DBMS Lab 2019-20 Spring Semester
Lab Day 4 (January 28, 2020) – 30 Marks
EVEN NUMBERED PCS

[Penalty for plagiarism/copying: You will be awarded 0 for all the problems for the lab day you were involved in plagiarism/copying and an additional 5 marks will be deducted out of the total of 40 in Lab. All persons involved will be awarded the same penalty irrespective of who has copied from whom. Decision of the lab teachers is final in this respect.]

Please note that you have to use exactly the same schema as given below as we will execute the SQLs on these schemas. Note that, you need not create the primary keys or foreign keys while creating the tables. They are given for your understanding of the schemas. Also, we finally need the SQLs. Hence, it is upto you whether you want to create the tables. But it is advisable that you create the tables and test your SQLs on sample data. Marking will be strict.

Consider the following relational schemas
Student (RollNo int Not Null, Name varchar(20) Not Null, YearOfAdmission int Not Null,
PRIMARY KEY(RollNo))

Friend(OwnRoll int Not Null, FriendRoll int Not Null,
PRIMARY KEY(OwnRoll, FriendRoll),
FOREIGN KEY fk_std1(OwnRoll) REFERENCES Student(RollNo),
FOREIGN KEY fk_std2(FriendRoll) REFERENCES Student(RollNo))

Movie(MID int Not Null, Title varchar(30) Not Null, YearOfRelease int Not Null, DirectorName varchar(20) Null,
PRIMARY KEY(MID)) [Assume all director names are unique. However, same director can direct many movies]

Rating(RollNo int Not Null, MID int Not Null, RatingDate date Not Null, Rating int Not Null,
PRIMARY KEY(RollNo, MID, RatingDate),
FOREIGN KEY fk_std4(RollNo) REFERENCES Student(RollNo),
FOREIGN KEY fk_mov2(MID) REFERENCES Movie(MID));

Write a single SQL for each of the following queries (3x10=30)

1. List the Rollno, Name, YearOfAdmission of all students who do not have even one friend having Amit as part of his name. (Output format: RollNo, Name, YearOfAdmission)
2. List the titles of all movies that have received ratings from at least one student. (Output format: Title)
3. List the director names with their corresponding lowest rated movie(s) (Output format: DirectorName, Title, LowestTotalRating). Note, total rating of a movie is the sum of all the ratings it has received.
4. List the pair of students with their names who have rated the same movie(s) at least once with the same rating (Output format: RollNo1, Name1, RollNo2, Name2, Title, Rating). Note, it is OK if the same pair of students shows up twice in the result set with their RollNo and YearOfAdmission reversed.
5. Determine is the average number of friends per student. (Output format: AvgNo). Note, your result should be just one row and one column which is the average of the number of friends each student has.

6. List the movies which have been given ratings by at least 4 students whose Year of Admission is after the Year of Release of the movie. (Output format: Title, YearOfRelease, Rollno, YearOfAdmission). Note, values for some of the columns may repeat for different rows.
7. List the students whose average rating over all movies (including multiple instances of rating the movies on different dates) is less than the average rating of those movies by his/her friends (including multiple instances of rating the movies on different dates). (Output format: RollNo1, AverageRating1, RollNo2, AverageRating2)
8. List the 4^{th} most popular movies(s). (Output format: Title, NumberOfStudentRatings). Note, upto the 3^{rd} most popular movie, all movies having the same popularity will have to be counted only once. For the 4^{th} position, all movies having that popularity are to be listed.
9. List the students who have not rated any movie that have been rated by his/her friend. (Output Format: RollNo1, Name1, RollNo2, Name2)
10. List the top 4 most popular movies. (Output format: Title, NumberOfStudentRatings). Note, popularity is determined by the number of ratings received a movie, which is the number of distinct students who have rated the movie. Also, here the number 4 denotes 4 distinct values of popularity with possibly multiple movies having the same popularity.

BONUS: List the students whose difference of ratings for the same movie on consecutive dates is greater than 1. (Output Format: RollNo, Name, Title, RatingDate1, Rating1, RatingDate2, Rating2). Note, if for the same movie, this happens more than once, you may either report that once or as many times as they occur. Also, for the same student, there might be multiple such movies.

[5 Marks]

Through Moodle, submit a text file containing all the Select SQL statements against the link for Even Numbered PCs. (Name it as Lab4_<Roll_no>.txt). On top of the text file, write your Roll Number, Name and PC Number. The PC Number should match with the PC Number in the attendance sheet.