DATABASE CREATION –

```
mysql> create database problemset03;
Query OK, 1 row affected (0.00 sec)

mysql> use problemset03;
Database changed
```

TABLE CREATION –

```
mysql> CREATE TABLE Movie(mID INT PRIMARY KEY,title VARCHAR(30),year
INT,director VARCHAR(20));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Reviewer(rID INT PRIMARY KEY,name VARCHAR(20));
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Rating(rID INT,mID INT,stars INT,ratingDate
DATE,FOREIGN KEY(rID) REFERENCES Reviewer(rID), FOREIGN KEY(mID)
REFERENCES Movie(mID));
Query OK, 0 rows affected (0.02 sec)
```

VALUE INSERTION –

```
mysql> INSERT INTO Movie(mID,title,year,director) VALUES (101,'Gone
with the Wind',1939,'Victor Fleming');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Movie(mID,title,year,director) VALUES (102,'Star
Wars',1977,'George Lucas');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Movie(mID,title,year,director) VALUES (103,'The
sound of music',1965,'Robert Wise');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Movie(mID,title,year,director) VALUES
(104,'E.T.',1982,'Steven Spielberg');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Movie(mID,title,year,director) VALUES
(105,'Titanic',1997,'James Cameron');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Movie(mID,title,year) VALUES (106,'Snow White',
1937);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Movie(mID,title,year,director) VALUES
(107,'Avatar',2009,'James Cameron');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Movie(mID,title,year,director) VALUES
(108,'Raiders of the Lost Ark',1981,'Steven Spielberg');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Movie;
+-----+-------------------------+------+------------------+
| mID | title                   | year | director         |
+-----+-------------------------+------+------------------+
| 101 | Gone with the Wind      | 1939 | Victor Fleming   |
| 102 | Star Wars               | 1977 | George Lucas     |
| 103 | The sound of music      | 1965 | Robert Wise      |
| 104 | E.T.                    | 1982 | Steven Spielberg |
| 105 | Titanic                 | 1997 | James Cameron    |
| 106 | Snow White              | 1937 | NULL             |
| 107 | Avatar                  | 2009 | James Cameron    |
| 108 | Raiders of the Lost Ark | 1981 | Steven Spielberg |
+-----+-------------------------+------+------------------+
8 rows in set (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (201,'Sarah Martinez');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (202,'Daniel Lewis');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (203,'Brittany
Harris');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (204,'Mike Anderson');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (205,'Chris Jackson');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (206,'Elizabeth
Thomas');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (207,'James Cameron');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Reviewer(rID,name) VALUES (208,'Ashley White');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Reviewer;
+-----+------------------+
| rID | name             |
+-----+------------------+
| 201 | Sarah Martinez   |
| 202 | Daniel Lewis     |
| 203 | Brittany Harris  |
| 204 | Mike Anderson    |
| 205 | Chris Jackson    |
| 206 | Elizabeth Thomas |
| 207 | James Cameron    |
| 208 | Ashley White     |
+-----+------------------+
8 rows in set (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(201,101,2,'20110122');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(201,101,4,'20110127');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Rating(rId,mId,stars) VALUES (202,106,4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(203,103,2,'20110120');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(203,108,4,'20110112');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(203,108,2,'20110130');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(204,101,3,'20110109');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(205,103,3,'20110127');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(205,104,2,'20110122');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Rating(rId,mId,stars) VALUES (205,108,4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(206,107,3,'20110115');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(206,106,5,'20110119');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
```

```
(207,107,5,'20110120');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rating(rId,mId,stars,ratingDate) VALUES
(208,104,3,'20110102');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Rating;
+------+------+-------+------------+
| rID  | mID  | stars | ratingDate |
+------+------+-------+------------+
|  201 |  101 |     2 | 2011-01-22 |
|  201 |  101 |     4 | 2011-01-27 |
|  202 |  106 |     4 | NULL       |
|  203 |  103 |     2 | 2011-01-20 |
|  203 |  108 |     4 | 2011-01-12 |
|  203 |  108 |     2 | 2011-01-30 |
|  204 |  101 |     3 | 2011-01-09 |
|  205 |  103 |     3 | 2011-01-27 |
|  205 |  104 |     2 | 2011-01-22 |
|  205 |  108 |     4 | NULL       |
|  206 |  107 |     3 | 2011-01-15 |
|  206 |  106 |     5 | 2011-01-19 |
|  207 |  107 |     5 | 2011-01-20 |
|  208 |  104 |     3 | 2011-01-02 |
+------+------+-------+------------+
14 rows in set (0.00 sec)
```

QUERIES -

1. Find the titles of all movies directed by Steven Spielberg.

```
mysql> select title from movie where director='Steven Spielberg';
+-------------------------+
| title                   |
+-------------------------+
| E.T.                    |
| Raiders of the Lost Ark |
+-------------------------+
2 rows in set (0.00 sec)
```

2. Find all years that have a movie that received a rating of 4 or 5, and sort them in increasing order .

```
mysql> select year, title from movie join Rating on
movie.mID=Rating.mID where Rating.stars=4 or Rating.stars=5 order by
Rating.stars;
+------+-------------------------+
| year | title                   |
+------+-------------------------+
| 1939 | Gone with the Wind      |
| 1937 | Snow White              |
| 1981 | Raiders of the Lost Ark |
| 1981 | Raiders of the Lost Ark |
| 1937 | Snow White              |
| 2009 | Avatar                  |
+------+-------------------------+
6 rows in set (0.00 sec)
```

3. Find the titles of all movies that have no ratings.

```
mysql> select title from movie where mID not in ( select mId from
Rating) ;     +-----------+
| title     |
+-----------+
| Star Wars |
| Titanic   |
+-----------+
2 rows in set (0.00 sec)
```

4. Some reviewers didn’t provide a date with their rating. Find the names of all reviewers who have ratings with a null value for the date.

```
mysql> select Reviewer.name from Reviewer join Rating on
Reviewer.rID=Rating.rID where Rating.RatingDate is Null;
+---------------+
| name          |
+---------------+
| Daniel Lewis  |
| Chris Jackson |
+---------------+
2 rows in set (0.00 sec)
```

5. Write a query to return the ratings data in a more readable format : reviewer name , movie title , stars and ratingDate. Also, sort the data , first by reviewer name, then by movie title ,and lastly by number of stars.

```
mysql> select name as Reviewers_Name,title as movie_title,stars as
No_of_Stars, ratingDate as DateOfRating from movie join Rating on
movie.mID=Rating.mID join  Reviewer on Rating.rID=Reviewer.rID order
by name,title,stars;
+------------------+-------------------------+-------------
+--------------+
| Reviewers_Name   | movie_title             | No_of_Stars |
DateOfRating |
+------------------+-------------------------+-------------
+--------------+
| Ashley White     | E.T.                    |           3 |
2011-01-02   |
| Brittany Harris  | Raiders of the Lost Ark |           2 |
2011-01-30   |
| Brittany Harris  | Raiders of the Lost Ark |           4 |
2011-01-12   |
| Brittany Harris  | The sound of music      |           2 |
2011-01-20   |
| Chris Jackson    | E.T.                    |           2 |
2011-01-22   |
| Chris Jackson    | Raiders of the Lost Ark |           4 | NULL
|
| Chris Jackson    | The sound of music      |           3 |
2011-01-27   |
| Daniel Lewis     | Snow White              |           4 | NULL
|
| Elizabeth Thomas | Avatar                  |           3 |
2011-01-15   |
| Elizabeth Thomas | Snow White              |           5 |
2011-01-19   |
| James Cameron    | Avatar                  |           5 |
2011-01-20   |
| Mike Anderson    | Gone with the Wind      |           3 |
2011-01-09   |
| Sarah Martinez   | Gone with the Wind      |           2 |
2011-01-22   |
| Sarah Martinez   | Gone with the Wind      |           4 |
2011-01-27   |
+------------------+-------------------------+-------------
+--------------+
14 rows in set (0.00 sec)
```

6.For all cases where the same reviewer rated the same movie twice and gave it a higher rating the second time , return the reviewer’s name and title of the movie.

7. For each movie that has at least one rating, find the highest number of stars that movie received.Return the movie title and number of stars. Sort by movie title.

```
mysql> select title,max(stars) from movie join Rating on
movie.mID=Rating.mID group by movie.mID order by movie.title;
+-------------------------+------------+
| title                   | max(stars) |
+-------------------------+------------+
| Avatar                  |          5 |
| E.T.                    |          3 |
| Gone with the Wind      |          4 |
| Raiders of the Lost Ark |          4 |
| Snow White              |          5 |
| The sound of music      |          3 |
+-------------------------+------------+
6 rows in set (0.00 sec)
```

8. For each movie, return the title and the ‘rating spread’, that is the difference between highest and lowest ratings given to that movie. Sort by rating spread from highest to lowest , then by movie title.

```
mysql> select title , max(stars)-min(stars) as RatingSpread from
movie join      Rating on movie.mID=Rating.mID group by movie.mID
```

```
order by RatingSpread desc, title ;
+-------------------------+--------------+
| title                   | RatingSpread |
+-------------------------+--------------+
| Avatar                  |            2 |
| Gone with the Wind      |            2 |
| Raiders of the Lost Ark |            2 |
| E.T.                    |            1 |
| Snow White              |            1 |
| The sound of music      |            1 |
+-------------------------+--------------+
6 rows in set (0.00 sec)
```

9.Find the difference between the average rating of movies released before 1980 and the average rating of movies released after 1980 . (Make sure to calculate the average rating for each movie , then the average of those averages for movies before 1980 and movies after) .

```
mysql> Create view AvgRatingMovie as select mID, avg(stars) as
stars1 from Rating group by mID;
Query OK, 0 rows affected (0.01 sec)

mysql> create view AvgBefore1980 as select avg(stars) as stars from
AvgRatingMovie , movie where AvgRatingMovie.mID=movie.mID and year <
1980;
Query OK, 0 rows affected (0.00 sec)

mysql> create view AvgAfter1980 as select avg(stars) as stars from
AvgRatingMovie , movie where AvgRatingMovie.mID=movie.mID and year >
1980;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from avgRating;
+------------+
| stars      |
+------------+
| 3.33333333 |
| 3.27776667 |
+------------+
2 rows in set (0.00 sec)

mysql> select avg(stars) from avgRating;
+------------+
| avg(stars) |
+------------+
|    3.30555 |
+------------+
1 row in set (0.00 sec)
```

10. Find the name of all reviewers who rated Gone with the Wind.

```
mysql> select name from Reviewer where rID in ( select rID from
Rating where
```

```
    -> mID in ( select mID from movie where title='Gone with the
Wind'));
+-----------------+
| name            |
+-----------------+
| Sarah Martinez |
| Mike Anderson  |
+-----------------+
2 rows in set (0.00 sec)
```

11. For any rating where the reviewer is the same as the director of the movie , return the reviewer name , movie title , and number of stars.

```
mysql> select Reviewer.name, movie.title , Rating.stars from
Reviewer join Rating on Reviewer.rID=Rating.rID join movie on
Rating.mID=movie.mID where Reviewer.name=movie.director;
+---------------+--------+-------+
| name          | title  | stars |
+---------------+--------+-------+
| James Cameron | Avatar |     5 |
+---------------+--------+-------+
1 row in set (0.00 sec)
```

12. Return all reviewer names and movie names together in a single list , alphabetised.

```
mysql> Select concat(name,'-',title) as Names from movie join Rating
on movie.mID=Rating.mID join Reviewer on Rating.rID=Reviewer.rID
order by name;
+------------------------------------------+
| Names                                    |
+------------------------------------------+
| Ashley White-E.T.                        |
| Brittany Harris-The sound of music       |
| Brittany Harris-Raiders of the Lost Ark |
| Brittany Harris-Raiders of the Lost Ark |
| Chris Jackson-The sound of music         |
| Chris Jackson-E.T.                       |
| Chris Jackson-Raiders of the Lost Ark    |
| Daniel Lewis-Snow White                  |
| Elizabeth Thomas-Avatar                  |
| Elizabeth Thomas-Snow White              |
| James Cameron-Avatar                     |
| Mike Anderson-Gone with the Wind         |
| Sarah Martinez-Gone with the Wind        |
| Sarah Martinez-Gone with the Wind        |
+------------------------------------------+
14 rows in set (0.01 sec)
```

13. Find the titles of all movies not reviewed by Chris Jackson .

```
mysql> select title from movie where mID in ( select mID from Rating
where rID in ( select rID from Reviewer where name='Chris
Jackson'));
+-------------------------+
| title                   |
+-------------------------+
| The sound of music      |
| E.T.                    |
| Raiders of the Lost Ark |
+-------------------------+
3 rows in set (0.00 sec)
```