

Total 10 descriptive queries: min 2 should be subqueries (#4, 5), 2 should be queries on joins (#1, 4, 5), and 2 should be aggregate queries (#1, 3, 4, 5)

1. Average song duration for each individual album

```
SELECT A.album_name, AVG(S.duration) AS avg_song_duration
FROM album A
INNER JOIN album_has A_H ON A.album_id = A_H.album_id
INNER JOIN song S ON A_H.song_id = S.song_id
GROUP BY A.album_name;
```

2. Album with the most recent release date

```
SELECT album_name, release_date FROM album
ORDER BY release_date DESC LIMIT 1;
```

3. Genre with the greatest number of songs

```
SELECT G.genre_name, COUNT(B.song_id) AS num_songs
FROM genre G
INNER JOIN song_belongs_to B ON G.genre_id = B.genre_id
GROUP BY G.genre_name
ORDER BY num_songs DESC LIMIT 1;
```

4. Most popular song for each artist (in the most playlists)

```
SELECT user_id, song_id, MAX(song_count)
FROM (
    SELECT artist_performs.user_id, playlist_contains.song_id,
    COUNT(playlist_id) AS song_count
    FROM playlist_contains PC
    LEFT JOIN artist_performs AP ON artist_performs.song_id =
    playlist_contains.song_id
    GROUP BY song_id
)
GROUP BY user_id;
```

5. Most popular artist (in the most playlists)

```
SELECT user_id, MAX(total_count)
FROM (
    SELECT artist_performs.user_id, COUNT(playlist_id) AS total_count
    FROM playlist_contains
    LEFT JOIN artist_performs ON artist_performs.song_id =
    playlist_contains.song_id
    GROUP BY user_id
);
```

6. Artist with the most songs

```
SELECT artist_name, MAX(total_count)
FROM artist LEFT JOIN (
    SELECT user_id, COUNT(song_id) AS total_count
    FROM artist_performs
) AS temp ON temp.user_id = artist.user_id;
```

```

SELECT artist.artist_name, temp.total_songs
FROM artist INNER JOIN (
    -- Subquery gets artist w/ the most songs
    SELECT user_id, COUNT(song_id) AS total_songs
    FROM artist_performs
    GROUP BY user_id
    ORDER BY total_songs DESC
    LIMIT 1
) AS temp ON artist.user_id = temp.user_id;

```

7. Song with the most recent release date

```

SELECT title, release_date FROM song ORDER BY release_date DESC LIMIT 1;

```

8. Most common plan among users

```

SELECT P.plan_name, COUNT(user_id) AS num_users
FROM [user] U
JOIN [plan] P ON U.plan_id = P.plan_id
GROUP BY P.plan_name
ORDER BY num_usersuser_count DESC
LIMIT 1;

```

9. Most popular song overall (in the most playlists)

```

SELECT title, MAX(count)
FROM (
    SELECT song_id, COUNT(playlist_id)
    FROM playlist_contains
    GROUP BY song_id
) AS temp
LEFT JOIN song ON song.song_id = temp.song_id;

SELECT song.title, temp.num_of_associated_playlists
FROM song INNER JOIN (
    SELECT song_id, COUNT(playlist_id) as num_of_associated_playlists
    FROM playlist_contains
    GROUP BY song_id
    ORDER BY num_of_associated_playlists DESC
    LIMIT 1
) AS temp ON song.song_id = temp.song_id;

```

10. User with the most playlists

```

SELECT username, MAX(playlist_count)
FROM user
LEFT JOIN (
    SELECT user_id, COUNT(playlist_id) AS playlist_count
    FROM playlist
    GROUP BY user_id
) ON user.user_id = playlist.user_id;

SELECT user.username, temp.total_playlists
FROM user INNER JOIN (
    -- Subquery gets user w/ the most playlists

```

```
SELECT user_id, COUNT(playlist_id) AS total_playlists
FROM playlist
GROUP BY user_id
ORDER BY total_playlists DESC
LIMIT 1
) AS temp ON user.user_id = temp.user_id;
```