SQL Project 2 – Triggers Solution

EECS 495 – Intro to Database Systems

TA: Muhammed Mas-ud Hussain

Q1 Solution:

For insert:

```
CREATE TRIGGER Q11 BEFORE INSERT ON made_money
FOR EACH ROW
        IF NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') THEN
                SET NEW.category ='Action';
        END IF;
```

For update:

```
CREATE TRIGGER Q12 BEFORE UPDATE ON made_money
FOR EACH ROW
BEGIN
        IF NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') THEN
                SET NEW.category ='Action';
        END IF;
END
```

Q2 Solution:

(Although ideally an update trigger should be written as well, but not required…(no marks deduction))

```
CREATE TRIGGER Q2 BEFORE INSERT ON appeared_in
FOR EACH ROW
BEGIN
        IF EXISTS(SELECT * FROM made_money WHERE movie=NEW.movie and Category='Comedy')
AND NOT EXISTS (SELECT * FROM appeared_in a, made_money b WHERE a.Movie = b.Movie AND a.Star
= NEW.Star AND b.category in ('Comedy', 'Drama', 'Romantic')) THEN
      UPDATE made_money SET category='Drama' WHERE movie=NEW.Movie;
        END IF;
END
```

Q3 Solution:

(Although ideally an update trigger should be written as well, but not required…(no marks deduction))

```
CREATE TRIGGER Q3 BEFORE INSERT ON married

FOR EACH ROW

BEGIN

   DECLARE done INT DEFAULT FALSE;

   DECLARE mday DATE;

   DECLARE dday DATE;

   DECLARE c_num INT;

   DECLARE cur1 CURSOR FOR SELECT b.couple_num, b.day FROM married b, in_couple c
WHERE b.couple_num = c.couple_num AND c.star IN (SELECT a.star From in_couple a WHERE
a.couple_num = 1);

   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;


   OPEN cur1;

   read_loop: LOOP

     FETCH cur1 INTO c_num, mday;


     IF done THEN

        LEAVE read_loop;

            END IF;


     IF EXISTS (SELECT day FROM divorced WHERE couple_num = c_num) THEN

                        SET dday = (SELECT day FROM divorced WHERE couple_num = c_num);

     ELSE

         SET dday = '2030-01-01';

     END IF;


     IF NEW.day between mday and dday THEN

                              SIGNAL SQLSTATE '12347'

                                       SET MESSAGE_TEXT = 'Check constraint on Married.DAY failed';
```

```
    END IF;
  END LOOP;
  CLOSE cur1;
END //
```

Q4 Solution:

(Although ideally an update trigger should be written as well, but not required…(no marks deduction))

```
CREATE TRIGGER Q4 BEFORE INSERT ON made_money
FOR EACH ROW
        IF NEW.how_much <1000 OR NEW.how_much >3000000000 THEN
      SIGNAL SQLSTATE '12345'
          SET MESSAGE_TEXT = 'check constraint on Made_Money.HOW_MUCH failed';
       ELSEIF NEW.category='Action' AND NEW.how_much <10000 THEN
      SIGNAL SQLSTATE '12345'
          SET MESSAGE_TEXT = 'check constraint on Made_Money.HOW_MUCH failed';
       ELSEIF NEW.category='Comedy' AND NEW.how_much >1000000000 THEN
      SIGNAL SQLSTATE '12345'
          SET MESSAGE_TEXT = 'check constraint on Made_Money.HOW_MUCH failed';
       END IF;
```

Q5 Solution:

(Although ideally an update trigger should be written as well, but not required…(no marks deduction))

```
CREATE TRIGGER Q5 BEFORE INSERT ON divorced
FOR EACH ROW
BEGIN
        IF NOT EXISTS (SELECT * FROM MARRIED WHERE
COUPLE_NUM=NEW.COUPLE_NUM) THEN
      SIGNAL SQLSTATE '12346'
          SET MESSAGE_TEXT = 'The provided couple_num is not Married yet';
```

```
        ELSEIF NEW.day < (SELECT day FROM MARRIED WHERE
COUPLE_NUM=NEW.COUPLE_NUM) THEN

                SET New.day = (SELECT day FROM MARRIED WHERE
COUPLE_NUM=NEW.COUPLE_NUM);

        END IF;

END
```

Q6 Solution:

```
CREATE TRIGGER Q6 AFTER INSERT ON made_money

FOR EACH ROW

        INSERT INTO log_data VALUES(New.movie, New.category);
```

INSERT results:

a) Insert a new movie, with values ("IRON MAN", 1000000, 2008-05-02, "ACTON") in MADE_MONEY table.
Result: Successfully inserted --- but the category of the inserted movie "Iron Man" is changed to "Action".

b) Update the CATEGORY of the movie "Fight Club" to "Horror in MADE_MONEY table.
Result: Successfully updated --- but, instead of "Horror", the category of the updated movie Fight Club" is actually unchanged (remains "Action").

c) Insert a new tuple in APPEARED_IN table, with values ("Matt Damon", "Bruce Almighty").
Result: Successfully inserted --- but, the category of the movie "Bruce Almighty" is reset to "Drama" in made_money table.

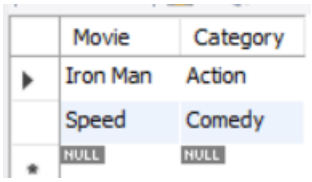
d) Insert a new tuple in MARRIED, with values (1, 2015-06-26).
Result: The insert is Unsuccessful. The error is: 'Check constraint on Married.DAY failed'.

e) Insert two new tuples in MADE_MONEY, having values ("Most Welcome", 8000, 2012-07-07, "Action") and ("Speed", 9000, 2010-03-28, "Comedy").
Result: The first insert is Unsuccessful. The error is: 'Check constraint on Made_Money.HOW_MUCH failed'. The second insert is Successfully inserted as it is, without any change in the given values.

f) Insert a new tuple in MADE_MONEY, having values ("Hangover", 1500000000, 2011-03-05, "Comedy").
Result: The insert is Unsuccessful. The error is:

g) Insert a new tuple in DIVORCED, with values (6, 2004-01-01).
Result: Successfully inserted --- but, the day of the newly inserted data is changed to (instead of '2004-01-01') the marriage day of the couple 6, which is 2005-06-25.

LOG Table state in the end:



	Movie	Category
▶	Iron Man	Action
	Speed	Comedy
*	NULL	NULL