```
create database db210;
use db210;

create table artist (
	artname varchar(100) primary key);

create table song (
	songname varchar(200) not null,
	artistname varchar(100) not null,
	release_date date not null,
	genre varchar(100) not null,
	foreign key (artistname) references artist(artname),
	CONSTRAINT song_artist PRIMARY KEY (songname, artistname));

create table user (
	username varchar(100) primary key);

create table album (
	albumname varchar(200) not null,
	artistname varchar(100) not null,
	release_date date not null,
	foreign key (artistname) references artist(artname),
	CONSTRAINT album_artist PRIMARY KEY (albumname, artistname));


create table playlist (
	playlistname varchar(200) primary key,
	user_name varchar(100) not null,
	pl_date datetime not null,
	foreign key (user_name) references user(username));

create table playlist_songs (
	song_title varchar(20) not null,
	playlist_name varchar(20) not null,
	foreign key (song_title) references song(songname),
	foreign key (playlist_name) references playlist(playlistname));

create table rating (
	user_name varchar(100) not null,
	album_title varchar(200),
```

```
    song_title varchar(200),
    playlist_title varchar(200),
    ratingdate date not null,
    rating ENUM('1','2','3','4','5') not null,
    foreign key (user_name) references user(username),
    foreign key (album_title) references album(albumname),
    foreign key (song_title) references song(songname),
    foreign key (playlist_title) references playlist(playlistname));

#**********READ ME**********#
#First insert name of artist and users into their single respective tables
#You need to insert a song first before inserting them into anything else
#songs matched to album by release date
#Playlist functionality is split into 2 tables
#First create playlist which is matched to a user
#to put songs into playlist you need to put songs into playlist_song table that matches via playlist name
#need a user before you can insert into rating
#**********READ ME**********#

insert into artist values ("a1");
insert into artist values ("a2");
insert into user values ("u1");
insert into user values ("u2");
insert into user values ("u3");

insert into song values ("a1song1" , "a1", "1999-12-23", "rock");
insert into song values ("a1song2" , "a1", "1999-12-23", "pop");
insert into song values ("a1song3" , "a1", "1990-12-23", "funk");
insert into song values ("a1song4" , "a1", "1994-01-01", "funk");


insert into song values ("a2song1" , "a2", "1990-12-23", "pop");
insert into song values ("a2song2" , "a2", "1999-12-23", "funk");
insert into song values ("a2song3" , "a2", "1990-12-23", "funk");

insert into album values ("a1album1", "a1", "1999-12-23");
insert into album values ("a2album1", "a2", "1999-12-23");

insert into playlist values ("u1playlist1", "u1", "1999-12-23");
```

```
insert into playlist values ("u1playlist2", "u1", "1999-12-23");
insert into playlist values ("u2playlist1", "u2", "1999-12-23");

insert into playlist_songs values ("a1song1", "u1playlist1");
insert into playlist_songs values ("a1song2", "u1playlist1");
insert into playlist_songs values ("a1song2", "u2playlist1");

insert into playlist_songs values ("a1song3", "u1playlist2");
insert into playlist_songs values ("a1song1", "u2playlist1");
insert into playlist_songs values ("a1song1", "u2playlist1");


insert into rating values ("u1", "a1album1", "a1song1", null, "1993-12-23", "4");
insert into rating values ("u3", "a1album1", "a1song1", null, "1993-12-23", "4");
insert into rating values ("u1", "a1album1" , "a1song2", null, "1993-12-23", "5");
insert into rating values ("u2", "a1album1" , "a2song1", null, "1993-12-23", "1");
insert into rating values ("u2", "a2album1" , null, null, "1993-12-23", "1");


-- select * from song;
-- select * from album;
-- select * from rating;
-- select * from playlist;
-- select * from playlist_songs;

#1
SELECT genre, count(*) as number_of_songs
FROM song
GROUP BY genre
ORDER BY count(*) DESC
LIMIT 3;

#2
SELECT DISTINCT song.artistname AS artist_name
FROM song
INNER JOIN album
ON album.artistname = song.artistname
WHERE album.release_date != song.release_date;

#3
```

```
select album_title as album_name,  avg(rating) as average_user_rating
from rating
WHERE ratingdate >= "1990-01-01" AND ratingdate <= "1999-12-31"
group by album_name
ORDER BY avg(rating) DESC, avg(rating) ASC
LIMIT 10;
#4
Select genre as genre_name, count(*) as number_of_song_ratings
from song
inner join rating
ON rating.song_title = song.songname
WHERE rating.ratingdate >= "1991-01-01" AND rating.ratingdate <= "1995-12-31"
group by genre
ORDER BY count(*) DESC
limit 3;
#5
Select playlist.user_name as username, playlistname as playlist_title, avg(rating) as
average_song_rating
from playlist
inner join rating
ON playlist.playlistname = rating.playlist_title
group by playlist_title HAVING avg(rating) > 4.0
order by avg(rating) DESC;
#6
select user_name AS username, (count(rating.album_title) + count(rating.song_title)) as
number_of_ratings
FROM rating
group by username
ORDER BY count(rating) DESC
limit 5;
#7
Select artistname as artist_name, count(*) as number_of_songs
from song
where release_date >= "1990-01-01" AND release_date <= "2010-12-31"
group by artistname
ORDER BY count(*) DESC
limit 10;
#8
Select song_title, count(distinct playlist_name) as number_of_playlist
from playlist_songs
```

```
group by song_title
order by count(distinct playlist_name) DESC, count(distinct playlist_name) ASC
limit 10;
#9
select songname as song_title, song.artistname as artist_name, count(rating)
from song
inner join rating
on songname = rating.song_title
inner join album
on song.artistname = album.artistname
where song.release_date != album.release_date
group by songname
ORDER by count(rating) DESC
limit 20;
#10
select distinct artistname as artist_title
from song
where artistname
not in (select artistname from song where release_date > "1993-12-31");
```