Database Schema

```
Create table artist(
    Id int auto_increment primary key,
    Name varchar(25) not null unique);

Create table album(
    Id int auto_increment primary key,
    Title varchar(20) not null,
    artistId int not null,
    Foreign key (artistId) references artist(id),
    unique(title, artistId),
    releaseDate date not null);

Create table Song(
    Id int auto_increment primary key,
    Title varchar(20) not null,
    artistId int not null,
    Foreign key (artistId) references artist(id),
    unique(title, artistId),
    albumId int,
    Foreign key (albumId) references album(id),
    Genre enum(‘pop’, ‘rap’, ‘jazz’, ‘hip-hop’, ‘rock’, ‘classical’, ‘top hit’) default ‘top hit’,
    releaseDate date,
    Constraint releaseDate_1 check(albumId is null and releaseDate is not null or albumId is
        not null and releaseDate is null)

Create table User(
    Id int auto_increment primary key,
    Name varchar(25) not null unique);

Create table Playlist(
    Id int auto_increment primary key,
    Title varchar(25) not null,
    userId int,
    Foreign key (userId) references user(id),
    unique(title, userId),
    createDate date not null,
    createTime time not null);
```

```
Create table songInPlaylist(
	playlistId int not null,
	Foreign key (playlistId) references playlist(id),
	songId int not null,
	Foreign key (songId) references song(id));

Create table songRating(
	userId int not null,
	Foreign key (userId) references user(id),
	songId int not null,
	Foreign key (songId) references song(id),
	unique(userId, songId),
	Rating int not null check(rating >= 1 and rating <= 5)
	ratingDate date not null);

Create table playlistRating(
	userId int not null,
	Foreign key (userId) references user(id),
	songId int not null,
	Foreign key (playlistId) references playlist(id),
	unique(userId, playlistId),
	Rating int not null check(rating >= 1 and rating <= 5)
	ratingDate date not null);

Create table albumRating(
	userId int not null,
	Foreign key (userId) references user(id),
	albumId int not null,
	Foreign key (albumId) references album(id),
	unique(userId, albumId),
	Rating int not null check(rating >= 1 and rating <= 5)
	ratingDate date not null);
```

Queries

- Query1
    - Select genre, count(*) as 'number_of_songs' from song group by genre order by number_of_songs desc limit 3;
- Query2
    - select artist.name as ‘artist_name’ from artist, song where artist.Id = song.artistId and song.albumId is not null and artist.id in(select artist.id from artist, song where artist.id = song.artistId and song.albumId is null);
- Query3
    - select a.title as album_name, avg(ar.rating) as average_user_rating from album a, albumRating ar where a.id = ar.albumid and year(ar.ratingDate)>=1990 and year(ar.ratingDate) <= 1999 group by album_name order by average_user_rating desc , album_name asc limit 10;
- Query4
    - select s.genre as genre_name, count(*) as number_of_song_ratings from song s, songrating r where s.id = r.songid and year(r.ratingDate) >= 1991 and year(r.ratingDate) <= 1995 group by s.genre order by number_of_song_ratings desc limit 3;
- Query5

    select u.name as username, p.title as playlist_title, avg(sr.rating) as average_song_rating from user u, playlist p, song s, songinplaylist sp, songrating sr where u.id = p.userid and p.id = sp.playlistid and s.id = sp.songid and s.id = sr.songid group by u.name, p.title having avg(sr.rating) > 4;
- Query6

    select user.name as username, sum(num) as number_of_ratings from(select sr.userid, count(*) as num from songrating sr group by sr.userid union select ar.userid, count(*) as num from albumrating ar group by ar.userid) as p, user where user.id = p.userid group by username order by number_of_ratings desc limit 5;
- Query7
    - select artist.name as artist_name, count(*) as number_of_songs from artist, (select title, releaseDate, artistId from song where albumid is null union select distinct song.title, album.releaseDate, album.artistId from song, album where song.albumid = album.id) as p where p.artistId = artist.id and year(p.releaseDate) >= 1990 and year(p.releaseDate) <= 2010 group by artist_name order by number_of_songs desc limit 10;
- Query8
    - select song.title as song_title , count(*) as number_of_plays from song, songinplaylist sp where song.id = sp.songid group by song_title order by number_of_plays desc, song_title asc limit 10;
- Query9

  - select song.title as song_title, artist.name as artist_name, count(*) as number_of_ratings from song, artist, songrating where song.artistid = artist.id and song.id = songrating.songid and song.albumid is null group by song_title, artist_name limit 20;
- Query10
  - select artist.name as artist_title from artist where artist.name not in (select distinct artist.name as artist_title from artist, (select title, releaseDate, artistId from song where albumid is null union select distinct song.title, album.releaseDate, album.artistId from song, album where song.albumid = album.id) as p where artist.id = p.artistId and year(p.releaseDate) > 1993);