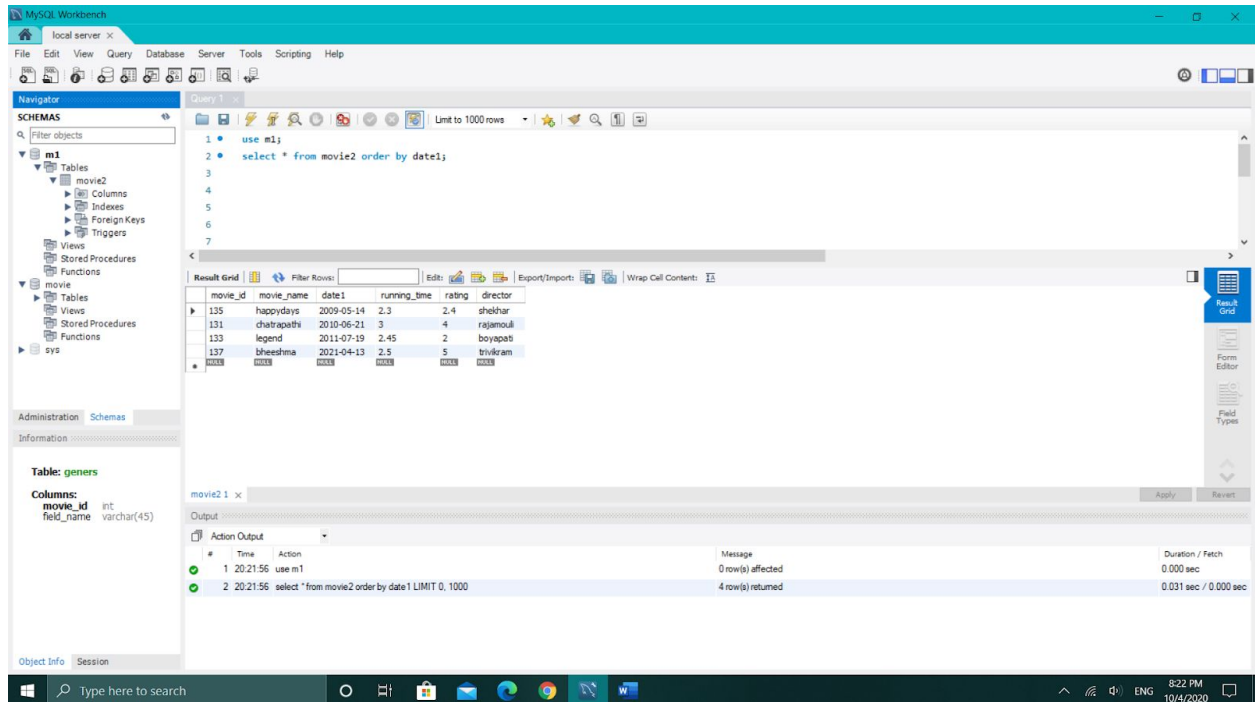


1.order by clause

ANS:- SELECT \* FROM movie1 order by date1;



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1. use m1;
2. select * from movie2 order by date1;
3.
4.
5.
6.
7.
```

The 'Result Grid' shows the following data:

movie_id	movie_name	date1	running_time	rating	director
135	happydays	2009-05-14	2.3	2.4	shelhar
131	chhatrapathi	2010-06-21	3	4	rajanouli
133	legend	2011-07-19	2.45	2	boyapati
137	bheeshma	2021-04-13	2.5	5	trivikram

The 'Action Output' pane shows the following output:

#	Time	Action	Message	Duration / Fetch
1	20:21:56	use m1	0 row(s) affected	0.000 sec
2	20:21:56	select * from movie2 order by date1 LIMIT 0, 1000	4 row(s) returned	0.031 sec / 0.000 sec

2.Group by and having.

Ans. select \* from movie1 group by votes having rating >= 3;

MySQL Workbench interface showing a query execution. The query is:

```
1. use movie;
2. select * from movie1 group by votes having rating >=3;
```

The result grid shows the following data:

movie_id	title	date1	runningtime	rating	votes	gross	director
151	sahoo	2020-06-24	3	3	8000	9000000	sujeeth
153	bahubali	2018-04-18	3.1	4	2000	10000000	rajemouli
155	merchi	2013-02-10	2.3	3.5	6000	4000000	konatala siva
159	intersteller	2020-06-24	2.35	3	4000	6000000	nolan

The output pane shows the execution steps and their durations:

#	Time	Action	Message	Duration / Fetch
1	20:21:56	use m1	0 row(s) affected	0.000 sec
2	20:21:56	select * from movie2 order by date1 LIMIT 0, 1000	4 row(s) returned	0.031 sec / 0.000 sec
3	20:27:09	use movie	0 row(s) affected	0.000 sec
4	20:27:10	select * from movie1 group by votes having rating >=3 LIMIT 0, 1000	4 row(s) returned	0.047 sec / 0.000 sec

### 3. Aggregate functions.

Ans. select avg(rating) from movies1;

MySQL Workbench interface showing a query execution. The query is:

```
1. use movie;
2. select avg(rating) from movie1;
```

The result grid shows the following data:

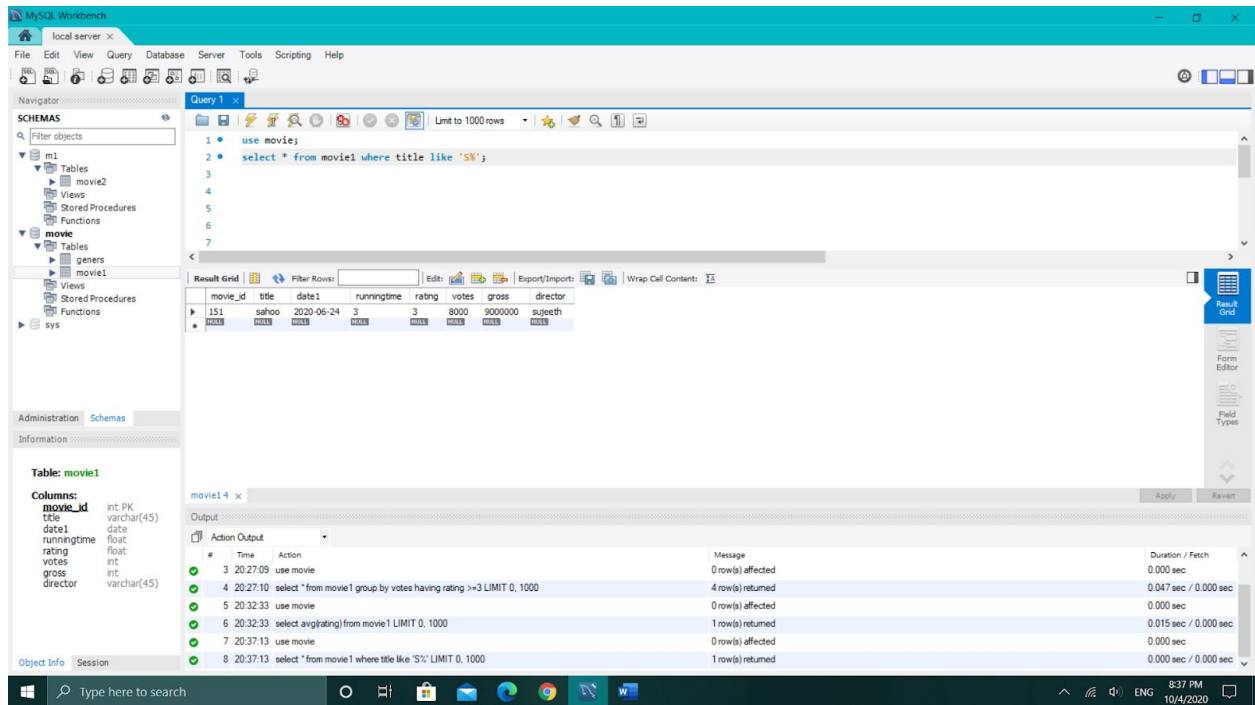
avg(rating)
3.2

The output pane shows the execution steps and their durations:

#	Time	Action	Message	Duration / Fetch
1	00:39:43	use movie	0 row(s) affected	0.000 sec
2	00:39:43	select avg(rating) from movie1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

#### 4. Logical operators especially with LIKE.

Ans. `select * from movies1 where title like 's%';`



The screenshot displays the MySQL Workbench interface. The 'Query 1' editor contains the following SQL code:

```
1 use movie1;
2 select * from movie1 where title like 'S%';
3
4
5
6
7
```

The 'Result Grid' shows the execution results for the query. The columns are: movie\_id, title, date1, runtime, rating, votes, gross, and director. The results are as follows:

movie_id	title	date1	runtime	rating	votes	gross	director
151	sahoo	2020-06-24	3	3	8000	9000000	sujeth

The 'Table: movie1' structure is also visible, showing columns: movie\_id (int PK), title (varchar(45)), date1 (date), runtime (float), rating (float), votes (int), gross (int), and director (varchar(45)).

The 'Action Output' pane shows the execution steps and their durations:

#	Time	Action	Message	Duration / Fetch
3	20:27:09	use movie1	0 row(s) affected	0.000 sec
4	20:27:10	select * from movie1 group by votes having rating >=3 LIMIT 0, 1000	4 row(s) returned	0.047 sec / 0.000 sec
5	20:32:33	use movie1	0 row(s) affected	0.000 sec
6	20:32:33	select avg(rating) from movie1 LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
7	20:37:13	use movie1	0 row(s) affected	0.000 sec
8	20:37:13	select * from movie1 where title like 'S%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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

Ans.

A. `select m.title, s.filed_name from genres s, movie1 m where m.movie_id in (select m.movie_id where m.movie_id=s.movie_id);`

MySQL Workbench interface showing a query execution. The query is:

```
1. use movie;
2. select m.title, s.field_name from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id);
```

The result grid shows the following data:

	title	field_name
1	sahoo	adventure
2	bahubali	action/drama
3	merchi	action/drama
4	pokiri	action/triller
5	intersteller	sc-fi/adventure

The output pane shows the following messages:

```
1 23:44:16 use movie 0 row(s) affected 0.000 sec
2 23:44:16 select m.title, s.field_name from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id); 5 row(s) returned 0.016 sec / 0.000 sec
```

B. select m.title , s.feild\_name from songs s, movie1 m where m.movie\_id=(select m.movie\_id where m.movie\_id=s.movie\_id and m.title="bahubali");

MySQL Workbench interface showing a query execution. The query is:

```
1. use movie;
2. select m.title, s.field_name from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id and m.title="bahubali");
```

The result grid shows the following data:

	title	field_name
1	bahubali	action/drama

The output pane shows the following messages:

```
1 23:44:16 use movie 0 row(s) affected 0.000 sec
2 23:44:16 select m.title, s.field_name from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id and m.title="bahubali"); 5 row(s) returned 0.016 sec / 0.000 sec
3 23:45:35 use movie 0 row(s) affected 0.015 sec
4 23:45:35 select m.title, s.field_name from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id and m.title="bahubali"); 1 row(s) returned 0.000 sec / 0.000 sec
```

C.select concat(m.title," ", s.feild\_name) as movie\_details from genres s, movie1 m where m.movie\_id in( select m.movie\_id where m.movie\_id=s.movie\_id and m.title in(select m.title where m.title="sahoo"));

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

```
1 use movie;
2 select concat(m.title," ",s.field_name)as movie_details from genres s, movie1 m where m.movie_id in(select m.movie_id where m.movie_id=s.movie_id and m.title in(select m.title
3
4
5
6
7
```

The left sidebar shows the database schema with tables 'movie1', 'movie2', and 'movie'. The 'movie' table is selected, showing its structure.

The bottom panel shows the execution output:

#	Time	Action	Message	Duration / Fetch
1	00:20:47	use movie	0 row(s) affected	0.000 sec
2	00:20:47	select concat(m.title," ",s.field_name)as movie_details from genres s, movie1 m where m.movie_id in(select m.mo...	1 row(s) returned	0.000 sec / 0.000 sec

The result grid shows one row for the movie 'sahoo,adventure'.