

## 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:

```
8 * insert into movie values(3,'RRR','2022-04-01','action/drama',9.5);
9 * insert into movie values(4,'master','2021-01-05','romance',8);
10 * insert into movie values(5,'Tenet','2020-09-01','action',7.5);
11 * insert into movie values(6,'Trance','2020-07-12','methodological',8);
12 * select * from movie;
13 * select * from movie order by date1;
14 * set sql_mode=(select replace(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
15 * select * from movie group by genre having rating >7;
16 * select avg(rating) from movie where left(date1,4)>2020;
```

The 'Result Grid' shows the output of the query 'select \* from movie order by date1;'. The results are as follows:

| # | 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      |
| 6 | 3       | RRR     | 2022-04-01 | action/drama   | 9.5    |

The 'Action Output' table shows the execution details of the query:

| #  | Time     | Action   | Message           | Duration / Fetch        |
|----|----------|--|-------------------|-------------------------|
| 59 | 14:21:05 | insert into songs values(4,'memories','vineeth','sudhu')       | 1 row(s) affected | 0.034 sec               |
| 60 | 14:21:05 | insert into songs values(1,'shape of you','ed sheran','ed s... | 1 row(s) affected | 0.033 sec               |
| 61 | 14:21:35 | desc songs   | 4 row(s) returned | 0.0013 sec / 0.0000...  |
| 62 | 14:24:12 | select * from movie order by date1 LIMIT 0, 1000               | 6 row(s) returned | 0.00075 sec / 0.0000... |

### 2. Group by and having

QUERY: 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:

```
5 * create table songs(film_id int,song_name varchar(50),singer varchar(45),composer varchar(50));
6 * insert into movie values(1,'pushpa','2021-07-21','action',8);
7 * insert into movie values(2,'acharya','2021-05-10','drama',9);
8 * insert into movie values(3,'RRR','2022-04-01','action/drama',9.5);
9 * insert into movie values(4,'master','2021-01-05','romance',8);
10 * insert into movie values(5,'Tenet','2020-09-01','action',7.5);
11 * insert into movie values(6,'Trance','2020-07-12','methodological',8);
12 * select * from movie;
13 * select * from movie order by date1;
```

The 'Result Grid' shows the output of the query 'select \* from movie group by genre having rating >7;'. The results are as follows:

| # | 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' table shows the execution details of the query:

| #  | Time     | Action   | Message           | Duration / Fetch        |
|----|----------|--|-------------------|-------------------------|
| 60 | 14:21:05 | insert into songs values(1,'shape of you','ed sheran','ed s... | 1 row(s) affected | 0.033 sec               |
| 61 | 14:21:35 | desc songs   | 4 row(s) returned | 0.0013 sec / 0.0000...  |
| 62 | 14:24:12 | select * from movie order by date1 LIMIT 0, 1000               | 6 row(s) returned | 0.00075 sec / 0.0000... |
| 63 | 14:26:15 | select * from movie group by genre having rating >7 LIM...     | 5 row(s) returned | 0.00035 sec / 0.0000... |

### 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:

```
13 select * from movie order by date1;
14 set sql_mode=(select replace(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
15 select * from movie group by genre having rating >7;
16 select avg(rating) from movie where left(date1,4)=2020;
17 select * from movie where title like 't%';
18 insert into songs values(1,'vaathi coming','anirudh','anirudh');
19 insert into songs values(2,'hello','haricharan','anup rubens');
20 insert into songs values(3,'holy','justin beiber','justin beiber');
21 insert into songs values(4,'memories','vineeth','sudhu');
```

The Result Grid shows the output of the query:

| # | avg(rating) |
|---|-------------|
| 1 | 7.75        |

The Action Output pane shows the execution details of the query:

| #  | Time     | Action   | Message           | Duration / Fetch       |
|----|----------|--|-------------------|------------------------|
| 61 | 14:21:35 | desc songs   | 4 row(s) returned | 0.0013 sec / 0.000...  |
| 62 | 14:24:12 | select * from movie order by date1 LIMIT 0, 1000             | 6 row(s) returned | 0.00075 sec / 0.000... |
| 63 | 14:26:15 | select * from movie group by genre having rating >7 LIMIT... | 5 row(s) returned | 0.00035 sec / 0.000... |
| 64 | 14:27:30 | select avg(rating) from movie where left(date1,4)=2020 ...   | 1 row(s) returned | 0.00079 sec / 0.000... |

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:

```
20 insert into songs values(3,'holy','justin beiber','justin beiber');
21 insert into songs values(4,'memories','vineeth','sudhu');
22 insert into songs values(1,'shape of you','ed sheran','ed sheran');
23 select movie.film_id,movie.title from movie where movie.film_id in
24 (select songs.film_id from songs where songs.singer='anirudh');
25 select * from songs where songs.film_id in
26 (select movie.film_id from movie where movie.genre='action');
27 select count(film_id) from movie where rating>7;
28
```

The Result Grid shows the output of the query:

| # | count(film_id) |
|---|----------------|
| 1 | 6              |

The Action Output pane shows the execution details of the query:

| #   | Time     | Action   | Message  | Duration / Fetch       |
|-----|----------|--|--|------------------------|
| 100 | 15:50:21 | select concat(m.title,"",s.song_name,"",s.singer,"",co...    | 2 row(s) returned                                    | 0.00049 sec / 0.000... |
| 101 | 15:55:21 | select concat(m.title,"",s.song_name,"",s.singer,"",co...    | Error Code: 1064. You have an error in your SQL s... | 0.00057 sec            |
| 102 | 15:56:29 | select count(film_id) from movie where rating>8 LIMIT 0, ... | 1 row(s) returned                                    | 0.00078 sec / 0.000... |
| 103 | 15:56:44 | select count(film_id) from movie where rating>7 LIMIT 0, ... | 1 row(s) returned                                    | 0.00034 sec / 0.000... |

#### 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:

```
14 • set sql_mode=(select replace(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
15 • select * from movie group by genre having rating >7;
16 • select avg(rating) from movie where left(date1,4)=2020;
17 • select * from movie where title like 't%';
18 • insert into songs values(1,'vaathi coming','anirudh','anirudh');
19 • insert into songs values(2,'hello','haricharan','anup rubens');
20 • insert into songs values(3,'holy','justin beiber','justin beiber');
21 • insert into songs values(4,'memories','vineeth','sudhu');
22 • insert into songs values(1,'shape of you','ed sheran','ed sheran');
```

The Result Grid shows the output of the query:

| # | 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 shows the execution of the query:

| #  | Time     | Action   | Message           | Duration / Fetch       |
|----|----------|--|-------------------|------------------------|
| 62 | 14:24:12 | select * from movie order by date1 LIMIT 0, 1000           | 6 row(s) returned | 0.00075 sec / 0.000... |
| 63 | 14:26:15 | select * from movie group by genre having rating >7 LIM... | 5 row(s) returned | 0.00035 sec / 0.000... |
| 64 | 14:27:30 | select avg(rating) from movie where left(date1,4)=2020 ... | 1 row(s) returned | 0.00079 sec / 0.000... |
| 65 | 14:28:02 | select * from movie where title like 't%' LIMIT 0, 1000    | 2 row(s) returned | 0.00076 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:

```
17 • create table songs(film_id int,song_name varchar(50),singer varchar(45),composer varchar(50),foreign key(film_id) references movie(film_i
18 • insert into songs values(1,'vaathi coming','anirudh','anirudh');
19 • insert into songs values(2,'hello','haricharan','anup rubens');
20 • insert into songs values(3,'holy','justin beiber','justin beiber');
21 • insert into songs values(4,'memories','vineeth','sudhu');
22 • insert into songs values(1,'shape of you','ed sheran','ed sheran');
23 • select film_id,title from movie where film_id in
    (select film_id from songs where singer='anirudh');
```

The Result Grid shows the output of the query:

| # | film_id | title  |
|---|---------|--------|
| 1 | 1       | pushpa |

The Action Output shows the execution of the query:

| #  | Time     | Action  | Message           | Duration / Fetch       |
|----|----------|---|-------------------|------------------------|
| 92 | 14:55:15 | insert into songs values(3,'holy','justin beiber','justin be... | 1 row(s) affected | 0.118 sec              |
| 93 | 14:55:15 | insert into songs values(4,'memories','vineeth','sudhu')        | 1 row(s) affected | 0.034 sec              |
| 94 | 14:55:16 | insert into songs values(1,'shape of you','ed sheran','ed s...  | 1 row(s) affected | 0.034 sec              |
| 95 | 14:59:31 | select film_id,title from movie where film_id in(select fil...  | 1 row(s) returned | 0.017 sec / 0.00001... |

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

OUTPUT 2 :

The screenshot shows MySQL Workbench with Query 1 executed. The query is as follows:

```

19 • insert into songs values(2,'hello','haricharan','anup rubens');
20 • insert into songs values(3,'holy','justin beiber','justin beiber');
21 • insert into songs values(4,'memories','vineeth','sudhu');
22 • insert into songs values(1,'shape of you','ed sheran','ed sheran');
23 • select movie.film_id,movie.title from movie where movie.film_id in
24       (select songs.film_id from songs where songs.singer='anirudh');
25 • select * from songs where songs.film_id in
26       (select movie.film_id from movie where movie.genre='action');
27

```

The Result Grid shows the following data:

| # | film_id | song_name     | singer    | composer  |
|---|---------|---------------|-----------|-----------|
| 1 | 1       | vaathi coming | anirudh   | anirudh   |
| 2 | 1       | shape of you  | ed sheran | ed sheran |

The Action Output shows the execution details for the query:

| #  | Time     | Action   | Message           | Duration / Fetch       |
|----|----------|--|-------------------|------------------------|
| 95 | 14:59:31 | select film_id,title from movie where film_id in(select fil... | 1 row(s) returned | 0.017 sec / 0.0001...  |
| 96 | 15:09:27 | select * from songs where film_id in(select film_id from ...   | 2 row(s) returned | 0.00033 sec / 0.000... |
| 97 | 15:20:39 | select movie.film_id,movie.title from movie where movi...      | 1 row(s) returned | 0.00055 sec / 0.000... |
| 98 | 15:22:14 | select * from songs where songs.film_id in ...                 | 2 row(s) returned | 0.00043 sec / 0.000... |

QUERY 3 : select concat(m.title,"", s.song\_name,"", s.singer) as song\_details from songs  
s,movie m where m.film\_id in  
( select m.film\_id where m.film\_id=s.film\_id and m.title in  
(select m.title where m.title="Master"));

OUTPUT 3 :

The screenshot shows MySQL Workbench with Query 3 executed. The query is as follows:

```

19 •
20 •
21 •
22 •
23 •
24 • songs where songs.singer='anirudh';
25 •
26 • movie.genre='action';
27 • songs s, movie m where m.film_id in (select m.film_id where m.film_id=s.film_id and m.title in(select m.title where m.title="Master"));

```

The Result Grid shows the following data:

| # | song_details            |
|---|-------------------------|
| 1 | master,memories,vineeth |

The Action Output shows the execution details for the query:

| #  | Time     | Action   | Message           | Duration / Fetch       |
|----|----------|--|-------------------|------------------------|
| 96 | 15:09:27 | select * from songs where film_id in(select film_id from ... | 2 row(s) returned | 0.00033 sec / 0.000... |
| 97 | 15:20:39 | select movie.film_id,movie.title from movie where movi...    | 1 row(s) returned | 0.00055 sec / 0.000... |
| 98 | 15:22:14 | select * from songs where songs.film_id in ...               | 2 row(s) returned | 0.00043 sec / 0.000... |
| 99 | 15:41:35 | select concat(m.title,"",s.song_name,"",s.singer) as so...   | 1 row(s) returned | 0.00099 sec / 0.000... |



QUERY 4 : select concat(m.title,"", s.song\_name,"", s.singer,"",composer) as  
 artist\_involved\_in\_action\_movies from songs s, movie m where m.film\_id in  
 (select m.film\_id where m.film\_id=s.film\_id and m.genre in  
 (select m.genre where m.genre="Action"));

OUTPUT 4 :

The screenshot shows the MySQL Workbench interface. The main window displays a query (Query 1) that concatenates movie titles, song names, singers, and composers for action movies. The query is as follows:

```

22 • js values(1,'shape of you','ed sheran','ed sheran');
23 • ilm_id,movie.title from movie where movie.film_id in
24   (select songs.film_id from songs where songs.singer='anirudh');
25 • ongs where songs.film_id in
26   (select movie.film_id from movie where movie.genre='action');
27 • .title,"", s.song_name,"", s.singer) as song_details from songs s, movie m where m.film_id in( select m.film_id where m.film_id=s.film_
28 • n.title,"", s.song_name,"", s.singer,"",composer) as artist_involved_in_action_movies from songs s, movie m where m.film_id in
29   (select m.film_id where m.film_id=s.film_id and m.genre in
30   (select m.genre where m.genre="Action"));
  
```

The results are displayed in the 'Result Grid' tab, showing two rows of data:

| # | artist_involved_in_action_movies     |
|---|--------------------------------------|
| 1 | pushpa,vaathi coming,anirudh,anirudh |
| 2 | pushpa,shape of you,ed sheran,ed ... |

Below the result grid, the 'Action Output' tab shows the execution progress of the query, including the time taken and the number of rows returned for each step.

| #   | Time     | Action   | Message           | Duration / Fetch       |
|-----|----------|--|-------------------|------------------------|
| 97  | 15:20:39 | select movie.film_id,movie.title from movie where movi...    | 1 row(s) returned | 0.00055 sec / 0.000... |
| 98  | 15:22:14 | select * from songs where songs.film_id in                   | 2 row(s) returned | 0.00043 sec / 0.000... |
| 99  | 15:41:35 | select concat(m.title,"", s.song_name,"", s.singer) as so... | 1 row(s) returned | 0.00099 sec / 0.000... |
| 100 | 15:50:21 | select concat(m.title,"", s.song_name,"", s.singer,"",co...  | 2 row(s) returned | 0.00049 sec / 0.000... |

The query is completed, and the status bar at the bottom indicates 'Query Completed'.