

## Triggers

1. Trigger to detect suspicious Bot Activity, If a user has placed more than 3 orders within the last 5 minutes then it detects it as a bot activity and add this data to the log table.

```
DELIMITER //
```

```
CREATE TRIGGER check_order_frequency_and_verify
AFTER INSERT ON Orders
FOR EACH ROW
BEGIN
    DECLARE order_count INT;

    -- Retrieve the count of orders placed by the user within the last 5
minutes
    SELECT COUNT(*)
    INTO order_count
    FROM Orders
    WHERE userID = NEW.userID
    AND orderID != NEW.orderID -- Exclude the current order
    AND order_date >= DATE_SUB(NOW(), INTERVAL 5 MINUTE);

    -- If the user has placed more than 2 orders within the last 5 minutes
    IF order_count > 2 THEN
        -- Log the action
        INSERT INTO log_table (user_id, action, timestamp)
        VALUES (NEW.userID, 'Multiple orders placed within a short interval.
Possible robot activity detected.', NOW());

        -- You can take further actions here such as alerting an administrator
or locking the user's account.
    END IF;
END//
```

```
DELIMITER ;
```

2. Trigger to insert records during a Transaction into the tables : Transactions,CustomersTransactions, transaction\_for\_order.

```
DROP TRIGGER IF EXISTS make_transactions;
```

```
DELIMITER $$
```

```
CREATE TRIGGER make_transactions AFTER INSERT ON orders  
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE t_id INT;
```

```
    SELECT MAX(transactionID) + 1 INTO t_id FROM Transactions;
```

```
    IF t_id IS NULL THEN
```

```
        SET t_id = 1; -- If the table is empty, set t_id to 1
```

```
    END IF;
```

```
    INSERT INTO Transactions VALUES (t_id, CURRENT_DATE(), 0,  
'Pending','NA');
```

```
    INSERT INTO CustomersTransactions VALUES (NEW.userID,t_id);
```

```
    INSERT INTO transaction_for_order VALUES (NEW.orderID,t_id);
```

```
END$$
```

```
DELIMITER ;
```

### 3. Trigger for updating the cart amount and product quantity

DELIMITER //

```
CREATE TRIGGER update_product_and_cart_after_cart_insert
AFTER INSERT ON Cart_has_Products
FOR EACH ROW
BEGIN
```

```
    DECLARE product_price DECIMAL(10, 2);
```

```
    -- Retrieve the product price from the Product table
```

```
    SELECT price INTO product_price
```

```
    FROM Product
```

```
    WHERE productID = NEW.productID;
```

```
    -- Update the product quantity in the Product table
```

```
    UPDATE Product
```

```
    SET quantity = quantity - NEW.product_quantity
```

```
    WHERE productID = NEW.productID;
```

```
    -- Update the cart amount in the Cart table
```

```
    UPDATE Cart
```

```
    SET total_amount = total_amount + (NEW.product_quantity *
product_price)
```

```
    WHERE cartID = NEW.cartID;
```

```
END//
```

```
DELIMITER ;
```

### Contributions:-

1. Nikhil Kumar (2022322): Designed the Basic Interface
2. Vipul (2022576): Wrote Embedded SQL queries in python
3. Nitin Kumar (2022337): Made Triggers