INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

15 de noviembre del 2019
Jerez, Zac
Ingeniería en sistemas computacionales
Semestre: 5

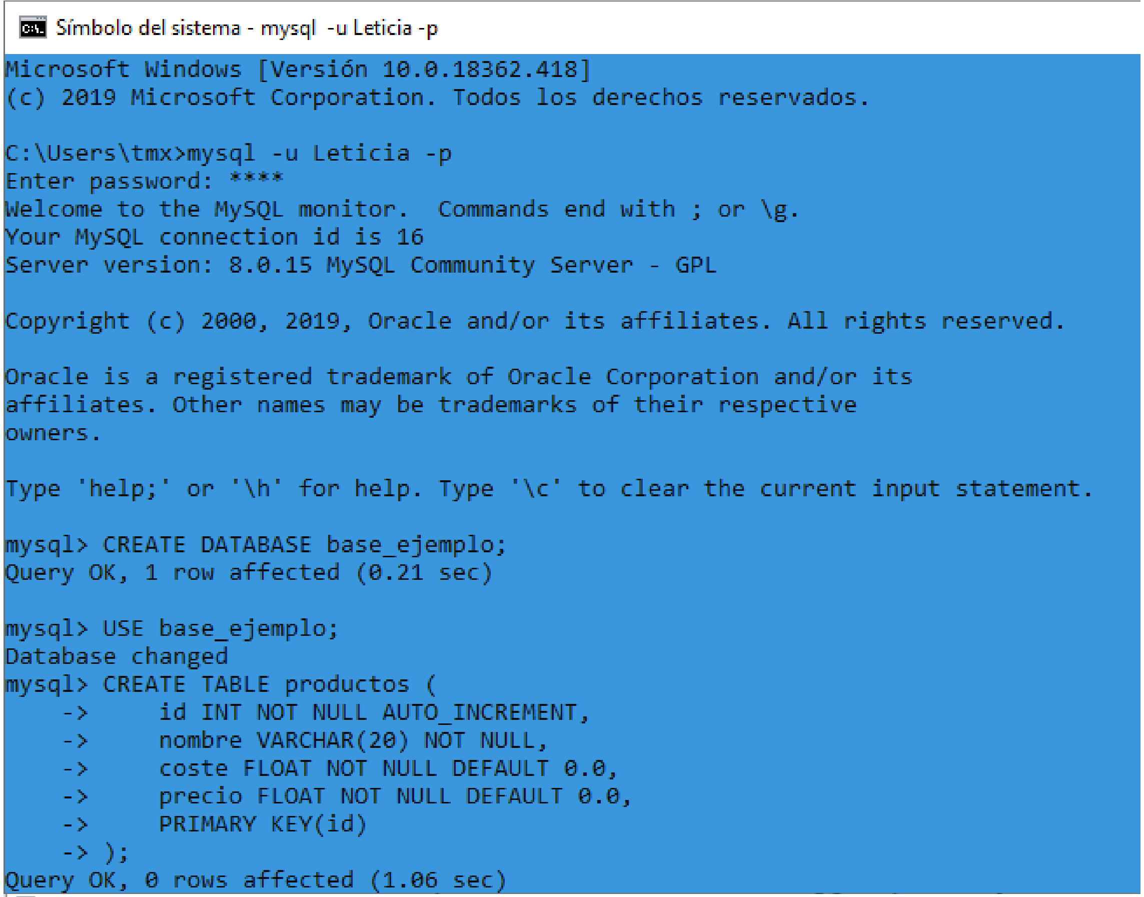
Unidad 5: SQL procedural
Alumna: Leticia carrera venegas
Correo: Letycv25@gmail.com
Num control:S17070155

Actividad: Triggers
Taller de bases de datos
Docente: ISC Salvador Acevedo Sandoval

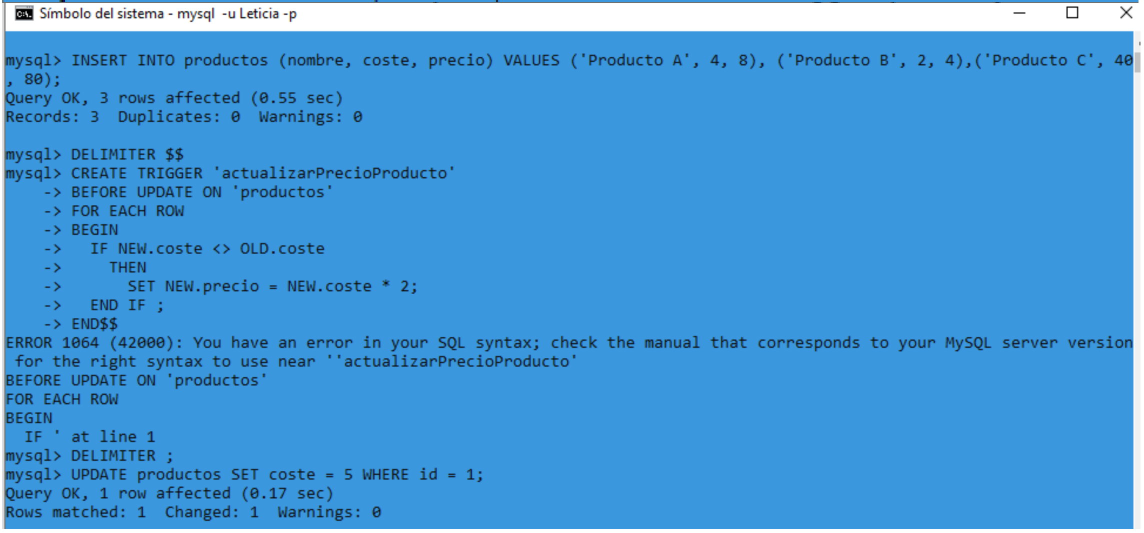
Practica TRIGGERS

Realizar las prácticas que se muestran en lo siguientes enlaces:

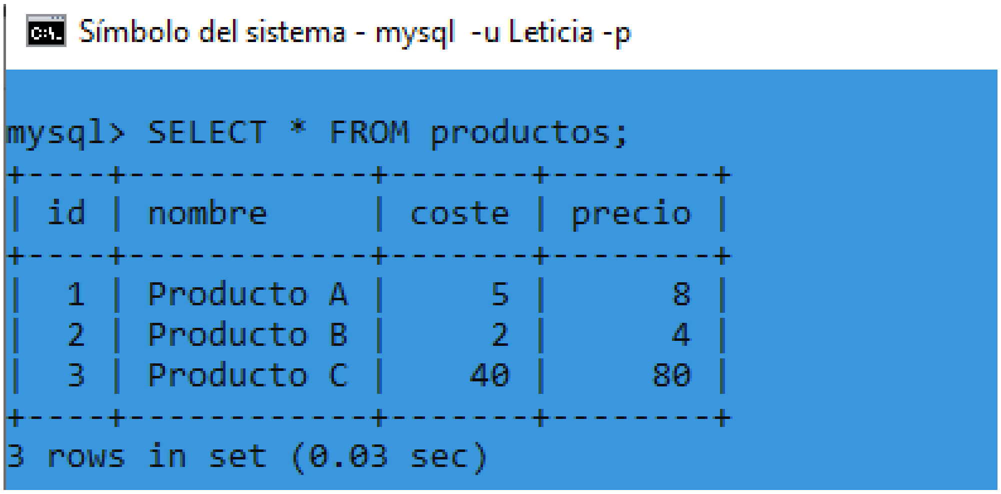
1. https://www.neoguias.com/como-crear-y-utilizar-triggers-en-mysql/



```
Microsoft Windows [Versión 10.0.18362.418]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\tmx>mysql -u Leticia -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.15 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE base_ejemplo;
Query OK, 1 row affected (0.21 sec)

mysql> USE base_ejemplo;
Database changed
mysql> CREATE TABLE productos (
    ->     id INT NOT NULL AUTO_INCREMENT,
    ->     nombre VARCHAR(20) NOT NULL,
    ->     coste FLOAT NOT NULL DEFAULT 0.0,
    ->     precio FLOAT NOT NULL DEFAULT 0.0,
    ->     PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (1.06 sec)
```

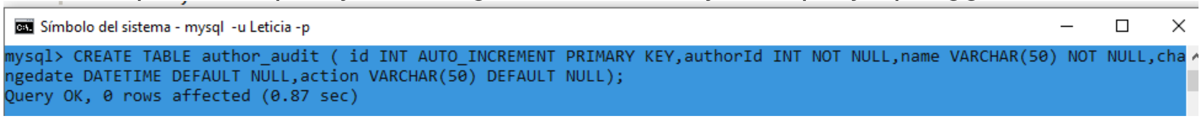


```
mysql> INSERT INTO productos (nombre, coste, precio) VALUES ('Producto A', 4, 8), ('Producto B', 2, 4),('Producto C', 40
, 80);
Query OK, 3 rows affected (0.55 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DELIMITER $$
mysql> CREATE TRIGGER 'actualizarPrecioProducto'
    -> BEFORE UPDATE ON 'productos'
    -> FOR EACH ROW
    -> BEGIN
    ->   IF NEW.coste <> OLD.coste
    ->     THEN
    ->       SET NEW.precio = NEW.coste * 2;
    ->   END IF ;
    -> END$$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
 for the right syntax to use near ''actualizarPrecioProducto'
BEFORE UPDATE ON 'productos'
FOR EACH ROW
BEGIN
  IF ' at line 1
mysql> DELIMITER ;
mysql> UPDATE productos SET coste = 5 WHERE id = 1;
Query OK, 1 row affected (0.17 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

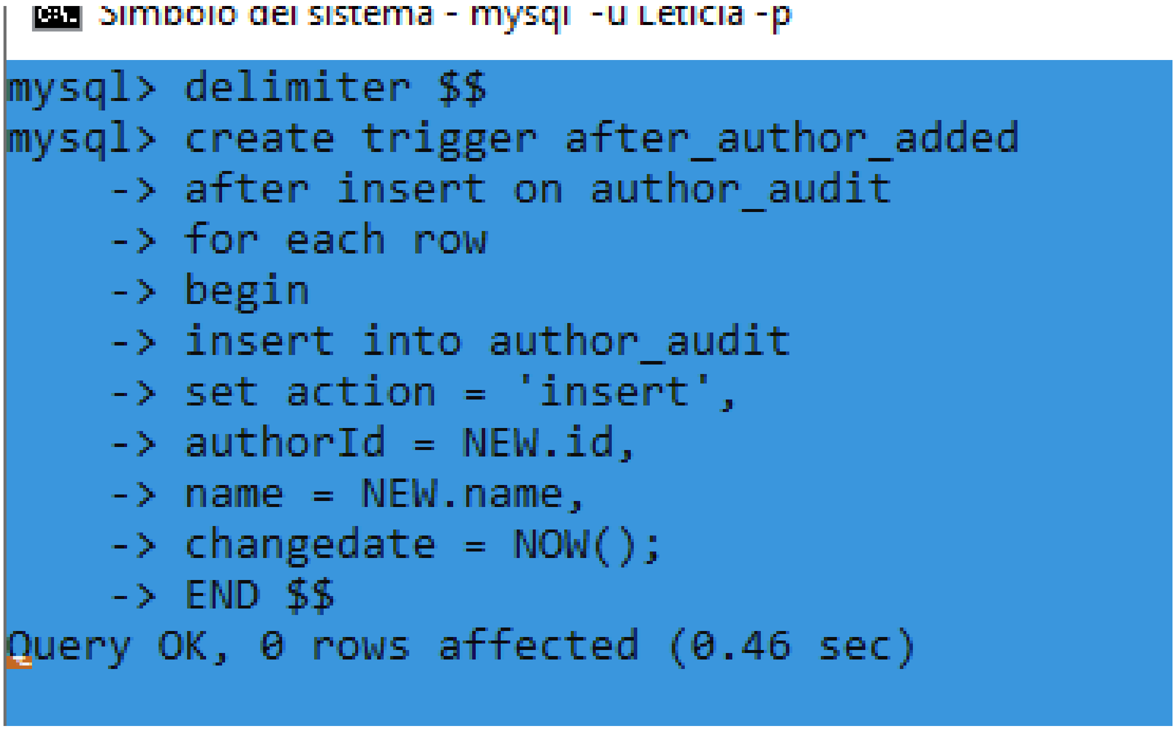


```
mysql> SELECT * FROM productos;
+----+------------+-------+--------+
| id | nombre     | coste | precio |
+----+------------+-------+--------+
|  1 | Producto A |     5 |      8 |
|  2 | Producto B |     2 |      4 |
|  3 | Producto C |    40 |     80 |
+----+------------+-------+--------+
3 rows in set (0.03 sec)
```

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



```
mysql> CREATE TABLE author_audit ( id INT AUTO_INCREMENT PRIMARY KEY,authorId INT NOT NULL,name VARCHAR(50) NOT NULL,cha
ngedate DATETIME DEFAULT NULL,action VARCHAR(50) DEFAULT NULL);
Query OK, 0 rows affected (0.87 sec)
```



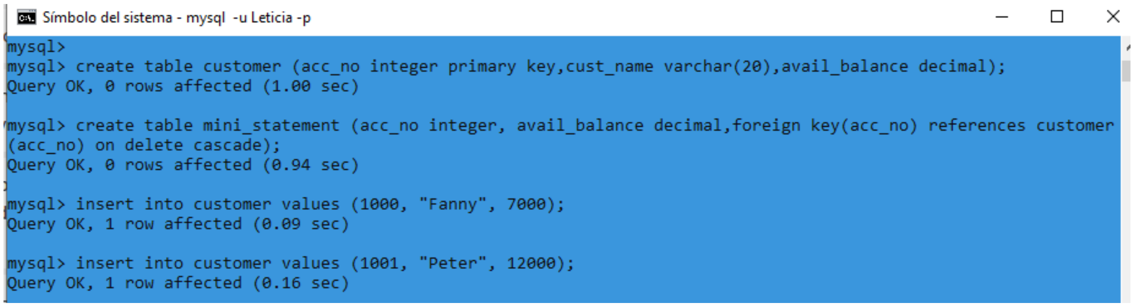
```
mysql> delimiter $$
mysql> create trigger after_author_added
    -> after insert on author_audit
    -> for each row
    -> begin
    -> insert into author_audit
    -> set action = 'insert',
    -> authorId = NEW.id,
    -> name = NEW.name,
    -> changedate = NOW();
    -> END $$
Query OK, 0 rows affected (0.46 sec)
```

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

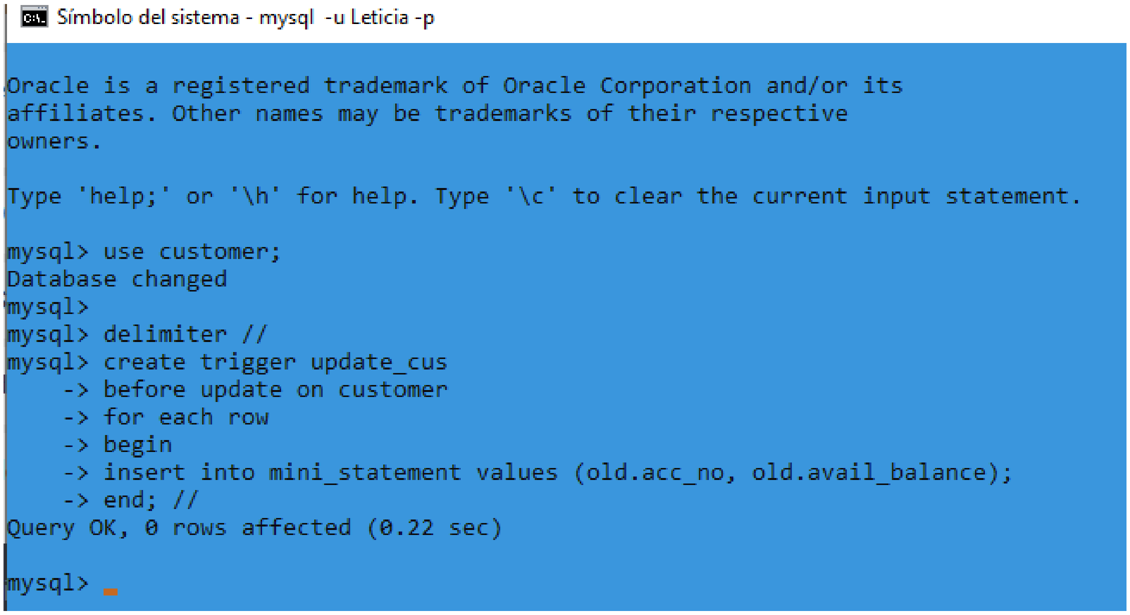
1. **Before Update Trigger:**
   As the name implies, it is a trigger which enacts before an update is

invoked. If we write an update statement, then the actions of the trigger will be performed before the update is implemented.

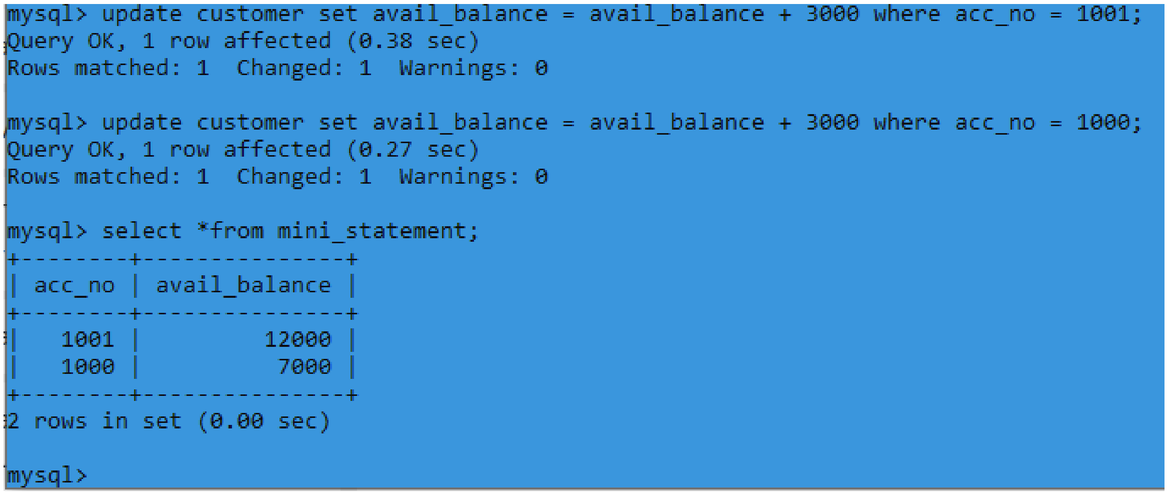
Example:



```
mysql>
mysql> create table customer (acc_no integer primary key,cust_name varchar(20),avail_balance decimal);
Query OK, 0 rows affected (1.00 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 (0.94 sec)

mysql> insert into customer values (1000, "Fanny", 7000);
Query OK, 1 row affected (0.09 sec)

mysql> insert into customer values (1001, "Peter", 12000);
Query OK, 1 row affected (0.16 sec)
```



```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use customer;
Database changed
mysql>
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 (0.22 sec)

mysql>
```

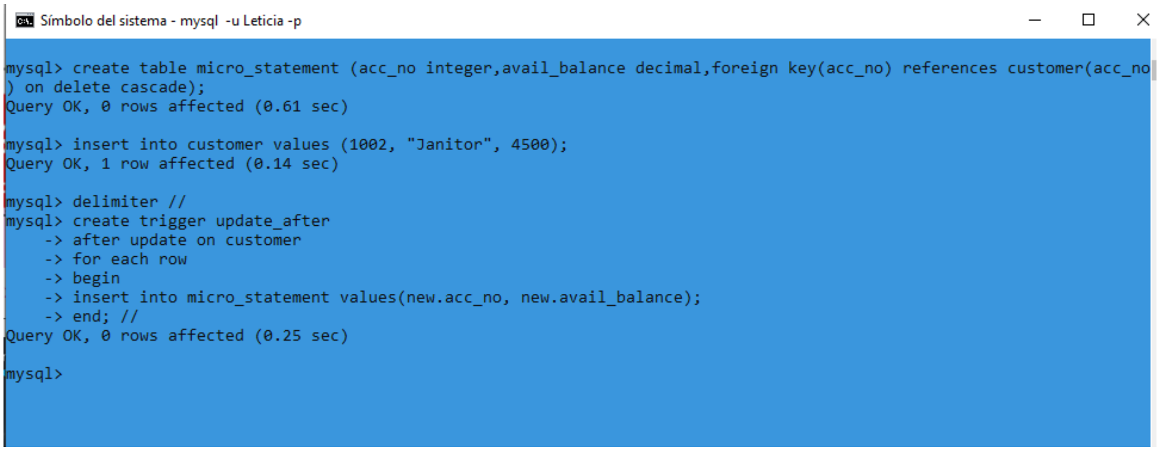


```
mysql> update customer set avail_balance = avail_balance + 3000 where acc_no = 1001;
Query OK, 1 row affected (0.38 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update customer set avail_balance = avail_balance + 3000 where acc_no = 1000;
Query OK, 1 row affected (0.27 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select *from mini_statement;
+--------+---------------+
| acc_no | avail_balance |
+--------+---------------+
|   1001 |         12000 |
|   1000 |          7000 |
+--------+---------------+
2 rows in set (0.00 sec)

mysql>
```

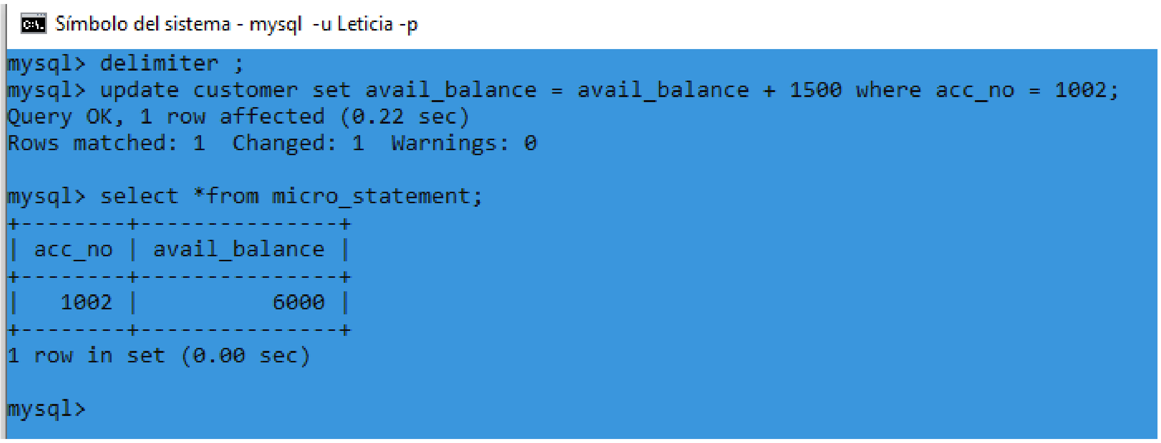
2. After Update Trigger:

As the name implies, this trigger is invoked after an updation occurs. (i.e., it gets implemented after an update statement is executed.).

Example:



```
mysql> create table micro_statement (acc_no integer,avail_balance decimal,foreign key(acc_no) references customer(acc_no
) on delete cascade);
Query OK, 0 rows affected (0.61 sec)

mysql> insert into customer values (1002, "Janitor", 4500);
Query OK, 1 row affected (0.14 sec)

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 (0.25 sec)

mysql>
```

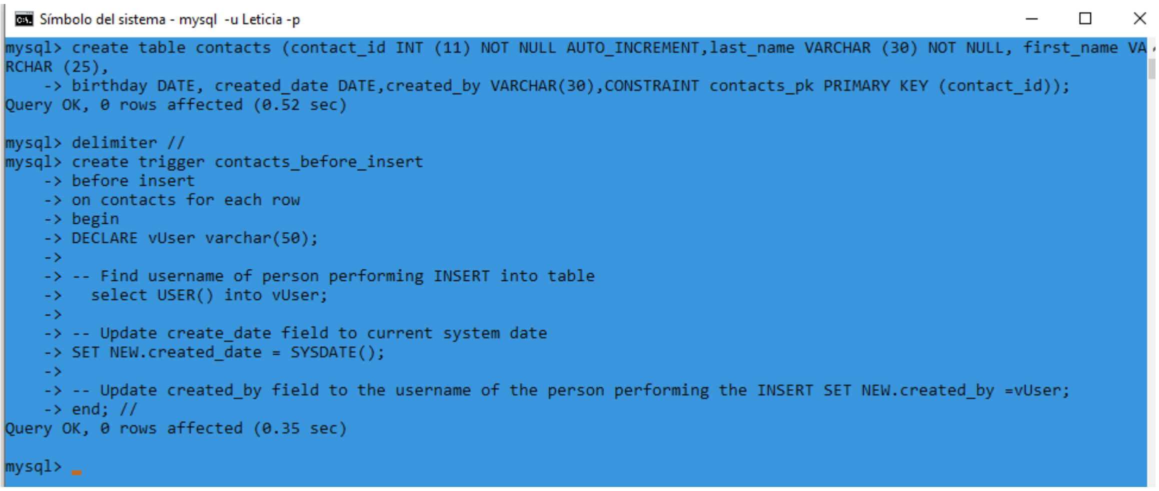


```
mysql> delimiter ;
mysql> update customer set avail_balance = avail_balance + 1500 where acc_no = 1002;
Query OK, 1 row affected (0.22 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select *from micro_statement;
+--------+---------------+
| acc_no | avail_balance |
+--------+---------------+
|   1002 |          6000 |
+--------+---------------+
1 row in set (0.00 sec)

mysql>
```

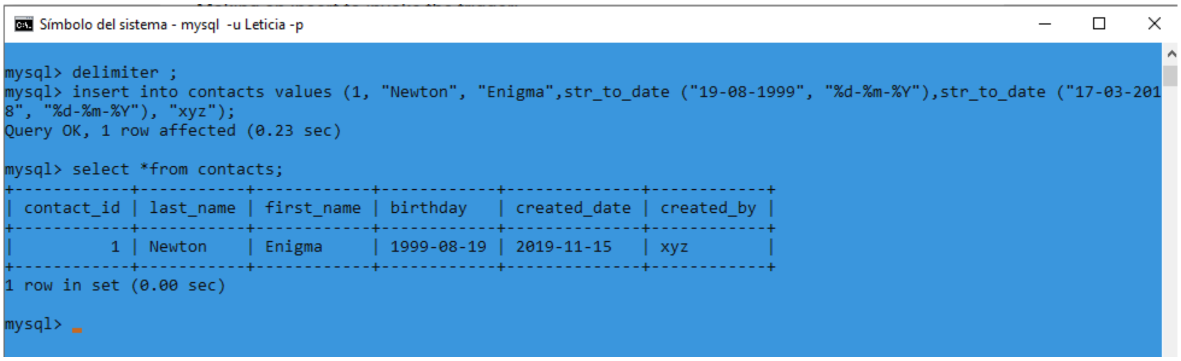
3. Before Insert Trigger:

As the name implies, this trigger is invoked before an insert, or before an insert statement is executed.

Example:



```
mysql> create table contacts (contact_id INT (11) NOT NULL AUTO_INCREMENT,last_name VARCHAR (30) NOT NULL, first_name VA
RCHAR (25),
    -> birthday DATE, created_date DATE,created_by VARCHAR(30),CONSTRAINT contacts_pk PRIMARY KEY (contact_id));
Query OK, 0 rows affected (0.52 sec)

mysql> delimiter //
mysql> create trigger contacts_before_insert
    -> before insert
    -> on contacts for each row
    -> begin
    -> DECLARE vUser varchar(50);
    ->
    -> -- Find username of person performing INSERT into table
    ->   select USER() into vUser;
    ->
    -> -- Update create_date field to current system date
    -> SET NEW.created_date = SYSDATE();
    ->
    -> -- Update created_by field to the username of the person performing the INSERT SET NEW.created_by =vUser;
    -> end; //
Query OK, 0 rows affected (0.35 sec)

mysql> _
```

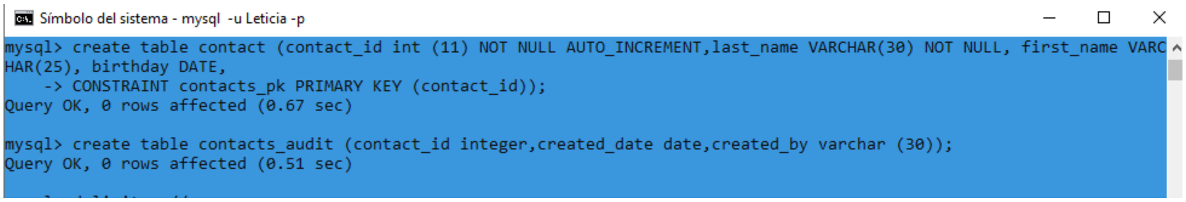


```
mysql> delimiter ;
mysql> insert into contacts values (1, "Newton", "Enigma",str_to_date ("19-08-1999", "%d-%m-%Y"),str_to_date ("17-03-201
8", "%d-%m-%Y"), "xyz");
Query OK, 1 row affected (0.23 sec)

mysql> select *from contacts;
+------------+-----------+------------+------------+--------------+------------+
| contact_id | last_name | first_name | birthday   | created_date | created_by |
+------------+-----------+------------+------------+--------------+------------+
|          1 | Newton    | Enigma     | 1999-08-19 | 2019-11-15   | xyz        |
+------------+-----------+------------+------------+--------------+------------+
1 row in set (0.00 sec)

mysql> _
```

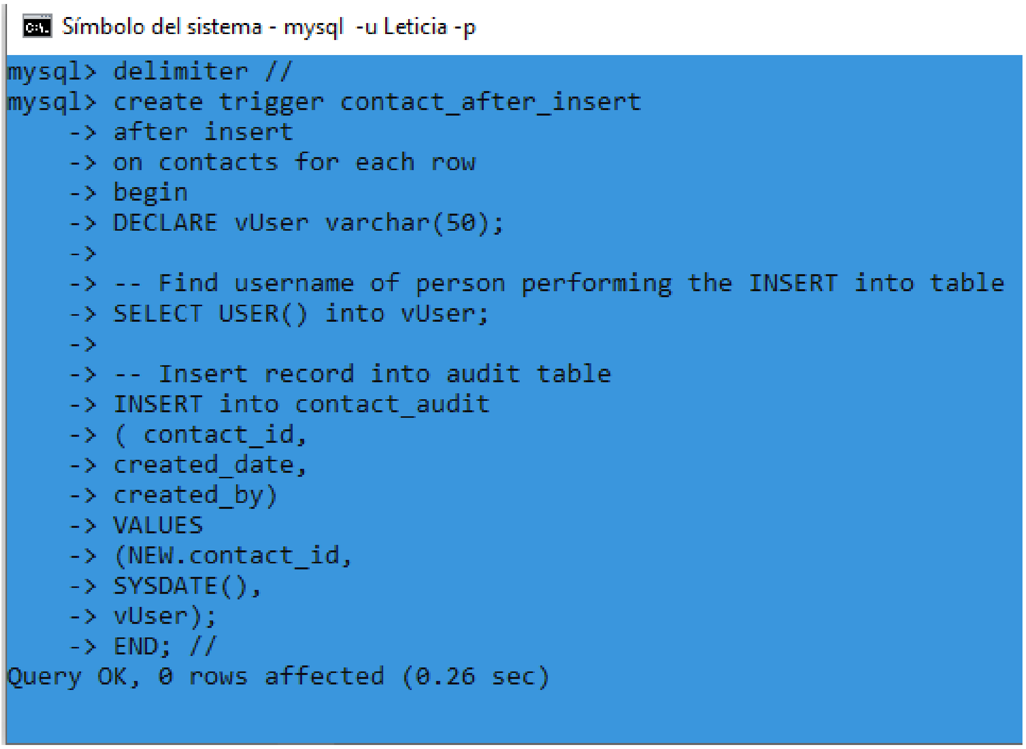
4. After Insert Trigger:

As the name implies, this trigger gets invoked after an insert is implemented.

Example:



```
mysql> create table contact (contact_id int (11) NOT NULL AUTO_INCREMENT,last_name VARCHAR(30) NOT NULL, first_name VARC
HAR(25), birthday DATE,
    -> CONSTRAINT contacts_pk PRIMARY KEY (contact_id));
Query OK, 0 rows affected (0.67 sec)

mysql> create table contacts_audit (contact_id integer,created_date date,created_by varchar (30));
Query OK, 0 rows affected (0.51 sec)
```

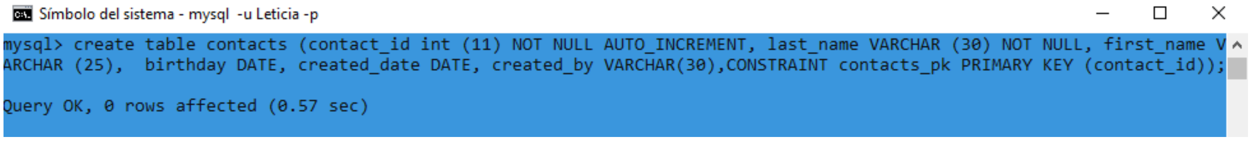


```
mysql> delimiter //
mysql> create trigger contact_after_insert
    -> after insert
    -> on contacts for each row
    -> begin
    -> DECLARE vUser varchar(50);
    ->
    -> -- Find username of person performing the INSERT into table
    -> SELECT USER() into vUser;
    ->
    -> -- Insert record into audit table
    -> INSERT into contact_audit
    -> ( contact_id,
    -> created_date,
    -> created_by)
    -> VALUES
    -> (NEW.contact_id,
    -> SYSDATE(),
    -> vUser);
    -> END; //
Query OK, 0 rows affected (0.26 sec)
```

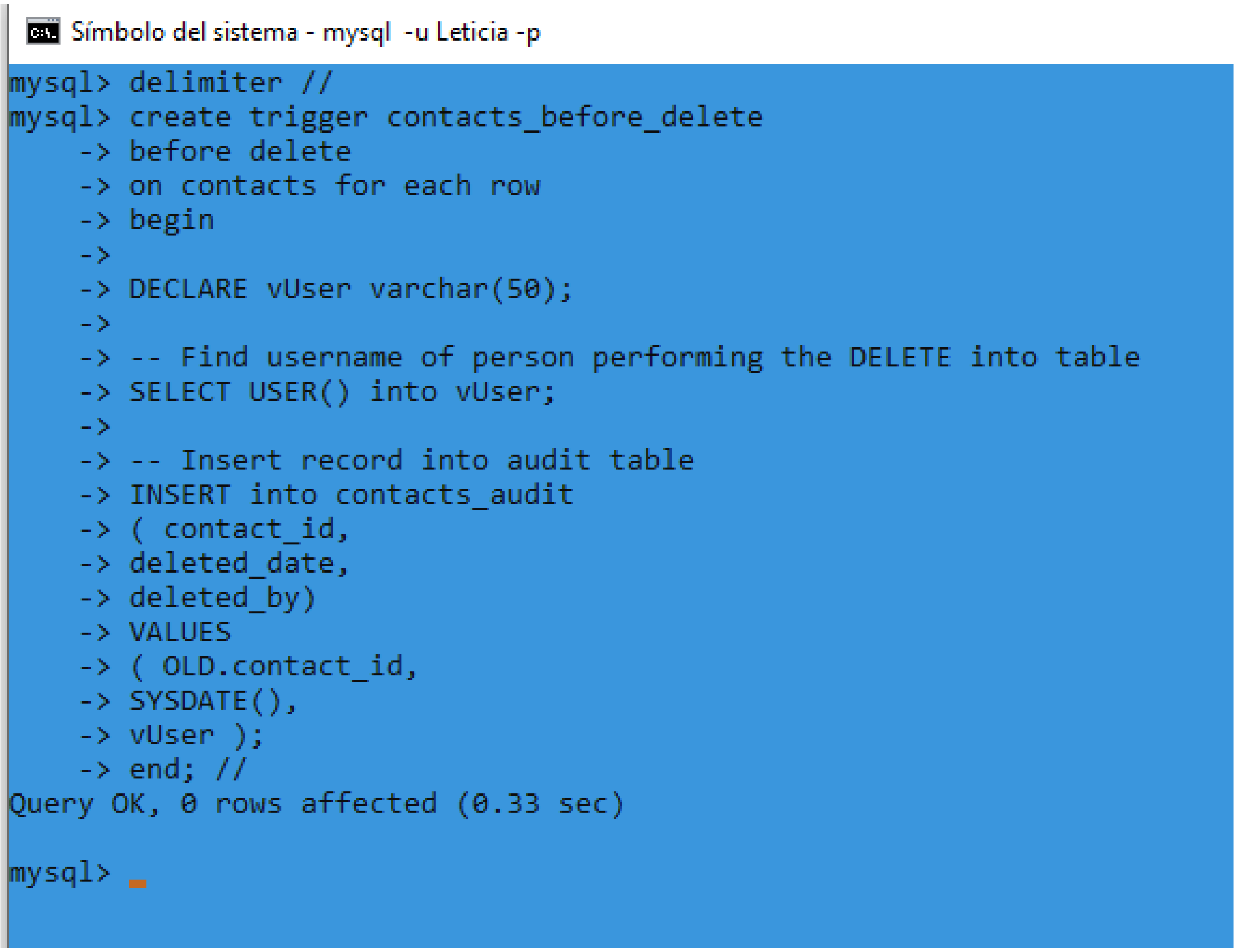
5. Before Delete Trigger:

As the name implies, this trigger is invoked before a delete occurs, or before deletion statement is implemented.

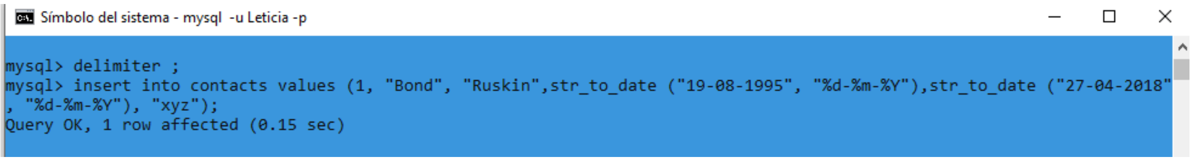
Example:



```
mysql> create table contacts (contact_id int (11) NOT NULL AUTO_INCREMENT, last_name VARCHAR (30) NOT NULL, first_name V
ARCHAR (25),  birthday DATE, created_date DATE, created_by VARCHAR(30),CONSTRAINT contacts_pk PRIMARY KEY (contact_id));

Query OK, 0 rows affected (0.57 sec)
```



```
mysql> delimiter //
mysql> create trigger contacts_before_delete
    -> before delete
    -> on contacts 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_audit
    -> ( contact_id,
    -> deleted_date,
    -> deleted_by)
    -> VALUES
    -> ( OLD.contact_id,
    -> SYSDATE(),
    -> vUser );
    -> end; //
Query OK, 0 rows affected (0.33 sec)

mysql>
```



```
mysql> delimiter ;
mysql> insert into contacts values (1, "Bond", "Ruskin",str_to_date ("19-08-1995", "%d-%m-%Y"),str_to_date ("27-04-2018"
, "%d-%m-%Y"), "xyz");
Query OK, 1 row affected (0.15 sec)
```