Schema

```
create table artist (
        id int auto_increment primary key,
        name varchar(100) not null unique,
        is_individual bit not null
);

create table album (
        id int auto_increment primary key,
        name varchar(100) not null,
        artist_id int not null,
        foreign key(artist_id) references artist(id),
        released_date date not null,
        constraint unique_name (artist_name, name) unique
);

create table song (
        id int auto_increment primary key,
        song_title varchar(100) not null,
        artist_id int not null,
        foreign key(artist_id) references artist(id),
        album_id int,
        foreign key(album_id) references album(id),
        released_date date,
);

create table genre (
        id int auto_increment primary key,
        name varchar(100) not null,
);

create table song_genre (
        id int auto_increment primary key,
        song_id int not null,
        foreign key(song_id) references song(id),
        genre_id int not null,
        foreign key(genre_id) references genre(id),
);
```

```
create table user (
        id int auto_increment primary key,
        username varchar(20) primary key unique
);

create table playlist(
        id int auto_increment primary key,
        title varchar(100) not null,
        created datetime not null,
        user_id int not null,
        foreign key(user_id) references user(id),
        constraint unique_name (user_id, title) unique
);

create table playlist_song (
        id int auto_increment primary key,
        song_id int not null,
        foreign key(song_id) references song(id),
        playlist_id not null,
        foreign key(playlist_id) references playlist(id)
);

create table rating(
        id int auto_increment primary key,
        user_id int not null,
        foreign key(user_id ) references user(id),
        song_id int,
        foreign key(song_id) references song(id),
        playlist_id int,
        foreign key(playlist_id) references playlist(id),
        album_id int,
        foreign key(album_id) references album(id),
        rating_input int check((rating_input > 0) and (rating_input< 6))
);
```

Queries

```
1)  select genre.name, count(song_genre.song_id) as ‘number_of_songs’
    from song_genre
    inner join genre on song_genre.genre_id = genre.id
    group by song_genre.genre_id
    order by ‘number_of_songs’ desc limit 3;

2)  select distinct artist.name
    from song|
    inner join artist on song.artist_id = artist.id
    where(song.album_id is null) and song.artist_id in (select artist_id from album);

3)  select album.name, avg(rating_input) as ‘average_rating’
    from rating
    inner join album on rating.album_id = album.id
    where (year(rating.released_date) between 1990 and 1999) and (rating.album_id is not
    null)
    group by rating.album_id
    order by album.name desc limit 10;

4)  select genre.name, count(rating.song_id) as ‘number_of_songs_rated’
    from rating
    inner join song on rating.song_id = song.song_id
    inner join song_genre on song_genre.song_id = song.id
    inner join genre on song_genre.genre_id = genre.id
    where year(rating.released_date) between 1991 and 1995
    group by song_genre.genre_id
    order by ‘number_of_songs_rated’ desc limit 3;

5)  select user.username, playlist.title, avg(rating.rating_input) as ‘average_song_rating’
    from rating, (select playlist.id, avg(rating.rating_input)
    from playlist_song
    inner join playlist on playlist_song.playlist_id = playlist.id
    inner join rating on playlist_song.song_id = rating.song_id
    where rating.playlist_id is null and rating.album_id is null
    group by rating.song_id) as ‘song_query’
    inner join playlist on rating.playlist_id = playlist.id
    where (playlist_song.playlist_id) = (song_query.playlist_id)
    group by rating.playlist_id;
```

```
6)  select user.username, count(rating.rating_input) as ‘number_of_ratings’
    from rating
    inner join user on rating.user_id = user.id
    where (rating.playlist_id is null)
    group by rating.user_id
    order by ‘number_of_ratings’ desc limit 5;

7)  select artist.name, count(song.id) as ‘number_of_songs’
    from song
    inner join artist on song.artist_id = artist.id
    where (year(song.released_date) between 1990 and 2010)
    group by song.artist_id
    order by ‘number_of_songs’ desc limit 10;

8)  select song.title, count(playlist_song.song_id) as ‘number_of_playlists’
    from playlist_song
    inner join song on playlist_song.song_id = song.id
    group by playlist_song.song_id
    order by ‘number_of_playlists’ desc, song.title asc limit 10;

9)  select song.song_title, artist.name, count(rating.rating_input) as ‘number_of_ratings’
    from song
    inner join artist on song.artist_id = artist.id
    inner join rating on song.song_id = rating.song_id
    where (song.album_id is null)
    group by song.id
    order by ‘number_of_ratings’ desc limit 20;

10) select artist.name
    from song, album
    inner join artist on song.artist_id = artist.id
    group by song.artist_id
    where (max(year(song.released_date)) < 1994) or (max(year(album.released_date)) <
    1994);
```