

Assignment-5

A. CREATING TABLES

1. movie table

The screenshot shows the MySQL Workbench interface with the 'movie' table being created. The SQL editor contains the following code:

```
1 create table movie (  
2     mov_id int not null,  
3     mov_title char(45) not null,  
4     mov_rel_date_year date not null,  
5     mov_lang CHAR(45) not null,  
6     mov_country CHAR(45) not null,  
7     mov_duration int not null,  
8     primary key(mov_id)  
9 );  
10  
11 insert into movie(mov_id,mov_title,mov_rel_date_year ,mov_lang ,mov_country ,mov_duration)  
12 values(300,'chakde','2017-06-15','hindi','india','2');  
13 insert into movie values(400,'gold','2016-06-18','hindi','india','3');  
14 insert into movie values(500,'3 Idiots','2008-07-14','hindi','india','2');  
15 insert into movie values(600,'taare zameen par','2008-07-10','hindi','india','2');  
16 insert into movie values(900,'Harry potter and the prisoner of askaban','2008-05-11','english','USA','4');  
17 insert into movie values(1000,'Harry potter and the deathly hallows 1','2011-05-21','english','USA','5');  
18 insert into movie values(700,'Mission Impossible 1','2012-06-14','english','USA','6');  
19  
20 select * from movie;
```

The output window shows the execution of the SQL script, with 7 rows returned for the final select statement.

2. mov_genre table

The screenshot shows the MySQL Workbench interface with the 'mov_genre' table being created. The SQL editor contains the following code:

```
1 create table mov_genre(  
2     genre_id int not null,  
3     genre_name varchar(45) not null,  
4     primary key(genre_id)  
5 );  
6 insert into mov_genre(genre_id,genre_name)  
7 values(300,'Drama,Family,sports');  
8 insert into mov_genre values(400,'Drama,History,sports');  
9 insert into mov_genre values(600,'Drama,Family');  
10 insert into mov_genre values(900,'Adventure,fantasy');  
11 insert into mov_genre values(1000,'Adventure,fantasy');  
12  
13 select * from mov_genre  
14  
15
```

The output window shows the execution of the SQL script, with 5 rows returned for the final select statement.

3.movie_director table

The screenshot shows the MySQL Workbench interface with the 'movie_director' table being created and populated. The SQL editor contains the following code:

```
1 create table movie_director(  
2     id int not null,  
3     director_fname varchar(45) not null,  
4     director_lname varchar(45) not null,  
5     primary key(id)  
6 );  
7 insert into movie_director(id,director_fname,director_lname)  
8     values(300,'shimit','amin');  
9 insert into movie_director values(500,'Rajkumar','Hirani');  
10 insert into movie_director values(900,'David','Yates');  
11 insert into movie_director values(700,'Brian De','Palma');  
12  
13 select * from movie_director;  
14  
15  
16  
17
```

The Output window shows the results of the queries:

#	Time	Action	Message	Duration / Fetch
1	11:05:30	SELECT * FROM movie LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
2	11:07:18	select * from movie_director LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

SQL script saved to: 'C:\Users\G Rithika\Downloads\ASSIGNMENT_5.sql'

4.movie_cast table

The screenshot shows the MySQL Workbench interface with the 'movie_cast' table being created and populated. The SQL editor contains the following code:

```
50  
51 create table movie_cast(  
52     mov_id int not null,  
53     act_id int not null,  
54     primary key(act_id)  
55 );  
56  
57 insert into movie_cast(mov_id,act_id)  
58     values(300,400);  
59 insert into movie_cast values(700,900);  
60 insert into movie_cast values(300,500);  
61 insert into movie_cast values(600,1000);  
62 insert into movie_cast values(900,300);  
63 insert into movie_cast values(1000,600);  
64
```

The Output window shows the results of the queries:

#	Time	Action	Message	Duration / Fetch
1	11:05:30	SELECT * FROM movie LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
2	11:07:18	select * from movie_director LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
3	11:07:53	select * from mov_genre LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SQL script saved to: 'C:\Users\G Rithika\Downloads\ASSIGNMENT_5.sql'

5.actor table

The screenshot shows the MySQL Workbench interface with the 'movies' schema selected. The SQL editor contains the following code:

```
create table actor(  
  act_id INT NOT NULL,  
  act_fname VARCHAR(255) NOT NULL,  
  actor_lname VARCHAR(255) NOT NULL,  
  actor_gender CHAR(10) NOT NULL,  
  actor_DOB DATE NOT NULL,  
  primary key(act_id)  
);  
  
insert into actor(act_id,act_fname,actor_lname,actor_gender,actor_DOB)  
values(300,'ranbir','kapoor','male','1998-07-11');  
insert into actor values(400,'rajkumar','rao','male','1997-02-17');  
insert into actor values(500,'anushka','sharma','female','2000-05-25');  
insert into actor values(600,'priyanka','chopra','female','1988-09-18');  
insert into actor values(900,'emma','watson','female','1991-05-11');  
insert into actor values(1000,'akshay','kumar','male','1974-12-14');
```

The Output window shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	11:05:30	SELECT * FROM movie LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
2	11:07:18	select * from movie_director LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
3	11:07:53	select * from mov_genre LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SQL script saved to 'C:\Users\G Rithika\Downloads\ASSIGNMENT_5.sql'

6.mov_rating table

The screenshot shows the MySQL Workbench interface with the 'movies' schema selected. The SQL editor contains the following code:

```
create table mov_rating(  
  mov_id INT NOT NULL,  
  num_o_stars INT NOT NULL,  
  primary key(mov_id)  
);  
  
insert into mov_rating(mov_id,num_o_stars)  
values(300,4);  
insert into mov_rating values(400,3);  
insert into mov_rating values(500,4);  
insert into mov_rating values(600,3);  
insert into mov_rating values(900,4);  
insert into mov_rating values(1000,4);  
insert into mov_rating values(700,5);
```

The Output window shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	11:05:30	SELECT * FROM movie LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
2	11:07:18	select * from movie_director LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
3	11:07:53	select * from mov_genre LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SQL script saved to 'C:\Users\G Rithika\Downloads\ASSIGNMENT_5.sql'

7.mov_award table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'movies' selected. The main editor window shows a SQL script for creating the 'mov_award' table and inserting data. The script is as follows:

```
103 create table mov_award(  
104     Award_id INT NOT NULL,  
105     mov_id INT NOT NULL,  
106     title VARCHAR(255) NOT NULL,  
107     primary key(Award_id)  
108 );  
109  
110  
111 insert into mov_award(Award_id,mov_id,title)  
112 values(1,300,'zee cine awards');  
113 insert into mov_award values(2,400,'iifa');  
114 insert into mov_award values(4,700,'iifa');  
115 insert into mov_award values(7,600,'guild award');  
116 insert into mov_award values(5,900,'people choice award');  
117 insert into mov_award values(6,700,'oscar');  
118
```

The bottom output window shows the execution results of the script:

#	Time	Action	Message	Duration / Fetch
1	11:05:30	SELECT * FROM movie LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
2	11:07:18	select * from movie_director LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
3	11:07:53	select * from mov_genre LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

B. INNER JOIN

Here the mov_id of both of the table, movie and mov_award are being matched and then the data that is movie title and title of the award is being displayed.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'movies' selected. The main editor window shows a SQL query for an INNER JOIN between the 'movie' and 'mov_award' tables. The query is as follows:

```
76 SELECT mov_award.title, movie.mov_title  
77 FROM movie  
78 INNER JOIN mov_award ON movie.mov_id = mov_award.mov_id WHERE mov_title='Mission Impossible 1';  
79  
80  
81
```

The bottom output window shows the execution results of the query:

#	Time	Action	Message	Duration / Fetch
33	16:42:15	select * from mov_award LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
34	17:16:28	SELECT * FROM mov_rating INNER JOIN movie ON mov_rating.column_name = movie.column_name LIMIT 0, 1000	Error Code: 1054. Unknown column 'mov_rating.column_name' in 'on clause'	0.000 sec / 0.000 sec
35	17:27:24	SELECT * FROM mov_rating INNER JOIN movie ON mov_rating.mov_id = movie.mov_id LIMIT 0, 1000	7 row(s) returned	0.047 sec / 0.000 sec
36	20:24:51	SELECT mov_award.title, movie.mov_title FROM movie INNER JOIN mov_award ON movie.mov_id = mov_a...	Error Code: 1054. Unknown column 'movie.mov_id' in 'on clause'	0.000 sec / 0.000 sec
37	20:25:46	SELECT mov_award.title, movie.mov_title FROM movie INNER JOIN mov_award ON movie.mov_id = mov_a...	6 row(s) returned	0.047 sec / 0.000 sec
38	20:28:23	SELECT mov_award.title, movie.mov_title FROM movie INNER JOIN mov_award ON movie.mov_id = mov_a...	2 row(s) returned	0.000 sec / 0.000 sec

Here mov_id and mov_genre_id are being compared belonging to the table movie and mov_genre, the matched data is being printed that is mov_genre_name and mov_title.

The screenshot shows MySQL Workbench with a query executed in the 'ASSIGNMENT_4' window. The query is as follows:

```

77 FROM movie
78 INNER JOIN mov_ward ON movie.mov_id = mov_ward.mov_id WHERE mov_title='Mission Impossible 1';
79
80 • SELECT mov_genre.genre_name, movie.mov_title
81 FROM movie
82 INNER JOIN mov_genre ON movie.mov_id = mov_genre.genre_id ORDER BY mov_lang ;

```

The 'Result Grid' displays the following data:

genre_name	mov_title
Adventure,fantasy	Harry potter and the prisoner of askaban
Adventure,fantasy	Harry potter and the deathly hallows 1
Drama,Family,sports	chakde
Drama,History,sports	gold
Drama,Family	taare zameen par

The 'Output' pane shows the execution log with the following actions:

#	Time	Action	Message	Duration / Fetch
41	14:07:08	SELECT mov_genre.genre_name, movie.mov_title FROM movie INNER JOIN mov_genre ON movie.mov_id =	Error Code: 1054. Unknown column 'mov_genre.gen_id' in 'on clause'	0.000 sec
42	14:07:27	USE movie_database	0 row(s) affected	0.000 sec
43	14:07:32	SELECT mov_genre.genre_name, movie.mov_title FROM movie INNER JOIN mov_genre ON movie.mov_id =	Error Code: 1054. Unknown column 'mov_genre.gen_id' in 'on clause'	0.000 sec
44	14:07:45	SELECT mov_ward.title, movie.mov_title FROM movie INNER JOIN mov_ward ON movie.mov_id = mov_a	2 row(s) returned	0.000 sec / 0.000 sec
45	14:08:11	SELECT mov_genre.genre_name, movie.mov_title FROM movie INNER JOIN mov_genre ON movie.mov_id =	5 row(s) returned	0.062 sec / 0.000 sec
46	14:08:34	SELECT mov_genre.genre_name, movie.mov_title FROM movie INNER JOIN mov_genre ON movie.mov_id =	5 row(s) returned	0.031 sec / 0.000 sec

Here 3 tables are being used which are mov_genre, movie and mov_rating and mov_genre_id and mov_id is compared on the other hand mov_genre_id and mov_rating is and the matched data is printed.

The screenshot shows MySQL Workbench with a query executed in the 'ASSIGNMENT_5' window. The query is as follows:

```

127
128
129 • select mov_genre.genre_name,movie.mov_title,movie.mov_lang,mov_rating.num_o_stars
130 from mov_genre
131 inner join movie on mov_genre.genre_id = movie.mov_id
132 inner join mov_rating on mov_genre.genre_id = mov_rating.mov_id;
133
134
135
136

```

The 'Result Grid' displays the following data:

genre_name	mov_title	mov_lang	num_o_stars
Drama,Family,sports	chakde	hindi	4
Drama,History,sports	gold	hindi	3
Drama,Family	taare zameen par	hindi	3
Adventure,fantasy	Harry potter and the prisoner of askaban	english	4
Adventure,fantasy	Harry potter and the deathly hallows 1	english	4

The 'Output' pane shows the execution log with the following actions:

#	Time	Action	Message	Duration / Fetch
1	09:12:51	SELECT mov_ward.title, movie.mov_title FROM movie INNER JOIN mov_ward ON movie.m...	2 row(s) returned	0.016 sec / 0.000 sec
2	09:14:48	SELECT mov_genre.genre_name, movie.mov_title FROM movie INNER JOIN mov_genre O...	5 row(s) returned	0.034 sec / 0.000 sec
3	09:17:05	select mov_genre.genre_name,movie.mov_title,movie.mov_lang,mov_rating.num_o_stars from	5 row(s) returned	0.063 sec / 0.000 sec

C.LEFT JOIN

Here 3 tables are being used movie, mov_director and mov_rating, mov_id and id of table movie and movie_director is being compared on the other hand mov_id of movie and mov_rating table are matched and the data selected are being printed.

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

```
141 SELECT movie.mov_title, movie.mov_rel_date_year, movie.mov_lang, movie_director.director_fname, movie_director.dir_
142 FROM movie
143 LEFT JOIN movie_director ON movie.mov_id = movie_director.id
144 LEFT JOIN mov_rating ON movie.mov_id = mov_rating.mov_id;
145
146
```

The result grid displays the following data:

mov_title	mov_rel_date_year	mov_lang	director_fname	director_lname	num_o_stars
chakde	2017-06-15	hindi	shimit	amin	4
gold	2016-06-18	hindi	MOON	MOON	3
3 idiots	2008-07-14	hindi	Rajkumar	Hirani	4
taare zameen par	2008-07-10	hindi	MOON	MOON	3
Mission Impossible 1	2012-06-14	english	Brian De	Palma	5
Harry potter and the prisoner of askaban	2008-05-11	english	David	Yates	4
Harry potter and the deathly hallows 1	2011-05-21	english	MOON	MOON	4

The bottom status bar indicates "Query Completed".

Here mov_id of table movie and mov_award with WHERE clause and the selected data is being printed.

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

```
88 ON movie.mov_id = movie_director.id;
89
90 SELECT movie.mov_title, mov_award.title
91 FROM movie
92 LEFT JOIN mov_award ON movie.mov_id = mov_award.mov_id
93 WHERE movie.mov_rel_date_year < '2011-12-12';
```

The result grid displays the following data:

mov_title	title
3 idiots	MOON
taare zameen par	guld award
Harry potter and the prisoner of askaban	people choice award
Harry potter and the deathly hallows 1	MOON

The bottom status bar indicates "Query Completed".

Here we are comparing mov_id of tables movie and mov_rating with ORDER BY clause and the selected data is being printed.

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

```
146
147 • SELECT movie.mov_title,movie.mov_lang,mov_rating.num_o_stars
148 FROM movie
149 LEFT JOIN mov_rating ON movie.mov_id = mov_rating.mov_id ORDER BY num_o_stars;
150
151
```

The Result Grid displays the following data:

mov_title	mov_lang	num_o_stars
gold	hindi	3
taare zameen par	hindi	3
chakde	hindi	4
3 idiots	hindi	4
Harry potter and the prisoner of askaban	english	4
Harry potter and the deathly hallows 1	english	4
Mission Impossible 1	english	5

The Output pane shows the action output: "1 10:52:49 SELECT movie.mov_title,movie.mov_lang,mov_rating.num_o_stars FROM movie LEFT JOIN ... 7 row(s) returned".

D.RIGHT JOIN

Here selected data is being printed with the usage of GROUP BY clause and sum() function, and mov_id and act_id are being matched from the table movie and mov_cast.

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

```
4
5
6 • select t.mov_title,t.mov_lang,sum(ti.act_id) as 'total views',sum(ti.mov_id) as 'total likes'
7 from movie as t
8 right join movie_cast as ti
9 on t.mov_id = ti.act_id
10 group by t.mov_title;
11
12
```

The Result Grid displays the following data:

mov_title	mov_lang	total views	total likes
chakde	hindi	300	900
gold	hindi	400	300
3 idiots	hindi	500	300
taare zameen par	hindi	600	1000
Harry potter and the prisoner of askaban	english	900	700
Harry potter and the deathly hallows 1	english	1000	600

The Output pane shows the action output: "1 19:53:31 select t.mov_title,t.mov_lang,sum(ti.act_id) as total views,sum(ti.mov_id) as total likes' from mo... 6 row(s) returned".

Here selected data is being printed by comparing mov_id of mov_award and movie table with usage of ORDER BY clause.

The screenshot shows the MySQL Workbench interface with a query executed in the SQL Editor. The query is a right join between the movie and mov_award tables, ordered by movie_title. The result grid displays the following data:

mov_title	mov_country	mov_lang	Award_id	title
chakde	india	hindi	1	zee cine awards
gold	india	hindi	2	ifa
Harry potter and the prisoner of askaban	USA	english	5	people choice award
Mission Impossible 1	USA	english	4	ifa
Mission Impossible 1	USA	english	6	oscar
taare zameen par	india	hindi	7	guild award

The Output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
5	19:14:39	select movie_mov_title,movie_mov_country,movie_mov_lang,mov_award Award_id,mov_awa...	6 row(s) returned	0.094 sec / 0.000 sec
6	19:15:57	select movie_mov_title,movie_mov_country,movie_mov_lang,mov_award Award_id,mov_awa...	6 row(s) returned	0.079 sec / 0.000 sec
7	19:17:54	select movie_mov_title,movie_mov_country,movie_mov_lang,mov_award Award_id,mov_awa...	6 row(s) returned	0.000 sec / 0.000 sec
8	19:19:07	select movie_mov_title,movie_mov_country,movie_mov_lang,mov_award Award_id,mov_awa...	6 row(s) returned	0.000 sec / 0.000 sec

Here 3 tables are being used which are mov_genre, movie and mov_rating and the data is matched and the selected ones are being printed accordingly.

The screenshot shows the MySQL Workbench interface with a query executed in the SQL Editor. The query is a right join between the mov_genre, movie, and mov_rating tables. The result grid displays the following data:

genre_name	mov_title	mov_lang	num_o_stars
Drama,Family,sports	chakde	hindi	4
Drama,History,sports	gold	hindi	3
None	None	None	4
Drama,Family	taare zameen par	hindi	3
None	None	None	5
Adventure,fantasy	Harry potter and the prisoner of askaban	english	4
Adventure,fantasy	Harry potter and the deathly hallows 1	english	4

The Output pane shows the execution log with the following entry:

#	Time	Action	Message	Duration / Fetch
1	18:56:16	select mov_genre.genre_name,movie_mov_title,movie_mov_lang,mov_rating.num_o_stars from ...	7 row(s) returned	0.390 sec / 0.000 sec