SQL Social-Network Query Exercises

Q1

(1/1 point)

Find the names of all students who are friends with someone named Gabriel.

select name

from highschooler

where id in (

select id1

from highschooler h1,friend f1

where h1.id = f1.id2

and name = 'Gabriel');

Q2

(1/1 point)

For every student who likes someone 2 or more grades younger than themselves, return that student's name and grade, and the name and grade of the student they like.

select distinct sName, sGrade, lName, lGrade

from (select h1.name as sName, h1.grade sGrade, h2.name as lName, h2.grade as lGrade, h1.grade-h2.grade as gradeDiff

from Highschooler h1, Likes, Highschooler h2

where h1.ID=ID1 and h2.ID=ID2)

where gradeDiff>1;

Q3

(1/1 point)

For every pair of students who both like each other, return the name and grade of both students. Include each pair only once, with the two names in alphabetical order.

select h1.name, h1.grade, h2.name, h2.grade from Likes l1, Likes l2, Highschooler h1, Highschooler h2

where l1.ID1=l2.ID2 and l2.ID1=l1.ID2 and l1.ID1=h1.ID and l1.ID2=h2.ID and h1.name<h2.name;

Q4

(1/1 point)

Find all students who do not appear in the Likes table (as a student who likes or is liked) and return their names and grades. Sort by grade, then by name within each grade.

select name,grade from Highschooler

where ID not in (select ID1 from Likes

union select ID2 from Likes) order by grade, name;

Q5

(1/1 point)

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 grades.

select distinct A.name, A.grade, B.name, B.grade

from Highschooler A, Likes, Highschooler B

where A.ID = Likes.ID1

and Likes.ID2 = B.ID and B.ID not in (select ID1 from Likes);

Q6

(1/1 point)

Find names and grades of students who only have friends in the same grade. Return the result sorted by grade, then by name within each grade.

select name, grade from Highschooler

where ID not in (

select ID1 from Highschooler A, Friend, Highschooler B

where A.ID = Friend.ID1 and Friend.ID2 = B.ID and A.grade <> B.grade)

order by grade, name;

Q7

(1/1 point)

For each student A who likes a student B where the two are not friends, find if they have a friend C in common (who can introduce them!). For all such trios, return the name and grade of A, B, and C.

select distinct A.name, A.grade, B.name, B.grade, C.name, C.grade

from Highschooler A, Likes, Highschooler B, Highschooler C, Friend F1, Friend F2

where A.ID = Likes.ID1 and Likes.ID2 = B.ID and

B.ID not in (select ID2 from Friend where ID1 = A.ID) and

A.ID = F1.ID1 and F1.ID2 = C.ID and

C.ID = F2.ID1 and F2.ID2 = B.ID;

Q8

(1/1 point)

Find the difference between the number of students in the school and the number of different first names.

select s.sNum-nm.nNum from

(select count(\*) as sNum from Highschooler) as s,

(select count(distinct name) as nNum from Highschooler) as nm;

Q9

(1/1 point)

Find the name and grade of all students who are liked by more than one other student.

select name, grade

from (select ID2, count(ID2) as numLiked from Likes group by ID2), Highschooler

where numLiked>1 and ID2=ID;