**North South University** 

**CSE 311L ID:**

**Lab Final Exam Name :**

**Time : 1h 30 min Sec :**

**Total Points : 100**

For the following relation schema:

*Students(ID, name, status)*

*GradeHistory(ID,courseID, courseCredit, grade\_point)*

*Friend(ID1,ID2)*

*Likes(ID1, ID2)*

*Friend table represents unidirectional relationships i.e if id1 = 1510 and id2 = 1381 means 1510 friends with 1381 but not vice versa.*

*Likes table represents unidirectional relationships i.e if id1 = 1689 and id2 = 1709 means 1689 Likes 1709 but not vice versa.*

The four years of undergraduate education are called: freshman, sophomore, junior, Senior.

|  |  |  |  |
| --- | --- | --- | --- |
| *Students* | *Friend* | *Likes* | *GradeHistory* |
|  |  |  |  |

Write an expression in SQL for each of the following queries:

**North South University** 

**CSE 311L ID:**

**Lab Final Exam Name :**

**Time : 1h 30 min Sec :**

**Total Points : 100**

**1) Answer any 8 questions out of 9 : Each question carries 10 points. *points : 80***

A. Find the names of all students who are friends with someone named Homer J. Simpson. B. For every pair of students who both like each other, return the name and status of both students. C. Find all students who do not appear in the Likes table (as a student who likes or is liked) and return their names and status. Sort by status, then by name within each status.

D. For every situation where student A likes student B,but we have no information about whom B likes (that is, B does not appear as an ID1 in the Likes table),return A and B's names and status. E. For each student A who likes student B where the two are not friends,but student B likes a different student C,For all such trios, return the name and grade of A, B, and C. F. Find the name and status of all students who are liked by more than one other student. G. What is the average number of friends per student? (Your result should be just one number.) H. Find the number of students who are either friends with Tom or are friends of Tom. Do not count Tom, even though technically he is a friend of a friend.

I. Find the name and status of the student(s) with the greatest number of friends.

2) Create a t*able called CGPA(ID, cgpa) which stores current c.g.p.a for each student and* ***points : 20*** *write a trigger on the GradeHistory table which automatically updates the CGPA table while any changes happen.*

*Select count(select s.id from students s join friend f on s.id= f.id1 and s.id=f.id2 where f.id1 = f.id2 or f.id2=f.id1 )/ count(id)*

*From students;*