DATABASE SCHEMA

```
Create database Music;

create table Artist (
        Name varchar(20) primary key not null);

create table Song (
        Song_name varchar(20)  not null,
        Artist_Name varchar(20) not null,
        Album_Name varchar(20),
        Release_date_of_song DATE,
       Genre_ID int not null,
        Primary key(Song_name, Artist_Name),
        Foreign key(Artist_Name) references Artist(Name),
        Foreign key(Album_Name) references Album(Name_of_album),
        Foreign key(Genre_ID) references Genre(ID));

create table Album (
        Name_of_album varchar(20) not null,
        Made_by_artist varchar(20) not null,
        Release_date DATE not null,
        Primary key(Name_of_album, Made_by_artist),
        Foreign key(Made_by_artist) references Artist(Name));

Create table genre(
       ID int primary key not null,
       Type varchar(200) not null);

Create table user(
        Username varchar(20) primary key not null);
```

```
Create table Playlist(
        Username varchar(20),
        Playlist_title varchar(20) not null,
        Date_Time_Creation DATE not null,
        Playlist_ID int primary key not null,
        Foreign key(Username) references user(Username));

Create table Playlist_Song(
        Playlist_ID int not null,
        Song_Title varchar(20) not null,
        Artist varchar(20) not null,
        Foreign key(Song_Title) references Song(Song_name),
        Foreign key(Artist) references Artist(Name),
        Foreign key(Playlist_ID) references Playlist(Playlist_ID));

Create table rating(
        Username varchar(20)
        Rating tinyint,
        Date_of_Rating DATE,
        Album_Name varchar(20),
        Album_Artist varchar(20),
        Song_Title varchar(20),
        Playlist_ID int,
        Foreign key(Username) references User(Username),
        Foreign key(Album_Name ) references Album(Name_of_album),
        Foreign key(Album_Artist) references Artist(Name),
        Foreign key(Song_Title ) references Song(Song_name),
        Foreign key(Playlist_ID) references Playlist(Playlist_ID));
```

QUERIES

1) **Which 3 genres are most represented in terms of number of songs in that genre? The result must have two columns, named genre and number_of_songs.**

```
Select type as genre, count(*) as number_of_songs
From Song, Genre
Where Genre_ID = ID
Group by genre
Order by count(*) desc
Limit 3;
```

2) **Find names of artists who have songs that are in albums as well as outside of albums (singles). The result must have one column, named artist_name**

```
(Select distinct(Made_by_artist) as artist_name From Album)
Where artist_name in
(Select distinct(Artist_Name) as artist_name From Song where Album_Name is null);
```

3) **What were the top 10 most highly rated albums (highest average user rating) in the period 1990-1999?. Break ties using alphabetical order of album names. The result must have two columns, named album_name and average_user_rating.**

```
Select Album_Name as album_name, avg(Rating) as average_user_rating
From rating
Where album_name IS NOT NULL
And Date_of_Rating like '199%'
Group by album_name
Order by average_user_rating desc
Limit 10;
```

4) **Which were the top 3 most rated genres (this is the number of ratings of songs in genres, not the actual rating scores) in the years 1991-1995? The result must have two columns, named genre_name and number_of_song_ratings.**

```
Select Type as genre_name, count(*) as number_of_song_ratings
From rating, song, genre
WHERE Song_Title = Song_name and Genre_ID = ID
Date_of_Rating BETWEEN '1991-00-00' AND '1996-00-00'
Group by Type
Order by number_of_song_ratings desc
Limit 3;
```

5) **Which users have a playlist that has an average song rating of 4.0 or more? (This is the average of the average song rating for each song in the playlist.) A user may appear multiple times in the result if more than one of their playlists make the cut. The result must 3 columns named username, playlist_title, average_song_rating**

```
Select Username,  Playlist_title, avg(avgRating) as average_song_rating
From (select Song_Title, avg(rating) as avgRating from rating group by Song_Title
where Song_Title is not null) temp, rating r, Playlist p
Where temp.Song_Title = r.Song_Title
And r.Playlist_ID = p.Playlist_ID
and average_song_rating >= 4
Group by Playlist_ID;
```

6) **Who are the top 5 most engaged users in terms of number of ratings that they have given to songs or albums? (In other words, they have given the most number of ratings to songs or albums combined.) The result must have 2 columns, named username and number_of_ratings.**

```
Select username as Username, count(*) as number_of_ratings
From rating
```

Where Album_Name IS NOT NULL or Song_Title IS NOT NULL
Group by username
Order by number_of_ratings desc
Limit 5;

7) **Find the top 10 most prolific artists (most number of songs) in the years 1990-2010? Count each song in an album individually. The result must have 2 columns, named artist_name and number_of_songs.**

Select Artist_Name as artist_name, count(*) as number_of_songs
From Song
WHERE Release_date_of_song BETWEEN '1990-00-00' AND '2011-00-00'
Group by artist_name
Order by number_of_songs desc
Limit 10;

8) **Find the top 10 songs that are in most number of playlists. Break ties in alphabetical order of song titles. The result must have a 2 columns, named song_title and number_of_playlists.**

Select Song_Title as song_title, count(*) as number_of_playlists
From Playlist_Song
Group by Song_Title
Order by number_of_playlists desc
Limit 10;

9) **Find the top 20 most rated singles (songs that are not part of an album).**
Most rated meaning number of ratings, not actual rating scores.
The result must have 3 columns, named song_title, artist_name, number_of_ratings.

Select Song_Title, Artist_Name, count(*) as number_of_ratings
From rating, Song
Where Song_Title = Song_name
And Release_date_of_song IS NOT NULL

```
Group by Song_Title
Order by number_of_ratings desc
Limit 20;
```

10) Find all artists who discontinued making music after 1993. The result should be a single column named artist_title

```
Select Name as artist_title
From Artist
Where artist_title not in (select Name
                From Artist, Album, Song
                Where Name = Made_by_artist and Made_by_artist = Artist_Name
                And Release_date > '1993-00-00'
                Union
                select Name
                From Artist, Album, Song
                Where Name = Made_by_artist and Made_by_artist = Artist_Name
                And Release_date_of_song > '1993-00-00'
               );
```