

TRIGGERS

Name: Yash Gupta

Roll No.: S20200010234

```
mysql> create table item (itemid int, iDesc varchar(10), quantity int, price int, category varchar(20));
Query OK, 0 rows affected (0.15 sec)

mysql> insert into item values(1, 'ponds',10,100, 'Cosmetics');
Query OK, 1 row affected (0.02 sec)

mysql> insert into item values(2, 'plants',90,100, 'Garden');
Query OK, 1 row affected (0.08 sec)

mysql> insert into item values(3, 'perfume',5,500, 'Cosmetics');
Query OK, 1 row affected (0.08 sec)

mysql> insert into item values(4, 'kb',10,10, 'toys');
Query OK, 1 row affected (0.08 sec)

mysql> insert into item values(5, 'train',80,60, 'toys');
Query OK, 1 row affected (0.01 sec)

mysql> insert into item values(6, 'knife',10,100, 'Kitchen');
Query OK, 1 row affected (0.08 sec)
```

Tables

```
mysql> create table sales (sid int, itemid varchar(10), qtysold int, price int, total
-> varchar(20) default 0);
Query OK, 0 rows affected (0.15 sec)

mysql> insert into sales(sid,itemid,qtysold,price) values(1,2,5,500);
Query OK, 1 row affected (0.09 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 (0.13 sec)
```

```
mysql> CREATE TABLE WorkCenters (
  -> id INT AUTO_INCREMENT PRIMARY KEY,
  -> name VARCHAR(100) NOT NULL,
  -> capacity INT NOT NULL
  -> );
Query OK, 0 rows affected (0.23 sec)

mysql> DROP TABLE IF EXISTS WorkCenterStats;
Query OK, 0 rows affected, 1 warning (0.08 sec)

mysql> CREATE TABLE WorkCenterStats(
  -> totalCapacity INT NOT NULL
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> DROP TABLE IF EXISTS members;
Query OK, 0 rows affected, 1 warning (0.08 sec)
```

Exercise Problems

```
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 (0.15 sec)

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

mysql> CREATE TABLE reminders (
  -> id INT AUTO_INCREMENT,
  -> memberId INT,
  -> message VARCHAR(255) NOT NULL,
  -> PRIMARY KEY (id , memberId)
  -> );
Query OK, 0 rows affected (0.20 sec)

mysql> DROP TABLE IF EXISTS Salaries;
Query OK, 0 rows affected, 1 warning (0.08 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 (0.04 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 (0.10 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```

mysql> DROP TABLE IF EXISTS SalaryArchives;
Query OK, 0 rows affected, 1 warning (0.00 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()
  -> );
ERROR 1068 (42000): Multiple primary key defined
mysql> DROP TABLE IF EXISTS Salaries;
Query OK, 0 rows affected (0.14 sec)

mysql> CREATE TABLE Salaries (
  -> employeeNumber INT PRIMARY KEY,
  -> salary DECIMAL(10,2) NOT NULL DEFAULT 0 );
Query OK, 0 rows affected (0.13 sec)

mysql> INSERT INTO Salaries(employeeNumber,salary) VALUES
  -> (1002,5000),
  -> (1056,7000),
  -> (1076,8000);
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> DROP TABLE IF EXISTS SalaryBudgets;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE TABLE SalaryBudgets(
  -> total DECIMAL(15,2) NOT NULL
  -> );
Query OK, 0 rows affected (0.13 sec)

mysql> INSERT INTO SalaryBudgets(total)
  -> SELECT SUM(salary)
  -> FROM Salaries;
Query OK, 1 row affected (0.08 sec)
Records: 1 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM SalaryBudgets;
+-----+
| total |
+-----+
| 20000.00 |
+-----+
1 row in set (0.07 sec)

```

```

mysql> CREATE TABLE SalaryArchives (
  -> id INT PRIMARY KEY AUTO_INCREMENT,
  -> employeeNumber INT,
  -> validFrom DATE NOT NULL,
  -> amount DEC(12 , 2 ) NOT NULL DEFAULT 0,
  -> deletedAt TIMESTAMP DEFAULT NOW()
  -> );
Query OK, 0 rows affected (0.15 sec)

```

Q1. Write a trigger program to update the total capacity in the WorkCenterStats table before a new work center is inserted into the WorkCenter table:

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER before_workcenters_insert
  -> BEFORE INSERT
  -> ON WorkCenters FOR EACH ROW
  -> BEGIN
  -> DECLARE rowcount INT;
  -> SELECT COUNT(*)
  -> INTO rowcount
  -> FROM WorkCenterStats;
  -> IF rowcount > 0 THEN
  -> UPDATE WorkCenterStats
  -> SET totalCapacity = totalCapacity + new.capacity;
  -> ELSE
  -> INSERT INTO WorkCenterStats(totalCapacity)
  -> VALUES(new.capacity);
  -> END IF;
  -> END $$
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> INSERT INTO WorkCenters(name, capacity) VALUES('Mold Machine',100);
Query OK, 1 row affected (0.09 sec)

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

Q2. Write a trigger program to inserts a reminder into the reminders table if the birth date of the member is NULL.

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER after_members_insert
  -> AFTER INSERT
  -> ON members FOR EACH ROW
  -> BEGIN
  -> IF NEW.birthDate IS NULL THEN
  -> INSERT INTO reminders(memberId, message)
  -> VALUES(new.id,CONCAT('Hi ', NEW.name, ', please update your date of  birth.'));
  -> END IF;
  -> END$$
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> DELIMITER ;
mysql> INSERT INTO members(name, email, birthDate)
  -> VALUES
  -> ('John Doe', 'john.doe@example.com', NULL),
  -> ('Jane Doe', 'jane.doe@example.com', '2000-01-01');
Query OK, 2 rows affected (0.10 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
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 (0.00 sec)
```

```
mysql> SELECT * FROM reminders;
+----+-----+-----+-----+
| id | memberId | message                                     |
+----+-----+-----+-----+
|  1 |          | Hi John Doe, please update your date of  birth. |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q3. Create MySQL Trigger that inserts a new row into the SalaryArchives table before a row from the Salaries table is deleted.

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER before_salaries_delete
  -> BEFORE DELETE
  -> ON salaries FOR EACH ROW
  -> BEGIN
  -> INSERT INTO SalaryArchives(employeeNumber,validFrom,amount)
  -> VALUES(old.employeeNumber,old.validFrom,old.amount);
  -> END$$
```

Q4. Create MySQL Trigger updates the total salary in the SalaryBudgets table after a row is deleted from the Salaries table.

```
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 (0.11 sec)
```

```
mysql> DELETE FROM Salaries
  -> WHERE employeeNumber = 1002;
Query OK, 1 row affected (0.04 sec)
```

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

```
mysql>
```

THANK YOU