Feedback — SQL Social-Network Modification Exercises

Help Center

You submitted this quiz on **Fri 20 Mar 2015 9:22 PM PDT**. You got a score of **3.00** out of **3.00**.

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)
English: There is a high school student with unique *ID* and a given *first name* in a certain *grade*.

Friend (ID1, ID2)
English: 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)
English: 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.

Your modifications will run over a small data set conforming to the schema. View the database. (You can also download the schema and data.)

For your convenience, here is a graph showing the various connections between the people in our database. 9th graders are blue, 10th graders are green, 11th graders are yellow, and 12th graders are purple. Undirected black edges indicate friendships, and directed red edges indicate that one person likes another person.

Social graph

Instructions: You are to write each of the following data modification commands using SQL. The workbench runs each modification using SQLite on the original state of the sample database. It then performs a query over the modified database to check whether your command made the correct modification, and restores the database to its original state.

You may perform these exercises as many times as you like, so we strongly encourage you to keep working with them until you complete the exercises with full credit.

NOTE: REMEMBER TO CLICK "Submit" WHEN YOU ARE DONE!
Please be patient as it does take time to check all of the exercises.

Question 1

It's time for the seniors to graduate. Remove all 12th graders from Highschooler.

You entered:

```
delete from Highschooler
where grade = 12
```

Run Command

Your Answer		Score	Explanation
delete from Highschooler where grade = 12	✓	1.00	**Correct** To check your data modification statement, we ran the following query after your modification: *select * from Highschooler order by ID*

Your Query Result:

1101	Haley	10
1247	Alexis	11
1316	Austin	11
1381	Tiffany	9
1468	Kris	10
1501	Jessica	11
1510	Jordan	9
1641	Brittany	10
1689	Gabriel	9
1709	Cassandra	9
1782	Andrew	10
1911	Gabriel	11

Expected Query Result:

1101	Haley	10
1247	Alexis	11
1316	Austin	11
1381	Tiffany	9
1468	Kris	10
1501	Jessica	11
1510	Jordan	9
1641	Brittany	10
1689	Gabriel	9
1709	Cassandra	9
1782	Andrew	10
1911	Gabriel	11

Total	1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your data modification command does not correctly reflect the problem at hand. All we check is that the verification query gets the right answer on the small sample database. Circumventing the system will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 2

If two students A and B are friends, and A likes B but not vice-versa, remove the Likes tuple.

You entered:

```
delete from Likes
where ID1 in (select L.ID1
                        from Likes as L, Friend as F
                        where F.ID1 = L.ID1
                              and F.ID2 = L.ID2
```

Run Command

Your Answer		Score	Explanation
delete from Likes where ID1 in (select L.ID1 from Likes as L, Friend as F where F.ID1 = L.ID1 and F.ID2 = L.ID2 and L.ID1 not in (select ID2	✓	1.00	**Correct** To check your data modification statement, we ran the following query after your modification: *select H1.name, H1.grade, H2.name, H2.grade from Likes L, Highschooler H1, Highschooler H2 where L.ID1 = H1.ID and L.ID2 = H2.ID order by H1.name, H1.grade*

Your Query Result:

Alexis	11	Kris	10
Andrew	10	Cassandra	9
Austin	11	Jordan	12
Cassandra	9	Gabriel	9

```
        from Likes

        where ID1
= L.ID2))
        and ID2 in
(select L.ID2

                from Li
kes as L, Friend
as F

                where
F.ID1 = L.ID1

                        a
nd F.ID2 = L.ID2

                        a
nd L.ID1 not in (s
elect ID2

                from Li
kes

                where I
D1 = L.ID2))
```

Gabriel	9	Cassandra	9
Jessica	11	Kyle	12
John	12	Haley	10
Kyle	12	Jessica	11

Expected Query Result:

Alexis	11	Kris	10
Andrew	10	Cassandra	9
Austin	11	Jordan	12
Cassandra	9	Gabriel	9
Gabriel	9	Cassandra	9
Jessica	11	Kyle	12
John	12	Haley	10
Kyle	12	Jessica	11

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your data modification command does not correctly reflect the problem at hand. All we check is that the verification query gets the right answer on the small sample database. Circumventing the system will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a

general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 3

For all cases where A is friends with B, and B is friends with C, add a new friendship for the pair A and C. Do not add duplicate friendships, friendships that already exist, or friendships with oneself. *(This one is a bit challenging; congratulations if you get it right.)*

You entered:

```
insert into Friend
select distinct F1.ID1 as A, F2.ID2 as C -- distinct is needed A
can get to C through multiple Bs
     from Friend as F1, Friend as F2
     where F1.ID2 = F2.ID1 -- this is B
```

Run Command

Your Answer **Score** **Explanation**

```
insert into Friend
select distinct F1.ID1
as A, F2.ID2 as C --
distinct is needed A c
an get to C through
multiple Bs
     from Friend as F
1, Friend as F2
     where F1.ID2 =
F2.ID1 -- this is B
          and F1.ID
1 <> F2.ID2 -- does n
ot add self as friend
          and F2.ID
2 not in (select ID2

               from F
riend

               wher
e ID1 = F1.ID1)  -- do
es not add duplicate f
riendship
```

✓ 1.00

Correct

To check your data modification statement, we ran the following query after your modification: *select ID, name, grade, (select count(*) from Friend where id1 = H.id) from Highschooler H order by ID*

Your Query Result:

1025	John	12	2
1101	Haley	10	3
1247	Alexis	11	7
1304	Jordan	12	8
1316	Austin	11	6
1381	Tiffany	9	6
1468	Kris	10	6
1501	Jessica	11	7
1510	Jordan	9	5
1641	Brittany	10	3

1661	Logan	12	4
1689	Gabriel	9	8
1709	Cassandra	9	7
1782	Andrew	10	10
1911	Gabriel	11	5
1934	Kyle	12	7

Expected Query Result:

1025	John	12	2
1101	Haley	10	3
1247	Alexis	11	7
1304	Jordan	12	8
1316	Austin	11	6
1381	Tiffany	9	6
1468	Kris	10	6
1501	Jessica	11	7
1510	Jordan	9	5
1641	Brittany	10	3
1661	Logan	12	4
1689	Gabriel	9	8
1709	Cassandra	9	7
1782	Andrew	10	10
1911	Gabriel	11	5
1934	Kyle	12	7

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your data modification command does not correctly reflect the problem at hand. All we check is that the verification query gets the right answer on the small sample database. Circumventing the system will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.