1. create table Artist (
    id int auto_increment primary key,
    name varchar(100) not null,
    unique (name));

2. create table Song(
    id int not null auto_increment primary key,
    title varchar(255) not null,
    artist_id int not null,
    foreign key (artist_id) references Artist(id),
    album_id int,
    foreign key (album_id) references Album(id),
    genre varchar(80) not null,
    release_date date not null,
    playlist_id int,
    foreign key (playlist_id) references Playlist(id),
    unique (title, artist_id));

3. create table Album (
    id int not null auto_increment primary key,
    artist_id int not null,
    foreign key (artist_id) references Artist(id),
    release_date not null,
    name varchar(255) not null
    unique (name, artist_id));

4. create table User (
    id int auto_increment primary key,
    username varchar(255) not null,
    unique (username));

5. create table Playlist (
    id int auto_increment primary key,
    title varchar(255) not null,
    user_id int not null,
    foreign key (user_id) references User(id),
    song_id int not null,
    foreign key (song_id) references Song(id),
    date_from datetime not null,
    unique (user_id, title));

6. create table Rating (
    user_id int not null,
    rate tinyint not null check (rating>= 0 and rating <= 5),
    album_id int not null,
    foreign key (album_id) references Album(id),
    song_id int not null,
    foreign key (song_id) references Song(id),
    playlist_id int,
    foreign key (playlist_id) references Playlist(id),
    date_made date not null);

---

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 genre, count(*) as 'number_of_songs'
   from Song S
   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 A.name as 'artist_name'
   from artist A, song S, album L
   where A.id = S.artist_id and A.id = L.artist_id
   and S.album_id = L.id
   having A.id in (select a.id from artist a, song s where a.id = s.artist_id and s.album_id 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 L.name as album_name, round(avg(R.rate),1) as average_user_rating,
   from album L, rating R
   where R.album_id = L.id and year(R.date_made) between 1990 and 1999
   group by R.album_id
   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 S.genre as 'genre_name',  count(*) as 'number_of_song_ratings'
  from song S, rating R
  where R.song_id = S.id and year(R.date_made) between 1991 and 1995
  group by S.genre
  order  by count(*) 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 column named **username**, **playlist_title**, **average_song_rating.**

```
 : select U.username, P.title as playlist_title , round(avg(R.rate),1) as
          average_song_rating
    from user U, playlist P, rating R
    where U.id = P.user_id and U.id = R.user_id and P.id = R.playlist_id
    group by U.username
    having average_song_rating >= 4.0;
```

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 results need 2 columns, named **username** and **number_of_ratings**

```
: select U.username, count(*) as number_of_ratings
  from user U, rating R
  where U.id = R.user_id
  group by username
  order by count(*) 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 A.name, count(*) as number_of_songs
  from artist A, song S, Album L
  where A.id = S.artist_id and S.album_id = L.id and year(S.release_date) between 1990
          and 2010
  group by A.name
  order by count(*) 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 2 columns, named song_title and number_of_playlists.

```
: select S.title, count (*) as number_of_playlists
  from song S, playlist P
  where S.playlist_id = P.id and S.id = P.song_id
  group by S.title
  order by count(*) 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.

```
: select S.title as song_title, A.name as artist_name, count(*) as number_of_ratings
  from song S, Artist A, rating R
  where A.id = S.artist_id, S.id = R.song_id and S.album_id is null
  group by S.title, A.name
  order by count(*) desc
  limit 20;
```

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

```
: select distinct(A.name) as artist_title
  from artist A, Song S, Album L
  where (A.id = L.artist_id, and L.name not in
                                (select L.name from Artist A, Album L
                                where A.id = L.artist_id and year(release_date) > 1993) )
  having A.id not in (
                select A.id from Artist A, Song S
                where A.id = S.artist_id and S.album_id is null and S.title not in ( select
                        title from Artist a, Song s where a.id  = s.artist_id and
                        year (s.release_date) > 1993))
                );
```