DBMS Lab 2017-18 Spring Semester

Lab Day 4 (January 30, 2018) – ODD NUMBERED PCS 50 Marks

Consider following database schemas

Student(RollNo int, name varchar(20), class int,
PRIMARY KEY(RollNo));

Friend(OwnRoll int, FriendRoll int,
PRIMARY KEY(OwnRoll,FriendRoll),
FOREIGN KEY fk_std1(OwnRoll) REFERENCES Student(RollNo),
FOREIGN KEY fk_std2(FriendRoll) REFERENCES Student(RollNo));

Likes(OwnRoll int, FriendRoll int,
PRIMARY KEY(OwnRoll,FriendRoll),
FOREIGN KEY fk_std1(OwnRoll) REFERENCES Student(RollNo),
FOREIGN KEY fk_std2(FriendRoll) REFERENCES Student(RollNo));

Movie(mID int, title varchar(30), year int, director varchar(20),
PRIMARY KEY(mID));

Rating(RollNo int, mID int, stars int, ratingDate date,PRIMARY KEY(RollNo,mID,stars),
FOREIGN KEY fk_std(RollNo) REFERENCES Student(RollNo),
FOREIGN KEY fk_mv(mID) REFERENCES Movie(mID));

Solve the following queries (5x10=50)

1. Find the names of all students who are friends with someone named Sachin. (Assume that there is only one student named Sachin)
2. Find the titles of all movies that have no ratings.
3. List the director name with their corresponding highest rated movie
4. List the pair of students with their class who rated same movie with same rating
5. What is the average number of friends per student? (Your result should be just one number.)
6. Find the difference between the number of students in the school and the number of different first names.
7. Find the list of movies and Directors rated more than 2 by pair of students who are in same class and likes each other
8. For every student who likes someone 2 or more class younger than themselves, return that student's name and class, and the name and class of the student they like.
9. List all students (Roll No, Name) each of whom has at least one friend who does not like him/her.
10. For all pairs of reviewers such that both reviewers gave a rating to the same movie, return the names of both reviewers. Eliminate duplicates, don't pair reviewers with themselves, and include each pair only once. For each pair, return the names in the pair in alphabetical order.

[Penalty for plagiarism/copying: You will be awarded 0 for all the problems for the lab day 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]

```
/* Delete the tables if they already exist */
drop table if exists Rating;
drop table if exists Movie;
drop table if exists Likes;
drop table if exists Friend;
drop table if exists Student;



/* Create the schema for our tables */
create table Student(RollNo int PRIMARY KEY, name varchar(20), class int);
create table Friend(OwnRoll int, FriendRoll int, FOREIGN KEY fk_std1(OwnRoll)
REFERENCES Student(RollNo), FOREIGN KEY fk_std2(FriendRoll) REFERENCES
Student(RollNo),PRIMARY KEY(OwnRoll,FriendRoll));
create table Likes(OwnRoll int, FriendRoll int, FOREIGN KEY fk_std1(OwnRoll)
REFERENCES Student(RollNo), FOREIGN KEY fk_std2(FriendRoll) REFERENCES
Student(RollNo),PRIMARY KEY(OwnRoll,FriendRoll));
create table Movie(mID int PRIMARY KEY, title varchar(30), year int, director
varchar(20));
create table Rating(RollNo int, mID int, stars int, ratingDate date, FOREIGN KEY
fk_std(RollNo) REFERENCES Student(RollNo), FOREIGN KEY fk_mv(mID) REFERENCES
Movie(mID),PRIMARY KEY(RollNo,mID,stars));
```

[Note that the datasets given are only representative. Testing may be done using other datasets also.]

```
/* Populate the table Student */
insert into Student values (1510, 'Anubhav', 9);
insert into Student values (1689, 'Deepika', 9);
insert into Student values (1381, 'Debajyoti', 9);
insert into Student values (1709, 'Sachin', 9);
insert into Student values (1101, 'Sunaina', 10);
insert into Student values (1782, 'Sharmila', 10);
insert into Student values (1468, 'Krushna', 10);
insert into Student values (1641, 'Parikshit', 10);
insert into Student values (1247, 'Deepak', 11);
insert into Student values (1316, 'Harsha', 11);
insert into Student values (1911, 'Deepika', 11);
insert into Student values (1501, 'Mayank', 11);
insert into Student values (1304, 'Anubhav', 12);
insert into Student values (1025, 'Rishabh', 12);
insert into Student values (1934, 'Shikha', 12);
insert into Student values (1661, 'Achal', 12);

/* Populate the table Friend */

insert into Friend values (1510, 1381);
insert into Friend values (1510, 1689);
insert into Friend values (1689, 1709);
insert into Friend values (1381, 1247);
insert into Friend values (1709, 1247);
insert into Friend values (1689, 1782);
insert into Friend values (1782, 1468);
insert into Friend values (1782, 1316);
insert into Friend values (1782, 1304);
insert into Friend values (1468, 1101);
```

```
insert into Friend values (1468, 1641);
insert into Friend values (1101, 1641);
insert into Friend values (1247, 1911);
insert into Friend values (1247, 1501);
insert into Friend values (1911, 1501);
insert into Friend values (1501, 1934);
insert into Friend values (1316, 1934);
insert into Friend values (1934, 1304);
insert into Friend values (1304, 1661);
insert into Friend values (1661, 1025);
insert into Friend values (1510, 1304);


/* Populate the table Likes */

insert into Likes values(1689, 1709);
insert into Likes values(1709, 1689);
insert into Likes values(1782, 1709);
insert into Likes values(1911, 1247);
insert into Likes values(1247, 1468);
insert into Likes values(1641, 1468);
insert into Likes values(1316, 1304);
insert into Likes values(1501, 1934);
insert into Likes values(1934, 1501);
insert into Likes values(1025, 1101);
insert into Likes values(1381, 1510);
insert into Likes values(1689, 1510);
insert into Likes values(1304, 1510);


/* Populate the table Movie */
insert into Movie values(101, 'Gone with the Wind', 1939, 'Victor Fleming');
insert into Movie values(102, 'Star Wars', 1977, 'George Lucas');
insert into Movie values(103, 'The Sound of Music', 1965, 'Robert Wise');
insert into Movie values(104, 'E.T.', 1982, 'Steven Spielberg');
insert into Movie values(105, 'Titanic', 1997, 'James Cameron');
insert into Movie values(106, 'Snow White', 1937, null);
insert into Movie values(107, 'Avatar', 2009, 'James Cameron');
insert into Movie values(108, 'Raiders of the Lost Ark', 1981, 'Steven
Spielberg');

/* Populate the table Rating */
insert into Rating values(1510, 101, 2, '2017-01-22');
insert into Rating values(1510, 101, 4, '2017-01-27');
insert into Rating values(1510, 103, 4, '2017-01-22');
insert into Rating values(1510, 104, 5, '2017-01-22');
insert into Rating values(1689, 106, 4, null);
insert into Rating values(1381, 103, 2, '2017-01-20');
insert into Rating values(1381, 108, 4, '2017-01-12');
insert into Rating values(1381, 108, 2, '2017-01-30');
insert into Rating values(1709, 101, 3, '2017-01-09');
insert into Rating values(1101, 103, 3, '2017-01-27');
insert into Rating values(1101, 104, 2, '2011-01-22');
insert into Rating values(1101, 108, 4, null);
insert into Rating values(1782, 107, 3, '2017-01-15');
insert into Rating values(1782, 106, 5, '2017-01-19');
insert into Rating values(1468, 107, 5, '2017-01-20');
```

```
insert into Rating values(1641, 104, 3, '2017-01-02');
insert into Rating values(1641, 107, 3, '2017-01-25');
```