Instituto Tecnológico Superior de Jerez – ITSJ

Estudiante: Miguel Angel Bazán garduño.

mabg211299@hotmail.com

5to Semestre Carrera: Ingeniería en sistemas computacionales (ISC).

Materia: Taller de Base De Datos.

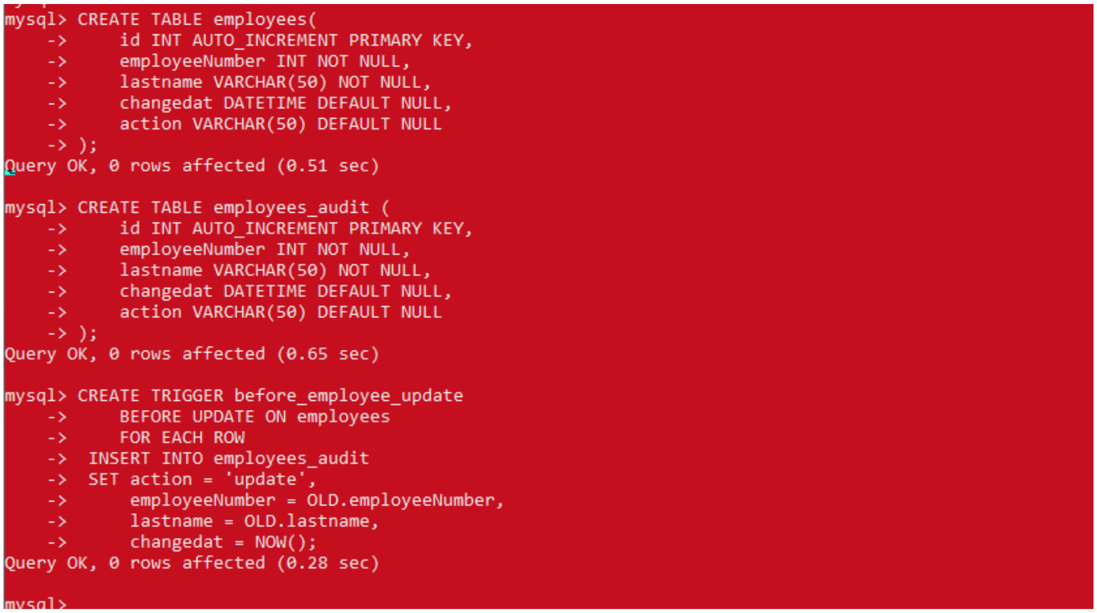
Actividad: Tutorial Triggers.

Docente: I.S.C. Salvador Acevedo Sandoval

Jerez de García Salinas Zacatecas

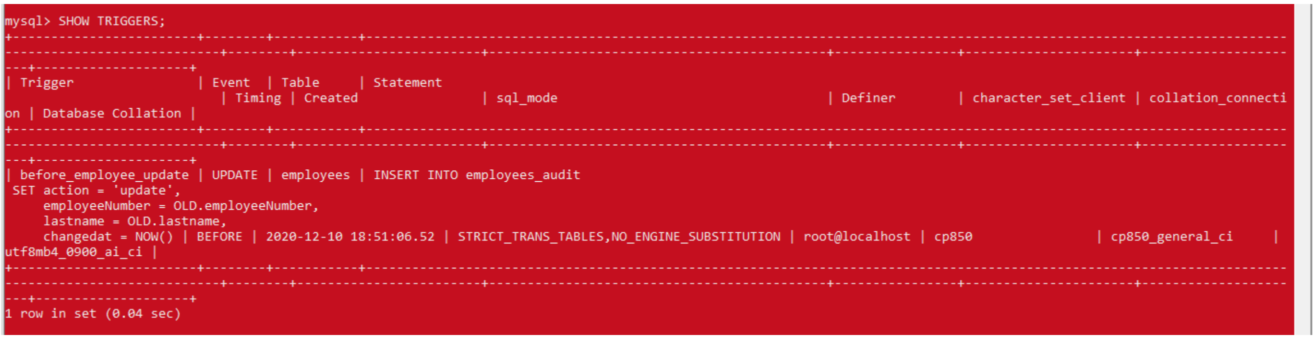
Create triggers:

Creación tabla y trigger:

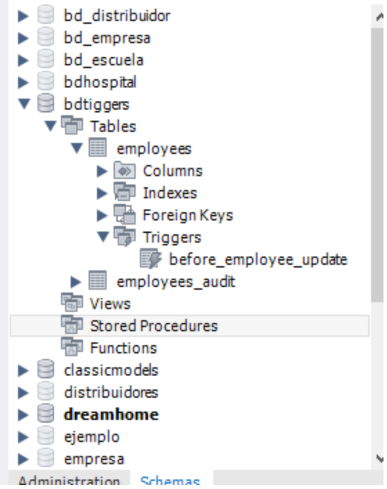


```
mysql> CREATE TABLE employees(
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     employeeNumber INT NOT NULL,
    ->     lastname VARCHAR(50) NOT NULL,
    ->     changedat DATETIME DEFAULT NULL,
    ->     action VARCHAR(50) DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.51 sec)

mysql> CREATE TABLE employees_audit (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     employeeNumber INT NOT NULL,
    ->     lastname VARCHAR(50) NOT NULL,
    ->     changedat DATETIME DEFAULT NULL,
    ->     action VARCHAR(50) DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.65 sec)

mysql> CREATE TRIGGER before_employee_update
    ->     BEFORE UPDATE ON employees
    ->     FOR EACH ROW
    ->  INSERT INTO employees_audit
    ->  SET action = 'update',
    ->      employeeNumber = OLD.employeeNumber,
    ->      lastname = OLD.lastname,
    ->      changedat = NOW();
Query OK, 0 rows affected (0.28 sec)

mysql>
```

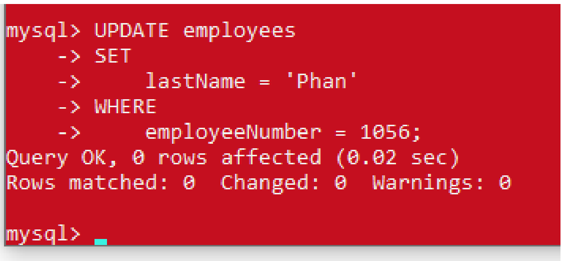
Visualización del trigger:



```
mysql> SHOW TRIGGERS;
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------
--------------------------+--------+-------------------------+---------------------------------------------+----------------+----------------------+-----------------
---+--------------------+
| Trigger                | Event  | Table     | Statement
                          | Timing | Created                 | sql_mode                                    | Definer        | character_set_client | collation_connecti
on | Database Collation |
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------
--------------------------+--------+-------------------------+---------------------------------------------+----------------+----------------------+-----------------
---+--------------------+
| before_employee_update | UPDATE | employees | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW() | BEFORE | 2020-12-10 18:51:06.52 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     |
utf8mb4_0900_ai_ci |
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------
--------------------------+--------+-------------------------+---------------------------------------------+----------------+----------------------+-----------------
---+--------------------+
1 row in set (0.04 sec)
```

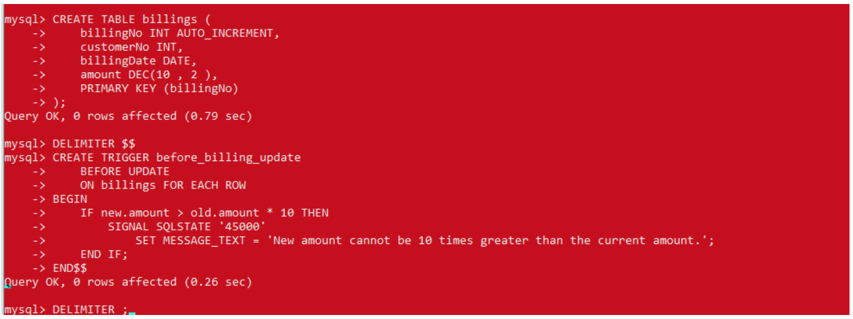


Actualización:



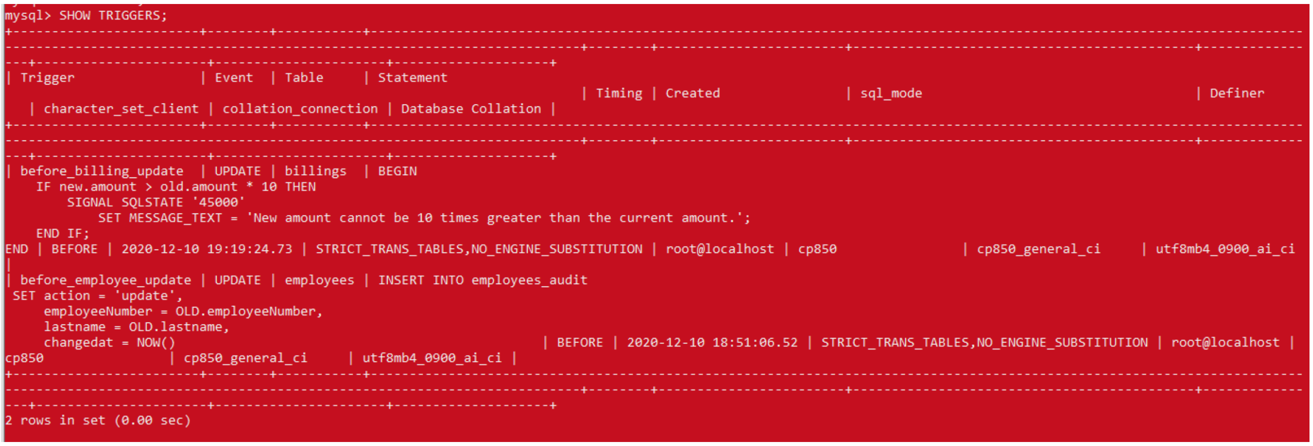
```
mysql> UPDATE employees
    -> SET
    ->     lastName = 'Phan'
    -> WHERE
    ->     employeeNumber = 1056;
Query OK, 0 rows affected (0.02 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql>
```

Drop triggers:

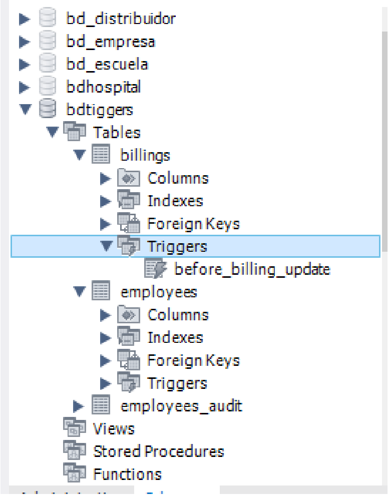


```
mysql> CREATE TABLE billings (
    ->     billingNo INT AUTO_INCREMENT,
    ->     customerNo INT,
    ->     billingDate DATE,
    ->     amount DEC(10 , 2 ),
    ->     PRIMARY KEY (billingNo)
    -> );
Query OK, 0 rows affected (0.79 sec)

mysql> DELIMITER $$
mysql> CREATE TRIGGER before_billing_update
    ->     BEFORE UPDATE
    ->     ON billings FOR EACH ROW
    -> BEGIN
    ->     IF new.amount > old.amount * 10 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = 'New amount cannot be 10 times greater than the current amount.';
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.26 sec)

mysql> DELIMITER ;
```

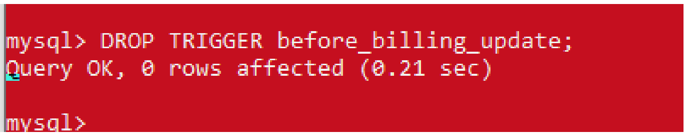
Visualización:



```
mysql> SHOW TRIGGERS;
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------+--------+------------------------+--------------------------------------------+-------------
---+----------------------+----------------------+--------------------+
| Trigger                | Event  | Table     | Statement
                                                                          | Timing | Created                | sql_mode                                   | Definer
   | character_set_client | collation_connection | Database Collation |
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------+--------+------------------------+--------------------------------------------+-------------
---+----------------------+----------------------+--------------------+
| before_billing_update  | UPDATE | billings  | BEGIN
    IF new.amount > old.amount * 10 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'New amount cannot be 10 times greater than the current amount.';
    END IF;
END | BEFORE | 2020-12-10 19:19:24.73 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci
|
| before_employee_update | UPDATE | employees | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW()                                                    | BEFORE | 2020-12-10 18:51:06.52 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost |
cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
+------------------------+--------+-----------+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------+--------+------------------------+--------------------------------------------+-------------
---+----------------------+----------------------+--------------------+
2 rows in set (0.00 sec)
```

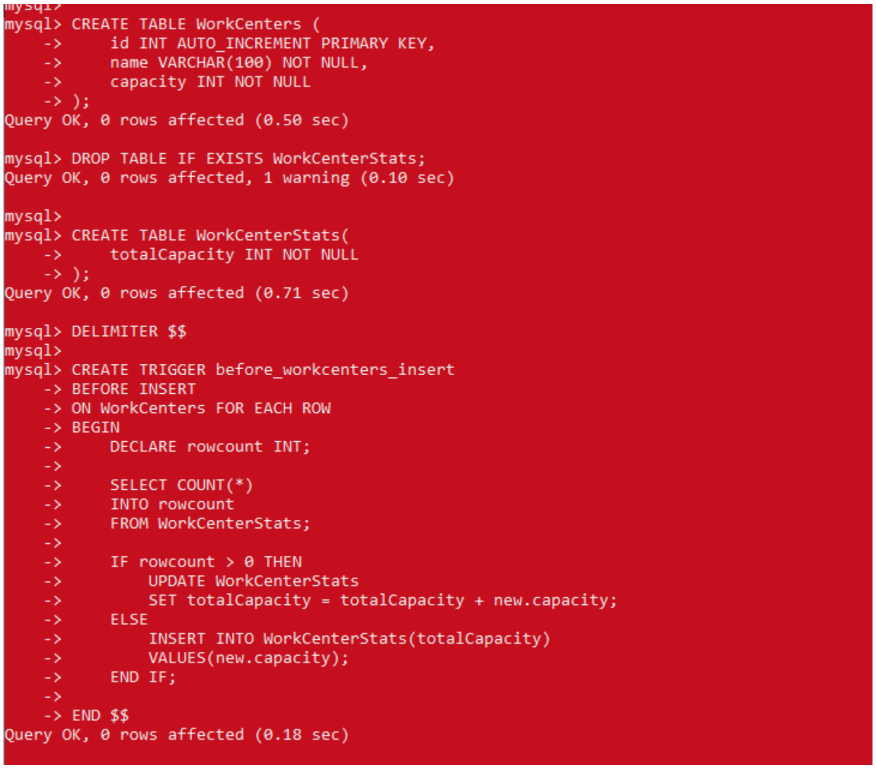


Eliminación:



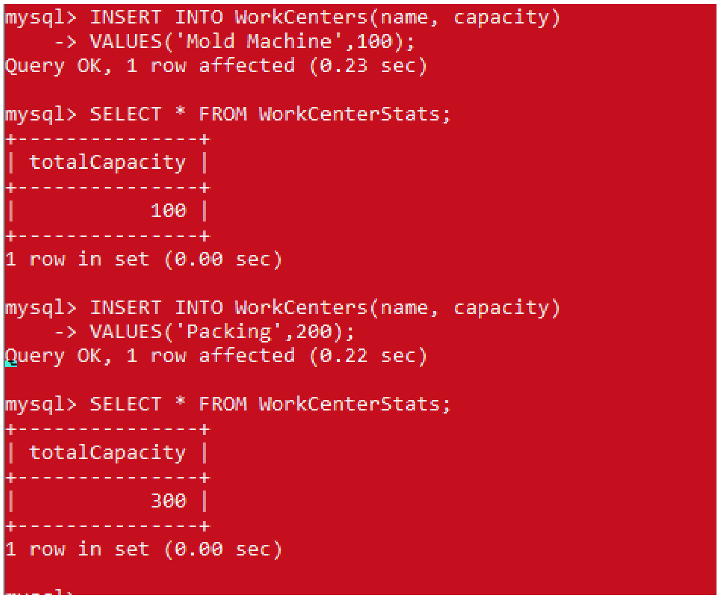
```
mysql> DROP TRIGGER before_billing_update;
Query OK, 0 rows affected (0.21 sec)

mysql>
```

BEFORE INSERT Trigger



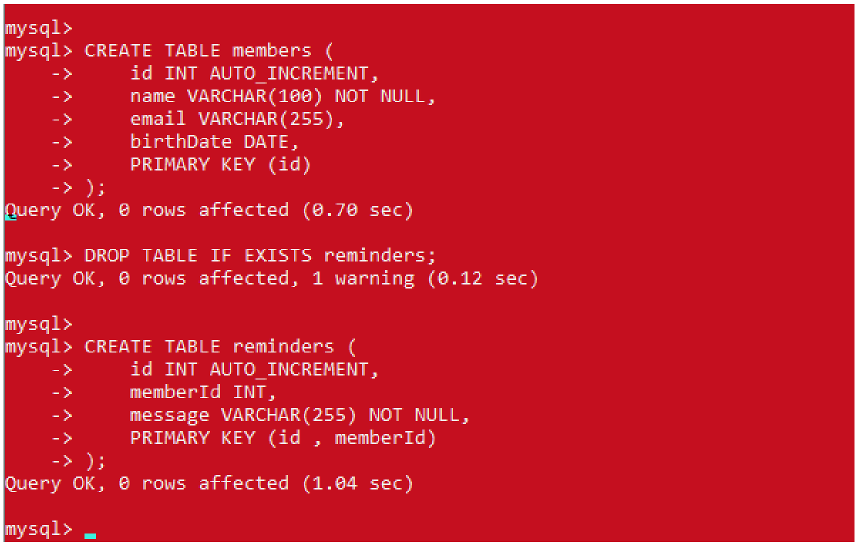
```
mysql>
mysql> CREATE TABLE WorkCenters (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100) NOT NULL,
    ->     capacity INT NOT NULL
    -> );
Query OK, 0 rows affected (0.50 sec)

mysql> DROP TABLE IF EXISTS WorkCenterStats;
Query OK, 0 rows affected, 1 warning (0.10 sec)

mysql>
mysql> CREATE TABLE WorkCenterStats(
    ->     totalCapacity INT NOT NULL
    -> );
Query OK, 0 rows affected (0.71 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_workcenters_insert
    -> BEFORE INSERT
    -> ON WorkCenters FOR EACH ROW
    -> BEGIN
    ->     DECLARE rowcount INT;
    ->
    ->     SELECT COUNT(*)
    ->     INTO rowcount
    ->     FROM WorkCenterStats;
    ->
    ->     IF rowcount > 0 THEN
    ->         UPDATE WorkCenterStats
    ->         SET totalCapacity = totalCapacity + new.capacity;
    ->     ELSE
    ->         INSERT INTO WorkCenterStats(totalCapacity)
    ->         VALUES(new.capacity);
    ->     END IF;
    ->
    -> END $$
Query OK, 0 rows affected (0.18 sec)
```

Probando Trigger:



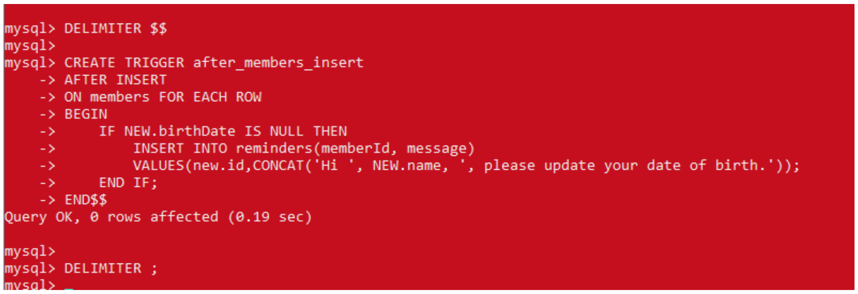
```
mysql> INSERT INTO WorkCenters(name, capacity)
    -> VALUES('Mold Machine',100);
Query OK, 1 row affected (0.23 sec)

mysql> SELECT * FROM WorkCenterStats;
+---------------+
| totalCapacity |
+---------------+
|           100 |
+---------------+
1 row in set (0.00 sec)

mysql> INSERT INTO WorkCenters(name, capacity)
    -> VALUES('Packing',200);
Query OK, 1 row affected (0.22 sec)

mysql> SELECT * FROM WorkCenterStats;
+---------------+
| totalCapacity |
+---------------+
|           300 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

AFTER INSERT triggers:



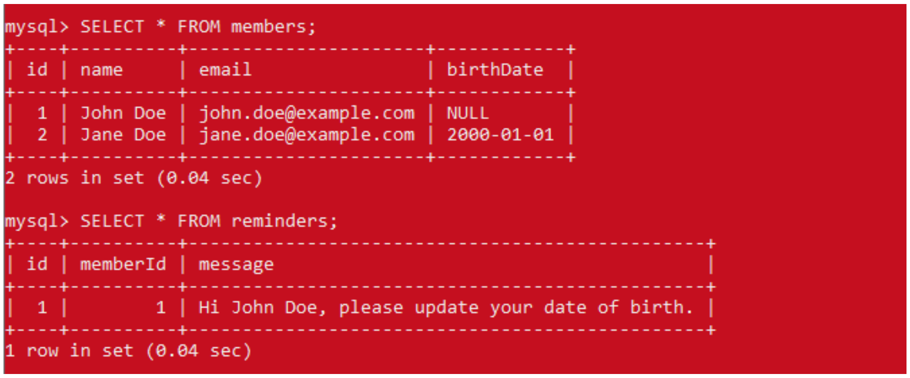
```
mysql>
mysql> CREATE TABLE members (
    ->     id INT AUTO_INCREMENT,
    ->     name VARCHAR(100) NOT NULL,
    ->     email VARCHAR(255),
    ->     birthDate DATE,
    ->     PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.70 sec)

mysql> DROP TABLE IF EXISTS reminders;
Query OK, 0 rows affected, 1 warning (0.12 sec)

mysql>
mysql> CREATE TABLE reminders (
    ->     id INT AUTO_INCREMENT,
    ->     memberId INT,
    ->     message VARCHAR(255) NOT NULL,
    ->     PRIMARY KEY (id , memberId)
    -> );
Query OK, 0 rows affected (1.04 sec)

mysql>
```

Creando TRIGGER:



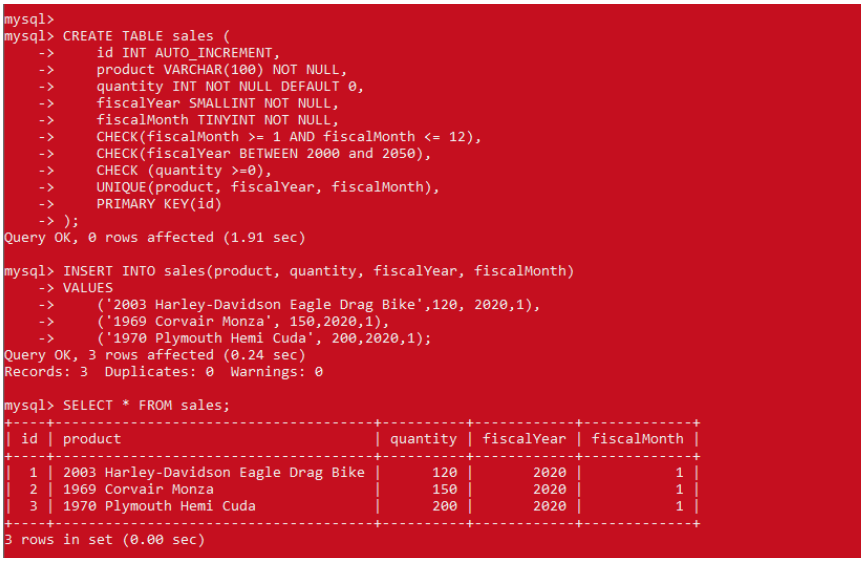
```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER after_members_insert
    -> AFTER INSERT
    -> ON members FOR EACH ROW
    -> BEGIN
    ->     IF NEW.birthDate IS NULL THEN
    ->         INSERT INTO reminders(memberId, message)
    ->         VALUES(new.id,CONCAT('Hi ', NEW.name, ', please update your date of birth.'));
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.19 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

Probando Trigger:



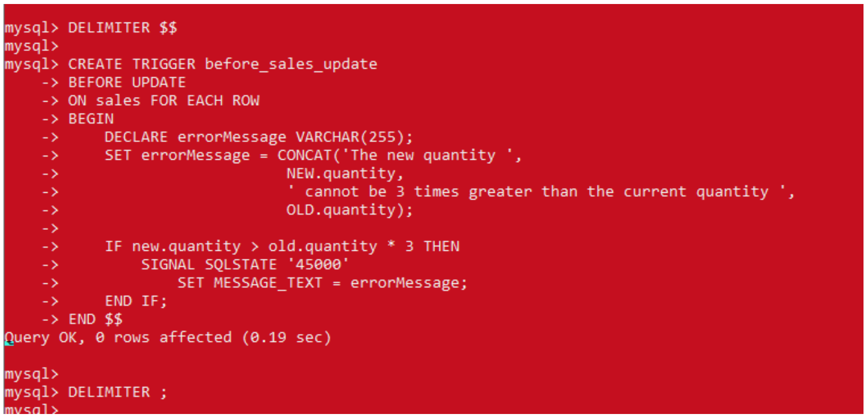
```
mysql> SELECT * FROM members;
+----+----------+----------------------+------------+
| id | name     | email                | birthDate  |
+----+----------+----------------------+------------+
|  1 | John Doe | john.doe@example.com | NULL       |
|  2 | Jane Doe | jane.doe@example.com | 2000-01-01 |
+----+----------+----------------------+------------+
2 rows in set (0.04 sec)

mysql> SELECT * FROM reminders;
+----+----------+--------------------------------------------+
| id | memberId | message                                    |
+----+----------+--------------------------------------------+
|  1 |        1 | Hi John Doe, please update your date of birth. |
+----+----------+--------------------------------------------+
1 row in set (0.04 sec)
```

Cree un activador ANTES DE ACTUALIZAR



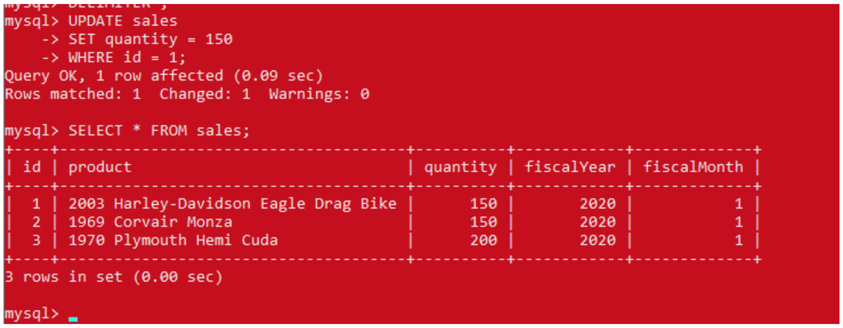
```
mysql>
mysql> CREATE TABLE sales (
    ->     id INT AUTO_INCREMENT,
    ->     product VARCHAR(100) NOT NULL,
    ->     quantity INT NOT NULL DEFAULT 0,
    ->     fiscalYear SMALLINT NOT NULL,
    ->     fiscalMonth TINYINT NOT NULL,
    ->     CHECK(fiscalMonth >= 1 AND fiscalMonth <= 12),
    ->     CHECK(fiscalYear BETWEEN 2000 and 2050),
    ->     CHECK (quantity >=0),
    ->     UNIQUE(product, fiscalYear, fiscalMonth),
    ->     PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (1.91 sec)

mysql> INSERT INTO sales(product, quantity, fiscalYear, fiscalMonth)
    -> VALUES
    ->     ('2003 Harley-Davidson Eagle Drag Bike',120, 2020,1),
    ->     ('1969 Corvair Monza', 150,2020,1),
    ->     ('1970 Plymouth Hemi Cuda', 200,2020,1);
Query OK, 3 rows affected (0.24 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM sales;
+----+--------------------------------------+----------+------------+-------------+
| id | product                              | quantity | fiscalYear | fiscalMonth |
+----+--------------------------------------+----------+------------+-------------+
|  1 | 2003 Harley-Davidson Eagle Drag Bike |      120 |       2020 |           1 |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 |
+----+--------------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)
```

Creación Trigger:



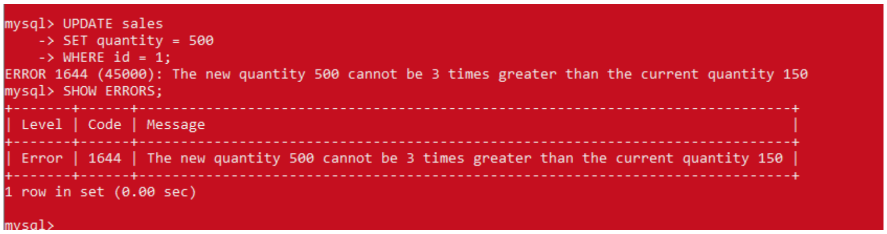
```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_sales_update
    -> BEFORE UPDATE
    -> ON sales FOR EACH ROW
    -> BEGIN
    ->     DECLARE errorMessage VARCHAR(255);
    ->     SET errorMessage = CONCAT('The new quantity ',
    ->                         NEW.quantity,
    ->                         ' cannot be 3 times greater than the current quantity ',
    ->                         OLD.quantity);
    ->
    ->     IF new.quantity > old.quantity * 3 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = errorMessage;
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.19 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

Probando Trigger:



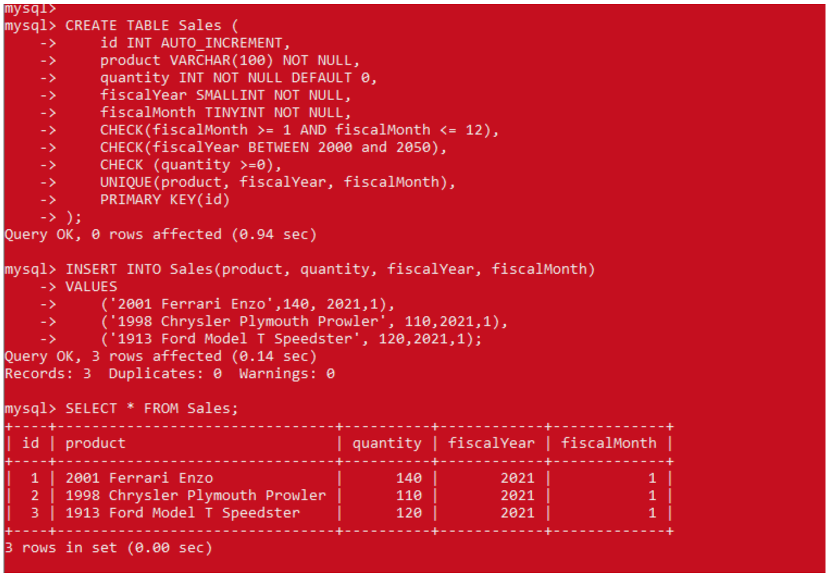
```
mysql> DELIMITER ;
mysql> UPDATE sales
    -> SET quantity = 150
    -> WHERE id = 1;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM sales;
+----+--------------------------------------+----------+------------+-------------+
| id | product                              | quantity | fiscalYear | fiscalMonth |
+----+--------------------------------------+----------+------------+-------------+
|  1 | 2003 Harley-Davidson Eagle Drag Bike |      150 |       2020 |           1 |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 |
+----+--------------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)

mysql>
```

Actualización ERROR.



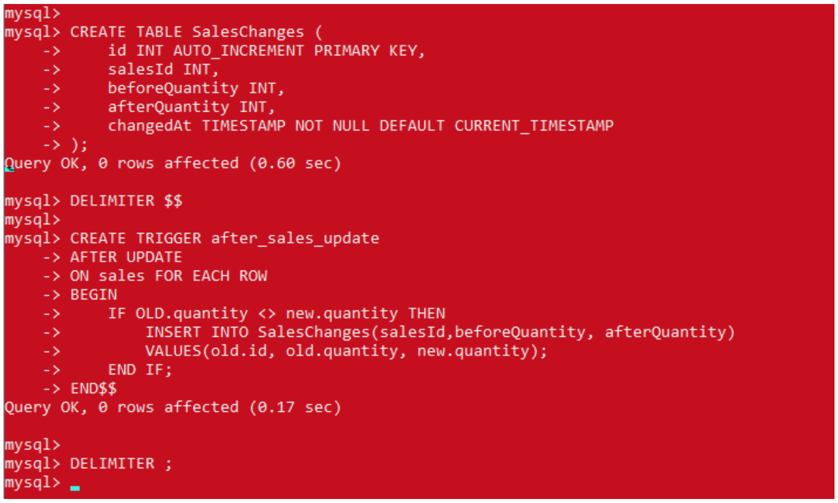
```
mysql> UPDATE sales
    -> SET quantity = 500
    -> WHERE id = 1;
ERROR 1644 (45000): The new quantity 500 cannot be 3 times greater than the current quantity 150
mysql> SHOW ERRORS;
+-------+------+------------------------------------------------------------------------------+
| Level | Code | Message                                                                      |
+-------+------+------------------------------------------------------------------------------+
| Error | 1644 | The new quantity 500 cannot be 3 times greater than the current quantity 150 |
+-------+------+------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Cree un activador DESPUÉS DE LA ACTUALIZACIÓN:



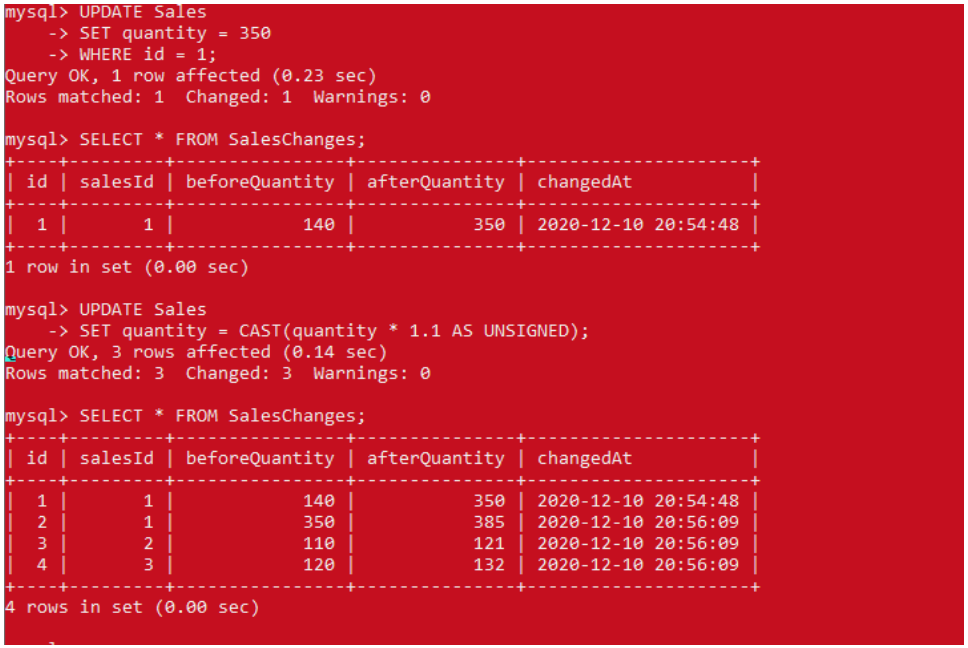
```
mysql>
mysql> CREATE TABLE Sales (
    ->     id INT AUTO_INCREMENT,
    ->     product VARCHAR(100) NOT NULL,
    ->     quantity INT NOT NULL DEFAULT 0,
    ->     fiscalYear SMALLINT NOT NULL,
    ->     fiscalMonth TINYINT NOT NULL,
    ->     CHECK(fiscalMonth >= 1 AND fiscalMonth <= 12),
    ->     CHECK(fiscalYear BETWEEN 2000 and 2050),
    ->     CHECK (quantity >=0),
    ->     UNIQUE(product, fiscalYear, fiscalMonth),
    ->     PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.94 sec)

mysql> INSERT INTO Sales(product, quantity, fiscalYear, fiscalMonth)
    -> VALUES
    ->     ('2001 Ferrari Enzo',140, 2021,1),
    ->     ('1998 Chrysler Plymouth Prowler', 110,2021,1),
    ->     ('1913 Ford Model T Speedster', 120,2021,1);
Query OK, 3 rows affected (0.14 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Sales;
+----+--------------------------------+----------+------------+-------------+
| id | product                        | quantity | fiscalYear | fiscalMonth |
+----+--------------------------------+----------+------------+-------------+
|  1 | 2001 Ferrari Enzo              |      140 |       2021 |           1 |
|  2 | 1998 Chrysler Plymouth Prowler |      110 |       2021 |           1 |
|  3 | 1913 Ford Model T Speedster    |      120 |       2021 |           1 |
+----+--------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)
```

Creación Trigger:



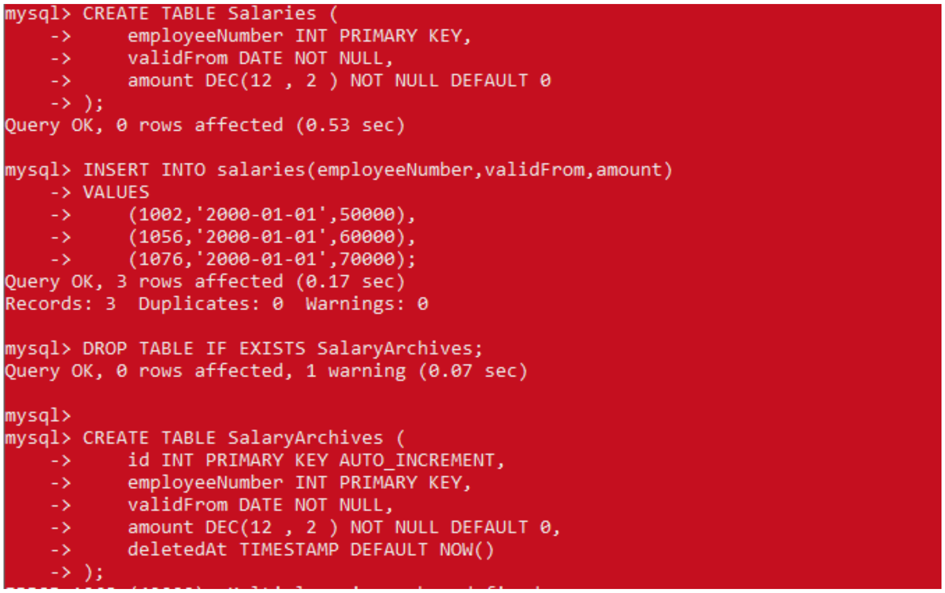
```
mysql>
mysql> CREATE TABLE SalesChanges (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     salesId INT,
    ->     beforeQuantity INT,
    ->     afterQuantity INT,
    ->     changedAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.60 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER after_sales_update
    -> AFTER UPDATE
    -> ON sales FOR EACH ROW
    -> BEGIN
    ->     IF OLD.quantity <> new.quantity THEN
    ->         INSERT INTO SalesChanges(salesId,beforeQuantity, afterQuantity)
    ->         VALUES(old.id, old.quantity, new.quantity);
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.17 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

Prueba:



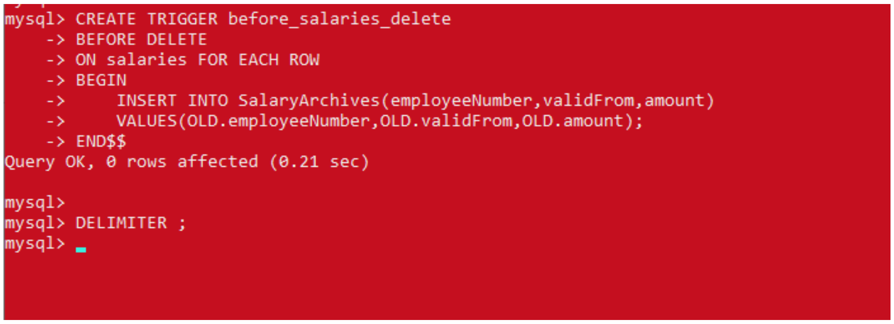
```
mysql> UPDATE Sales
    -> SET quantity = 350
    -> WHERE id = 1;
Query OK, 1 row affected (0.23 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM SalesChanges;
+----+---------+----------------+---------------+---------------------+
| id | salesId | beforeQuantity | afterQuantity | changedAt           |
+----+---------+----------------+---------------+---------------------+
|  1 |       1 |            140 |           350 | 2020-12-10 20:54:48 |
+----+---------+----------------+---------------+---------------------+
1 row in set (0.00 sec)

mysql> UPDATE Sales
    -> SET quantity = CAST(quantity * 1.1 AS UNSIGNED);
Query OK, 3 rows affected (0.14 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT * FROM SalesChanges;
+----+---------+----------------+---------------+---------------------+
| id | salesId | beforeQuantity | afterQuantity | changedAt           |
+----+---------+----------------+---------------+---------------------+
|  1 |       1 |            140 |           350 | 2020-12-10 20:54:48 |
|  2 |       1 |            350 |           385 | 2020-12-10 20:56:09 |
|  3 |       2 |            110 |           121 | 2020-12-10 20:56:09 |
|  4 |       3 |            120 |           132 | 2020-12-10 20:56:09 |
+----+---------+----------------+---------------+---------------------+
4 rows in set (0.00 sec)
```

Cree un disparador ANTES DELETE:



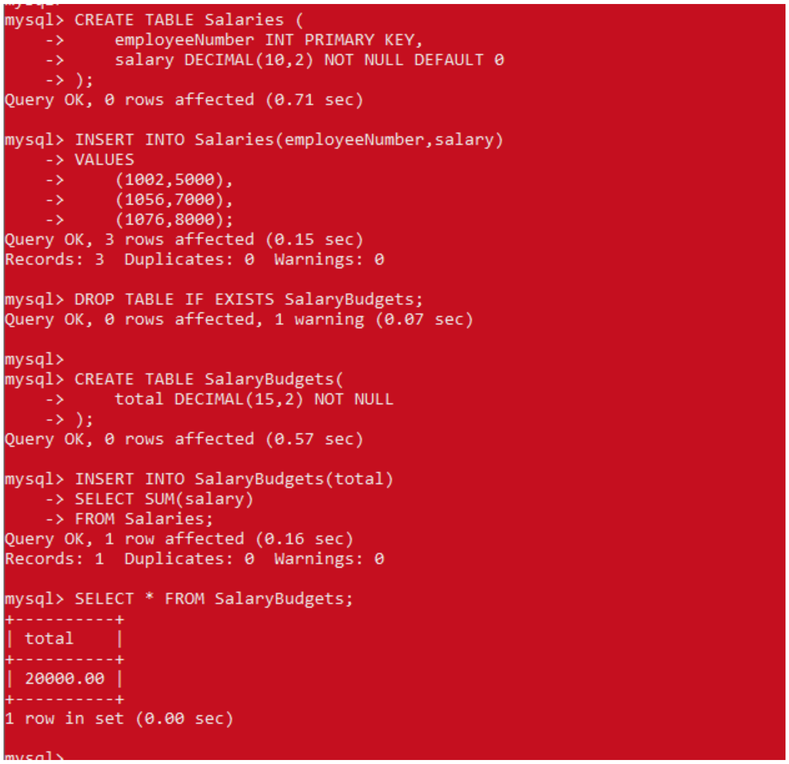
```
mysql> CREATE TABLE Salaries (
    ->     employeeNumber INT PRIMARY KEY,
    ->     validFrom DATE NOT NULL,
    ->     amount DEC(12 , 2 ) NOT NULL DEFAULT 0
    -> );
Query OK, 0 rows affected (0.53 sec)

mysql> INSERT INTO salaries(employeeNumber,validFrom,amount)
    -> VALUES
    ->     (1002,'2000-01-01',50000),
    ->     (1056,'2000-01-01',60000),
    ->     (1076,'2000-01-01',70000);
Query OK, 3 rows affected (0.17 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DROP TABLE IF EXISTS SalaryArchives;
Query OK, 0 rows affected, 1 warning (0.07 sec)

mysql>
mysql> CREATE TABLE SalaryArchives (
    ->     id INT PRIMARY KEY AUTO_INCREMENT,
    ->     employeeNumber INT PRIMARY KEY,
    ->     validFrom DATE NOT NULL,
    ->     amount DEC(12 , 2 ) NOT NULL DEFAULT 0,
    ->     deletedAt TIMESTAMP DEFAULT NOW()
    -> );
```

Creación TRIGGER:



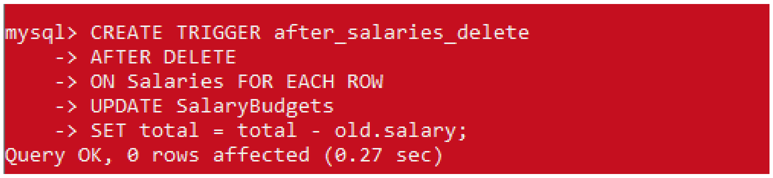
```
mysql> CREATE TRIGGER before_salaries_delete
    -> BEFORE DELETE
    -> ON salaries FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO SalaryArchives(employeeNumber,validFrom,amount)
    ->     VALUES(OLD.employeeNumber,OLD.validFrom,OLD.amount);
    -> END$$
Query OK, 0 rows affected (0.21 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

Crear un disparador DESPUÉS DEL BORRAR:



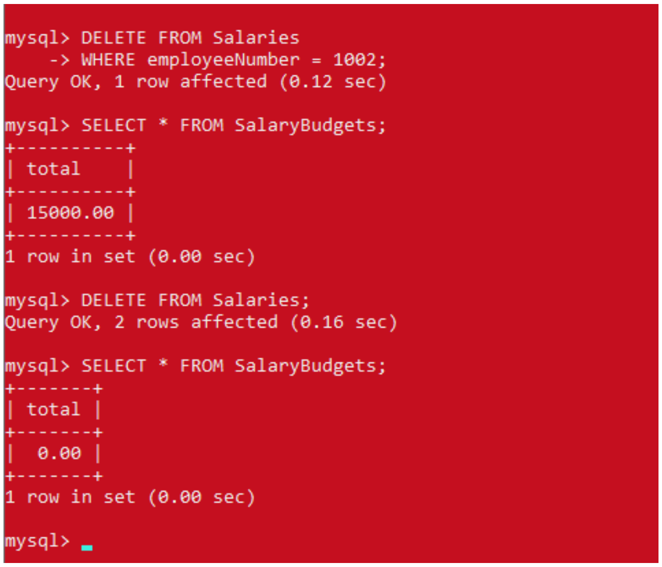
```
mysql> CREATE TABLE Salaries (
    ->     employeeNumber INT PRIMARY KEY,
    ->     salary DECIMAL(10,2) NOT NULL DEFAULT 0
    -> );
Query OK, 0 rows affected (0.71 sec)

mysql> INSERT INTO Salaries(employeeNumber,salary)
    -> VALUES
    ->     (1002,5000),
    ->     (1056,7000),
    ->     (1076,8000);
Query OK, 3 rows affected (0.15 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DROP TABLE IF EXISTS SalaryBudgets;
Query OK, 0 rows affected, 1 warning (0.07 sec)

mysql>
mysql> CREATE TABLE SalaryBudgets(
    ->     total DECIMAL(15,2) NOT NULL
    -> );
Query OK, 0 rows affected (0.57 sec)

mysql> INSERT INTO SalaryBudgets(total)
    -> SELECT SUM(salary)
    -> FROM Salaries;
Query OK, 1 row affected (0.16 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM SalaryBudgets;
+----------+
| total    |
+----------+
| 20000.00 |
+----------+
1 row in set (0.00 sec)
```

Creación de Trigger:



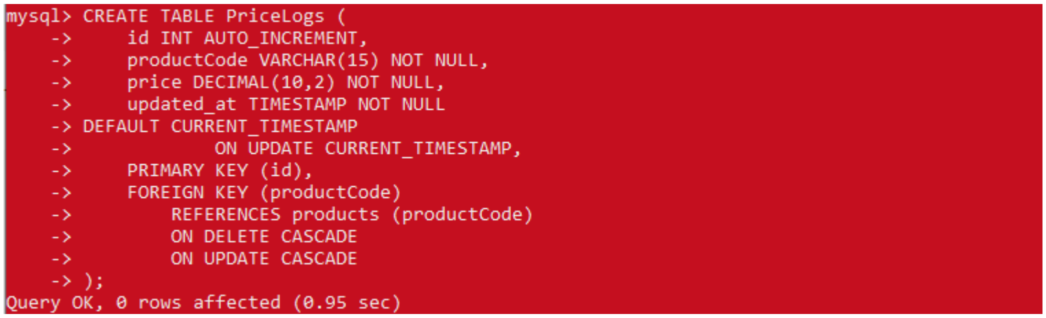
```
mysql> CREATE TRIGGER after_salaries_delete
    -> AFTER DELETE
    -> ON Salaries FOR EACH ROW
    -> UPDATE SalaryBudgets
    -> SET total = total - old.salary;
Query OK, 0 rows affected (0.27 sec)
```

Prueba:



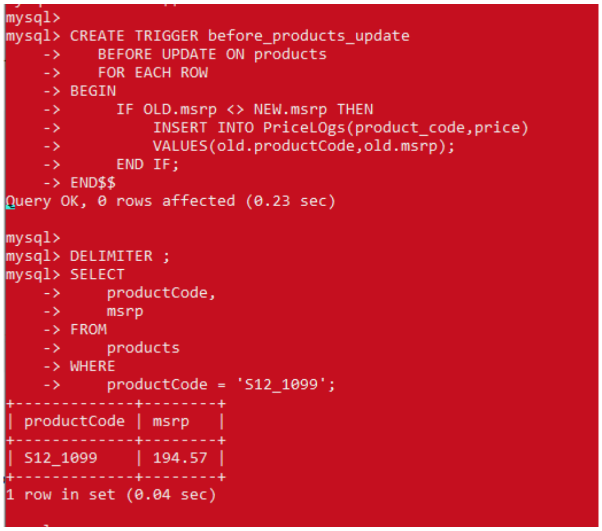
```
mysql> DELETE FROM Salaries
    -> WHERE employeeNumber = 1002;
Query OK, 1 row affected (0.12 sec)

mysql> SELECT * FROM SalaryBudgets;
+----------+
| total    |
+----------+
| 15000.00 |
+----------+
1 row in set (0.00 sec)

mysql> DELETE FROM Salaries;
Query OK, 2 rows affected (0.16 sec)

mysql> SELECT * FROM SalaryBudgets;
+-------+
| total |
+-------+
|  0.00 |
+-------+
1 row in set (0.00 sec)

mysql>
```

Cree varios activadores para una tabla que tengan el mismo evento y tiempo de activación:

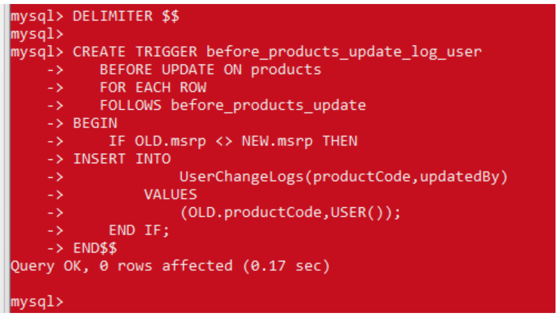


```
mysql> CREATE TABLE PriceLogs (
    ->     id INT AUTO_INCREMENT,
    ->     productCode VARCHAR(15) NOT NULL,
    ->     price DECIMAL(10,2) NOT NULL,
    ->     updated_at TIMESTAMP NOT NULL
    -> DEFAULT CURRENT_TIMESTAMP
    ->         ON UPDATE CURRENT_TIMESTAMP,
    ->     PRIMARY KEY (id),
    ->     FOREIGN KEY (productCode)
    ->         REFERENCES products (productCode)
    ->         ON DELETE CASCADE
    ->         ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.95 sec)
```

Creacion Trigger:

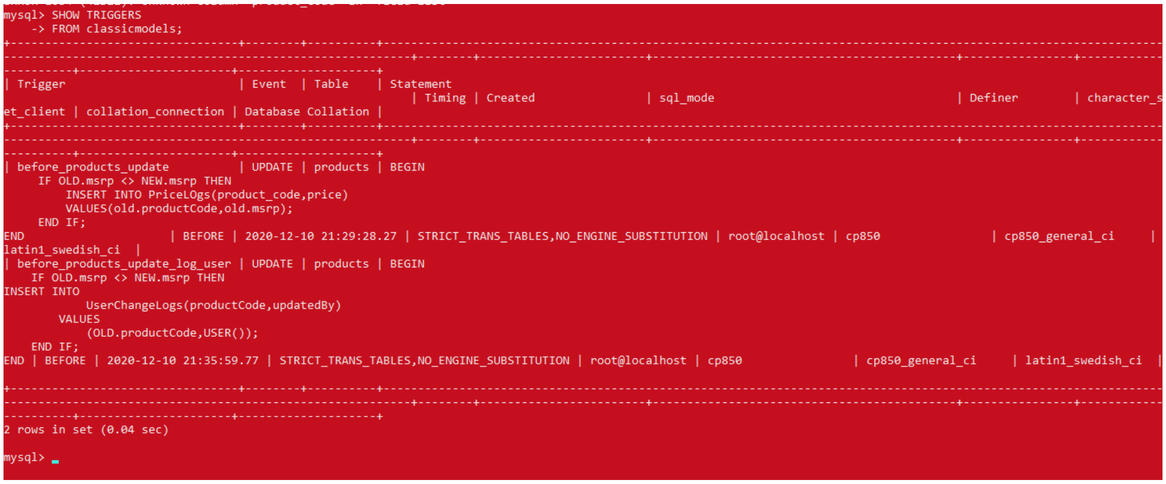


```
mysql>
mysql> CREATE TRIGGER before_products_update
    ->     BEFORE UPDATE ON products
    ->     FOR EACH ROW
    -> BEGIN
    ->      IF OLD.msrp <> NEW.msrp THEN
    ->          INSERT INTO PriceLOgs(product_code,price)
    ->          VALUES(old.productCode,old.msrp);
    ->      END IF;
    -> END$$
Query OK, 0 rows affected (0.23 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT
    ->     productCode,
    ->     msrp
    -> FROM
    ->     products
    -> WHERE
    ->     productCode = 'S12_1099';
+-------------+--------+
| productCode | msrp   |
+-------------+--------+
| S12_1099    | 194.57 |
+-------------+--------+
1 row in set (0.04 sec)
```



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_products_update_log_user
    ->     BEFORE UPDATE ON products
    ->     FOR EACH ROW
    ->     FOLLOWS before_products_update
    -> BEGIN
    ->     IF OLD.msrp <> NEW.msrp THEN
    -> INSERT INTO
    ->             UserChangeLogs(productCode,updatedBy)
    ->         VALUES
    ->             (OLD.productCode,USER());
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.17 sec)

mysql>
```

Ver Trigger:



```
mysql> SHOW TRIGGERS
    -> FROM classicmodels;
+----------------------------------+--------+----------+----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------+--------+------------------------+--------------------------------------------+----------------+----------------
----------+----------------------+--------------------+
| Trigger                          | Event  | Table    | Statement
                                                                | Timing | Created                | sql_mode                                   | Definer        | character_s
et_client | collation_connection | Database Collation |
+----------------------------------+--------+----------+----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------+--------+------------------------+--------------------------------------------+----------------+----------------
----------+----------------------+--------------------+
| before_products_update           | UPDATE | products | BEGIN
     IF OLD.msrp <> NEW.msrp THEN
         INSERT INTO PriceLOgs(product_code,price)
         VALUES(old.productCode,old.msrp);
     END IF;
END                         | BEFORE | 2020-12-10 21:29:28.27 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     |
latin1_swedish_ci  |
| before_products_update_log_user | UPDATE | products | BEGIN
    IF OLD.msrp <> NEW.msrp THEN
INSERT INTO
            UserChangeLogs(productCode,updatedBy)
        VALUES
            (OLD.productCode,USER());
    END IF;
END | BEFORE | 2020-12-10 21:35:59.77 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     | latin1_swedish_ci  |
+----------------------------------+--------+----------+----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------+--------+------------------------+--------------------------------------------+----------------+----------------
----------+----------------------+--------------------+
2 rows in set (0.04 sec)

mysql>
```