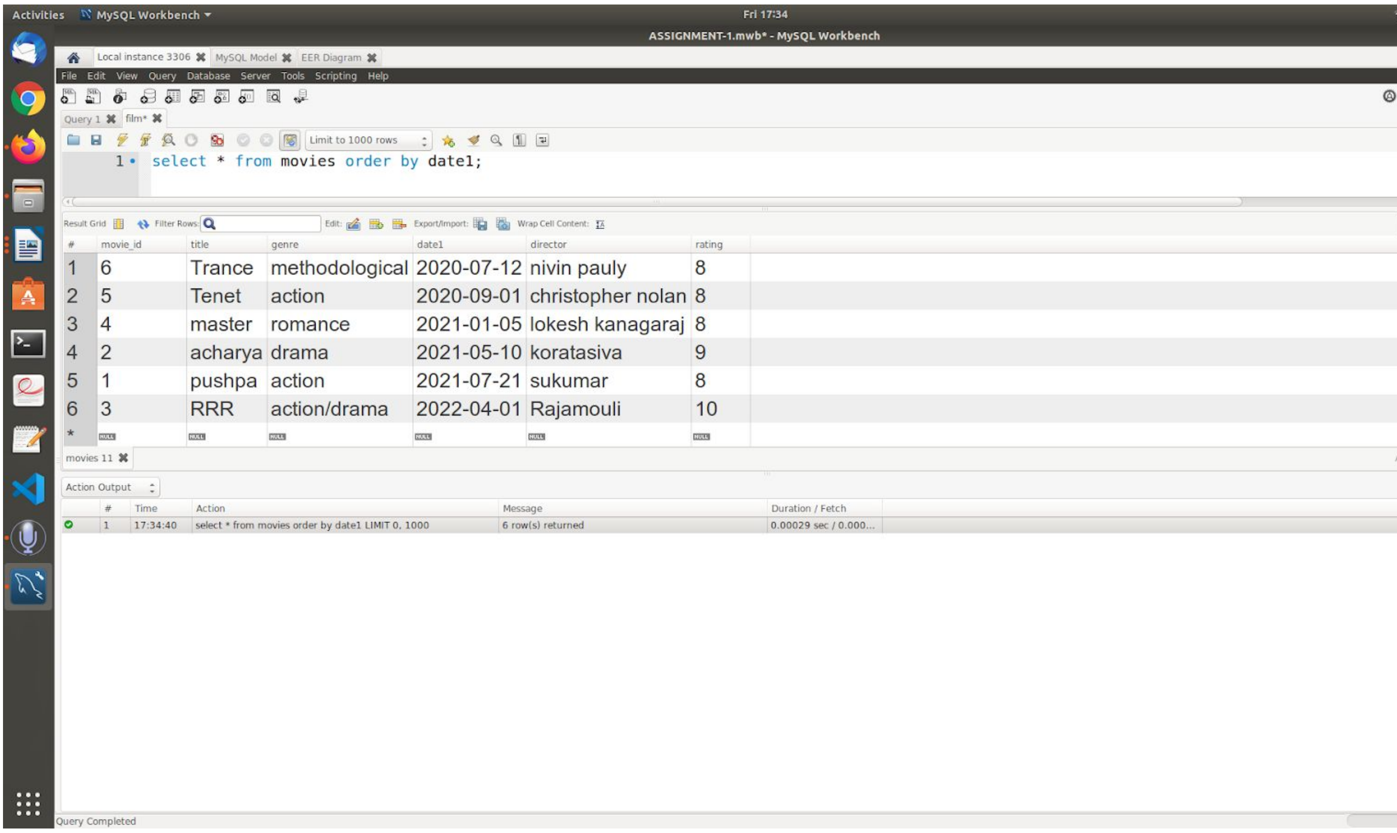
ASSIGNMENT-4

1. Order by Clause

SQL Query : select * from movies order by date1;

Output :



```
select * from movies order by date1;
```

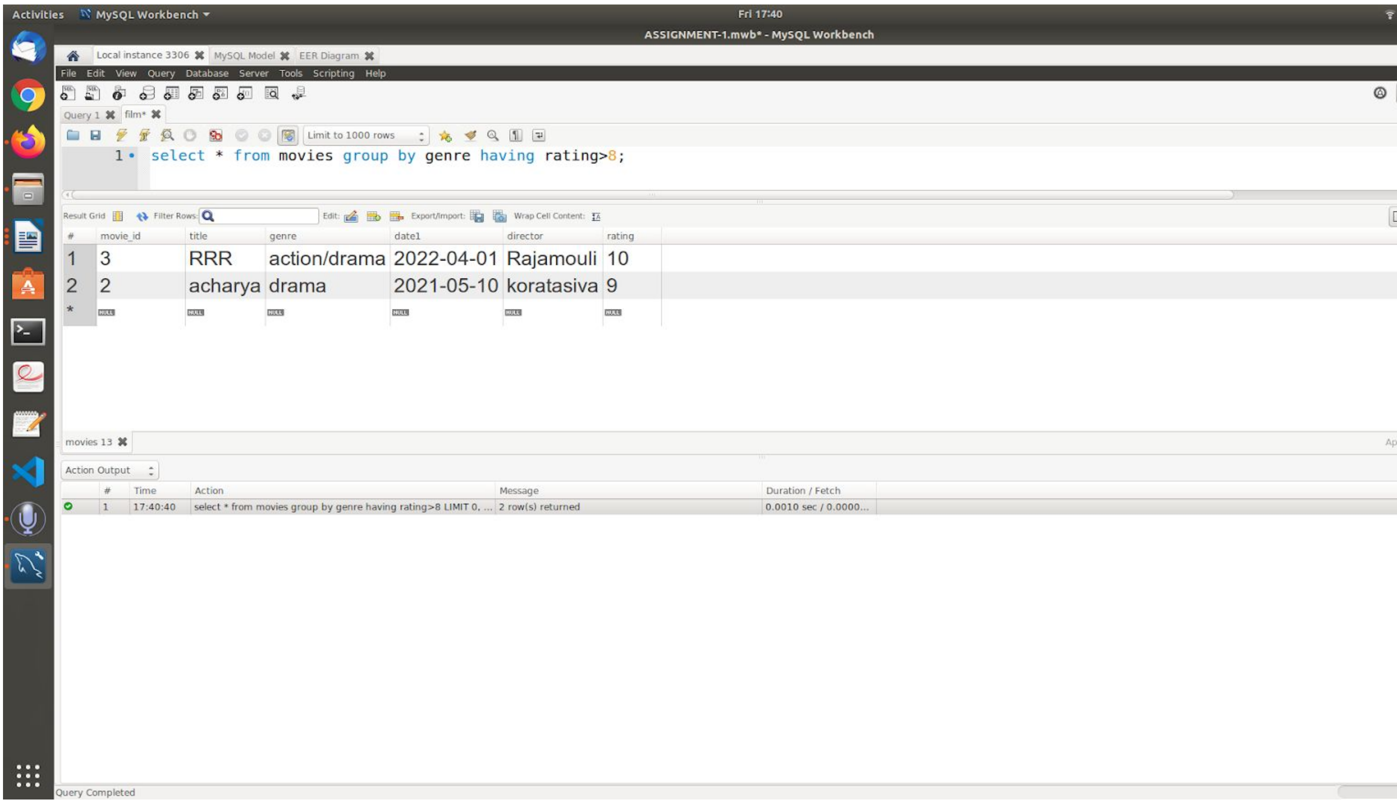
#	movie_id	title	genre	date1	director	rating
1	6	Trance	methodological	2020-07-12	nivin pauly	8
2	5	Tenet	action	2020-09-01	christopher nolan	8
3	4	master	romance	2021-01-05	lokesh kanagaraj	8
4	2	acharya	drama	2021-05-10	koratasiva	9
5	1	pushpa	action	2021-07-21	sukumar	8
6	3	RRR	action/drama	2022-04-01	Rajamouli	10
*	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	17:34:40	select * from movies order by date1 LIMIT 0, 1000	6 row(s) returned	0.00029 sec / 0.000...

2. Group by and having

SQL Query : select * from movies group by genre having rating>8;

Output :



```
select * from movies group by genre having rating>8;
```

#	movie_id	title	genre	date1	director	rating
1	3	RRR	action/drama	2022-04-01	Rajamouli	10
2	2	acharya	drama	2021-05-10	koratasiva	9
*	NULL	NULL	NULL	NULL	NULL	NULL

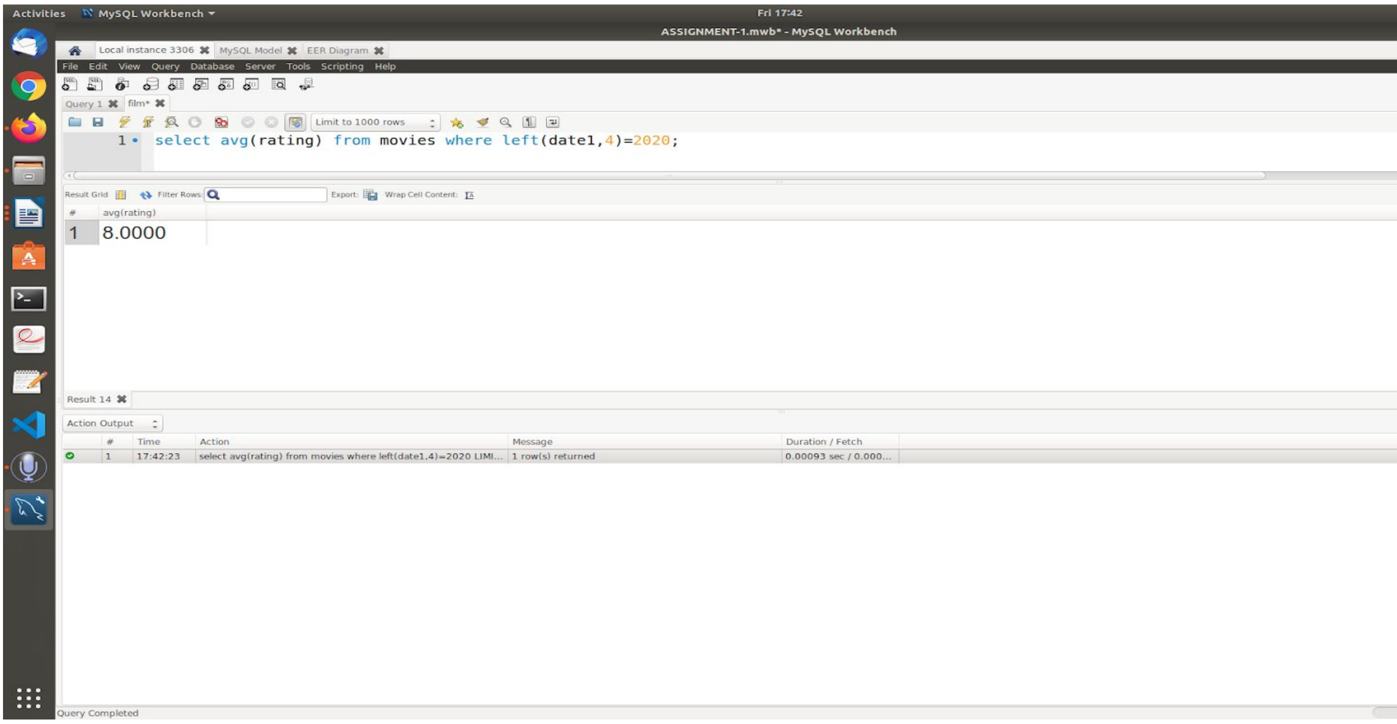
	#	Time	Action	Message	Duration / Fetch
✓	1	17:40:40	select * from movies group by genre having rating>8 LIMIT 0, ...	2 row(s) returned	0.0010 sec / 0.0000...

Query Completed

3. Aggregate functions

SQL Query : select avg(rating) from movies where left(date1,4)=2020;

Output :



```
select avg(rating) from movies where left(date1,4)=2020;
```

#	avg(rating)
1	8.0000

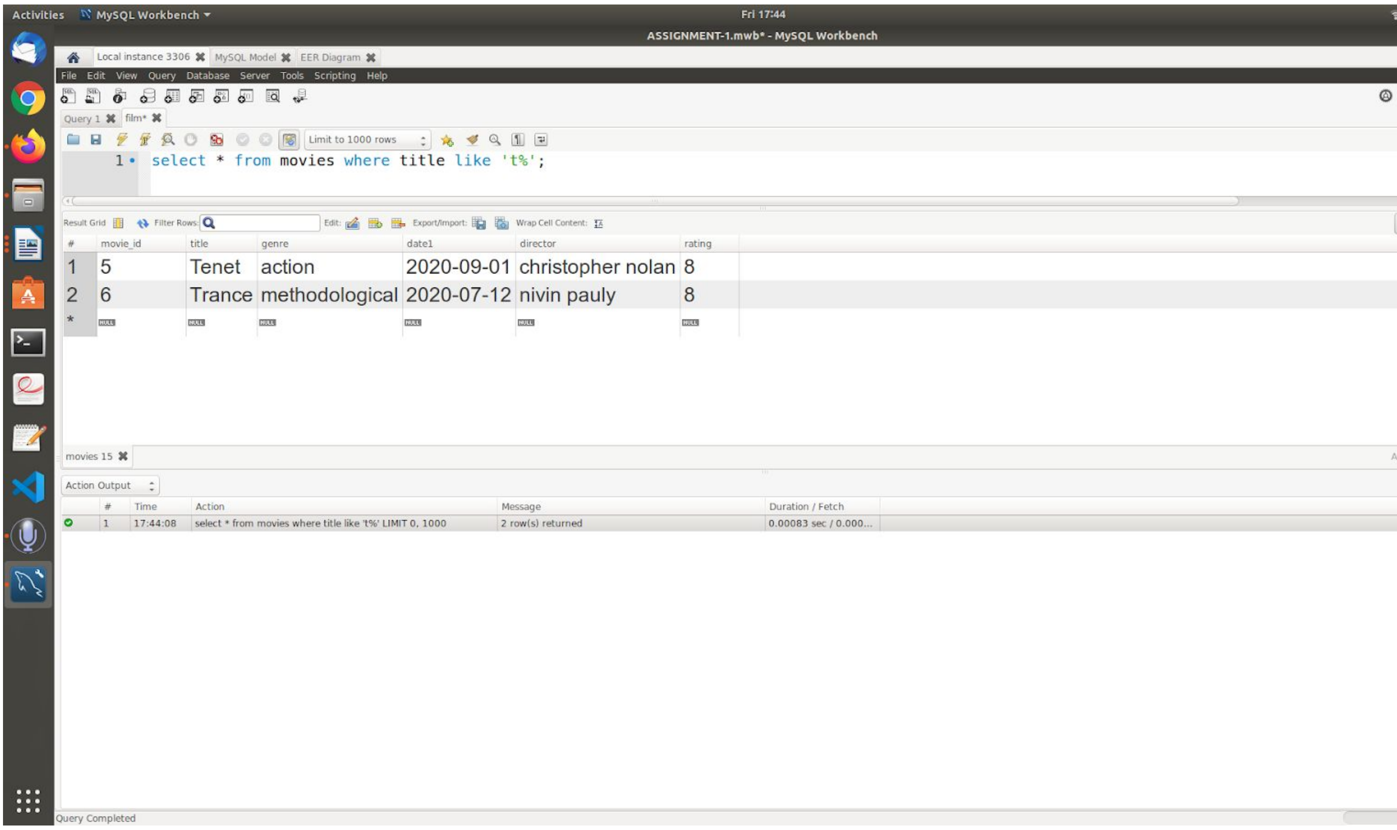
	#	Time	Action	Message	Duration / Fetch
✓	1	17:42:23	select avg(rating) from movies where left(date1,4)=2020 LIMI...	1 row(s) returned	0.00093 sec / 0.000...

Query Completed

4. Logical operators especially with LIKE

SQL Query : select * from movies where title like 't%';

Output :



```
select * from movies where title like 't%';
```

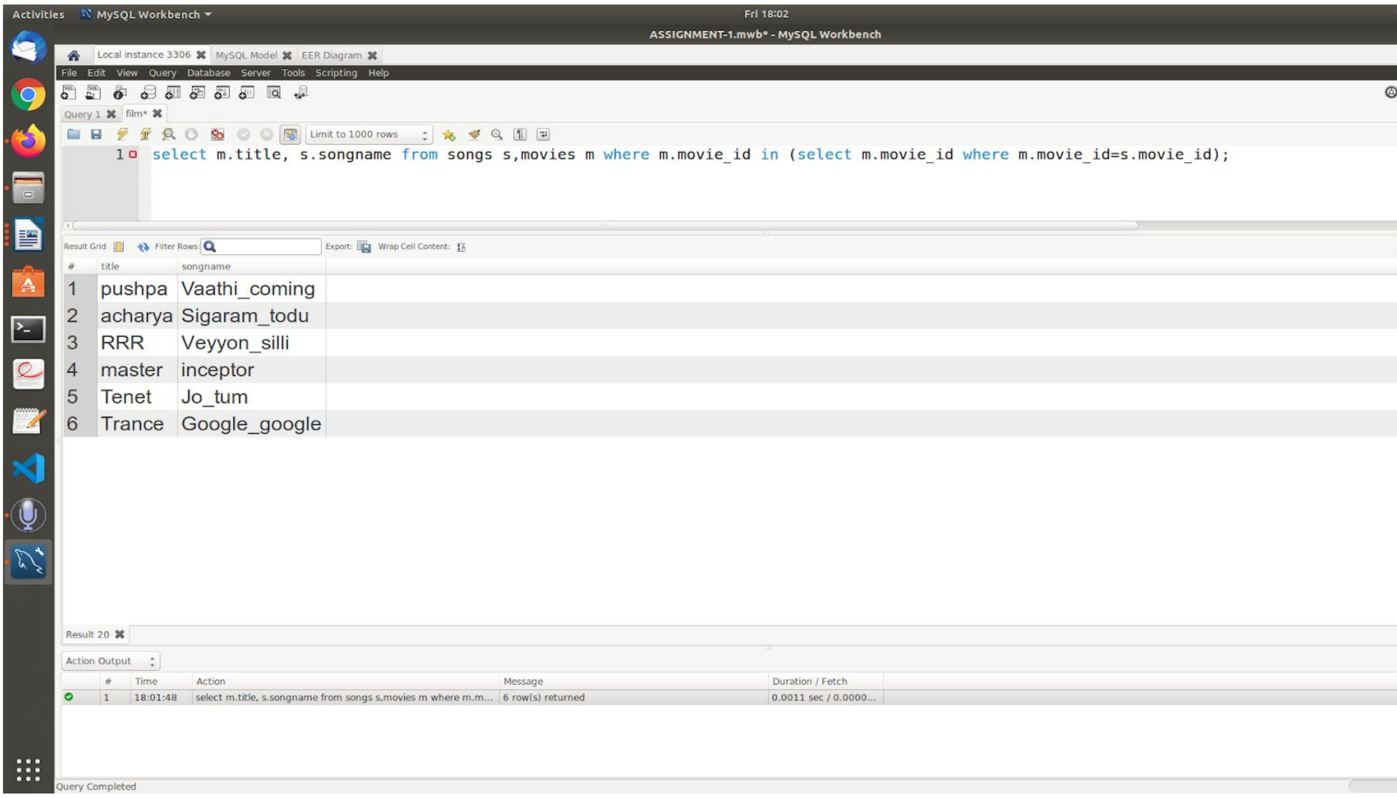
#	movie_id	title	genre	date1	director	rating
1	5	Tenet	action	2020-09-01	christopher nolan	8
2	6	Trance	methodological	2020-07-12	nivin pauly	8
*	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	17:44:08	select * from movies where title like 't%' LIMIT 0, 1000	2 row(s) returned	0.00083 sec / 0.000...

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

i) SQL Query :select m.title, s.songname from songs s,movies m where m.movie_id in (select m.movie_id where m.movie_id=s.movie_id);

Output:



```
select m.title, s.songname from songs s,movies m where m.movie_id in (select m.movie_id where m.movie_id=s.movie_id);
```

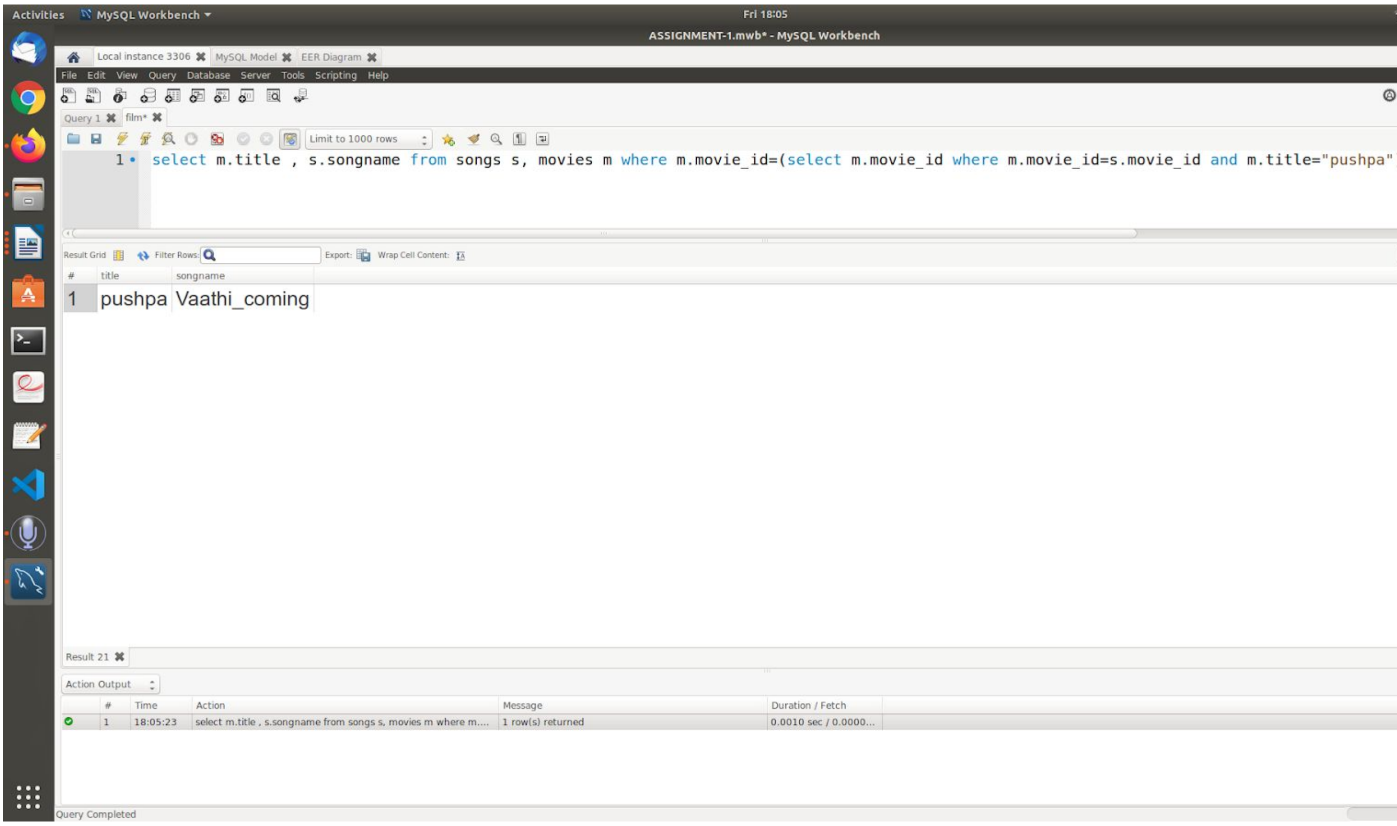
#	title	songname
1	pushpa	Vaathi_coming
2	acharya	Sigaram_todu
3	RRR	Veyyon_silli
4	master	inceptor
5	Tenet	Jo_tum
6	Trance	Google_google

#	Time	Action	Message	Duration / Fetch
1	18:01:48	select m.title, s.songname from songs s,movies m where m.m...	6 row(s) returned	0.0011 sec / 0.0000...

ii)SQL Query :

select m.title , s.songname from songs s, movies m where m.movie_id=(select m.movie_id where m.movie_id=s.movie_id and m.title="pushpa");

Output:



```
select m.title , s.songname from songs s, movies m where m.movie_id=(select m.movie_id where m.movie_id=s.movie_id and m.title="pushpa"
```

#	title	songname
1	pushpa	Vaathi_coming

Result 21

Action Output

	#	Time	Action	Message	Duration / Fetch
✓	1	18:05:23	select m.title , s.songname from songs s, movies m where m....	1 row(s) returned	0.0010 sec / 0.0000...

Query Completed

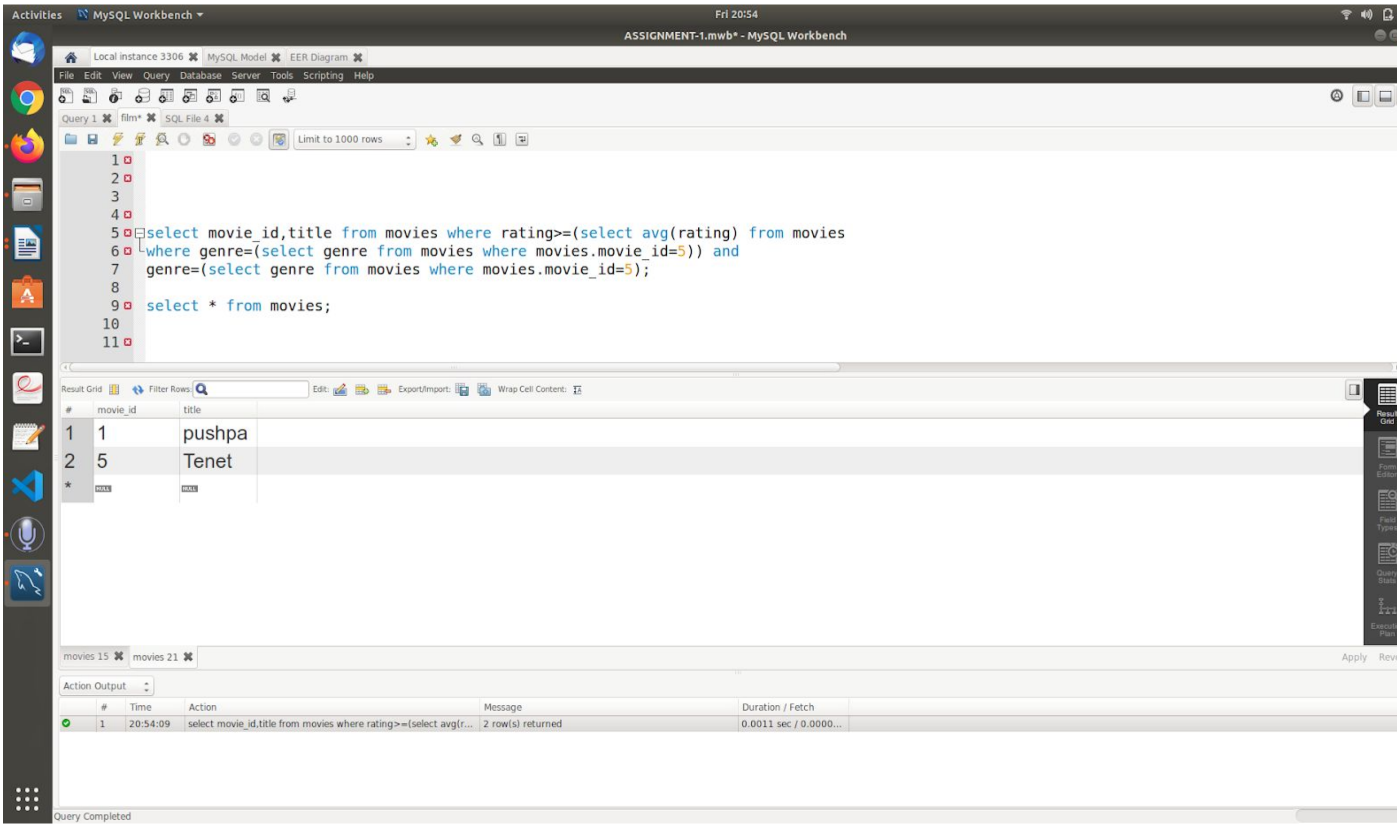
iii)SQL Query :

select movie_id,title from movies where rating>=(select avg(rating) from movies where genre=(select genre from movies where movies.movie_id=5)) and genre=(select genre from movies where movies.movie_id=5);

Explanation :

It selects the genre of movie_id=5 and then finds the average of ratings corresponding to that particular genre ,now the outer queries displays the details of movies having rating greater than equal to the value returned by inner query and genre same as the one with movie_id=5

Output:



```
select movie_id,title from movies where rating>=(select avg(rating) from movies
where genre=(select genre from movies where movies.movie_id=5)) and
genre=(select genre from movies where movies.movie_id=5);

select * from movies;
```

#	movie_id	title
1	1	pushpa
2	5	Tenet
*	NULL	NULL

	#	Time	Action	Message	Duration / Fetch
✓	1	20:54:09	select movie_id,title from movies where rating>=(select avg(r...	2 row(s) returned	0.0011 sec / 0.0000...

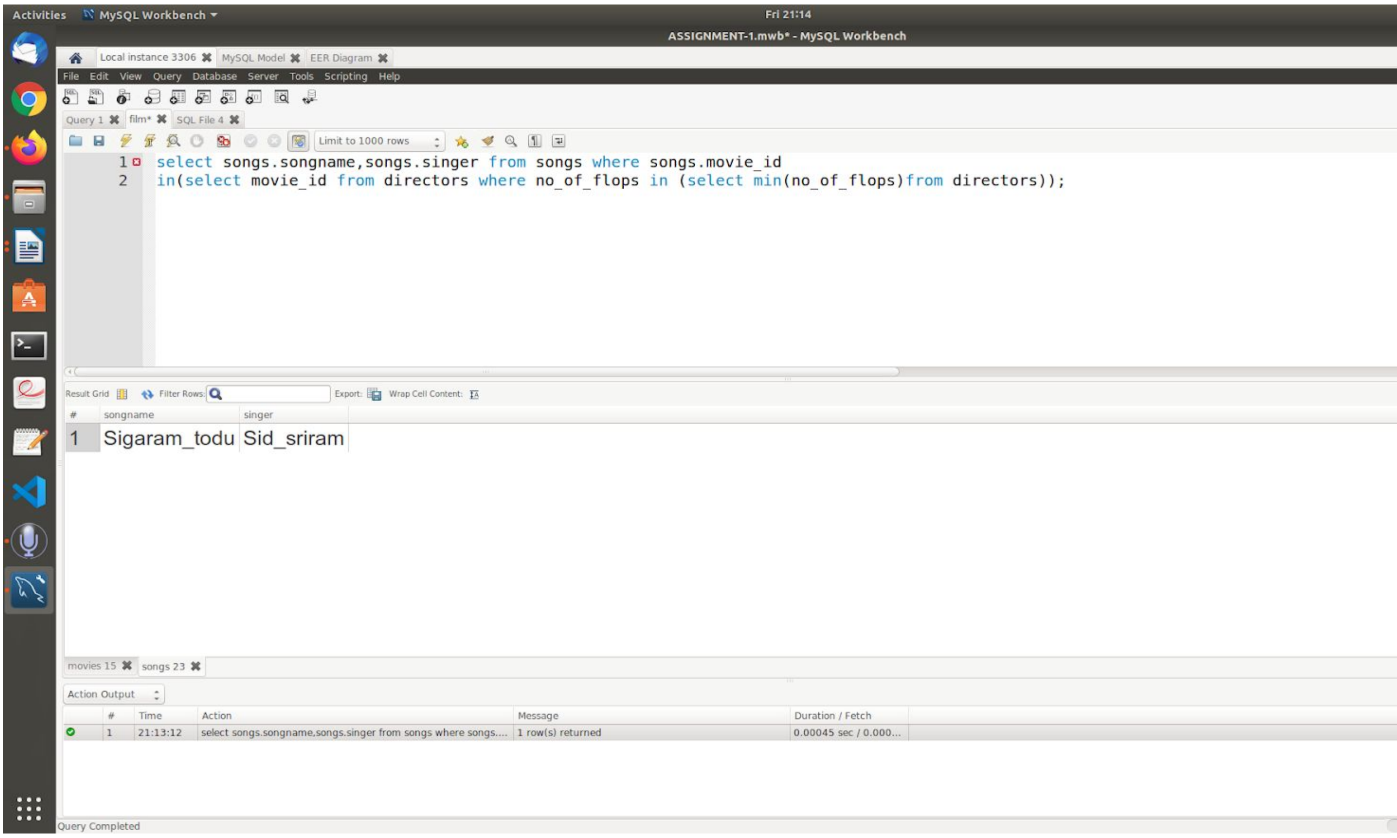
‘iv)SQL Query

select songs.songname,songs.singer from songs where songs.movie_id in(select movie_id from directors where no_of_flops in (select min(no_of_flops)from directors));

Explanation :

Displays song name and singer from songs table where number_of_flops directed by particular director is minimum

Output :



```
select songs.songname,songs.singer from songs where songs.movie_id
in(select movie_id from directors where no_of_flops in (select min(no_of_flops)from directors));
```

#	songname	singer
1	Sigaram_todu	Sid_sriram

	#	Time	Action	Message	Duration / Fetch
✓	1	21:13:12	select songs.songname,songs.singer from songs where songs....	1 row(s) returned	0.00045 sec / 0.000...

Query Completed