CSC 174 - Advanced Database Management Systems

# Homework Assignment 3: 6\_b\_trigger

#### Overview:

- Create triggers to maintain the consistency of the derived attribute NoOfStudents in the course table.
  - 1. When adding a new record in the enrollment table, update the corresponding value of NoOfStudents enrolled in the course.
  - 2. When deleting a record in the enrollment table, update the corresponding value of the NoOfStudents
  - ★ Note: I included the set up that allows me to run these commands. There is some inconsistency in the data because I changed some data after taking screenshots, but it works.

### Before executing the scenarios we need to create the triggers.

Run 'source CSC174/hw3/6\_a\_trigger.sql' this will execute the .sql file and create the trigger.
 \*This may vary depending on the path of the files.\*

```
mysql> source CSC174/hw3/6_a_trigger.sql
Query OK, 0 rows affected (0.01 sec)
```

Figure 1: Running the sql file to create triggers.

#### Scenario 1: Enrolling new students into a Course.

- 1a). Successful enrollment.
  - We want to enroll Ivana Tinkle and Yuri Nator into Professor Ying Jin's Advanced
    Database Management Systems course. The Advanced Database course currently has 3
    students enrolled. We know this from the 'SELECT \*FROM Course WHERE
    CourseNo=12345;' and we can see which of those students are in that course with the
    Procedure 'CALL GetStudentCourse(12345);'

```
mysql> SELECT * FROM Course WHERE CourseNo=12345;
  CourseNo | CourseName
                                                                  | NoOfStudents | InstructorID | TASSN
                                                                                            111111111 | 123456789 |
      12345 | Advanced Database Management Systems |
  row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
                                                                                                        | Email
  SSN
                 StudentName
                                       | Address
  123456789 | Moe Ron | 16514 Golden Valley Pkwy, Lathrop, CA 95330 | moeron@fake.com
214407735 | Seymour Butz | 2727 W March Ln, Stockton, CA 95219 | seymourbutts@fai
678954321 | Amanda Hugginkiss | 4600 Madison Ave, Sacramento, CA 95841 | amandahugginkiss
                                                                                                         seymourbutts@fake.com
                                                                                                         amandahugginkiss@fake.com
  rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
mysql>
```

Figure 2: Three students in Advanced Database course.

Yuri Nator turned in his add slip before Ivana Tinkle. So we enroll Yuri first by running,
After running the 'INSERT INTO Enrolled VALUES (890123456, 12345, 'B');' We can
check if the trigger updated the NoOfStudents in the class with the same commands
above

```
mysql> INSERT INTO Enrolled VALUES (890123456, 12345, 'B');
Query OK, 1 row affected (0.00 sec)
mysql> SELECT * FROM Course WHERE CourseNo=12345;
  CourseNo | CourseName
                                                                      | NoOfStudents | InstructorID | TASSN
      12345 | Advanced Database Management Systems |
                                                                                       4 | 111111111 | 123456789
1 row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
                                                                                                      | Email
  SSN
                | StudentName | Address
  123456789 | Moe Ron | 16514 Golden Valley Pkwy, Lathrop, CA 95330 | moeron@fake.com
214407735 | Seymour Butz | 2727 W March Ln, Stockton, CA 95219 | seymourbutts@fake.com
678954321 | Hugh Jass | 4600 Madison Ave, Sacramento, CA 95841 | hughjass@fake.com
890123456 | Yuri Nator | 3501 Truxel Rd, Sacramento, CA 95834 | yurinator@fake.com
                                                                                                        seymourbutts@fake.com
                                                                                                        hughjass@fake.com
                                                                                                      | yurinator@fake.com
4 rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
mysql>
```

Figure 3: NoOfStudents was updated from 3 to 4.

The registration office has received Ivana Tinkle's add slip, meaning we can now add her into the course. To do so we run 'INSERT INTO Enrolled VALUES (789012345, 12345, 'B');'. Then run the above commands to verify.

```
mysql> INSERT INTO Enrolled VALUES (789012345, 12345, 'B');
Query OK, 1 row affected (0.01 sec)
mysgl> SELECT * FROM Course WHERE CourseNo=12345;
                                                                   | NoOfStudents | InstructorID | TASSN
  CourseNo | CourseName
      12345 | Advanced Database Management Systems | 5 | 111111111 | 123456789 |
1 row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
 SSN
              | StudentName | Address
                                                                                                   | Email
  123456789 | Moe Ron
                                    | 16514 Golden Valley Pkwy, Lathrop, CA 95330 | moeron@fake.com
  214407735 | Seymour Butz | 2727 W March Ln, Stockton, CA 95219 | seymourbutts@fake.com 678954321 | Hugh Jass | 4600 Madison Ave, Sacramento, CA 95841 | hughjass@fake.com 789012345 | Ivana Tinkle | 2475 Sunrise Blvd, Rancho Cordova, CA 95670 | ivanatinkle@@fake.com 890123456 | Yuri Nator | 3501 Truxel Rd, Sacramento, CA 95834 | yurinator@fake.com
5 rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
mysql>
```

- Figure 4: NoOfStudents was updated from 4 to 5.
- Yuri Nator and Ivana Tinkle are both happy about being enrolled in Advanced Database Management Systems. They are now one step closer to graduating.

### 1b). Unsuccessful enrollment.

A friend of Ivana Tinkle was so excited to be enrolled in Advanced Database
 Management Systems that she wrote the wrong Student ID on her add slip. To simulate this scenario run 'INSERT INTO Enrolled VALUES (734572095, 12345, 'B');'

```
mysql> INSERT INTO Enrolled VALUES (734572095, 12345, 'B');

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`cs174119`.`Enrolled`, CONSTRAINT `Enrolled_SSN_FK` FOREIGN KEY (`SSN`)

REFERENCES `Student` (`SSN`))

mysql> |
```

#### Figure 5: ERROR due to Constraints.

 Thanks to our foreign key constraints we were not able to add this invalid Student into this course. Unfortunately, Ivana Tinkle's friend will need to re-submit an add slip.

### Scenario 2: Dropping old students from a Course.

- 2a). Successful drop. \*I changed some names around but it's the same\*
  - Despite the joyous addition of classes for Ivana Tinkle and Yuri Nator, it seems as though two students already enrolled in Advanced Database Management Systems are not doing well in class. These two students are Moe Ron and Hugh Jass. Hugh Jass drop slip is the first to make it to the registration office. So to drop him we need to run 'DELETE FROM Enrolled WHERE SSN=678954321 AND CourseNo=12345;'.

```
ysql> SELECT * FROM Course WHERE CourseNo=12345;
 CourseNo | CourseName
                                                       | NoOfStudents | InstructorID | TASSN
                                                                    5
     12345 | Advanced Database Management Systems |
                                                                            111111111 | 123456789 |
1 row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
 SSN
             | StudentName | Address
                                                                                | Email
 123456789 | Moe Ron
                               16514 Golden Valley Pkwy, Lathrop, CA 95330 |
                                                                                 moeron@fake.com
                               2727 W March Ln, Stockton, CA 95219
4600 Madison Ave, Sacramento, CA 95841
2475 Sunrise Blvd, Rancho Cordova, CA 95670
              Seymour Butz
 214407735
                                                                                  seymourbutts@fake.com
 678954321
              Hugh Jass
                                                                                 hughjass@fake.com
              Ivana Tinkle
                                                                                 ivanatinkle@@fake.com
  789012345
 890123456
              Yuri Nator
                               3501 Truxel Rd, Sacramento, CA 95834
                                                                                  vurinator@fake.com
  rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
mysql> DELETE FROM Enrolled WHERE SSN=678954321 AND CourseNo=12345;
Ouerv OK, 1 row affected (0.00 sec)
mysql> SELECT * FROM Course WHERE CourseNo=12345;
 CourseNo | CourseName
                                                      | NoOfStudents | InstructorID | TASSN
     12345 | Advanced Database Management Systems |
                                                                            111111111 | 123456789
 row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
             | StudentName | Address
  123456789 |
                               16514 Golden Valley Pkwy, Lathrop, CA 95330 |
                                                                                 moeron@fake.com
                               2727 W March Ln, Stockton, CA 95219
2475 Sunrise Blvd, Rancho Cordova, CA 95670
               Seymour Butz
 214407735
                                                                                  seymourbutts@fake.com
  789012345
               Ivana Tinkle
                                                                                  ivanatinkle@@fake.com
 890123456
                               3501 Truxel Rd, Sacramento, CA 95834
              Yuri Nator
                                                                                 yurinator@fake.com
  rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
```

Figure 6: Hugh has been dropped. NoOfStudents updated from 5 to 4.

 Next drop slip to be received is from Moe Ron. To drop him from the course we need to run 'DELETE FROM Enrolled WHERE SSN=123456789 AND CourseNo=12345;'.
 Then run the previous to verify the trigger.

```
mysql> DELETE FROM Enrolled WHERE SSN=123456789 AND CourseNo=12345;
Query OK, 1 row affected (0.00 sec)
mysql> SELECT * FROM Course WHERE CourseNo=12345;
 CourseNo | CourseName
                                                    | NoOfStudents | InstructorID | TASSN
                                                             3 | 111111111 | 123456789
    12345 | Advanced Database Management Systems |
l row in set (0.00 sec)
mysql> CALL GetStudentCourse(12345);
            | StudentName | Address
                                                                            | Email
                             2727 W March Ln, Stockton, CA 95219
 214407735 | Seymour Butz |
                                                                            | seymourbutts@fake.com
              Ivana Tinkle
 789012345
                              2475 Sunrise Blvd, Rancho Cordova, CA 95670 | ivanatinkle@@fake.com
 890123456 | Yuri Nator
                             3501 Truxel Rd, Sacramento, CA 95834
                                                                            | yurinator@fake.com
3 rows in set (0.00 sec)
Query OK, O rows affected (0.00 sec)
```

- Figure 7: Moe Ron has been dropped. NoOfStudents updated from 4 to 3.
- Hugh Jass and Moe Ron have been successfully dropped and can continue their days away from grades lower than a C for now.

## • 2b). Unsuccessful drop.

It seems that Seymour Butz wants to drop Advanced Database Management Systems for another course, but he needs to drop this course first because of the time constraint. Sadly, he filled in the wrong Student ID. To simulate this we will run, 'DELETE FROM Enrolled WHERE SSN=215507725 AND CourseNo =12345;'

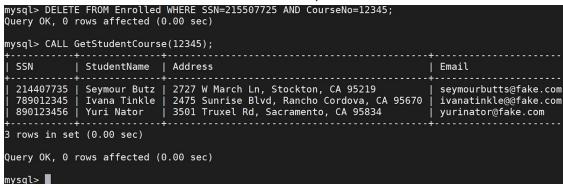


Figure 8: No one was dropped from the course.

### **Initial Set up:**

 First, we need to ensure that the tables have been created. To do so I run the command in MySQL, 'source CSC174/hw3/1\_create\_table.sql'. This may vary depending on where your .sql files are located.

```
mysql> source CSC174/hw3/1_create_table.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Figure 1: Tables have been created.

Now that the tables are created we need to populate the tables with data. I did this by running the command, 'source CSC174/hw3/2\_populate\_db.sql'. I've included SELECT statements to show all data in the data for verification and the views.

```
mysql> source CSC174/hw3/2_populate_db.sql
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
```

Figure 2: Populating the tables is successful.

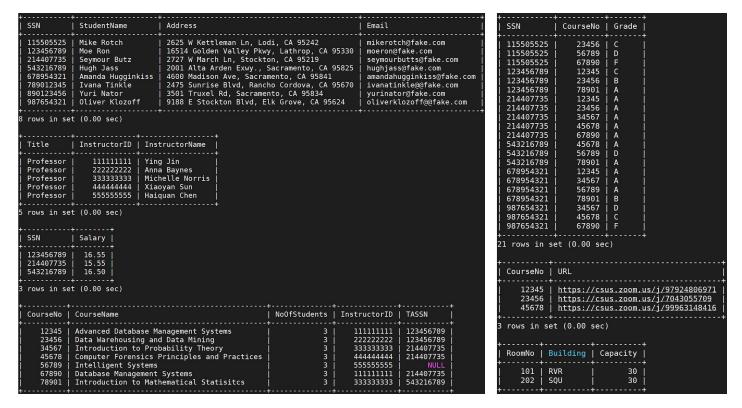


Figure 3: Tables (Student, Instructor, TA, Course)

Figure 4: Tables Cont.(Enrolled, OnlineCourse, Classroom, )

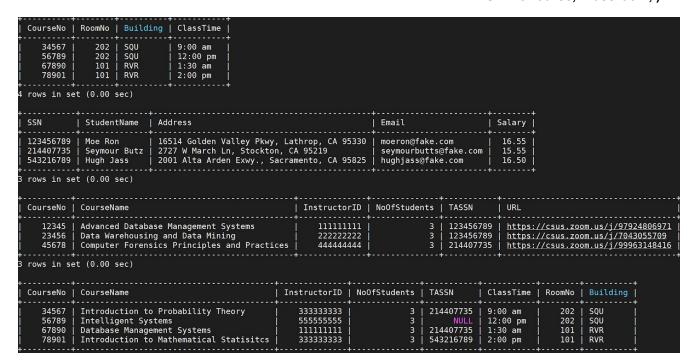


Figure 5: Tables Cont. (TradionalCourse) and Views (TAView, OnlineCourseView, TraditionalCourse View)

After the tables are populated we can now create the other views by running
 'source CSC174/hw3/3\_create\_view.sql'. I've provided some SELECT statements to show the
 contents of TA\_Course and Student\_Grade. TA\_Course returns TA information and the course
 they assist in. Student\_Grade returns the SSNs of students and the number of A's they received
 if they at least have 2 A's.

```
mysql> source CSC174/hw3/3_create view.sql
Query OK, O rows affected (0.00 sec)
              | TA_Email
 TA Name
                                      | Course Name
           | moeron@fake.com
 Moe Ron
                                       Advanced Database Management Systems
              moeron@fake.com
 Moe Ron
                                        Data Warehousing and Data Mining
 Seymour Butz | seymourbutts@fake.com
                                        Introduction to Probability Theory
 Seymour Butz | seymourbutts@fake.com
                                      | Computer Forensics Principles and Practices
                                        Database Management Systems
 Seymour Butz | seymourbutts@fake.com
 Hugh Jass | hughjass@fake.com
                                      | Introduction to Mathematical Statisitcs
 rows in set (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
 Student_SSN | NumberOfA
                       5
   214407735 |
                       2
   543216789
   678954321
                       3
 rows in set (0.00 sec)
mysql>
```

Figure 6: Views TA Course and Student Grade.

Now that this step is completed, we can continue to create the function Course\_Instructor. This
will return the name of the Instructor teaching that course name given to the function. We do this
by running, 'source CSC174/hw3/4\_func.sql'. In the file, I have provided all potential ways to
execute the function.

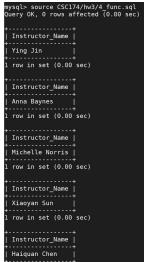


Figure 7: Successful execution of Function Course\_Instructor.

Next is the Procedure, Get\_TA, and GetStudentCourse. Get\_TA returns a list of TA's that assist
the corresponding Instructor's ID. While GetStudentCourse returns the student info of those
enrolled in the following course ID. To run this, 'source CSC174/hw3/5\_proc.sql'. In this file, I
included all possible calls for the following procedures.

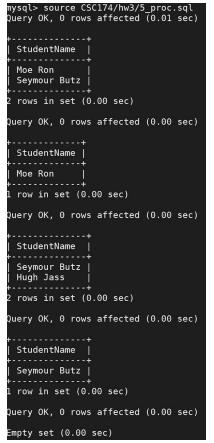


Figure 8: (Left) Get\_TA. 5 InstructorID to call. One is TAless so it's empty.

	StudentName	Address	Email
123456789     214407735     678954321	Moe Ron   Seymour Butz   Amanda Hugginkiss	16514 Golden Valley Pkwy, Lathrop, CA 9!   2727 W March Ln, Stockton, CA 95219   4600 Madison Ave, Sacramento, CA 95841	5330   moeron@fake.com   seymourbutts@fake.com   amandahugginkiss@fake.com
B rows in set	t (0.00 sec)	+	
Query OK, 0 i	rows affected (0.00 s	sec)	
SSN	++   StudentName   Add	ress	++   Email
115505525     123456789     214407735	Moe Ron   165:	5 W Kettleman Ln, Lodi, CA 95242 14 Golden Valley Pkwy, Lathrop, CA 95330 7 W March Ln, Stockton, CA 95219	mikerotch@fake.com     moeron@fake.com     seymourbutts@fake.com
3 rows in set	t (0.00 sec)		++
Query OK, 0 i	rows affected (0.00 s	sec)	
Query OK, 0 : +   SSN	rows affected (0.00 s +	sec) +   Address	Email
+	+	+	seymourbutts@fake.com     amandahugginkiss@fake.com
SSN   214407735   678954321   987654321		Address   Address   2727 W March Ln, Stockton, CA 95219   4600 Madison Ave, Sacramento, CA 95841	seymourbutts@fake.com     amandahugginkiss@fake.com
SSN   214407735   678954321   987654321   3 rows in set	StudentName   Seymour Butz   Amanda Hugginkiss   Oliver Klozoff	Address   Address   2727 W March Ln, Stockton, CA 95219   4600 Madison Ave, Sacramento, CA 95841   9188 E Stockton Blvd, Elk Grove, CA 956;	seymourbutts@fake.com     amandahugginkiss@fake.com
SSN   214407735   678954321   987654321   3 rows in set	StudentName   Seymour Butz   Amanda Hugginkiss   Oliver Klozoff   Coliver Stozoff   Stozoff	Address   Address   2727 W March Ln, Stockton, CA 95219   4600 Madison Ave, Sacramento, CA 95841   9188 E Stockton Blvd, Elk Grove, CA 956;	seymourbutts@fake.com     amandahugginkiss@fake.com
SSN	StudentName   Seymour Butz   Amanda Hugginkiss   Oliver Klozoff t (0.00 sec) rows affected (0.00 sec)   StudentName	+ Address   Address   2727 W March Ln, Stockton, CA 95219   4600 Madison Ave, Sacramento, CA 95841   9188 E Stockton Blvd, Elk Grove, CA 956; 	seymourbutts@fake.com     amandahugginkiss@fake.com   24   oliverklozoff@@fake.com     eliverklozoff@@fake.com     Email     seymourbutts@fake.com

Figure 9: GetStudentCourse, returns student info in the course. 7 Courses to call but only shown is 4 courses.

SSN	StudentName	Address		Email	
115505525   543216789   678954321	Mike Rotch Hugh Jass Amanda Hugginkis	2625 W Kettleman Ln, Lodi, CA 95242   2001 Alta Arden Exwy., Sacramento, CA 95825   4600 Madison Ave, Sacramento, CA 95841		mikerotch@fake.com hughjass@fake.com amandahugginkiss@fak	e.co
 3 rows in set	(0.00 sec)	+	+		
Query OK, 0 r	ows affected (0.0	0 sec)			
·	+		+		
SSN	StudentName	Address	Emai	il į	
115505525   214407735   987654321	Mike Rotch   Seymour Butz   Oliver Klozoff	2625 W Kettleman Ln, Lodi, CA 95242 2727 W March Ln, Stockton, CA 95219 9188 E Stockton Blvd, Elk Grove, CA 95624	seyn	mikerotch@fake.com   seymourbutts@fake.com   oliverklozoff@@fake.com	
 3 rows in set	+ : (0.00 sec)		+	+	
Query OK, 0 r	ows affected (0.0	0 sec)			
SSN	StudentName	Address	İ	Email	
123456789   543216789   678954321	Moe Ron Hugh Jass Amanda Hugginkis	16514 Golden Valley Pkwy, Lathrop, CA 9   2001 Alta Arden Exwy., Sacramento, CA 9   4600 Madison Ave, Sacramento, CA 95841		moeron@fake.com hughjass@fake.com amandahugginkiss@fak	e.co
rows in set	(0.00 sec)	+	+	+	
Ouerv OK. 0 r	ows affected (0.0	9 sec)			

Figure 10: The remainder of the courses not shown in 9.

- Now we can run the 'source CSC174/hw3/6\_a\_trigger.sql'. More on this topic above.
- Last is to drop everything we created. To do this run 'source CSC174/hw3/7 drop all.sql'

```
mysql> source CSC174/hw3/7_drop_all.sql
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, O rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, O rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, O rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
mysql> ■
```

Figure 11: Dropping everything.