CS 122A Course Assignment 5
SQL Constraints, Triggers, Views
Fall 2019

Due: Monday, November 25th (11:59 pm)

For Problem 1 you do not need to use mysql, however for Problem 2 and Problem 3 you should test your answers on mysql (use the schema and data provided for assignment 4).

Problem 1 - SQL Constraints

Part A:

Using CREATE TABLE statement create tables for :

- Student: sid, gpa, major
  Major is either CS or EECS. GPA is between 0 and 4.
- Course: cid, level
  Level is either Introductory or Advanced
- Taken_courses: sid, cid

Sid refers to Student.sid and cid to Courses.cid. If a student record is deleted from the Student table, its record must also needs to be deleted from Taken_courses table, however if a course is deleted from the Course table, records associated with that cid in Taken_courses table should not change. Furthermore, if the course cid is changed in the Course table, the change should be reflected in the Taken_course table. *Write the create table statements for each of the above tables with the above mentioned constraints.*

After creating tables, *create an assertion* that disallows students with a GPA below 3.0 from taking more than 2 advanced courses, i.e., a tuple <sid,cid> cannot be in the Taken_courses table, if sid corresponds to a student whose GPA is below 3.0, the cid corresponds to an Advanced level course, and the student has already taken 2 advanced courses.

Part B:

Consider the following relational schema and briefly answer the following questions:

Emp(eid:integer, ename:string, age:integer, salary:real)
Works(eid:integer, did:integer, pcttime:integer)
Dept(did:integer, budget:real, managerid:integer)

a) Define an attribute-based constraint on Emp that will ensure that every employee makes at most $10,000.

b) Define a tuple-based constraint on Dept that will ensure that all managers have age > 50.

c) Define an assertion on Dept that will ensure that all managers have age > 50.

d) Compare assertion in (c) with the equivalent tuple-based constraint in (b). Explain which is better.

e) Define a row-level trigger that no update can reduce employee's salary

Problem 2 - Views

UCI has adopted Smart Waste Management and is replacing all their trash cans with smart bins. There are different kinds of users for the Smart Waste Management System such as App Users, Data Analysts, and Facility Managers. The campus management would like to control what data is accessible to which type of users by creating appropriate views and issuing appropriate GRANT permissions. They have asked you as database experts to help them by creating such views . They have three types of users: App Users, Sustainability Analysts, and Facility Managers.

App Users: are interested in knowing the locations of various types of bins to throw away their trash. However, the campus only wants to show information of the bins which are not currently full (the last load observation for that bin is less than its capacity) and are located inside buildings. *Current time is assumed as 2019-10-26 13:00:00.*

Example:

Waste bin id	x	y	Type of bin
142	1000	1400	Recycle

Sustainability Analysts: are interested in learning the patterns of waste disposal by students on UCI Campus to promote sustainability among students. The security policy, however, requires that analyst should not be allowed to have access to data about exactly which student threw away the trash and what type of trash they disposed. They only need to know information about the bin's location, department names of the student, and the total weight of the trash thrown away by them in the bins.

Example:

Waste bin id	x	y	Department	Total weight
183	100	100	Computer Science	100

Facility Managers: are interested in keeping track of user's waste disposal activities. They want to know what is their name, how many times each day they disposed off trash in each type of bin.

Example:

Name	Day	Compost Bin	LandFill Bin	Recycle Bin
ABC	2019-10-01	10	20	5

Your task in this question is to make views that allows UCI Campus to provide customized access to data based on the needs of their users and/or to prevent unauthorized access. Note that to prevent unauthorized access, you generally will need to specify GRANT permissions as well. But since we have not modeled users and their roles in the assignment so far, we will ignore the GRANT permissions and your task is to only create appropriate views.

Use the following queries to verify your answers:

```
Select * from App_Users;
Select* from Sustainability_Analysts;
Select* from Facility_Managers;
```

Problem 3 - Triggers

Identifying Malfunctioning Sensors: Our goal in this question is to write a trigger that replaces erroneous load sensor values by NULL when the erroneous values are inserted. Note that previous weight values recorded by the sensor that are not erroneous should not get replaced with null, only if any of the following conditions holds for a record before insertion, weight for that record should set to null and then get inserted. A load sensor value is considered to be erroneous if the following condition holds: *Erroneous Sensor Detection Condition:* If: (a) A load sensor value differs from the previous value of the same sensor by more than 1000 units. The value will be considered erroneous only if the previous value was recent, i.e., recorded in the last 24 hours. OR (b) There is a null reading already recorded by that load sensor. (**Hint:** For finding the time difference use DATEDIFF function).

After writing your trigger, run the following statements:

```
INSERT INTO LoadObservation(sensor_id, oid, Weight, timestamp) VALUES (350, 50001, 15000, '2017-07-07 20:00:55');
INSERT INTO LoadObservation(sensor_id, oid, Weight, timestamp) VALUES (350, 50002, 15500, '2017-07-17 22:00:55');
INSERT INTO LoadObservation(sensor_id, oid, Weight, timestamp) VALUES (350, 50003, 17000, '2017-07-18 20:45:55');
INSERT INTO LoadObservation(sensor_id, oid, Weight, timestamp) VALUES (350, 50004, 17500, '2017-07-20 20:50:55');
```

Then run the following query and write the results. Is the result what you would expect? Make sure you include your results and explanations in your answers.

```
Select * from LoadObservation where sensor_id = 350 and oid > 50000;
```

Trash Violations: In this question, you will write a trigger that will record trash violations from object recognition observations. You will first need to create a trash violation table named TrashViolations with the attributes of TVID (unique id), user_id (user who threw trash in the wrong bin), timestamp, waste_bin_id (id of the bin where the trash was thrown), trash_type(the type of trash thrown), that will be used to store the trash code violations.

A trash violation happens when trash is thrown in the wrong bin. For example, if compost trash is thrown in the recycle bin or recyclable trash is thrown in landfill. Your goal is to write a trigger that executes when entries for a trash throwing event is inserted into the ObjectRecognitionObservations table, and inserts the violations appropriately in the TrashViolations table. [**Hint**: You will need WasteBin(and other related tables), LocationSensor, LocationObservation table as well, to create an entry for the TrashViolations table]

After creating TrashViolations table and capacity_check trigger, run the following statements:

```
INSERT INTO LocationObservation(sensor_id, oid, timestamp, X, Y) VALUES (1, 100001, '2017-11-15 14:00:00', 5459, 3576);
INSERT INTO ObjectRecognitionObservation(sensor_id, oid, timestamp, trash_type) VALUES (354, 200001,  '2017-11-15 14:00:00', 'LandFill');
```

Run the following query and write the results. Is the result what you would expect? Make sure you include your results and explanations in your answers.

```
Select * from TrashViolations;
```

Deliverables:

Your assignment has to be submitted to **gradescope**. Only **one team member** has to submit the following three files :

1. **SQL script** containing the SQL statements for Problem 2 and 3 - which can be executed on MySQL command line processor. The name of the sql file should be last names of each team member placed together. For example if Edgar Codd, Donald Chamberlin and Peter Chen were teammates, they would submit: codd_chamberlin_chen_assignment5_script.sql.
2. A **PDF file** containing solution for all problems. The name of the pdf file should be last names of each team member placed together. For example if Edgar Codd, Donald Chamberlin and Peter Chen were teammates, they would submit codd_chamberlin_chen_assignment5.pdf.

Instructions

Creating the result of a SQL file: You need to create a TXT file to include your queries and its results. Points may be deducted if you don't follow the instructions. Here are the instructions. **Please do not copy text and type each command. When copying text from a PDF file in the terminal, it does not work sometimes.**

1. Open a blank text file in a text editor and copy the following template into the file. Then, paste your SQL statements after the "-- Paste .." comment.

```
USE `cs122a_test`;

-- Paste your SQL statements in the below
```

2. Save it as “script.sql”.

3. You are going to use **mysql** command-line tool (http://dev.mysql.com/doc/refman/5.0/en/mysql.html) to execute your script and generate an output. Execute the following command to load your SQL script and generate an output. Here, we assume that “script.sql” is located in your home directory. The result file name should be script_output.txt. Do not convert it to other formats such as DOC or PDF. In the command prompt (or terminal), execute the following command (not after executing mysql). In case you cannot run below command, please run script.sql and copy the output to script_output.txt file.

```
mysql --force --comments -v -v -u root < yourHomePath/script.sql > yourHomePath/script_output.txt
```

Do not omit any options. Especially the option **–v** is repeated twice on purpose. If the root account has a password, use the following command.

```
mysql --force --comments -v -v -u root -pYOURPASSWORD < yourHomePath/script.sql > yourHomePath/script_output.txt
```

* OS Specific instructions -

Windows

1. Put your script in a folder that your account can access. (e.g., d:\)
2. Open a command prompt (cmd) and go to the MySQL folder.

```
cd C:\Program Files (x86)\MySQL\MySQL Server 8.0\bin
```

3. Execute the following command.

```
mysql --force --comments -v -v -u root < d:\script.sql > d:\script_output.txt
```

OS X

1. Put your script in your home directory. (e.g., /Users/youraccount)
2. Open a terminal and execute the following command.

```
/usr/local/mysql/bin/mysql --force --comments -v -v -u root < /Users/youraccount/script.sql > /Users/youraccount/script_output.txt
```

If you have a problem to execute mysql tool, here is the default location that you can find it. It might be a good idea to move script.sql to a folder that your account has an access like the above.

OS X: /usr/local/mysql/bin/mysql
Windows: C:\Program Files (x86)\MySQL\MySQL Server 8.0\bin

Refer to the following guideline for the mysql command-line tool.
For OS X | For Windows