Name	Sujal Sandeep Dingankar
UID	2024301005
Experiment No.	9
Aim	**Create Trigger in MySQL.**
Command	Donor Table :-

```
mysql> select * from donor;
+-----+-----------+-----------+------------+---------------------------+-------------+------+--------------+
| id  | firstname | lastname  | contact    | email                     | bloodgroup | Age  | healthstatus |
+-----+-----------+-----------+------------+---------------------------+-------------+------+--------------+
| 101 | Sujal     | Dingankar | 7798802841 | sujal.dingankar@spit.ac.in | O+         |   20 | Healthy      |
| 102 | Shreeya   | Nemade    | 7698231082 | shreeya.nemade@spit.ac.in  | A+         |   19 | Healthy      |
| 103 | Harsha    | Surwase   | 5612798561 | harsha.surwase@spit.ac.in  | B+         |   21 | Healthy      |
| 104 | Avi       | Patil     | 1234598745 | avinash.patil@spit.ac.in   | AB+        |   18 | Healthy      |
| 105 | Shruti    | Bhuvad    | 455612345  | shruti.bhuvad@spit.ac.in   | A+         |   22 | Healthy      |
| 106 | John      | Artino    | 7894561230 | john.doe@example.com       | O+         |   25 | Healthy      |
| 107 | Jane      | Smith     | 7894567890 | jane.smith@example.com     | O+         |   30 | Healthy      |
| 108 | Mark      | Taylor    | 7894563456 | mark.taylor@example.com    | O+         |   19 | Healthy      |
| 109 | Lucy      | Brown     | 7894569876 | lucy.brown@example.com     | A+         |   35 | Healthy      |
| 110 | Manu      | Pandey    | 4561298742 | manu.pandey@spit.ac.in     | AB+        |   26 | Healthy      |
| 111 | Rakesh    | Ramane    | 4531022236 | rakesh.ramane@spit.ac.in   | B+         |   40 | Healthy      |
| 112 | Alice     | Johnson   | 1234567890 | alice.johnson@example.com  | A+         |   30 | Healthy      |
+-----+-----------+-----------+------------+---------------------------+-------------+------+--------------+
12 rows in set (0.00 sec)
```

Donee Table :-

```
mysql> select * from donne;
+-----+------------+----------+------------+-------------------------------+------------+--------------+------------+--------------+
| id  | firstname  | lastname | contact    | email                         | bloodgroup | healthstatus | request_id | request_date |
+-----+------------+----------+------------+-------------------------------+------------+--------------+------------+--------------+
| 101 | Shubham    | Bhuvad   | 7798521123 | shubham.bhuvad@spit.ac.in     | A+         | Healthy      | R001       | 2024-10-05   |
| 102 | Sanjay     | Kadam    | 4568912345 | sanjay.bhuvad@spit.ac.in      | AB+        | Healthy      | R002       | 2024-10-06   |
| 103 | Harshvardan| Nemade   | 777894561  | harshavardan.nemade@spit.ac.in| O+         | Healthy      | R003       | 2024-10-14   |
| 104 | Soham      | Bhojane  | 564781234  | soham.bhojane@spit.ac.in      | A+         | Healthy      | R004       | 2024-10-20   |
| 105 | Shruti     | Bhuvad   | 7798802841 | sujal.dingankar@spit.ac.in    | O+         | Healthy      | R005       | 2024-10-25   |
| 112 | Alice      | Williams | NULL       | NULL                          | O+         | NULL         | NULL       | NULL         |
+-----+------------+----------+------------+-------------------------------+------------+--------------+------------+--------------+
6 rows in set (0.00 sec)
```

Blood Table :-

```
mysql> select * from blood;
+-----+------------+----------+-------------+-------------+
| id  | bloodgroup | quantity | expiry_date | bloodStatus |
+-----+------------+----------+-------------+-------------+
| 101 | A+         |      550 | 2024-10-13  | Available   |
| 102 | B+         |      100 | 2024-10-09  | Available   |
| 103 | AB+        |      150 | 2024-10-15  | Expired     |
| 104 | O+         |      320 | 2024-10-20  | Available   |
| 105 | A+         |      560 | 2024-10-14  | Available   |
+-----+------------+----------+-------------+-------------+
5 rows in set (0.00 sec)
```

Blood_Donation Table :-

```
mysql> select * from blood_donation;
+----+----------+----------+----------+---------------+--------------+
| id | donar_id | blood_id | donee_id | donation_date | receive_date |
+----+----------+----------+----------+---------------+--------------+
|  1 |      101 |      101 |      101 | 2024-10-01    | 2024-10-05   |
|  2 |      102 |      102 |     NULL | 2024-10-15    | NULL         |
|  3 |      103 |      103 |     NULL | 2024-10-20    | NULL         |
|  4 |      104 |      104 |      104 | 2024-10-16    | 2024-10-12   |
|  5 |      101 |      101 |     NULL | 2024-11-04    | NULL         |
|  6 |      101 |      101 |     NULL | 2024-10-10    | NULL         |
|  7 |      102 |      102 |     NULL | 2024-10-10    | NULL         |
|  8 |      102 |      102 |     NULL | 2024-10-10    | NULL         |
|  9 |      102 |      102 |     NULL | 2024-10-10    | NULL         |
| 10 |      102 |      102 |     NULL | 2024-10-10    | NULL         |
| 11 |      102 |      102 |     NULL | 2024-10-11    | NULL         |
+----+----------+----------+----------+---------------+--------------+
11 rows in set (0.00 sec)
```

Create trigger named after_donation_insert is created to automatically update blood quantities after a new donation is recorded.

```
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER after_donation_insert
    -> AFTER INSERT ON blood_donation
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE donor_bloodgroup VARCHAR(10);
    ->
    ->     -- Retrieve the donor's bloodgroup
    ->     SELECT bloodgroup INTO donor_bloodgroup FROM donor WHERE id = NEW.donar_id;
    ->
    ->     -- Update the quantity in the blood table for the donor's bloodgroup
    ->     UPDATE blood
    ->     SET quantity = quantity + 50 -- Assuming each donation is 50 ml
    ->     WHERE bloodgroup = donor_bloodgroup;
    -> END//
Query OK, 0 rows affected (0.01 sec)
```

A test donation is inserted to check if the trigger updates the blood quantity accordingly.

```
mysql> DELIMITER ;
mysql> INSERT INTO blood_donation (donar_id, blood_id, donation_date) VALUES (102, 102, '2024-10-10');
Query OK, 1 row affected (0.01 sec)
```

This query retrieves current records for blood group A+ to verify the update.

```
mysql> SELECT * FROM blood WHERE bloodgroup = 'A+';
+-----+-------------+----------+-------------+-------------+
| id  | bloodgroup  | quantity | expiry_date | bloodStatus |
+-----+-------------+----------+-------------+-------------+
| 101 | A+          |      450 | 2024-10-13  | Available   |
| 105 | A+          |      460 | 2024-10-14  | Available   |
+-----+-------------+----------+-------------+-------------+
2 rows in set (0.00 sec)
```

This query lists blood group A+ records ordered by quantity to see the most recent updates clearly.

```
mysql> SELECT * FROM blood WHERE bloodgroup = 'A+' ORDER BY quantity DESC;
+-----+------------+----------+-------------+-------------+
| id  | bloodgroup | quantity | expiry_date | bloodStatus |
+-----+------------+----------+-------------+-------------+
| 105 | A+         |      460 | 2024-10-14  | Available   |
| 101 | A+         |      450 | 2024-10-13  | Available   |
+-----+------------+----------+-------------+-------------+
2 rows in set (0.00 sec)
```

Create trigger after_request_insert is created to automatically adjust the quantity of blood in the blood table whenever a new request is inserted into the donne table.

A variable requested_quantity is declared and initialized to 50, which represents the amount of blood (in ml) to be deducted for each request.

The UPDATE statement decreases the blood quantity in the blood table by the requested_quantity for the specific blood group related to the newly inserted record in the donne table.

```
mysql> DELIMITER //
mysql> CREATE TRIGGER after_request_insert
    -> AFTER INSERT ON donne
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE requested_quantity INT DEFAULT 50; -- Assuming each request is 50 ml
    ->
    ->     -- Update the blood quantity in the blood table
    ->     UPDATE blood
    ->     SET quantity = quantity - requested_quantity
    ->     WHERE bloodgroup = NEW.bloodgroup;
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)
```

A test record is inserted into the donne table for a blood request made by Rahul Sharma, indicating he needs blood from the A+ group.

```
mysql> INSERT INTO donne (id, firstname, lastname, contact, email, bloodgroup, healthstatus, request_id, request_date)
   -> VALUES (106, 'Rahul', 'Sharma', '7894561230', 'rahul.sharma@example.com', 'A+', 'Healthy', 'R010', '2024-10-15');
Query OK, 1 row affected (0.01 sec)
```

This query retrieves the current records for blood group A+ from the blood table to verify whether the quantity has been updated correctly after the trigger executed due to the insert in the donne table.

```
mysql> SELECT * FROM blood WHERE bloodgroup = 'A+';
+-----+------------+----------+-------------+-------------+
| id  | bloodgroup | quantity | expiry_date | bloodStatus |
+-----+------------+----------+-------------+-------------+
| 101 | A+         |      500 | 2024-10-13  | Available   |
| 105 | A+         |      510 | 2024-10-14  | Available   |
+-----+------------+----------+-------------+-------------+
2 rows in set (0.00 sec)
```

The after_donation_delete trigger is created to automatically adjust the quantity of blood in the blood table whenever a record is deleted from the blood_donation table.

The variable donated_bloodgroup stores the blood group of the donor associated with the deleted donation, while donated_quantity is initialized to 50 ml, representing the amount of blood donated.

The trigger retrieves the donor's blood group using a SELECT statement and updates the blood quantity in the blood table by subtracting the donated_quantity for that specific blood group.

```
mysql> DELIMITER //
mysql> CREATE TRIGGER after_donation_delete
    -> AFTER DELETE ON blood_donation
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE donated_bloodgroup VARCHAR(20);
    ->     DECLARE donated_quantity INT DEFAULT 50;
    ->
    ->     -- Retrieve the donor's blood group
    ->     SELECT bloodgroup INTO donated_bloodgroup
    ->     FROM donor
    ->     WHERE id = OLD.donar_id;
    ->
    ->     -- Decrease the blood quantity in the blood table
    ->     UPDATE blood
    ->     SET quantity = quantity - donated_quantity
    ->     WHERE bloodgroup = donated_bloodgroup;
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)
```

This command deletes a specific record from the blood_donation table (in this case, with ID = 1), simulating the removal of a blood donation.

```
mysql> DELETE FROM blood_donation WHERE id = 1;
Query OK, 1 row affected (0.01 sec)
```

This query retrieves the current records for blood group O+ from the blood table to verify whether the quantity has been adjusted correctly after the trigger executed due to the deletion in the blood_donation table.

```
mysql> SELECT * FROM blood WHERE bloodgroup = 'O+';
+-----+-----------+----------+-------------+-------------+
| id  | bloodgroup | quantity | expiry_date | bloodStatus |
+-----+-----------+----------+-------------+-------------+
| 104 | O+         |      270 | 2024-10-20  | Available   |
+-----+-----------+----------+-------------+-------------+
1 row in set (0.00 sec)
```

	Finally we drop udpate delete trigger.
	`mysql> Drop trigger after_donation_delete;`
	`Query OK, 0 rows affected (0.01 sec)`

Conclusion	From this experiment, I have learned about creating triggers and perform operations on it in MySQL & also implemented them.