University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

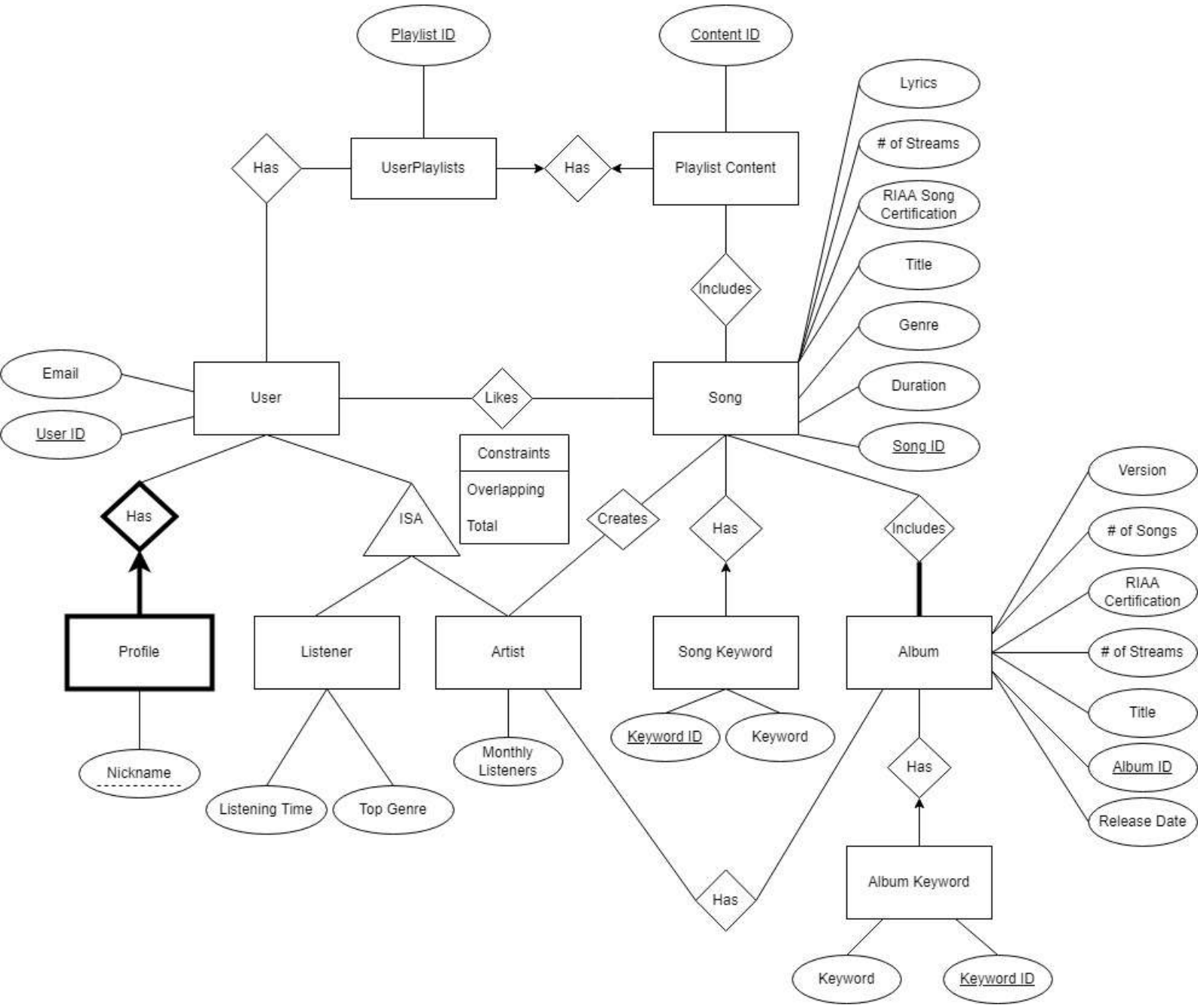
Date: July 23, 2022

Group Number: 29

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Rahul Shandilya	63475768	w0d2b	shandilyaholy@gmail.com
Melissa Lin	44974509	l3j6a	melissa22lin@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia


Playlist ID
Content ID
Lyrics
# of Streams
RIAA Song Certification
Title
Genre
Duration
Song ID
Has
UserPlaylists
Has
Playlist Content
Includes
Email
User ID
User
Likes
Song
Constraints
Overlapping
Total
Has
ISA
Creates
Has
Includes
Version
# of Songs
RIAA Certification
# of Streams
Title
Album ID
Release Date
Profile
Listener
Artist
Song Keyword
Album
Nickname
Listening Time
Top Genre
Monthly Listeners
Keyword ID
Keyword
Has
Has
Album Keyword
Keyword
Keyword ID

Milestone 2

ER DIAGRAM CHANGES:

- Removed playlistID in User
- Added attributes to Song:
    - Lyrics
    - # of Streams
    - RIAA Song Certification
- Added attributes to Album:
    - # of Streams
    - RIAA Song Certification
    - Version
    - Release Date

RELATIONS PRE-NORMALIZATION:

Listener(User ID: integer, Email: string, Listening Time: integer, Top Genre: string)
Artist(User ID: integer, Email: string, Monthly Listeners: integer)
ProfileHas(**User ID**: integer, Nickname: string)
UserLikesSong(**User ID**: integer**, Song ID**: integer)
Song(SongID: integer, Title: string, Genre: string, Duration: real, Lyrics: string, # of Streams: string, RIAA Song Certification: string)
SongHasKeyword(KeywordID: integer, Keyword: string, **Song ID**: integer)
UserPlaylists(PlaylistID: integer)
UserHasUserPlaylists(**User ID**: integer**, Playlist ID**: integer)
PlaylistContent(ContentID: integer, **Playlist ID**: integer) *playlist id must be unique
PlaylistContentIncludesSong(**Content ID**: integer, **Song ID**: integer)
Album(AlbumID: integer, # of Songs: integer, Title: string, # of Streams: integer, RIAA Certification: string, Version: integer, Release Date: Date)
AlbumHasKeyword(KeywordID: integer, Keyword: string, **AlbumID**: integer)
AlbumIncludesSong(**Album ID**: integer, **SongID**: integer) *album cannot be null
ArtistCreatesSong(**User ID**: integer, **Song ID**: integer)
ArtistHasAlbum(**User ID**: integer, **Album ID**: integer)

FUNCTIONAL DEPENDENCIES:

1. Song: # of streams -> certification
2. Song: lyrics -> title
3. Album: # of streams -> certification
4. Album: Title, Version -> Release Date, # of Songs

NORMALIZATION:

Song:

Relation:
Song(SongID: integer, Title: string, Genre: string, Duration: real, Lyrics: string, # of Streams: string, RIAA Song Certification: string)

FDs:

1. SongID -> Title,Genre,Duration,Lyrics,# of Stream, RIAA Song Certification
2. # of Streams -> RIAA Song Certification
3. Lyrics -> Title

[SongID]+ = {SongID, Title,Genre,Duration,Lyrics,# of Stream, RIAA Song Certification}

[# of streams]+ = {# of streams, RIAA Song Certification}

[Lyrics]+ = {Lyrics, Title}

Since closure 2 and 3 do not give all the attributes in the relation, they are not superkeys. Hence, they violate BCNF, i.e., the relation is not in BCNF.

Minimal Keys:

Left	Middle	Right
SongID	# of streams, lyrics	Title,Genre,Duration, RIAA Song Certification

[SongID]+ = {Title,Genre,Duration,Lyrics,# of Stream, RIAA Song Certification}

SongID is the only minimal key of the relation.

FDs 2 and 3 have right hand side values that are not part of the minimal keys. Hence, they violate 3NF, i.e., the relation is not in 3NF.

So we need to decompose it.

Minimal Covers:

- Reduce RHS and LHS:
  i) SongID -> Title,
  ii) SongID -> Genre,
  iii) SongID -> Duration,
  iv) SongID -> Lyrics,
  v) SongID -> # of streams,
  vi) SongID -> RIAA Song Certification,
  vii) # of streams -> RIAA Song Certification,
  viii) Lyrics -> Title

- Remove Redundant FDs:
  (i) SongID -> Title, (This can be derived using (iv) and (viii))
  (ii) SongID -> Genre,
  (iii) SongID -> Duration,
  (iv) SongID -> Lyrics,
  (v) SongID -> # of streams,
  (vi) SongID -> RIAA Song Certification, (This can be derived using (v) and (vii))
  (vii) # of streams -> RIAA Song Certification,
  (viii) Lyrics -> Title

Hence the minimal cover is:

- SongID -> Genre, Duration, Lyrics, # of streams
- # of streams -> RIAA Song Certification
- Lyrics -> Title

Decomposing using synthesis, we get,

- Song(SongID: integer, Genre: string, Duration: real, **Lyrics**: string, **# of streams**: integer)
- StreamSongCertification(# of streams: integer, RIAA Song Certification: string)
- LyricsTitle(Lyrics: string, Title: string)

Album:

Relation:
Album(AlbumID: integer, # of Songs: integer, Title: string, # of Streams: integer, RIAA Certification: string, Version: integer, Release Date: Date)

FDs:

1. AlbumID -> Title, # of Songs, # of Streams, RIAA Certification, Version, Release Date
2. # of Streams -> RIAA Certification
3. Title, Version -> Release Date, # of Songs

[AlbumID]+ = {AlbumID, Title, # of Songs, # of Streams, RIAA Certification, Version, Release Date}

[# of Streams]+ = {# of Streams, RIAA Certification}

[Title, Version]+ = {Title, Version, Release Date, # of Songs}

Since closure 2 and 3 do not give all the attributes in the relation, they are not superkeys. Hence, they violate BCNF, i.e., the relation is not in BCNF.

Minimal Keys:

Left	Middle	Right
AlbumID	# of streams, Title, Version	# of Songs, RIAA Certification, Release Date

[AlbumID]+ = {AlbumID, Title, # of Songs, # of Streams, RIAA Certification, Version, Release Date}

AlbumID is the only minimal key of the relation.

FDs 2 and 3 have right hand side values that are not part of the minimal keys. Hence, they violate 3NF, i.e., the relation is not in 3NF.

So we need to decompose it.

Minimal Covers:

- Reduce RHS and LHS:
  AlbumID -> Title,
  AlbumID -> # of Songs,
  AlbumID -> # of Streams,
  AlbumID -> RIAA Certification,
  AlbumID -> Version,
  AlbumID -> Release Date,
  # of Streams -> RIAA Certification,
  Title, Version -> Release Date
  Title, Version -> # of Songs

- Remove Redundant FDs:
  AlbumID -> Title,
  AlbumID -> # of Songs, (This can be derived from other FDs)
  AlbumID -> # of Streams,
  AlbumID -> RIAA Certification, (This can be derived from other FDs)
  AlbumID -> Version,
  AlbumID -> Release Date, (This can be derived from other FDs)
  # of Streams -> RIAA Certification,
  Title, Version -> Release Date,
  Title, Version -> # of Songs

Hence the minimal cover is:

- AlbumID -> Title, # of Streams, Version
- # of Streams -> RIAA Certification
- Title, Version -> Release Date, # of Songs

Decomposing using synthesis, we get,

- Album(AlbumID: integer, **Title**: string, **Version**: integer, **# of streams**: integer)
- StreamAlbumCertification(# of streams: integer, RIAA Certification: string)
- TitleRelease(Title: string, Version: integer, Release Date: Date, # of Songs: integer)

FINAL RELATIONAL MODEL POST-NORMALIZATION:

Listener(User ID: integer, Email: string, Listening Time: integer, Top Genre: string)
Artist(User ID: integer, Email: string, Monthly Listeners: integer)
ProfileHas(**User ID**: integer, Nickname: string)
UserLikesSong(**User ID**: integer**, Song ID**: integer)
Song(SongID: integer, Genre: string, Duration: real, **Lyrics**: string, **# of Streams**: integer)
StreamSongCertification(# of Streams: integer, RIAA Song Certification: string)
LyricsTitle(Lyrics: string, Title: string)
SongHasKeyword(KeywordID: integer, Keyword: string, **Song ID**: integer)
UserPlaylists(PlaylistID: integer)
UserHasUserPlaylists(**User ID**: integer**, Playlist ID**: integer)
PlaylistContent(ContentID: integer, **Playlist ID**: integer) *playlist id must be unique
PlaylistContentIncludesSong(**Content ID**: integer, **Song ID**: integer)
Album(AlbumID: integer, **Title**: string, **Version**: integer, **# of Streams**: integer)
StreamAlbumCertification(# of Streams: integer, RIAA Certification: string)
TitleRelease(Title: string, Version: integer, Release Date: Date, # of Songs: integer)
AlbumHasKeyword(KeywordID: integer, Keyword: string, **AlbumID**: integer)
AlbumIncludesSong(**Album ID**: integer, **SongID**: integer) *album cannot be null
ArtistCreatesSong(**User ID**: integer, **Song ID**: integer)
ArtistHasAlbum(**User ID**: integer, **Album ID**: integer)

SQL DDL:

```
CREATE TABLE Listener(UserID INTEGER PRIMARY KEY,
                Email VARCHAR(50),
                ListeningTime INTEGER,
                TopGenre VARCHAR(20));

CREATE TABLE Artist(UserID INTEGER PRIMARY KEY,
                Email VARCHAR(50),
                MonthlyListeners INTEGER);

CREATE TABLE ProfileHas(UserID INTEGER,
                    Nickname VARCHAR(30),
                    PRIMARY KEY (UserID, Nickname),
                    FOREIGN KEY (UserID) REFERENCES Listener);

CREATE TABLE Song(SongID INTEGER PRIMARY KEY,
```

```
                Genre VARCHAR(20),
                Duration REAL,
                Lyrics VARCHAR(200),
                NumStreams INTEGER,
                FOREIGN KEY (Lyrics) REFERENCES LyricsTitle(Lyrics),
                FOREIGN KEY (NumStreams) REFERENCES
StreamSongCertification(NumStreams));

CREATE TABLE UserLikesSong(UserID INTEGER,
                    SongID INTEGER,
                    PRIMARY KEY (UserID, SongID),
                    FOREIGN KEY (UserID) REFERENCES Listener,
                    FOREIGN KEY (SongID) REFERENCES Song);

CREATE TABLE StreamSongCertification(NumStreams INTEGER PRIMARY KEY,
                        Certification VARCHAR(20));

CREATE TABLE LyricsTitle(Lyrics VARCHAR(200) PRIMARY KEY,
                    Title VARCHAR(200));

CREATE TABLE SongHasKeyword(KeywordID INTEGER PRIMARY KEY,
                    Keyword VARCHAR(30),
                    SongID INTEGER,
                    FOREIGN KEY (SongID) REFERENCES Song);

CREATE TABLE UserPlaylists(PlaylistID INTEGER PRIMARY KEY);

CREATE TABLE UserHasUserPlaylists(UserID INTEGER,
                        PlaylistID INTEGER,
                        PRIMARY KEY (UserID, PlaylistID),
                        FOREIGN KEY (UserID) REFERENCES Listener,
                        FOREIGN KEY (PlaylistID) REFERENCES UserPlaylists);

CREATE TABLE PlaylistContent(ContentID INTEGER PRIMARY KEY,
                    PlaylistID INTEGER,
                    UNIQUE (PlaylistID),
                    FOREIGN KEY (PlaylistID) REFERENCES UserPlaylists);

CREATE TABLE PlaylistContentIncludesSong(ContentID INTEGER,
                            SongID INTEGER,
```

```
                    PRIMARY KEY (ContentID, SongID),
                    FOREIGN KEY (ContentID) REFERENCES PlaylistContent,
                    FOREIGN KEY (SongID) REFERENCES Song);

CREATE TABLE StreamAlbumCertification(NumStreams INTEGER PRIMARY KEY,
                    Certification VARCHAR(30));

CREATE TABLE TitleRelease(Title VARCHAR(30),
                    AlbumVersion INTEGER,
                    ReleaseDate DATE,
                    NumSongs INTEGER,
                    PRIMARY KEY (Title, AlbumVersion));

CREATE TABLE Album(AlbumID INTEGER PRIMARY KEY,
                    Title VARCHAR(20),
                    AlbumVersion INTEGER,
                    NumStreams INTEGER,
                    FOREIGN KEY (Title, AlbumVersion) REFERENCES TitleRelease,
                    FOREIGN KEY (NumStreams) REFERENCES StreamAlbumCertification);

CREATE TABLE AlbumHasKeyword(KeywordID INTEGER PRIMARY KEY,
                    Keyword VARCHAR(30),
                    AlbumID INTEGER,
                    FOREIGN KEY (AlbumID) REFERENCES Album);

CREATE TABLE AlbumIncludesSong(AlbumID INTEGER NOT NULL,
                    SongID INTEGER,
                    PRIMARY KEY (AlbumID, SongID),
                    FOREIGN KEY (AlbumID) REFERENCES Album,
                    FOREIGN KEY (SongID) REFERENCES Song);

CREATE TABLE ArtistCreatesSong(UserID INTEGER,
                    SongID INTEGER,
                    PRIMARY KEY (UserID, SongID),
                    FOREIGN KEY (UserID) REFERENCES Artist,
                    FOREIGN KEY (SongID) REFERENCES Song);

CREATE TABLE ArtistHasAlbum(UserID INTEGER,
                    AlbumID INTEGER,
                    PRIMARY KEY (UserID, AlbumID),
```

```
FOREIGN KEY (UserID) REFERENCES Artist,
FOREIGN KEY (AlbumID) REFERENCES Album);
```

POPULATED TABLES (screenshots of data are provided below):

```
INSERT INTO Listener (UserID,Email,ListeningTime,TopGenre) VALUES( 0,
'holyheck@gmail.com', 1359022,'Pop');
INSERT INTO Listener (UserID,Email,ListeningTime,TopGenre) VALUES( 1,
'reeeee@gmail.com', 300,'Rap');
INSERT INTO Listener (UserID,Email,ListeningTime,TopGenre) VALUES( 2,
'dolphinsareok@outlook.com', 65123,'K-Pop');
INSERT INTO Listener (UserID,Email,ListeningTime,TopGenre) VALUES( 52,
'apple@hotmail.com', 5,'Classical');
INSERT INTO Listener (UserID,Email,ListeningTime,TopGenre) VALUES( 69,
'nice@hotmail.com', 420,'Monkey-Trap');

INSERT INTO Artist (UserID, Email, MonthlyListeners)
VALUES
(
        0,
        'holyheck@gmail.com',
        1630928
    );
INSERT INTO Artist (UserID, Email, MonthlyListeners)
VALUES
    (
        3,
        'shandilya@gmail.com',
        250
    );
INSERT INTO Artist (UserID, Email, MonthlyListeners)
VALUES    (
        4,
        'ilovedolphins@outlook.com',
        1205304
    );
INSERT INTO Artist (UserID, Email, MonthlyListeners)
VALUES(
        5,
        'ihatedolphins@hotmail.com',
        340792
    );
INSERT INTO Artist (UserID, Email, MonthlyListeners)
```

```
VALUES(
        6,
        'imaboomer@yahoo.com',
        1
    );

INSERT INTO ProfileHas (UserID, Nickname)
VALUES(0, 'Melissa');
INSERT INTO ProfileHas (UserID, Nickname)
VALUES (1, 'Rahul');
INSERT INTO ProfileHas (UserID, Nickname)
VALUES    (2, 'Mark');
INSERT INTO ProfileHas (UserID, Nickname)
VALUES    (52, 'Michelle');
INSERT INTO ProfileHas (UserID, Nickname)
VALUES     (69, 'ironcat');

INSERT INTO StreamSongCertification (NumStreams, Certification) VALUES
(10000000, 'Diamond');
INSERT INTO StreamSongCertification (NumStreams, Certification) VALUES (2000000,
'Multi-Platinum');
INSERT INTO StreamSongCertification (NumStreams, Certification) VALUES (500000,
'Gold');
INSERT INTO StreamSongCertification (NumStreams, Certification) VALUES (1000000,
'Platinum');
INSERT INTO StreamSongCertification (NumStreams, Certification) VALUES (100000,
'None');

INSERT INTO LyricsTitle(Lyrics, Title) VALUES
('Fly','Baby Shark');
INSERT INTO LyricsTitle(Lyrics, Title) VALUES
('Never','Never ');
INSERT INTO LyricsTitle(Lyrics, Title) VALUES
('Auuuugh','ARRRRRGH');
INSERT INTO LyricsTitle(Lyrics, Title) VALUES
('NAAAAAAAAAAAAAH','Trumpet Boiii');
INSERT INTO LyricsTitle(Lyrics, Title) VALUES
('Country','Country Road');

INSERT INTO
```

```
   Song (SongID, Genre,Duration,Lyrics,NumStreams)
VALUES (0, 'Pop', 3, 'Fly', 10000000);

INSERT INTO
   Song (SongID, Genre,Duration,Lyrics,NumStreams)
VALUES (1, 'Rap', 6, 'Never', 2000000);

INSERT INTO Song (SongID, Genre,Duration,Lyrics,NumStreams)
VALUES (53, 'Monkey-Trap', 4, 'Auuuugh', 500000);

INSERT INTO Song (SongID, Genre,Duration,Lyrics,NumStreams)
VALUES (47, 'Jazz', 1, 'NAAAAAAAAAAAAH', 1000000);

INSERT INTO Song (SongID, Genre,Duration,Lyrics,NumStreams)
VALUES (14, 'Country', 5, 'Country', 100000);

INSERT INTO UserLikesSong (UserID, SongID)
VALUES(0, 0);
INSERT INTO UserLikesSong (UserID, SongID)
VALUES(0, 1);
INSERT INTO UserLikesSong (UserID, SongID)
VALUES (2, 53);
INSERT INTO UserLikesSong (UserID, SongID)
VALUES (52, 0);
INSERT INTO UserLikesSong (UserID, SongID)
VALUES(69, 0);

INSERT INTO SongHasKeyword (KeywordID, Keyword, SongID) VALUES (0,
'sociopath', 0);
INSERT INTO SongHasKeyword (KeywordID, Keyword, SongID) VALUES (1, 'apathy',
0);
INSERT INTO SongHasKeyword (KeywordID, Keyword, SongID) VALUES (2, 'turn',
53);
INSERT INTO SongHasKeyword (KeywordID, Keyword, SongID) VALUES (3, 'me', 53);
INSERT INTO SongHasKeyword (KeywordID, Keyword, SongID) VALUES (4, 'on', 53);

INSERT INTO UserPlaylists (PlaylistID) VALUES (0);
INSERT INTO UserPlaylists (PlaylistID) VALUES (1);
INSERT INTO UserPlaylists (PlaylistID) VALUES (2);
INSERT INTO UserPlaylists (PlaylistID) VALUES (3);
```

```
INSERT INTO UserPlaylists (PlaylistID) VALUES (444);

INSERT INTO UserHasUserPlaylists (UserID, PlaylistID) VALUES (0, 0);
INSERT INTO UserHasUserPlaylists (UserID, PlaylistID) VALUES (0, 1);
INSERT INTO UserHasUserPlaylists (UserID, PlaylistID) VALUES (2, 2);
INSERT INTO UserHasUserPlaylists (UserID, PlaylistID) VALUES (52, 3);
INSERT INTO UserHasUserPlaylists (UserID, PlaylistID) VALUES (52, 444);

INSERT INTO PlaylistContent (ContentID, PlaylistID) VALUES (0, 0);
INSERT INTO PlaylistContent (ContentID, PlaylistID) VALUES (1, 1);
INSERT INTO PlaylistContent (ContentID, PlaylistID) VALUES (2, 2);
INSERT INTO PlaylistContent (ContentID, PlaylistID) VALUES (3, 3);
INSERT INTO PlaylistContent (ContentID, PlaylistID) VALUES (444, 444);

INSERT INTO PlaylistContentIncludesSong (ContentID, SongID) VALUES (0, 0);
INSERT INTO PlaylistContentIncludesSong (ContentID, SongID) VALUES (0, 1);
INSERT INTO PlaylistContentIncludesSong (ContentID, SongID) VALUES (2, 47);
INSERT INTO PlaylistContentIncludesSong (ContentID, SongID) VALUES (2, 53);
INSERT INTO PlaylistContentIncludesSong (ContentID, SongID) VALUES (444, 53);

INSERT INTO StreamAlbumCertification (NumStreams, Certification) VALUES
(10000000, 'Diamond');
INSERT INTO StreamAlbumCertification (NumStreams, Certification) VALUES
(2000000, 'Multi-Platinum');
INSERT INTO StreamAlbumCertification (NumStreams, Certification) VALUES (500000,
'Gold');
INSERT INTO StreamAlbumCertification (NumStreams, Certification) VALUES
(1000000, 'Platinum');
INSERT INTO StreamAlbumCertification (NumStreams, Certification) VALUES (100000,
'None');

INSERT INTO TitleRelease(Title, AlbumVersion, ReleaseDate, NumSongs) VALUES
('032 Funk', 1, '01-APR-21', 5);
INSERT INTO TitleRelease(Title, AlbumVersion, ReleaseDate, NumSongs) VALUES
('30', 1, '16-MAY-20', 10);
INSERT INTO TitleRelease(Title, AlbumVersion, ReleaseDate, NumSongs) VALUES
('Light Switch', 1, '30-MAR-22', 1);
INSERT INTO TitleRelease(Title, AlbumVersion, ReleaseDate, NumSongs) VALUES
('Pigeon!', 1, '25-FEB-22', 8);
```

```
INSERT INTO TitleRelease(Title, AlbumVersion, ReleaseDate, NumSongs) VALUES
('ZZZ', 1, '16-NOV-18', 9);

INSERT INTO Album(AlbumID, Title, AlbumVersion, NumStreams) VALUES (0, '032
Funk', 1, 2000000);
INSERT INTO Album(AlbumID, Title, AlbumVersion, NumStreams) VALUES (1, '30', 1,
10000000);
INSERT INTO Album(AlbumID, Title, AlbumVersion, NumStreams) VALUES (2, 'Light
Switch', 1, 500000);
INSERT INTO Album(AlbumID, Title, AlbumVersion, NumStreams) VALUES (5,
'Pigeon!', 1, 100000);
INSERT INTO Album(AlbumID, Title, AlbumVersion, NumStreams) VALUES (99, 'ZZZ',
1, 1000000);

INSERT INTO AlbumHasKeyword (KeywordID, Keyword, AlbumID) VALUES (0, 'Funk',
0);
INSERT INTO AlbumHasKeyword (KeywordID, Keyword, AlbumID) VALUES (1,
'Celebration', 0);
INSERT INTO AlbumHasKeyword (KeywordID, Keyword, AlbumID) VALUES (2,
'Incheon Airport Freestyle', 0);
INSERT INTO AlbumHasKeyword (KeywordID, Keyword, AlbumID) VALUES (3, 'Sad',
1);
INSERT INTO AlbumHasKeyword (KeywordID, Keyword, AlbumID) VALUES (4,
'Upbeat', 2);

INSERT INTO AlbumIncludesSong (AlbumID, SongID) VALUES (0, 0);
INSERT INTO AlbumIncludesSong (AlbumID, SongID) VALUES (0, 1);
INSERT INTO AlbumIncludesSong (AlbumID, SongID) VALUES (1, 47);
INSERT INTO AlbumIncludesSong (AlbumID, SongID) VALUES (2, 53);
INSERT INTO AlbumIncludesSong (AlbumID, SongID) VALUES (2, 14);

INSERT INTO ArtistHasAlbum (UserID, AlbumID) VALUES (0, 0);
INSERT INTO ArtistHasAlbum (UserID, AlbumID) VALUES (0, 1);
INSERT INTO ArtistHasAlbum (UserID, AlbumID) VALUES (3, 2);
INSERT INTO ArtistHasAlbum (UserID, AlbumID) VALUES (4, 5);
INSERT INTO ArtistHasAlbum (UserID, AlbumID) VALUES (5, 99);
```

Listener Table:

```
    USERID EMAIL                                              LISTENINGTIME
---------- -------------------------------------------------- -------------
TOPGENRE
--------------------
         0 holyheck@gmail.com                                       1359022
Pop

         1 reeeee@gmail.com                                             300
Rap

         2 dolphinsareok@outlook.com                                  65123
K-Pop


    USERID EMAIL                                              LISTENINGTIME
---------- -------------------------------------------------- -------------
TOPGENRE
--------------------
        52 apple@hotmail.com                                            5
Classical

        69 nice@hotmail.com                                           420
Monkey-Trap
```

Artist Table:

```
   USERID EMAIL                                           MONTHLYLISTENERS
--------- -------------------------------------------------- ---------------
        0 holyheck@gmail.com                                        1630928
        3 shandilya@gmail.com                                           250
        4 ilovedolphins@outlook.com                                1205304
        5 ihatedolphins@hotmail.com                                 340792
        6 imaboomer@yahoo.com                                            1
```

ProfileHas Table:

USERID	NICKNAME
0	Melissa
1	Rahul
2	Mark
52	Michelle
69	ironcat

StreamSongCertification Table:

NUMSTREAMS	CERTIFICATION
10000000	Diamond
2000000	Multi-Platinum
500000	Gold
1000000	Platinum
100000	None

```
SQL> select * from lyricstitle;

LYRICS
--------------------------------------------------------------------------------
TITLE
--------------------
Baby Shark Doo Doo Dooo
Baby Shark

Auuuugh
ARRRRRRRRRRGH

NAAAAAAAAAAAAAH
Trumpet Boiii


LYRICS
--------------------------------------------------------------------------------
TITLE
--------------------
Country Road Take Home
Country Road

Baby
Baby Shark

Never
Never


LYRICS
--------------------------------------------------------------------------------
TITLE
--------------------
Fly me to the moon
Baby Shark

Never
Never
```

LyricsTitle Table Continued:

```
Fly
Baby Shark


LYRICS
--------------------------------------------------------------------------------
TITLE
--------------------
Country
Country Road
```

```
SQL> select * from song;

    SONGID GENRE                DURATION
---------- -------------------- ----------
LYRICS
--------------------------------------------------------------------------------
NUMSTREAMS
----------
         0 Pop                          3
Fly
  10000000

         1 Rap                          6
Never
   2000000

    SONGID GENRE                DURATION
---------- -------------------- ----------
LYRICS
--------------------------------------------------------------------------------
NUMSTREAMS
----------

        53 Monkey-Trap                  4
Auuuugh
    500000

        47 Jazz                         1
NAAAAAAAAAAAAAH

    SONGID GENRE                DURATION
---------- -------------------- ----------
LYRICS
--------------------------------------------------------------------------------
NUMSTREAMS
----------
   1000000

        14 Country                      5
Country
    100000
```

```
SQL> select * from userlikessong;

    USERID     SONGID
---------- ----------
         0          0
         0          1
         2         53
        52          0
        69          0
```

```
SQL> select * from songhaskeyword;

 KEYWORDID KEYWORD                          SONGID
---------- ------------------------------ ----------
         0 sociopath                               0
         1 apathy                                  0
         2 turn                                   53
         3 me                                     53
         4 on                                     53
```

```
SQL> select * from userplaylists;

PLAYLISTID
----------
         0
         1
         2
         3
       444
```

```
 SQL> select * from userhasuserplaylists;

    USERID PLAYLISTID
---------- ----------
         0          0
         0          1
         2          2
        52          3
        52        444
```

```
SQL> select * from playlistcontent;

 CONTENTID PLAYLISTID
---------- ----------
         0          0
         1          1
         2          2
         3          3
       444        444
```

```
SQL> select * from playlistcontentincludessong;

 CONTENTID     SONGID
---------- ----------
         0          0
         0          1
         2         47
         2         53
       444         53
```

```
SQL> select * from streamalbumcertification;

NUMSTREAMS CERTIFICATION
---------- ------------------------------
  10000000 Diamond
   2000000 Multi-Platinum
    500000 Gold
   1000000 Platinum
    100000 None
```

```
SQL> select * from titlerelease;

TITLE                          ALBUMVERSION RELEASEDA   NUMSONGS
------------------------------ ------------ --------- ----------
032 Funk                                  1 01-APR-21          5
30                                        1 16-MAY-20         10
Light Switch                              1 30-MAR-22          1
Pigeon!                                   1 25-FEB-22          8
ZZZ                                       1 16-NOV-18          9
```

```
SQL> select * from album;

   ALBUMID TITLE                ALBUMVERSION NUMSTREAMS
---------- -------------------- ------------ ----------
         0 032 Funk                        1    2000000
         1 30                              1   10000000
         2 Light Switch                    1     500000
         5 Pigeon!                         1     100000
        99 ZZZ                             1    1000000
```

```
SQL> select * from albumhaskeyword;

 KEYWORDID KEYWORD                          ALBUMID
---------- ------------------------------ ----------
         0 Funk                                    0
         1 Celebration                             0
         2 Incheon Airport Freestyle               0
         3 Sad                                     1
         4 Upbeat                                  2
```

```
SQL> select * from albumincludessong;

    ALBUMID     SONGID
---------- ----------
         0          0
         0          1
         1         47
         2         14
         2         53
```

```
SQL> select * from artisthasalbum;

    USERID    ALBUMID
---------- ----------
         0          0
         0          1
         3          2
         4          5
         5         99
```