SCHEMA:

```
CREATE DATABASE IF NOT EXISTS `hw4`;
USE `hw4`;

DROP TABLE IF EXISTS `playlist_rating`;
DROP TABLE IF EXISTS `song_rating`;
DROP TABLE IF EXISTS `album_rating`;
DROP TABLE IF EXISTS `playlist_songs`;
DROP TABLE IF EXISTS `playlist`;
DROP TABLE IF EXISTS `genre`;
DROP TABLE IF EXISTS `song`;
DROP TABLE IF EXISTS `album`;
DROP TABLE IF EXISTS `artist`;
DROP TABLE IF EXISTS `user`;

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

CREATE TABLE `album` (
  `album_id` integer NOT NULL,
  `album_name` varchar(50) NOT NULL,
  `released_by` varchar(30) NOT NULL,
  `release_date` date NOT NULL,
  PRIMARY KEY (`album_id`),
  CONSTRAINT `uc_album` UNIQUE (`album_name`, `released_by`),
  FOREIGN KEY (`released_by`) references `artist` (`artist_name`) ON DELETE CASCADE
);

CREATE TABLE `song` (
  `song_id` integer NOT NULL,
  `song_title` varchar(50) NOT NULL,
  `recorded_by` varchar(50) NOT NULL,
  `release_date` date NOT NULL,
  `album_id` integer,
  PRIMARY KEY (`song_id`),
  CONSTRAINT `uc_song` UNIQUE (`song_title`, `recorded_by`),
  FOREIGN KEY (`album_id`) references `album` (`album_id`) ON DELETE CASCADE
```

```
);

CREATE TABLE `genre` (
  `song_id` integer NOT NULL,
  `genre_name` varchar(20) NOT NULL,
  PRIMARY KEY (`song_id`,`genre_name`),
  FOREIGN KEY (`song_id`) references `song` (`song_id`) ON DELETE CASCADE
);

CREATE TABLE `user` (
  `username` varchar(50) NOT NULL,
  PRIMARY KEY (`username`)
);

CREATE TABLE `playlist` (
  `playlist_id` integer NOT NULL,
  `playlist_title` varchar(50) NOT NULL,
  `created_at` datetime NOT NULL,
  `username` varchar(50) NOT NULL,
  PRIMARY KEY (`playlist_id`),
  CONSTRAINT `uc_playlist` UNIQUE (`playlist_title`, `username`),
  FOREIGN KEY (`username`) references `user` (`username`) ON DELETE CASCADE
);

CREATE TABLE `playlist_songs` (
  `playlist_id` integer NOT NULL,
  `song_id` integer NOT NULL,
  PRIMARY KEY (`playlist_id`,`song_id`),
  FOREIGN KEY (`playlist_id`) references `playlist` (`playlist_id`) ON DELETE CASCADE,
  FOREIGN KEY (`song_id`) references `song` (`song_id`) ON DELETE CASCADE
);

CREATE TABLE `song_rating` (
  `song_id` integer NOT NULL,
  `username` varchar(50) NOT NULL,
  `rating` integer NOT NULL,
  `created_at` date NOT NULL,
  PRIMARY KEY (`username`,`song_id`),
  FOREIGN KEY (`username`) references `user` (`username`) ON DELETE CASCADE,
  FOREIGN KEY (`song_id`) references `song` (`song_id`) ON DELETE CASCADE
```

```
);


CREATE TABLE `playlist_rating` (
  `playlist_id` integer NOT NULL,
  `username` varchar(50) NOT NULL,
  `rating` integer NOT NULL,
  `created_at` date NOT NULL,
  PRIMARY KEY (`username`,`playlist_id`),
  FOREIGN KEY (`username`) references `user` (`username`) ON DELETE CASCADE,
  FOREIGN KEY (`playlist_id`) references `playlist` (`playlist_id`) ON DELETE CASCADE
);

CREATE TABLE `album_rating` (
  `album_id` integer NOT NULL,
  `username` varchar(50) NOT NULL,
  `rating` integer NOT NULL,
  `created_at` date NOT NULL,
  PRIMARY KEY (`username`,`album_id`),
  FOREIGN KEY (`username`) references `user` (`username`) ON DELETE CASCADE,
  FOREIGN KEY (`album_id`) references `album` (`album_id`) ON DELETE CASCADE
);
```

QUERIES:

1.

```
SELECT genre, COUNT(*) AS "number_of_songs" FROM genre GROUP BY genre ORDER
BY number_of_songs DESC LIMIT 3;
```

2.

```
SELECT artist_name FROM artist WHERE artist_name IN
(SELECT DISTINCT a.artist_name FROM artist a JOIN song s ON a.artist_name =
s.recorded_by JOIN album l ON l.released_by = a.artist_name
WHERE s.song_id IN (SELECT song_id FROM song WHERE album_id IS NULL))
AND artist_name IN
(SELECT DISTINCT a.artist_name FROM artist a JOIN song s ON a.artist_name =
s.recorded_by JOIN album l ON l.released_by = a.artist_name
WHERE s.song_id IN (SELECT song_id FROM song WHERE album_id IS NOT NULL));
```

3.
```
SELECT a.album_name, AVG(r.rating) AS average_user_rating FROM album a JOIN
album_rating r ON a.album_id = r.album_id
WHERE year(r.created_at) between 1990 AND 1999 GROUP BY a.album_name
ORDER BY average_user_rating DESC, album_name ASC LIMIT 10;
```

4.
```
SELECT genre_name, COUNT(*) AS number_of_song_ratings FROM genre g JOIN
song_rating s ON s.song_id = g.song_id
WHERE year(s.created_at) between 1991 AND 1995 GROUP BY genre_name
ORDER BY number_of_song_ratings DESC LIMIT 3;
```

5.
```
SELECT p.username, p.playlist_title, AVG(avg_song_rating) AS average_song_rating FROM
playlist p JOIN playlist_songs ps ON ps.playlist_id = p.playlist_id JOIN
(SELECT song_id, AVG(rating) AS avg_song_rating FROM song_rating GROUP BY song_id)
s ON s.song_id = ps.song_id
GROUP BY p.playlist_title, p.username HAVING average_song_rating>=4;
```

6.
```
SELECT COALESCE(u_name, username) AS username, COALESCE(num_album_ratings,0) +
COALESCE(num_song_ratings,0) AS number_of_ratings FROM (
SELECT * FROM
(SELECT username AS u_name, COUNT(*) AS num_album_ratings FROM album_rating a
GROUP BY a.username) ar
LEFT OUTER JOIN
(SELECT username, COUNT(*) AS num_song_ratings FROM song_rating s GROUP BY
s.username) sr ON sr.username = ar.u_name
UNION
SELECT * FROM
(SELECT username, COUNT(*) AS num_album_ratings FROM album_rating a GROUP BY
a.username) ar
RIGHT OUTER JOIN
(SELECT username, COUNT(*) AS num_song_ratings FROM song_rating s GROUP BY
s.username) sr ON sr.username = ar.username) final ORDER BY number_of_ratings DESC
LIMIT 5;
```

7.
```
SELECT recorded_by AS artist_name, COUNT(*) AS number_of_songs FROM song WHERE year(release_date) between 1990 AND 2010 GROUP BY recorded_by ORDER BY number_of_songs DESC LIMIT 10;
```

8.
```
SELECT s.song_title, COUNT(*) AS number_of_playlists FROM playlist_songs ps JOIN song s ON ps.song_id = s.song_id GROUP BY s.song_title ORDER BY number_of_playlists DESC, song_title ASC LIMIT 10;
```

9.
```
SELECT song_title, recorded_by AS artist_name, COUNT(*) AS number_of_ratings FROM song_rating sr JOIN song s ON sr.song_id = s.song_id
WHERE s.album_id IS NULL GROUP BY song_title ORDER BY number_of_ratings DESC LIMIT 20;
```

```
-- 10
SELECT recorded_by FROM song WHERE recorded_by NOT IN
(SELECT recorded_by FROM song WHERE year(release_date) > 1993);
```