

## ASSIGNMENT - 4

### 1. Order by Clause

QUERY : select \* from movie order by date1;

OUTPUT:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
32 • select song_name,singer from songs where songs.film_id in
33 • (select direction.film_id from direction where numofflops=
34 • (select min(numofflops) from direction));
35 • use movies;
36 • select * from movie order by date1;
37
```

The Result Grid displays the following data:

#	film_id	title	date1	genre	rating
1	6	Trance	2020-07-12	methodological	8
2	5	Tenet	2020-09-01	action	7.5
3	4	master	2021-01-05	romance	8
4	2	acharya	2021-05-10	drama	9
5	1	pushpa	2021-07-21	action	8

The Action Output pane shows the execution steps:

#	Time	Action	Message	Duration / Fetch
1	11:19:21	select * from movie order by date1 LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clickin...	0.036 sec
2	11:20:38	use movies	0 row(s) affected	0.00053 sec
3	11:20:40	use movies	0 row(s) affected	0.00042 sec
4	11:20:46	select * from movie order by date1 LIMIT 0, 1000	6 row(s) returned	0.0011 sec / 0.0000

Query Completed

### 2. Group by and having

QUERY:

set sql\_mode=(select replace(@@sql\_mode,'ONLY\_FULL\_GROUP\_BY',''));

select \* from movie group by genre having rating >7;

OUTPUT:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
36 • select * from movie order by date1;
37 • set sql_mode=(select replace(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
38 • select * from movie group by genre having rating >7;
39
40
```

The Result Grid displays the following data:

#	film_id	title	date1	genre	rating
1	1	pushpa	2021-07-21	action	8
2	3	RRR	2022-04-01	action/drama	9.5
3	2	acharya	2021-05-10	drama	9
4	6	Trance	2020-07-12	methodological	8
5	4	master	2021-01-05	romance	8

The Action Output pane shows the execution steps:

#	Time	Action	Message	Duration / Fetch
4	11:20:46	select * from movie order by date1 LIMIT 0, 1000	6 row(s) returned	0.0011 sec / 0.0000...
5	11:43:59	select * from movie group by genre having rating >7 LIM...	Error Code: 1055. Expression #1 of SELECT list is ...	0.00030 sec
6	11:44:50	set sql_mode=(select replace(@@sql_mode,'ONLY_FUL...	0 row(s) affected	0.00059 sec
7	11:44:53	select * from movie group by genre having rating >7 LIM...	5 row(s) returned	0.00092 sec / 0.000...

Query Completed

### 3. Aggregate functions

QUERY: `select avg(rating) from movie where left(date1,4)=2020;`

OUTPUT:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
37 • set sql_mode=(select replace(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
38 • select * from movie group by genre having rating >7;
39 • select avg(rating) from movie where left(date1,4)=2020;
40
41
```

The Result Grid shows the output of the query:

#	avg(rating)
1	7.75

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
5	11:43:59	select * from movie group by genre having rating >7 LIMIT...	Error Code: 1055. Expression #1 of SELECT list is ...	0.00030 sec
6	11:44:50	set sql_mode=(select replace(@@sql_mode,'ONLY_FUL...	0 row(s) affected	0.00059 sec
7	11:44:53	select * from movie group by genre having rating >7 LIM...	5 row(s) returned	0.00092 sec / 0.000...
8	11:46:41	select avg(rating) from movie where left(date1,4)=2020 ...	1 row(s) returned	0.00091 sec / 0.000...

Query Completed

QUERY: `select count(film_id) from movie where rating>7;`

OUTPUT:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
38 • select * from movie group by genre having rating >7;
39 • select avg(rating) from movie where left(date1,4)=2020;
40 • select count(film_id) from movie where rating>7;
41
42
```

The Result Grid shows the output of the query:

#	count(film_id)
1	6

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
6	11:44:50	set sql_mode=(select replace(@@sql_mode,'ONLY_FUL...	0 row(s) affected	0.00059 sec
7	11:44:53	select * from movie group by genre having rating >7 LIM...	5 row(s) returned	0.00092 sec / 0.000...
8	11:46:41	select avg(rating) from movie where left(date1,4)=2020 ...	1 row(s) returned	0.00091 sec / 0.000...
9	11:47:30	select count(film_id) from movie where rating>7 LIMIT 0, ...	1 row(s) returned	0.00033 sec / 0.000...

Query Completed

#### 4. Logical operators especially with LIKE

QUERY:select \* from movie where title like 't%';

OUTPUT:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
39 • select avg(rating) from movie where left(date1,4)=2020;
40 • select count(film_id) from movie where rating>7;
41 • select * from movie where title like 't%';
42
43
```

The Results window displays the output of the third query, showing two rows of movie data:

#	film_id	title	date1	genre	rating
1	5	Tenet	2020-09-01	action	7.5
2	6	Trance	2020-07-12	methodological	8

The Action Output window shows the execution details for the query:

#	Time	Action	Message	Duration / Fetch
7	11:44:53	select * from movie group by genre having rating>7 LIMIT 0, ...	5 row(s) returned	0.00092 sec / 0.000...
8	11:46:41	select avg(rating) from movie where left(date1,4)=2020 ...	1 row(s) returned	0.00091 sec / 0.000...
9	11:47:30	select count(film_id) from movie where rating>7 LIMIT 0, ...	1 row(s) returned	0.00033 sec / 0.000...
10	11:48:21	select * from movie where title like 't%' LIMIT 0, 1000	2 row(s) returned	0.00036 sec / 0.000...

#### 5. At least 4 Nested queries specific to your Database, out of which at least 2 should have multiple subquery.

QUERY 1 : select film\_id,title from movie where film\_id in (select film\_id from songs where singer='anirudh');

OUTPUT 1:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
40 • select count(film_id) from movie where rating>7;
41 • select * from movie where title like 't%';
42 • select film_id,title from movie where film_id in
43 (select film_id from songs where singer='anirudh');
44
45
```

The Results window displays the output of the third query, showing one row of movie data:

#	film_id	title
1	1	pushpa

The Action Output window shows the execution details for the query:

#	Time	Action	Message	Duration / Fetch
8	11:46:41	select avg(rating) from movie where left(date1,4)=2020 ...	1 row(s) returned	0.00091 sec / 0.000...
9	11:47:30	select count(film_id) from movie where rating>7 LIMIT 0, ...	1 row(s) returned	0.00033 sec / 0.000...
10	11:48:21	select * from movie where title like 't%' LIMIT 0, 1000	2 row(s) returned	0.00036 sec / 0.000...
11	11:49:20	select film_id,title from movie where film_id in (select fil...	1 row(s) returned	0.035 sec / 0.00000...

QUERY 2:|select \* from songs where songs.film\_id in  
(select movie.film\_id from movie where movie.genre='action');  
OUTPUT 2 :

MySQL Workbench interface showing the execution of Query 2. The query is as follows:

```

41. select * from movie where title like 't%';
42. select film_id,title from movie where film_id in
43. (select film_id from songs where singer='anirudh');
44. select * from songs where songs.film_id in
45. (select movie.film_id from movie where movie.genre='action');
46

```

The Result Grid displays the following data:

#	film_id	song_name	singer	composer
1	1	vaathi coming	anirudh	anirudh

The Action Output tab shows the execution progress of the query, indicating that 1 row(s) were returned.

QUERY 3 :select film\_id,title from movie where rating>  
(select avg(rating) from movie where genre=  
(select genre from movie where movie.film\_id=1))and  
genre= (select genre from movie where movie.film\_id=1);  
OUTPUT 3:

MySQL Workbench interface showing the execution of Query 3. The query is as follows:

```

43. (select film_id from songs where singer='anirudh');
44. select * from songs where songs.film_id in
45. (select movie.film_id from movie where movie.genre='action');
46. select film_id,title from movie where rating>
47. (select avg(rating) from movie where genre=
48. (select genre from movie where movie.film_id=1))and
49. genre= (select genre from movie where movie.film_id=1);

```

The Result Grid displays the following data:

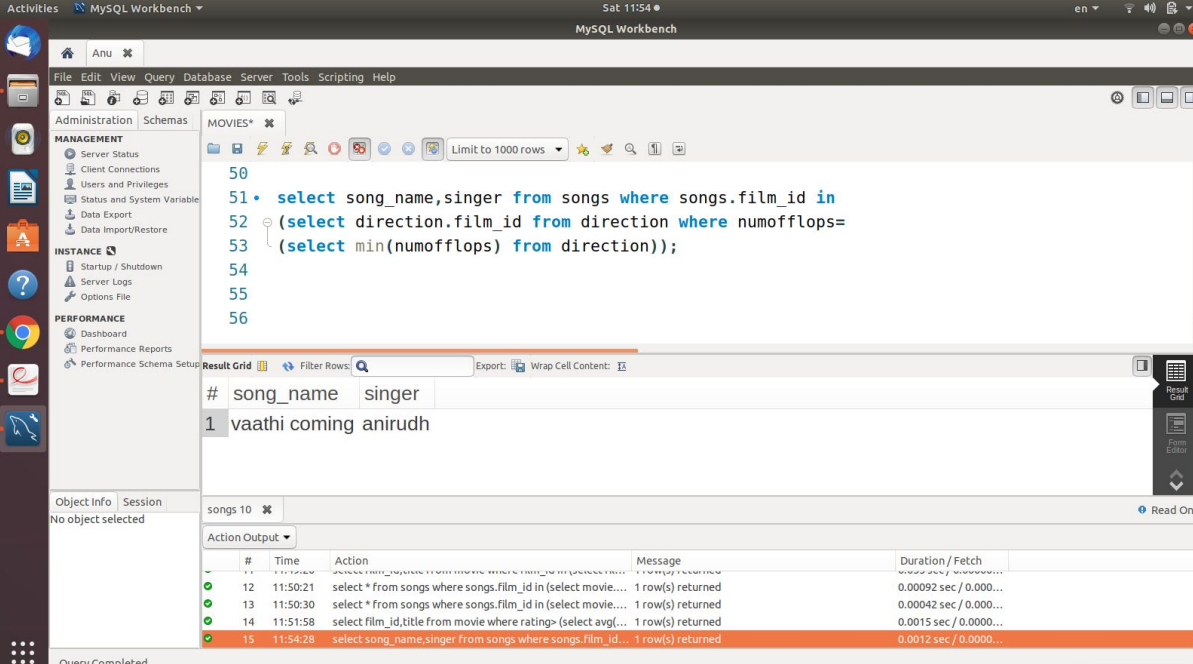
#	film_id	title
1	1	pushpa

The Action Output tab shows the execution progress of the query, indicating that 1 row(s) were returned.



QUERY 4 :select song\_name,singer from songs where songs.film\_id in  
(select direction.film\_id from direction where numofflops=  
(select min(numofflops) from direction));

OUTPUT 4:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```

50
51 • select song_name,singer from songs where songs.film_id in
52 (select direction.film_id from direction where numofflops=
53 (select min(numofflops) from direction));
54
55
56

```

The Result Grid shows the following data:

#	song_name	singer
1	vaathi coming	anirudh

The Action Output shows the following execution steps:

#	Time	Action	Message	Duration / Fetch
12	11:50:21	select * from songs where songs.film_id in (select movie...	1 row(s) returned	0.00092 sec / 0.000...
13	11:50:30	select * from songs where songs.film_id in (select movie...	1 row(s) returned	0.00042 sec / 0.000...
14	11:51:58	select film_id,title from movie where rating> (select avg(...	1 row(s) returned	0.0015 sec / 0.0000...
15	11:54:28	select song_name,singer from songs where songs.film_id...	1 row(s) returned	0.0012 sec / 0.0000...

Query Completed

QUERY3 EXPLANATION: /The inner query returns the average of the ratings of those movies with genre same as the genre of movies with movie\_id=134. The outer query displays the movie details with rating > the value returned by the inner query and with genre same as that with movie\_id=1./

QUERY 4 EXPLANATION: /\*Displays the song name and singer from the songs of movies directed by directors with no\_of\_flops greater than the minimum number of flops of any director from the director table\*/