DBMS PROJECT REPORT

MUSIC DATABASE SYSTEM

BY

JASWANTH YERRAMSETTI 21CSB0B68

SUBMITTED TO :

DR.T.RAMAKRISHNUDU

CONTENT

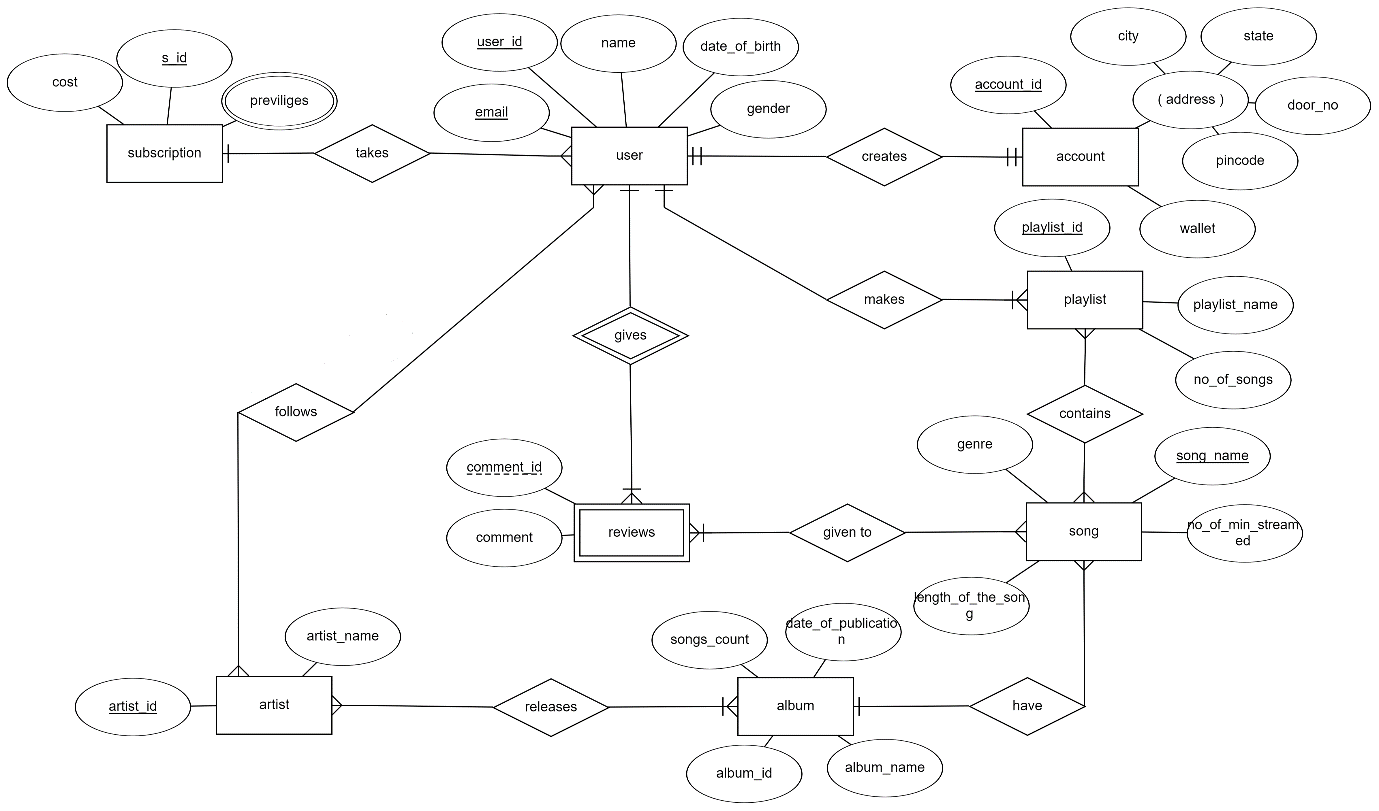
* PROBLEM STATEMENT
* ER DIAGRAM
* RELATIONAL SCHEMA
* ASSUMPTIONS
* ENTITIES AND RELATIONSHIPS
* NORMALIZATION
* TABLES CREATION
* SQL QUERIES

PROBLEM STATEMENT

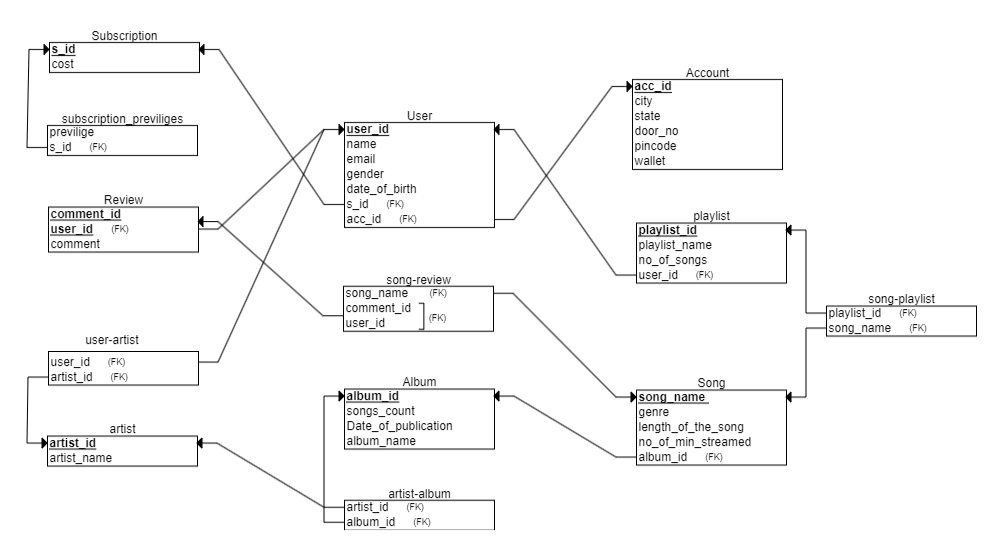
*Music plays an important role in this era of busy society. So, we choose to make a Database management project on Music.*

*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



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.*

1. **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.*

1. **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.*

1. **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).*

1. ***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.*

1. ***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.*

1. **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.*

1. **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);

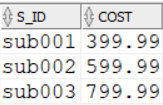
INSERT INTO subscription (s\_id, cost)

VALUES ('sub002', 599.99);

INSERT INTO subscription (s\_id, cost)

VALUES ('sub003', 799.99);

select \* from subscription;



**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);

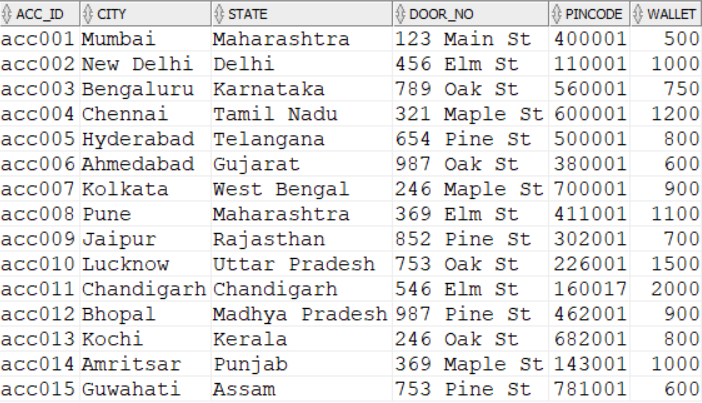
INSERT INTO account (acc\_id, city, state, door\_no, pincode, wallet)

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;



**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');

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');

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;



**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');

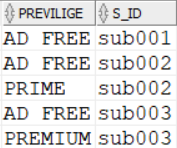
INSERT INTO subscription\_previliges(previlige, s\_id)

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

INSERT INTO subscription\_previliges(previlige, s\_id)

VALUES ('PREMIUM','sub003');

SELECT \* FROM subscription\_previliges;

****

**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');

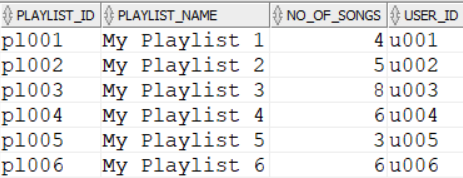
INSERT INTO playlist (playlist\_id, playlist\_name, no\_of\_songs, user\_id)

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;



**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');

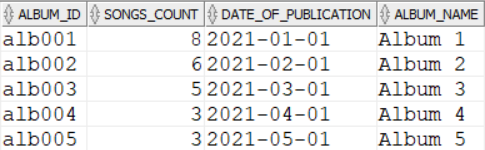
insert into album(album\_id, songs\_count, date\_of\_publication, album\_name)

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;



**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');

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');

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');

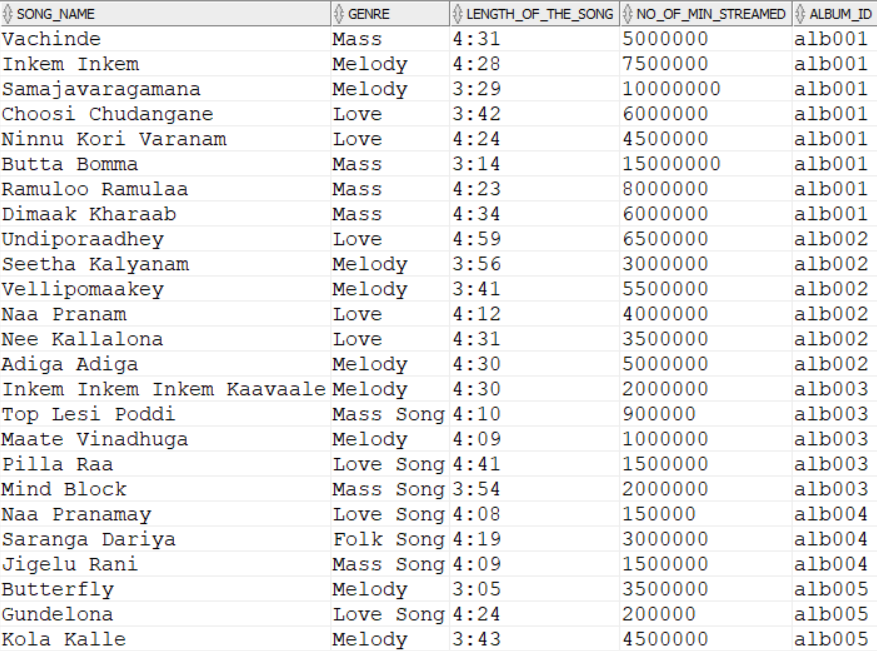
INSERT INTO song (song\_name, genre, length\_of\_the\_song, no\_of\_min\_streamed, album\_id)

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;



**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.');

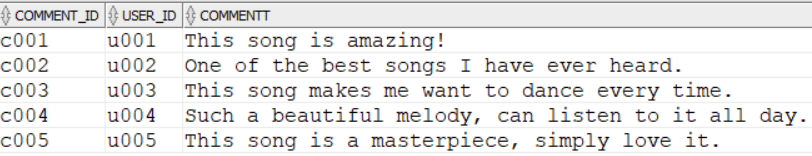
INSERT INTO review (comment\_id, user\_id, commentt) VALUES

('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;



**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');

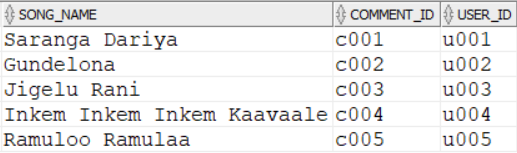
INSERT INTO song\_review (song\_name, comment\_id, user\_id) VALUES

('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;



**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');

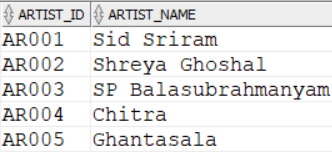
insert into artist (artist\_id, artist\_name) values

('AR004', 'Chitra');

insert into artist (artist\_id, artist\_name) values

('AR005', 'Ghantasala');

SELECT \* FROM ARTIST;



**Artist\_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');

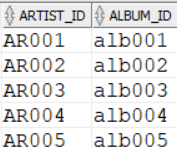
INSERT INTO artist\_album (artist\_id, album\_id) VALUES

('AR004', 'alb004');

INSERT INTO artist\_album (artist\_id, album\_id) VALUES

('AR005', 'alb005');

SELECT \* FROM ARTIST\_ALBUM;



**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');

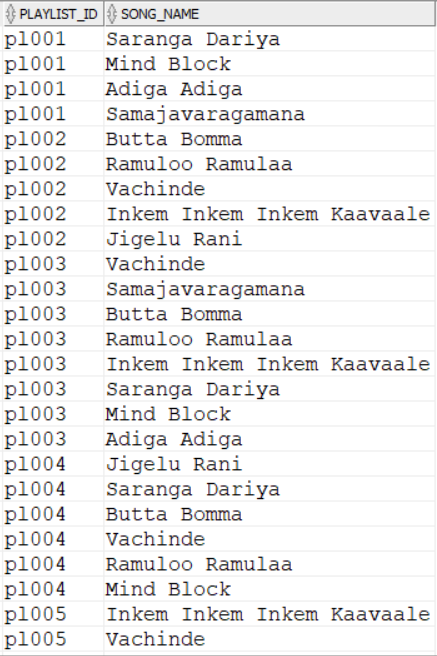
INSERT INTO song\_playlist (playlist\_id, song\_name)

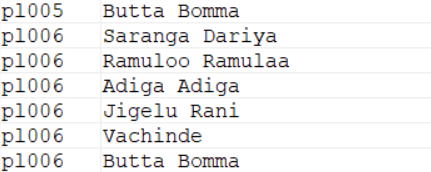
VALUES ('pl006', 'Vachinde');

INSERT INTO song\_playlist (playlist\_id, song\_name)

VALUES ('pl006', 'Butta Bomma');

select \* from song\_playlist;





**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');

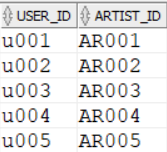
INSERT INTO user\_artist (user\_id, artist\_id)

VALUES ('u004', 'AR004');

INSERT INTO user\_artist (user\_id, artist\_id)

VALUES ('u005', 'AR005');

select \* from user\_artist;



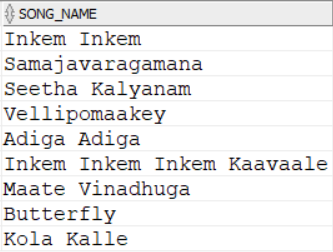
SQL QUERIES

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

QUERY :

SELECT SONG\_NAME FROM SONG WHERE GENRE='Melody';

OUTPUT:



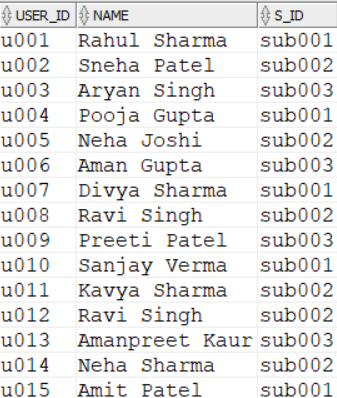
2. . Write an SQL query for finding which user took what subscription.

QUERY :

SELECT USERR.USER\_ID,USERR.NAME,SUBSCRIPTION.S\_ID FROM USERR

JOIN SUBSCRIPTION ON USERR.S\_ID=SUBSCRIPTION.S\_ID;

OUTPUT:



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

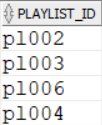
QUERY:

SELECT PLAYLIST\_ID FROM SONG\_PLAYLIST

WHERE SONG\_NAME='Ramuloo Ramulaa'

GROUP BY PLAYLIST\_ID HAVING COUNT(\*)>=1;

OUTPUT:



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.COMMENT\_ID

JOIN USERR ON REVIEW.USER\_ID=USERR.USER\_ID

WHERE SONG\_NAME='Inkem Inkem Inkem Kaavaale';

OUTPUT:



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:



THANK YOU