

Project Assignment 3

To understand integrity and be able to create tables with constraints and to create triggers.

Overview

Students at 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:

1. **Highschooler (ID, name, grade)**, ID is the Primary key.
English: There is a high school student with unique ID and a given first name in a certain grade.
2. **Friend (ID1, ID2)**, Primary key (ID1, ID2), ID1 is a Foreign key references Highschooler(ID), ID2 is also a Foreign key references Highschooler(ID)
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).
3. **Likes (ID1, ID2)**, Primary key (ID1, ID2), ID1 is a Foreign key references Highschooler(ID), ID2 is also a Foreign key references Highschooler(ID)
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. One student could like more than one other students.
4. **graduates(ID, name)**, ID is the Primary key. It is also a Foreign key references Highschooler(ID)
English: The student with ID and name is graduated.

Tasks

Task 1 Create tables (2 points each table)

Q0: Write the create table statement to define the schema, primary key and foreign key(s) for each table. Each statement begins on a new line. Don't forget using 'engine=innodb' to support your referential integrity implementation.

Once the tables are created, you can use statements in *cs211-project3-social-data.sql* (which has been attached) to load data into database.

Task2 Create triggers (2 points each)

Important notes:

Your submission are graded using MariaDB. If you use **alternatives**, e.g. MS SQL, please ensure the queries you submit are executable on **MariaDB** Server.

Each question is independent, please consider each question's trigger(s) will be the only trigger(s) for the database.

Use test cases in file *cs211-project3-test cases.txt* to verify your trigger.

You may need to change the Delimiter to `//` when creating a trigger, and change it back to `;` when the creating is done.

Q1: Write a trigger that makes new students named 'Friendly' automatically like everyone else in their grade. That is, after the trigger runs, we should have ('Friendly', A) in the Likes table for every other Highschooler A in the same grade as 'Friendly'.

Q2: Write a trigger to manage the grade attribute of new Highschoolers. If the inserted tuple has a value less than 9 or greater than 12, change the value to 9.

Q3: Write a trigger so when a student A graduates, i.e., when his/her grade is updated to exceed 12, he or she will not be friend with those who were in the same grade before (for example B, (A, B) and (B, A) both removed from friend table). Note that graduation is an action happening on the highschooler table.

Q4: Write a trigger that automatically backup students when they graduate. When their grade is updated to exceed 12, (a) to copy the student ID and name to the graduates table, (b) update the student's grade to -1.

Submission

Write your SQL statements in one text file `cs211_Proj3_yourusername.sql` (containing your solutions) as your submission to moodle submission page. Your file should look like this:

```
-- Q0
CREATE ...

-- Q2
CREATE ...

-- Q3
...
```