Table for artist

```
CREATE TABLE artist (
name varchar(50) NOT NULL,
PRIMARY KEY (name)
);
```

Table for album

```
CREATE TABLE album (
name varchar(100) NOT NULL,
artist_name varchar(50) NOT NULL,
release_date date NOT NULL,
PRIMARY KEY (name),
KEY artist_name_idx (artist_name),
CONSTRAINT artist_name FOREIGN KEY (artist_name) REFERENCES artist (name)
);
```

Table for song

```
CREATE TABLE song (
song_name varchar(100) NOT NULL,
artist varchar(50) NOT NULL,
release_date date DEFAULT NULL,
album_name varchar(100) DEFAULT NULL,
PRIMARY KEY (song_name,artist),
KEY artist_idx (artist),
CONSTRAINT artist FOREIGN KEY (artist) REFERENCES artist (name)
);
```

Table for genre

```
CREATE TABLE genre (
genre varchar(20) NOT NULL,
PRIMARY KEY (genre)
);
```

Table for song and genre

```
CREATE TABLE song_genre (
song_name varchar(50) DEFAULT NULL,
genre varchar(20) DEFAULT NULL,
KEY song_name_idx (song_name),
KEY genre_idx (genre),
CONSTRAINT genre FOREIGN KEY (genre) REFERENCES genre (genre),
CONSTRAINT song_name FOREIGN KEY (song_name) REFERENCES song (song_name)
);
```

Table for user

```
CREATE TABLE user (
user_name varchar(100) NOT NULL,
PRIMARY KEY (user_name)
);
```

Table for playlist

```
CREATE TABLE play_list (
list_name varchar(100) NOT NULL,
song_name varchar(100) NOT NULL,
user_name varchar(50) NOT NULL,
create_date date NOT NULL,
PRIMARY KEY (list_name,user_name),
KEY user_name_idx (user_name),
CONSTRAINT create_user FOREIGN KEY (user_name) REFERENCES user (user_name)
);
```

Table for rating

```
CREATE TABLE rating (
user_name varchar(50) NOT NULL,
name varchar(100) NOT NULL,
rating int NOT NULL,
rating_date date NOT NULL,
KEY user_name_idx (user_name),
CONSTRAINT user_name FOREIGN KEY (user_name) REFERENCES user (user_name)
);
```

Query

1

```
select genre, count(*) as    number_of_songs
from song_genre
group by genre
order by number_of_songs desc
limit 3;
```

2

```
select distinct artist_name
from album
where artist_name in (
select distinct artist
from song
where album_name = null);
```

3

```
select name as album_name, avg(rating) as average_user_rating
from rating
where name in (select name
from album
) and rating_date between "1990-01-01" and "1999-12-31"
group by name
order by average_user_rating desc, name asc
limit 10;
```

4

```
select count(song_genre.song_name) as number_of_song_ratings , song_genre.genre as genre_name
from song_genre
right join rating on song_genre.song_name = rating.name
where rating is not null and rating.rating_date between "1991-01-01" and "1995-12-31"
group by genre_name
limit 3;
```

5

```
select play_list.user_name as username, play_list.list_name as playlist_title, avg(rating.rating) as average_song_rating
from play_list
right join rating on play_list.song_name = rating.name
group by list_name
having average_song_rating > 4;
```

6

```
SELECT user_name as username, count(*) as number_of_ratings
from rating
where name in (select song_name from song) or (select name from album)
group by username
order by number_of_ratings desc
limit 5;
```

7

```
select artist as artist_name, count(*) as number_of_songs
from song
where release_date between "1990-01-01" and "2010-12-31"
group by artist
order by number_of_songs desc
limit 10;
```

8

```
select song_name as song_title, count(song_name) as number_of_playlists
from play_list
group by song_name
order by number_of_playlists desc, song_title asc
limit 10;
```

9

```
select song_name as song_title , genre as Genre_name, count as number_of_ratings
from
(select *, row_number() over(partition by genre order by count desc, song_name asc) as rnk
from
(select song_name, genre, count(rating) as count
from song_genre
join rating on song_name = rating.name
group by genre, song_name) a) s
where rnk between 1 and 20;
```

10

```
select distinct artist
from song
where artist not in (select artist from song where release_date > "1993-12-31");
```