

# DATABASE MANAGEMENT AND SYSTEMS

## LAB – 08

NAME – RAHUL KATINNI

ROLL NO – S20200010091

### QUESTION -1

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> create table WorkCenters(
  -> id int auto_increment primary key,
  -> name varchar(100) not null,
  -> capacity int not null
  -> );
Query OK, 0 rows affected (0.13 sec)

mysql> create table WorkCenterStats(
  -> totalCapacity int not null
  -> );
Query OK, 0 rows affected (0.07 sec)

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 (0.02 sec)

mysql> delimiter ;
mysql> insert into WorkCenters(name,capacity) values('Mold Machine',100);
Query OK, 1 row affected (0.02 sec)

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

```

## QUESTION-2

Write a trigger program to insert a reminder into the reminders table if the birth date of the member is greater than 100 years.

```
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 (0.10 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 (0.09 sec)  
  
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 (0.03 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 (0.01 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  | 1        | Hi John Doe,please update your date of birth. |
+-----+-----+-----+
1 row in set (0.00 sec)
```

## QUESTION-3

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

```
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 (0.07 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 (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```

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 (0.09 sec)

mysql> delimiter $$
mysql> create trigger before_salaries_delete
  -> before delete
  -> on salaries for each row
  -> begin
  -> insert into SalaryArchives(employeeNumber,validFrom,amount)
  -> values(old.employeeNumber,old.validFrom,old.amount);
  -> end $$
Query OK, 0 rows affected (0.05 sec)

mysql> delimiter ;
mysql> delete from salaries
  -> where employeeNumber =1002;
Query OK, 1 row affected (0.02 sec)

mysql> select *from SalaryArchives;
+-----+-----+-----+-----+-----+
| id | employeeNumber | validFrom | amount | deletedAt |
+-----+-----+-----+-----+-----+
| 1 | 1002 | 2000-01-01 | 50000.00 | 2021-10-29 18:33:07 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> delete from salaries;
Query OK, 2 rows affected (0.03 sec)

mysql> select *from SalaryArchives;
+-----+-----+-----+-----+-----+
| id | employeeNumber | validFrom | amount | deletedAt |
+-----+-----+-----+-----+-----+
| 1 | 1002 | 2000-01-01 | 50000.00 | 2021-10-29 18:33:07 |
| 2 | 1056 | 2000-01-01 | 60000.00 | 2021-10-29 18:33:46 |
| 3 | 1076 | 2000-01-01 | 70000.00 | 2021-10-29 18:33:46 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

## QUESTION-4

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

```

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

mysql> CREATE TABLE Salaries (employeeNumber INT PRIMARY KEY,salary DECIMAL(10,2) NOT NULL DEFAULT 0);
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO Salaries(employeeNumber,salary) VALUES (1002,5000),(1056,7000),(1076,8000);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE SalaryBudgets(total DECIMAL(15,2) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO SalaryBudgets(total) SELECT SUM(salary) FROM Salaries;
Query OK, 1 row affected (0.05 sec)
Records: 1 Duplicates: 0 Warnings: 0

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

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 (0.02 sec)

mysql> delete from Salaries
-> where employeeNumber = 1002;
Query OK, 1 row affected (0.05 sec)

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

```

# THANK YOU