Laporan Praktikum 8

Nama : Muhammad ‘Azmi Salam
NIM : 2406010
Kelas : C2 - 2024

1. Bikin Trigger

```
DELIMITER //

CREATE TRIGGER trg_after_insert_order_item
AFTER INSERT ON order_items
FOR EACH ROW
BEGIN
  DECLARE item_price INT;

  -- Ambil harga item dari menu_items
  SELECT price INTO item_price
  FROM menu_items
  WHERE item_id = NEW.item_id;

  -- Update total_amount di tabel orders
  UPDATE orders
  SET total_amount = total_amount + (item_price * NEW.quantity)
  WHERE order_id = NEW.order_id;
END;
//

DELIMITER ;
```

```
DELIMITER //

CREATE TRIGGER trg_after_insert_order_item
AFTER INSERT ON order_items
FOR EACH ROW
BEGIN
  DECLARE item_price INT;

  -- Ambil harga item dari menu_items
  SELECT price INTO item_price
  FROM menu_items
```

```
  WHERE item_id = NEW.item_id;

  -- Update total_amount di tabel orders
  UPDATE orders
  SET total_amount = total_amount + (item_price * NEW.quantity)
  WHERE order_id = NEW.order_id;
END;
//

DELIMITER ;
```

```
MariaDB [black_beans]> SELECT total_amount FROM orders WHERE order_id = 1;
+--------------+
| total_amount |
+--------------+
|     58500.00 |
+--------------+
1 row in set (0.018 sec)

MariaDB [black_beans]> INSERT INTO order_items (order_id, item_id, quantity)
    -> VALUES (1, 2, 3);  -- order_id = 1, item_id = 2, quantity = 3
Query OK, 1 row affected (0.048 sec)

MariaDB [black_beans]> SELECT total_amount FROM orders WHERE order_id = 1;
+--------------+
| total_amount |
+--------------+
|    148500.00 |
+--------------+
1 row in set (0.001 sec)

MariaDB [black_beans]>
```

2. Bikin Stored Function

```
DELIMITER //

CREATE FUNCTION get_best_seller_by_category(input_category_id INT)
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
   DECLARE best_seller_name VARCHAR(255);

   SELECT i.name
   INTO best_seller_name
   FROM menu_items i
   JOIN order_items oi ON i.item_id = oi.item_id
   WHERE i.category_id = input_category_id
```

```
    GROUP BY i.item_id
    ORDER BY SUM(oi.quantity) DESC
    LIMIT 1;

    RETURN best_seller_name;
END //

DELIMITER ;
```

```
MariaDB [black_beans]> SELECT get_best_seller_by_category(3) AS best_seller;
+------------------+
| best_seller      |
+------------------+
| Chocolate Muffin |
+------------------+
1 row in set (0.005 sec)

MariaDB [black_beans]>
```

```
MariaDB [black_beans]> SELECT get_best_seller_by_category(3) AS best_seller;
+------------------+
| best_seller      |
+------------------+
| Chocolate Muffin |
+------------------+
1 row in set (0.005 sec)

MariaDB [black_beans]> SELECT get_best_seller_by_category(1) AS best_seller;
+-------------+
| best_seller |
+-------------+
| Latte       |
+-------------+
1 row in set (0.002 sec)

MariaDB [black_beans]> SELECT get_best_seller_by_category(2) AS best_seller;
+-------------+
| best_seller |
+-------------+
| Black Tea   |
+-------------+
1 row in set (0.001 sec)

MariaDB [black_beans]>
```

3. Bikin Prosedur

```
DELIMITER //

CREATE PROCEDURE get_orders_and_revenue_by_date(IN input_date DATE)
BEGIN
    -- Tampilkan daftar order pada tanggal tersebut
    SELECT
        o.order_id,
```

```
    o.order_date,
    i.name AS item_name,
    oi.quantity,
    (oi.quantity * i.price) AS subtotal
  FROM orders o
  JOIN order_items oi ON o.order_id = oi.order_id
  JOIN menu_items i ON oi.item_id = i.item_id
  WHERE DATE(o.order_date) = input_date;

  -- Tampilkan total omzet pada tanggal tersebut
  SELECT
    SUM(oi.quantity * i.price) AS total_revenue
  FROM orders o
  JOIN order_items oi ON o.order_id = oi.order_id
  JOIN menu_items i ON oi.item_id = i.item_id
  WHERE DATE(o.order_date) = input_date;
END //

DELIMITER ;
```

```
MariaDB [black_beans]> CALL get_orders_and_revenue_by_date('2025-04-06');
+----------+---------------------+------------------+----------+----------+
| order_id | order_date          | item_name        | quantity | subtotal |
+----------+---------------------+------------------+----------+----------+
|        1 | 2025-04-06 09:30:00 | Espresso         |        1 | 25000.00 |
|        1 | 2025-04-06 09:30:00 | Croissant        |        1 | 20000.00 |
|        1 | 2025-04-06 09:30:00 | Latte            |        3 | 90000.00 |
|        2 | 2025-04-06 14:00:00 | Latte            |        1 | 30000.00 |
|        2 | 2025-04-06 14:00:00 | Green Tea        |        1 | 28000.00 |
|        3 | 2025-04-06 19:30:00 | Latte            |        1 | 30000.00 |
|        4 | 2025-04-06 10:00:00 | Chocolate Muffin |        1 | 22000.00 |
|        4 | 2025-04-06 10:00:00 | Americano        |        1 | 27000.00 |
|        5 | 2025-04-06 15:15:00 | Cappuccino       |        1 | 32000.00 |
|        5 | 2025-04-06 15:15:00 | Black Tea        |        1 | 25000.00 |
|        6 | 2025-04-06 20:30:00 | Latte            |        1 | 30000.00 |
|        6 | 2025-04-06 20:30:00 | Cheese Danish    |        1 | 24000.00 |
+----------+---------------------+------------------+----------+----------+
12 rows in set (0.001 sec)

+---------------+
| total_revenue |
+---------------+
|     383000.00 |
+---------------+
1 row in set (0.074 sec)

Query OK, 0 rows affected (0.076 sec)

MariaDB [black_beans]>
```

4. Kreasi Sendiri

a. Membuat prosedur create new order (tanpa item dulu)

```
DELIMITER //
CREATE PROCEDURE create_new_order(
  IN p_employee_id INT,
  IN p_promo_id INT
)
BEGIN
  INSERT INTO orders (order_date, employee_id, promo_id, total_amount)
  VALUES (NOW(), p_employee_id, p_promo_id, 0.00);
END;
//
DELIMITER ;
```

```
MariaDB [black_beans]> call create_new_order(1, 1);
Query OK, 1 row affected (0.006 sec)

MariaDB [black_beans]> select * from orders;
+----------+---------------------+-------------+----------+--------------+
| order_id | order_date          | employee_id | promo_id | total_amount |
+----------+---------------------+-------------+----------+--------------+
|        1 | 2025-04-06 09:30:00 |           1 |        1 |    148500.00 |
|        2 | 2025-04-06 14:00:00 |           2 |     NULL |     48000.00 |
|        3 | 2025-04-06 19:30:00 |           3 |        2 |     40000.00 |
|        4 | 2025-04-06 10:00:00 |           4 |        3 |     48600.00 |
|        5 | 2025-04-06 15:15:00 |           5 |        4 |     43200.00 |
|        6 | 2025-04-06 20:30:00 |           6 |     NULL |     59000.00 |
|        7 | 2025-04-16 21:10:51 |           1 |        1 |     50000.00 |
|        9 | 2025-05-08 22:09:53 |           1 |        1 |         0.00 |
+----------+---------------------+-------------+----------+--------------+
8 rows in set (0.001 sec)

MariaDB [black_beans]>
```

b. Membuat prosedur menambahkan item ke orderan

```
DELIMITER //
CREATE PROCEDURE add_item_to_order(
  IN p_order_id INT,
  IN p_item_id INT,
  IN p_quantity INT
)
BEGIN
  INSERT INTO order_items (order_id, item_id, quantity)
  VALUES (p_order_id, p_item_id, p_quantity);
END;
//
DELIMITER ;

```

```
MariaDB [black_beans]> call add_item_to_order(9, 1, 3);
Query OK, 6 rows affected (0.016 sec)

MariaDB [black_beans]> select * from orders;
+----------+---------------------+-------------+----------+--------------+
| order_id | order_date          | employee_id | promo_id | total_amount |
+----------+---------------------+-------------+----------+--------------+
|        1 | 2025-04-06 09:30:00 |           1 |        1 |    148500.00 |
|        2 | 2025-04-06 14:00:00 |           2 |     NULL |     48000.00 |
|        3 | 2025-04-06 19:30:00 |           3 |        2 |     40000.00 |
|        4 | 2025-04-06 10:00:00 |           4 |        3 |     48600.00 |
|        5 | 2025-04-06 15:15:00 |           5 |        4 |     43200.00 |
|        6 | 2025-04-06 20:30:00 |           6 |     NULL |     59000.00 |
|        7 | 2025-04-16 21:10:51 |           1 |        1 |     50000.00 |
|        9 | 2025-05-08 22:09:53 |           1 |        1 |     67500.00 |
+----------+---------------------+-------------+----------+--------------+
8 rows in set (0.004 sec)

MariaDB [black_beans]>
```

c. **Membuat fungsi untuk mengecek menu ada atau tidak**

```
DELIMITER $$

CREATE FUNCTION get_item_status(p_item_id INT)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
  DECLARE count_item INT;
  DECLARE status_item VARCHAR(20);

  SELECT COUNT(*) INTO count_item
  FROM menu_items
  WHERE item_id = p_item_id;

  IF count_item > 0 THEN
    SET status_item = 'available';
  ELSE
    SET status_item = 'unavailable';
  END IF;

  RETURN status_item;
END $$

DELIMITER ;
```

```
MariaDB [black_beans]> select get_item_status(13);
+---------------------+
| get_item_status(13) |
+---------------------+
| available           |
+---------------------+
1 row in set (0.004 sec)

MariaDB [black_beans]> select get_item_status(14);
+---------------------+
| get_item_status(14) |
+---------------------+
| unavailable         |
+---------------------+
1 row in set (0.001 sec)

MariaDB [black_beans]>
```

d. Membuat fungsi untuk menghitung harga akhir dari sebuah orderan

```
DELIMITER //

CREATE FUNCTION calculate_order_total(p_order_id INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
  DECLARE total DECIMAL(10,2);
  DECLARE discount DECIMAL(5,2) DEFAULT 0;
  DECLARE promo INT;

  -- Hitung total harga tanpa diskon
  SELECT SUM(mi.price * oi.quantity)
  INTO total
  FROM order_items oi
  JOIN menu_items mi ON oi.item_id = mi.item_id
  WHERE oi.order_id = p_order_id;

  -- Ambil promo_id dari order
  SELECT promo_id
  INTO promo
  FROM orders
  WHERE order_id = p_order_id;

  -- Ambil diskon jika ada promo
  IF promo IS NOT NULL THEN
    SELECT discount_percent
    INTO discount
    FROM promotions
    WHERE promo_id = promo;
  END IF;

  -- Kembalikan total setelah diskon
  RETURN IFNULL(total, 0.00) * (1 - (discount / 100));
END;
//

DELIMITER ;
```

```
MariaDB [black_beans]> SELECT calculate_order_total(1) AS total_setelah_disk
on;
+----------------------+
| total_setelah_diskon |
+----------------------+
|            121500.00 |
+----------------------+
1 row in set (0.001 sec)

MariaDB [black_beans]> SELECT calculate_order_total(9) AS total_setelah_disk
on;
+----------------------+
| total_setelah_diskon |
+----------------------+
|             67500.00 |
+----------------------+
1 row in set (0.002 sec)

MariaDB [black_beans]>
```

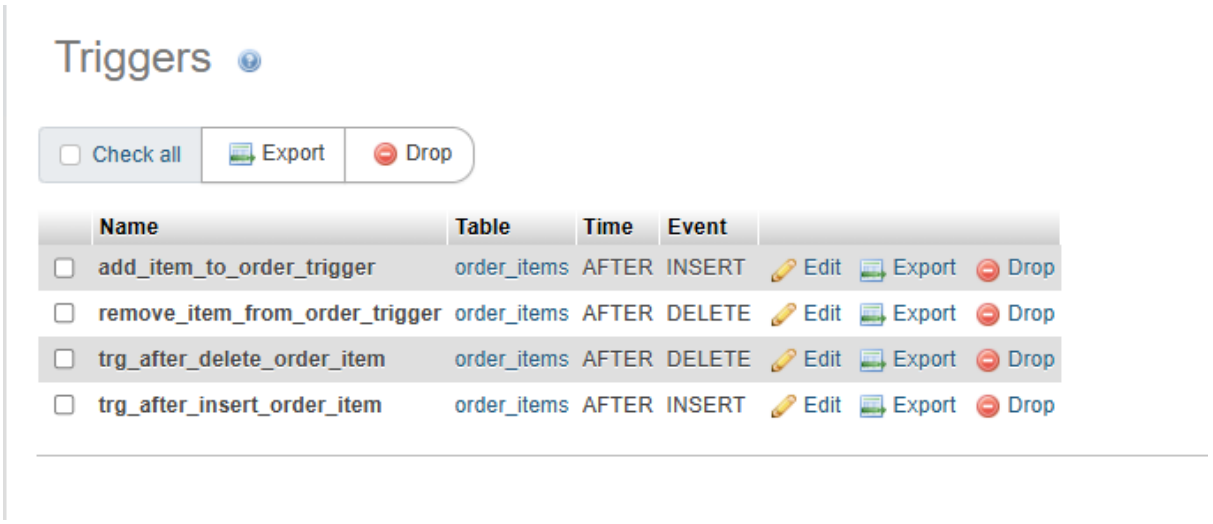
e. Membuat trigger untuk mencatat waktu ketika add item

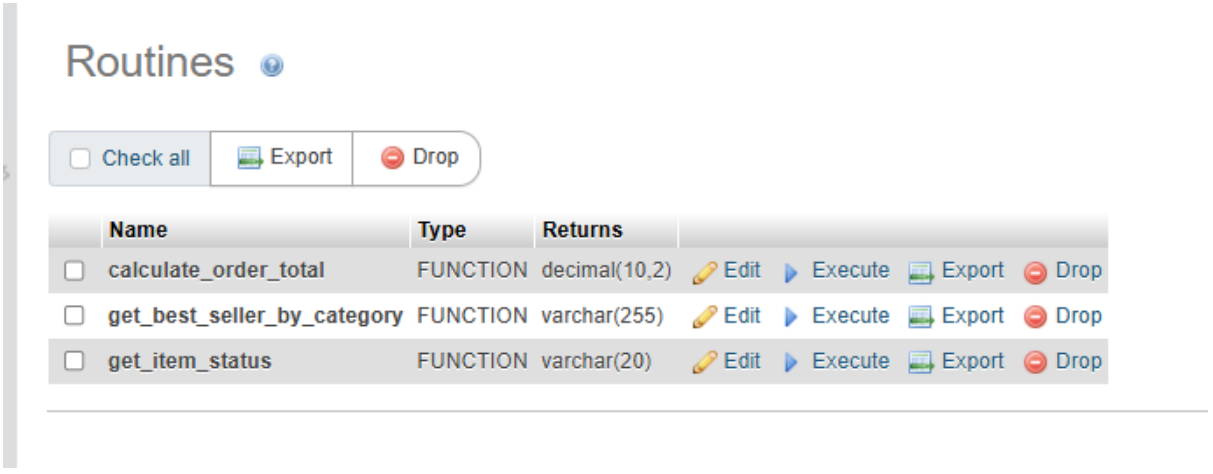
```
1  DELIMITER $$
2
3  CREATE TRIGGER add_item_to_order_trigger
4  AFTER INSERT ON order_items
5  FOR EACH ROW
6  BEGIN
7
8    INSERT INTO order_items_log (order_id, item_id, action, timestamp)
9    VALUES (NEW.order_id, NEW.item_id, 'Added', NOW());
10 END $$
11
12 DELIMITER ;
13
```

f. Membuat trigger untuk mencatat waktu ketika hapus item

```
DELIMITER $$

CREATE TRIGGER remove_item_from_order_trigger
AFTER DELETE ON order_items
FOR EACH ROW
BEGIN
  
  INSERT INTO order_items_log (order_id, item_id, action, timestamp)
  VALUES (OLD.order_id, OLD.item_id, 'Removed', NOW());
END $$

DELIMITER ;

```

BUKTI





Routines

Check all | Export | Drop

	Name	Type	Returns				
☐	SP_FilterMenuByMaxHarga	PROCEDURE		Edit	Execute	Export	Drop
☐	add_item_to_order	PROCEDURE		Edit	Execute	Export	Drop
☐	create_new_order	PROCEDURE		Edit	Execute	Export	Drop
☐	getMenu	PROCEDURE		Edit	Execute	Export	Drop
☐	get_orders_and_revenue_by_date	PROCEDURE		Edit	Execute	Export	Drop
☐	sp_lokal_variabel	PROCEDURE		Edit	Execute	Export	Drop