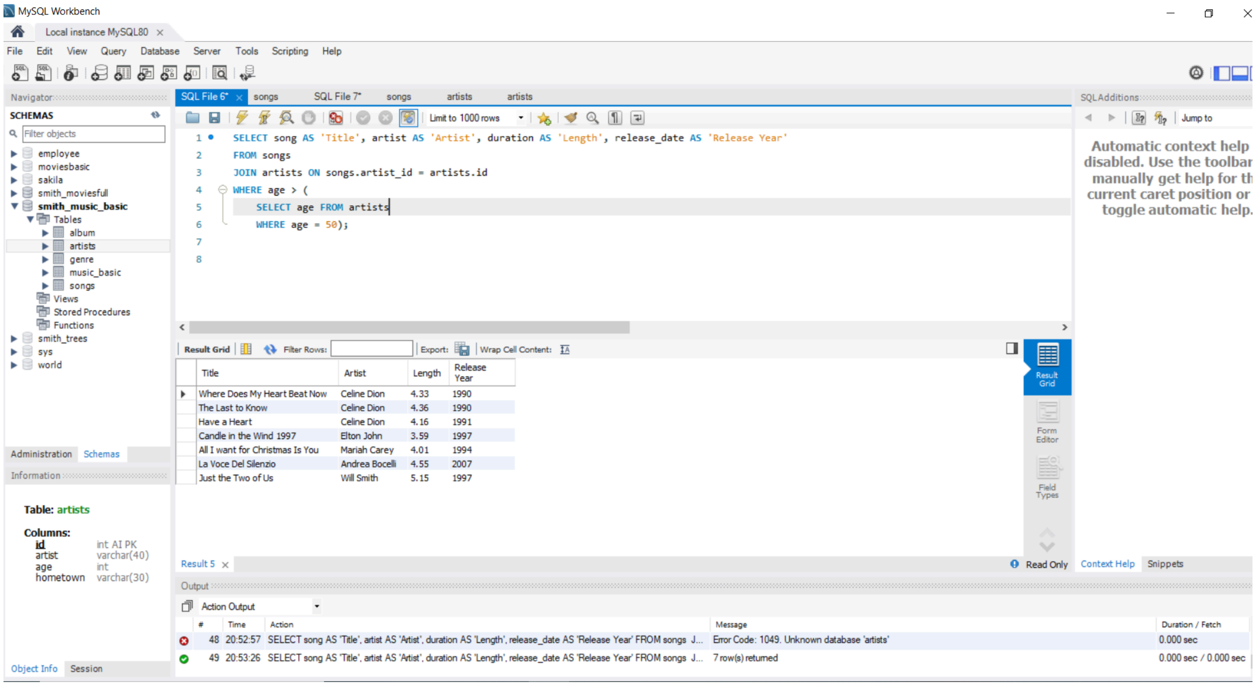
1) Write a query that JOINS two tables and uses a subquery to list the songs made by artists over the age of 50.



```
SELECT song AS 'Title', artist AS 'Artist', duration AS 'Length', release_date AS 'Release Year'
FROM songs
JOIN artists ON songs.artist_id = artists.id
WHERE age > (
    SELECT age FROM artists
    WHERE age = 50);
```

Title	Artist	Length	Release Year
Where Does My Heart Beat Now	Celine Dion	4.33	1990
The Last to Know	Celine Dion	4.36	1990
Have a Heart	Celine Dion	4.16	1991
Candle in the Wind 1997	Elton John	3.59	1997
All I want for Christmas Is You	Mariah Carey	4.01	1994
La Voce Del Silenzio	Andrea Bocelli	4.55	2007
Just the Two of Us	Will Smith	5.15	1997

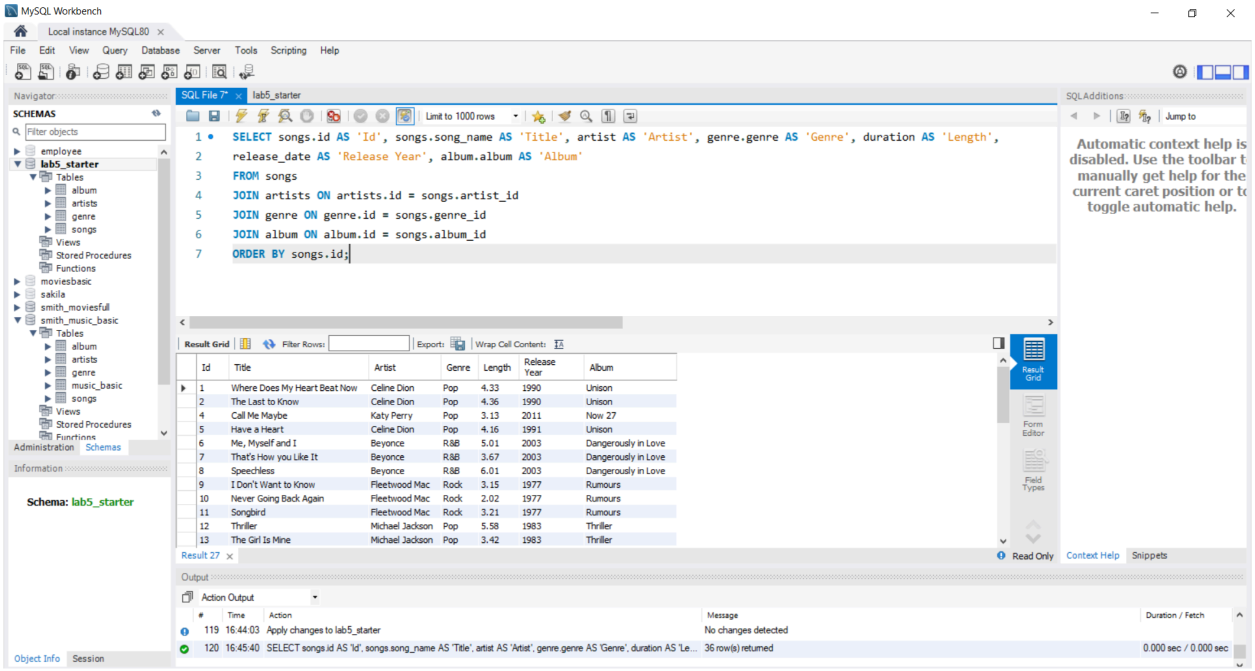
Table: artists

Columns:

id	int AI PK
artist	varchar(40)
age	int
hometown	varchar(30)

```
#   Time      Action                                                                                       Message                                      Duration / Fetch
48  20:52:57  SELECT song AS 'Title', artist AS 'Artist', duration AS 'Length', release_date AS 'Release Year' FROM songs J...  Error Code: 1049. Unknown database 'artists'  0.000 sec
49  20:53:26  SELECT song AS 'Title', artist AS 'Artist', duration AS 'Length', release_date AS 'Release Year' FROM songs J...  7 row(s) returned                            0.000 sec / 0.000 sec
```

2) Write a query that JOINS all tables created in the previous lab to display the data in a similar format to music_basic.



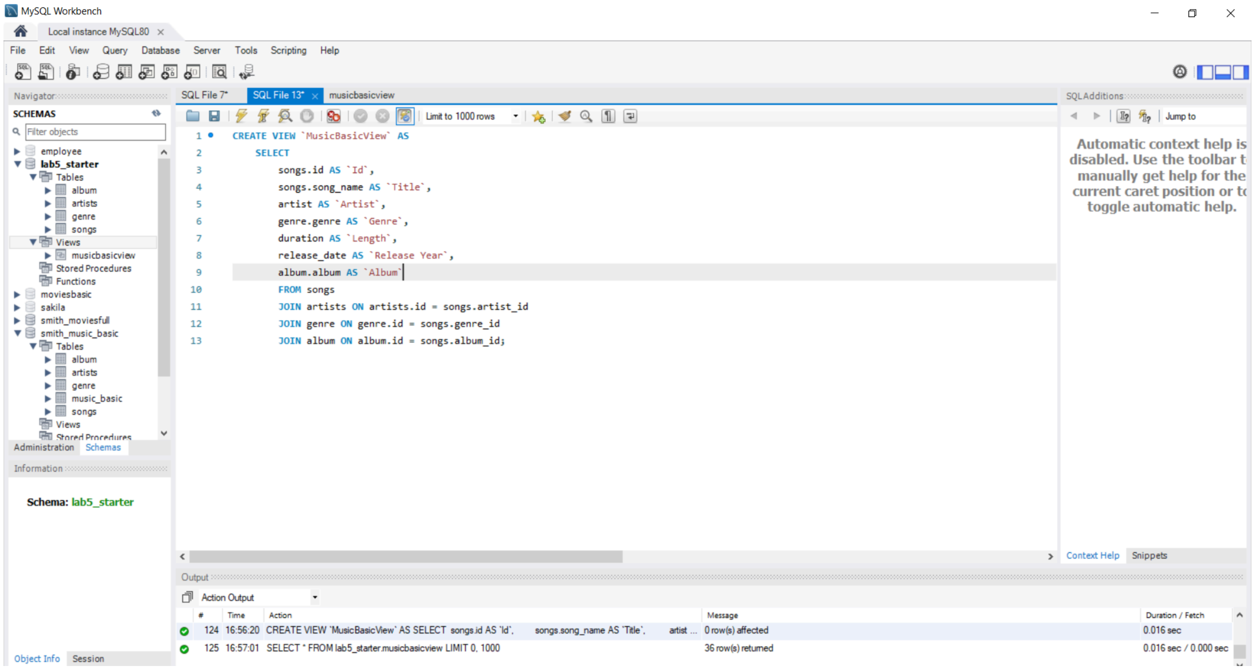
Schema: lab5_starter

```
SELECT songs.id AS 'Id', songs.song_name AS 'Title', artist AS 'Artist', genre.genre AS 'Genre', duration AS 'Length',
release_date AS 'Release Year', album.album AS 'Album'
FROM songs
JOIN artists ON artists.id = songs.artist_id
JOIN genre ON genre.id = songs.genre_id
JOIN album ON album.id = songs.album_id
ORDER BY songs.id;
```

Id	Title	Artist	Genre	Length	Release Year	Album
1	Where Does My Heart Beat Now	Celine Dion	Pop	4.33	1990	Unison
2	The Last to Know	Celine Dion	Pop	4.36	1990	Unison
4	Call Me Maybe	Katy Perry	Pop	3.13	2011	Now 27
5	Have a Heart	Celine Dion	Pop	4.16	1991	Unison
6	Me, Myself and I	Beyonce	R&B	5.01	2003	Dangerously in Love
7	That's How you Like It	Beyonce	R&B	3.67	2003	Dangerously in Love
8	Speechless	Beyonce	R&B	6.01	2003	Dangerously in Love
9	I Don't Want to Know	Fleetwood Mac	Rock	3.15	1977	Rumours
10	Never Going Back Again	Fleetwood Mac	Rock	2.02	1977	Rumours
11	Songbird	Fleetwood Mac	Rock	3.21	1977	Rumours
12	Thriller	Michael Jackson	Pop	5.58	1983	Thriller
13	The Girl Is Mine	Michael Jackson	Pop	3.42	1983	Thriller

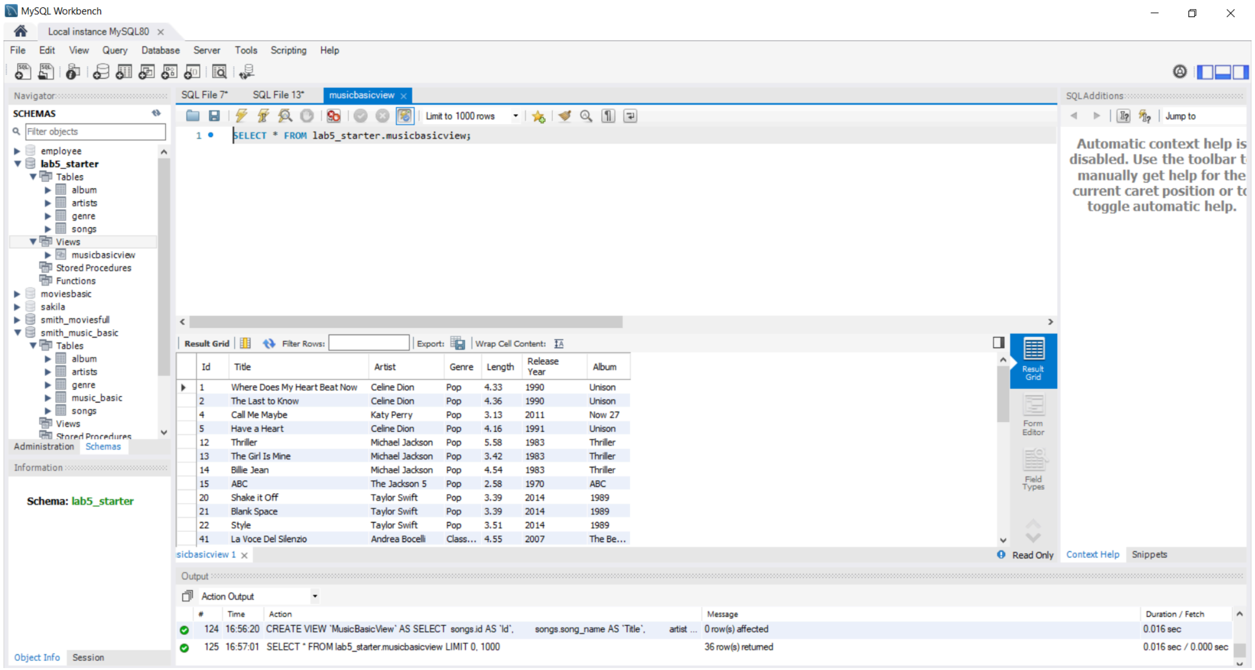
```
#    Time      Action                                                                                          Message               Duration / Fetch
119  16:44:03  Apply changes to lab5_starter                                                                   No changes detected
120  16:45:40  SELECT songs.id AS 'Id', songs.song_name AS 'Title', artist AS 'Artist', genre.genre AS 'Genre', duration AS 'Le...  36 row(s) returned    0.000 sec / 0.000 sec
```

3) Create a View table using the query created in Question 2. Display the contents of the view table and include the results in your screenshot.



```
CREATE VIEW `MusicBasicView` AS
    SELECT
        songs.id AS `Id`,
        songs.song_name AS `Title`,
        artist AS `Artist`,
        genre.genre AS `Genre`,
        duration AS `Length`,
        release_date AS `Release Year`,
        album.album AS `Album`
        FROM songs
        JOIN artists ON artists.id = songs.artist_id
        JOIN genre ON genre.id = songs.genre_id
        JOIN album ON album.id = songs.album_id;
```

```
#    Time      Action                                                                          Message              Duration / Fetch
124  16:56:20  CREATE VIEW `MusicBasicView` AS SELECT  songs.id AS `Id`,   songs.song_name AS `Title`,   artist ...  0 row(s) affected    0.016 sec
125  16:57:01  SELECT * FROM lab5_starter.musicbasicview LIMIT 0, 1000                                         36 row(s) returned   0.016 sec / 0.000 sec
```

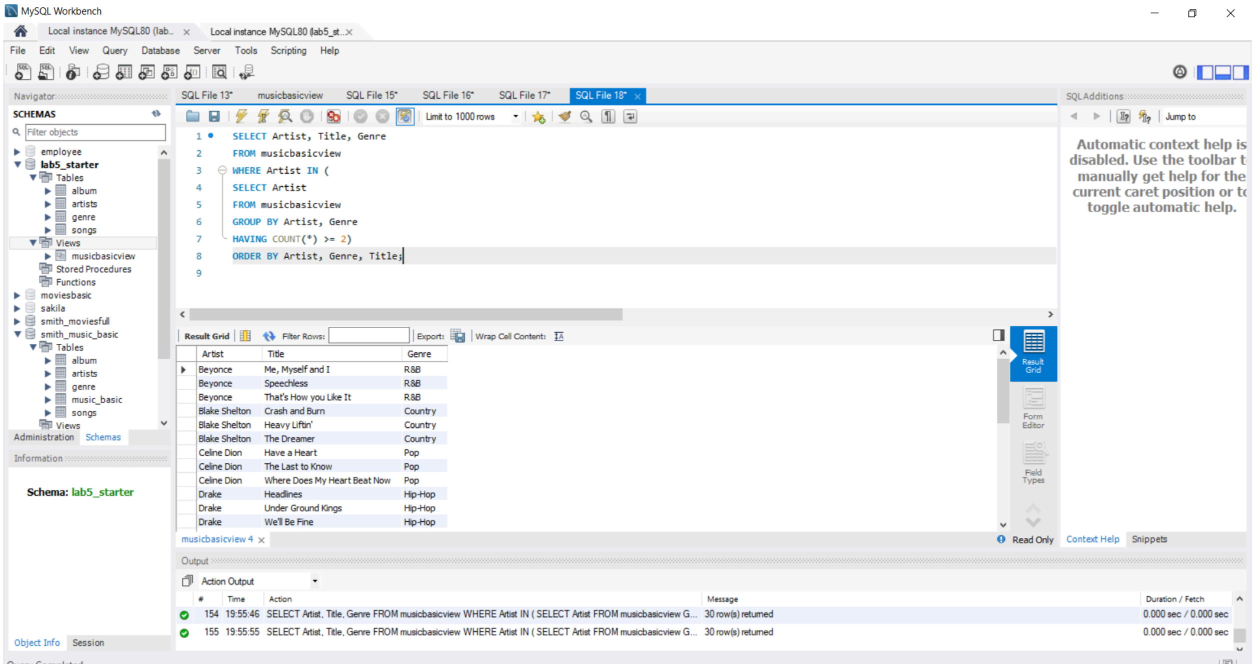


```
SELECT * FROM lab5_starter.musicbasicview;
```

Id	Title	Artist	Genre	Length	Release Year	Album
1	Where Does My Heart Beat Now	Celine Dion	Pop	4.33	1990	Unison
2	The Last to Know	Celine Dion	Pop	4.36	1990	Unison
4	Call Me Maybe	Katy Perry	Pop	3.13	2011	Now 27
5	Have a Heart	Celine Dion	Pop	4.16	1991	Unison
12	Thriller	Michael Jackson	Pop	5.58	1983	Thriller
13	The Girl Is Mine	Michael Jackson	Pop	3.42	1983	Thriller
14	Billie Jean	Michael Jackson	Pop	4.54	1983	Thriller
15	ABC	The Jackson 5	Pop	2.58	1970	ABC
20	Shake it Off	Taylor Swift	Pop	3.39	2014	1989
21	Blank Space	Taylor Swift	Pop	3.39	2014	1989
22	Style	Taylor Swift	Pop	3.51	2014	1989
41	La Voce Del Silenzio	Andrea Bocelli	Class...	4.55	2007	The Be...

```
#    Time      Action                                                                          Message              Duration / Fetch
124  16:56:20  CREATE VIEW `MusicBasicView` AS SELECT  songs.id AS `Id`,   songs.song_name AS `Title`,   artist ...  0 row(s) affected    0.016 sec
125  16:57:01  SELECT * FROM lab5_starter.musicbasicview LIMIT 0, 1000                                         36 row(s) returned   0.016 sec / 0.000 sec
```

4) Using a complex query and the newly created MusicBasicView, find the artists HAVING at least two songs in a genre



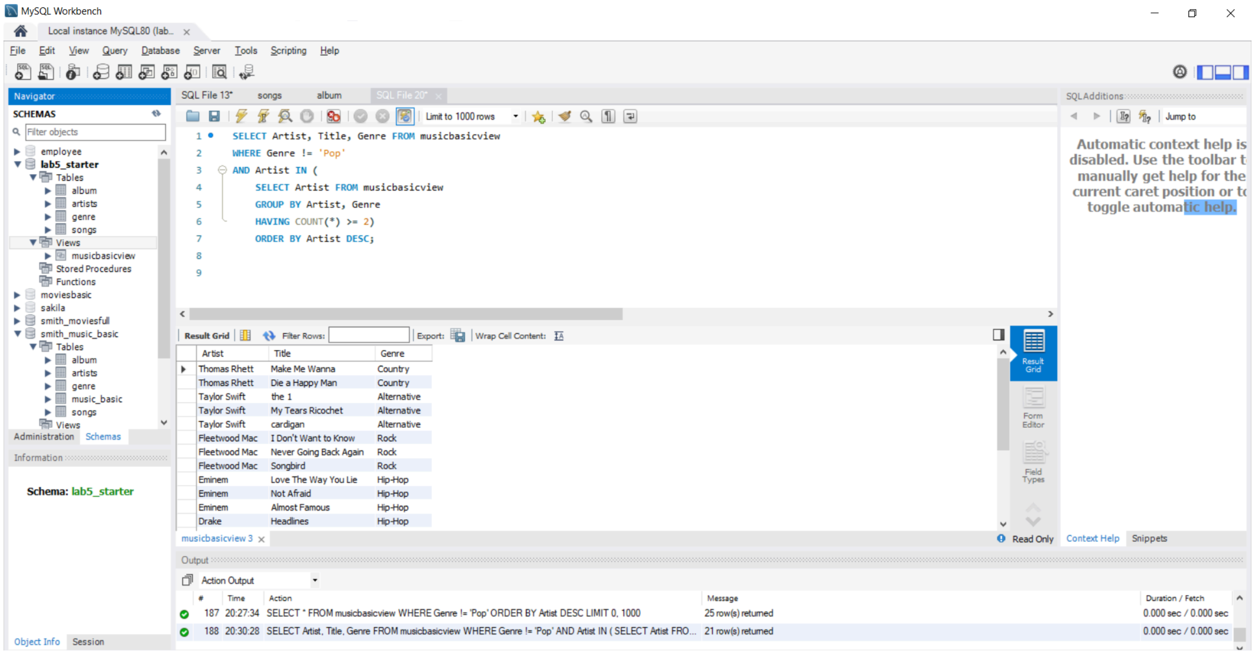
```
SELECT Artist, Title, Genre
FROM musicbasicview
WHERE Artist IN (
SELECT Artist
FROM musicbasicview
GROUP BY Artist, Genre
HAVING COUNT(*) >= 2)
ORDER BY Artist, Genre, Title;
```

Artist	Title	Genre
Beyonce	Me, Myself and I	R&B
Beyonce	Speechless	R&B
Beyonce	That's How you Like It	R&B
Blake Shelton	Crash and Burn	Country
Blake Shelton	Heavy Liftin'	Country
Blake Shelton	The Dreamer	Country
Celine Dion	Have a Heart	Pop
Celine Dion	The Last to Know	Pop
Celine Dion	Where Does My Heart Beat Now	Pop
Drake	Headlines	Hip-Hop
Drake	Under Ground Kings	Hip-Hop
Drake	We'll Be Fine	Hip-Hop

```
#    Time      Action                                                                              Message              Duration / Fetch
154  19:55:46  SELECT Artist, Title, Genre FROM musicbasicview WHERE Artist IN ( SELECT Artist FROM musicbasicview G...  30 row(s) returned  0.000 sec / 0.000 sec
155  19:55:55  SELECT Artist, Title, Genre FROM musicbasicview WHERE Artist IN ( SELECT Artist FROM musicbasicview G...  30 row(s) returned  0.000 sec / 0.000 sec
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

5) Modify your query from question 4 to only include records where the Genre is not Pop and sort the output by artist name in descending order



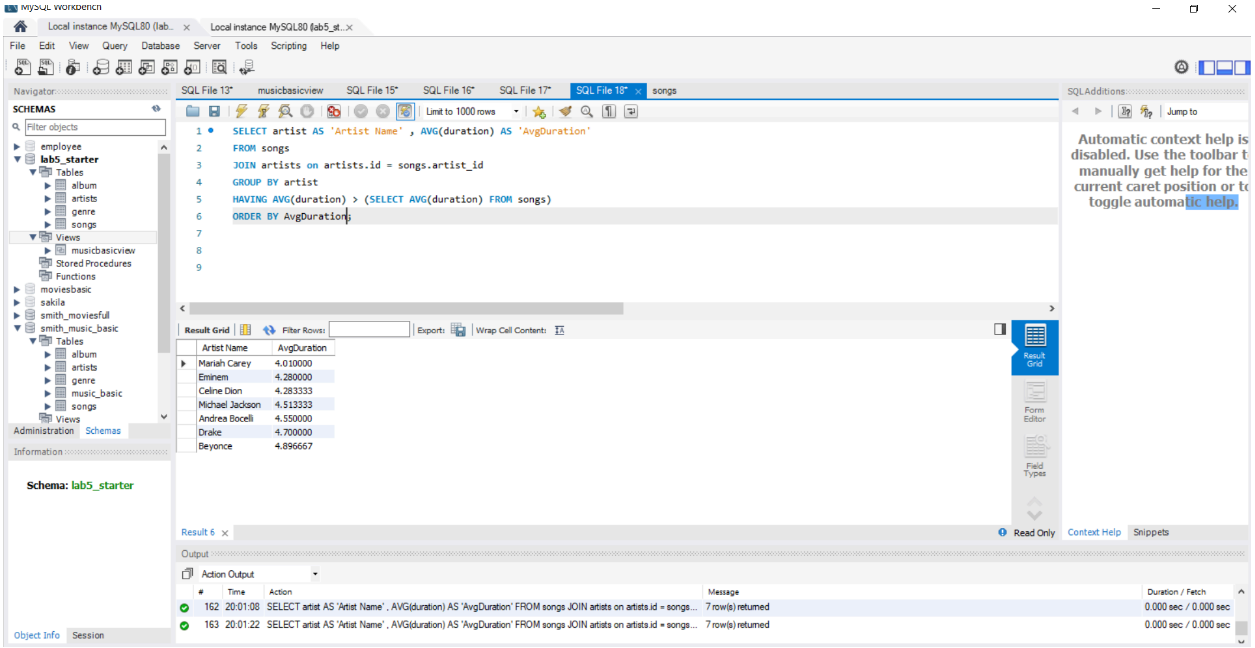
```
SELECT Artist, Title, Genre FROM musicbasicview
WHERE Genre != 'Pop'
AND Artist IN (
    SELECT Artist FROM musicbasicview
    GROUP BY Artist, Genre
    HAVING COUNT(*) >= 2)
    ORDER BY Artist DESC;
```

Artist	Title	Genre
Thomas Rhett	Make Me Wanna	Country
Thomas Rhett	Die a Happy Man	Country
Taylor Swift	the 1	Alternative
Taylor Swift	My Tears Ricochet	Alternative
Taylor Swift	cardigan	Alternative
Fleetwood Mac	I Don't Want to Know	Rock
Fleetwood Mac	Never Going Back Again	Rock
Fleetwood Mac	Songbird	Rock
Eminem	Love The Way You Lie	Hip-Hop
Eminem	Not Afraid	Hip-Hop
Eminem	Almost Famous	Hip-Hop
Drake	Headlines	Hip-Hop

```
#    Time      Action                                                                                   Message              Duration / Fetch
187  20:27:34  SELECT * FROM musicbasicview WHERE Genre != 'Pop' ORDER BY Artist DESC LIMIT 0, 1000          25 row(s) returned  0.000 sec / 0.000 sec
188  20:30:28  SELECT Artist, Title, Genre FROM musicbasicview WHERE Genre != 'Pop' AND Artist IN ( SELECT Artist FRO...  21 row(s) returned  0.000 sec / 0.000 sec
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

6) Write a query that finds the average duration per artist (i.e., The average duration of a Thomas Rhett song is 3.47). Only show the artists with an average duration greater than the overall average duration.



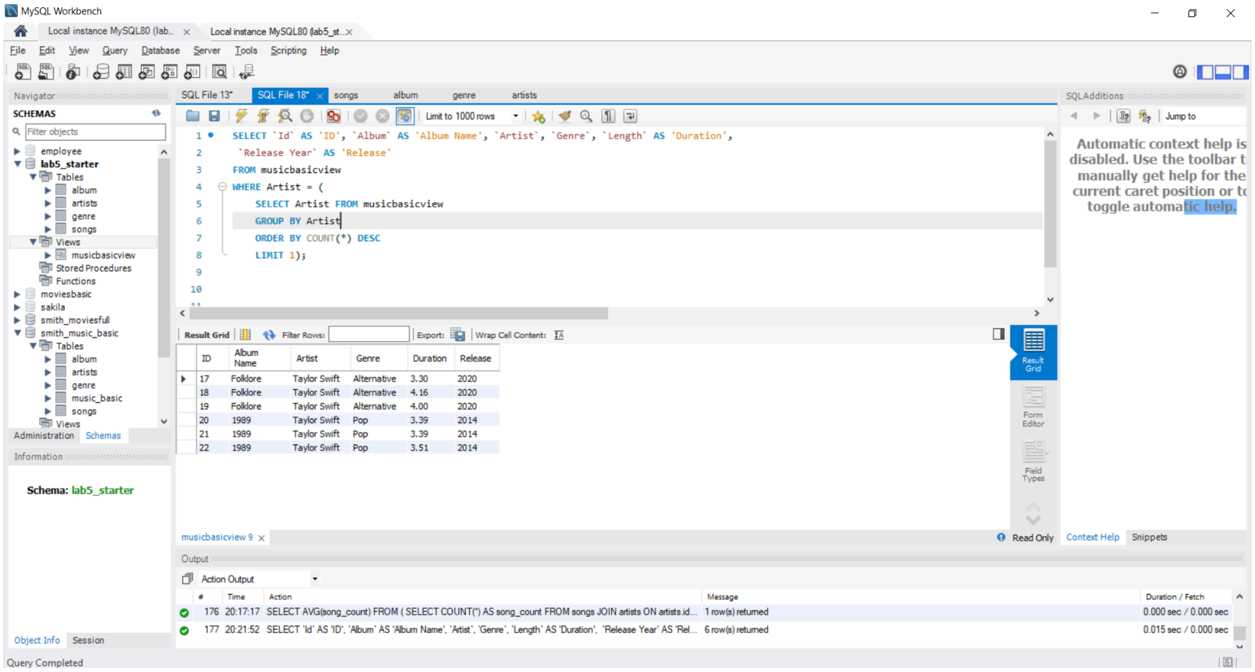
```
SELECT artist AS 'Artist Name' , AVG(duration) AS 'AvgDuration'
FROM songs
JOIN artists on artists.id = songs.artist_id
GROUP BY artist
HAVING AVG(duration) > (SELECT AVG(duration) FROM songs)
ORDER BY AvgDuration;
```

Artist Name	AvgDuration
Mariah Carey	4.010000
Eminem	4.280000
Celine Dion	4.283333
Michael Jackson	4.513333
Andrea Bocelli	4.550000
Drake	4.700000
Beyonce	4.896667

```
#    Time      Action                                                                                  Message            Duration / Fetch
162  20:01:08  SELECT artist AS 'Artist Name' , AVG(duration) AS 'AvgDuration' FROM songs JOIN artists on artists.id = songs...  7 row(s) returned  0.000 sec / 0.000 sec
163  20:01:22  SELECT artist AS 'Artist Name' , AVG(duration) AS 'AvgDuration' FROM songs JOIN artists on artists.id = songs...  7 row(s) returned  0.000 sec / 0.000 sec
```

MySQL Workbench
Local instance MySQL80 (lab...
Local instance MySQL80 (lab5_st...
File Edit View Query Database Server Tools Scripting Help
SQL File 13* musicbasicview SQL File 15* SQL File 16* SQL File 17* SQL File 18* songs
Limit to 1000 rows
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result 6
Read Only
Context Help
Snippets
Output
Action Output
Schema: lab5_starter
Object Info
Session

7) Write a query that shows all songs where the song's artist is the artist with the most songs was.

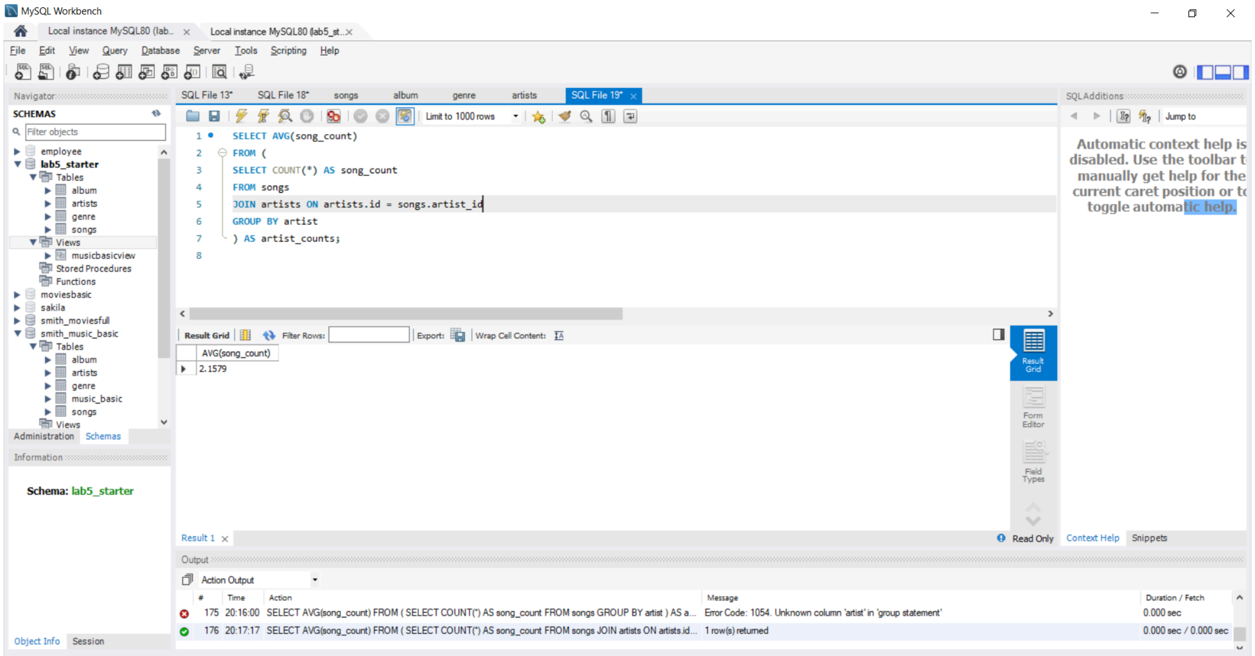


```
SELECT `Id` AS 'ID', `Album` AS 'Album Name', `Artist`, `Genre`, `Length` AS 'Duration',
 `Release Year` AS 'Release'
FROM musicbasicview
WHERE Artist = (
    SELECT Artist FROM musicbasicview
    GROUP BY Artist
    ORDER BY COUNT(*) DESC
    LIMIT 1);
```

ID	Album Name	Artist	Genre	Duration	Release
17	Folklore	Taylor Swift	Alternative	3.30	2020
18	Folklore	Taylor Swift	Alternative	4.16	2020
19	Folklore	Taylor Swift	Alternative	4.00	2020
20	1989	Taylor Swift	Pop	3.39	2014
21	1989	Taylor Swift	Pop	3.39	2014
22	1989	Taylor Swift	Pop	3.51	2014

```
#    Time      Action                                                                                        Message            Duration / Fetch
176  20:17:17  SELECT AVG(song_count) FROM ( SELECT COUNT(*) AS song_count FROM songs JOIN artists ON artists.id...  1 row(s) returned  0.000 sec / 0.000 sec
177  20:21:52  SELECT `Id` AS 'ID', `Album` AS 'Album Name', `Artist`, `Genre`, `Length` AS 'Duration',  `Release Year` AS 'Rel...  6 row(s) returned  0.015 sec / 0.000 sec
```

8) Write a query that returns the average number of songs an artist has in the songs table. (Hint: You need to find the average of each artist's COUNT)



Schema: lab5_starter

```
SELECT AVG(song_count)
FROM (
SELECT COUNT(*) AS song_count
FROM songs
JOIN artists ON artists.id = songs.artist_id
GROUP BY artist
) AS artist_counts;
```

AVG(song_count)
2.1579

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

```
#    Time      Action                                                                               Message                                                        Duration / Fetch
175  20:16:00  SELECT AVG(song_count) FROM ( SELECT COUNT(*) AS song_count FROM songs GROUP BY artist ) AS a...   Error Code: 1054. Unknown column 'artist' in 'group statement'   0.000 sec
176  20:17:17  SELECT AVG(song_count) FROM ( SELECT COUNT(*) AS song_count FROM songs JOIN artists ON artists.id...   1 row(s) returned                                               0.000 sec / 0.000 sec
```