Prevent inserting a new order if the user already has 3 or more orders in the same month.

**Step-by-Step Implementation**

**1. Create the Trigger**

sql

DELIMITER $$

CREATE TRIGGER trg\_limit\_orders\_per\_month

BEFORE INSERT ON orders

FOR EACH ROW

BEGIN

DECLARE order\_count INT;

SELECT COUNT(\*) INTO order\_count

FROM orders

WHERE user\_id = NEW.user\_id

AND MONTH(order\_date) = MONTH(NEW.order\_date)

AND YEAR(order\_date) = YEAR(NEW.order\_date);

IF order\_count >= 3 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'User has already placed 3 orders this month.';

END IF;

END$$

DELIMITER ;

**2. Test the Trigger**

**Insert order #1 (allowed)**

sql

INSERT INTO orders (order\_id, user\_id, order\_date)

VALUES (201, 1, '2024-07-01');

**Insert order #2 (allowed)**

sql

INSERT INTO orders (order\_id, user\_id, order\_date)

VALUES (202, 1, '2024-07-10');

**Insert order #3 (allowed)**

sql

INSERT INTO orders (order\_id, user\_id, order\_date)

VALUES (203, 1, '2024-07-15');

**Insert order #4 (blocked)**

sql

INSERT INTO orders (order\_id, user\_id, order\_date)

VALUES (204, 1, '2024-07-20');

**Error Message:** User has already placed 3 orders this month.

**Notes**

* The trigger runs **before every insert** and uses NEW.order\_date to check the month and year.
* It relies on the existing data in orders, so if orders are inserted in bulk or through a script, it will still block the 4th.

You can also create a similar trigger for:

* Preventing duplicate order items in the same order?
* Enforcing max order amount per user per month?
* Logging blocked attempts to a separate table?