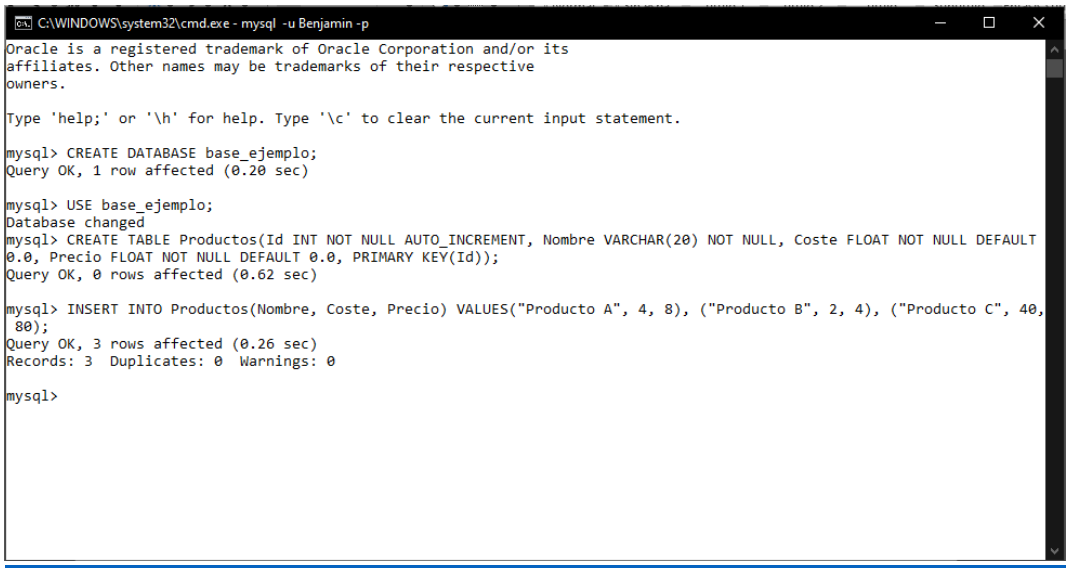
Práctica TRIGGERS

Realizar las practicas que se muestran en los siguientes enlaces:

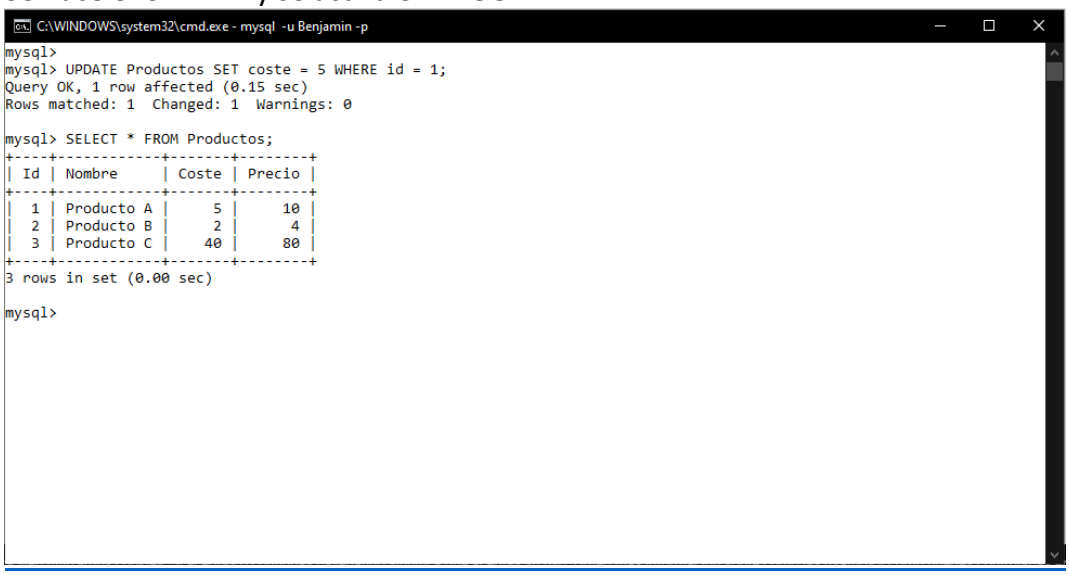
1. https://www.neoguias.com/como-crear-y-utilizar-triggers-en-mysql/

   Se crea la base de datos y se añade la tabla Productos que contendrá tres registros

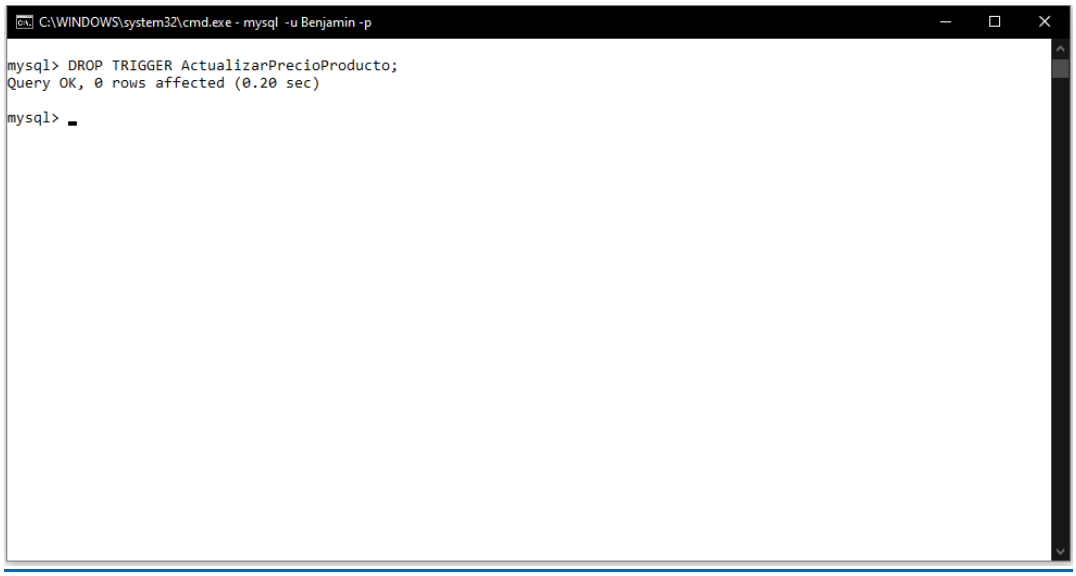


   Se crea el Trigger para antes de un UPDATE

Se hace el UPDATE y se activa el TRIGGER

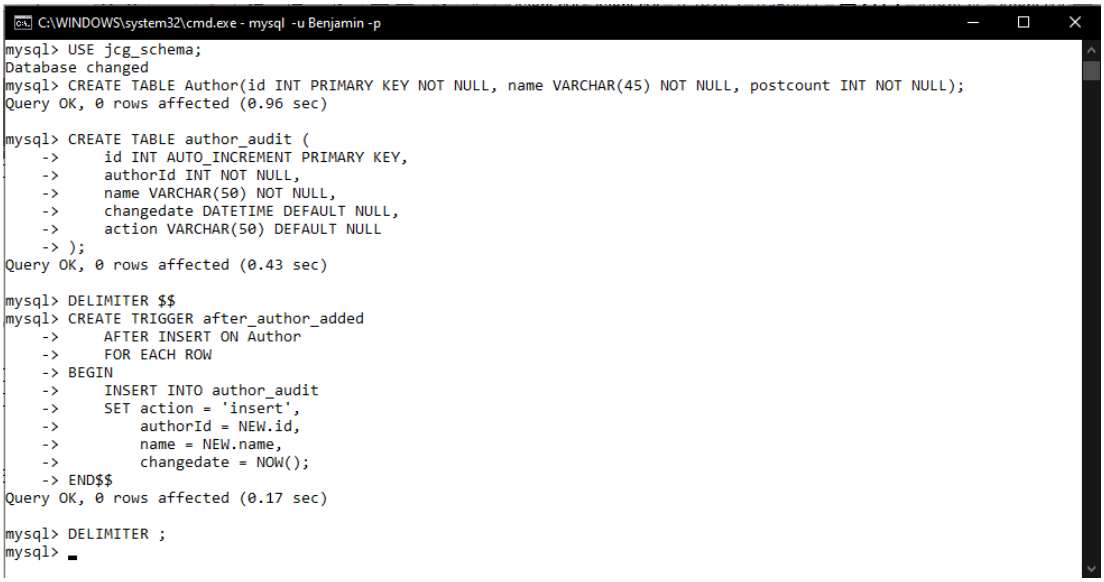


Se elimina el TRIGGER

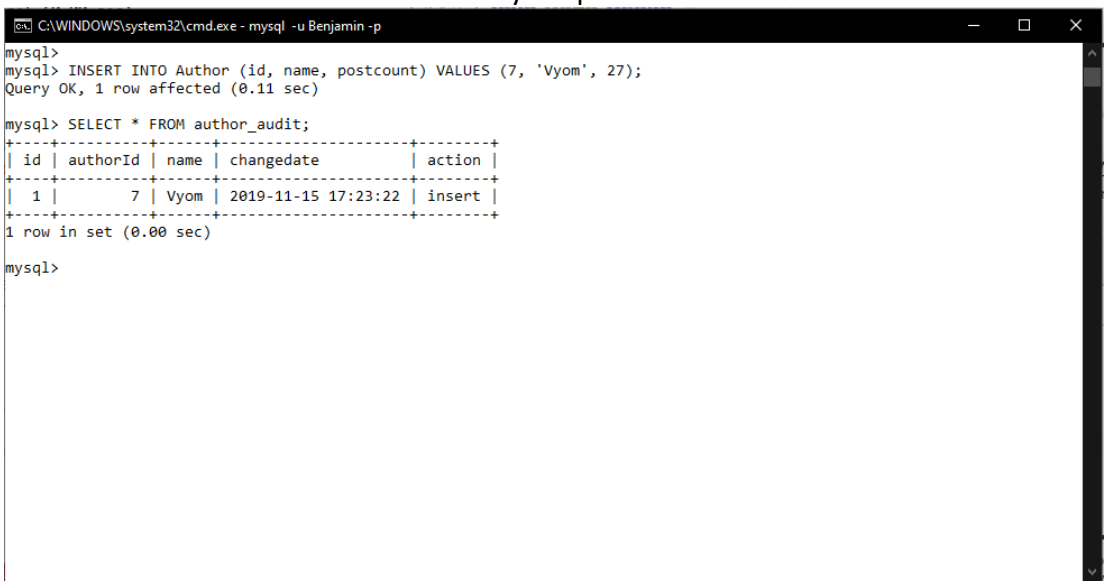


2. https://examples.javacodegeeks.com/core-java/sql/mysql-triggers-tutorial/

Se crea la base de datos y la tabla Autor. Se crea también el TRIGGER para después del INSERT en la tabla antes mencionada



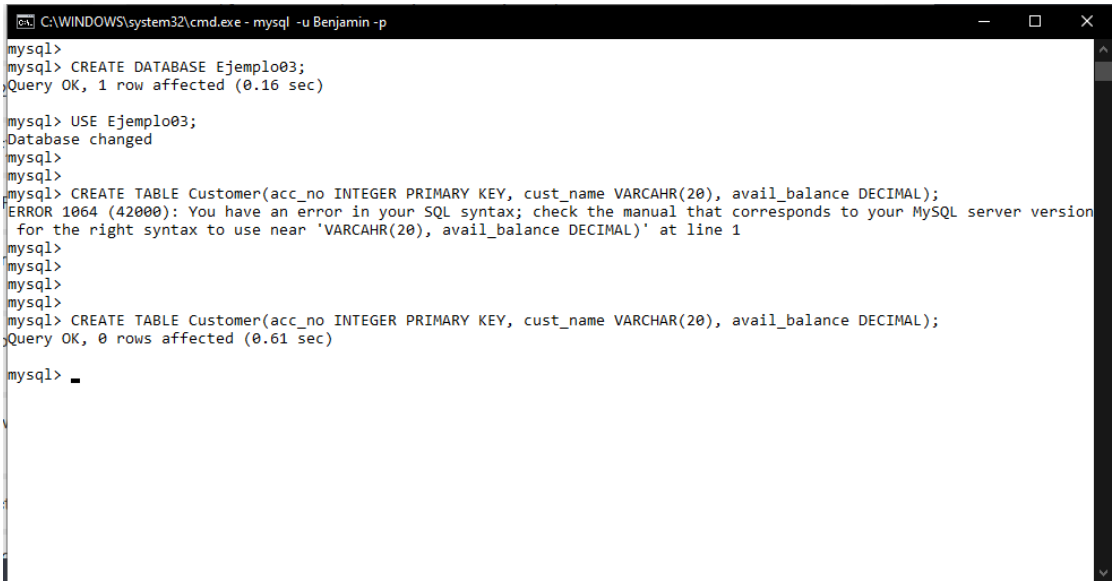
Se realiza la inserción en la tabla autor y después de esta se activa el TRIGGER.

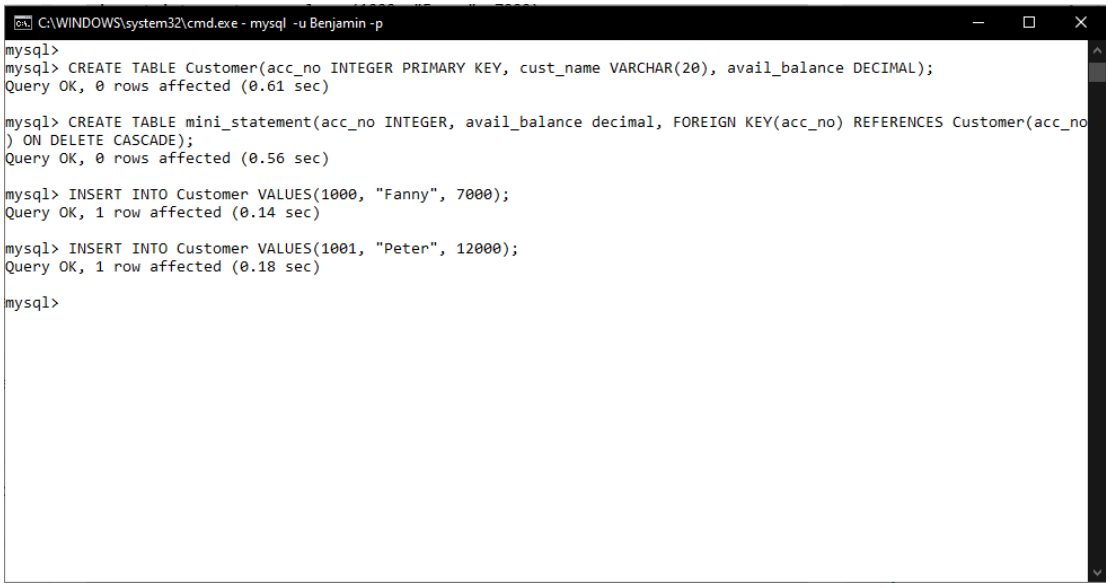


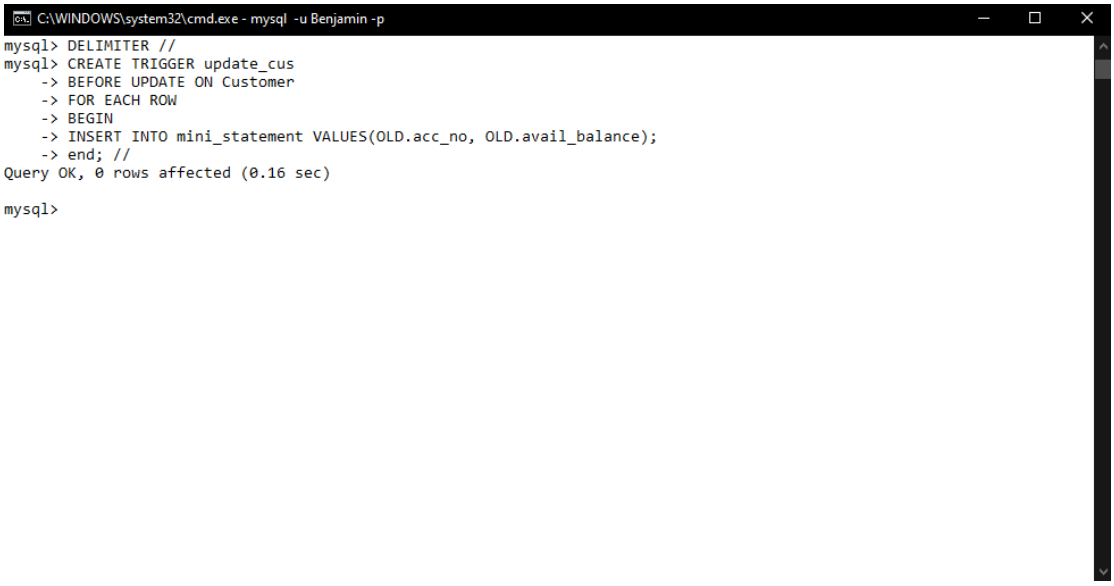
Se elimina el TRIGGER

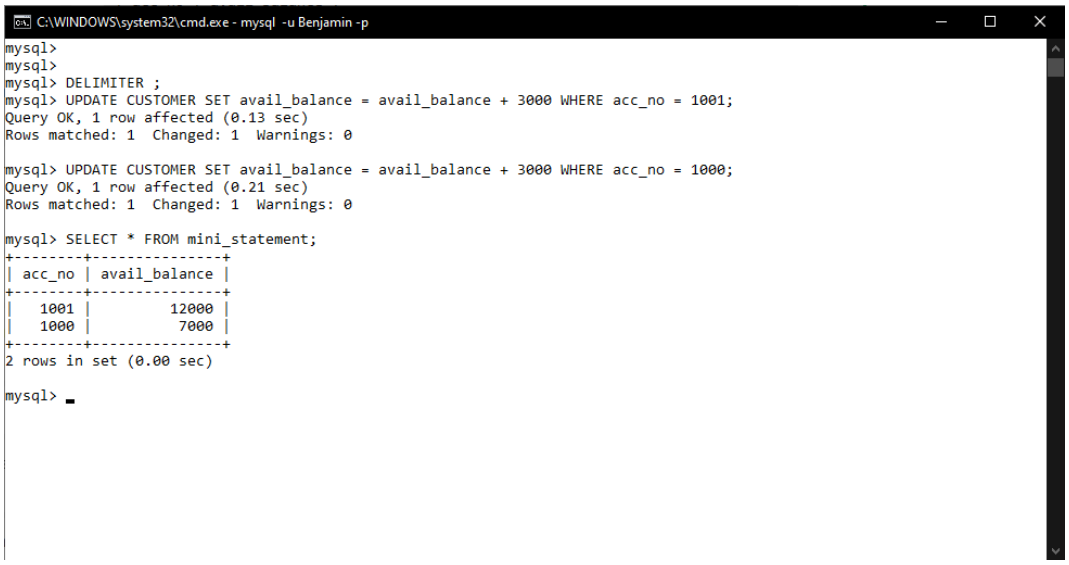
3. https://www.geeksforgeeks.org/different-types-of-mysql-triggers-with-examples/

Before Update Trigger:

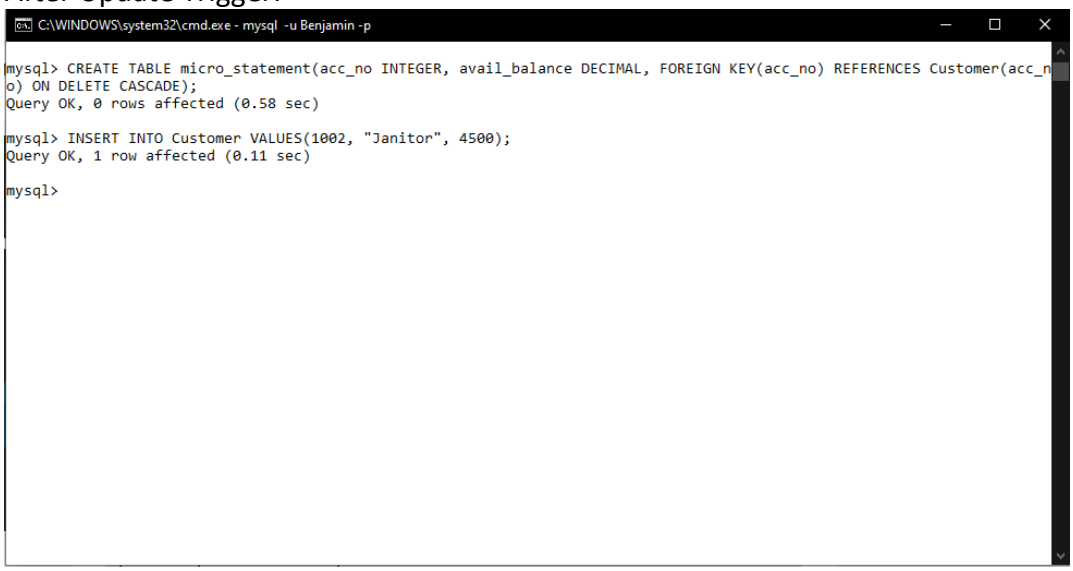


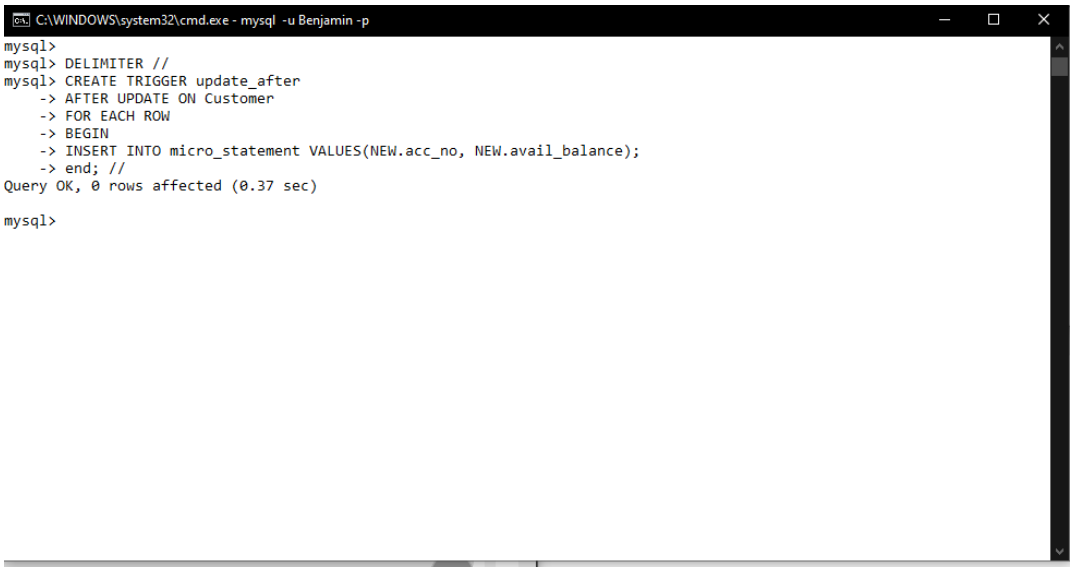

C:\WINDOWS\system32\cmd.exe - mysql -u Benjamin -p
mysql>
mysql> CREATE TABLE Customer(acc_no INTEGER PRIMARY KEY, cust_name VARCHAR(20), avail_balance DECIMAL);
Query OK, 0 rows affected (0.61 sec)
mysql> CREATE TABLE mini_statement(acc_no INTEGER, avail_balance decimal, FOREIGN KEY(acc_no) REFERENCES Customer(acc_no
) ON DELETE CASCADE);
Query OK, 0 rows affected (0.56 sec)
mysql> INSERT INTO Customer VALUES(1000, "Fanny", 7000);
Query OK, 1 row affected (0.14 sec)
mysql> INSERT INTO Customer VALUES(1001, "Peter", 12000);
Query OK, 1 row affected (0.18 sec)
mysql>

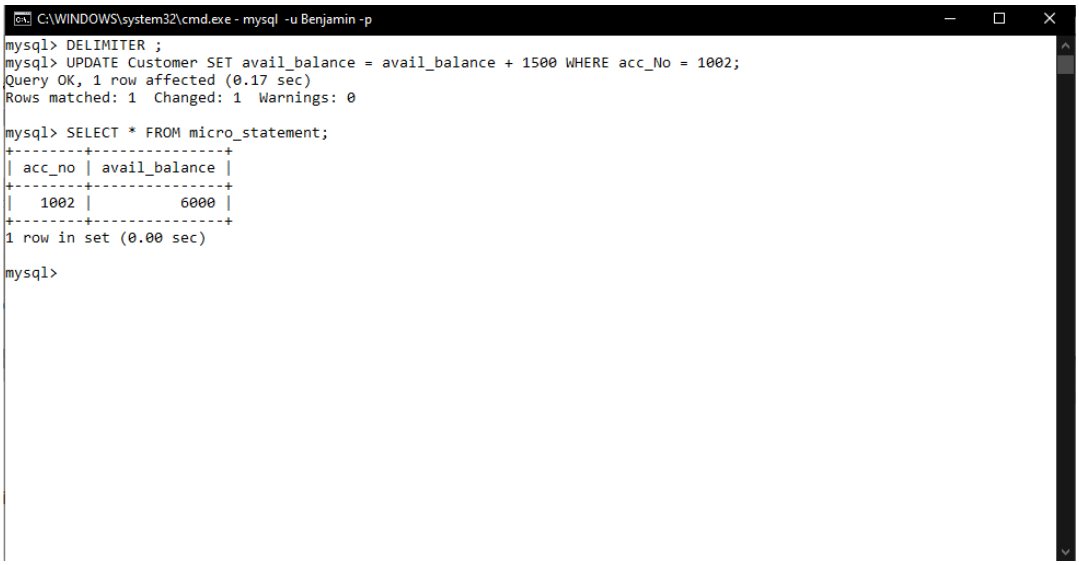

C:\WINDOWS\system32\cmd.exe - mysql -u Benjamin -p
mysql> DELIMITER //
mysql> CREATE TRIGGER update_cus
-> BEFORE UPDATE ON Customer
-> FOR EACH ROW
-> BEGIN
-> INSERT INTO mini_statement VALUES(OLD.acc_no, OLD.avail_balance);
-> end; //
Query OK, 0 rows affected (0.16 sec)
mysql>



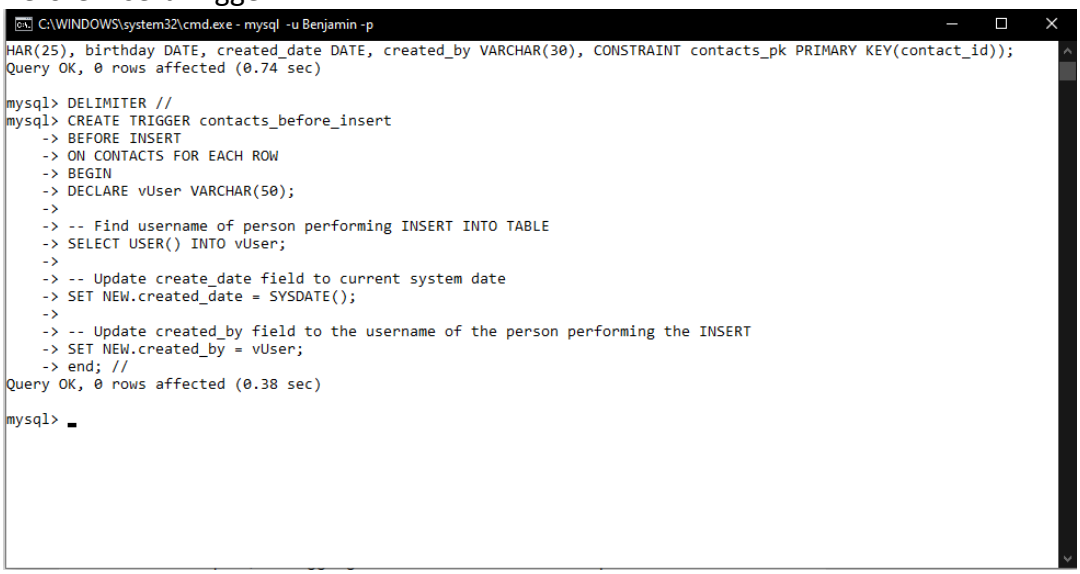
After Update Trigger:

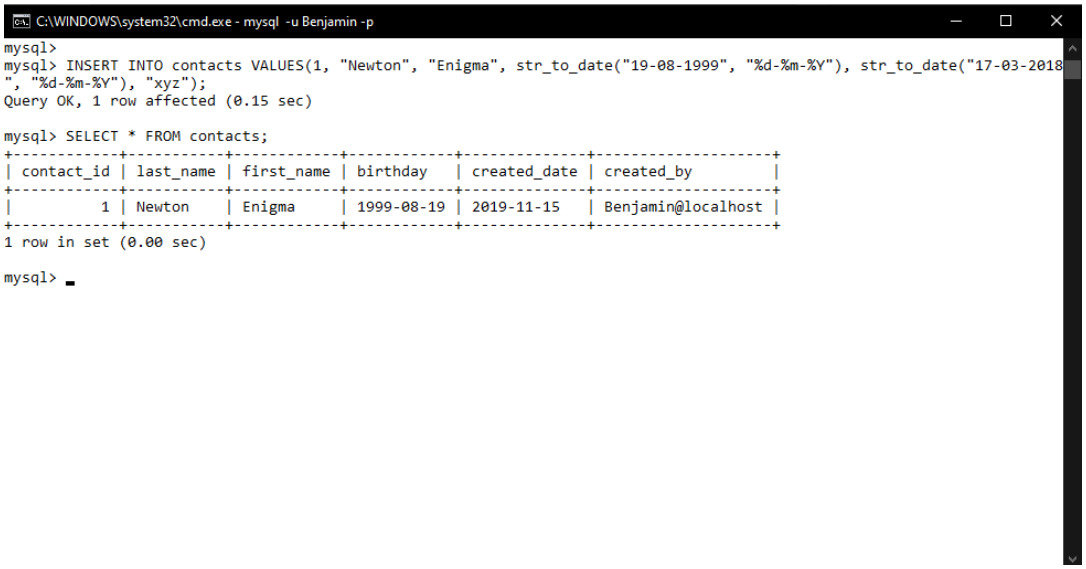



C:\WINDOWS\system32\cmd.exe - mysql -u Benjamin -p
mysql>
mysql> DELIMITER //
mysql> CREATE TRIGGER update_after
-> AFTER UPDATE ON Customer
-> FOR EACH ROW
-> BEGIN
-> INSERT INTO micro_statement VALUES(NEW.acc_no, NEW.avail_balance);
-> end; //
Query OK, 0 rows affected (0.37 sec)
mysql>

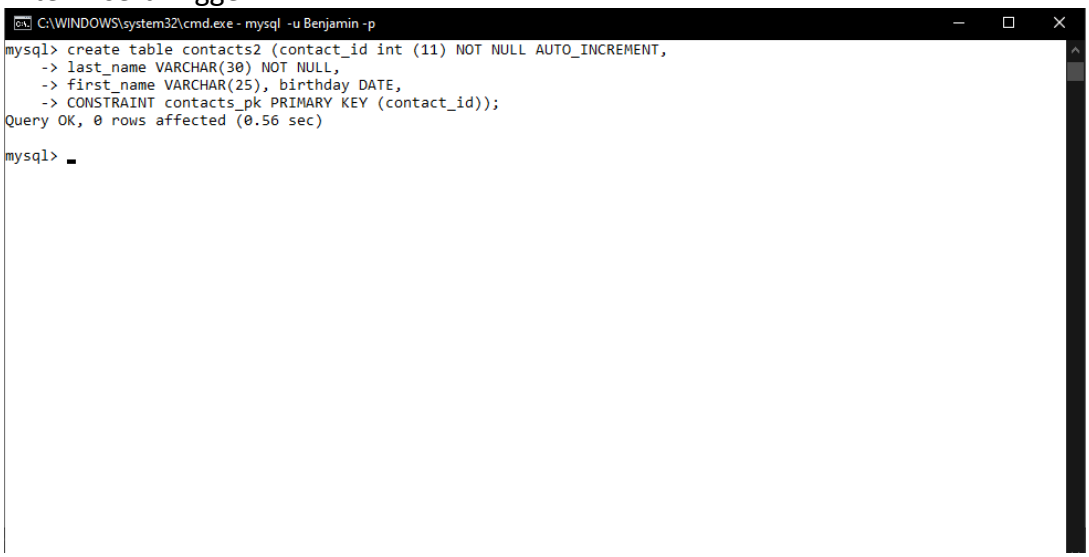

C:\WINDOWS\system32\cmd.exe - mysql -u Benjamin -p
mysql> DELIMITER ;
mysql> UPDATE Customer SET avail_balance = avail_balance + 1500 WHERE acc_No = 1002;
Query OK, 1 row affected (0.17 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> SELECT * FROM micro_statement;
+--------+---------------+
| acc_no | avail_balance |
+--------+---------------+
| 1002 | 6000 |
+--------+---------------+
1 row in set (0.00 sec)
mysql>

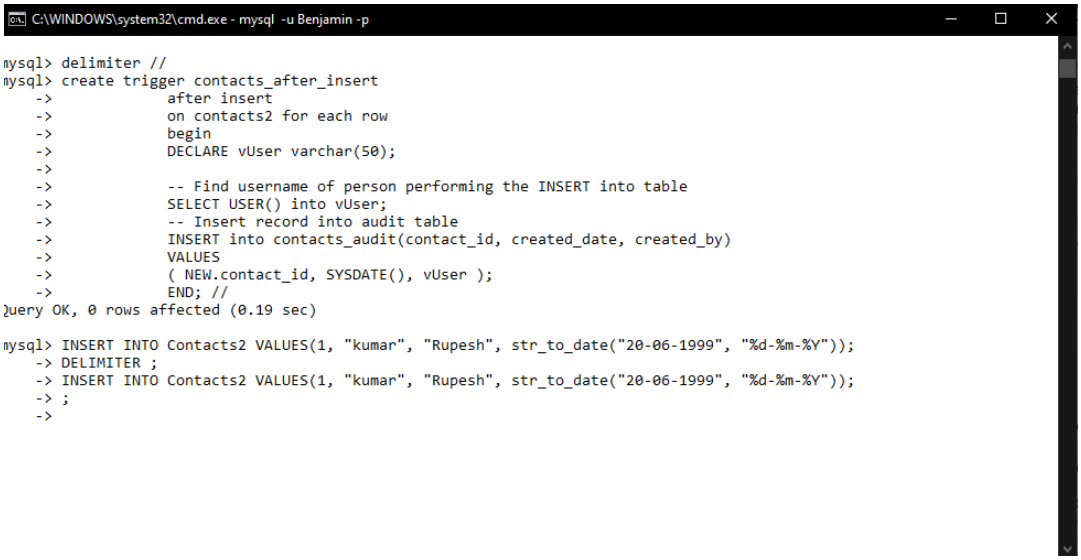
Before Insert Trigger:

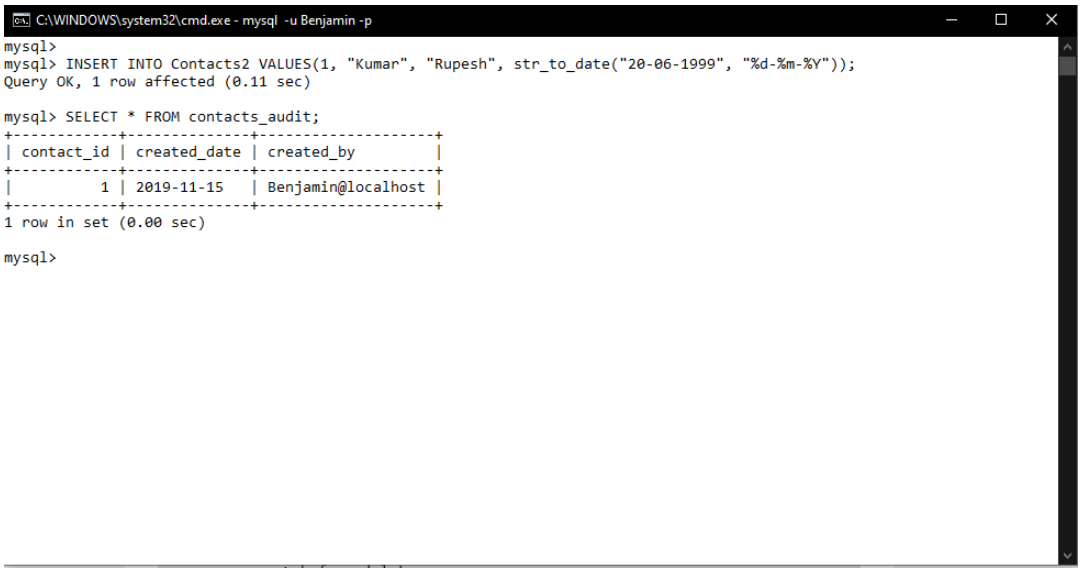




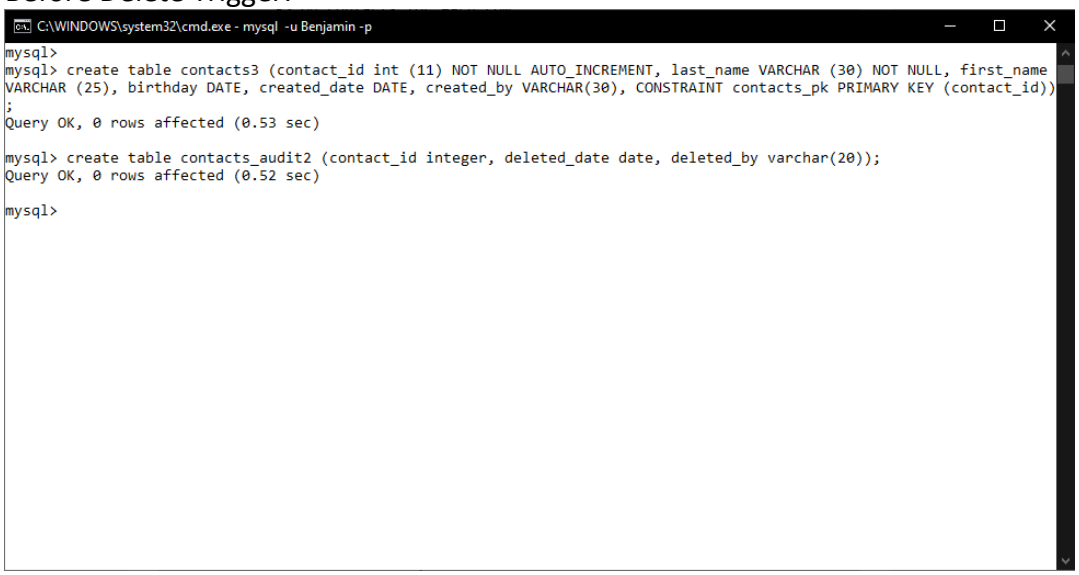
After Insert Trigger:

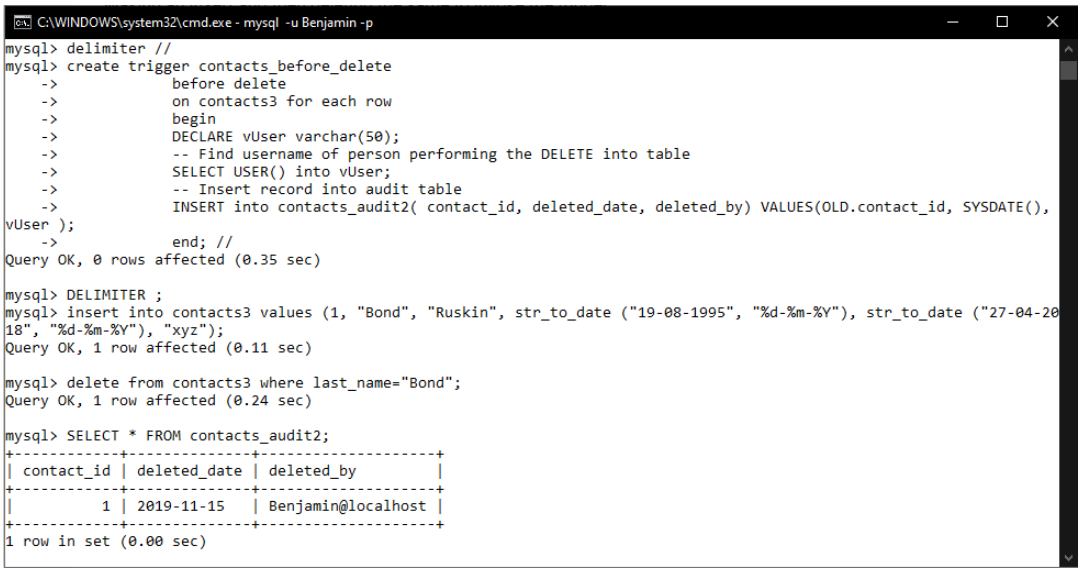




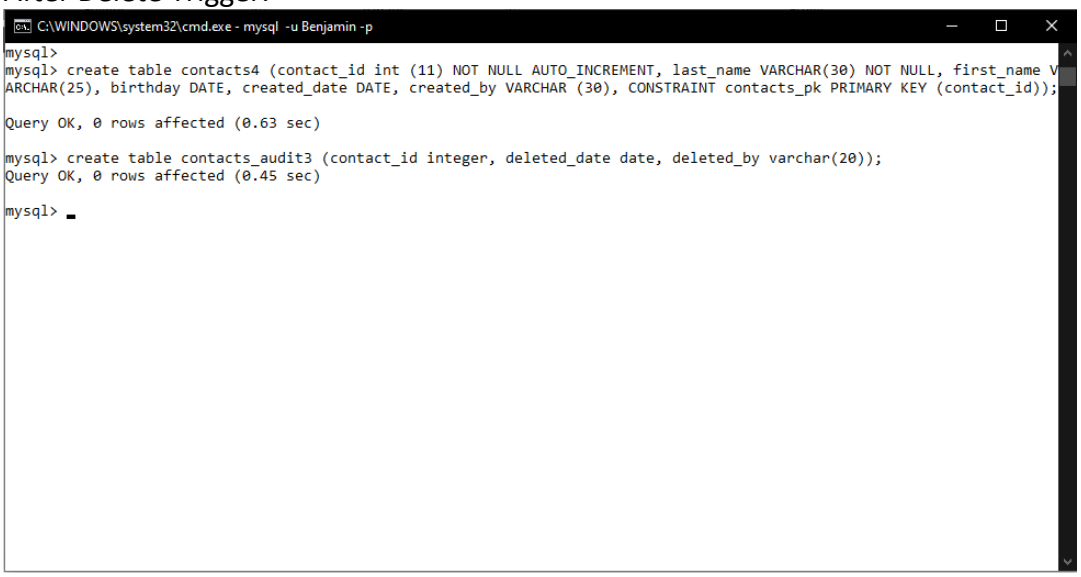


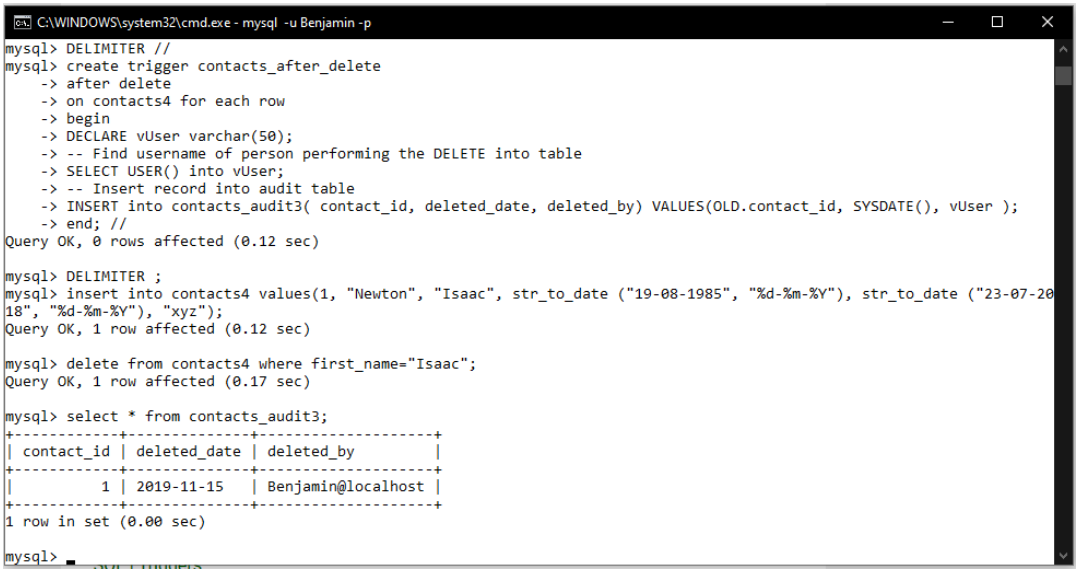
Before Delete Trigger:





After Delete Trigger:




C:\WINDOWS\system32\cmd.exe - mysql -u Benjamin -p
mysql> DELIMITER //
mysql> create trigger contacts_after_delete
-> after delete
-> on contacts4 for each row
-> begin
-> DECLARE vUser varchar(50);
-> -- Find username of person performing the DELETE into table
-> SELECT USER() into vUser;
-> -- Insert record into audit table
-> INSERT into contacts_audit3(contact_id, deleted_date, deleted_by) VALUES(OLD.contact_id, SYSDATE(), vUser);
-> end; //
Query OK, 0 rows affected (0.12 sec)
mysql> DELIMITER ;
mysql> insert into contacts4 values(1, "Newton", "Isaac", str_to_date ("19-08-1985", "%d-%m-%Y"), str_to_date ("23-07-20
18", "%d-%m-%Y"), "xyz");
Query OK, 1 row affected (0.12 sec)
mysql> delete from contacts4 where first_name="Isaac";
Query OK, 1 row affected (0.17 sec)
mysql> select * from contacts_audit3;
+------------+--------------+--------------------+
| contact_id | deleted_date | deleted_by |
+------------+--------------+--------------------+
| 1 | 2019-11-15 | Benjamin@localhost |
+------------+--------------+--------------------+
1 row in set (0.00 sec)
mysql>