DBMS PROJECT REPORT

MUSIC DATABASE SYSTEM

AKASH VANKA

B210703EC

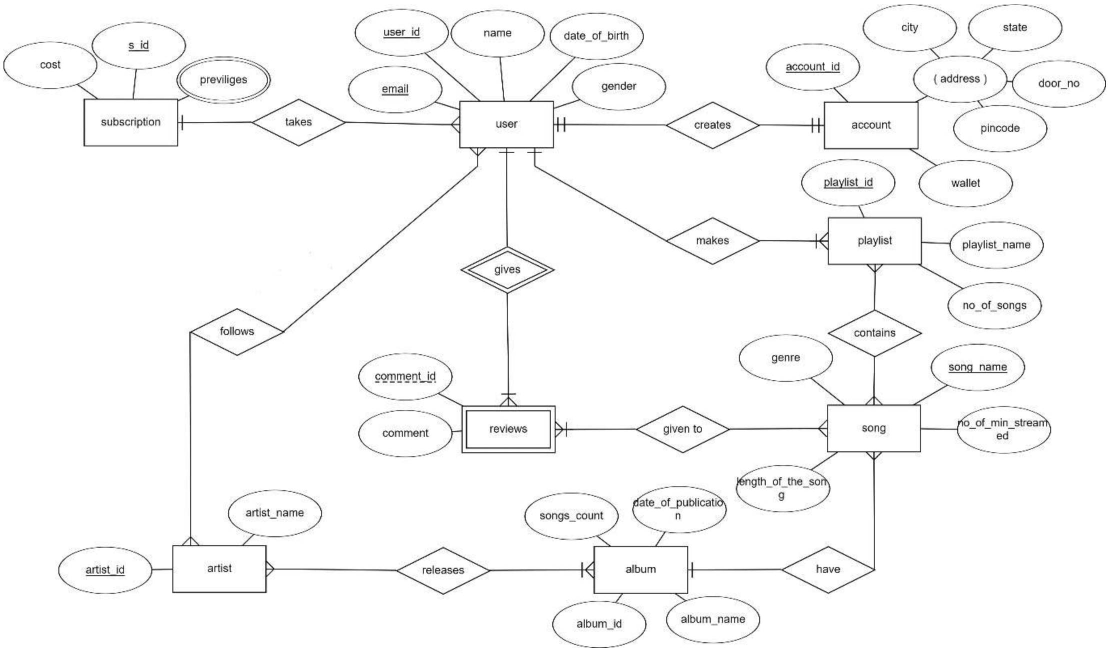
CONTENT

PROBLEM STATEMENT

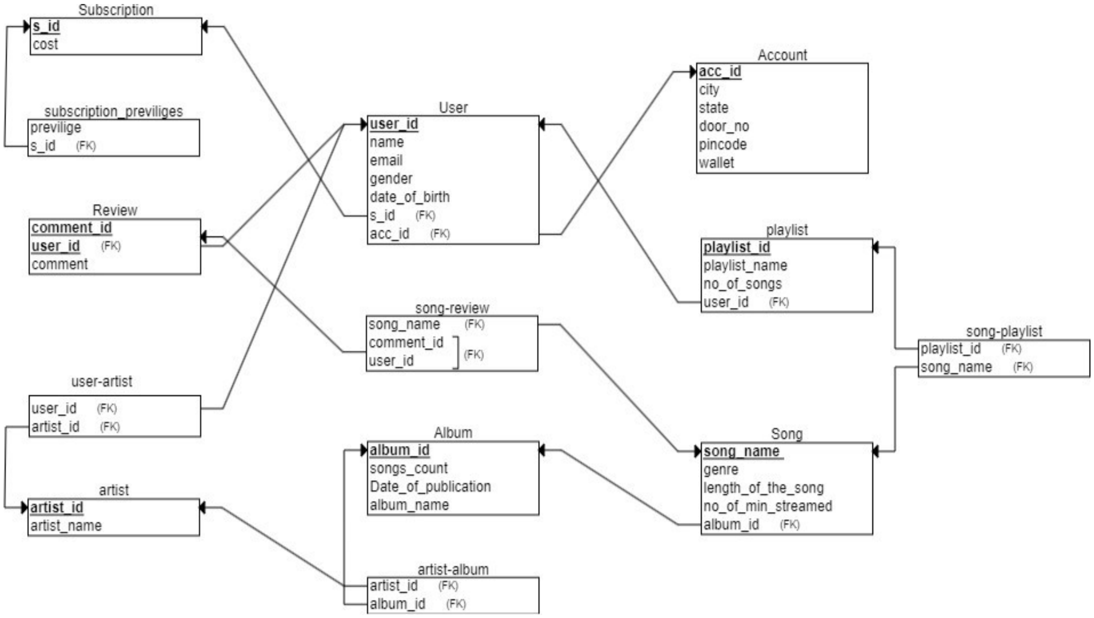
Music plays an important role in this era of busy society. So, we choose to make a Database management project on Music.

A Music Database contains information about music albums and songs played by the users. This includes information about Songs and albums i.e, which album it is from, Singers and composers, Genres of the songs and Release date. This Database also includes data about Most played songs. We would also like to classify songs according to their release times as old and new songs. This kind of classification in database helps users to find their favorite songs in less time.

ER-DIAGRAM


cost
s_id
previliges
subscription
takes
user_id
name
date_of_birth
email
gender
user
creates
account_id
account
city
state
(address)
door_no
pincode
wallet
playlist_id
makes
playlist
playlist_name
no_of_songs
gives
follows
contains
genre
song_name
comment_id
comment
reviews
given to
song
no_of_min_stream ed
length_of_the_son g
artist_name
songs_count
date_of_publicatio n
artist_id
artist
releases
album
have
album_id
album_name

RELATIONAL SCHEMA



ASSUMPTIONS

1. We assumed that one user has one account.
2. We assumed that one user can have only one subscription but one subscription can be taken by many users.
3. We assumed that one user can give any number of comments to a song but one comment can be given by single user.
4. We assumed that one user can follow any number of artists and an artist can have any number of followers(users).
5. We assumed that any number of reviews can be given to any number of songs.
6. We assumed that a song can be in any number of playlists created by user and a playlist can have any number of songs.
7. We assumed that a user can create any number of playlists but a playlist must be created by a single user.
8. We assumed that an album can have any number of songs but a song must belong to single album.
9. We assumed that an album can be sung by any number of artists and an artist can sing in many albums.

ENTITIES & RELATIONSHIPS

1. **User** :

- *This entity represents a user who was browsing the database*
- *This entity has user_id, name, date_of_birth, gender, email attributes.*
- *user_id is the primary key.*

2. **Account :**

- *Users creates an account to buy a subscription.*
- *This entity has account_id, wallet, address(composite attribute) attributes.*
- *account_id is the primary key.*
- *It is in 'creates' relationship with User entity.*

3. **Subscription:**

- *This entity specifies about the subscription bought by user using account.*
- *This entity has cost, s_id, previliges attributes.*
- *s_id is the primary key.*
- *It is in 'takes' relationship with User.*

4. **Reviews***:*

- *This entity stores information about a review given by a user to a song.*
- *It is a weak entity with comment_id and comment as attributes.*
- *It is in 'gives' relationship with strong entity User to form primary key (user_id,comment_id).*

5. ***Playlist:***

- *This entity stores information of playlist created by a user.*
- *It has playlist_id, playlist_name and no_of_songs as attributes.*
- *Playlist_id is the primary key.*
- *It is in 'makes' relationship with user.*

6. ***Song:***

- *This entity stores information of a song.*
- *It has song_name, genre and no_of_min_streamed as attributes.*
- *song_name is the primary key.*
- *It is in 'contains' relationship with playlist and 'given to' relationship with review.*

7. **Album:**

- *This entity stores information of song albums and songs in them.*
- *It has songs_count, date_of_publication, album_name, album_id attributes.*
- *album_id is the primary key.*
- *It is in 'have' relationship with song.*

8. **Artist:**

- *This entity stores information of the artist.*
- *It has artist_name and artist_id attributes.*
- *artist_id is the primary key.*
- *It is in 'releases' relationship with album and 'follows' relationship with User.*

NORMALIZATION

1. User:

user_id->(name, date_of_birth, gender, email,account_id,s_id)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, user_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. user_id to all other attributes so the table is in BCNF.

2. Account:

account_id->(account_id, wallet, address)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, account_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. account_id to all other attributes so the table is in BCNF.

3.Subscription:

S_id->(cost)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, s_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. s_id to all other attributes so the table is in BCNF.

4.subscription-previliges:

s_id->(previliges)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, s_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. s_id to all other attributes so the table is in BCNF.

5.Reviews:

(comment_id,user_id)->(comment)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, *(comment_id,user_id)* and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. *(comment_id,user_id)* to all other attributes so the table is in BCNF.

6.Song:

song_name->(no_of_min_streamed)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, song_name and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. song_name to all other attributes so the table is in BCNF.

7. Playlist:

Playlist_id->(playlist_name , no_of_songs)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, playlist_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. playlist_id to all other attributes so the table is in BCNF.

8. Album:

album_id->(songs_count, date_of_publication, album_name)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, album_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. album_id to all other attributes so the table is in BCNF.

9. Artist:

artist_id->(artist_name)

1NF:As the table contains primary key and all the attributes are atomic attributes and there is no multivalued attributes so the table is in 1NF.

2NF:In this table there is only one primary key i.e, artist_id and it is only single attribute so there is no partial dependency so the table is in 2NF.

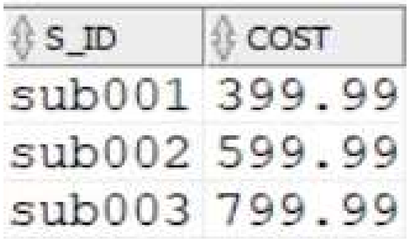
3NF:In this table all functional dependencies are from candidatekey(primeattribute) to non prime attributes.So There is no transitive dependency so the table is in 3NF.

BCNF:Here all Functional dependencies are from super key i.e. artist_id to all other attributes so the table is in BCNF.

TABLE CREATIONS

Subscription:

```
create table subscription(
s_id varchar(256) not null primary key,
cost number
);

INSERT INTO subscription (s_id, cost)
VALUES ('sub001', 399.99);
INSERT INTO subscription (s_id, cost)
VALUES ('sub002', 599.99);
INSERT INTO subscription (s_id, cost)
VALUES ('sub003', 799.99);

select * from subscription;
```

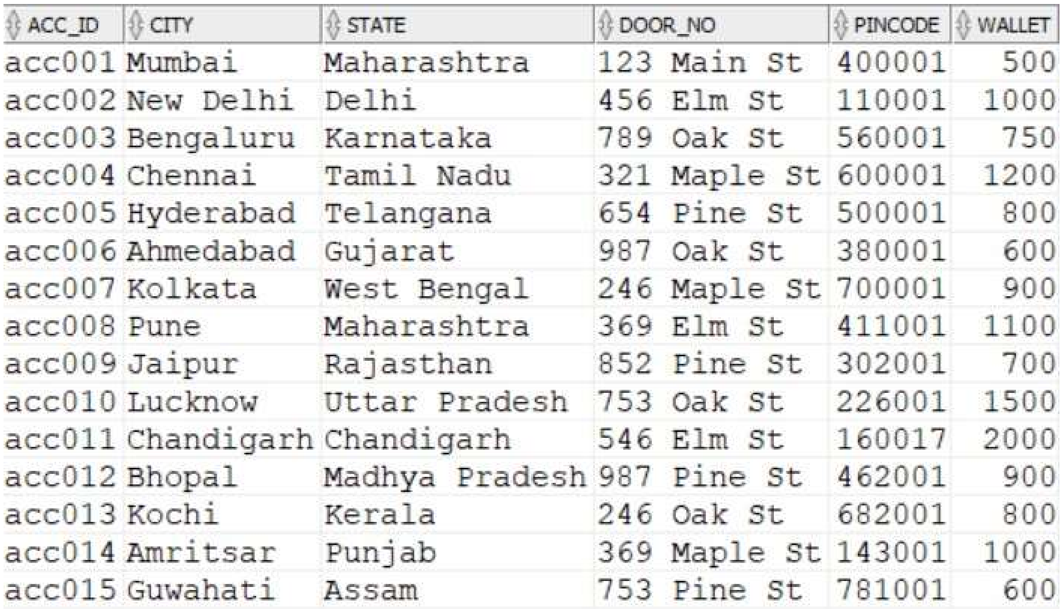


S_ID	COST
sub001	399.99
sub002	599.99
sub003	799.99

Account:

```
acc_id varchar(256) not null primary key,
city varchar(256),
state varchar(256),
door_no varchar(256),
pincode number,
wallet number
);

INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc001', 'Mumbai', 'Maharashtra', '123 Main St', 400001,
500.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc002', 'New Delhi', 'Delhi', '456 Elm St', 110001, 1000.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc003', 'Bengaluru', 'Karnataka', '789 Oak St', 560001,
750.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc004', 'Chennai', 'Tamil Nadu', '321 Maple St', 600001,
1200.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc005', 'Hyderabad', 'Telangana', '654 Pine St', 500001,
800.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc006', 'Ahmedabad', 'Gujarat', '987 Oak St', 380001,
600.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc007', 'Kolkata', 'West Bengal', '246 Maple St', 700001,
900.00);
```

```
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc008', 'Pune', 'Maharashtra', '369 Elm St', 411001,
1100.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc009', 'Jaipur', 'Rajasthan', '852 Pine St', 302001, 700.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc010', 'Lucknow', 'Uttar Pradesh', '753 Oak St', 226001,
1500.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc011', 'Chandigarh', 'Chandigarh', '546 Elm St', 160017,
2000.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc012', 'Bhopal', 'Madhya Pradesh', '987 Pine St', 462001,
900.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc013', 'Kochi', 'Kerala', '246 Oak St', 682001, 800.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc014', 'Amritsar', 'Punjab', '369 Maple St', 143001,
1000.00);
INSERT INTO account (acc_id, city, state, door_no, pincode, wallet)
VALUES ('acc015', 'Guwahati', 'Assam', '753 Pine St', 781001, 600.00);

select * from account;
```



ACC_ID	CITY	STATE	DOOR_NO	PINCODE	WALLET
acc001	Mumbai	Maharashtra	123 Main St	400001	500
acc002	New Delhi	Delhi	456 Elm St	110001	1000
acc003	Bengaluru	Karnataka	789 Oak St	560001	750
acc004	Chennai	Tamil Nadu	321 Maple St	600001	1200
acc005	Hyderabad	Telangana	654 Pine St	500001	800
acc006	Ahmedabad	Gujarat	987 Oak St	380001	600
acc007	Kolkata	West Bengal	246 Maple St	700001	900
acc008	Pune	Maharashtra	369 Elm St	411001	1100
acc009	Jaipur	Rajasthan	852 Pine St	302001	700
acc010	Lucknow	Uttar Pradesh	753 Oak St	226001	1500
acc011	Chandigarh	Chandigarh	546 Elm St	160017	2000
acc012	Bhopal	Madhya Pradesh	987 Pine St	462001	900
acc013	Kochi	Kerala	246 Oak St	682001	800
acc014	Amritsar	Punjab	369 Maple St	143001	1000
acc015	Guwahati	Assam	753 Pine St	781001	600

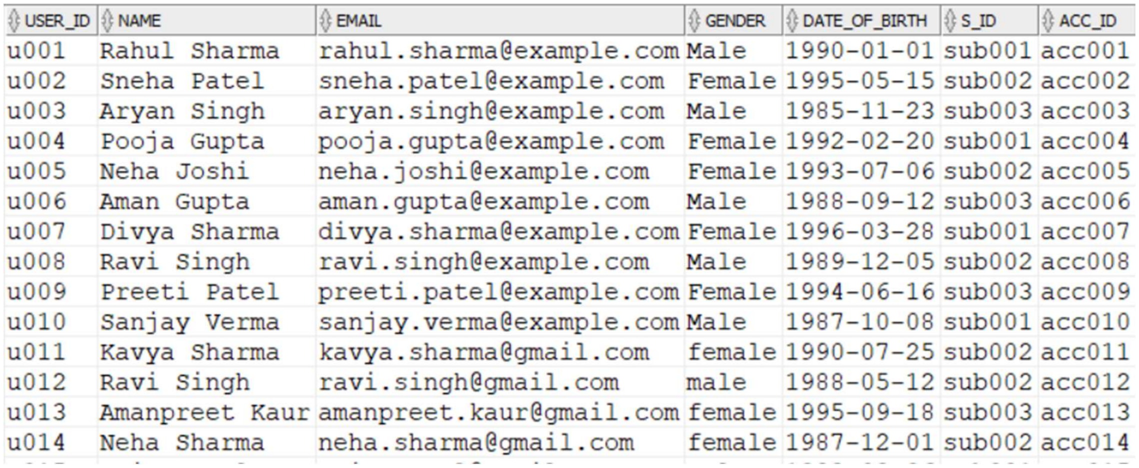
User:

```
create table userr(

user_id varchar(256) not null primary key,

name varchar(256),

email varchar(256),

gender varchar(256),

date_of_birth varchar(256),

s_id varchar(256),

acc_id varchar(256),

foreign key (s_id) references subscription(s_id) on delete cascade,

foreign key (acc_id) references account(acc_id) on delete cascade

);

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u001', 'Rahul Sharma', 'rahul.sharma@example.com',
'Male', '1990-01-01', 'sub001', 'acc001');
```

```sql
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u002', 'Sneha Patel', 'sneha.patel@example.com', 'Female',
'1995-05-15', 'sub002', 'acc002');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u003', 'Aryan Singh', 'aryan.singh@example.com', 'Male',
'1985-11-23', 'sub003', 'acc003');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u004', 'Pooja Gupta', 'pooja.gupta@example.com',
'Female', '1992-02-20', 'sub001', 'acc004');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u005', 'Neha Joshi', 'neha.joshi@example.com', 'Female',
'1993-07-06', 'sub002', 'acc005');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u006', 'Aman Gupta', 'aman.gupta@example.com', 'Male',
'1988-09-12', 'sub003', 'acc006');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u007', 'Divya Sharma', 'divya.sharma@example.com',
'Female', '1996-03-28', 'sub001', 'acc007');

INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id,
acc_id)

VALUES ('u008', 'Ravi Singh', 'ravi.singh@example.com', 'Male',
'1989-12-05', 'sub002', 'acc008');
```

```sql
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u009', 'Preeti Patel', 'preeti.patel@example.com', 'Female', '1994-06-16', 'sub003', 'acc009');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u010', 'Sanjay Verma', 'sanjay.verma@example.com', 'Male', '1987-10-08', 'sub001', 'acc010');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u011', 'Kavya Sharma', 'kavya.sharma@gmail.com', 'female', '1990-07-25', 'sub002', 'acc011');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u012', 'Ravi Singh', 'ravi.singh@gmail.com', 'male', '1988-05-12', 'sub002', 'acc012');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u013', 'Amanpreet Kaur', 'amanpreet.kaur@gmail.com', 'female', '1995-09-18', 'sub003', 'acc013');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u014', 'Neha Sharma', 'neha.sharma@gmail.com', 'female', '1987-12-01', 'sub002', 'acc014');
INSERT INTO userr (user_id, name, email, gender, date_of_birth, s_id, acc_id)
VALUES ('u015', 'Amit Patel', 'amit.patel@gmail.com', 'male', '1992-03-06', 'sub001', 'acc015');
```

```
select * from userr;
```

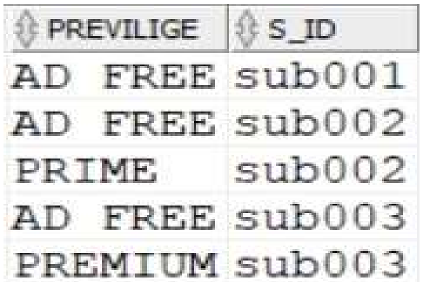


USER_ID	NAME	EMAIL	GENDER	DATE_OF_BIRTH	S_ID	ACC_ID
u001	Rahul Sharma	rahul.sharma@example.com	Male	1990-01-01	sub001	acc001
u002	Sneha Patel	sneha.patel@example.com	Female	1995-05-15	sub002	acc002
u003	Aryan Singh	aryan.singh@example.com	Male	1985-11-23	sub003	acc003
u004	Pooja Gupta	pooja.gupta@example.com	Female	1992-02-20	sub001	acc004
u005	Neha Joshi	neha.joshi@example.com	Female	1993-07-06	sub002	acc005
u006	Aman Gupta	aman.gupta@example.com	Male	1988-09-12	sub003	acc006
u007	Divya Sharma	divya.sharma@example.com	Female	1996-03-28	sub001	acc007
u008	Ravi Singh	ravi.singh@example.com	Male	1989-12-05	sub002	acc008
u009	Preeti Patel	preeti.patel@example.com	Female	1994-06-16	sub003	acc009
u010	Sanjay Verma	sanjay.verma@example.com	Male	1987-10-08	sub001	acc010
u011	Kavya Sharma	kavya.sharma@gmail.com	female	1990-07-25	sub002	acc011
u012	Ravi Singh	ravi.singh@gmail.com	male	1988-05-12	sub002	acc012
u013	Amanpreet Kaur	amanpreet.kaur@gmail.com	female	1995-09-18	sub003	acc013
u014	Neha Sharma	neha.sharma@gmail.com	female	1987-12-01	sub002	acc014

subscription_previliges:

```
create table subscription_previliges(

previlige varchar(256),

s_id varchar(256),

foreign key (s_id) references subscription(s_id) on delete cascade

);


INSERT INTO subscription_previliges(previlige, s_id)

VALUES ('AD FREE','sub001');

INSERT INTO subscription_previliges(previlige, s_id)

VALUES ('AD FREE','sub002');

INSERT INTO subscription_previliges(previlige, s_id)

VALUES ('PRIME','sub002');

INSERT INTO subscription_previliges(previlige, s_id)

VALUES ('AD FREE','sub003');
```

```
INSERT INTO subscription_previliges(previlige, s_id)
VALUES ('PREMIUM','sub003');

SELECT * FROM subscription_previliges;
```

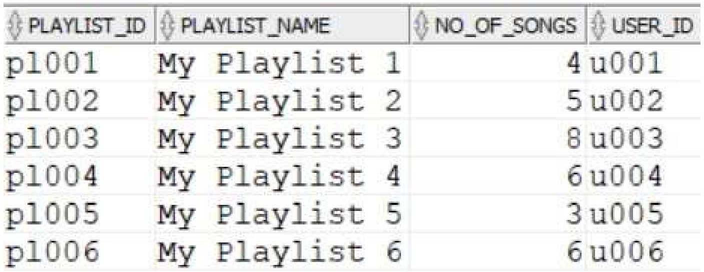


PREVILIGE	S_ID
AD FREE	sub001
AD FREE	sub002
PRIME	sub002
AD FREE	sub003
PREMIUM	sub003

playlist:

```
create table playlist(
playlist_id varchar(256) not null primary key,
playlist_name varchar(256),
no_of_songs number,
user_id varchar(256),
foreign key (user_id) references userr(user_id) on delete cascade
);

INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)
VALUES ('pl001', 'My Playlist 1', 4, 'u001');
INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)
VALUES ('pl002', 'My Playlist 2', 5, 'u002');
```

```
INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)

VALUES ('pl003', 'My Playlist 3', 8, 'u003');

INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)

VALUES ('pl004', 'My Playlist 4', 6, 'u004');

INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)

VALUES ('pl005', 'My Playlist 5', 3, 'u005');

INSERT INTO playlist (playlist_id, playlist_name, no_of_songs, user_id)

VALUES ('pl006', 'My Playlist 6', 6, 'u006');


select * from playlist;
```

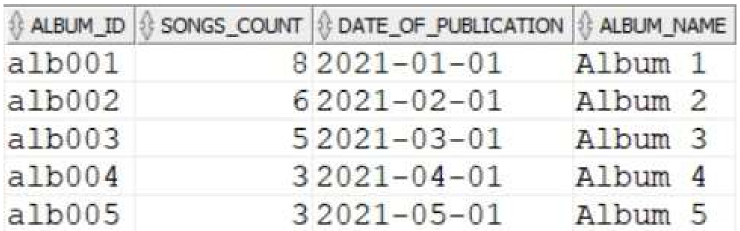


PLAYLIST_ID	PLAYLIST_NAME	NO_OF_SONGS	USER_ID
pl001	My Playlist 1	4	u001
pl002	My Playlist 2	5	u002
pl003	My Playlist 3	8	u003
pl004	My Playlist 4	6	u004
pl005	My Playlist 5	3	u005
pl006	My Playlist 6	6	u006

album:

```
create table album(

album_id varchar(256) not null primary key,

songs_count number,

date_of_publication varchar(256),

album_name varchar(256)

);
```

```
insert into album(album_id, songs_count, date_of_publication, album_name)

values ('alb001', 8, '2021-01-01', 'Album 1');

insert into album(album_id, songs_count, date_of_publication, album_name)

values ('alb002', 6, '2021-02-01', 'Album 2');

insert into album(album_id, songs_count, date_of_publication, album_name)

values ('alb003', 5, '2021-03-01', 'Album 3');

insert into album(album_id, songs_count, date_of_publication, album_name)

values ('alb004', 3, '2021-04-01', 'Album 4');

insert into album(album_id, songs_count, date_of_publication, album_name)

values ('alb005', 3, '2021-05-01', 'Album 5');

select * from album;
```



ALBUM_ID	SONGS_COUNT	DATE_OF_PUBLICATION	ALBUM_NAME
alb001	8	2021-01-01	Album 1
alb002	6	2021-02-01	Album 2
alb003	5	2021-03-01	Album 3
alb004	3	2021-04-01	Album 4
alb005	3	2021-05-01	Album 5

song:

```
create table song(

song_name varchar(256) not null primary key,

genre varchar(256),

length_of_the_song varchar(256),
no_of_min_streamed varchar(256),
album_id varchar(256),
foreign key (album_id) references album(album_id) on delete
cascade
);

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Vachinde', 'Mass', '4:31', '5000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Inkem Inkem', 'Melody', '4:28', '7500000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Samajavaragamana', 'Melody', '3:29', '10000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Choosi Chudangane', 'Love', '3:42', '6000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Ninnu Kori Varanam', 'Love', '4:24', '4500000', 'alb001');
```

```sql
INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Butta Bomma', 'Mass', '3:14', '15000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Ramuloo Ramulaa', 'Mass', '4:23', '8000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Dimaak Kharaab', 'Mass', '4:34', '6000000', 'alb001');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Undiporaadhey', 'Love', '4:59', '6500000', 'alb002');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Seetha Kalyanam', 'Melody', '3:56', '3000000', 'alb002');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Vellipomaakey', 'Melody', '3:41', '5500000', 'alb002');
```

```
INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Naa Pranam', 'Love', '4:12', '4000000', 'alb002');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Nee Kallalona', 'Love', '4:31', '3500000', 'alb002');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Adiga Adiga', 'Melody', '4:30', '5000000', 'alb002');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Inkem Inkem Inkem Kaavaale', 'Melody', '4:30', '2000000',
'alb003');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Top Lesi Poddi', 'Mass Song', '4:10', '900000', 'alb003');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Maate Vinadhuga', 'Melody', '4:09', '1000000', 'alb003');
```

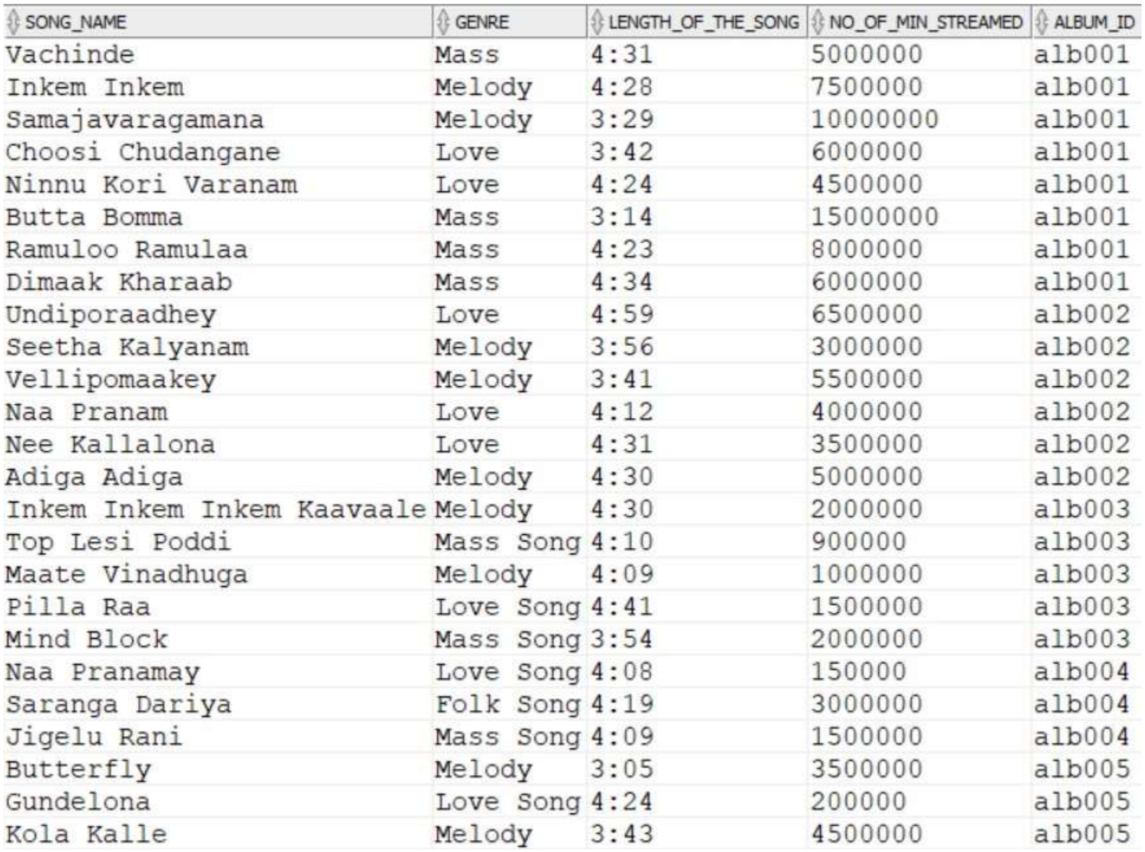
```sql
INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Pilla Raa', 'Love Song', '4:41', '1500000', 'alb003');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Mind Block', 'Mass Song', '3:54', '2000000', 'alb003');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Naa Pranamay', 'Love Song', '4:08', '150000', 'alb004');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Saranga Dariya', 'Folk Song', '4:19', '3000000', 'alb004');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Jigelu Rani', 'Mass Song', '4:09', '1500000', 'alb004');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
VALUES ('Butterfly', 'Melody', '3:05', '3500000', 'alb005');

INSERT INTO song (song_name, genre, length_of_the_song,
no_of_min_streamed, album_id)
```

VALUES ('Gundelona', 'Love Song', '4:24', '200000', 'alb005');

INSERT INTO song (song_name, genre, length_of_the_song, no_of_min_streamed, album_id)

VALUES ('Kola Kalle', 'Melody', '3:43', '4500000', 'alb005');

select * from song;



SONG_NAME	GENRE	LENGTH_OF_THE_SONG	NO_OF_MIN_STREAMED	ALBUM_ID
Vachinde	Mass	4:31	5000000	alb001
Inkem Inkem	Melody	4:28	7500000	alb001
Samajavaragamana	Melody	3:29	10000000	alb001
Choosi Chudangane	Love	3:42	6000000	alb001
Ninnu Kori Varanam	Love	4:24	4500000	alb001
Butta Bomma	Mass	3:14	15000000	alb001
Ramuloo Ramulaa	Mass	4:23	8000000	alb001
Dimaak Kharaab	Mass	4:34	6000000	alb001
Undiporaadhey	Love	4:59	6500000	alb002
Seetha Kalyanam	Melody	3:56	3000000	alb002
Vellipomaakey	Melody	3:41	5500000	alb002
Naa Pranam	Love	4:12	4000000	alb002
Nee Kallalona	Love	4:31	3500000	alb002
Adiga Adiga	Melody	4:30	5000000	alb002
Inkem Inkem Inkem Kaavaale	Melody	4:30	2000000	alb003
Top Lesi Poddi	Mass Song	4:10	900000	alb003
Maate Vinadhuga	Melody	4:09	1000000	alb003
Pilla Raa	Love Song	4:41	1500000	alb003
Mind Block	Mass Song	3:54	2000000	alb003
Naa Pranamay	Love Song	4:08	150000	alb004
Saranga Dariya	Folk Song	4:19	3000000	alb004
Jigelu Rani	Mass Song	4:09	1500000	alb004
Butterfly	Melody	3:05	3500000	alb005
Gundelona	Love Song	4:24	200000	alb005
Kola Kalle	Melody	3:43	4500000	alb005

review:

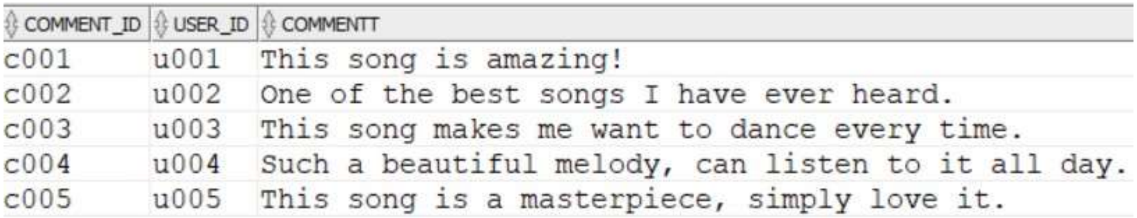
create table review(

comment_id varchar(256),

user_id varchar(256),

commentt varchar(256),

```
primary key (comment_id,user_id),
foreign key (user_id) references userr(user_id) on delete cascade
);

INSERT INTO review (comment_id, user_id, commentt) VALUES
('c001', 'u001', 'This song is amazing!');

INSERT INTO review (comment_id, user_id, commentt) VALUES
('c002', 'u002', 'One of the best songs I have ever heard.');

INSERT INTO review (comment_id, user_id, commentt) VALUES
('c003', 'u003', 'This song makes me want to dance every time.');

INSERT INTO review (comment_id, user_id, commentt) VALUES
('c004', 'u004', 'Such a beautiful melody, can listen to it all day.');

INSERT INTO review (comment_id, user_id, commentt) VALUES
('c005', 'u005', 'This song is a masterpiece, simply love it.');

SELECT * FROM REVIEW;
```

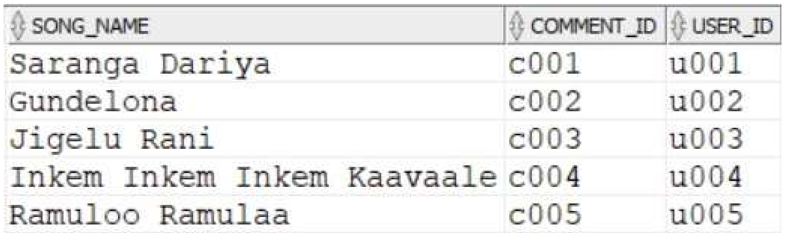


```
COMMENT_ID  USER_ID  COMMENTT
c001        u001     This song is amazing!
c002        u002     One of the best songs I have ever heard.
c003        u003     This song makes me want to dance every time.
c004        u004     Such a beautiful melody, can listen to it all day.
c005        u005     This song is a masterpiece, simply love it.
```

Song_review:

```
create  table  song_review(
song_name   varchar(256),
comment_id varchar(256),
user_id varchar(256),
foreign key (song_name) references song(song_name) on delete
cascade,
foreign key (comment_id,user_id) references
review(comment_id,user_id) on delete cascade
);


INSERT INTO song_review (song_name, comment_id, user_id)
VALUES
('Saranga Dariya', 'c001', 'u001');
INSERT INTO song_review (song_name, comment_id, user_id)
VALUES
('Gundelona', 'c002', 'u002');
INSERT INTO song_review (song_name, comment_id, user_id)
VALUES
('Jigelu Rani', 'c003', 'u003');
INSERT INTO song_review (song_name, comment_id, user_id)
VALUES
('Inkem Inkem Inkem Kaavaale', 'c004', 'u004');
INSERT INTO song_review (song_name, comment_id, user_id)
VALUES
('Ramuloo Ramulaa', 'c005', 'u005');
```

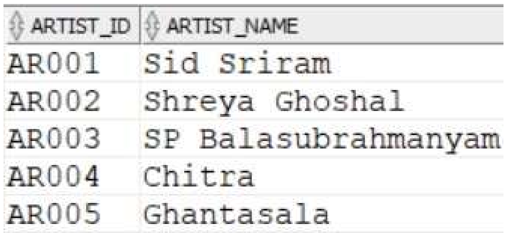
select * from song_review;



SONG_NAME	COMMENT_ID	USER_ID
Saranga Dariya	c001	u001
Gundelona	c002	u002
Jigelu Rani	c003	u003
Inkem Inkem Inkem Kaavaale	c004	u004
Ramuloo Ramulaa	c005	u005

artist:

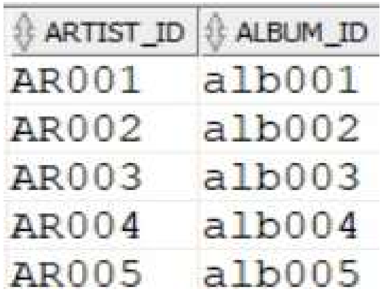
```
create table artist(
artist_id varchar(256) not null primary key,
artist_name varchar(256)
);

insert into artist (artist_id, artist_name) values
('AR001', 'Sid Sriram');
insert into artist (artist_id, artist_name) values
('AR002', 'Shreya Ghoshal');
insert into artist (artist_id, artist_name) values
('AR003', 'SP Balasubrahmanyam');
insert into artist (artist_id, artist_name) values
('AR004', 'Chitra');
insert into artist (artist_id, artist_name) values
('AR005', 'Ghantasala');
SELECT * FROM ARTIST;
```



ARTIST_ID	ARTIST_NAME
AR001	Sid Sriram
AR002	Shreya Ghoshal
AR003	SP Balasubrahmanyam
AR004	Chitra
AR005	Ghantasala

Artist_album:

```
create table artist_album(
artist_id varchar(256),
album_id varchar(256),
foreign key (artist_id) references artist(artist_id) on delete cascade,
foreign key (album_id) references album(album_id) on delete cascade
);


INSERT INTO artist_album (artist_id, album_id) VALUES
('AR001', 'alb001');
INSERT INTO artist_album (artist_id, album_id) VALUES
('AR002', 'alb002');
INSERT INTO artist_album (artist_id, album_id) VALUES
('AR003', 'alb003');
INSERT INTO artist_album (artist_id, album_id) VALUES
('AR004', 'alb004');
INSERT INTO artist_album (artist_id, album_id) VALUES
('AR005', 'alb005');
SELECT * FROM ARTIST_ALBUM;
```



ARTIST_ID	ALBUM_ID
AR001	alb001
AR002	alb002
AR003	alb003
AR004	alb004
AR005	alb005

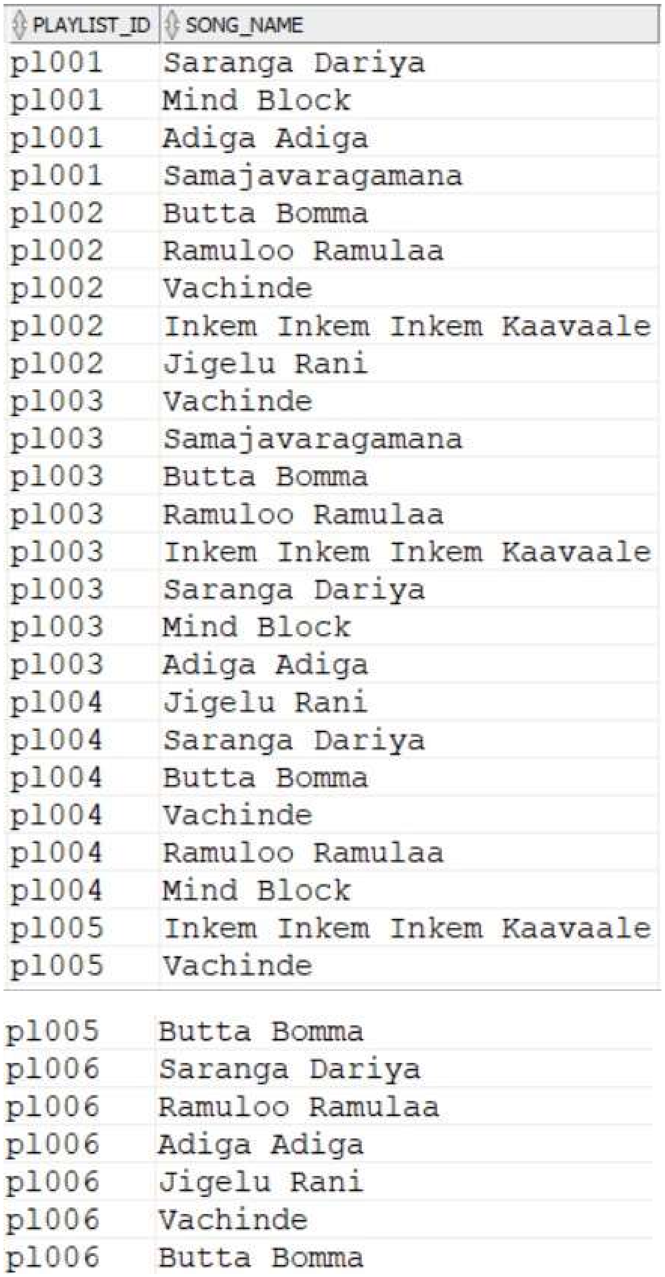
Song_playlist:

```
create table song_playlist(
playlist_id varchar(256),
song_name varchar(256),
foreign key (playlist_id) references playlist(playlist_id) on delete cascade,
foreign key (song_name) references song(song_name) on delete cascade
);


INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl001', 'Saranga Dariya');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl001', 'Mind Block');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl001', 'Adiga Adiga');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl001', 'Samajavaragamana');


INSERT INTO song_playlist (playlist_id, song_name)
```

```
VALUES ('pl002', 'Butta Bomma');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl002', 'Ramuloo Ramulaa');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl002', 'Vachinde');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl002', 'Inkem Inkem Inkem Kaavaale');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl002', 'Jigelu Rani');

INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Vachinde');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Samajavaragamana');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Butta Bomma');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Ramuloo Ramulaa');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Inkem Inkem Inkem Kaavaale');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Saranga Dariya');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Mind Block');
```

```
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl003', 'Adiga Adiga');

INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Jigelu Rani');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Saranga Dariya');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Butta Bomma');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Vachinde');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Ramuloo Ramulaa');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl004', 'Mind Block');

INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl005', 'Inkem Inkem Inkem Kaavaale');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl005', 'Vachinde');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl005', 'Butta Bomma');

INSERT INTO song_playlist (playlist_id, song_name)
```

```
VALUES ('pl006', 'Saranga Dariya');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl006', 'Ramuloo Ramulaa');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl006', 'Adiga Adiga');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl006', 'Jigelu Rani');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl006', 'Vachinde');
INSERT INTO song_playlist (playlist_id, song_name)
VALUES ('pl006', 'Butta Bomma');

select * from song_playlist;
```



PLAYLIST_ID	SONG_NAME
pl001	Saranga Dariya
pl001	Mind Block
pl001	Adiga Adiga
pl001	Samajavaragamana
pl002	Butta Bomma
pl002	Ramuloo Ramulaa
pl002	Vachinde
pl002	Inkem Inkem Inkem Kaavaale
pl002	Jigelu Rani
pl003	Vachinde
pl003	Samajavaragamana
pl003	Butta Bomma
pl003	Ramuloo Ramulaa
pl003	Inkem Inkem Inkem Kaavaale
pl003	Saranga Dariya
pl003	Mind Block
pl003	Adiga Adiga
pl004	Jigelu Rani
pl004	Saranga Dariya
pl004	Butta Bomma
pl004	Vachinde
pl004	Ramuloo Ramulaa
pl004	Mind Block
pl005	Inkem Inkem Inkem Kaavaale
pl005	Vachinde
pl005	Butta Bomma
pl006	Saranga Dariya
pl006	Ramuloo Ramulaa
pl006	Adiga Adiga
pl006	Jigelu Rani
pl006	Vachinde
pl006	Butta Bomma

user_artist:

create table user_artist(

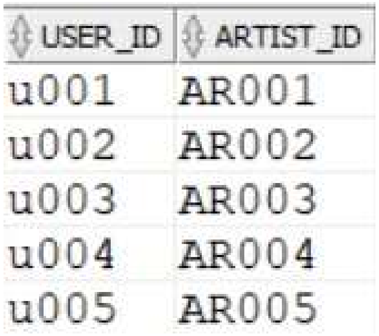
user_id varchar(256),

artist_id varchar(256),

foreign key (user_id) references userr(user_id) on delete cascade,

foreign key (artist_id) references artist(artist_id) on delete cascade

```
);

INSERT INTO user_artist (user_id, artist_id)
VALUES ('u001', 'AR001');

INSERT INTO user_artist (user_id, artist_id)
VALUES ('u002', 'AR002');

INSERT INTO user_artist (user_id, artist_id)
VALUES ('u003', 'AR003');

INSERT INTO user_artist (user_id, artist_id)
VALUES ('u004', 'AR004');

INSERT INTO user_artist (user_id, artist_id)
VALUES ('u005', 'AR005');

select * from user_artist;
```



USER_ID	ARTIST_ID
u001	AR001
u002	AR002
u003	AR003
u004	AR004
u005	AR005

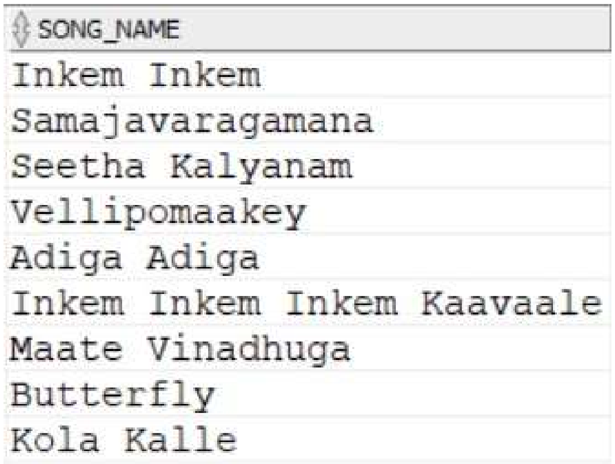
SQL QUERIES

1.Write an SQL query for finding all the songs of the genre MELODY.

QUERY :

SELECT SONG_NAME FROM SONG WHERE GENRE='Melody';

OUTPUT:



```
SONG_NAME
Inkem Inkem
Samajavaragamana
Seetha Kalyanam
Vellipomaakey
Adiga Adiga
Inkem Inkem Inkem Kaavaale
Maate Vinadhuga
Butterfly
Kola Kalle
```

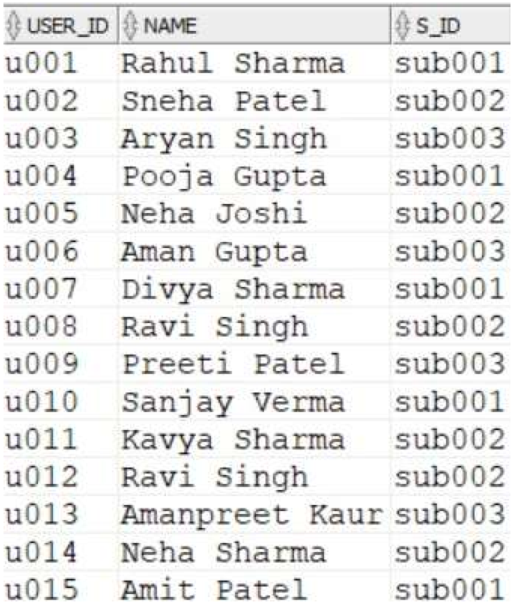
2. . Write an SQL query for finding which user took what subscription.

QUERY :

SELECT USERR.USER_ID,USERR.NAME,SUBSCRIPTION.S_ID FROM USERR

JOIN SUBSCRIPTION ON USERR.S_ID=SUBSCRIPTION.S_ID;

OUTPUT:



USER_ID	NAME	S_ID
u001	Rahul Sharma	sub001
u002	Sneha Patel	sub002
u003	Aryan Singh	sub003
u004	Pooja Gupta	sub001
u005	Neha Joshi	sub002
u006	Aman Gupta	sub003
u007	Divya Sharma	sub001
u008	Ravi Singh	sub002
u009	Preeti Patel	sub003
u010	Sanjay Verma	sub001
u011	Kavya Sharma	sub002
u012	Ravi Singh	sub002
u013	Amanpreet Kaur	sub003
u014	Neha Sharma	sub002
u015	Amit Patel	sub001

3. Write an SQL query for finding which playlists contains the song RAMULO RAMULA.

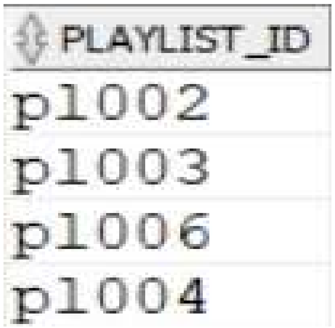
QUERY:

SELECT PLAYLIST_ID FROM SONG_PLAYLIST

WHERE SONG_NAME='Ramuloo Ramulaa'

GROUP BY PLAYLIST_ID HAVING COUNT(*)>=1;

OUTPUT:



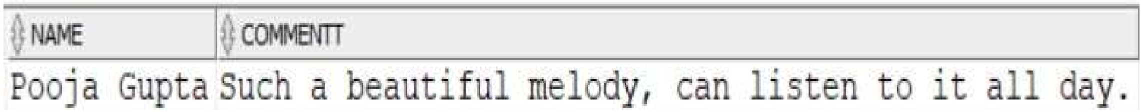
PLAYLIST_ID
pl002
pl003
pl006
pl004

4. Write an SQL query to find which users have written comments on the song INKEM INKEM INKEM KAVALI.

QUERY:

```
SELECT USERR.NAME,REVIEW.COMMENTT
FROM SONG_REVIEW

JOIN REVIEW ON
SONG_REVIEW.COMMENT_ID=REVIEW.COMMEN
T_ID

JOIN USERR ON
REVIEW.USER_ID=USERR.USER_ID

WHERE SONG_NAME='Inkem Inkem Inkem
Kaavaale';
```

OUTPUT:



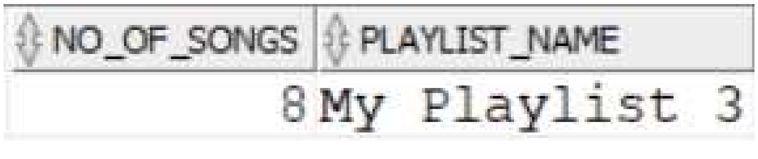
NAME	COMMENTT
Pooja Gupta	Such a beautiful melody, can listen to it all day.

5. Write an SQL query for finding the playlist with max no songs.

QUERY:

```
SELECT NO_OF_SONGS,PLAYLIST_NAME FROM
PLAYLIST

ORDER BY NO_OF_SONGS DESC

OFFSET 0 ROWS FETCH NEXT 1 ROW ONLY;
```

OUTPUT:



NO_OF_SONGS	PLAYLIST_NAME
8	My Playlist 3

THANK YOU