

Name: Amber Khurshid

Section: BAI-4A

Roll No: 22P-9295

Lab Task #08

Create a database name person and create the tables above.

Create database person;

Use person;

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

```
mysql> create database person;
Query OK, 1 row affected (0.01 sec)

mysql> use person
Database changed
mysql> 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 (0.02 sec)
```

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

```
mysql> CREATE TABLE Summary (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> total_user INT,
    -> Yahoo INT,
    -> Hotmail INT,
    -> Gmail INT
    -> );
Query OK, 0 rows affected (0.01 sec)
```

1. Write a procedure that take id, total_user, Yahoo, Hotmail, Gmail values as an input and insert the data into the table 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 ;

```
mysql> DELIMITER #
mysql> CREATE PROCEDURE insert_summary ( IN id INT , IN total_user INT, IN Yahoo INT, IN Hotmail INT, IN Gmail INT ) BEG
IN INSERT INTO Summary (id, total_user, Yahoo, Hotmail, Gmail) VALUES (id, total_user, Yahoo, Hotmail, Gmail); END#
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

2. Write a procedure that take user_id, username, password, email values as an input and insert the data into the table 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 ;

```
mysql> DELIMITER #
mysql> 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#
mysql>
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

3. Write a procedure that output the average value of yahoo

DELIMITER #

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

DELIMITER ;

```
mysql> DELIMITER #
mysql> CREATE PROCEDURE AVG_YAHOO(OUT avg_yahoo decimal) BEGIN SELECT AVG(Yahoo) INTO avg_yahoo FROM Summary; END#
mysql>
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

4. Write a procedure that output the min value of Gmail.

DELIMITER #

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

DELIMITER ;

```
mysql> DELIMITER #
mysql> CREATE PROCEDURE MIN_GMAIL(OUT min_gmail decimal) BEGIN SELECT MIN(Gmail) INTO min_gmail FROM Summary; END#
mysql>
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

5. Write a procedure that output the max value of Hotmail.

DELIMITER #

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

DELIMITER ;

```
mysql> DELIMITER #
mysql> CREATE PROCEDURE MAX_HOTMAIL(OUT max_hotmail decimal) BEGIN SELECT MAX(Hotmail) INTO max_hotmail FROM Summary; END#
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

6. Write a procedure that increment the total user by 1 when yahoo value is less than equal to Hotmail.

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 ;

```
mysql> DELIMITER #
mysql> 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 (0.01 sec)

mysql> DELIMITER ;
mysql>
```

7. Create trigger(s) to update the records in the summary table. The summary table will contain only one record and will be updated each time a new entry is made to the users table, or a user is deleted from the users table. You need to create the following two triggers.

I. Whenever a new entry is made to the user's table then total_user attribute in the summary table must be incremented by 1.

DELIMITER #

CREATE TRIGGER tr_insert_users AFTER INSERT ON Users FOR EACH ROW BEGIN UPDATE Summary SET total_user = total_user + 1; END#

DELIMITER ;

```
mysql> DELIMITER #
mysql> 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 (0.02 sec)

mysql> DELIMITER ;
mysql>
```

II. Whenever a record is deleted from the user's table then total_user attribute in the summary table must be decremented by 1.

DELIMITER #

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

DELIMITER ;

```
mysql> DELIMITER #
mysql> 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 (0.01 sec)

mysql> DELIMITER ;
mysql>
```

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(5, 'amina', 'amina123', 'amina@hotmail.com');
```

```
CALL insert_users(6, 'john', 'j987', 'john@gmail.com');
```

```
CALL insert_users(1, 'amber_kh', 'am123', 'amber@yahoo.com');
```

```
CALL insert_users(2, 'ayesha', 'ayesh123', 'ayesha@hotmail.com');
```

```
CALL insert_users(7, 'anderson', 'anderson123', 'anderson@yahoo.com');
```

```
CALL insert_users(8, 'jemima', 'password123', 'jemima@hotmail.com');
```

```
CALL insert_users(9, 'mashaal', 'xyz456', 'mashaal@gmail.com');
```

```
CALL insert_users(10, 'arisha', 'aris123', 'arisha@yahoo.com');
```

```
CALL insert_users(3, 'alina', 'pass123', 'ayesha@gmail.com');
```

```
CALL insert_users(4, 'sana', 'sana456', 'sana@yahoo.com');
```

```
mysql> CALL insert_summary(0,0,0,0,0);
Query OK, 1 row affected (0.01 sec)

mysql> CALL insert_users(5, 'amina', 'amina123', 'amina@hotmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> CALL insert_users(6, 'john', 'j987', 'john@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(1, 'amber_kh', 'am123', 'amber@yahoo.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(2, 'ayesha', 'ayesh123', 'ayesha@hotmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(7, 'anderson', 'anderson123', 'anderson@yahoo.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(8, 'jemima', 'password123', 'jemima@hotmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(9, 'mashaal', 'xyz456', 'mashaal@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(10, 'arisha', 'aris123', 'arisha@yahoo.com');
Query OK, 1 row affected (0.00 sec)

mysql> CALL insert_users(3, 'alina', 'pass123', 'ayesha@gmail.com');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> CALL insert_users(4, 'sana', 'sana456', 'sana@yahoo.com');
Query OK, 1 row affected (0.00 sec)
```

SELECT * FROM Summary;

```
mysql> SELECT * FROM Summary;
+----+-----+-----+-----+-----+
| id | total_user | Yahoo | Hotmail | Gmail |
+----+-----+-----+-----+-----+
|  1 |          10 |      0 |        0 |      0 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

SELECT * FROM users;

```
mysql> SELECT * FROM users;
```

id	user_id	username	password	email
1	5	amina	amina123	amina@hotmail.com
2	6	john	j987	john@gmail.com
3	1	amber_kh	am123	amber@yahoo.com
4	2	ayesha	ayesh123	ayesha@hotmail.com
5	7	anderson	anderson123	anderson@yahoo.com
6	8	jemima	password123	jemima@hotmail.com
7	9	mashaal	xyz456	mashaal@gmail.com
8	10	arisha	aris123	arisha@yahoo.com
9	3	alina	pass123	ayesha@gmail.com
10	4	sana	sana456	sana@yahoo.com

```
10 rows in set (0.00 sec)

mysql>
```

DELETE FROM Users WHERE id = 8;

SELECT * FROM Summary;

```
mysql> DELETE FROM Users WHERE id = 8;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Summary;
```

id	total_user	Yahoo	Hotmail	Gmail
1	9	0	0	0

```
1 row in set (0.00 sec)

mysql>
```

TASK PRO MAXTask: 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;
```



```
mysql> CREATE VIEW p_view AS
-> SELECT * FROM PRODUCT;
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER #
mysql> 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#
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> SET @update_from_view := 1;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE p_view SET P_PRICE = 100 WHERE P_MIN > 20;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> SET @update_from_view := NULL;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE PRODUCT SET P_PRICE = 100 WHERE P_MIN > 20;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```