INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

INGENIERÍA EN SISTEMAS COMPUTACIONALES

5to Semestre

Fecha de entrega: 11/12/2020

Actividad 3: Ejercicios TRIGGERS.

Tema 5: SQL Procedural.

Materia: Taller de Base de Datos.

Nombre del Alumno: Marín Ramírez Mario.

Número de Control: S18070186

Correo electrónico: mariomarin502t@gmail.com

Profesor: I.S.C. Salvador Acevedo Sandoval.

1. *Create Triggers.*

```
mysql> CREATE Table employees(employeeNumber int AUTO_INCREMENT PRIMARY KEY, lastName varchar(30), fisrtName varchar(30), extension varchar(30), email varchar(30), offi
ceCode int, reportsTo varchar(30), jobTitle varchar(30));
Query OK, 0 rows affected (0.03 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 (0.03 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 (0.01 sec)

mysql> UPDATE employees
    -> SET
    ->     lastName = 'Phan'
    -> WHERE
    ->     employeeNumber = 1056;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

2. *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 (0.03 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 (0.01 sec)
```

```
mysql> DROP TRIGGER before_billing_update;
Query OK, 0 rows affected (0.01 sec)
```

3. Create a BEFORE INSERT trigger

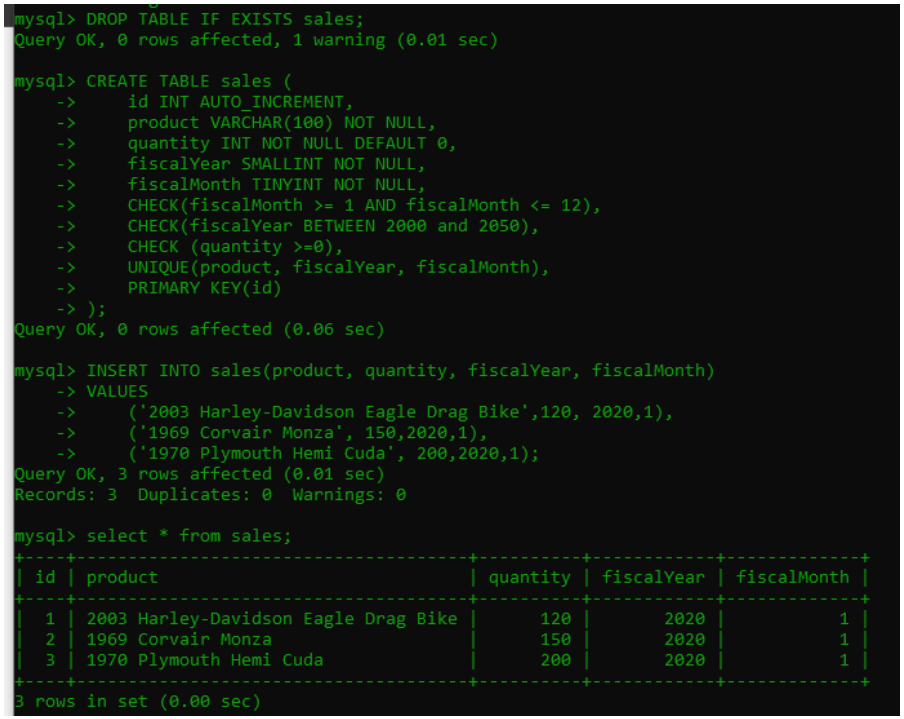
```
mysql>
mysql> DROP TABLE IF EXISTS WorkCenters;
Query OK, 0 rows affected, 1 warning (0.01 sec)

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 (0.03 sec)

mysql> DROP TABLE IF EXISTS WorkCenterStats;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql>
mysql> CREATE TABLE WorkCenterStats(
    ->     totalCapacity INT NOT NULL
    -> );
Query OK, 0 rows affected (0.03 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 (0.01 sec)
```

4. *Create an AFTER INSERT trigger*

```
mysql> DROP TABLE IF EXISTS members;
Query OK, 0 rows affected, 1 warning (0.01 sec)

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 (0.03 sec)

mysql> DROP TABLE IF EXISTS reminders;
Query OK, 0 rows affected, 1 warning (0.01 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 (0.03 sec)

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 (0.01 sec)
```

5. *Create a BEFORE UPDATE trigger*



```
mysql> DROP TABLE IF EXISTS sales;
Query OK, 0 rows affected, 1 warning (0.01 sec)

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 (0.06 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 (0.01 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)
```

6. *Create an AFTER UPDATE trigger*

```
mysql> DROP TABLE IF EXISTS Sales;
Query OK, 0 rows affected (0.03 sec)

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 (0.04 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 (0.01 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)
```

7. *Create a BEFORE DELETE trigger*

```
mysql> DROP TABLE IF EXISTS Salaries;
Query OK, 0 rows affected, 1 warning (0.01 sec)

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 (0.03 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 (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DROP TABLE IF EXISTS SalaryArchives;
Query OK, 0 rows affected, 1 warning (0.01 sec)

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()
```