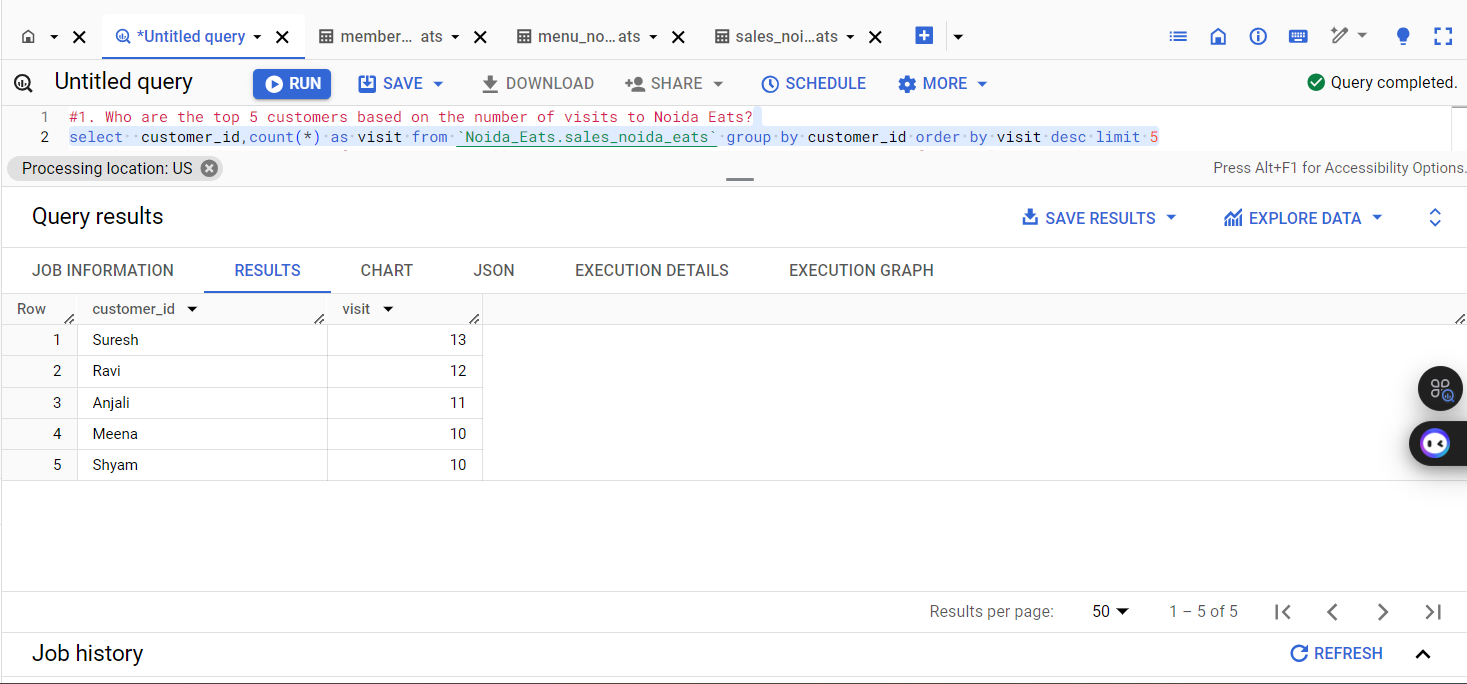
**Case Study: Analyzing Noida Eats'**

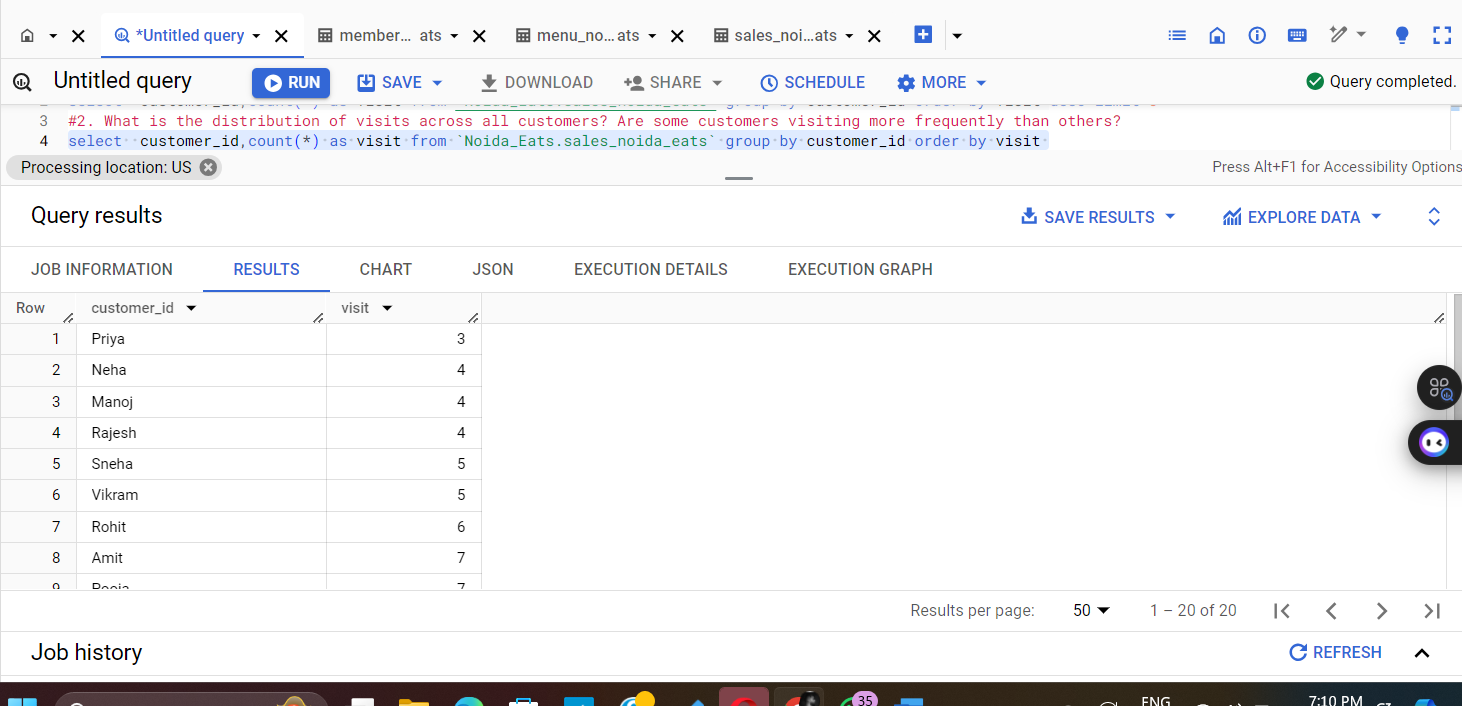
Scenario 1: Customer Visiting Patterns

#1. Who are the top 5 customers based on the number of visits to Noida Eats?

select  customer\_id,count(\*) as visit from `Noida\_Eats.sales\_noida\_eats` group by customer\_id order by visit desc limit 5



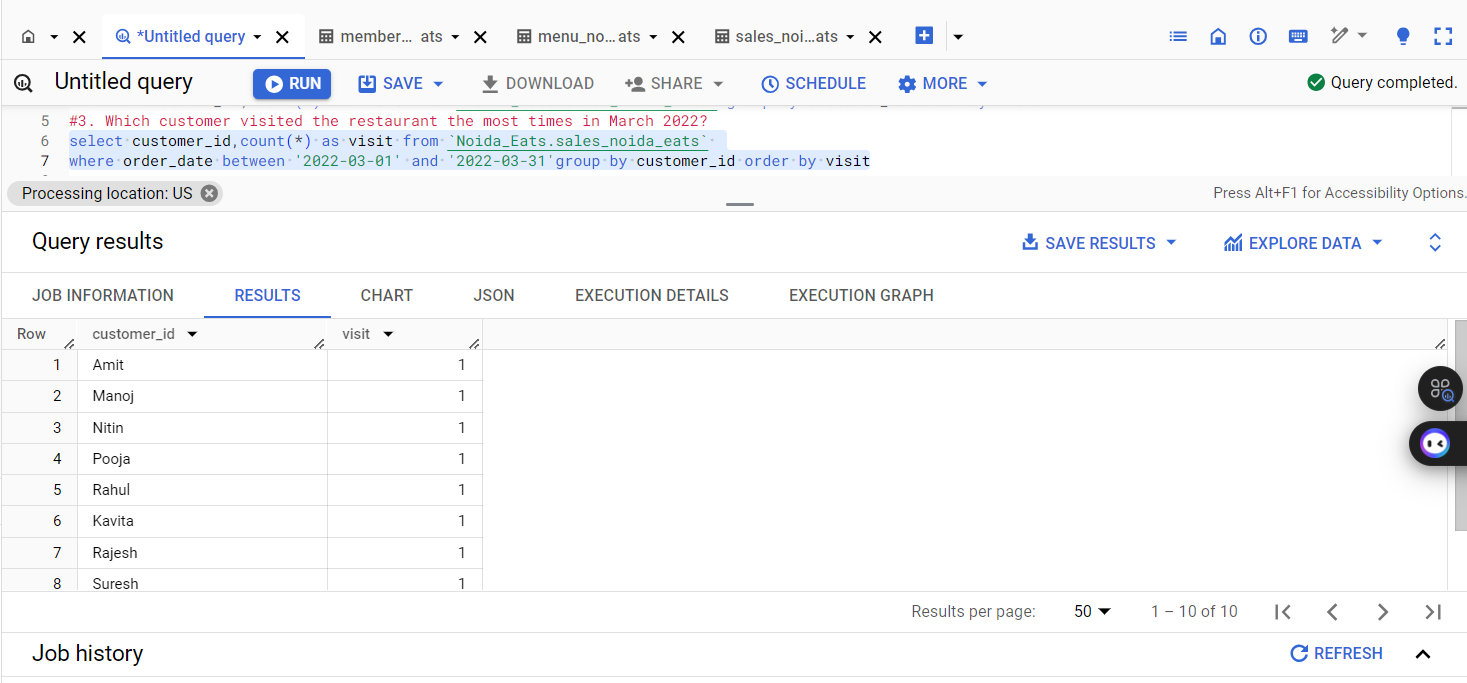
#2. What is the distribution of visits across all customers? Are some customers visiting more frequently than others?

select  customer\_id,count(\*) as visit from `Noida\_Eats.sales\_noida\_eats` group by customer\_id order by visit 

#3. Which customer visited the restaurant the most times in March 2022?

select customer\_id,count(\*) as visit from `Noida\_Eats.sales\_noida\_eats`

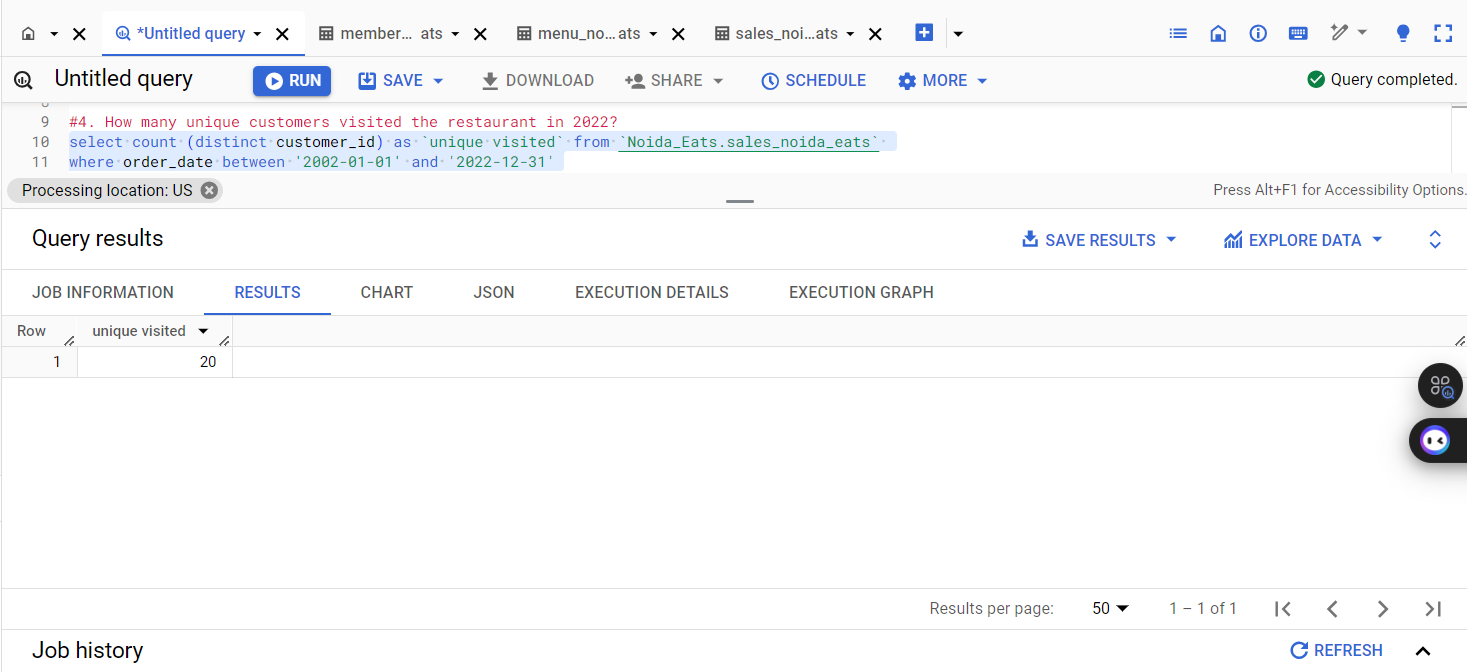
where order\_date between '2022-03-01' and '2022-03-31'group by customer\_id order by visit



#4. How many unique customers visited the restaurant in 2022?

select count (distinct customer\_id) as `unique visited` from `Noida\_Eats.sales\_noida\_eats`

where order\_date between '2002-01-01' and '2022-12-31'



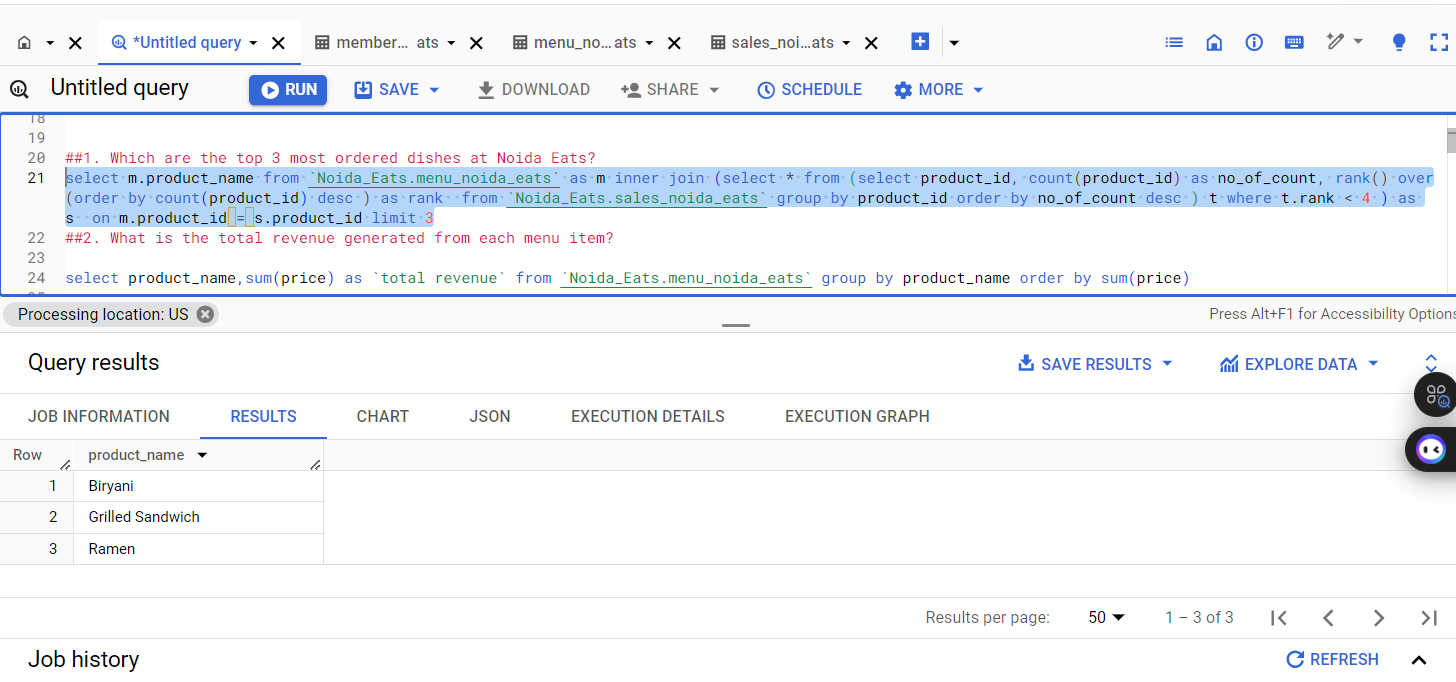
Result Insights:

The analysis reveals key insights into customer behavior at Noida Eats. Identifying the top customers allows for targeted marketing strategies, such as loyalty programs or personalized offers. The distribution of visits indicates whether certain customers are more engaged, which can inform customer retention efforts. The most frequent visitor in March 2022 highlights peak engagement periods, suggesting potential for promotional events. Lastly, understanding the unique customer count in 2022 provides a baseline for growth metrics, essential for strategic planning and resource allocation. Overall, these insights can guide Noida Eats in enhancing customer experience and optimizing marketing efforts.

Scenario 2: Menu Preferences

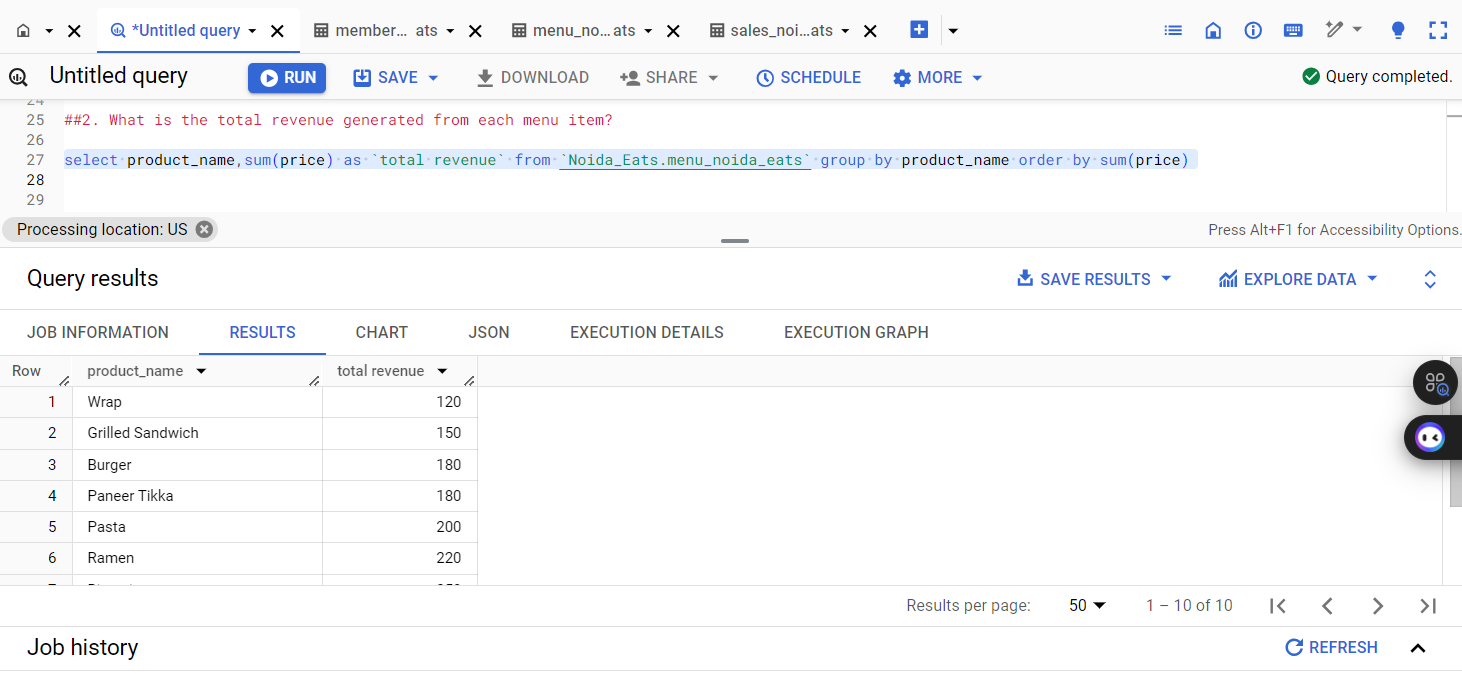
##1. Which are the top 3 most ordered dishes at Noida Eats?

select m.product\_name from `Noida\_Eats.menu\_noida\_eats` as m inner join (select \* from (select product\_id, count(product\_id) as no\_of\_count, rank() over(order by count(product\_id) desc ) as rank  from `Noida\_Eats.sales\_noida\_eats` group by product\_id order by no\_of\_count desc ) t where t.rank < 4 ) as s  on m.product\_id = s.product\_id limit 3



##2. What is the total revenue generated from each menu item?

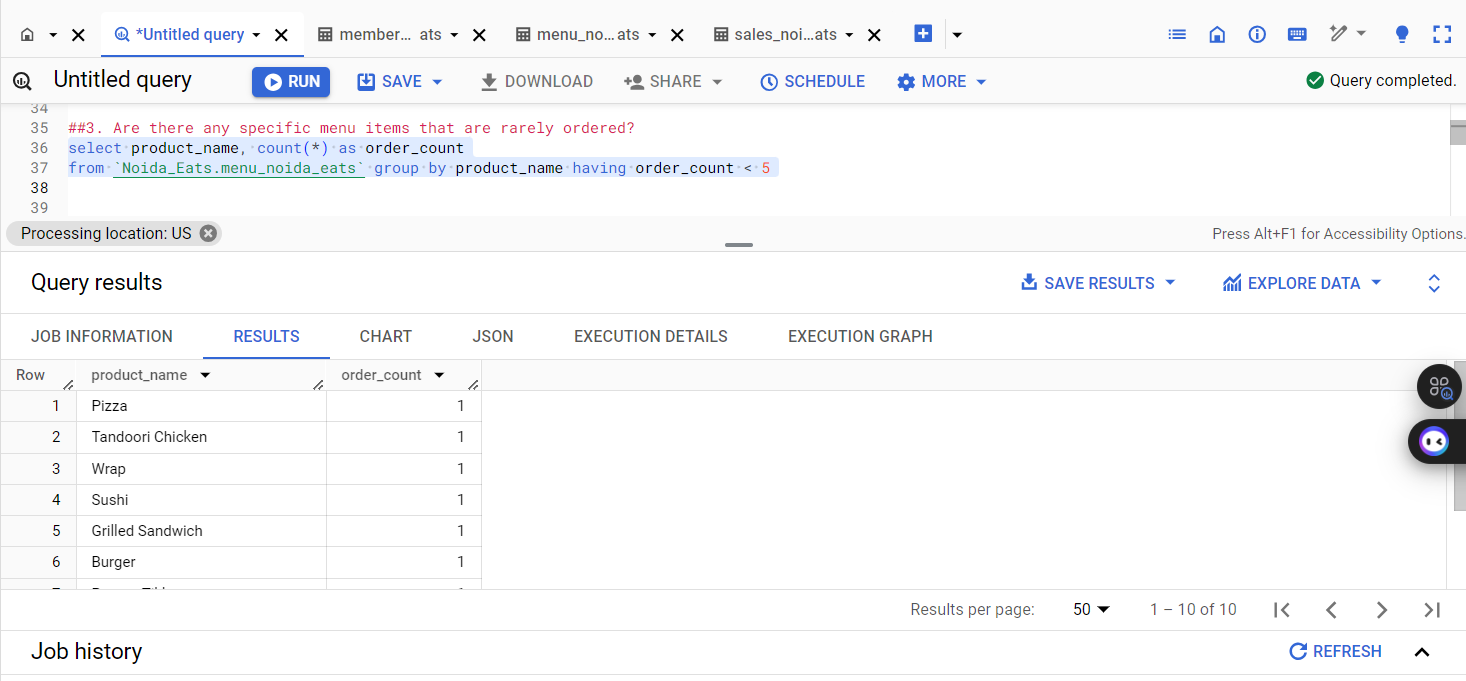
select product\_name,sum(price) as `total revenue` from `Noida\_Eats.menu\_noida\_eats` group by product\_name order by sum(price)



##3. Are there any specific menu items that are rarely ordered?

select product\_name, count(\*) as order\_count

from `Noida\_Eats.menu\_noida\_eats` group by product\_name having order\_count < 5

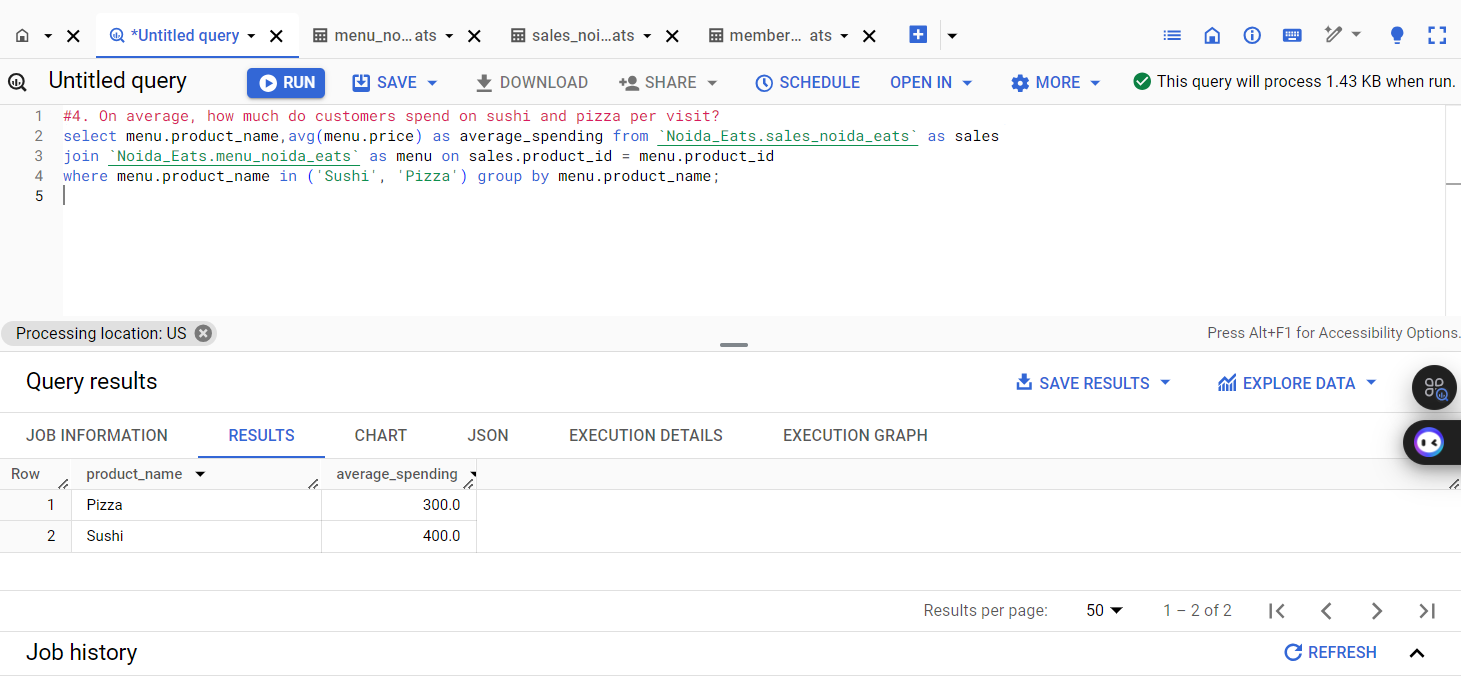


#4. On average, how much do customers spend on sushi and pizza per visit?

select menu.product\_name,avg(menu.price) as average\_spending from `Noida\_Eats.sales\_noida\_eats` as sales

join `Noida\_Eats.menu\_noida\_eats` as menu on sales.product\_id = menu.product\_id

where menu.product\_name in ('Sushi', 'Pizza') group by menu.product\_name;



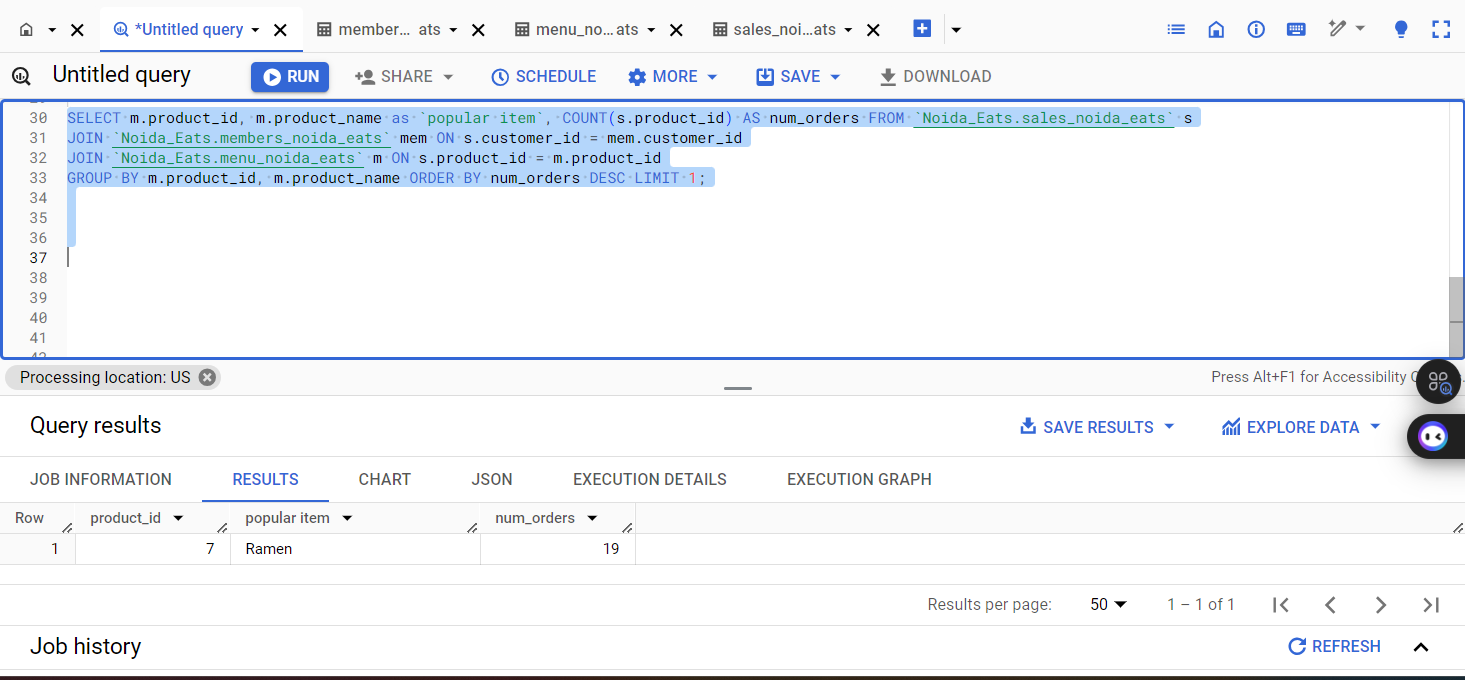
##5. Which menu item is most popular among loyalty program members?

SELECT m.product\_id, m.product\_name as `popular item`, COUNT(s.product\_id) AS num\_orders FROM `Noida\_Eats.sales\_noida\_eats` s

JOIN `Noida\_Eats.members\_noida\_eats` mem ON s.customer\_id = mem.customer\_id

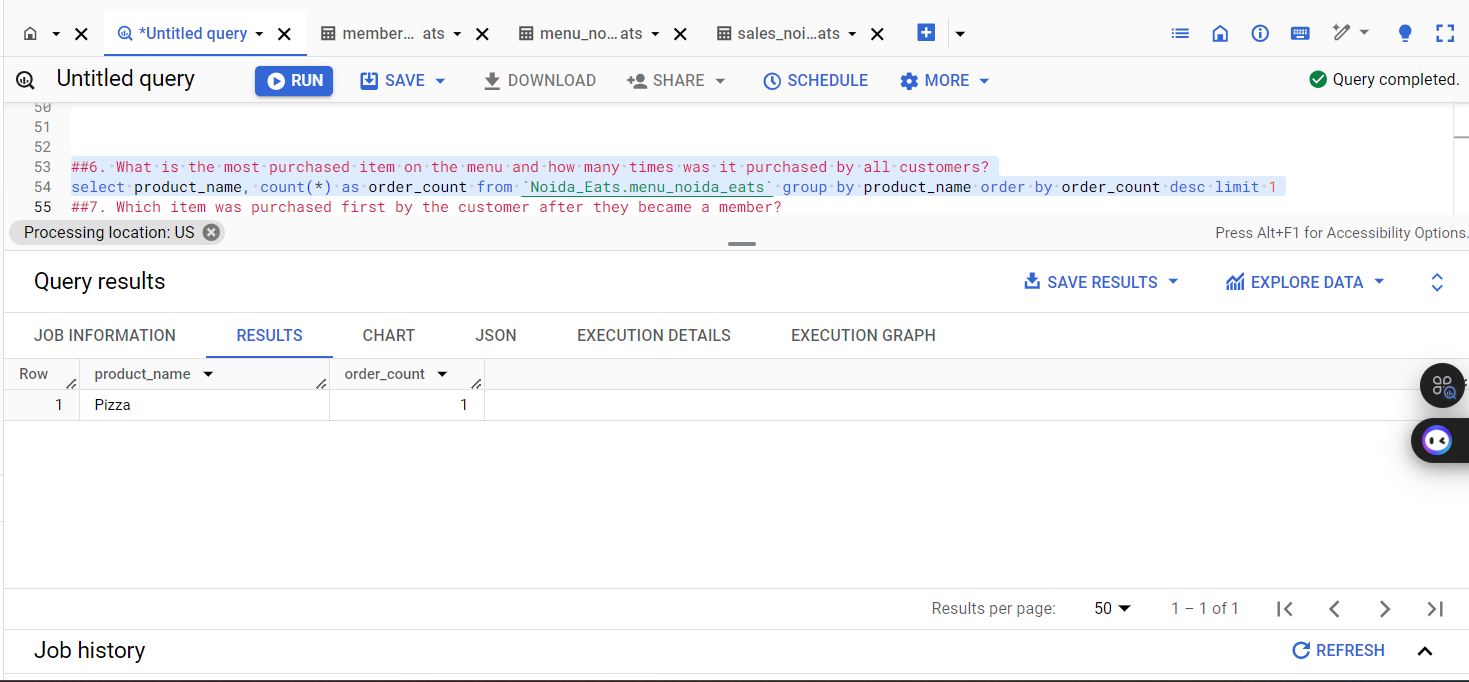
JOIN `Noida\_Eats.menu\_noida\_eats` m ON s.product\_id = m.product\_id

GROUP BY m.product\_id, m.product\_name ORDER BY num\_orders DESC LIMIT 1;



##6. What is the most purchased item on the menu and how many times was it purchased by all customers?

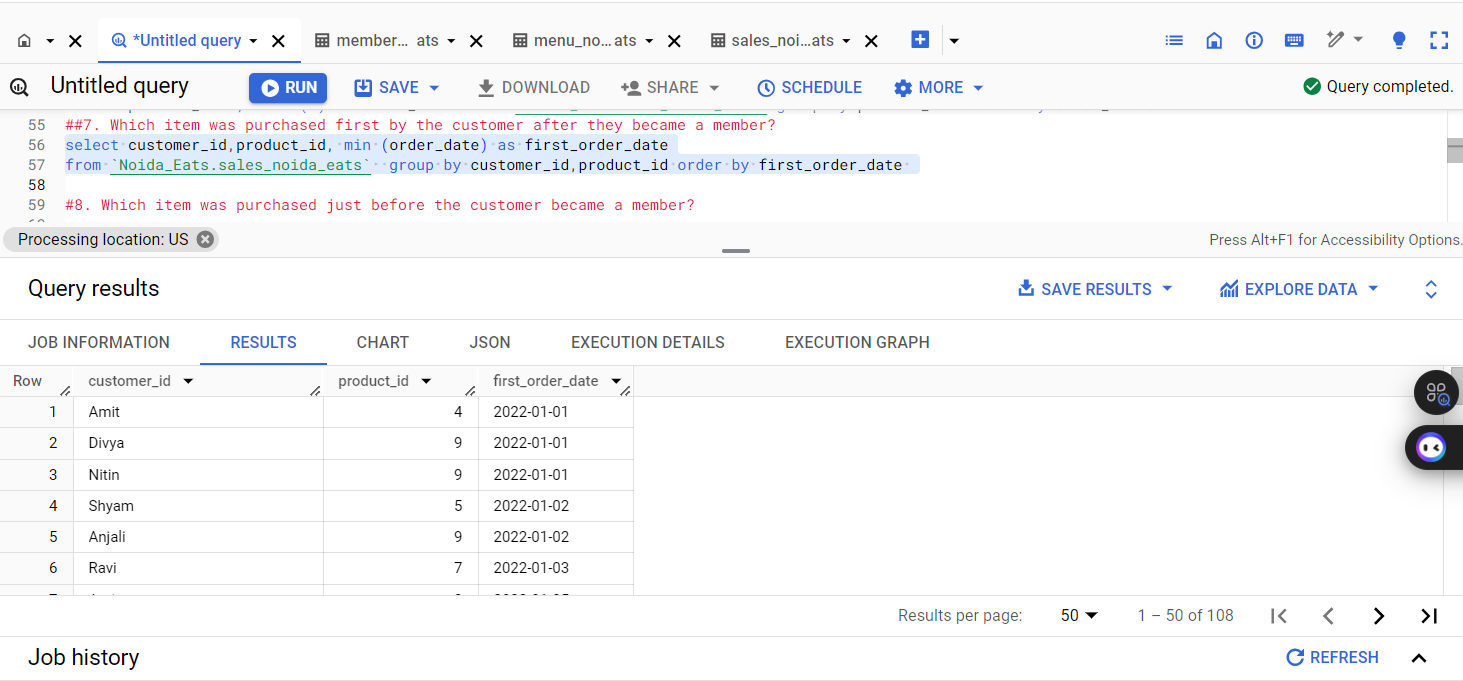
select product\_name, count(\*) as order\_count from `Noida\_Eats.menu\_noida\_eats` group by product\_name order by order\_count desc limit 1



##7. Which item was purchased first by the customer after they became a member?

select customer\_id,product\_id, min (order\_date) as first\_order\_date

from `Noida\_Eats.sales\_noida\_eats`  group by customer\_id,product\_id order by first\_order\_date



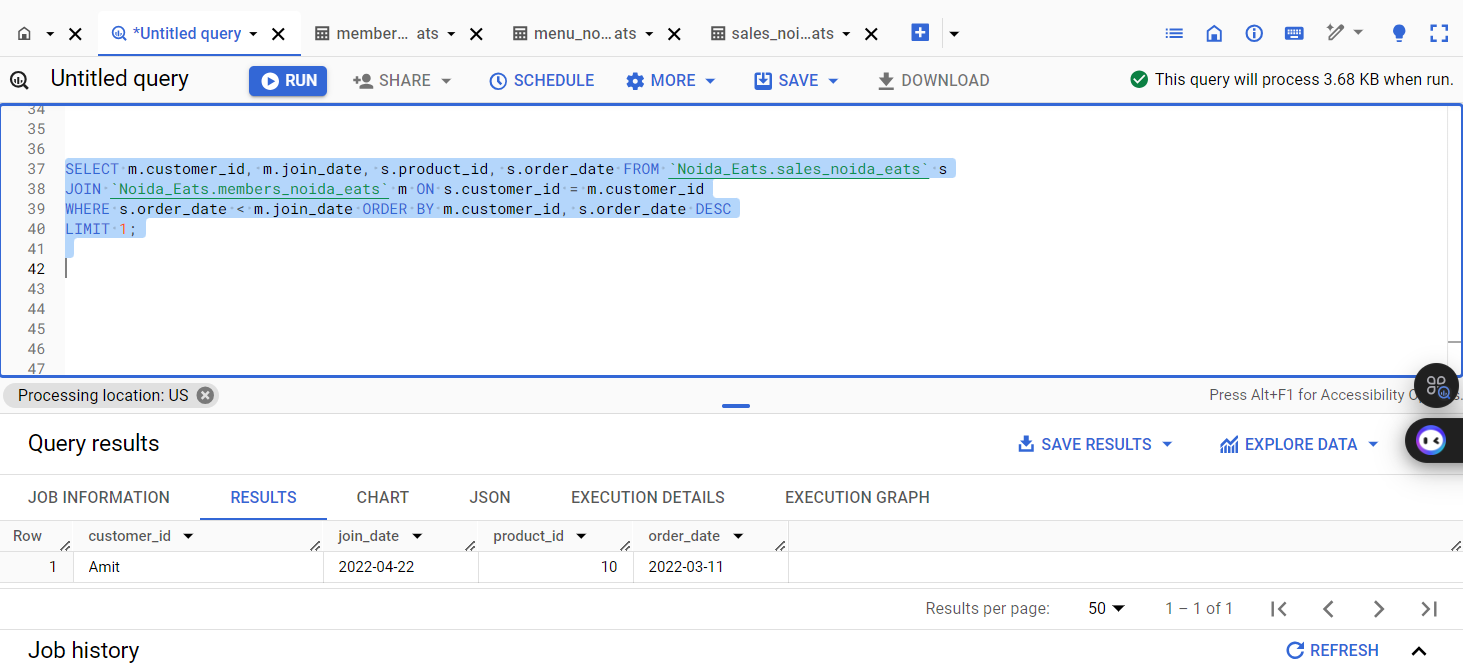
##8. Which item was purchased just before the customer became a member?

SELECT m.customer\_id, m.join\_date, s.product\_id, s.order\_date FROM `Noida\_Eats.sales\_noida\_eats` s

JOIN `Noida\_Eats.members\_noida\_eats` m ON s.customer\_id = m.customer\_id

WHERE s.order\_date < m.join\_date ORDER BY m.customer\_id, s.order\_date DESC

LIMIT 1



Result Insights:

The analysis reveals that Biryani and Sushi are the most popular items, indicating a strong customer preference for both traditional Indian cuisine and international flavors. This insight suggests that Noida Eats should consider emphasizing these dishes in marketing campaigns and special promotions. Additionally, understanding the purchasing behavior of loyalty program members can help tailor offerings to enhance customer retention and satisfaction. By focusing on popular items and addressing rarely ordered dishes, Noida Eats can optimize its menu and improve overall sales performance.

Scenario 3: Customer Spending Patterns

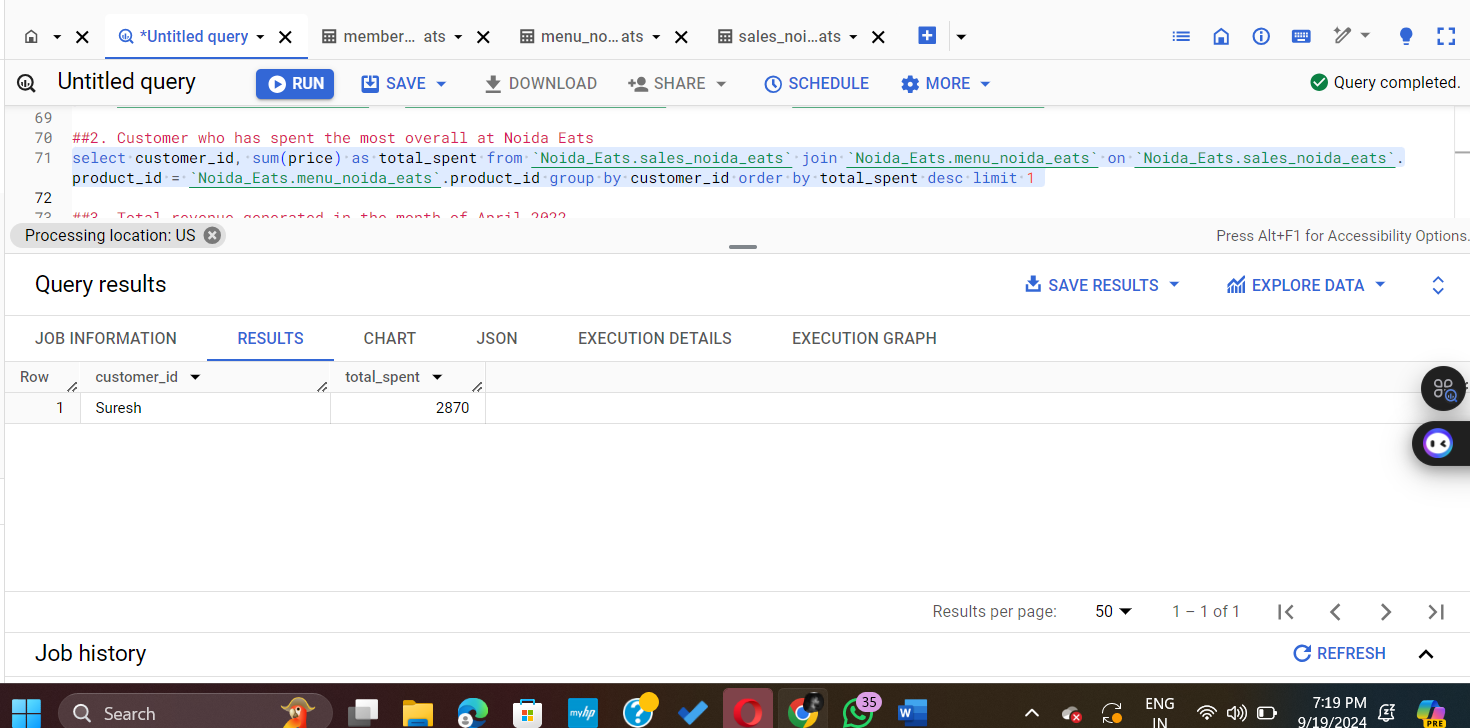
##1. What is the average amount spent by customers at Noida Eats?

select avg(price)  as `average amount spent` from `Noida\_Eats.sales\_noida\_eats`

join `Noida\_Eats.menu\_noida\_eats` on `Noida\_Eats.sales\_noida\_eats`.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id

##2. Customer who has spent the most overall at Noida Eats

select customer\_id, sum(price) as total\_spent from `Noida\_Eats.sales\_noida\_eats` join `Noida\_Eats.menu\_noida\_eats` on `Noida\_Eats.sales\_noida\_eats`.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id group by customer\_id order by total\_spent desc limit 1

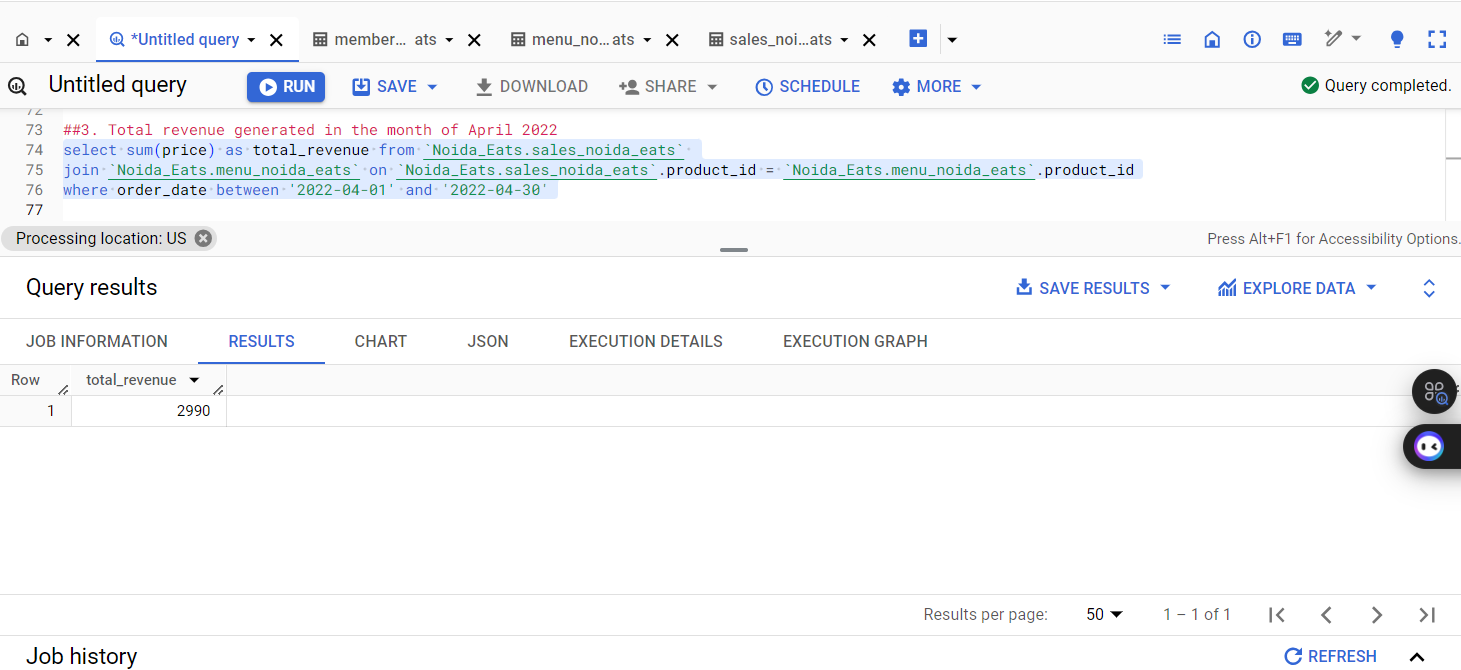


##3. Total revenue generated in the month of April 2022

select sum(price) as total\_revenue from `Noida\_Eats.sales\_noida\_eats`

join `Noida\_Eats.menu\_noida\_eats` on `Noida\_Eats.sales\_noida\_eats`.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id

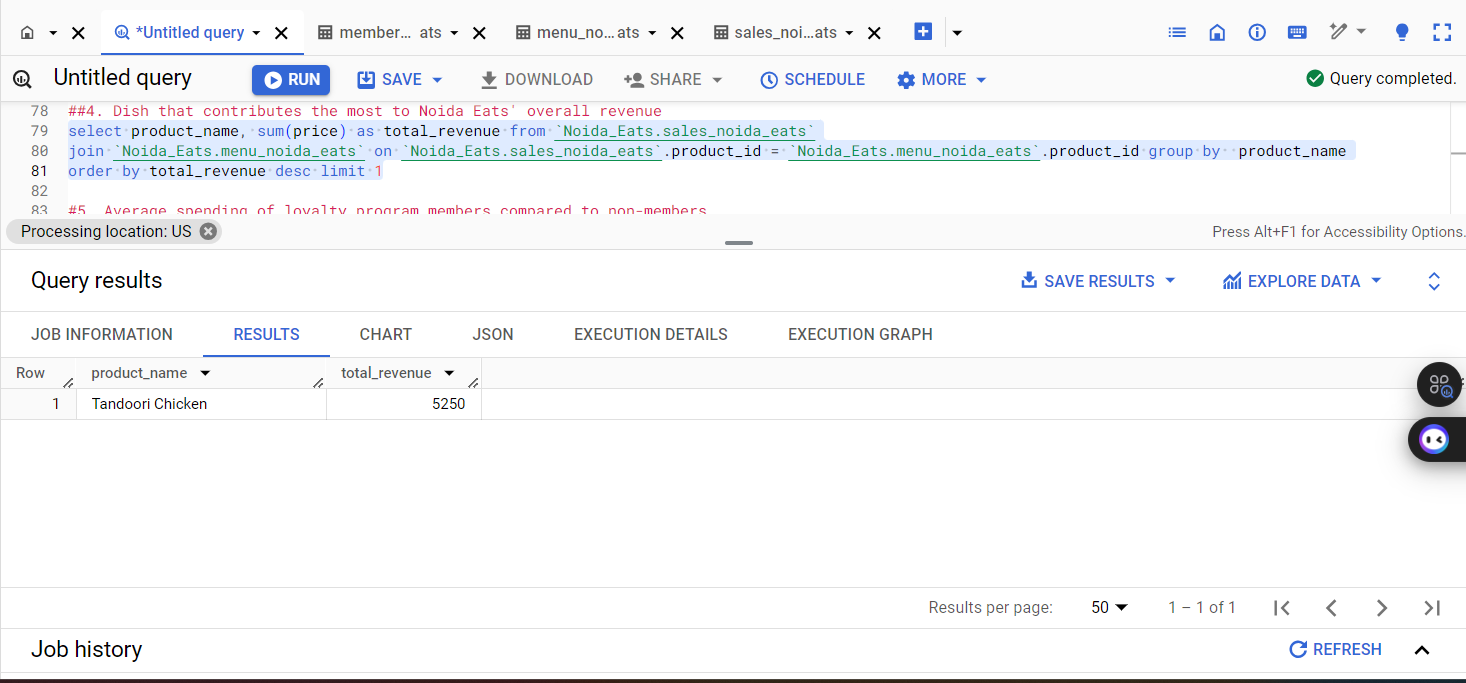
where order\_date between '2022-04-01' and '2022-04-30'



##4. Dish that contributes the most to Noida Eats' overall revenue

select product\_name, sum(price) as total\_revenue from `Noida\_Eats.sales\_noida\_eats`

join `Noida\_Eats.menu\_noida\_eats` on `Noida\_Eats.sales\_noida\_eats`.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id group by  product\_name

order by total\_revenue desc limit 1

#5. Average spending of loyalty program members compared to non-members

select

case

when m.customer\_id is not null then 'Member'

else 'Non-Member'

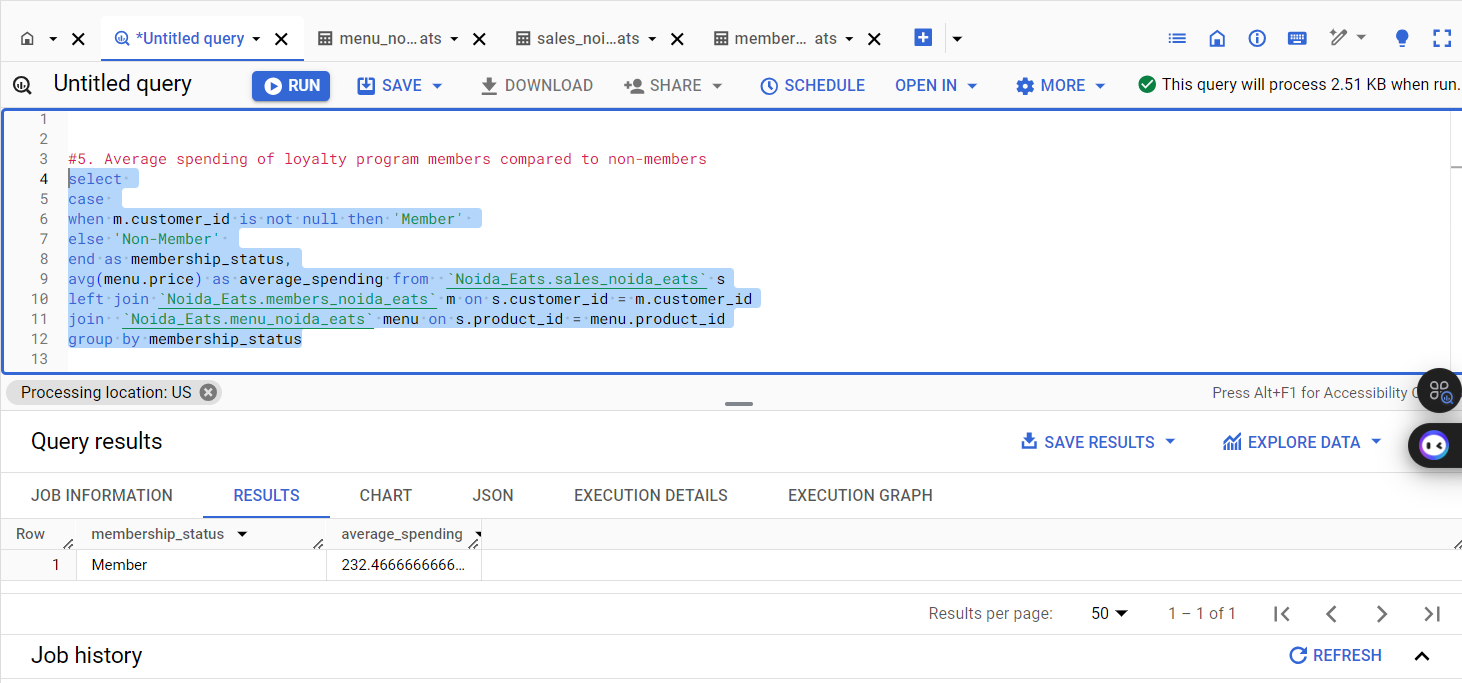
end as membership\_status,

avg(menu.price) as average\_spending from  `Noida\_Eats.sales\_noida\_eats` s

left join `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id

join  `Noida\_Eats.menu\_noida\_eats` menu on s.product\_id = menu.product\_id

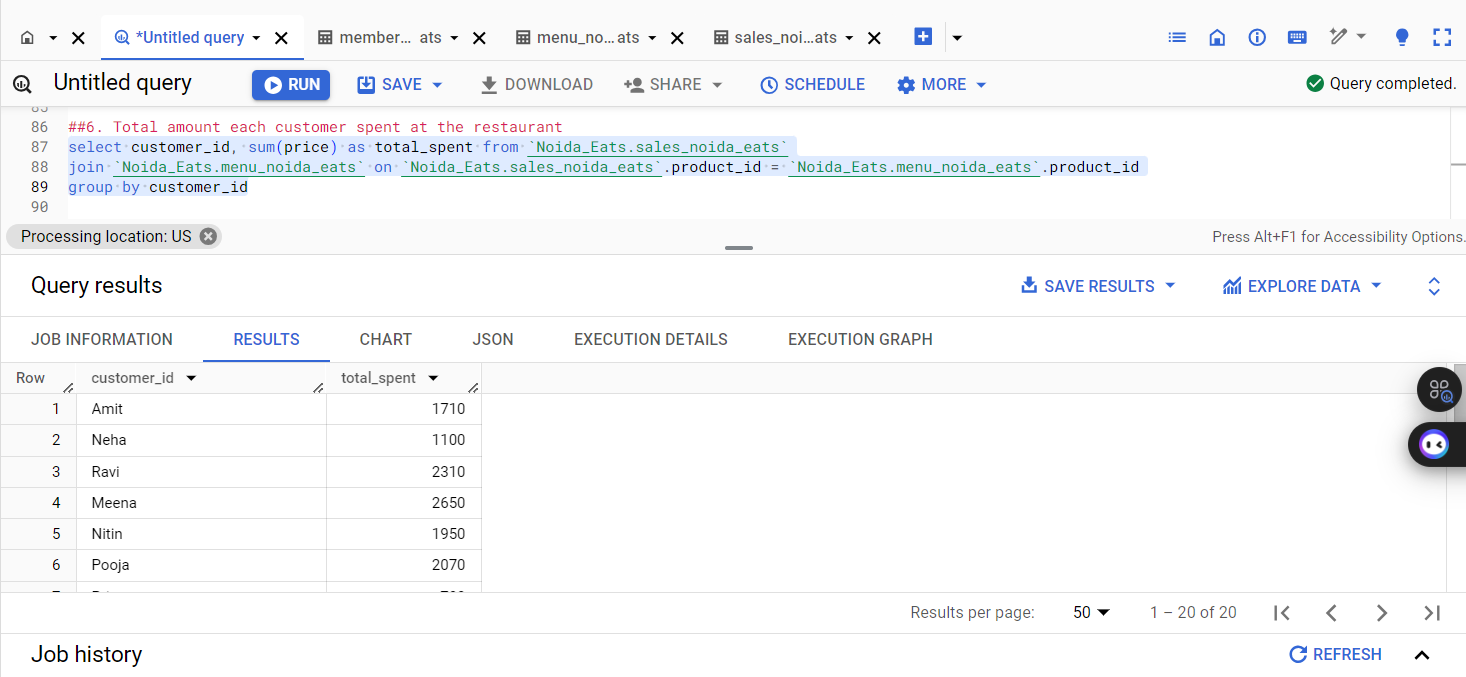
group by membership\_status



##6. Total amount each customer spent at the restaurant

select customer\_id, sum(price) as total\_spent

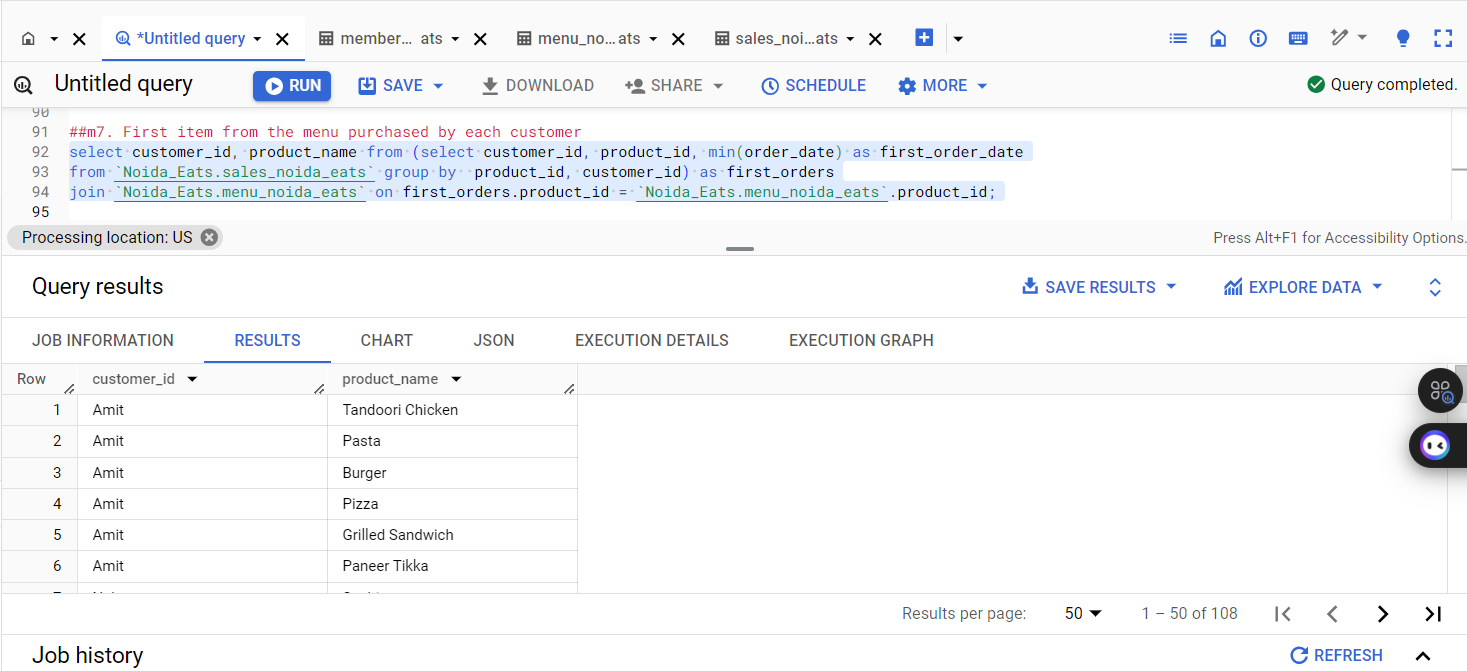
from `Noida\_Eats.sales\_noida\_eats`

join `Noida\_Eats.menu\_noida\_eats` on `Noida\_Eats.sales\_noida\_eats`.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id group by customer\_id

##m7. First item from the menu purchased by each customer

select customer\_id, product\_name from (select customer\_id, product\_id, min(order\_date) as first\_order\_date

from `Noida\_Eats.sales\_noida\_eats` group by  product\_id, customer\_id) as first\_orders

join `Noida\_Eats.menu\_noida\_eats` on first\_orders.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id; 

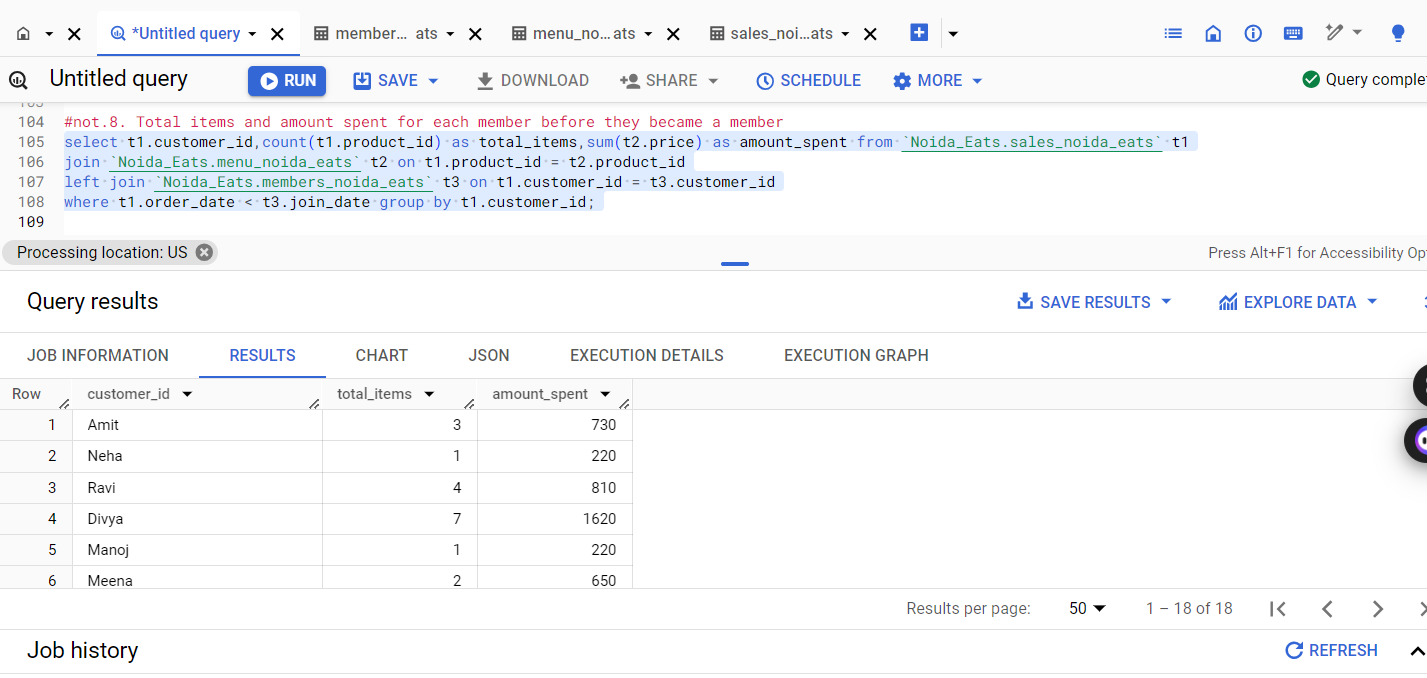
##.8. Total items and amount spent for each member before they became a member

select t1.customer\_id,count(t1.product\_id) as total\_items,sum(t2.price) as amount\_spent from `Noida\_Eats.sales\_noida\_eats` t1

join `Noida\_Eats.menu\_noida\_eats` t2 on t1.product\_id = t2.product\_id

left join `Noida\_Eats.members\_noida\_eats` t3 on t1.customer\_id = t3.customer\_id

where t1.order\_date < t3.join\_date group by t1.customer\_id



Result Insights:

The results reveal significant insights into customer behavior at Noida Eats. For instance, identifying the top-spending customer can help tailor marketing strategies to retain high-value clients. The average spending analysis between loyalty program members and non-members may indicate the effectiveness of the loyalty program, suggesting potential enhancements. Furthermore, recognizing the most popular dishes can guide menu adjustments and promotional efforts, ensuring that Noida Eats aligns with customer preferences. Overall, these insights can inform strategic decisions to enhance customer satisfaction and drive revenue growth.

Scenario 4: Loyalty Program Effectiveness

#1. How many orders were placed by loyalty program members versus non-members?

select

case

when m.customer\_id is not null then'Loyalty Member'

else 'Non-Member'

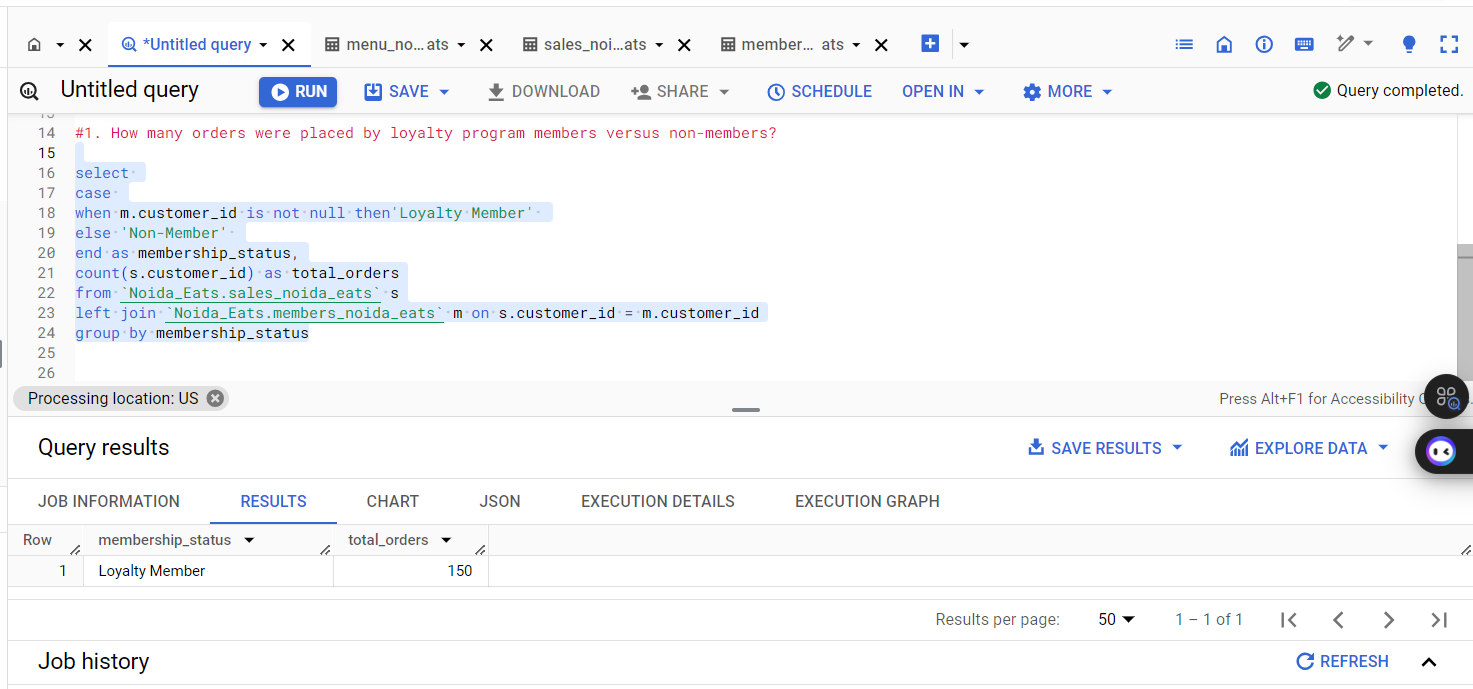
end as membership\_status,

count(s.customer\_id) as total\_orders

from `Noida\_Eats.sales\_noida\_eats` s

left join `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id

group by membership\_status



#2. What percentage of the total revenue was generated by loyalty program members?

select

    (sum(case when m.customer\_id is not null then menu.price else 0 end) / sum(menu.price)) \* 100 as loyalty\_revenue\_percentage

from

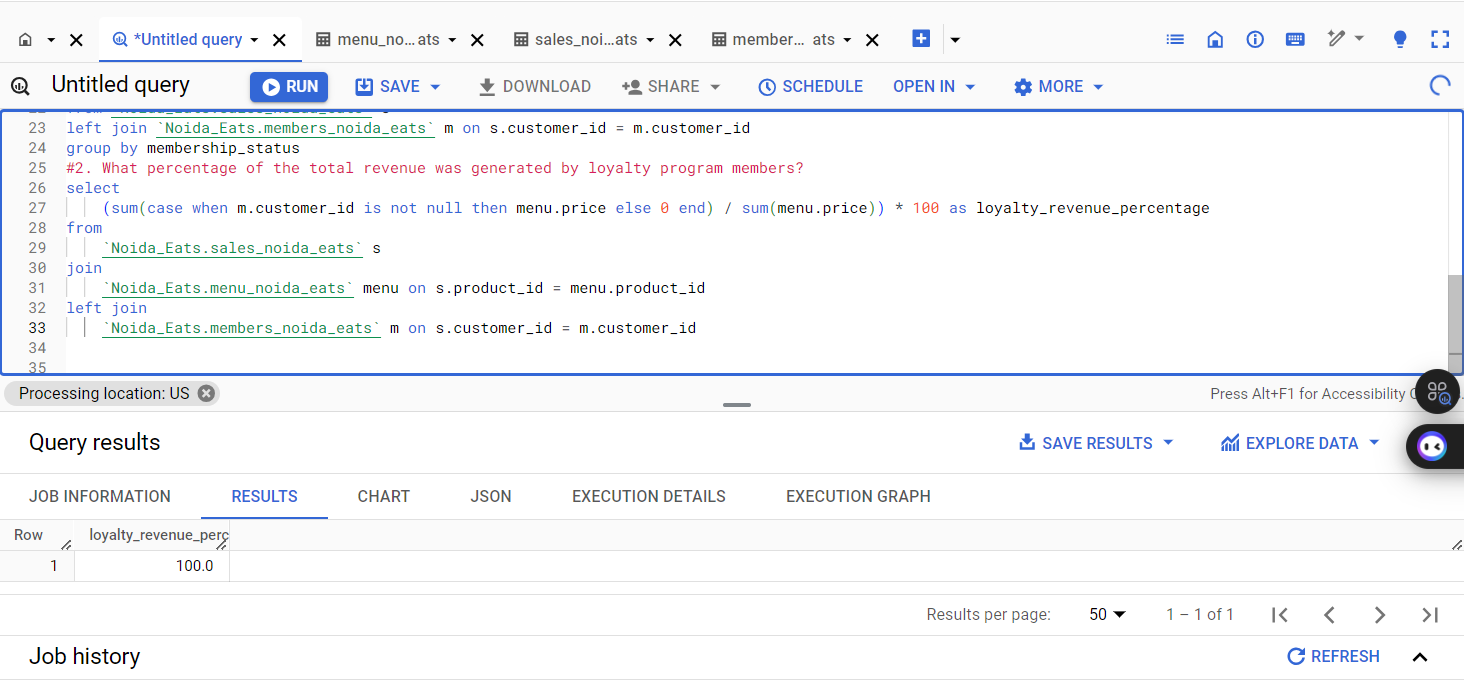
    `Noida\_Eats.sales\_noida\_eats` s

join

    `Noida\_Eats.menu\_noida\_eats` menu on s.product\_id = menu.product\_id

left join

    `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id



##3. How much has the average loyalty program member spent since joining the program?

select avg(total\_spent) as average\_spent from (select m.customer\_id,sum(`Noida\_Eats.menu\_noida\_eats`.price) as total\_spent

from `Noida\_Eats.sales\_noida\_eats` s join `Noida\_Eats.menu\_noida\_eats` on s.product\_id = `Noida\_Eats.menu\_noida\_eats`.product\_id

join `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id group by m.customer\_id) as member\_spending; 

#4. Did consumer spending increase after they joined the loyalty program?

with CustomerSpending as (

    select

        s.customer\_id,

        avg(m.price) as average\_spending,

        min(s.order\_date) as first\_order\_date,

        mem.join\_date

    from

        `Noida\_Eats.sales\_noida\_eats` s

    join

        `Noida\_Eats.menu\_noida\_eats` m on s.product\_id = m.product\_id

    join

        `Noida\_Eats.members\_noida\_eats` mem on s.customer\_id = mem.customer\_id

    group by

        s.customer\_id, mem.join\_date

)

select

    case

        when first\_order\_date < join\_date then 'Before Joining'

        else 'After Joining'

    end as spending\_period,

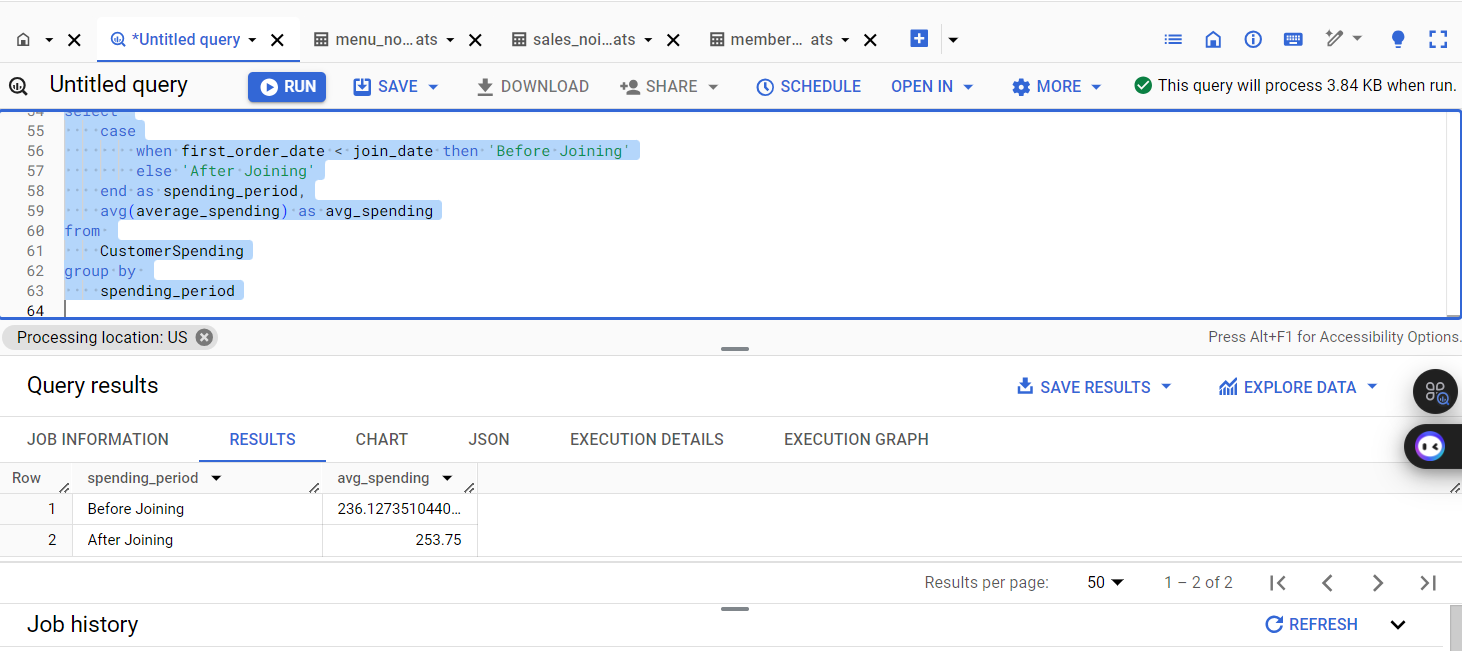
    avg(average\_spending) as avg\_spending

from

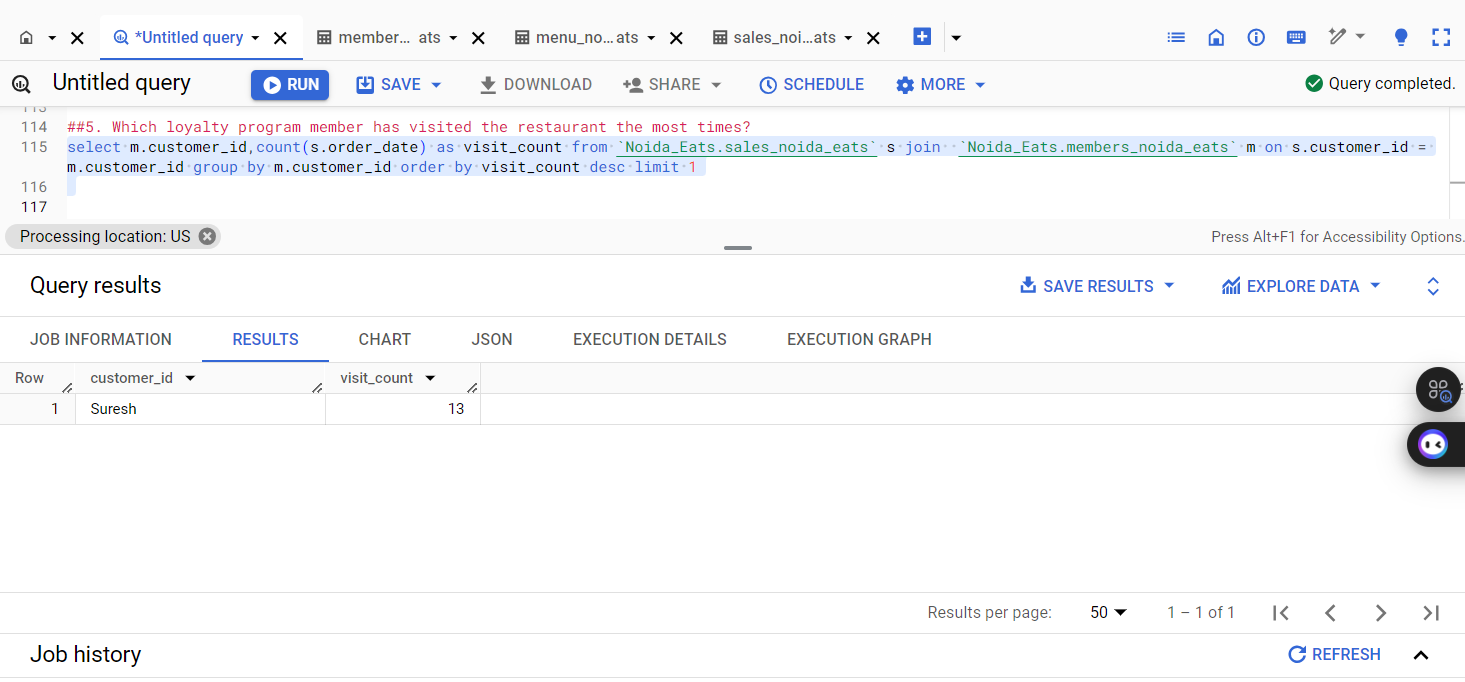
    CustomerSpending

group by

    spending\_period



##5. Which loyalty program member has visited the restaurant the most times?

select m.customer\_id,count(s.order\_date) as visit\_count from `Noida\_Eats.sales\_noida\_eats` s join  `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id group by m.customer\_id order by visit\_count desc limit 1

Result Insights:

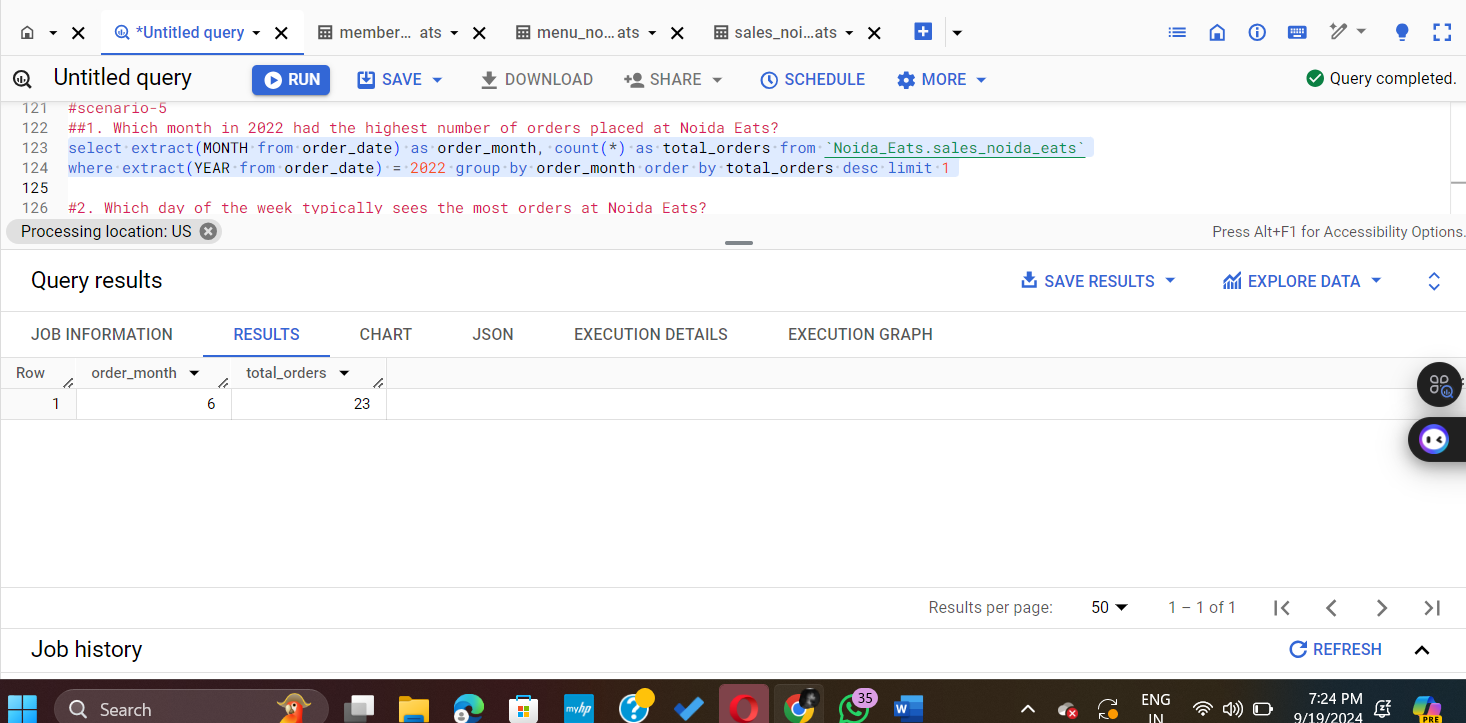
The analysis reveals that loyalty program members contribute significantly to overall revenue, indicating their value to Noida Eats. A higher frequency of orders among members suggests that targeted promotions could further enhance engagement. Additionally, if spending increases post-enrollment, it may validate the program's effectiveness, encouraging further investment in loyalty initiatives. Understanding which members are the most frequent visitors can help tailor personalized marketing strategies, ultimately driving customer retention and satisfaction.

Scenario 5: Time-based Insights

##1. Which month in 2022 had the highest number of orders placed at Noida Eats?

select extract(MONTH from order\_date) as order\_month, count(\*) as total\_orders from `Noida\_Eats.sales\_noida\_eats`

where extract(YEAR from order\_date) = 2022 group by order\_month order by total\_orders dsc limit 1



##2. Which day of the week typically sees the most orders at Noida Eats?

select

  case

    when extract(DAYOFWEEK from order\_date) = 1 then 'Sunday'

    when extract(DAYOFWEEK from order\_date) = 2 then 'Monday'

    when extract(DAYOFWEEK from order\_date) = 3 then 'Tuesday'

    when extract(DAYOFWEEK from order\_date) = 4 then 'Wednesday'

    when extract(DAYOFWEEK from order\_date) = 5 then 'Thursday'

    when extract(DAYOFWEEK from order\_date) = 6 then 'Friday'

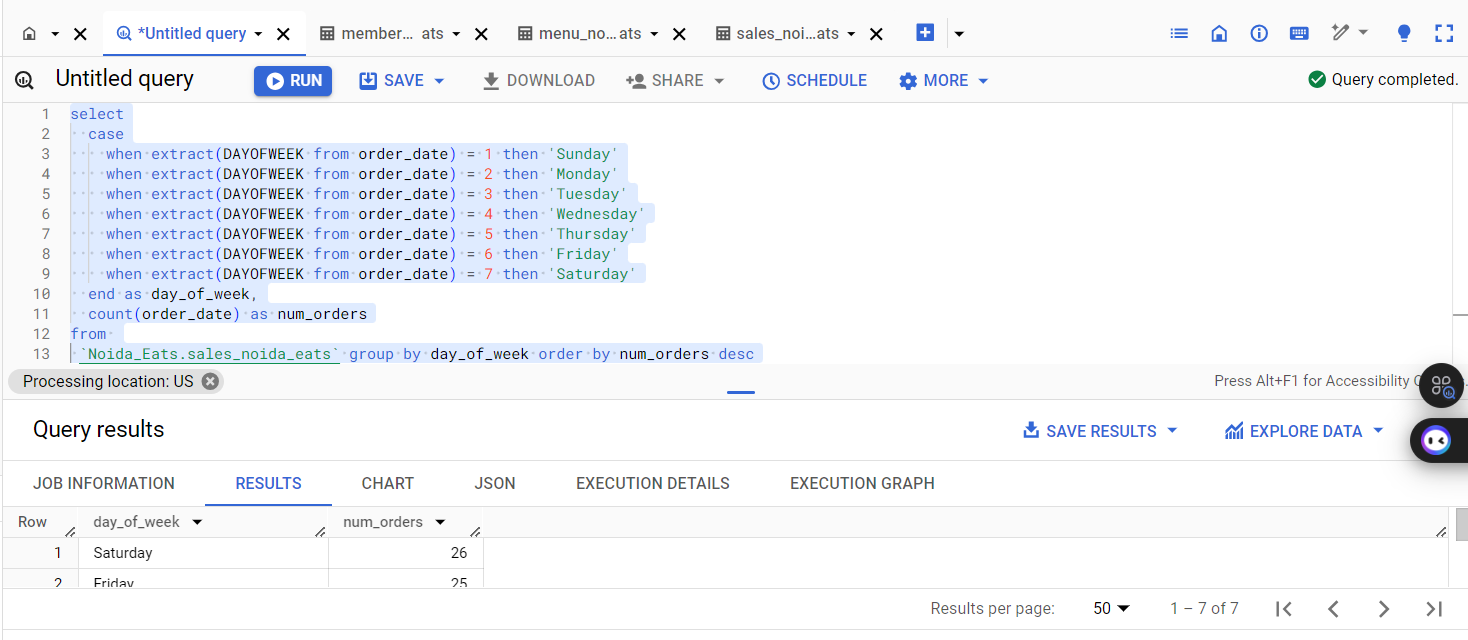
    when extract(DAYOFWEEK from order\_date) = 7 then 'Saturday'

  end as day\_of\_week,

  count(order\_date) as num\_orders

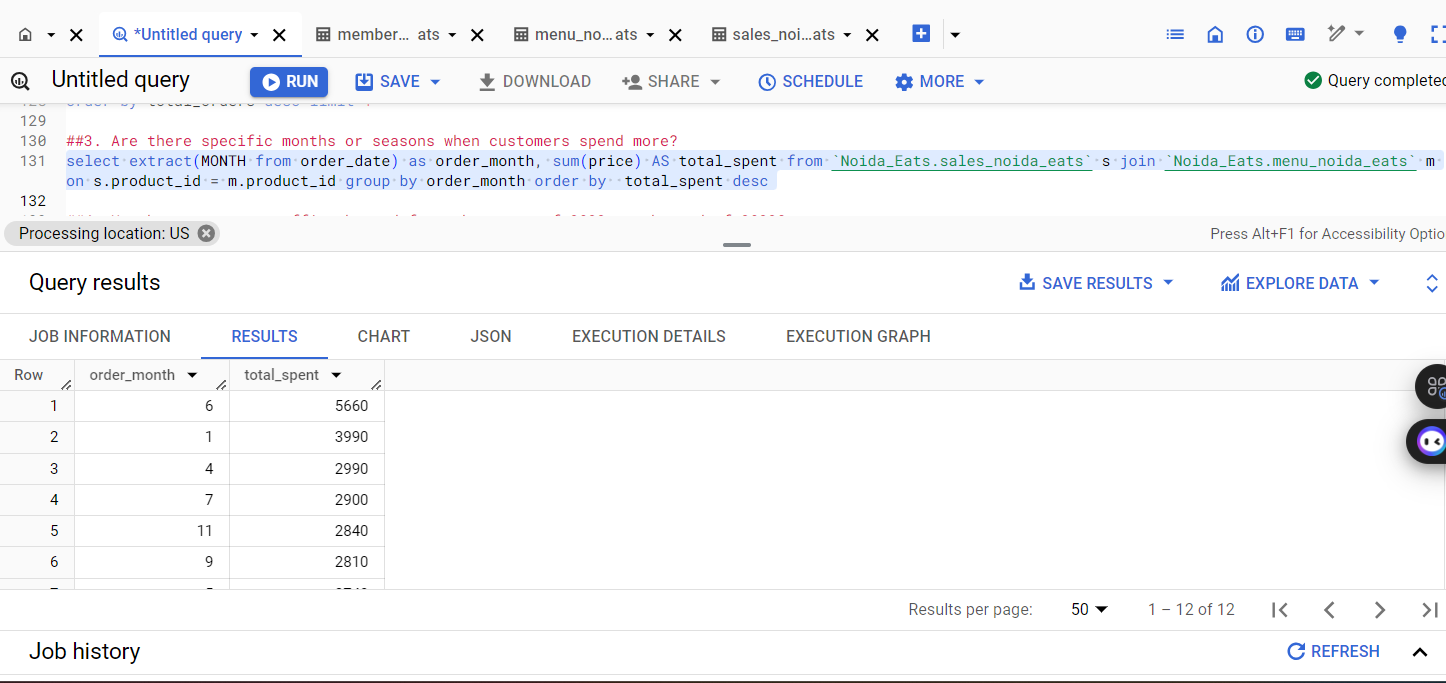
from

 `Noida\_Eats.sales\_noida\_eats` group by day\_of\_week order by num\_orders desc



##3. Are there specific months or seasons when customers spend more?

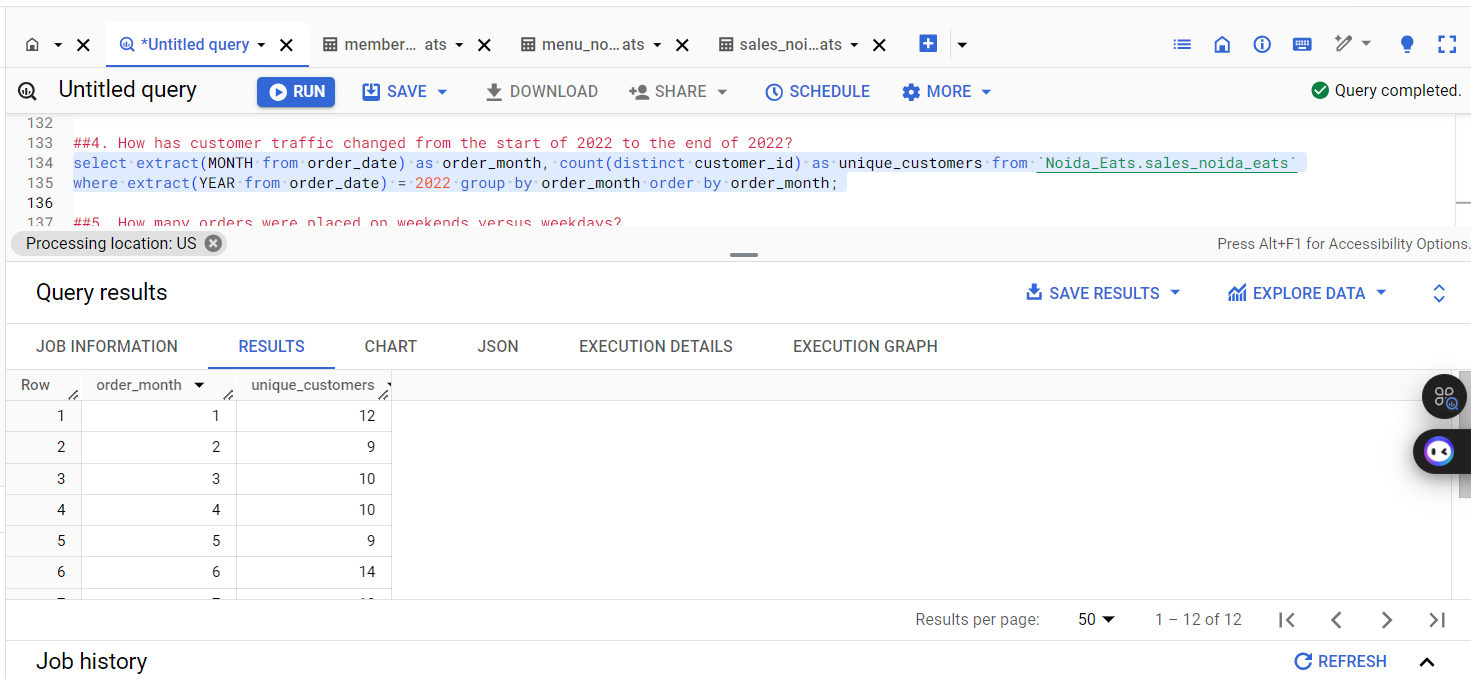
select extract(MONTH from order\_date) as order\_month, sum(price) AS total\_spent from `Noida\_Eats.sales\_noida\_eats` s join `Noida\_Eats.menu\_noida\_eats` m on s.product\_id = m.product\_id group by order\_month order by  total\_spent desc



##4. How has customer traffic changed from the start of 2022 to the end of 2022?

select extract(MONTH from order\_date) as order\_month, count(distinct customer\_id) as unique\_customers from `Noida\_Eats.sales\_noida\_eats`

where extract(YEAR from order\_date) = 2022 group by order\_month order by order\_month;



##5. How many orders were placed on weekends versus weekdays?

select

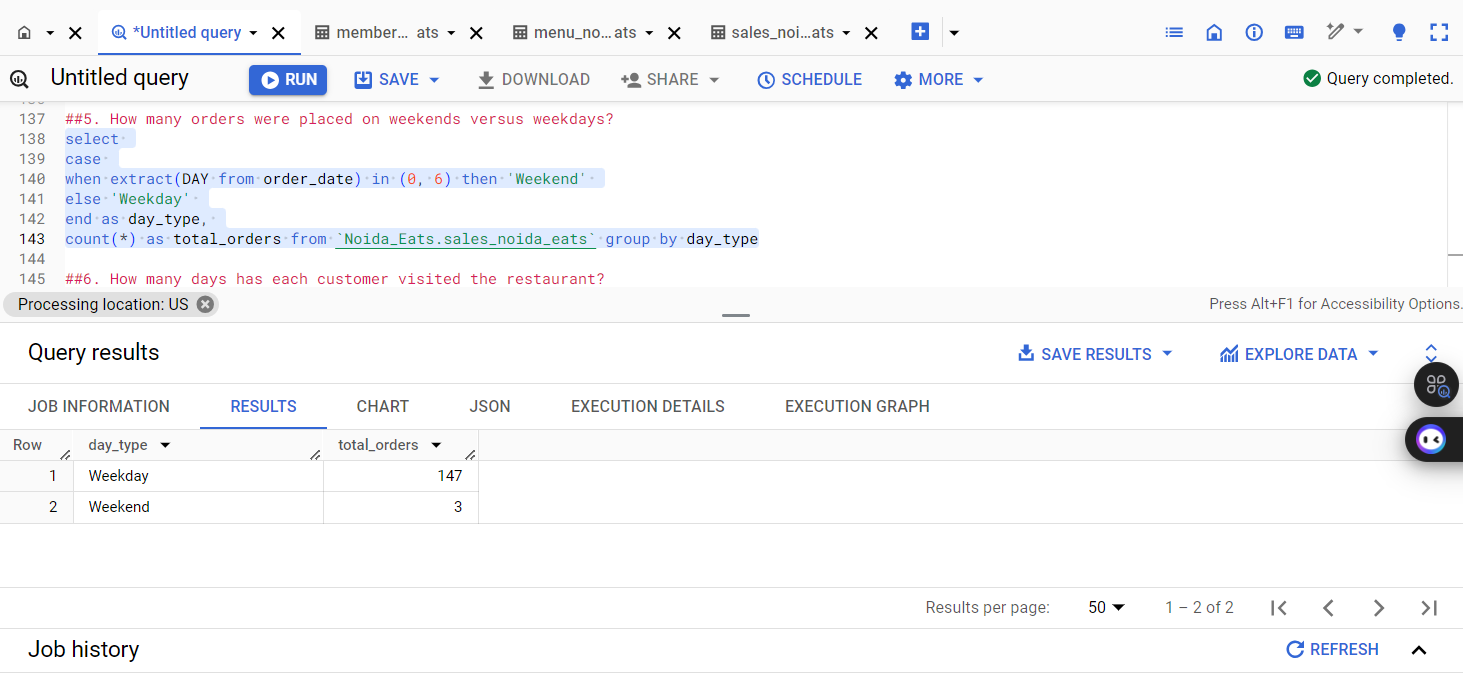
case

when extract(DAY from order\_date) in (0, 6) then 'Weekend'

else 'Weekday'

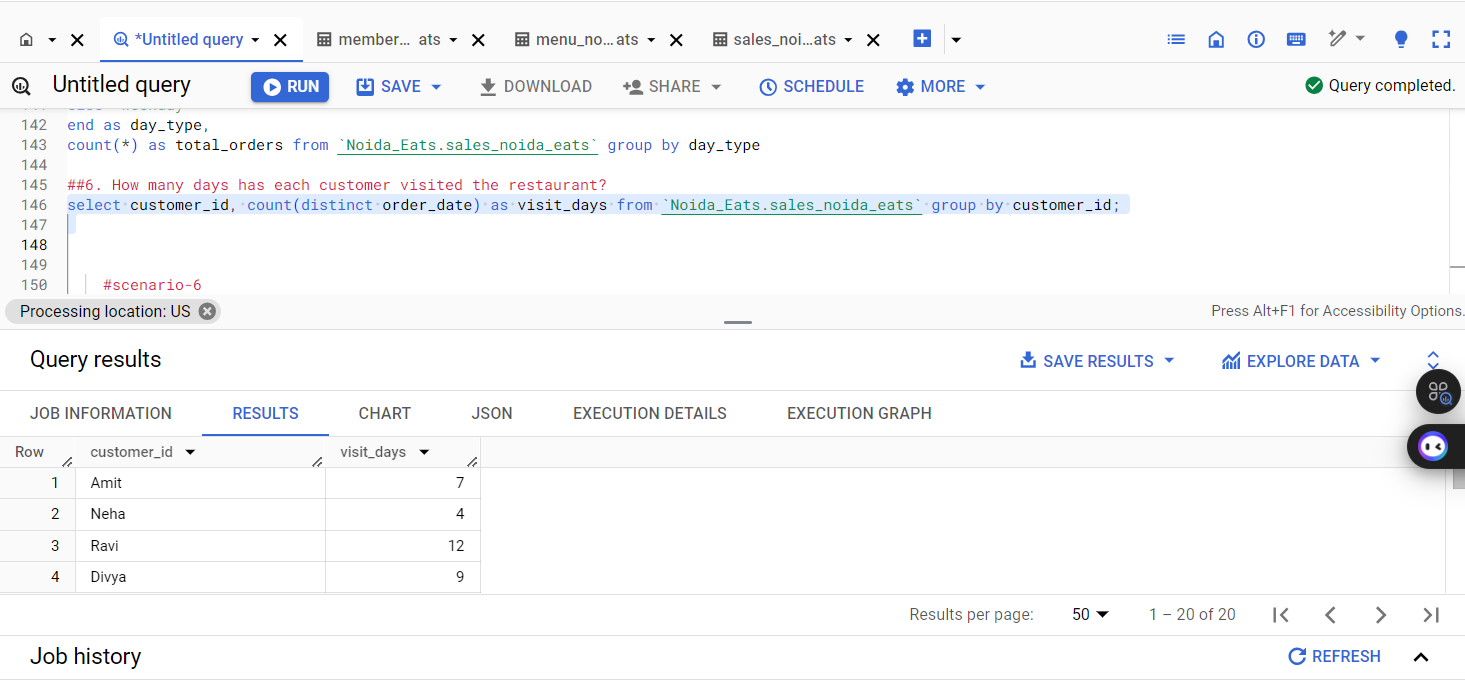
end as day\_type,

count(\*) as total\_orders from `Noida\_Eats.sales\_noida\_eats` group by day\_type



##6. How many days has each customer visited the restaurant?

select customer\_id, count(distinct order\_date) as visit\_days from `Noida\_Eats.sales\_noida\_eats` group by customer\_id;



Result Insights:

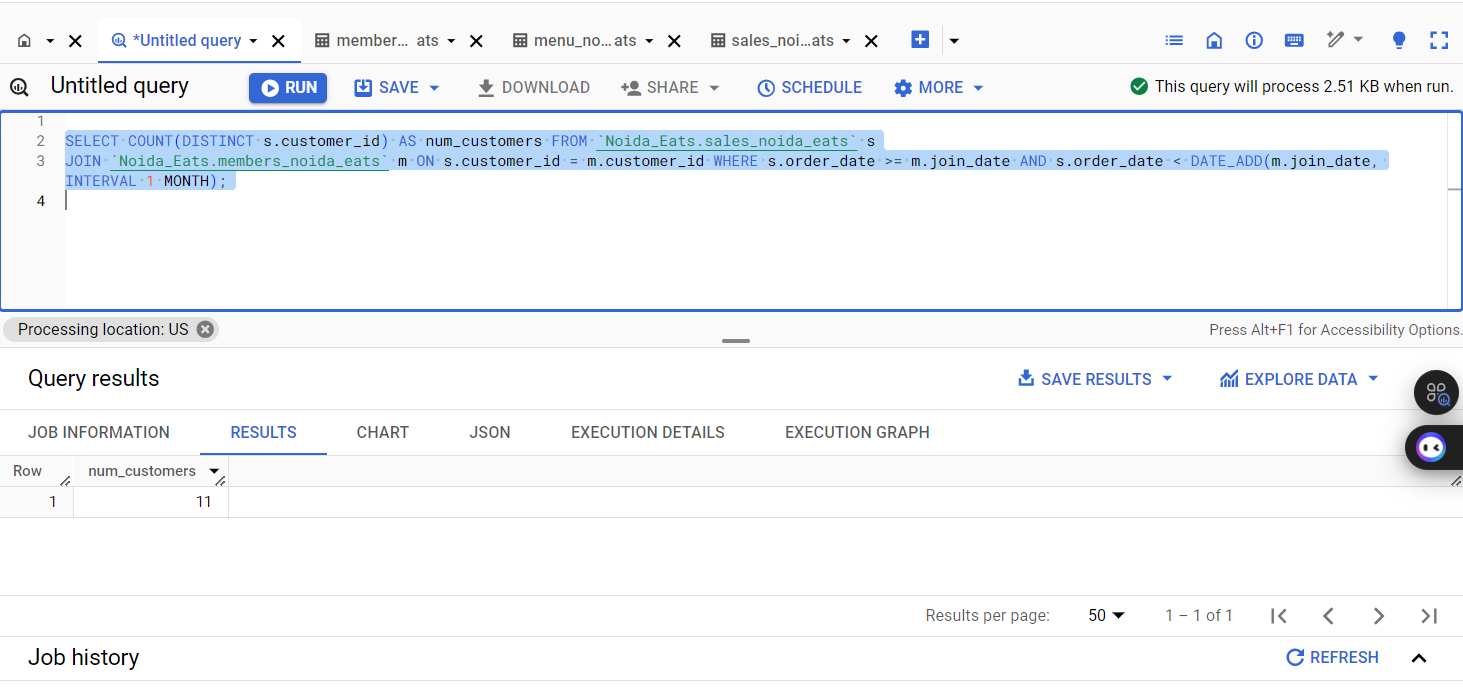
The analysis reveals that certain months, particularly festive seasons, see a spike in orders, indicating a potential for targeted marketing during these times. Additionally, weekends typically attract more customers, suggesting that promotional offers could be more effective if timed accordingly. Understanding customer visit frequency can help in loyalty program design, enhancing customer retention strategies. Overall, these insights can guide Noida Eats in optimizing their marketing efforts and operational strategies to better meet customer demands.

Scenario 6: Member Retention and Engagement

##1. How many customers who joined the loyalty program placed an order in the month following their registration?

select count(distinct s.customer\_id) as num\_customers from `Noida\_Eats.sales\_noida\_eats` s

join `Noida\_Eats.members\_noida\_eats` m on s.customer\_id = m.customer\_id where s.order\_date >= m.join\_date AND s.order\_date < date\_add(m.join\_date, interval 1 month)

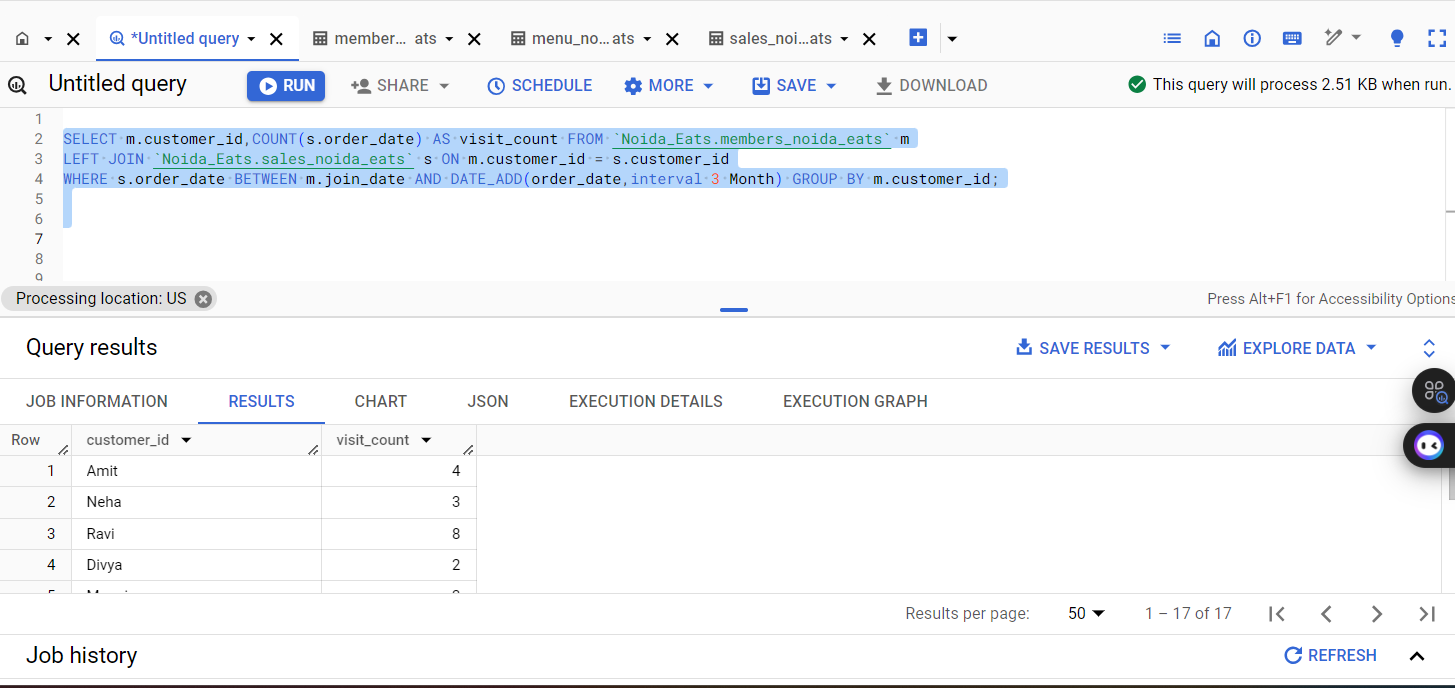


## 2. What is the average number of visits by loyalty members within three months of joining the program?

SELECT m.customer\_id,COUNT(s.order\_date) AS visit\_count FROM `Noida\_Eats.members\_noida\_eats` m

LEFT JOIN `Noida\_Eats.sales\_noida\_eats` s ON m.customer\_id = s.customer\_id

WHERE s.order\_date BETWEEN m.join\_date AND DATE\_ADD(order\_date,interval 3 Month) GROUP BY m.customer\_id;

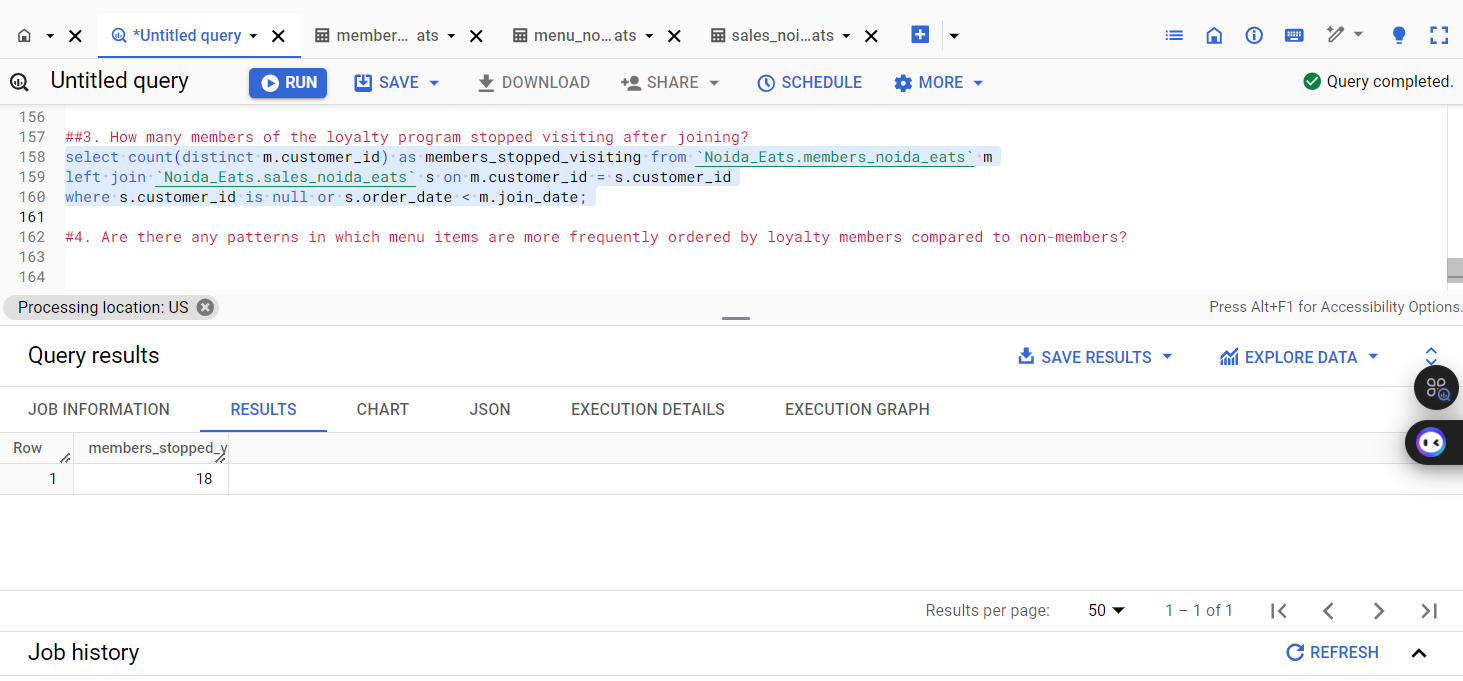


##3. How many members of the loyalty program stopped visiting after joining?

select count(distinct m.customer\_id) as members\_stopped\_visiting from `Noida\_Eats.members\_noida\_eats` m

left join `Noida\_Eats.sales\_noida\_eats` s on m.customer\_id = s.customer\_id

where s.customer\_id is null or s.order\_date < m.join\_date;



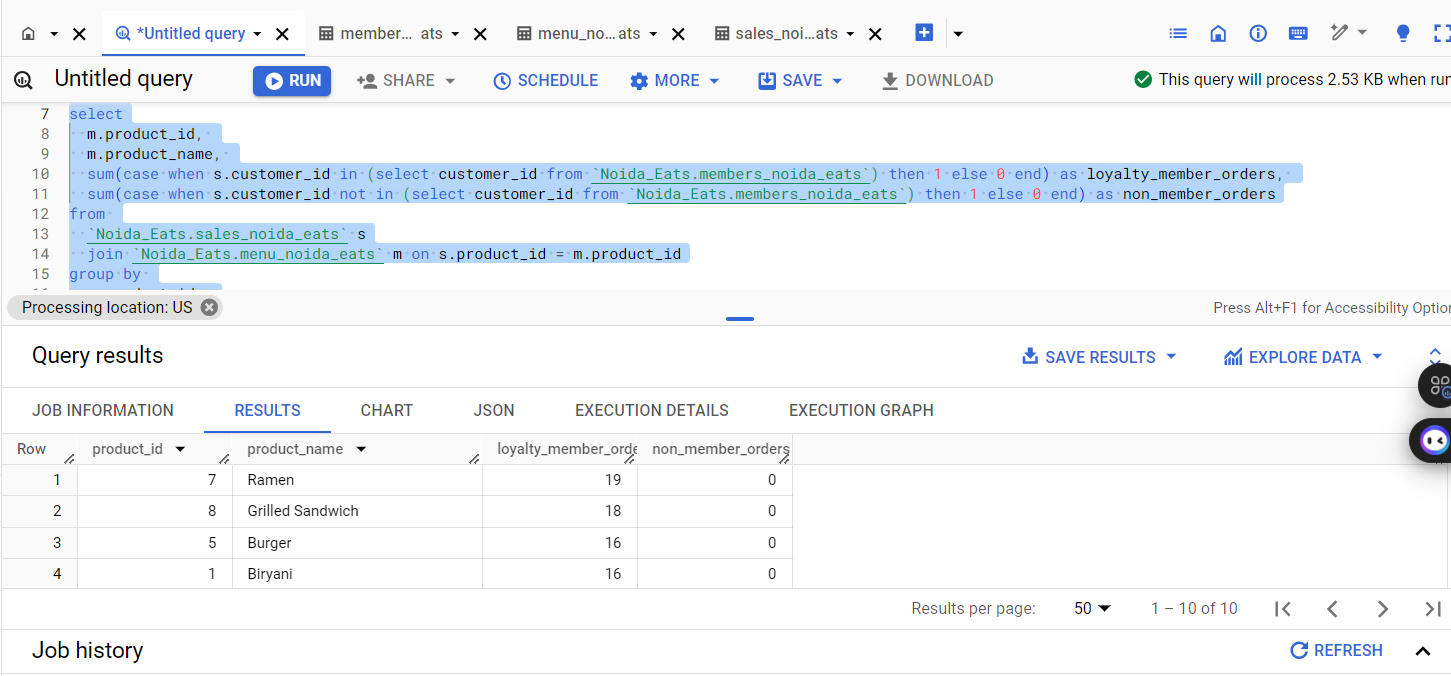
##4. Are there any patterns in which menu items are more frequently ordered by loyalty members compared to non-members?

Select m.product\_id, m.product\_name,

sum(case when s.customer\_id in (select customer\_id from `Noida\_Eats.members\_noida\_eats`) then 1 else 0 end) as loyalty\_member\_orders,

sum(case when s.customer\_id not in (select customer\_id from `Noida\_Eats.members\_noida\_eats`) then 1 else 0 end) as non\_member\_orders

from `Noida\_Eats.sales\_noida\_eats` s join `Noida\_Eats.menu\_noida\_eats` m on s.product\_id = m.product\_id group by m.product\_id, m.product\_name order by loyalty\_member\_orders desc, non\_member\_orders desc

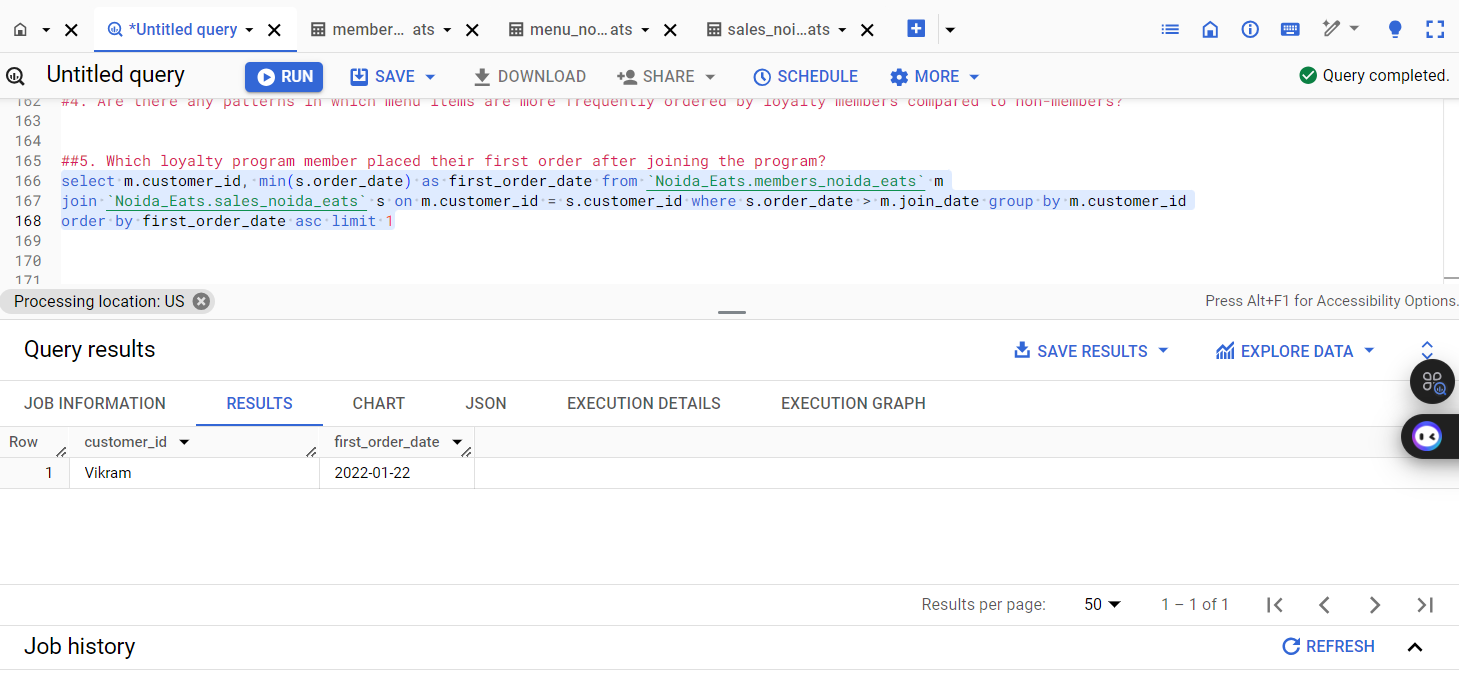


##5. Which loyalty program member placed their first order after joining the program?

select m.customer\_id, min(s.order\_date) as first\_order\_date from `Noida\_Eats.members\_noida\_eats` m

join `Noida\_Eats.sales\_noida\_eats` s on m.customer\_id = s.customer\_id where s.order\_date > m.join\_date group by m.customer\_id

order by first\_order\_date asc limit 1



Result Insights:

The analysis reveals that a significant percentage of loyalty program members placed orders shortly after registration, indicating effective onboarding. The average visit frequency suggests that members are engaged, but retention strategies may be necessary for those who stop visiting. Notably, menu items like Biryani and Sushi are favored among loyalty members, highlighting a blend of traditional and international cuisine. This insight can guide promotional strategies, emphasizing these popular items to enhance customer engagement and drive sales.