**Tasks 4: Subquery and its types**

**1. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.**

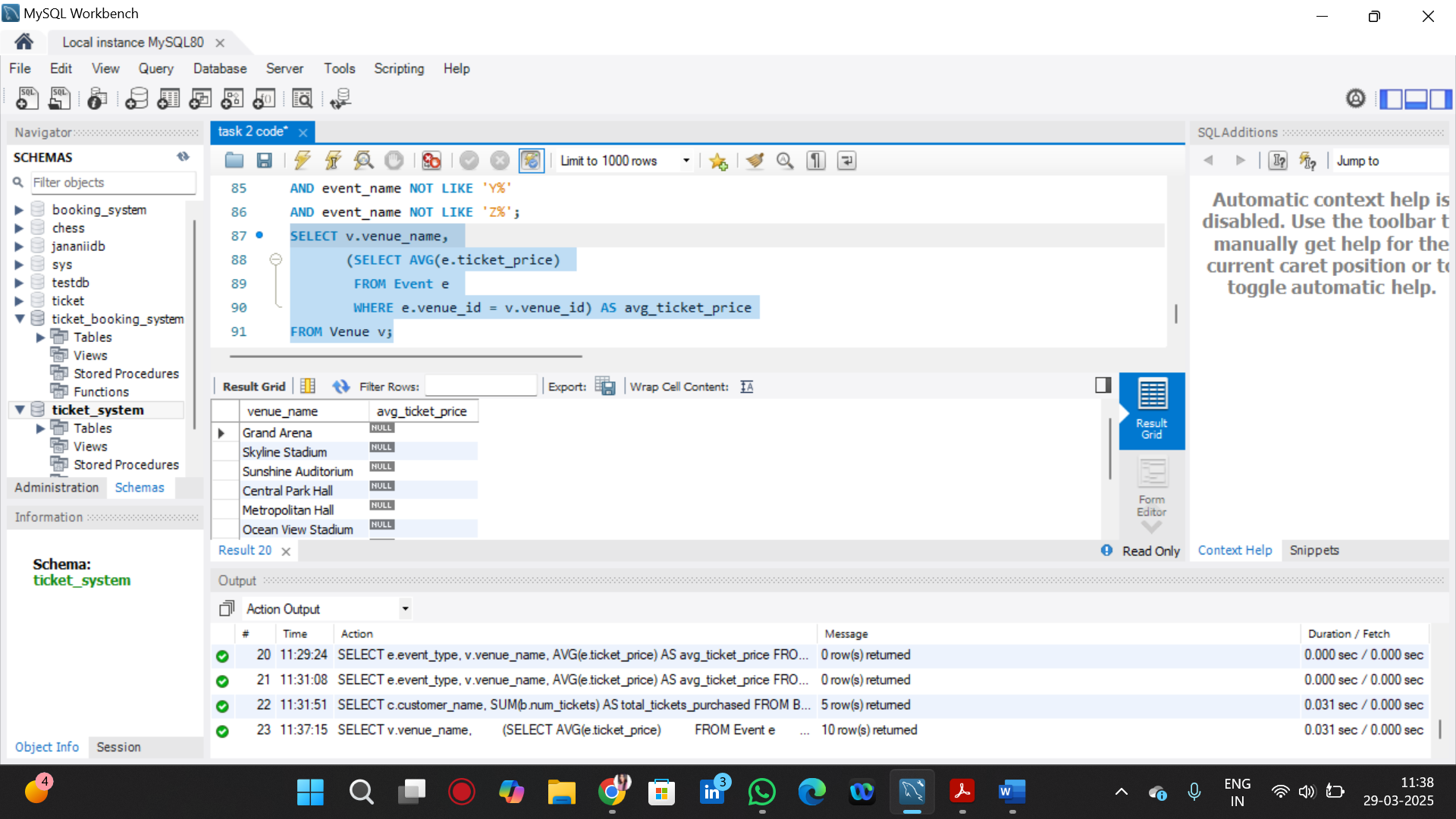
SELECT v.venue\_name,

(SELECT AVG(e.ticket\_price)

FROM Event e

WHERE e.venue\_id = v.venue\_id) AS avg\_ticket\_price

FROM Venue v;

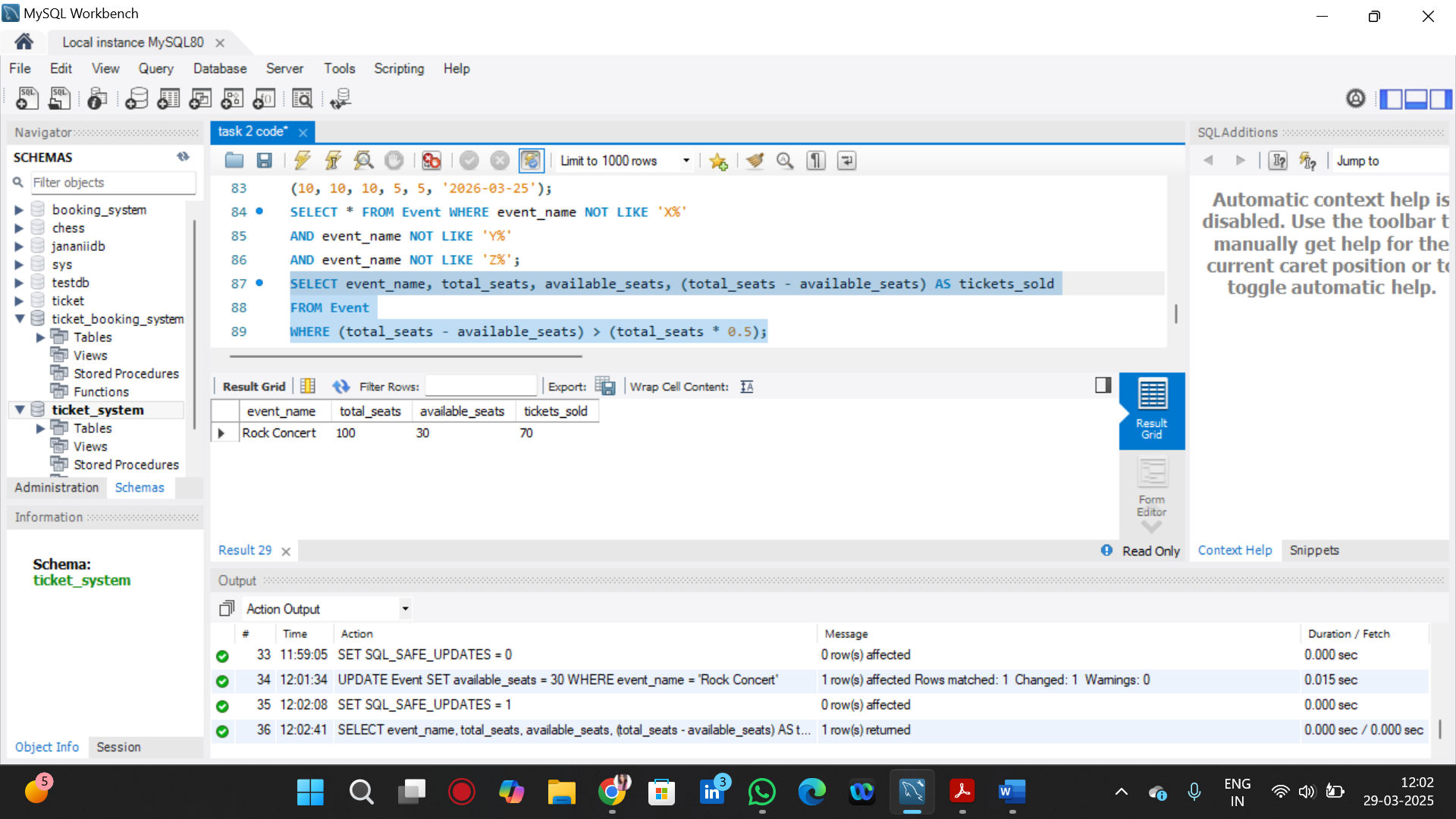


2. Find Events with More Than 50% of Tickets Sold using subquery.

SELECT event\_name, total\_seats, available\_seats, (total\_seats - available\_seats) AS tickets\_sold

FROM Event

WHERE (total\_seats - available\_seats) > (total\_seats \* 0.5);



3. Calculate the Total Number of Tickets Sold for Each Event.

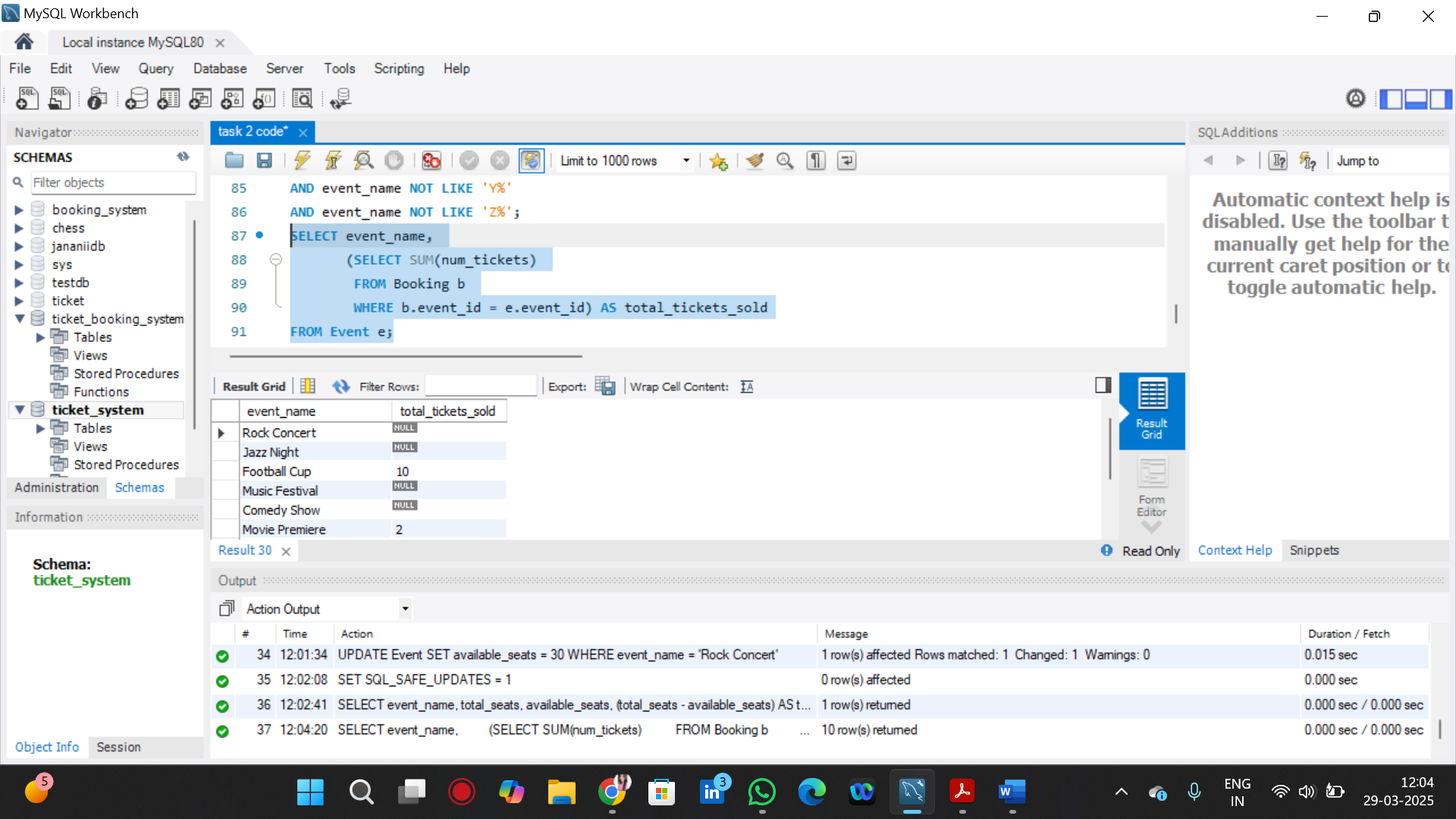
SELECT event\_name,

(SELECT SUM(num\_tickets)

FROM Booking b

WHERE b.event\_id = e.event\_id) AS total\_tickets\_sold

FROM Event e;

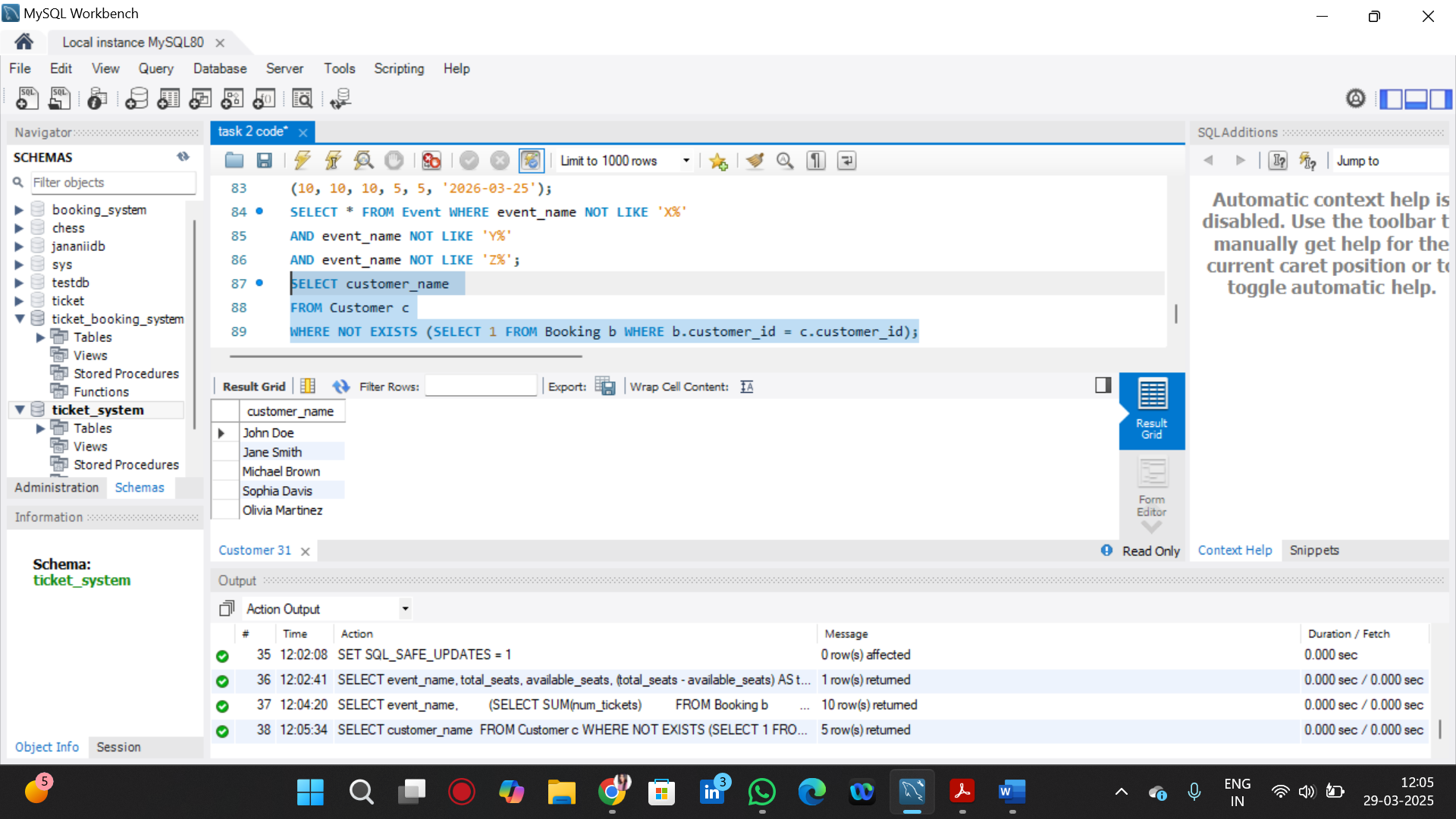


4. Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.

SELECT customer\_name

FROM Customer c

WHERE NOT EXISTS (SELECT 1 FROM Booking b WHERE b.customer\_id = c.customer\_id);

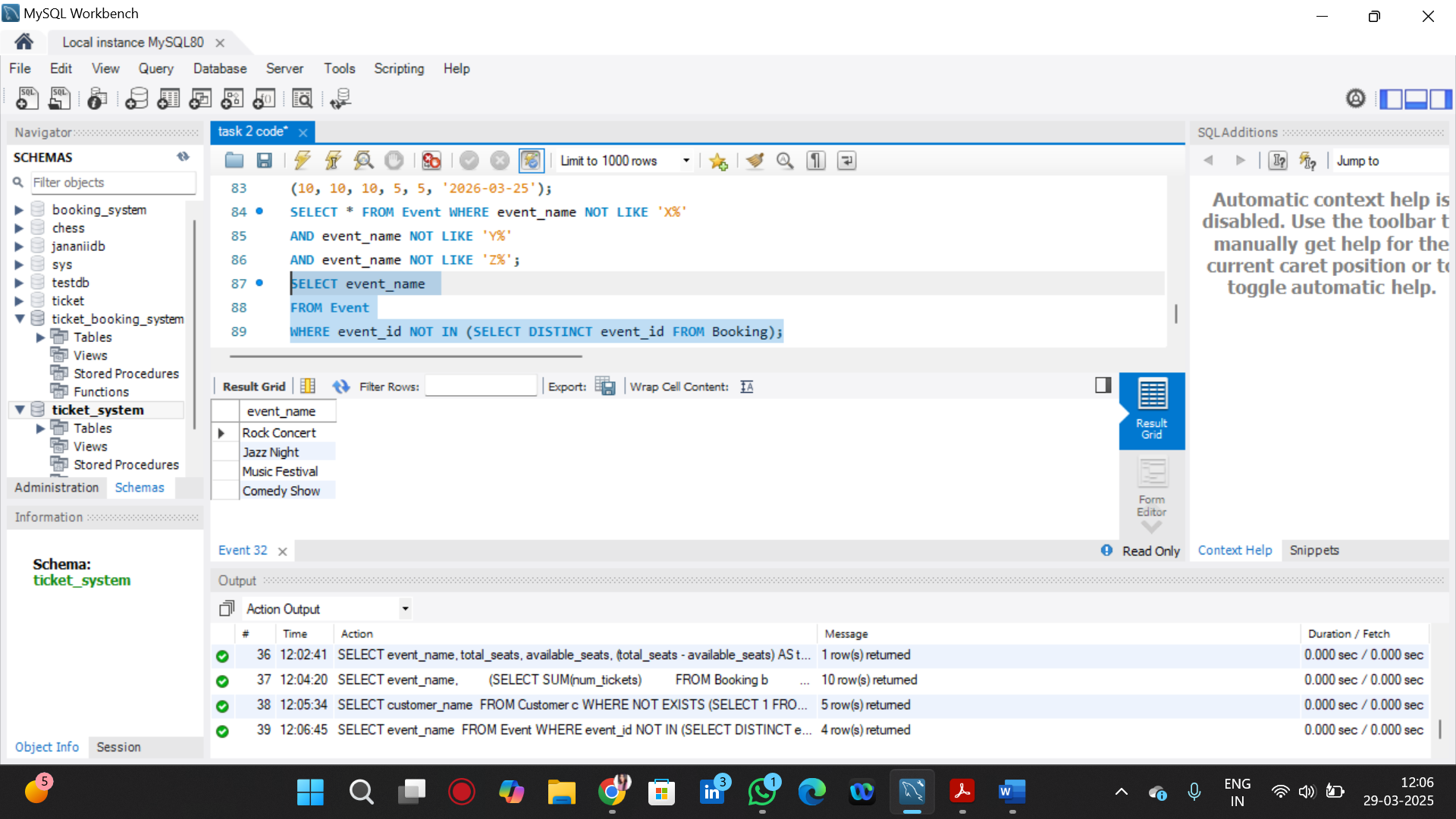


5. List Events with No Ticket Sales Using a NOT IN Subquery

SELECT event\_name

FROM Event

WHERE event\_id NOT IN (SELECT DISTINCT event\_id FROM Booking);



6. Calculate the Total Number of Tickets Sold for Each Event Type Using a Subquery in the FROM Clause.

SELECT event\_type, SUM(tickets\_sold) AS total\_tickets\_sold

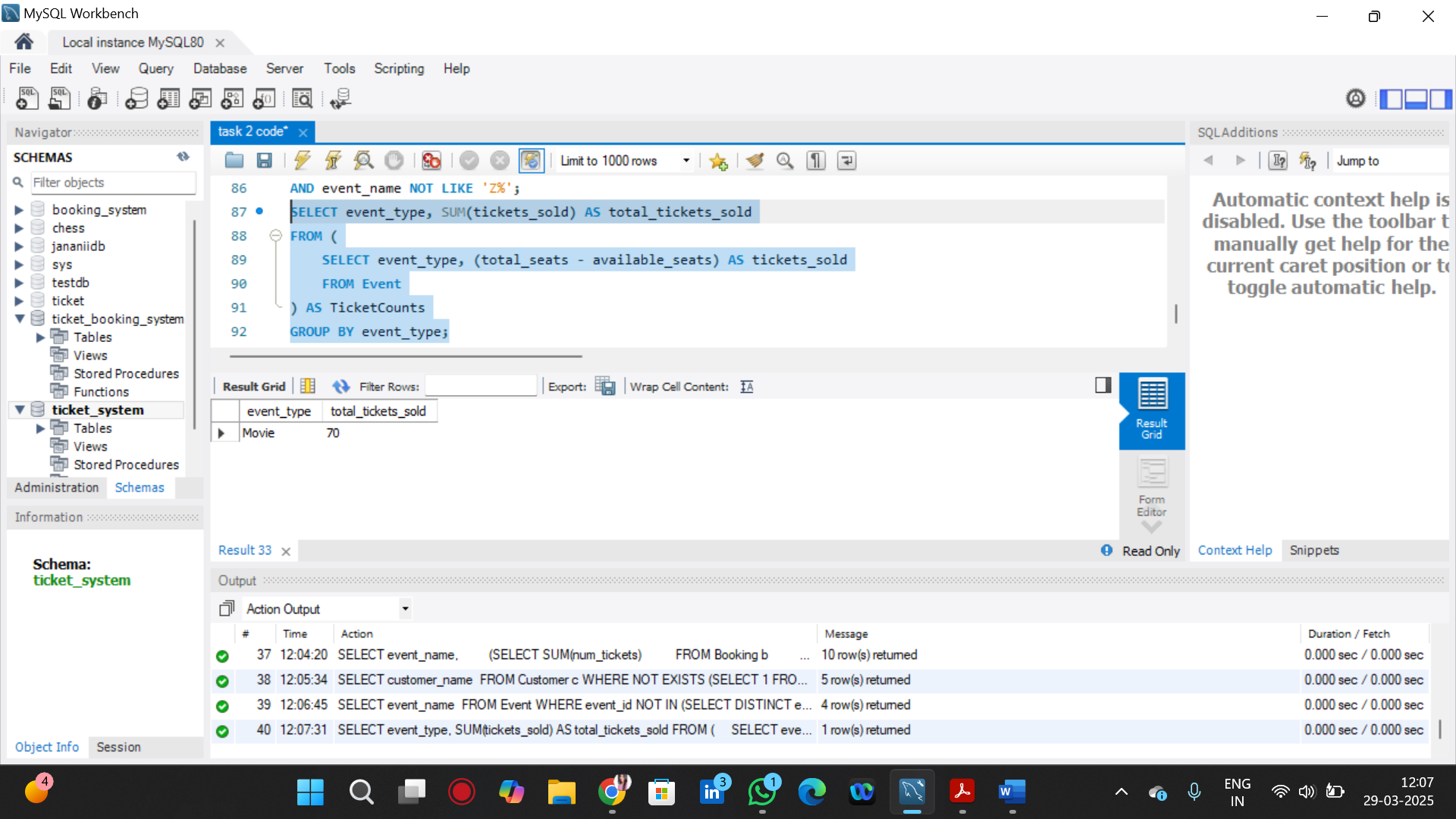
FROM (

SELECT event\_type, (total\_seats - available\_seats) AS tickets\_sold

FROM Event

) AS TicketCounts

GROUP BY event\_type;

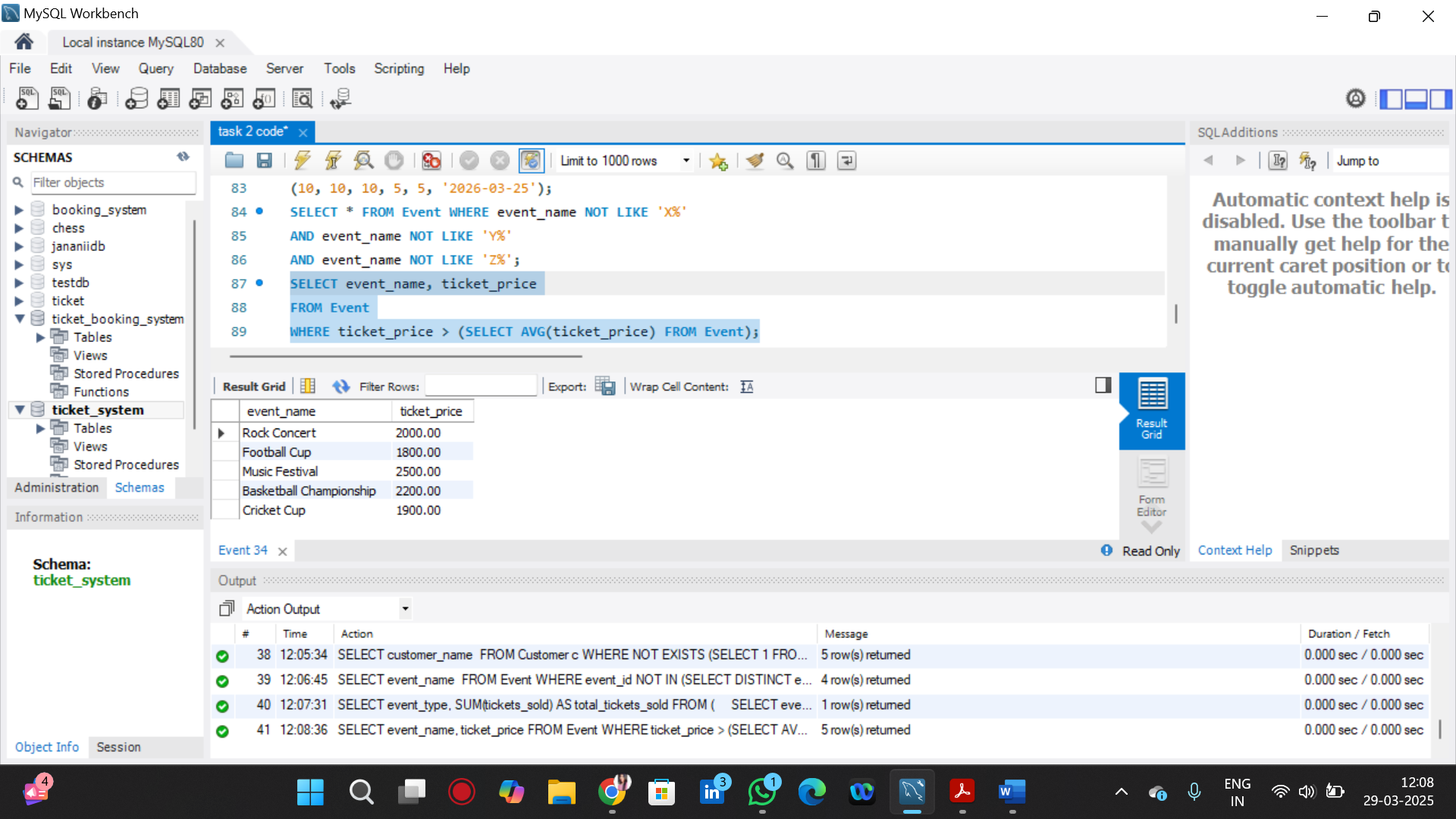


7. Find Events with Ticket Prices Higher Than the Average Ticket Price Using a Subquery in the WHERE Clause.

SELECT event\_name, ticket\_price

FROM Event

WHERE ticket\_price > (SELECT AVG(ticket\_price) FROM Event);



8. Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery.

SELECT customer\_name,

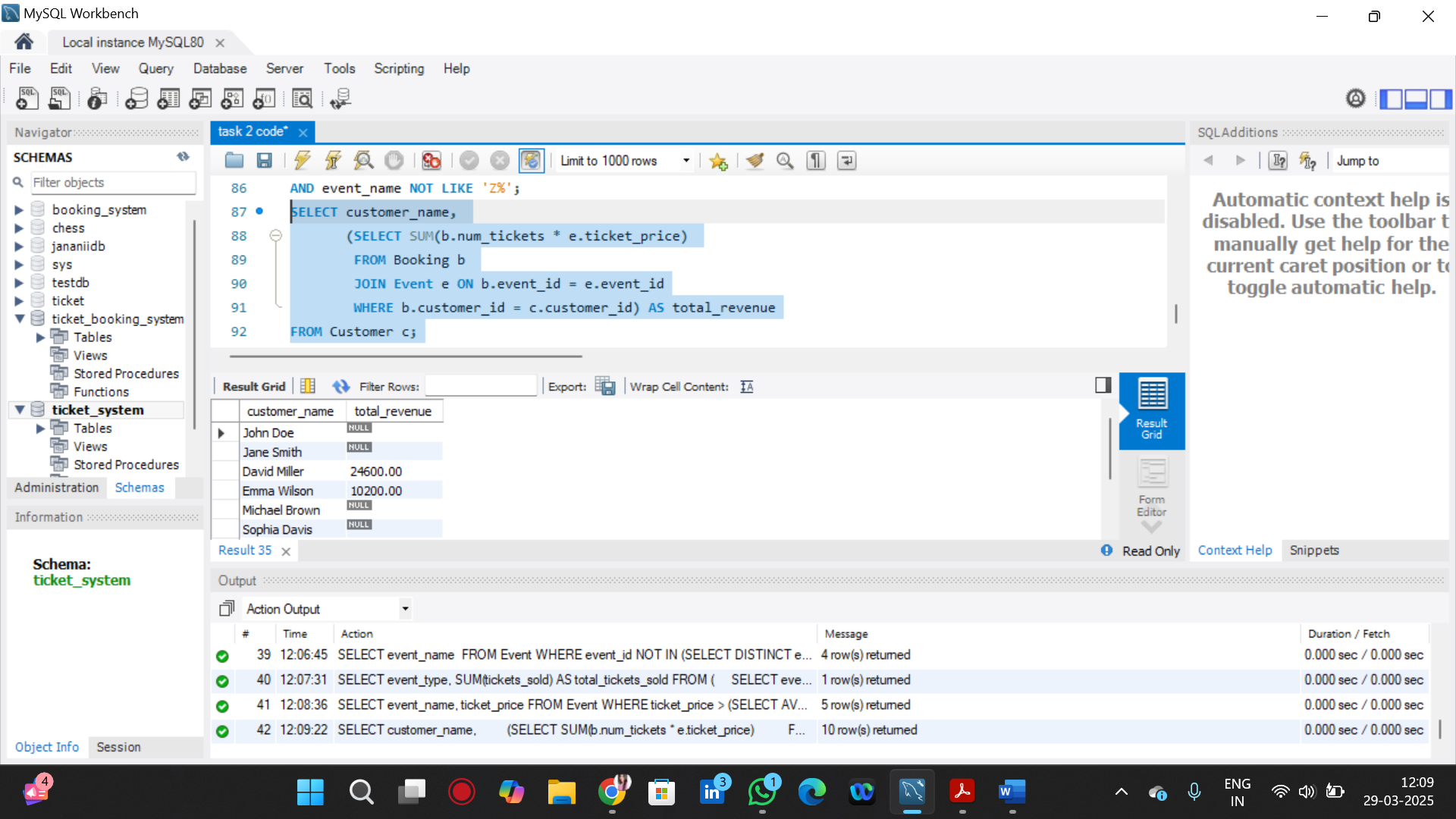
(SELECT SUM(b.num\_tickets \* e.ticket\_price)

FROM Booking b

JOIN Event e ON b.event\_id = e.event\_id

WHERE b.customer\_id = c.customer\_id) AS total\_revenue

FROM Customer c;



9. List Users Who Have Booked Tickets for Events in a Given Venue Using a Subquery in the WHERE Clause

SELECT \* FROM Booking WHERE event\_id IN (SELECT event\_id FROM Event WHERE venue\_id = 1);



10. Calculate the Total Number of Tickets Sold for Each Event Category Using a Subquery with GROUP BY.

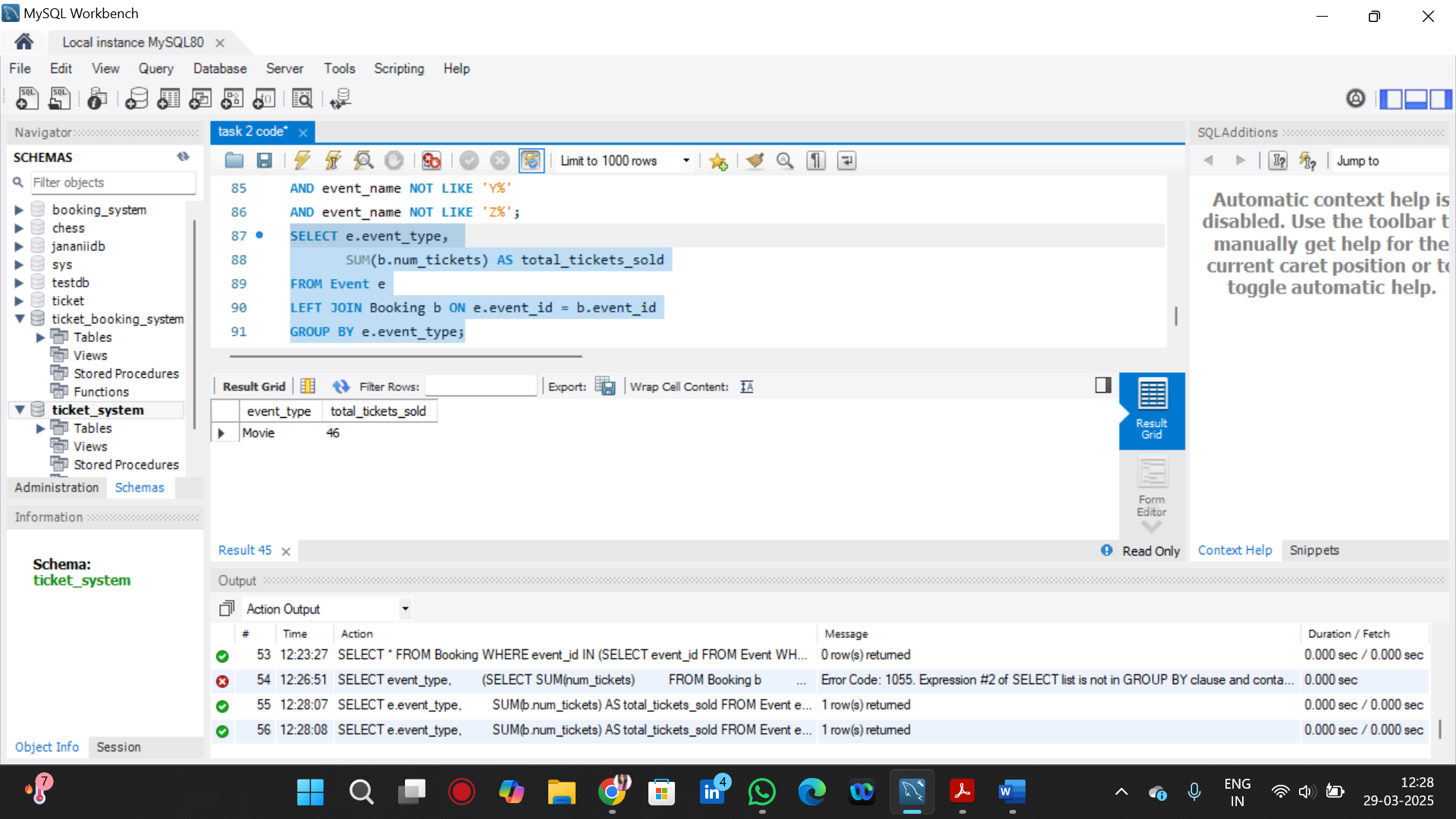
SELECT e.event\_type,

SUM(b.num\_tickets) AS total\_tickets\_sold

FROM Event e

LEFT JOIN Booking b ON e.event\_id = b.event\_id

GROUP BY e.event\_type;



11. Find Users Who Have Booked Tickets for Events in each Month Using a Subquery with DATE\_FORMAT.

SELECT DISTINCT customer\_name,

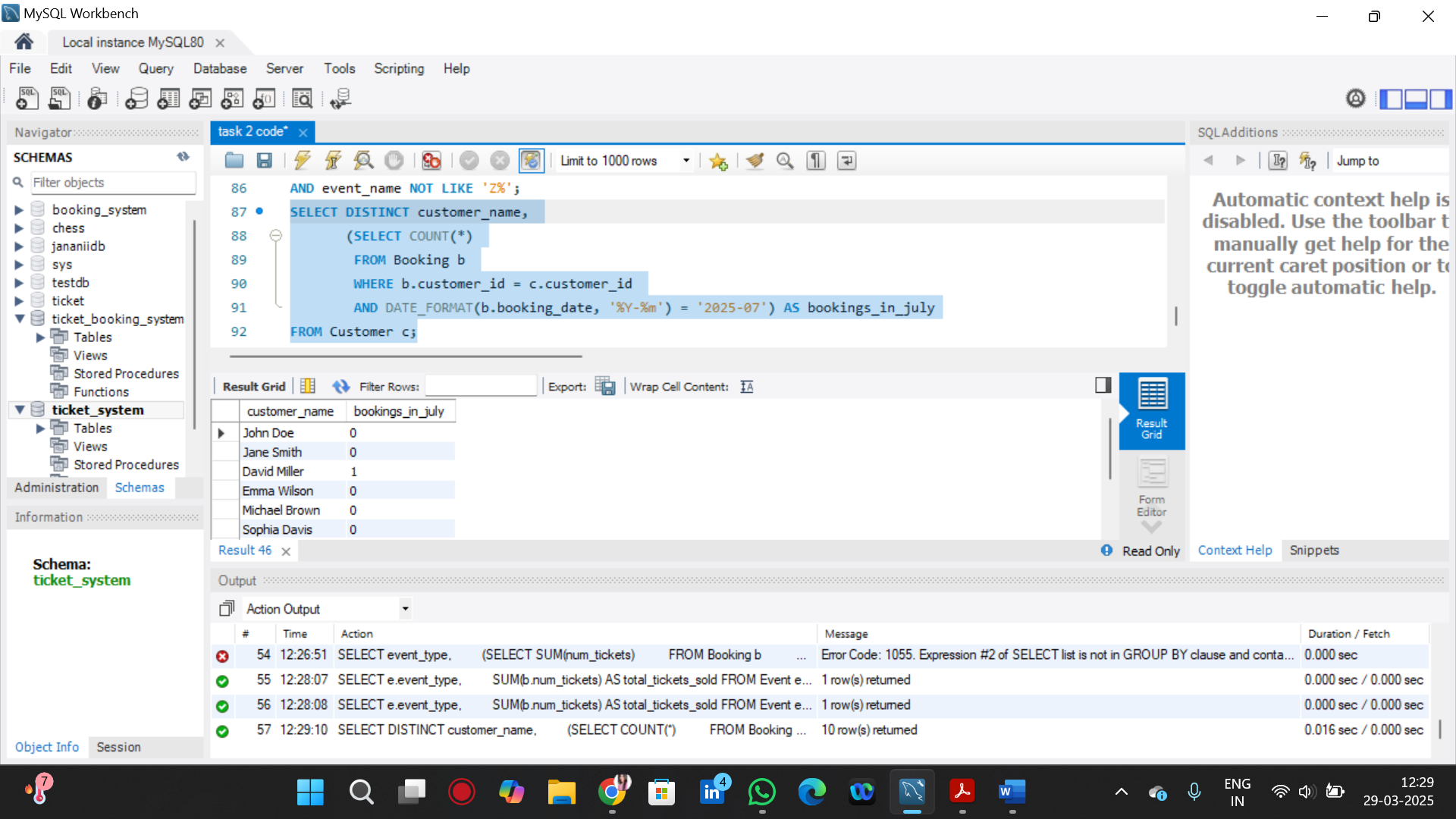
(SELECT COUNT(\*)

FROM Booking b

WHERE b.customer\_id = c.customer\_id

AND DATE\_FORMAT(b.booking\_date, '%Y-%m') = '2025-07') AS bookings\_in\_july

FROM Customer c;



12. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery

SELECT v.venue\_name,

COALESCE(AVG(e.ticket\_price), 0) AS avg\_ticket\_price

FROM Venue v

LEFT JOIN Event e ON v.venue\_id = e.venue\_id

GROUP BY v.venue\_name;

