

# ASSIGNMENT NO. 1

Q.1) List of the actors whose last name is "Bear"

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'firstprog' selected. Under 'firstprog', the 'actors' table is highlighted. The 'Table: actors' information panel shows columns: id (int PK), first\_name (varchar(100)), last\_name (varchar(100)), gender (char(1)), and film\_count (int). The main query editor contains the following SQL:

```
1 use firstprog;
2 SELECT last_name FROM actors where last_name = 'Bear';
```

The 'Result Grid' at the bottom shows one result row:

last_name
Bear

The status bar at the bottom indicates 'Query Completed' and 'actors 4 x'.

Q.2) List of movie names whose rank is between 7 and 8

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'firstprog' selected. Under 'firstprog', the 'movies' table is highlighted. The 'Table: movies' information panel shows columns: id (int PK), name (varchar(100)), year (int), and rank (float). The main query editor contains the following SQL:

```
1 use firstprog;
2 SELECT movies.name, movies.rank
3 FROM movies
4 WHERE movies.rank BETWEEN 7 and 8;
```

The 'Result Grid' at the bottom shows the following results:

name	rank
Animal House	7.5
Apollo 13	7.5
Few Good Men, A	7.5
JFK	7.8
Little Mermaid, The	7.3
Lost in Translation	8
O Brother, Where Art Thou?	7.8
Ocean's Eleven	7.5
Office Space	7.6
Pi	7.5
Planes, Trains & Automobiles	7.2
Snatch.	7.9
Str of Echoes	7

The status bar at the bottom indicates 'Query Completed' and 'movies 4 x'.

Q.3) List of the movies that have contains the letter 'a' in any position

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'movies' database selected. The 'Table: movies' information is shown, including columns: id (int PK), name (varchar(100)), year (int), and rank (float). The main query editor contains the following SQL code:

```
1 use firstprog;  
2 SELECT movies.name  
3 FROM movies  
4 WHERE movies.name LIKE '%a%';
```

The 'Result Grid' at the bottom shows the results of the query, listing movie names that contain the letter 'a':

name
Aliens
Animal House
Apollo 13
Batman Begins
Braveheart
Fargo
Few Good Men, A
Garden State
Godfather, The
Hollow Man
Little Mermaid, The
Lost in Translation
Matrix, The
O Brother, Wher...
Ocean's Eleven
Office Space

The status bar at the bottom indicates 'Query Completed'.

This screenshot is identical to the one above, showing the same MySQL Workbench interface and query results. The 'Result Grid' lists the following movie names:

name
Garden State
Godfather, The
Hollow Man
Little Mermaid, The
Lost in Translation
Matrix, The
O Brother, Wher...
Ocean's Eleven
Office Space
Prates of the Ca...
Planes, Trains & ...
Shawshank Rede...
Snatch
Star Wars
Titanic
Vanilla Sky

The status bar at the bottom indicates 'Query Completed'.

Q.4) List of the directors whose movies are ranked with greater than 4 ranking.

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

```
1 use firstprog;
2 SELECT directors.first_name, movies.name, movies.rank
3 FROM directors
4 JOIN movies_directors
5 ON directors.id = movies_directors.director_id
6 JOIN movies
7 ON movies_directors.movie_id=movies.id
8 WHERE movies.rank > 4
```

The result grid displays the following data:

first_name	name	rank
James (1)	Aliens	8.2
John (1)	Animal House	7.5
Ron	Apollo 13	7.5
Mel (1)	Braveheart	8.3
Ethan	Fargo	8.2
Joel	Fargo	8.2
Rob	Few Good Men, A	7.5
David	Fight Club	8.5
Herbert (1)	Footloose	5.8
Zach	Garden State	8.3
Francis Ford	Godfather, The	9
Paul (1)	Hollow Man	5.3
Oliver (1)	JFK	7.8

The left sidebar shows the database schema with tables: actors, directors, directors\_genres, movies, movies\_directors, and movies. The 'movies\_directors' table is selected, showing columns: director\_id and movie\_id.

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

```
1 use firstprog;
2 SELECT directors.first_name, movies.name, movies.rank
3 FROM directors
4 JOIN movies_directors
5 ON directors.id = movies_directors.director_id
6 JOIN movies
7 ON movies_directors.movie_id=movies.id
8 WHERE movies.rank > 4
```

The result grid displays the following data:

first_name	name	rank
Paul (1)	Hollow Man	5.3
Oliver (1)	JFK	7.8
Quentin	Kill Bill: Vol. 1	8.4
Quentin	Kill Bill: Vol. 2	8.2
Ron	Little Mermaid, The	7.3
John	Little Mermaid, The	7.3
Sofia	Lost in Translation	8
Andy	Matrix, The	8.5
Larry	Matrix, The	8.5
Christopher	Memento	8.7
Clint	Mystic River	8.1
Ethan	O Brother, Wher...	7.8
Joel	O Brother, Wher...	7.8

The left sidebar shows the database schema with tables: actors, directors, directors\_genres, movies, movies\_directors, and movies. The 'movies\_directors' table is selected, showing columns: director\_id and movie\_id.

MySQL Workbench

Local instance MySQL80 x MySQL Model\* x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

firstprog

Tables

actors

directors

directors\_genres

movies

Columns

id

name

year

rank

Indexes

Foreign Keys

Triggers

movies\_directors

Columns

director\_id

movie\_id

Administration

Schemas

Information

Table: directors

Columns:

id int PK

first\_name varchar(100)

last\_name varchar(100)

Object Info Session

Result 7 x

```
1 use firstprog;
2 SELECT directors.first_name, movies.name, movies.rank
3 FROM directors
4 JOIN movies_directors
5 ON directors.id = movies_directors.director_id
6 JOIN movies
7 ON movies_directors.movie_id=movies.id
8 WHERE movies.rank > 4
```

Result Grid

first_name	name	rank
Ethan	O Brother, Wher...	7.8
Joel	O Brother, Wher...	7.8
Steven	Ocean's Eleven	7.5
Mike (1)	Office Space	7.6
Darren	Pi	7.5
John (1)	Planes, Trains & ...	7.2
Quentin	Pulp Fiction	8.7
Quentin	Reservoir Dogs	8.3
Frank	Shawshank Rede...	9
Andrew	Shrek	8.1
Vicky	Shrek	8.1
Guy	Snatch.	7.9
George	Star Wars	8.8

Query Completed

Type here to search

10°C

4:07 PM 5/13/2022

MySQL Workbench

Local instance MySQL80 x MySQL Model\* x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

firstprog

Tables

actors

directors

directors\_genres

movies

Columns

id

name

year

rank

Indexes

Foreign Keys

Triggers

movies\_directors

Columns

director\_id

movie\_id

Administration

Schemas

Information

Table: directors

Columns:

id int PK

first\_name varchar(100)

last\_name varchar(100)

Object Info Session

Result 7 x

```
1 use firstprog;
2 SELECT directors.first_name, movies.name, movies.rank
3 FROM directors
4 JOIN movies_directors
5 ON directors.id = movies_directors.director_id
6 JOIN movies
7 ON movies_directors.movie_id=movies.id
8 WHERE movies.rank > 4
```

Result Grid

first_name	name	rank
Darren	Pi	7.5
John (1)	Planes, Trains & ...	7.2
Quentin	Pulp Fiction	8.7
Quentin	Reservoir Dogs	8.3
Frank	Shawshank Rede...	9
Andrew	Shrek	8.1
Vicky	Shrek	8.1
Guy	Snatch.	7.9
George	Star Wars	8.8
David	Str of Echoes	7
James (1)	Titanic	6.9
Jay	UHF	6.6
Cameron	Vanilla Sky	6.9

Query Completed

Type here to search

10°C

4:08 PM 5/13/2022

Q.5) List of "Adventure" genre movies acted by the actors with the lastname – Bear

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a search filter. The 'movies' table is selected, showing its columns: id (int PK), name (varchar(100)), year (int), and rank (float). The main query editor contains the following SQL code:

```
1 use firstprog;
2 SELECT movies_genres.genre, actors.first_name, actors.last_name, movies.name
3 FROM actors
4 JOIN roles
5 ON actors.id = roles.actor_id
6 JOIN movies
7 ON roles.movie_id = movies.id
8 JOIN movies_genres
9 ON movies.id = movies_genres.movie_id
10 WHERE movies_genres.genre = 'Adventure' and actors.last_name = 'Bear'
11
12
13
```

The 'Result Grid' at the bottom shows the following data:

genre	first_name	last_name	name
Adventure	Lightrning	Bear	Star Wars

The status bar at the bottom indicates 'Query Completed' and 'Read Only'.