SQL Social-Network Query Exercises

COURSE - Databases: Relational Databases and SQL

StanfordOnline

By Njabulo Hlabangana

Background

Students at your hometown high school have decided to organize their social network using databases. So far, they have collected information about sixteen students in four grades, 9-12. Here's the schema:

Highschooler (ID, name, grade)
There is a high school student with unique *ID* and a given *first name* in a certain *grade*.

Friend (ID1, ID2)
The student with *ID1* is friends with the student with *ID2*. Friendship is mutual, so if (123, 456) is in the Friend table, so is (456, 123).

Likes (ID1, ID2)
The student with *ID1* likes the student with *ID2*. Liking someone is not necessarily mutual, so if (123, 456) is in the Likes table, there is no guarantee that (456, 123) is also present.

Queries are executed using SQLite

Highschooler

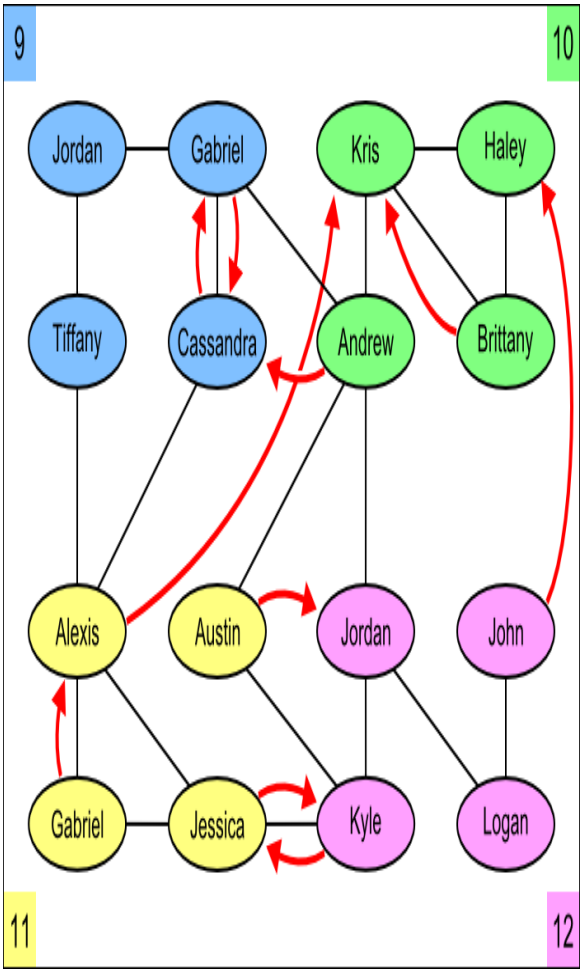
ID	name	grade
1510	Jordan	9
1689	Gabriel	9
1381	Tiffany	9
1709	Cassandra	9
1101	Haley	10
1782	Andrew	10
1468	Kris	10
1641	Brittany	10
1247	Alexis	11
1316	Austin	11
1911	Gabriel	11
1501	Jessica	11
1304	Jordan	12
1025	John	12
1934	Kyle	12
1661	Logan	12

Likes

ID1	ID2
1689	1709
1709	1689
1782	1709
1911	1247
1247	1468
1641	1468
1316	1304
1501	1934
1934	1501
1025	1101

Friend

ID1	ID2
1510	1381
1510	1689
1689	1709
1381	1247
1709	1247
1689	1782
1782	1468
1782	1316
1782	1304
1468	1101
1468	1641
1101	1641
1247	1911
1247	1501
1911	1501
1501	1934
1316	1934
1934	1304
1304	1661
1661	1025
1381	1510
1689	1510
1709	1689
1247	1381
1247	1709
1782	1689
1468	1782
1316	1782
1304	1782
1101	1468
1641	1468
1641	1101
1911	1247
1501	1247
1501	1911
1934	1501
1934	1316
1304	1934
1661	1304
1025	1661



SQL Social-Network Query Exercises

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

 select name
from highschooler
where ID in
(select ID2
from friend
where ID1 in
(select ID from highschooler where name = 'Gabriel')
union
select ID1
from friend
where ID2 in
(select ID from highschooler where name = 'Gabriel'))


--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 name_lk,grade_lk,name_lkd,grade_lkd
from
(select distinct name_lk,grade_lk,name_lkd,grade_lkd
from
(select l.ID1 as lkID,name as name_lk,grade as grade_lk, l.ID2 as lk_ID2
from likes l ,highschooler
where l.ID1 = ID)t1
JOIN
(select l2.ID2 as lkd_ID2, name as name_lkd,grade as grade_lkd
 from likes l2,highschooler
 where l2.ID2 = ID)t2
on lk_ID2 = lkd_ID2)
where grade_lk - grade_lkd >= 2

--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 distinct name_lk, grade_lk,name_lkd, grade_lkd
from
(select l.ID1 as lkID,name as name_lk,grade as grade_lk, l.ID2 as lk_ID2
from likes l ,highschooler
where l.ID1 = ID)t1
JOIN
(select l2.ID2 as lkd_ID2, name as name_lkd,grade as grade_lkd
 from likes l2,highschooler
 where l2.ID2 = ID)t2
on lk_ID2 = lkd_ID2
where lkID in
(select ID1
 from likes
 where ID1 in (select distinct ID2 from likes) and ID2 in (select distinct
ID1 from likes)) and name_lk < name_lkd
ORDER BY name_lk,name_lkd
```

```
--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



--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 name_lk,grade_lk,name_lkd,grade_lkd
from
(select l.ID1 as lkID,name as name_lk,grade as grade_lk, l.ID2 as lk_ID2
from likes l ,highschooler
where l.ID1 = ID)t1
JOIN
(select l2.ID2 as lkd_ID2, name as name_lkd,grade as grade_lkd
 from likes l2,highschooler
 where l2.ID2 = ID)t2
on lk_ID2 = lkd_ID2
where lkd_ID2 not in (select ID1 from likes) and lkd_ID2 in (select ID2 from
likes)



--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_lk,grade_lk
from
(select l.ID1 as lkID,name as name_lk,grade as grade_lk, l.ID2 as lk_ID2
from friend l ,highschooler
where l.ID1 = ID)t1
JOIN
(select l2.ID2 as lkd_ID2, name as name_lkd,grade as grade_lkd
 from friend l2,highschooler
 where l2.ID2 = ID)t2
on lk_ID2 = lkd_ID2
group by lkID
having avg(grade_lk) = avg(grade_lkd)
order by grade_lk,name_lk
```

```
--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 name_A,grade_A,name_B,grade_B,name_C,grade_C
from
(select distinct l.ID2 as I1,f.ID2 as fr1,name as name_A,grade as grade_A
from friend f,highschooler ,likes l
where f.ID1 not in (select distinct friend.ID1
from friend,likes
where friend.ID1 = likes.ID1 and friend.ID2 = likes.ID2) and l.ID1 = f.ID1
and ID = f.ID1)
JOIN
(select distinct f1.ID1 as I2, f1.ID2 as fr2, h.name as name_B,h.grade as
grade_B,h2.name as name_C,h2.grade as grade_C
from friend f1,likes l1,highschooler h,highschooler h2
where f1.ID1 not in (select distinct friend.ID1
from friend,likes
where friend.ID1 = likes.ID2 and friend.ID2 = likes.ID2) and l1.ID2 = f1.ID1
and f1.ID1 = h.ID and h2.ID = f1.ID2)
on I1 = I2 and fr1 = fr2
order by name_A



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

select num_of_students - names
from
(select count(ID) as num_of_students, count(distinct name) as names
from highschooler)



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

select distinct name_lkd,grade_lkd
from
(select l.ID1 as lkID,name as name_lk,grade as grade_lk, l.ID2 as lk_ID2
from likes l ,highschooler
where l.ID1 = ID)t1
JOIN
(select l2.ID2 as lkd_ID2, name as name_lkd,grade as grade_lkd
 from likes l2,highschooler
 where l2.ID2 = ID)t2
on lk_ID2 = lkd_ID2
group by lkd_ID2
having count(lkID) > 1
```