Name : Tazmeen Afroz
Roll No: 22p-9252
Class & Section : BAI-4A
LAB-TASK-9

Create a database name person and create the tables above.

```
CREATE DATABASE person;
USE person;
```

Table Creation

```
CREATE TABLE Users (
  id INT AUTO_INCREMENT  PRIMARY KEY,
  user_id INT,
  username VARCHAR(255),
  password VARCHAR(255),
  email VARCHAR(255)
);
```

```
MariaDB [(none)]> CREATE DATABASE person;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE person;
Database changed
MariaDB [person]>
MariaDB [person]> CREATE TABLE Users (
    ->   id INT AUTO_INCREMENT  PRIMARY KEY,
    ->   user_id INT,
    ->   username VARCHAR(255),
    ->   password VARCHAR(255),
    ->   email VARCHAR(255)
    -> ) ;
Query OK, 0 rows affected (0.038 sec)
```

```
CREATE TABLE Summary (
  id INT AUTO_INCREMENT PRIMARY KEY,
  total_user INT,
  Yahoo INT,
  Hotmail INT,
  Gmail INT
);
```

```
MariaDB [person]> CREATE TABLE Summary (
    ->   id INT AUTO_INCREMENT PRIMARY KEY,
    ->   total_user INT,
    ->   Yahoo INT,
    ->   Hotmail INT,
    ->   Gmail INT
    -> ) ;
Query OK, 0 rows affected (0.033 sec)
```

Procedure 1: Insert into Summary

```
DELIMITER //

CREATE PROCEDURE insert_summary (
  IN id INT ,
  IN total_user INT,
  IN Yahoo INT,
  IN Hotmail INT,
  IN Gmail INT
)
BEGIN
  INSERT INTO Summary (id, total_user, Yahoo, Hotmail, Gmail)
  VALUES (id, total_user, Yahoo, Hotmail, Gmail);
END//

DELIMITER ;
```

```
MariaDB [person]> DELIMITER //
MariaDB [person]> CREATE PROCEDURE insert_summary (
    ->   IN id INT ,
    ->   IN total_user INT,
    ->   IN Yahoo INT,
    ->   IN Hotmail INT,
    ->   IN Gmail INT
    -> )
    -> BEGIN
    ->   INSERT INTO Summary (id, total_user, Yahoo, Hotmail, Gmail)
    ->   VALUES (id, total_user, Yahoo, Hotmail, Gmail);
    -> END//
Query OK, 0 rows affected (0.018 sec)
```

Procedure 2: Insert into Users

```
DELIMITER //

CREATE PROCEDURE insert_users(
  IN user_id INT,
  IN username VARCHAR(255),
  IN password VARCHAR(255),
  IN email VARCHAR(255)
)
BEGIN
  INSERT INTO Users (user_id, username, password, email)
  VALUES (user_id, username, password, email);
END//

DELIMITER ;
```

```
MariaDB [person]> CREATE PROCEDURE insert_users(
    ->   IN user_id INT,
    ->   IN username VARCHAR(255),
    ->   IN password VARCHAR(255),
    ->   IN email VARCHAR(255)
    -> )
    -> BEGIN
    ->   INSERT INTO Users (user_id, username, password, email)
    ->   VALUES (user_id, username, password, email);
    -> END//
Query OK, 0 rows affected (0.020 sec)
```

Procedure 3: Average Yahoo

```
DELIMITER //

CREATE PROCEDURE AVG_YAHOO(OUT  avg_yahoo decimal)
BEGIN
  SELECT AVG(Yahoo) INTO avg_yahoo
  FROM Summary;
END//

DELIMITER ;
```

Procedure 4: Min Gmail

```
DELIMITER //

CREATE PROCEDURE MIN_GMAIL(OUT  min_gmail decimal)
BEGIN
  SELECT MIN(Gmail) INTO min_gmail
  FROM Summary;
END//

DELIMITER ;
```

Procedure 5: Max Hotmail

```
DELIMITER //

CREATE PROCEDURE MAX_HOTMAIL(OUT  max_hotmail decimal)
BEGIN
  SELECT MAX(Hotmail) INTO max_hotmail
  FROM Summary;
END//

DELIMITER ;
```

```
MariaDB [person]> CREATE PROCEDURE AVG_YAHOO(OUT  avg_yahoo decimal)
    -> BEGIN
    ->   SELECT AVG(Yahoo) INTO avg_yahoo
    ->   FROM Summary;
    -> END//
Query OK, 0 rows affected (0.014 sec)

MariaDB [person]> CREATE PROCEDURE MIN_GMAIL(OUT  min_gmail decimal)
    -> BEGIN
    ->   SELECT MIN(Gmail) INTO min_gmail
    ->   FROM Summary;
    -> END//
Query OK, 0 rows affected (0.019 sec)

MariaDB [person]> CREATE PROCEDURE MAX_HOTMAIL(OUT  max_hotmail decimal)
    -> BEGIN
    ->   SELECT MAX(Hotmail) INTO max_hotmail
    ->   FROM Summary;
    -> END//
Query OK, 0 rows affected (0.020 sec)
```

Procedure 6: Increment Total User

```
DELIMITER //

CREATE PROCEDURE INC_TOTAL_USER(
  IN Yahoo INT,
  IN Hotmail INT
)
BEGIN
UPDATE summary
    SET total_user = total_user + 1
    WHERE Yahoo <= Hotmail;
END//

DELIMITER ;
```

```
MariaDB [person]> CREATE PROCEDURE INC_TOTAL_USER(
    ->   IN Yahoo INT,
    ->   IN Hotmail INT
    -> )
    -> BEGIN
    -> UPDATE summary
    ->     SET total_user = total_user + 1
    ->     WHERE Yahoo <= Hotmail;
    -> END//
Query OK, 0 rows affected (0.015 sec)
```

Trigger 1: Insert into Users

```
DELIMITER //

CREATE TRIGGER tr_insert_users
AFTER INSERT ON Users
FOR EACH ROW
BEGIN
  UPDATE Summary
```

```
  SET total_user = total_user + 1;
END//

DELIMITER ;
```

Trigger 2: Delete from Users

```
DELIMITER //

CREATE TRIGGER tr_delete_users
AFTER DELETE ON Users
FOR EACH ROW
BEGIN
  UPDATE Summary
  SET total_user = total_user - 1;
END//

DELIMITER ;
```

```
MariaDB [person]> CREATE TRIGGER tr_insert_users
    -> AFTER INSERT ON Users
    -> FOR EACH ROW
    -> BEGIN
    ->   UPDATE Summary
    ->   SET total_user = total_user + 1;
    -> END//
Query OK, 0 rows affected (0.015 sec)

MariaDB [person]> CREATE TRIGGER tr_delete_users
    -> AFTER DELETE ON Users
    -> FOR EACH ROW
    -> BEGIN
    ->   UPDATE Summary
    ->   SET total_user = total_user - 1;
    -> END//
Query OK, 0 rows affected (0.020 sec)
```

Insert data in the users table (at least 5-10 records) with different email addresses using the stored procedure created in (Exercise 2).
Check the summary table each time you add a new user or delete a user to check if the trigger(s) are working correctly.

```
CALL insert_summary(0,0,0,0,0);

CALL insert_users(1, 'john_doe', 'password123', 'john@yahoo.com');
CALL insert_users(2, 'jane_smith', 'qwerty123', 'jane@hotmail.com');
CALL insert_users(3, 'bob_johnson', 'abc123', 'bob@gmail.com');
CALL insert_users(4, 'sarah_wilson', 'pass456', 'sarah@yahoo.com');
CALL insert_users(5, 'mike_brown', 'password789', 'mike@hotmail.com');
CALL insert_users(6, 'emily_taylor', 'qwerty456', 'emily@gmail.com');
CALL insert_users(7, 'david_anderson', 'letmein123', 'david@yahoo.com');
```

CALL insert_users(8, 'jessica_thompson', 'password987', 'jessica@hotmail.com');
CALL insert_users(9, 'michael_jackson', 'abc456', 'michael@gmail.com');
CALL insert_users(10, 'ashley_garcia', 'pass123', 'ashley@yahoo.com');

```
MariaDB [person]> CALL insert_users(1, 'john_doe', 'password123', 'john@yahoo.com');
    -> CALL insert_users(2, 'jane_smith', 'qwerty123', 'jane@hotmail.com');
    -> CALL insert_users(3, 'bob_johnson', 'abc123', 'bob@gmail.com');
    -> CALL insert_users(4, 'sarah_wilson', 'pass456', 'sarah@yahoo.com');
    -> CALL insert_users(5, 'mike_brown', 'password789', 'mike@hotmail.com');
    -> CALL insert_users(6, 'emily_taylor', 'qwerty456', 'emily@gmail.com');
    -> CALL insert_users(7, 'david_anderson', 'letmein123', 'david@yahoo.com');
    -> CALL insert_users(8, 'jessica_thompson', 'password987', 'jessica@hotmail.com');
    -> CALL insert_users(9, 'michael_jackson', 'abc456', 'michael@gmail.com');
    -> CALL insert_users(10, 'ashley_garcia', 'pass123', 'ashley@yahoo.com');
    -> //
Query OK, 1 row affected (0.009 sec)

Query OK, 1 row affected (0.015 sec)

Query OK, 1 row affected (0.033 sec)

Query OK, 1 row affected (0.039 sec)

Query OK, 1 row affected (0.048 sec)

Query OK, 1 row affected (0.054 sec)

Query OK, 1 row affected (0.064 sec)

Query OK, 1 row affected (0.070 sec)

Query OK, 1 row affected (0.076 sec)

Query OK, 1 row affected (0.082 sec)
```

SELECT * FROM Summary;
SELECT * FROM users;

```
MariaDB [person]> SELECT * FROM Summary;
+----+------------+-------+---------+-------+
| id | total_user | Yahoo | Hotmail | Gmail |
+----+------------+-------+---------+-------+
|  1 |         10 |     0 |       0 |     0 |
+----+------------+-------+---------+-------+
1 row in set (0.000 sec)

MariaDB [person]> SELECT * FROM users;
ERROR 1146 (42S02): Table 'person.users' doesn't exist
MariaDB [person]> SELECT * FROM Users;
+----+---------+------------------+-------------+---------------------+
| id | user_id | username         | password    | email               |
+----+---------+------------------+-------------+---------------------+
|  1 |       1 | john_doe         | password123 | john@yahoo.com      |
|  2 |       2 | jane_smith       | qwerty123   | jane@hotmail.com    |
|  3 |       3 | bob_johnson      | abc123      | bob@gmail.com       |
|  4 |       4 | sarah_wilson     | pass456     | sarah@yahoo.com     |
|  5 |       5 | mike_brown       | password789 | mike@hotmail.com    |
|  6 |       6 | emily_taylor     | qwerty456   | emily@gmail.com     |
|  7 |       7 | david_anderson   | letmein123  | david@yahoo.com     |
|  8 |       8 | jessica_thompson | password987 | jessica@hotmail.com |
|  9 |       9 | michael_jackson  | abc456      | michael@gmail.com   |
| 10 |      10 | ashley_garcia    | pass123     | ashley@yahoo.com    |
+----+---------+------------------+-------------+---------------------+
10 rows in set (0.001 sec)
```

DELETE FROM Users WHERE id = 5;
SELECT * FROM Summary;

```
MariaDB [person]> DELETE FROM Users WHERE id = 5;
Query OK, 1 row affected (0.009 sec)

MariaDB [person]> SELECT * FROM Summary;
+----+------------+-------+---------+-------+
| id | total_user | Yahoo | Hotmail | Gmail |
+----+------------+-------+---------+-------+
|  1 |          9 |     0 |       0 |     0 |
+----+------------+-------+---------+-------+
1 row in set (0.001 sec)
```

TASK PRO MAX : Create a trigger that restricts only view to make updates to the product table. Direct updates to the product table must be allowed. (Hint Use IF statement)

```
 USE sales_co;
CREATE VIEW p_view AS
SELECT * FROM PRODUCT;
DELIMITER //
CREATE TRIGGER prevent_view_update
BEFORE UPDATE ON PRODUCT
FOR EACH ROW
BEGIN
    IF @update_from_view = 1 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Updates from views are not allowed on this table.';
    END IF;
END//
DELIMITER ;
SET @update_from_view := 1;
    UPDATE p_view SET P_PRICE = 100 WHERE P_MIN > 20;
SET @update_from_view := NULL;
UPDATE PRODUCT SET P_PRICE = 100 WHERE P_MIN > 20;
```

```
MariaDB [sales_co]> SET @update_from_view := 1;
    -> UPDATE p_view SET P_PRICE = 100 WHERE P_MIN > 20;
    -> SET @update_from_view := NULL;
    -> //
Query OK, 0 rows affected (0.000 sec)

ERROR 1644 (45000): Updates from views are not allowed on this table.
```

```
MariaDB [sales_co]> UPDATE PRODUCT SET P_PRICE = 100 WHERE P_MIN > 20;
    -> //
Query OK, 3 rows affected (0.008 sec)
Rows matched: 3  Changed: 3  Warnings: 0

MariaDB [sales_co]>
```