Database Management Systems: Lab 6

Name: Rithvik Rajesh Matta
SRN: PES2UG23CS485
Section: H

Task 1:

1. Create a trigger that automatically decreases the total_quantity of an item in the stall_items table whenever a new row is inserted into the purchased table.

   Before:

```
mysql> SELECT * FROM item WHERE ItemID = 1;
+--------+----------+------+--------+---------+----------+
| ItemID | ItemName | Type | Price  | StallID | Quantity |
+--------+----------+------+--------+---------+----------+
|      1 | Burger   | Veg  | 100.00 |       1 |       10 |
+--------+----------+------+--------+---------+----------+
1 row in set (0.00 sec)
```

   Command:

```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER decrease_item_quantity
    -> AFTER INSERT ON purchase
    -> FOR EACH ROW
    -> BEGIN
    ->     UPDATE item
    ->     SET Quantity = Quantity - NEW.Quantity
    ->     WHERE ItemID = NEW.ItemID;
    -> END $$
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> DELIMITER ;
```

   After:

```
mysql> INSERT INTO purchase (PurchaseID, ParticipantSRN, StallID, ItemID, Quantity, PurchaseTime)
    -> VALUES (201, 'SRN001', 1, 1, 2, NOW());
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM item WHERE ItemID = 1;
+--------+----------+------+--------+---------+----------+
| ItemID | ItemName | Type | Price  | StallID | Quantity |
+--------+----------+------+--------+---------+----------+
|      1 | Burger   | Veg  | 100.00 |       1 |        8 |
+--------+----------+------+--------+---------+----------+
1 row in set (0.00 sec)
```

2. Create a trigger that prevents a participant from purchasing more than 5 units of any single item in a single transaction (i.e., quantity > 5) in the purchased table.

Before:

```
mysql> SELECT * FROM purchase;
+------------+----------------+---------+--------+----------+---------------------+
| PurchaseID | ParticipantSRN | StallID | ItemID | Quantity | PurchaseTime        |
+------------+----------------+---------+--------+----------+---------------------+
|        201 | SRN001         |       1 |      1 |        2 | 2025-09-19 17:53:38 |
|        801 | P1001          |       1 |    701 |        2 | 2025-07-09 10:30:00 |
|        802 | P1002          |       6 |    702 |        1 | 2025-07-09 11:15:00 |
|        803 | P1017          |       6 |    702 |        3 | 2025-07-10 14:00:00 |
+------------+----------------+---------+--------+----------+---------------------+
4 rows in set (0.00 sec)
```

Command:

```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER prevent_large_purchase
    -> BEFORE INSERT ON purchase
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.Quantity > 5 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = 'Purchase of more than 5 units of a single item is not allowed';
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> DELIMITER ;
```

After:

```
mysql> INSERT INTO purchase (PurchaseID, ParticipantSRN, StallID, ItemID, Quantity, PurchaseTime)
    -> VALUES (301, 'SRN001', 1, 1, 3, NOW());
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO purchase (PurchaseID, ParticipantSRN, StallID, ItemID, Quantity, PurchaseTime)
    -> VALUES (302, 'SRN001', 1, 1, 6, NOW());
ERROR 1644 (45000): Purchase of more than 5 units of a single item is not allowed
```

Task 2:

1. Write a stored procedure GetStallSales that takes a stall_id as input and prints the total revenue generated from that stall based on the purchased table and item prices from stall_items.

Before:

```
mysql> SELECT * FROM item LIMIT 5;
+--------+------------------+---------+--------+---------+----------+
| ItemID | ItemName         | Type    | Price  | StallID | Quantity |
+--------+------------------+---------+--------+---------+----------+
|      1 | Burger           | Veg     | 100.00 |       1 |        5 |
|    701 | Mushroom Risotto | Veg     | 120.00 |       1 |       25 |
|    702 | Fish Tacos       | Non-Veg | 150.00 |       6 |       40 |
+--------+------------------+---------+--------+---------+----------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM purchase LIMIT 5;
+------------+----------------+---------+--------+----------+---------------------+
| PurchaseID | ParticipantSRN | StallID | ItemID | Quantity | PurchaseTime        |
+------------+----------------+---------+--------+----------+---------------------+
|        201 | SRN001         |       1 |      1 |        2 | 2025-09-19 17:53:38 |
|        301 | SRN001         |       1 |      1 |        3 | 2025-09-19 17:56:34 |
|        801 | P1001          |       1 |    701 |        2 | 2025-07-09 10:30:00 |
|        802 | P1002          |       6 |    702 |        1 | 2025-07-09 11:15:00 |
|        803 | P1017          |       6 |    702 |        3 | 2025-07-10 14:00:00 |
+------------+----------------+---------+--------+----------+---------------------+
5 rows in set (0.00 sec)
```

Command:

```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetStallSales(IN input_stall_id INT)
    -> BEGIN
    ->     SELECT i.StallID,
    ->            SUM(i.Price * p.Quantity) AS TotalRevenue
    ->     FROM purchase p
    ->     JOIN item i ON p.ItemID = i.ItemID
    ->     WHERE i.StallID = input_stall_id
    ->     GROUP BY i.StallID;
    -> END $$
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> DELIMITER ;
```

After:

```
mysql> CALL GetStallSales(1);
+---------+--------------+
| StallID | TotalRevenue |
+---------+--------------+
|       1 |       740.00 |
+---------+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

2. Create a procedure RegisterParticipant that registers a participant (SRN) for an event (event_id) by inserting into the registration table. The procedure should take the event_id, SRN, and registration_id as parameters.

Before:

```
mysql> SELECT * FROM registration LIMIT 5;
+-------+---------+----------------+
| RegNo | EventID | ParticipantSRN |
+-------+---------+----------------+
|   503 |       2 | P1001          |
|   507 |       2 | P1003          |
|   505 |       5 | P1001          |
|   506 |       5 | P1002          |
+-------+---------+----------------+
4 rows in set (0.02 sec)
```

Command:

```
mysql> CREATE PROCEDURE RegisterParticipant (
    ->     IN p_event_id INT,
    ->     IN p_srn VARCHAR(20),
    ->     IN p_regno INT
    -> )
    -> BEGIN
    ->     INSERT INTO registration (RegNo, EventID, ParticipantSRN)
    ->     VALUES (p_regno, p_event_id, p_srn);
    -> END $$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

After:

```
mysql> CALL RegisterParticipant(101, 'SRN001', 6001);
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> SELECT * FROM registration WHERE RegNo = 6001;
+-------+---------+----------------+
| RegNo | EventID | ParticipantSRN |
+-------+---------+----------------+
|  6001 |     101 | SRN001         |
+-------+---------+----------------+
1 row in set (0.00 sec)
```

Task 3:

1. Create a function GetEventCost that accepts an event_id and returns the total price a participant would pay to register for that event (i.e., return the event's price from the event table).

   Before:

```
mysql> SELECT * FROM event_schedule LIMIT 5;
+-------------------+---------+---------------------+--------------+-------+--------+--------+--------+
| Date_of_conduction | EventID | EventName           | Block        | Floor | RoomNo | Price  | TeamID |
+-------------------+---------+---------------------+--------------+-------+--------+--------+--------+
| 2025-07-15        |       1 | Programming Workshop | B-Block      |     2 | 201    | 900.00 |      1 |
| 2025-07-16        |       2 | Coding Contest       | C-Block      |     1 | 102    | 500.00 |      2 |
| 2048-07-18        |       5 | Robotics Challenge   | D-Block      |     3 | 301    | 700.00 |      1 |
| 2025-09-02        |       6 | AI Hackathon         | Seminar Hall |     2 | 205    | 900.00 |      4 |
| 2025-09-20        |     101 | Test Event           | A            |     1 | 101    |   0.00 |   NULL |
+-------------------+---------+---------------------+--------------+-------+--------+--------+--------+
5 rows in set (0.00 sec)
```

   Command:

```
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION GetEventCost(input_event_id INT)
    -> RETURNS DECIMAL(6,2)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE event_price DECIMAL(6,2);
    ->
    ->     SELECT Price INTO event_price
    ->     FROM event_schedule
    ->     WHERE EventID = input_event_id;
    ->
    ->     RETURN event_price;
    -> END $$
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> DELIMITER ;
```

   After:

```
mysql> SELECT GetEventCost(1) AS EventCost;
+-----------+
| EventCost |
+-----------+
|    900.00 |
+-----------+
1 row in set (0.00 sec)
```

2. Create a function GetParticipantPurchaseTotal(SRN) that returns the total amount a participant has spent across all stalls.

Before:

```
mysql> SELECT * FROM purchase LIMIT 5;
+------------+----------------+---------+--------+----------+---------------------+
| PurchaseID | ParticipantSRN | StallID | ItemID | Quantity | PurchaseTime        |
+------------+----------------+---------+--------+----------+---------------------+
|        201 | SRN001         |       1 |      1 |        2 | 2025-09-19 17:53:38 |
|        301 | SRN001         |       1 |      1 |        3 | 2025-09-19 17:56:34 |
|        801 | P1001          |       1 |    701 |        2 | 2025-07-09 10:30:00 |
|        802 | P1002          |       6 |    702 |        1 | 2025-07-09 11:15:00 |
|        803 | P1017          |       6 |    702 |        3 | 2025-07-10 14:00:00 |
+------------+----------------+---------+--------+----------+---------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM item LIMIT 5;
+--------+------------------+---------+--------+---------+----------+
| ItemID | ItemName         | Type    | Price  | StallID | Quantity |
+--------+------------------+---------+--------+---------+----------+
|      1 | Burger           | Veg     | 100.00 |       1 |        5 |
|    701 | Mushroom Risotto | Veg     | 120.00 |       1 |       25 |
|    702 | Fish Tacos       | Non-Veg | 150.00 |       6 |       40 |
+--------+------------------+---------+--------+---------+----------+
3 rows in set (0.00 sec)
```

Command:

```
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION GetParticipantPurchaseTotal(input_srn VARCHAR(20))
    -> RETURNS DECIMAL(10,2)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE total_spent DECIMAL(10,2);
    ->
    ->     SELECT SUM(i.Price * p.Quantity) INTO total_spent
    ->     FROM purchase p
    ->     JOIN item i ON p.ItemID = i.ItemID
    ->     WHERE p.ParticipantSRN = input_srn;
    ->
    ->     RETURN IFNULL(total_spent, 0);
    -> END $$
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> DELIMITER ;
```

After:

```
mysql> SELECT GetParticipantPurchaseTotal('SRN001') AS TotalSpent;
+-------------+
| TotalSpent  |
+-------------+
|      500.00 |
+-------------+
1 row in set (0.00 sec)
```