TRIGGERS

1. LOGIN TABLE

```
CREATE DEFINER = CURRENT_USER TRIGGER
`FlipZon`.`LOGIN_BEFORE_INSERT` BEFORE INSERT ON `LOGIN` FOR
EACH ROW
BEGIN
IF NEW.username IS NULL THEN
SET NEW.username = "USERNAME MANDATORY";
END IF;
END
```

2. CART_ITEMS TABLE

```
CREATE DEFINER = CURRENT_USER TRIGGER
`FlipZon`.`CART_ITEMS_BEFORE_INSERT` BEFORE INSERT ON
`CART_ITEMS` FOR EACH ROW
BEGIN
IF NEW.ProductName IS NULL THEN
SET NEW.ProductName = "PRODUCT NAME MANDATORY";
END IF;
END
```

EMBEDDED QUERIES

1. CREATE A TABLE IF IT DOES NOT ALREADY EXIST

```
cursor.execute("USE FlipZon")
cursor.execute("CREATE TABLE IF NOT EXISTS DEMO_TABLE(name
varchar(200), username varchar(200))")
cursor.execute("SHOW TABLES")
for x in cursor:
    print(x)
```

2. DELETE THE TABLE IF IT ALREADY EXISTS

```
cursor.execute("USE FlipZon")
cursor.execute("DROP TABLE IF EXISTS DEMO_TABLE")
```

OLAP QUERIES

1. Show the total revenue generated by each product category for a specific time period.

    ```
    SELECT c.CategoryName, SUM(p.ProductPrice * oi.Quantity) AS
    Revenue
    FROM CATEGORY c
    JOIN PRODUCT p ON c.ProductId = p.ProductId
    JOIN CART_ITEMS oi ON p.ProductId = oi.ProductId
    JOIN CART o ON oi.CartId = o.CartId
    JOIN ORDERS od ON o.CustId = od.CustId
    WHERE od.OrderDate BETWEEN '2022-01-01' AND '2022-12-31'
    GROUP BY c.CategoryName
    ORDER BY Revenue DESC;
    ```

2. Show the top 5 customers who have spent the most money on orders in the past year.

    ```
    SELECT c.CustName, SUM(o.OrderAmount) AS TotalSpent
    FROM CUSTOMER c
    JOIN ORDERS o ON c.CustId = o.CustId
    WHERE o.OrderDate BETWEEN '2022-01-01' AND '2022-12-31'
    GROUP BY c.CustName
    ORDER BY TotalSpent DESC
    LIMIT 5;
    ```

3. Show the number of orders and total revenue generated for each month of the year.

    ```
    SELECT MONTH(OrderDate) AS Month, COUNT(*) AS NumOrders,
    SUM(OrderAmount) AS Revenue
    FROM ORDERS
    WHERE OrderDate BETWEEN '2022-01-01' AND '2022-12-31'
    GROUP BY MONTH(OrderDate)
    ORDER BY MONTH(OrderDate);
    ```

4. Show the total revenue generated by each courier company for a specific time period.

    ```
    SELECT td.CourierName, SUM(o.OrderAmount) AS Revenue
    FROM TRACKING_DETAILS td
    JOIN ORDERS o ON td.TrackingNum = o.TrackingNum
    WHERE o.OrderDate BETWEEN '2022-01-01' AND '2022-12-31'
    GROUP BY td.CourierName
    ORDER BY Revenue DESC;
    ```