, '3:02', 'Indie',
'Graveyard');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (4909, '3:08', 'Pop', 'Treat
You Better');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (6877, '3:29', 'Pop', 'Mercy');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7659, '3:02', 'Indie', 'Youth');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8978, '3:54', 'Indie',
'Suburbia');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (9847, '3:57', 'Indie', 'Talk
Me Down');

INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9760,
'thank u, next', to_date('2019-02-08', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9786,
'My Everything', to_date('2014-08-25', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (4562,
'Map of The Soul: Persona', to_date('2019-04-12', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (2386,
'The Most Beautiful Moment in Life Pt.2', to_date('2015-11-30', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0927,
'BADLANDS', to_date('2015-08-28', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0789,
'Single', to_date('2019-09-13', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9867,
'Illuminate', to_date('2016-09-23', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0892,
'Blue Neighbourhood', to_date('2015-12-04', 'yyyy-mm-dd'), 3);

CREATE TABLE creates (
   User_id NUMBER REFERENCES members(User_id),
   Playlist_id NUMBER REFERENCES playlist(Playlist_id),
   PRIMARY KEY (User_id, Playlist_id));
```

```
CREATE TABLE releases (
   Artist_id NUMBER REFERENCES artist(Artist_id),
   Album_id NUMBER REFERENCES album(Album_id),
   PRIMARY KEY (Artist_id, Album_id));

CREATE TABLE plays (
   User_id NUMBER REFERENCES members(User_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (User_id, Song_id));

CREATE TABLE listens_to (
   User_id NUMBER REFERENCES members(User_id),
   Podcast_id NUMBER REFERENCES podcast(Podcast_id),
   PRIMARY KEY (User_id, Podcast_id));

CREATE TABLE contains (
   Album_id NUMBER REFERENCES album(Album_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (Album_id, Song_id));

CREATE TABLE has (
   Playlist_id NUMBER REFERENCES playlist(Playlist_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (Playlist_id, Song_id));

INSERT INTO creates (User_id, Playlist_id) VALUES (1232, 7892);
INSERT INTO creates (User_id, Playlist_id) VALUES (2344, 5409);
INSERT INTO creates (User_id, Playlist_id) VALUES (3459, 8943);
INSERT INTO creates (User_id, Playlist_id) VALUES (6923, 6781);

INSERT INTO releases (Artist_id, Album_id) VALUES (7852, 9760);
INSERT INTO releases (Artist_id, Album_id) VALUES (7852, 9786);
INSERT INTO releases (Artist_id, Album_id) VALUES (8916, 4562);
INSERT INTO releases (Artist_id, Album_id) VALUES (8916, 2386);
INSERT INTO releases (Artist_id, Album_id) VALUES (3404, 0927);
INSERT INTO releases (Artist_id, Album_id) VALUES (3404, 0789);
INSERT INTO releases (Artist_id, Album_id) VALUES (0734, 9867);
INSERT INTO releases (Artist_id, Album_id) VALUES (7488, 0892);

INSERT INTO plays (User_id, Song_id) VALUES (1232, 4523);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 3542);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 3894);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 6554);
```

```
INSERT INTO plays (User_id, Song_id) VALUES (1232, 0983);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 8123);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 8767);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 4909);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 1431);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 3542);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 0833);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 8905);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 3854);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 2894);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 6877);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 8978);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 1459);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 9304);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 3409);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 7895);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 3098);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 7659);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 9847);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 0833);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 3894);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 8905);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 6554);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 3854);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 0983);

INSERT INTO listens_to(User_id, Podcast_id) VALUES(1232, 3025);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(2344, 8934);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(2344, 3025);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(3459, 2436);

INSERT INTO contains (Album_id, Song_id) VALUES (9760, 4523);
INSERT INTO contains (Album_id, Song_id) VALUES (9760, 1459);
INSERT INTO contains (Album_id, Song_id) VALUES (9760, 1431);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 9304);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 7584);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 3542);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 0833);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 3894);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 8905);
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 6554);
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 3854);
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 0983);
```

```
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 8123);
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 2894);
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 3409);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 7895);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 3098);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 8767);
INSERT INTO contains (Album_id, Song_id) VALUES (9867, 4909);
INSERT INTO contains (Album_id, Song_id) VALUES (9867, 6877);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 7659);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 8978);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 9847);

INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 4523);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 3542);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 3894);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 6554);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 0983);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 8123);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 8767);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 4909);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 1431);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 3542);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 0833);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 8905);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 3854);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 2894);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 6877);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 8978);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 1459);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 9304);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 3409);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 7895);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 3098);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 7659);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 9847);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 0833);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 3894);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 8905);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 6554);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 3854);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 0983);
```

DESIGNING SIMPLE AND ADVANCED QUERIES

SIMPLE QUERIES

1. SELECT *
   FROM members;

2. SELECT *
   FROM artist
   WHERE Album_num > 1;

	ARTIST_ID	ARTIST_NAME	COUNTRY	ALBUM_NUM
1	7852	Ariana Grande	America	2
2	8916	BTS	Korea	2
3	3404	Halsey	America	2

3. SELECT Episodes AS New_Podcasts, Podcast_date
   FROM podcast
   WHERE Podcast_date > to_date('2019-06-01', 'yyyy-mm-dd');

	NEW_PODCASTS	PODCAST_DATE
1	Can Seaweed Help Global Warming?	13-SEP-19
2	The Poland Story	15-AUG-19

4. SELECT Playlist_name ,Total_songs AS Number_of_Songs_in_Playlist
   FROM playlist
   WHERE Total_songs = 8;

	PLAYLIST_NAME	NUMBER_OF_SONGS_IN_PLAYLIST
1	Party Mode	8
2	Everyday Favs	8
3	Chill Vibes	8

5. SELECT Title AS Indie_Songs, Song_length
   FROM song
   WHERE Genre = 'Indie';

	INDIE_SONGS	SONG_LENGTH
1	New Americana	3:04
2	Graveyard	3:02
3	Youth	3:02
4	Suburbia	3:54
5	Talk Me Down	3:57

6. SELECT Album_name AS Album_2015, Album_year
   FROM album
   WHERE Album_year > to_date('2015-01-01', 'yyyy-mm-dd') AND Album_year < to_date('2016-01-01', 'yyyy-mm-dd');

	ALBUM_2015	ALBUM_YEAR
1	The Most Beautiful Moment in Life Pt.2	30-NOV-15
2	BADLANDS	28-AUG-15
3	Blue Neighbourhood	04-DEC-15

7. SELECT *
   FROM podcast
   WHERE Categories = 'True Crime' ;

	PODCAST_ID	TITLE	PODCAST_DATE	PODCAST_LENGTH	CATEGORIES	EPISODES
1	2436	Uncover	28-MAY-19	29:45	True Crime	Behind Bars

ADVANCED QUERIES

1. SELECT Title, 'in Playlist:', Playlist_name
   FROM song s, has h, playlist p
   WHERE s.Song_id = h.Song_id
   AND h.Playlist_id = p.Playlist_id
   AND Playlist_name = 'Chill Vibes';

	TITLE	'INPLAYLIST:'	PLAYLIST_NAME
1	thank u, next	in Playlist:	Chill Vibes
2	One Last Time	in Playlist:	Chill Vibes
3	Castle	in Playlist:	Chill Vibes
4	Nightmare	in Playlist:	Chill Vibes
5	Without Me	in Playlist:	Chill Vibes
6	Youth	in Playlist:	Chill Vibes
7	Talk Me Down	in Playlist:	Chill Vibes

2. SELECT Title, 'is by' , Artist_name
   FROM song s, album a, artist b, releases r, contains c
   WHERE s.Song_id = c.Song_id
   AND c.Album_id = a.Album_id
   AND a.Album_id = r.Album_id
   AND r.Artist_id = b.Artist_id
   AND Artist_name = 'BTS' ;

	TITLE	'ISBY'	ARTIST_NAME
1	RUN	is by	BTS
2	Butterfly	is by	BTS
3	Ma City	is by	BTS
4	Boy With Luv	is by	BTS
5	Mikrokosmos	is by	BTS
6	Jamais Vu	is by	BTS

3. CREATE VIEW sub_playlist AS
   (SELECT * FROM album
   WHERE Album_name LIKE 'M%');
   SELECT Album_name, Album_year
   FROM sub_playlist
   WHERE Album_id = 4562;

	ALBUM_NAME	ALBUM_YEAR
1	Map of The Soul: Persona	12-APR-19

4. CREATE VIEW kpop_songs AS
   (SELECT *
   FROM song
   WHERE Genre = 'Kpop');
   SELECT *
   FROM kpop_songs
   WHERE Song_length LIKE '3%';

	SONG_ID	SONG_LENGTH	GENRE	TITLE
1	833	3:50	Kpop	Boy With Luv
2	3894	3:44	Kpop	Mikrokosmos
3	8905	3:47	Kpop	Jamais Vu
4	6554	3:57	Kpop	RUN
5	3854	3:59	Kpop	Butterfly

5. SELECT Playlist_name, COUNT(Song_id) AS Total_Songs_in_Playlist
   FROM playlist p, has h
   WHERE p.Playlist_id = h.Playlist_id
   GROUP BY Playlist_name;

	PLAYLIST_NAME	TOTAL_SONGS_IN_PLAYLIST
1	Party Mode	8
2	Everyday Favs	8
3	All Time BTS	6
4	Chill Vibes	8

6. (SELECT *
   FROM song)
   MINUS
   (SELECT s.*
   FROM song s, playlist p, has h
   WHERE p.Playlist_name = 'Party Mode'
   AND p.Playlist_id = h.Playlist_id
   AND h.Song_id = s.Song_id);

	SONG_ID	SONG_LENGTH	GENRE	TITLE
1	833	3:50	Kpop	Boy With Luv
2	1431	3:32	RnB	Imagine
3	1459	3.27	Pop	thank u, next
4	2894	3:04	Indie	New Americana
5	3098	3:22	Pop	Without Me
6	3409	4:38	Electro Pop	Castle
7	3854	3:59	Kpop	Butterfly
8	6877	3:29	Pop	Mercy
9	7584	3:19	RnB	Bang Bang
10	7659	3:02	Indie	Youth
11	7895	3:52	Metal	Nightmare
12	8905	3:47	Kpop	Jamais Vu
13	8978	3:54	Indie	Suburbia
14	9304	3:17	Pop	One Last Time
15	9847	3:57	Indie	Talk Me Down

UNIX SHELL IMPLEMENTATION

Create_tables.sh

```
#!/bin/sh
sqlplus64
"[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson
.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))"<<EOF

CREATE TABLE members (
   User_id NUMBER PRIMARY KEY,
   Username VARCHAR2(20),
   Country VARCHAR2(30),
   Email VARCHAR2(30),
   Full_name VARCHAR(25),
   Birthday DATE);

CREATE TABLE artist (
   Artist_id NUMBER PRIMARY KEY,
   Artist_name VARCHAR2(20) NOT NULL,
   Country VARCHAR2(30),
   Album_num NUMBER);

CREATE TABLE podcast (
   Podcast_id NUMBER PRIMARY KEY,
   Title VARCHAR2(35),
   Podcast_date DATE,
   Podcast_length VARCHAR2(5),
   Categories VARCHAR2(25),
   Episodes VARCHAR2(50));

CREATE TABLE playlist (
   Playlist_id NUMBER PRIMARY KEY,
   Playlist_name VARCHAR2(25),
   Total_songs NUMBER);

CREATE TABLE song (
   Song_id NUMBER PRIMARY KEY,
   Song_length VARCHAR2(5),
   Genre VARCHAR2(15),
   Title VARCHAR2(35));
```

```
CREATE TABLE album (
   Album_id NUMBER PRIMARY KEY,
   Album_name VARCHAR2(40) NOT NULL,
   Album_year DATE,
   Song_number NUMBER);

CREATE TABLE creates (
   User_id NUMBER REFERENCES members(User_id),
   Playlist_id NUMBER REFERENCES playlist(Playlist_id),
   PRIMARY KEY (User_id, Playlist_id));

CREATE TABLE releases (
   Artist_id NUMBER REFERENCES artist(Artist_id),
   Album_id NUMBER REFERENCES album(Album_id),
   PRIMARY KEY (Artist_id, Album_id));

CREATE TABLE plays (
   User_id NUMBER REFERENCES members(User_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (User_id, Song_id));

CREATE TABLE listens_to (
   User_id NUMBER REFERENCES members(User_id),
   Podcast_id NUMBER REFERENCES podcast(Podcast_id),
   PRIMARY KEY (User_id, Podcast_id));

CREATE TABLE contains (
   Album_id NUMBER REFERENCES album(Album_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (Album_id, Song_id));

CREATE TABLE has (
   Playlist_id NUMBER REFERENCES playlist(Playlist_id),
   Song_id NUMBER REFERENCES song(Song_id),
   PRIMARY KEY (Playlist_id, Song_id));


CREATE VIEW sub_playlist AS
   (SELECT *
   FROM album
   WHERE Album_name LIKE 'M%');
```

```
CREATE VIEW kpop_songs AS
   (SELECT *
   FROM song
   WHERE Genre = 'Kpop');

exit;
EOF
```

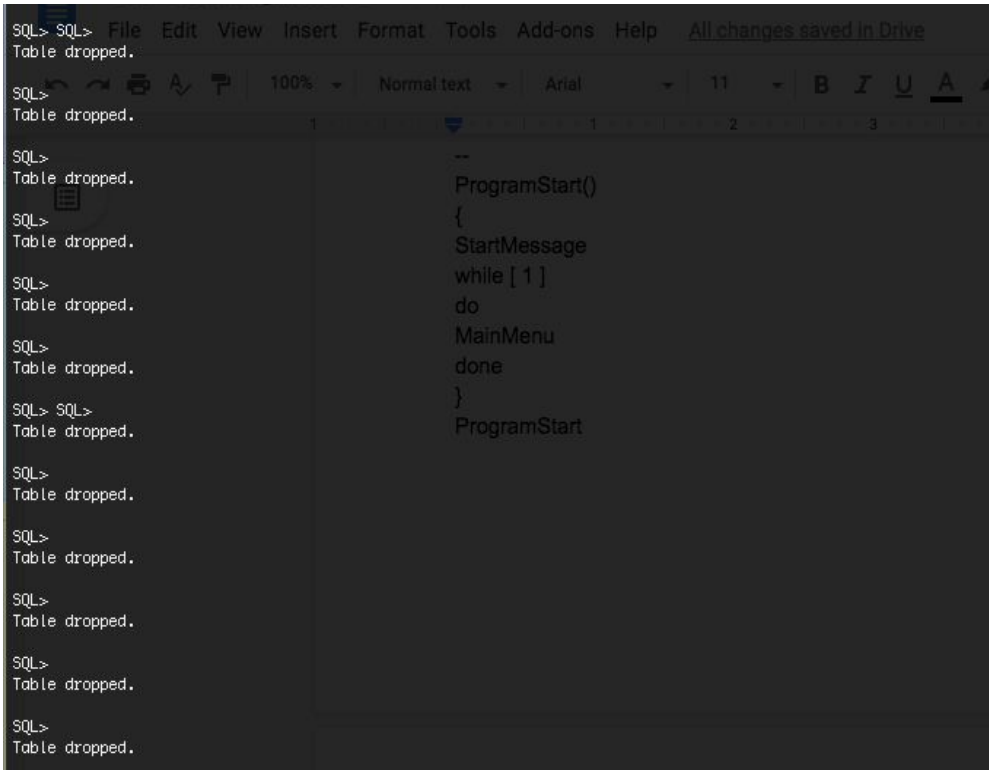
Output:

```
flevina@elara:~$ bash create_tables.sh

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 14 23:11:20 2019

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5
Table created.
```

Drop_tables.sh

```
#!/bin/sh
sqlplus64
"[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson
.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))"<<EOF

DROP TABLE creates;
DROP TABLE releases;
DROP TABLE plays;
DROP TABLE listens_to;
DROP TABLE contains;
DROP TABLE has;
DROP TABLE members;
DROP TABLE artist;
DROP TABLE podcast;
DROP TABLE playlist;
DROP TABLE song;
DROP TABLE album;
exit;
EOF
```

Output:



Populate_tables.sh

```
#!/bin/sh

sqlplus64
"[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson
.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))"<<EOF

INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(1232, '[redacted]', 'Indonesia', '[redacted]@gmail.com', '[redacted]', to_date('1999-09-08',
'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(2344, '[redacted]', 'India', '[redacted]@gmail.com', '[redacted]', to_date('1999-11-28',
'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(3459, '[redacted]', 'Canada', '[redacted]@gmail.com', '[redacted]',
to_date('1999-07-19', 'yyyy-mm-dd'));
INSERT INTO members (User_id, Username, Country, Email, Full_name, Birthday) VALUES
(6923, 'army_1306', 'Canada', 'bts_ftw13@gmail.com', 'Bangtan Bangtan', to_date('2013-06-13',
'yyyy-mm-dd'));

INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (7852, 'Ariana
Grande', 'America', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (8916, 'BTS',
'Korea', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (3404, 'Halsey',
'America', 2);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (0734, 'Shawn
Mendes', 'Canada', 1);
INSERT INTO artist (Artist_id, Artist_name, Country, Album_num) VALUES (7488, 'Troye
Sivan', 'Australia', 1);

INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories,
Episodes) VALUES (3025, 'TED Talks Daily',  to_date('2019-09-13', 'yyyy-mm-dd'), '13:49',
'Environment', 'Can Seaweed Help Global Warming?');
INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories,
Episodes) VALUES (8934, 'MISFITS', to_date('2019-08-15', 'yyyy-mm-dd'), '53:38', 'Comedy',
'The Poland Story');
INSERT INTO podcast (Podcast_id, Title, Podcast_date, Podcast_length, Categories,
Episodes) VALUES (2436, 'Uncover', to_date('2019-05-28', 'yyyy-mm-dd'), '29:45', 'True Crime',
'Behind Bars');
```

```
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (7892, 'Party Mode', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (5409, 'Everyday Favs', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (8943, 'Chill Vibes', 8);
INSERT INTO playlist (Playlist_id, Playlist_name, Total_Songs) VALUES (6781, 'All Time BTS', 6);

INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (4523, '2:59', 'RnB', '7 Rings');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (1459, '3.27', 'Pop', 'thank u, next');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (1431, '3:32', 'RnB', 'Imagine');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (9304, '3:17', 'Pop', 'One Last Time');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7584, '3:19', 'RnB', 'Bang Bang');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3542, '3:35', 'Pop', 'Break Free');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (0833, '3:50', 'Kpop', 'Boy With Luv');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3894, '3:44', 'Kpop', 'Mikrokosmos');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8905, '3:47', 'Kpop', 'Jamais Vu');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (6554, '3:57', 'Kpop', 'RUN');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3854, '3:59', 'Kpop', 'Butterfly');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (0983, '4:17', 'Kpop', 'Ma City');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8123, '4:09', 'Electro Pop', 'Colors');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (2894, '3:04', 'Indie', 'New Americana');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3409, '4:38', 'Electro Pop', 'Castle');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7895, '3:52', 'Metal', 'Nightmare');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (3098, '3:22', 'Pop', 'Without Me');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8767, '3:02', 'Indie', 'Graveyard');
```

INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (4909, '3:08', 'Pop', 'Treat You Better');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (4909, '3:08', 'Pop', 'Treat You Better');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (6877, '3:29', 'Pop', 'Mercy');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (7659, '3:02', 'Indie', 'Youth');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (8978, '3:54', 'Indie', 'Suburbia');
INSERT INTO song (Song_id, Song_length, Genre, Title) VALUES (9847, '3:57', 'Indie', 'Talk Me Down');

INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9760, 'thank u, next', to_date('2019-02-08', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9786, 'My Everything', to_date('2014-08-25', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (4562, 'Map of The Soul: Persona', to_date('2019-04-12', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (2386, 'The Most Beautiful Moment in Life Pt.2', to_date('2015-11-30', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0927, 'BADLANDS', to_date('2015-08-28', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0789, 'Single', to_date('2019-09-13', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (9867, 'Illuminate', to_date('2016-09-23', 'yyyy-mm-dd'), 3);
INSERT INTO album (Album_id, Album_name, Album_year, Song_number) VALUES (0892, 'Blue Neighbourhood', to_date('2015-12-04', 'yyyy-mm-dd'), 3);

INSERT INTO creates (User_id, Playlist_id) VALUES (1232, 7892);
INSERT INTO creates (User_id, Playlist_id) VALUES (2344, 5409);
INSERT INTO creates (User_id, Playlist_id) VALUES (3459, 8943);
INSERT INTO creates (User_id, Playlist_id) VALUES (6923, 6781);

INSERT INTO releases (Artist_id, Album_id) VALUES (7852, 9760);
INSERT INTO releases (Artist_id, Album_id) VALUES (7852, 9786);
INSERT INTO releases (Artist_id, Album_id) VALUES (8916, 4562);
INSERT INTO releases (Artist_id, Album_id) VALUES (8916, 2386);
INSERT INTO releases (Artist_id, Album_id) VALUES (3404, 0927);
INSERT INTO releases (Artist_id, Album_id) VALUES (3404, 0789);
INSERT INTO releases (Artist_id, Album_id) VALUES (0734, 9867);
INSERT INTO releases (Artist_id, Album_id) VALUES (7488, 0892);

INSERT INTO plays (User_id, Song_id) VALUES (1232, 4523);

```
INSERT INTO plays (User_id, Song_id) VALUES (1232, 3542);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 3894);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 6554);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 0983);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 8123);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 8767);
INSERT INTO plays (User_id, Song_id) VALUES (1232, 4909);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 1431);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 3542);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 0833);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 8905);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 3854);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 2894);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 6877);
INSERT INTO plays (User_id, Song_id) VALUES (2344, 8978);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 1459);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 9304);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 3409);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 7895);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 3098);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 7659);
INSERT INTO plays (User_id, Song_id) VALUES (3459, 9847);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 0833);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 3894);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 8905);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 6554);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 3854);
INSERT INTO plays (User_id, Song_id) VALUES (6923, 0983);

INSERT INTO listens_to(User_id, Podcast_id) VALUES(1232, 3025);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(2344, 8934);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(2344, 3025);
INSERT INTO listens_to(User_id, Podcast_id) VALUES(3459, 2436);

INSERT INTO contains (Album_id, Song_id) VALUES (9760, 4523);
INSERT INTO contains (Album_id, Song_id) VALUES (9760, 1459);
INSERT INTO contains (Album_id, Song_id) VALUES (9760, 1431);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 9304);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 7584);
INSERT INTO contains (Album_id, Song_id) VALUES (9786, 3542);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 0833);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 3894);
INSERT INTO contains (Album_id, Song_id) VALUES (4562, 8905);
```

```
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 6554);
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 3854);
INSERT INTO contains (Album_id, Song_id) VALUES (2386, 0983);
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 8123);
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 2894);
INSERT INTO contains (Album_id, Song_id) VALUES (0927, 3409);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 7895);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 3098);
INSERT INTO contains (Album_id, Song_id) VALUES (0789, 8767);
INSERT INTO contains (Album_id, Song_id) VALUES (9867, 4909);
INSERT INTO contains (Album_id, Song_id) VALUES (9867, 6877);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 7659);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 8978);
INSERT INTO contains (Album_id, Song_id) VALUES (0892, 9847);

INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 4523);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 3542);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 3894);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 6554);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 0983);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 8123);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 8767);
INSERT INTO has (Playlist_id, Song_id) VALUES (7892, 4909);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 1431);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 3542);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 0833);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 8905);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 3854);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 2894);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 6877);
INSERT INTO has (Playlist_id, Song_id) VALUES (5409, 8978);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 1459);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 9304);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 3409);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 7895);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 3098);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 7659);
INSERT INTO has (Playlist_id, Song_id) VALUES (8943, 9847);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 0833);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 3894);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 8905);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 6554);
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 3854);
```

```
INSERT INTO has (Playlist_id, Song_id) VALUES (6781, 0983);
exit;
EOF
```

Output:

```
flevina@europa:~$ bash populate_tables.sh
SQL*Plus: Release 12.1.0.2.0 Production on Thu Oct 24 11:20:01 2019
Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.
```

```
#!/bin/sh
MainMenu()
{
    while [ "$CHOICE" != "START"
    do
        clear
        echo
"============================
echo "|              Oracle All Inclus
|"
echo "|
Main Menu
-
Select Desired Operation(s):
|"
echo "|      <CTRL
```

Queries.sh

```
#!/bin/sh
sqlplus64
"[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson
.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))"<<EOF

SELECT Title, 'in Playlist:', Playlist_name
FROM song s, has h, playlist p
WHERE s.Song_id = h.Song_id
	AND h.Playlist_id = p.Playlist_id
	AND Playlist_name = 'Chill Vibes';

SELECT Title, 'is by' , Artist_name
FROM song s, album a, artist b, releases r, contains c
WHERE s.Song_id = c.Song_id
	AND c.Album_id = a.Album_id
	AND a.Album_id = r.Album_id
	AND r.Artist_id = b.Artist_id
	AND Artist_name = 'BTS' ;

SELECT Album_name, Album_year
FROM sub_playlist
WHERE Album_id = 4562;

SELECT *
FROM kpop_songs
WHERE Song_length LIKE '3%';

SELECT Playlist_name, COUNT(Song_id) AS Total_Songs_in_Playlist
FROM playlist p, has h
WHERE p.Playlist_id = h.Playlist_id
GROUP BY Playlist_name;

(SELECT *
FROM song)
MINUS
(SELECT s.*
FROM song s, playlist p, has h
WHERE p.Playlist_name = 'Party Mode'
	AND p.Playlist_id = h.Playlist_id
	AND h.Song_id = s.Song_id);
```

exit;
EOF

Output:

```
SQL> SQL>   2    3    4    5
TITLE                            'INPLAYLIST: PLAYLIST_NAME
---------------------------------- ------------ ------------------------
thank u, next                    in Playlist: Chill Vibes
One Last Time                    in Playlist: Chill Vibes
Castle                           in Playlist: Chill Vibes
Nightmare                        in Playlist: Chill Vibes
Without Me                       in Playlist: Chill Vibes
There is Nothing Holding Me Back in Playlist: Chill Vibes
Youth                            in Playlist: Chill Vibes
Talk Me Down                     in Playlist: Chill Vibes

8 rows selected.

SQL> SQL> SQL> SQL>   2    3    4    5    6    7
TITLE                            'ISBY ARTIST_NAME
---------------------------------- ----- --------------------
RUN                              is by BTS
Butterfly                        is by BTS
Ma City                          is by BTS
Boy With Luv                     is by BTS
Mikrokosmos                      is by BTS
Jamais Vu                        is by BTS

6 rows selected.

SQL> SQL> SQL>   2    3
ALBUM_NAME                                ALBUM_YEA
----------------------------------------- ---------
Map of The Soul: Persona                  12-APR-19

SQL> SQL> SQL> SQL>   2    3
   SONG_ID SONG_ GENRE           TITLE
---------- ----- --------------- ----------------------------------
       833 3:50  Kpop            Boy With Luv
      3894 3:44  Kpop            Mikrokosmos
      8905 3:47  Kpop            Jamais Vu
      6554 3:57  Kpop            RUN
      3854 3:59  Kpop            Butterfly
```

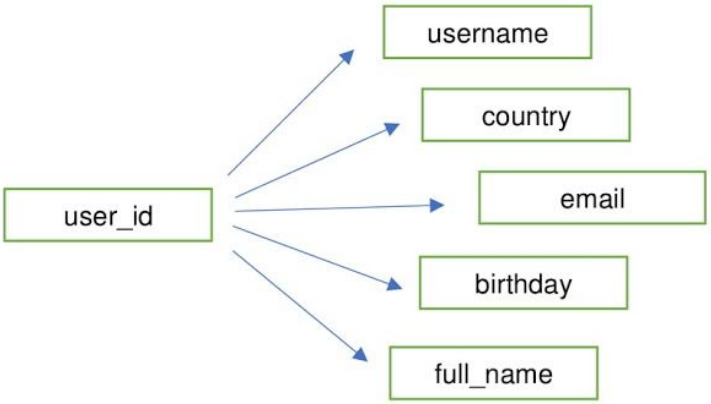
```
SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8
   SONG_ID SONG_ GENRE           TITLE
---------- ----- --------------- ----------------------------------
       833 3:50  Kpop            Boy With Luv
      1431 3:32  RnB             Imagine
      1459 3.27  Pop             thank u, next
      2894 3:04  Indie           New Americana
      3098 3:22  Pop             Without Me
      3409 4:38  Electro Pop     Castle
      3854 3:59  Kpop            Butterfly
      5634 3:19  Pop             There is Nothing Holding Me Back
      6877 3:29  Pop             Mercy
      7584 3:19  RnB             Bang Bang
      7659 3:02  Indie           Youth

   SONG_ID SONG_ GENRE           TITLE
---------- ----- --------------- ----------------------------------
      7895 3:52  Metal           Nightmare
      8905 3:47  Kpop            Jamais Vu
      8978 3:54  Indie           Suburbia
      9304 3:17  Pop             One Last Time
      9847 3:57  Indie           Talk Me Down

16 rows selected.
```

FUNCTIONAL DEPENDENCIES

1. Table 1 - **members**
    - {user_id} → username
    - {user_id} → country
    - {user_id} → email
    - {user_id} → full_name
    - {user_id} → birthday
2. Table 2 - **artists**
    - {artist_id} → artist_name
    - {artist_id} → country
    - {artist_id} → album_num
3. Table 3 - **podcasts**
    - {podcast_id} → title
    - {podcast_id} → podcast_date
    - {podcast_id} → podcast_length
    - {podcast_id} → categories
    - {podcast_id} → episodes
4. Table 4 - **playlists**
    - {playlist_id} → playlist_name
    - {playlist_id} → total_songs
5. Table 5 - **songs**
    - {song_id} → song_length
    - {song_id} → genre
    - {song_id} → title
6. Table 6 - **albums**
    - {album_id} → album_name
    - {album_id} → album_year
    - {album_id} → song_number
7. Relationship 1 - **releases** *(one to many relationship)*
    - {album_id} → artist_id
8. Relationship 2 - **contains** *(one to many relationship)*
    - {song_id} → album_id
9. Relationship 3 - **plays**
    - This relationship is M to N, it does not have any functional dependency
10. Relationship 4 - **has**
    - This relationship is M to N, it does not have any functional dependency
11. Relationship 5 - **creates**
    - This relationship is M to N, it does not have any functional dependency
12. Relationship 6 - **plays**
    - This relationship is M to N, it does not have any functional dependency

3NF NORMALIZATION

Table 1 – **members**



Decomposing Table 1 to 3NF:

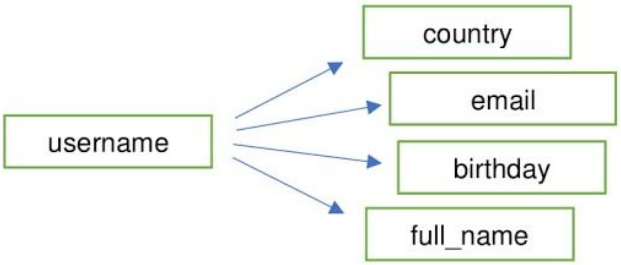
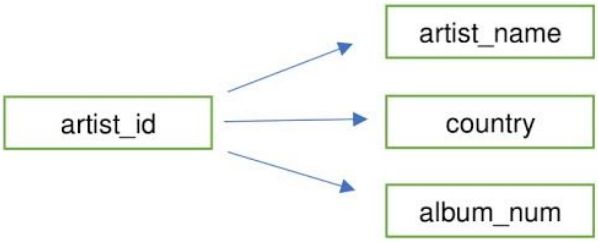


Table 2 – **artists**



Decomposing Table 2 to 3NF:

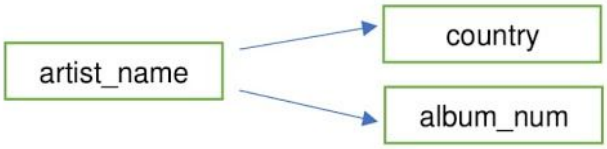
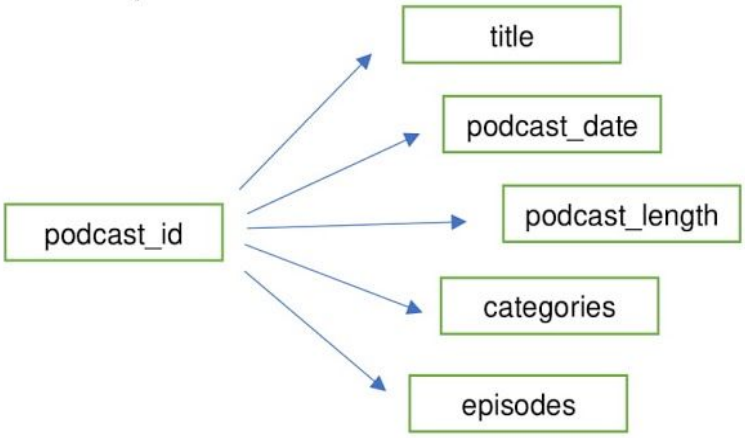


Table 3 – **podcasts**



Decomposing Table 3 to 3NF:

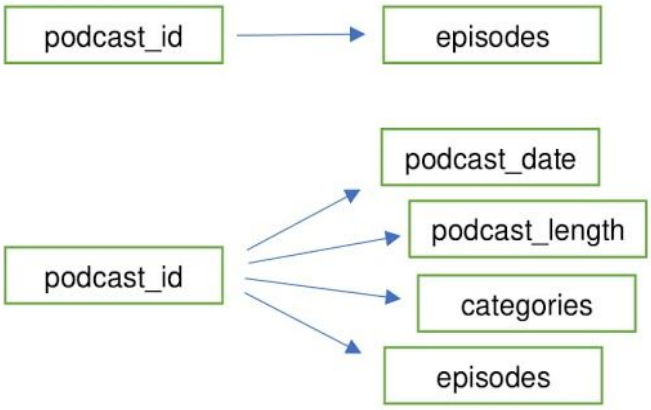
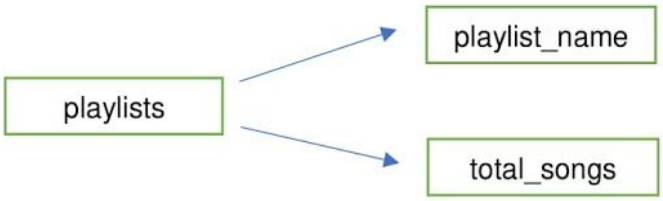


Table 4 – **playlists**



Decomposing Table 4 to 3NF:

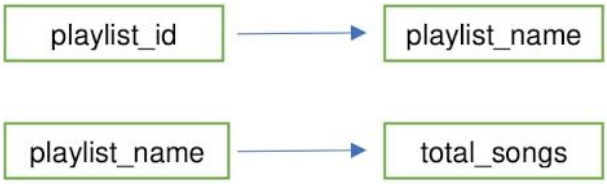
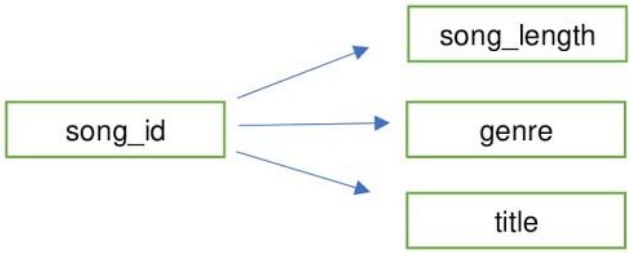


Table 5 – **songs**



Decomposing Table 5 to 3NF:

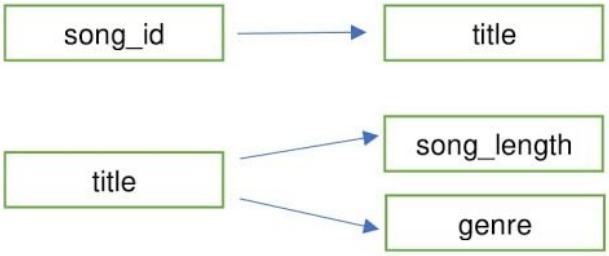
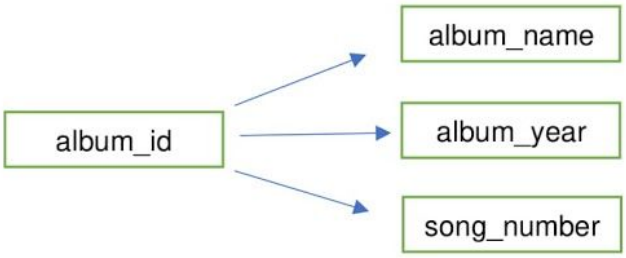
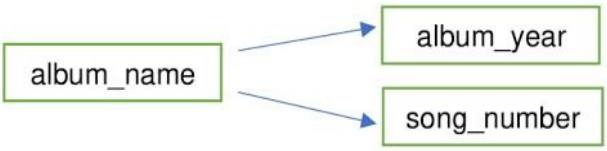


Table 6 – **albums**



Decomposing Table 6 to 3NF:



Relationship 1 – **releases**

Relationship 2 – **contains**

3NF BERNSTEIN'S ALGORITHM

Table

members(<u>user_id</u>, username, email, country, name, birthday)

<u>Functional Dependencies for this table:</u>

user_id → username, country, email
name, email → birthday
username, email → name
username → country, email

<u>Bernstein's Algorithm</u>

<u>Step 1:</u>

R(user_id, username, email, name, birthday, country)

FDs: {user_id → username, country, email
name, email → birthday
username, email → name
username → country, email}

<u>Step 2:</u>

a) Rewriting FDs so that each RHS is exactly one attribute
   1. **user_id → username**
   2. user_id → country
   3. user_id → email
   4. name, email → birthday
   5. username, email → name
   6. username → country
   7. username → email

b) Get rid of any redundant FDs
   1. $user\_id^+$ = {user_id, country, email} - no *<u>username</u>* in closure **(first FD is not redundant)**
   2. $user\_id^+$ = {user_id, username, email, name, birthday, country} - *<u>country</u>* is in closure **(second FD is redundant)**
   3. $user\_id^+$ = {user_id, username, email, name, birthday, country} - *<u>email</u>* is in closure **(third FD is redundant)**

4. name, email$^+$ = {name, email} - no *birthday* in closure **(forth FD is not redundant)**
5. username, email$^+$ = {username, email, country} - no *name* in closure **(fifth FD is not redundant)**
6. username$^+$ = {username, email, name, birthday} - no *country* in closure **(sixth FD is not redundant)**
7. username$^+$ = {username, country} - no *email* in closure **(seventh FD is not redundant)**

c) Find minimal dependencies
   1. user_id → username
   2. name, email → birthday
   3. username, email → name
   4. username → country
   5. username → email

Step 3: Finding keys

1. If only on the LHS, definitely part of key
   - user_id
2. If only on the RHS, definitely not part of key
   - country, birthday
3. Test closure of other attributes
   - user_id$^+$= {user_id, username, country, email, name, birthday} - is key
   - *user_id, username$^+$= {user_id, username, country, email, name, birthday} - is key*
   - user_id, email$^+$= {user_id, username, country, email, name, birthday} - is key
   - user_id, name$^+$= {user_id, username, country, email, name, birthday} - is key

Step 4: Deriving Relation Schema

R1(user_id, username)	***with FD: user_id → username***	→	**a key relation**
R2(name, email, birthday)	with FD: name, email → birthday		
R3(username, email, name)	with FD: username, email → name		
R4(username, country)	with FD: username → country		
R5(username, email)	with FD: username → email		

BCNF DECOMPOSITION

R(user_id, username, country, email, name, birthday}

user_id → username, country, email
name, email → birthday
username, email → name
username → country, email

<u>Decomposing R to BCNF</u>

user_id$^+$ = {user_id, username, country, email, name, birthday) **is a key**

name, email$^+$ = {name, email, birthday} *is not a key, R is not BCNF with respect to* **name, email → birthday**

1. ***Decompose R in R11 and R12***
   R11(user_id, username, country, email, name)
   R12 (name, email, birthday) with FD: name, email → birthday

2. ***Check the next FDs in terms of R11***

username, email$^+$ = {username, email, country, name} *is not a key, R11 is not BCNF with respect to* **username, email → name**

3. ***Decompose R11 in R111 and R112***
   R111(user_id, username, country, email)
   R112(username, email, name)

4. ***Check the next FDs in terms of R111***

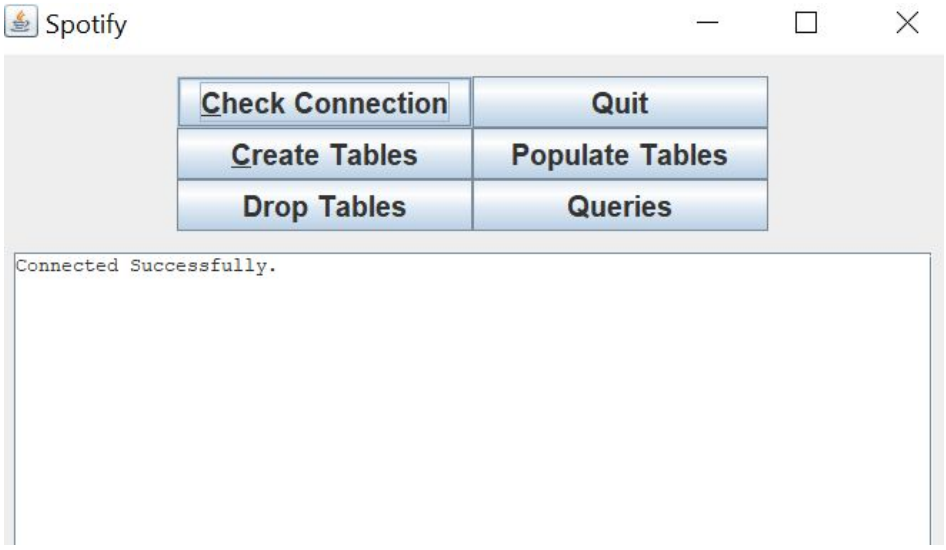
username$^+$ = {username, country, email} *is not a key so R111 is not BCNF with respect to* **username → country, email**

5. ***Decompose R111 in R1111 and R1112***
   R1111(user_id, username)
   R1112(username, country, email)

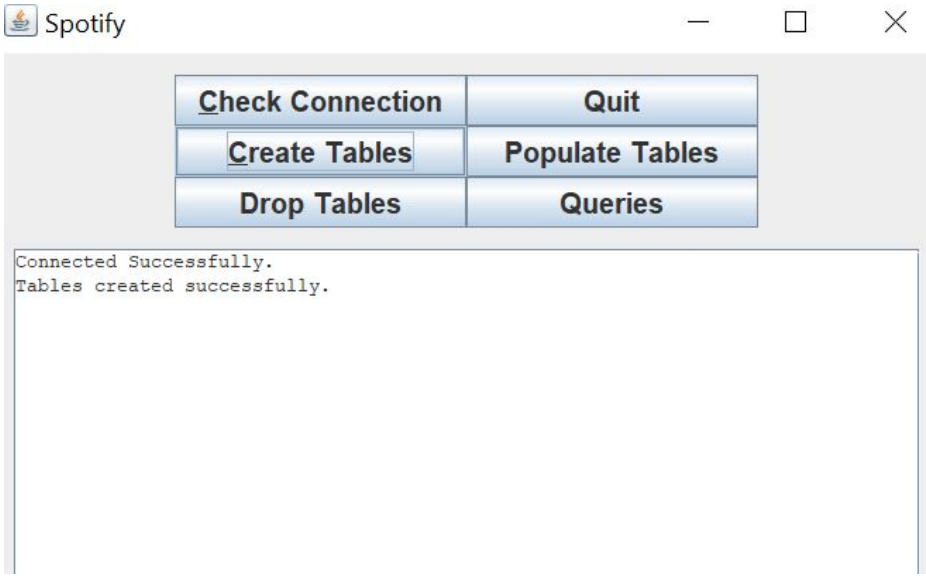
6. ***Final BCNF Schema for R***
   R1(name, email, birthday)
   R2(username, email, name)
   R3(username, country, email)
   R4(user_id, username)

WEB BASED USER INTERFACE

Checking the Connection:

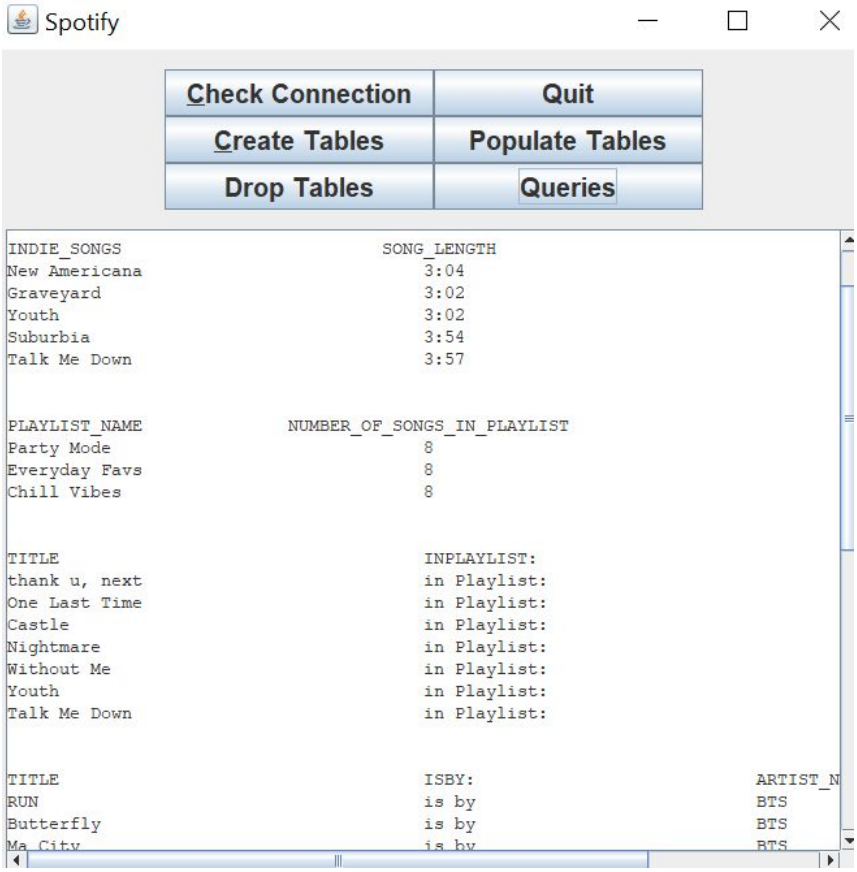


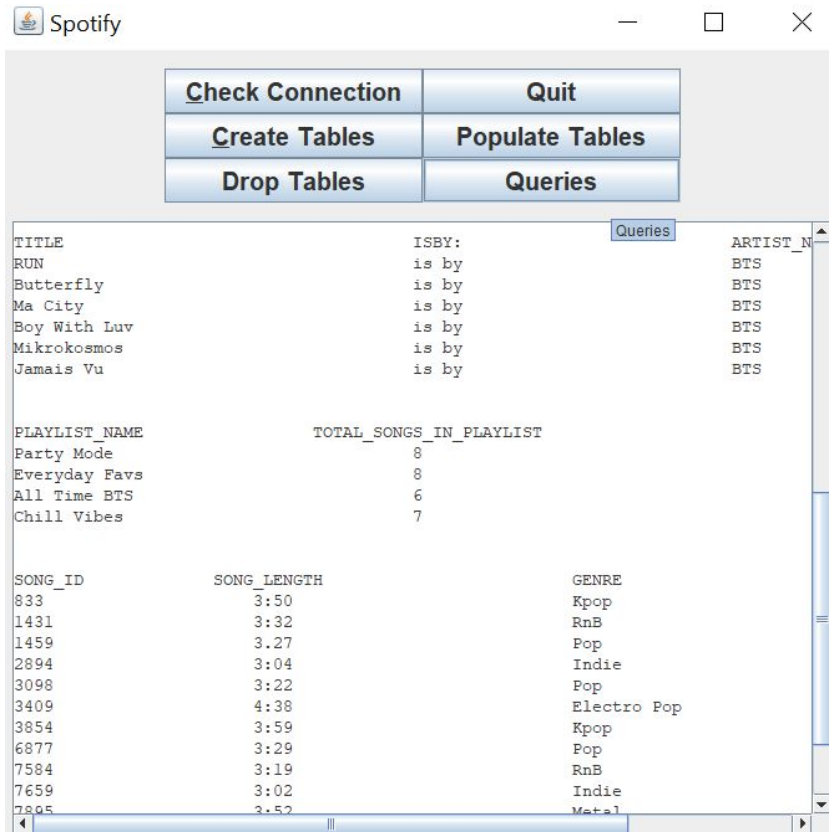
Creating the tables:

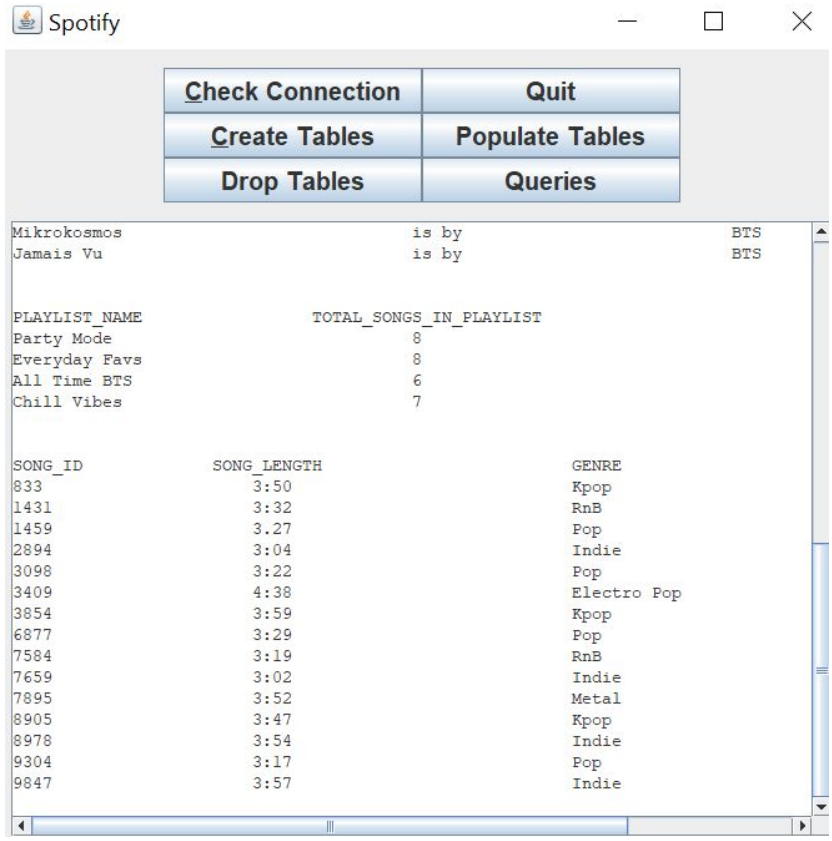


Populating the tables:

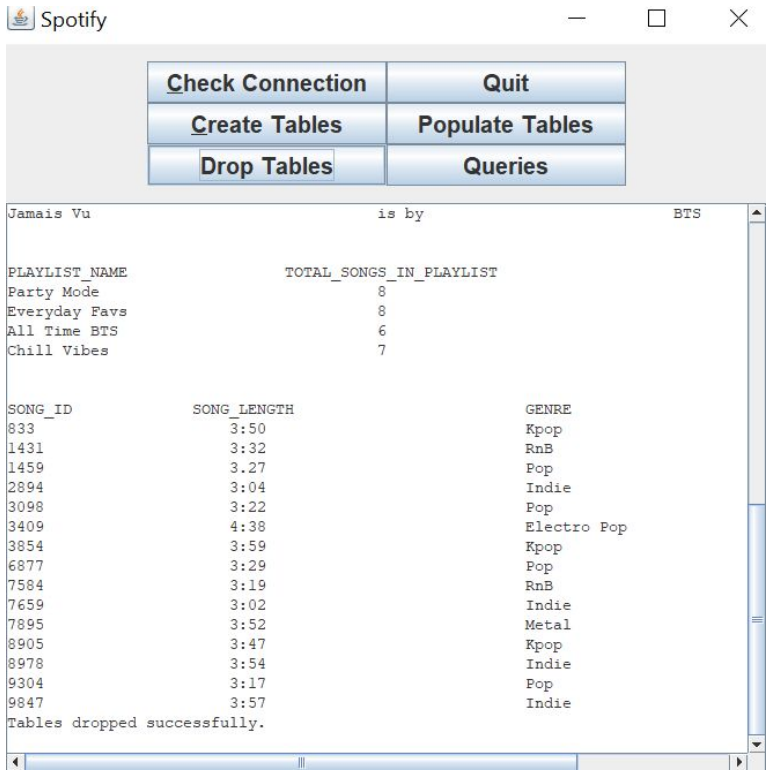
Printing the Queries:



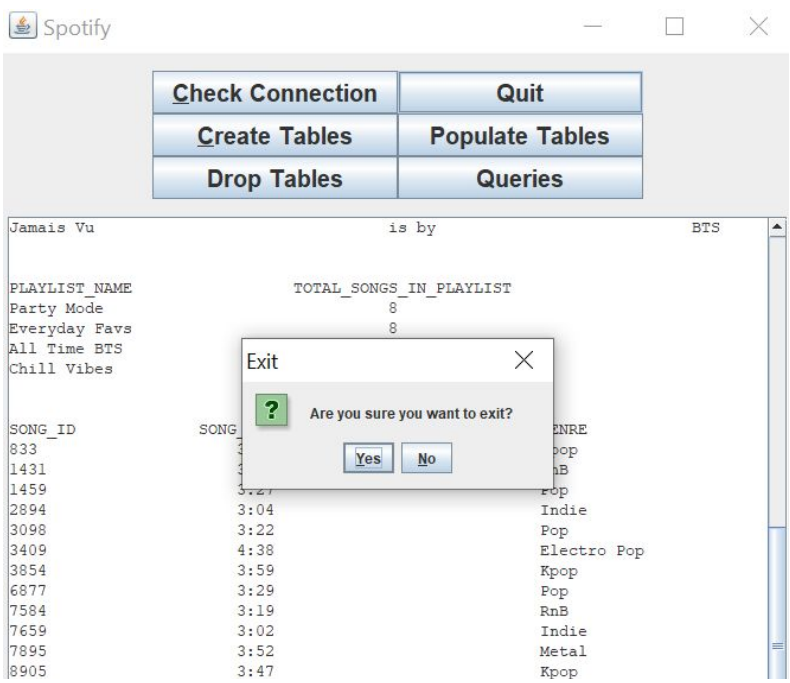

Spotify
Check Connection
Quit
Create Tables
Populate Tables
Drop Tables
Queries
Queries
TITLE ISBY: ARTIST_N
RUN is by BTS
Butterfly is by BTS
Ma City is by BTS
Boy With Luv is by BTS
Mikrokosmos is by BTS
Jamais Vu is by BTS
PLAYLIST_NAME TOTAL_SONGS_IN_PLAYLIST
Party Mode 8
Everyday Favs 8
All Time BTS 6
Chill Vibes 7
SONG_ID SONG_LENGTH GENRE
833 3:50 Kpop
1431 3:32 RnB
1459 3.27 Pop
2894 3:04 Indie
3098 3:22 Pop
3409 4:38 Electro Pop
3854 3:59 Kpop
6877 3:29 Pop
7584 3:19 RnB
7659 3:02 Indie


Spotify
Check Connection
Quit
Create Tables
Populate Tables
Drop Tables
Queries
Mikrokosmos is by BTS
Jamais Vu is by BTS
PLAYLIST_NAME TOTAL_SONGS_IN_PLAYLIST
Party Mode 8
Everyday Favs 8
All Time BTS 6
Chill Vibes 7
SONG_ID SONG_LENGTH GENRE
833 3:50 Kpop
1431 3:32 RnB
1459 3.27 Pop
2894 3:04 Indie
3098 3:22 Pop
3409 4:38 Electro Pop
3854 3:59 Kpop
6877 3:29 Pop
7584 3:19 RnB
7659 3:02 Indie
7895 3:52 Metal
8905 3:47 Kpop
8978 3:54 Indie
9304 3:17 Pop
9847 3:57 Indie

Dropping the tables:



Quitting and closing the connection:



RELATIONAL ALGEBRA NOTATION

SIMPLE QUERIES

1. SELECT *
   FROM members;

   σ **(members)**

2. SELECT *
   FROM artist
   WHERE Album_num > 1;

   $\sigma_{\text{Album\_num > 1}}$ **(artist)**

3. SELECT Episodes AS New_Podcasts, Podcast_date
   FROM podcast
   WHERE Podcast_date > to_date('2019-06-01', 'yyyy-mm-dd');

   $\rho_{\text{Episodes} \rightarrow \text{New\_podcast}}$ $(\pi_{\text{Episodes, Podcast\_date}}$ $(\sigma_{\text{Podcast\_date > to\_date('2019-06-01', 'yyyy-mm-dd')}}$ **(podcast)))**

4. SELECT Playlist_name ,Total_songs AS Number_of_Songs_in_Playlist
   FROM playlist
   WHERE Total_songs = 8;

   $\rho_{\text{Total\_songs} \rightarrow \text{Number\_of\_songs\_in\_playlist}}$ $(\pi_{\text{Playlist\_name, Total\_songs}}$ $(\sigma_{\text{Total\_songs = 8}}$ **(playlist)))**

5. SELECT Title AS Indie_Songs, Song_length
   FROM song
   WHERE Genre = 'Indie';

   $\rho_{\text{Title} \rightarrow \text{Indie\_Songs}}$ $(\pi_{\text{Title, Song\_length}}$ $(\sigma_{\text{Genre = 'Indie'}}$ **(song)))**

6. SELECT Album_name AS Album_2015, Album_year
   FROM album
   WHERE Album_year > to_date('2015-01-01', 'yyyy-mm-dd') AND Album_year < to_date('2016-01-01', 'yyyy-mm-dd');

$$\rho_{\text{Album\_name} \rightarrow \text{Album\_2015}}(\pi_{\text{Album\_name, Album\_year}} (\sigma_{\text{Album\_year > to\_date('2015-01-01', 'yyyy-mm-dd') AND Album\_year < to\_date('2016-01-01', 'yyyy-mm-dd')}} (\text{album})))$$

7. SELECT *
   FROM podcast
   WHERE Categories = 'True Crime' ;

$$\sigma_{\text{Categories = 'True Crime'}} (\text{podcast})$$

ADVANCED QUERIES

1. SELECT Title, 'in Playlist:', Playlist_name
   FROM song s, has h, playlist p
   WHERE s.Song_id = h.Song_id
   AND h.Playlist_id = p.Playlist_id
   AND Playlist_name = 'Chill Vibes';

$$\pi_{\text{Title, Playlist\_name}} (\sigma_{\text{Playlist\_name = 'Chill Vibes'}} (\text{song} \bowtie \text{has} \bowtie \text{playlist}))$$

2. SELECT Title, 'is by' , Artist_name
   FROM song s, album a, artist b, releases r, contains c
   WHERE s.Song_id = c.Song_id
   AND c.Album_id = a.Album_id
   AND a.Album_id = r.Album_id
   AND r.Artist_id = b.Artist_id
   AND Artist_name = 'BTS' ;

$$\pi_{\text{Title, Artist\_name}} (\sigma_{\text{Artist\_name = 'BTS'}} (\text{song} \bowtie \text{contains} \bowtie \text{album} \bowtie \text{releases} \bowtie \text{artist}))$$

3. CREATE VIEW sub_playlist AS
   (SELECT * FROM album
   WHERE Album_name LIKE 'M%');
   SELECT Album_name, Album_year
   FROM sub_playlist
   WHERE Album_id = 4562;

$$\pi_{\text{Album\_name, Album\_year}} (\sigma_{\text{Album\_id = 4562}} (\text{sub\_playlist}))$$

4. CREATE VIEW kpop_songs AS
   (SELECT *
   FROM song
   WHERE Genre = 'Kpop');
   SELECT *
   FROM kpop_songs
   WHERE Song_length LIKE '3%';

$$\sigma_{\text{song\_length LIKE '3\%'}} (\text{kpop\_songs}))$$

5. SELECT Playlist_name, COUNT(Song_id) AS Total_Songs_in_Playlist
   FROM playlist p, has h
   WHERE p.Playlist_id = h.Playlist_id
   GROUP BY Playlist_name;

$${}_{\text{Playlist\_name}}F_{\text{COUNT Song\_id}} (\text{playlist} \bowtie \text{has})$$

6. (SELECT *
   FROM song)
   MINUS
   (SELECT s.*
   FROM song s, playlist p, has h
   WHERE p.Playlist_name = 'Party Mode'
   AND p.Playlist_id = h.Playlist_id
   AND h.Song_id = s.Song_id);

$$\Sigma (\text{song}) - \sigma_{\text{Playlist\_name = 'Party Mode'}} (\text{playlist} \bowtie \text{has} \bowtie \text{song})$$

CONCLUDING REMARKS ON DESIGN EXPERIENCE

The design experience was a very rewarding process to have been a part of. Though the design process took a very long time to complete, we were able to learn many skills crucial to creating a database management system. This would include things like familiarizing ourselves with various SQL commands, applying our knowledge to create functional dependencies and decomposing relations, such as the 3NF and BCNF database we created in assignments 7 and 8, creating the appropriate Java UI and connecting it to the database, and applying the appropriate order of steps in designing the database system as a whole, among other things. That being said, the design process was completed by dividing the tasks into small groups. This was quite beneficial for us as a group, as we were all able to focus on the specific details pertaining to each task, rather than being overwhelmed by the number of tasks to complete. Each task could typically be completed within 2-4 hours, with the exception of the ER diagram and the Java UI, which took roughly 1-3 days to complete. However, our efforts paid off, as seen through the successfully operating, UI-connected, database management system. Overall, we are very satisfied with the database system we created, and are ultimately grateful for acquiring the crucial knowledge and skills that will aid us in our future endeavors.