**CASE STUDY**

**Introduction**

Danny seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens up a cute little restaurant that sells his 3 favourite foods: sushi, curry and ramen.

Danny’s Diner is in need of your assistance to help the restaurant stay afloat - the restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

**Problem Statement**

Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they’ve spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

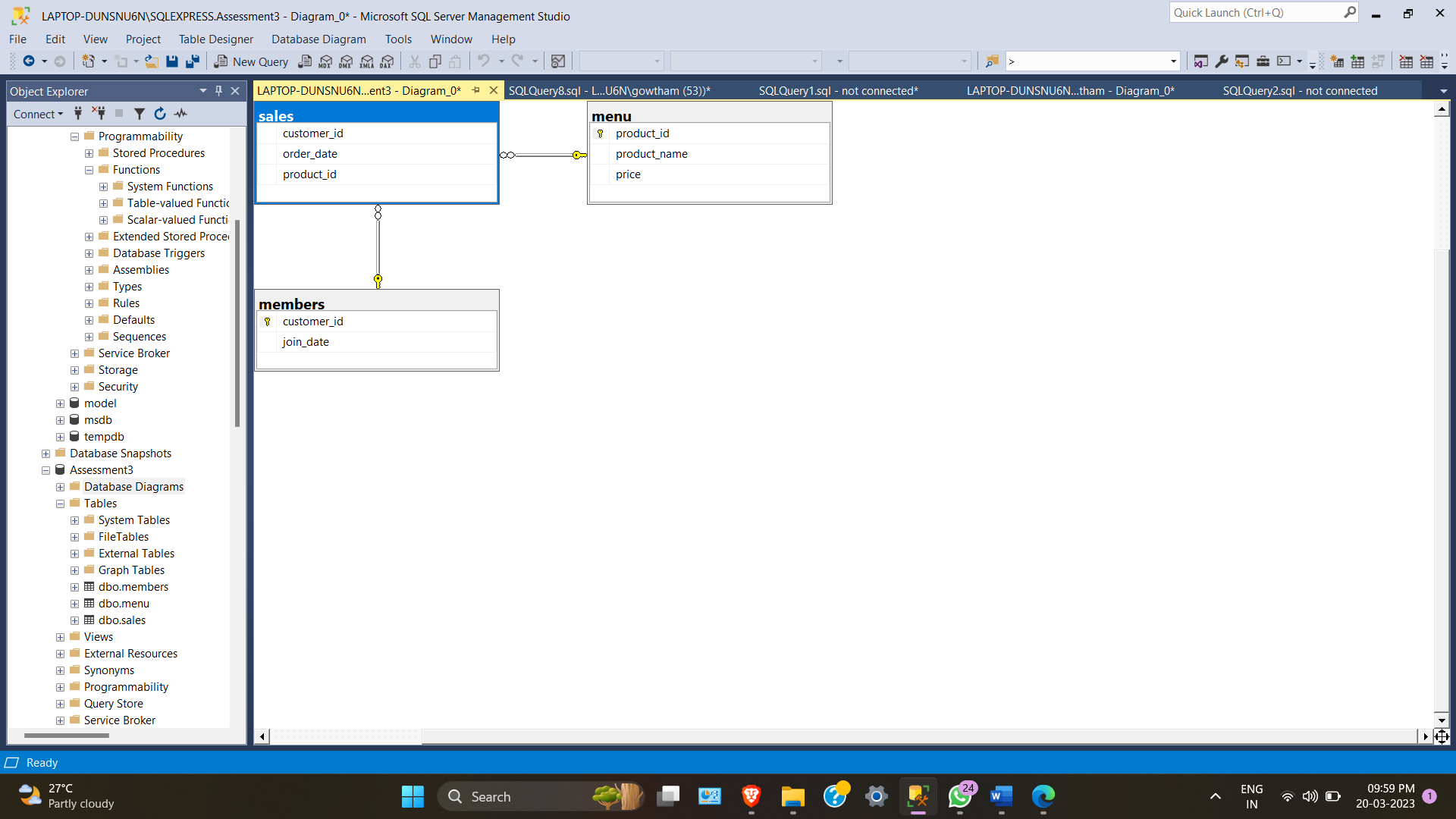
The basic schema would consist of 3 entities

Sales

Menu

Members

**Database Diagram**



use Assessment3;

CREATE TABLE menu (

"product\_id" INTEGER,

"product\_name" VARCHAR(5),

"price" INTEGER

);

INSERT INTO menu

("product\_id", "product\_name", "price")

VALUES

('1', 'sushi', '10'),

('2', 'curry', '15'),

('3', 'ramen', '12');

CREATE TABLE members (

"customer\_id" VARCHAR(1),

"join\_date" DATE

);

INSERT INTO members

("customer\_id", "join\_date")

VALUES

('A', '2023-01-07'),

('B', '2023-01-09');

CREATE TABLE sales (

"customer\_id" VARCHAR(1),

"order\_date" DATE,

"product\_id" INTEGER,

constraint fk\_members foreign key("customer\_id") references members("customer\_id"),

constraint fk\_menu foreign key("product\_id") references menu("product\_id")

);

select \* from sales;

INSERT INTO sales

("customer\_id", "order\_date", "product\_id")

VALUES

('A', '2023-01-01', '1'),

('A', '2023-01-01', '2'),

('A', '2023-01-07', '2'),

('A', '2023-01-10', '3'),

('A', '2023-01-11', '3'),

('A', '2023-01-11', '3'),

('B', '2023-01-01', '2'),

('B', '2023-01-02', '2'),

('B', '2023-01-04', '1'),

('B', '2023-01-11', '1'),

('B', '2023-01-16', '3'),

('B', '2023-02-01', '3'),

('C', '2023-01-01', '3'),

('C', '2023-01-01', '3'),

('C', '2023-01-07', '3');

Select \*From members

Select \*From menu

Select \*From Sales

ANSWER THE FOLLOWING QUERIES

1. What is the total amount each customer spent at the restaurant?

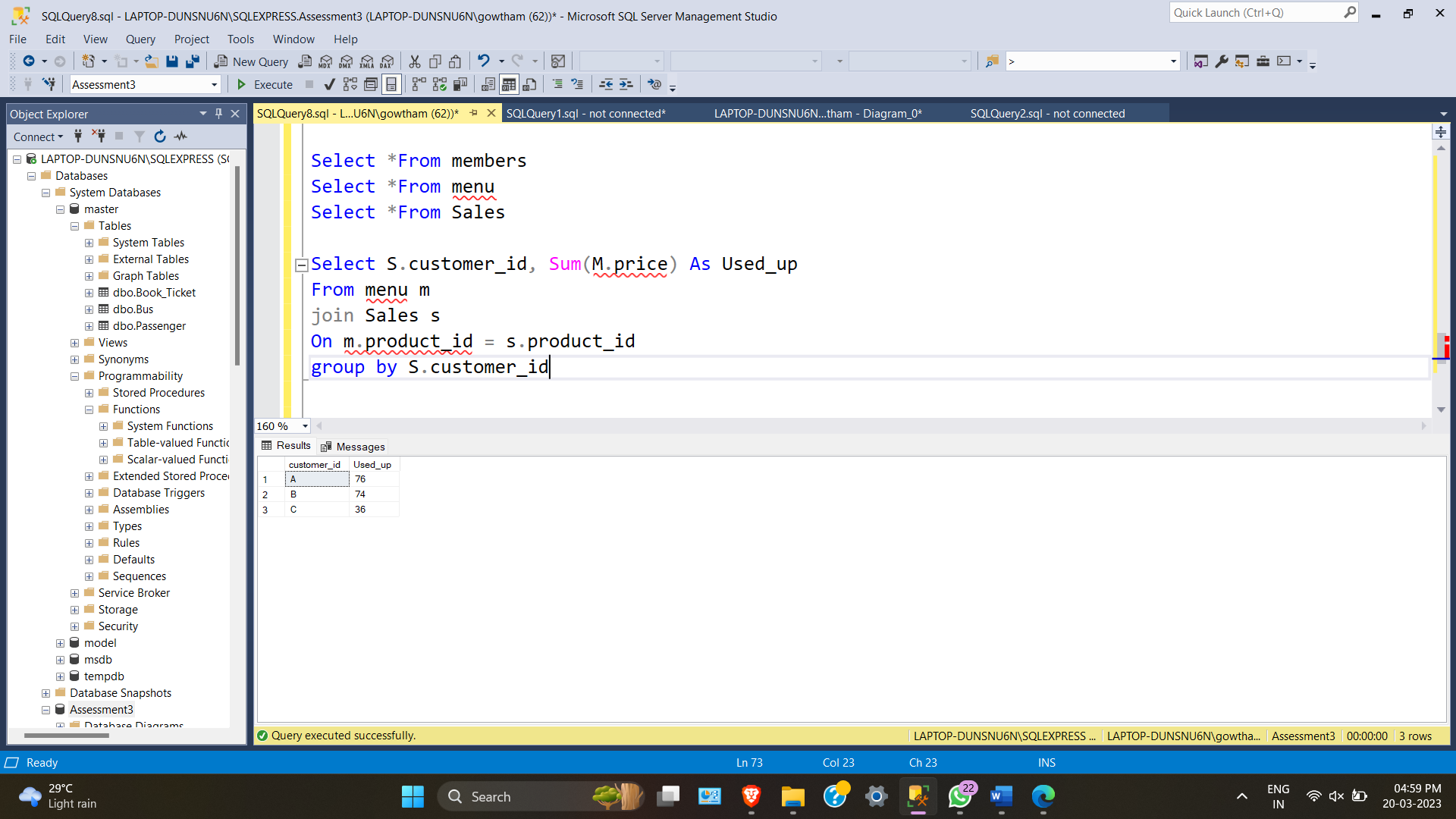
Select S.customer\_id, Sum(M.price) As Used\_up

From menu m

join Sales s

On m.product\_id = s.product\_id

group by S.customer\_id

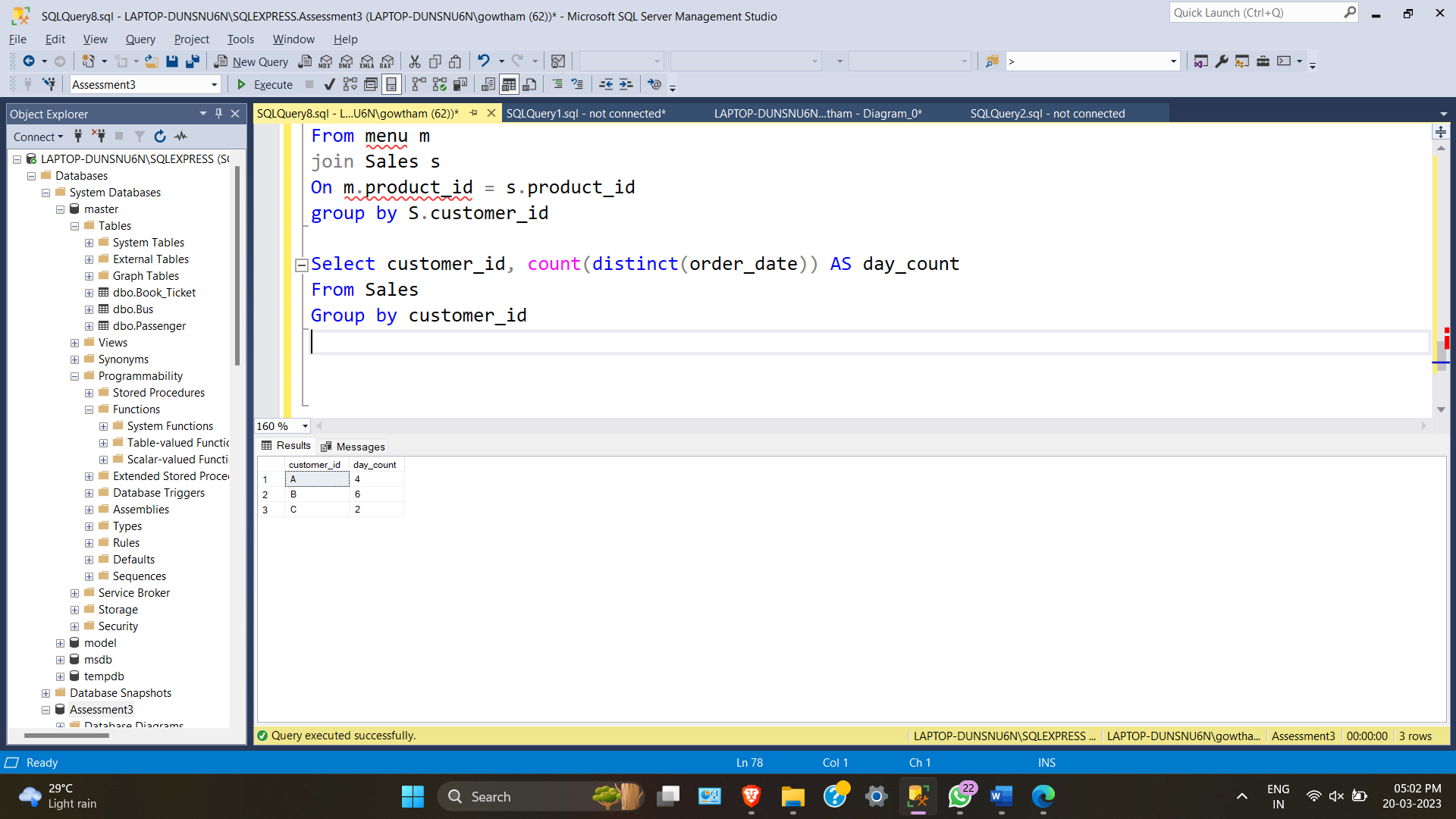


1. How many days has each customer visited the restaurant?

Select customer\_id, count(distinct(order\_date)) AS day\_count

From Sales

Group by customer\_id

Which item was the most popular for each customer?

1. Which item was purchased first by the customer after they became a member?
2. Which item was purchased just before the customer became a member?
3. What is the total items and amount spent for each member before they became a member?
4. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
5. In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi - how many points do customer A and B have at the end of January?
6. What was the first item from the menu purchased by each customer?

WITH ordered\_sales\_cte AS

(

SELECT

customer\_id,

order\_date,

product\_name,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rank

FROM dbo.sales AS s

JOIN dbo.menu AS m

ON s.product\_id = m.product\_id

)

SELECT

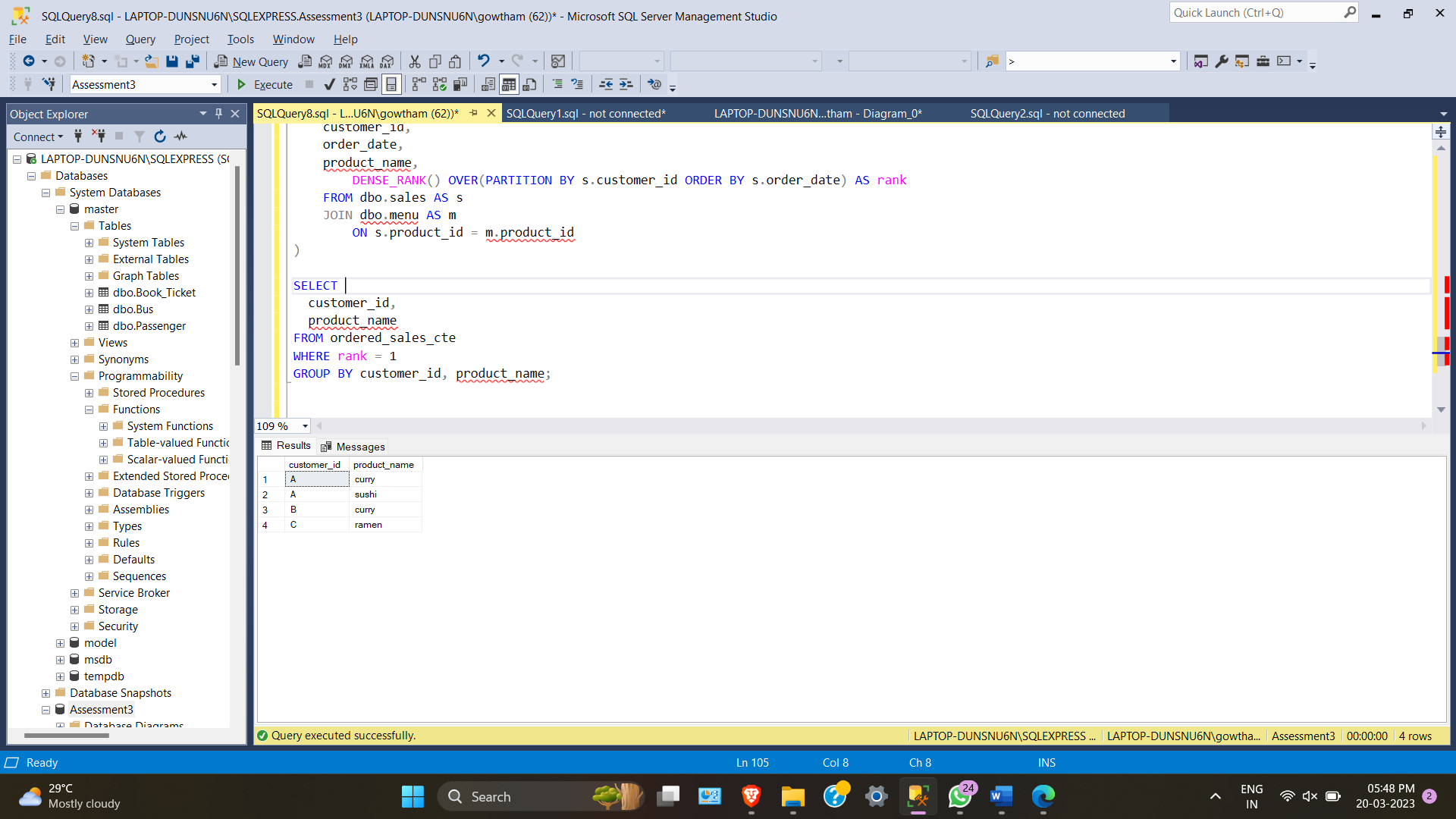
customer\_id,

product\_name

FROM ordered\_sales\_cte

WHERE rank = 1

GROUP BY customer\_id, product\_name;



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

SELECT

TOP 1 (COUNT(s.product\_id)) AS most\_purchased,

product\_name

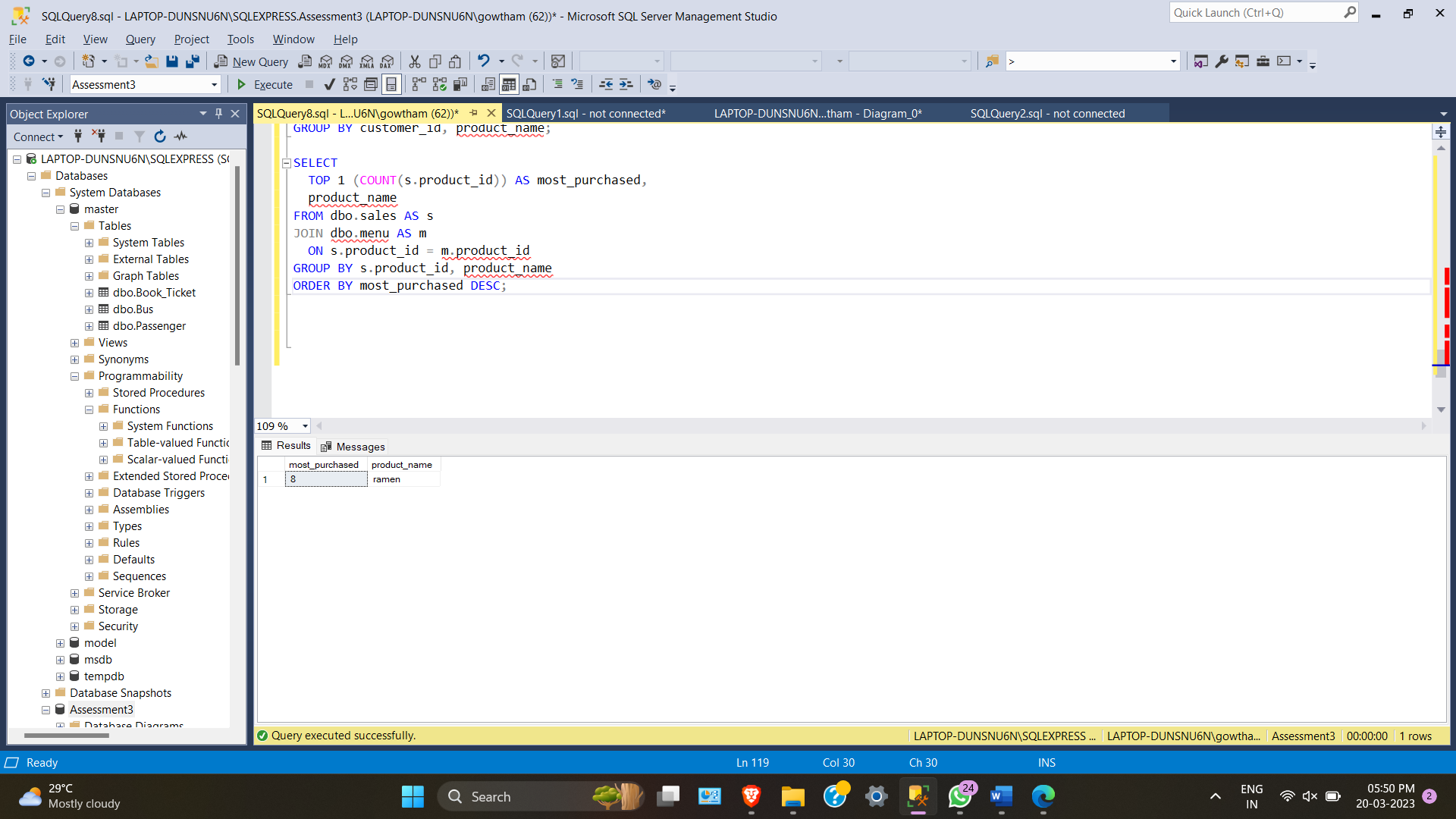
FROM dbo.sales AS s

JOIN dbo.menu AS m

ON s.product\_id = m.product\_id

GROUP BY s.product\_id, product\_name

ORDER BY most\_purchased DESC;



5)Which item was the most popular for each customer?

WITH fav\_item\_cte AS

(

SELECT

s.customer\_id,

m.product\_name,

COUNT(m.product\_id) AS order\_count,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY COUNT(s.customer\_id) DESC) AS rank

FROM dbo.menu AS m

JOIN dbo.sales AS s

ON m.product\_id = s.product\_id

GROUP BY s.customer\_id, m.product\_name

)

SELECT

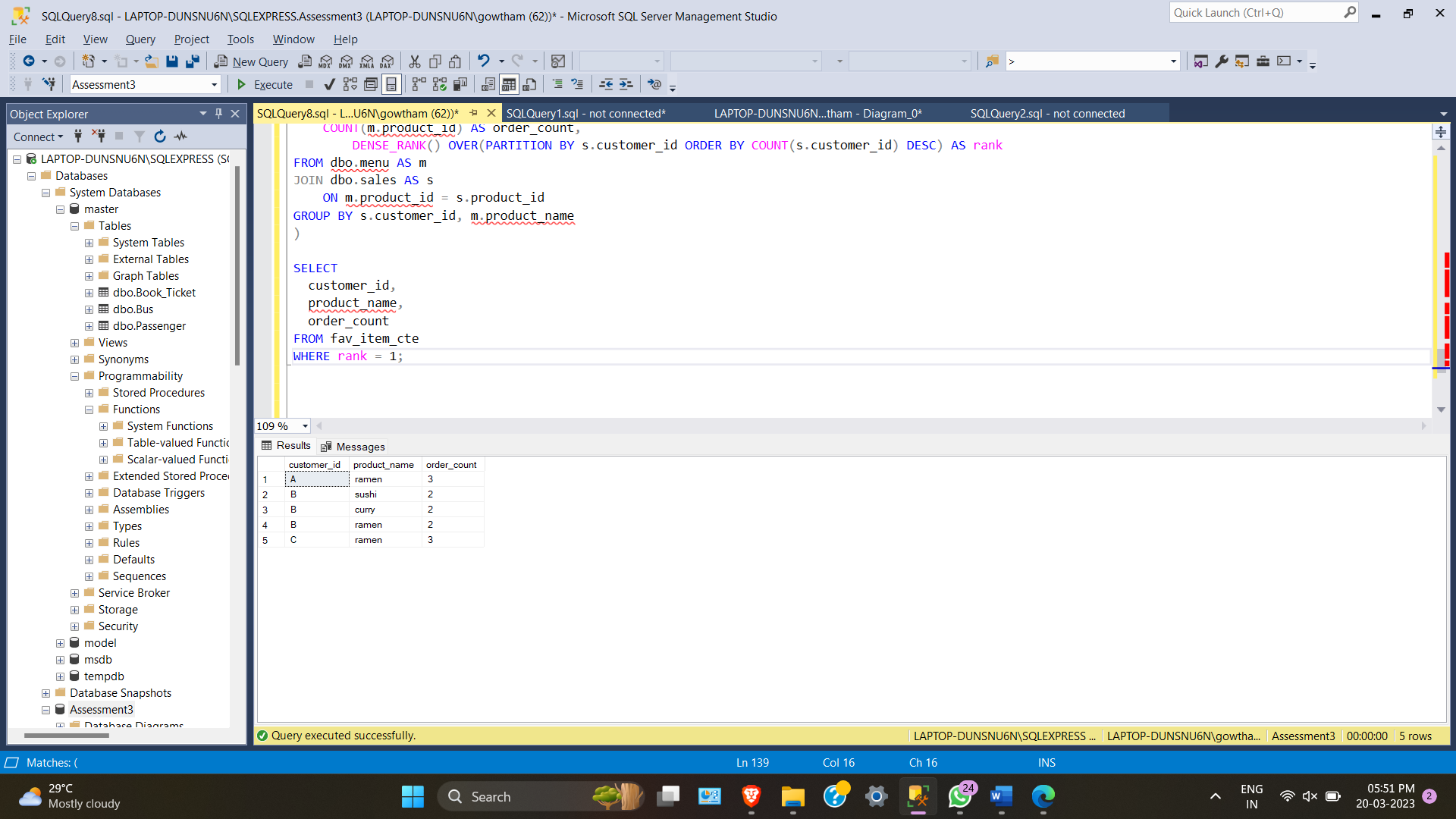
customer\_id,

product\_name,

order\_count

FROM fav\_item\_cte

WHERE rank = 1;



6)Which item was purchased first by the customer after they became a member?

WITH member\_sales\_cte AS

(

SELECT

s.customer\_id,

m.join\_date,

s.order\_date,

s.product\_id,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rank

FROM sales AS s

JOIN members AS m

ON s.customer\_id = m.customer\_id

WHERE s.order\_date >= m.join\_date)

SELECT

s.customer\_id,

s.order\_date,

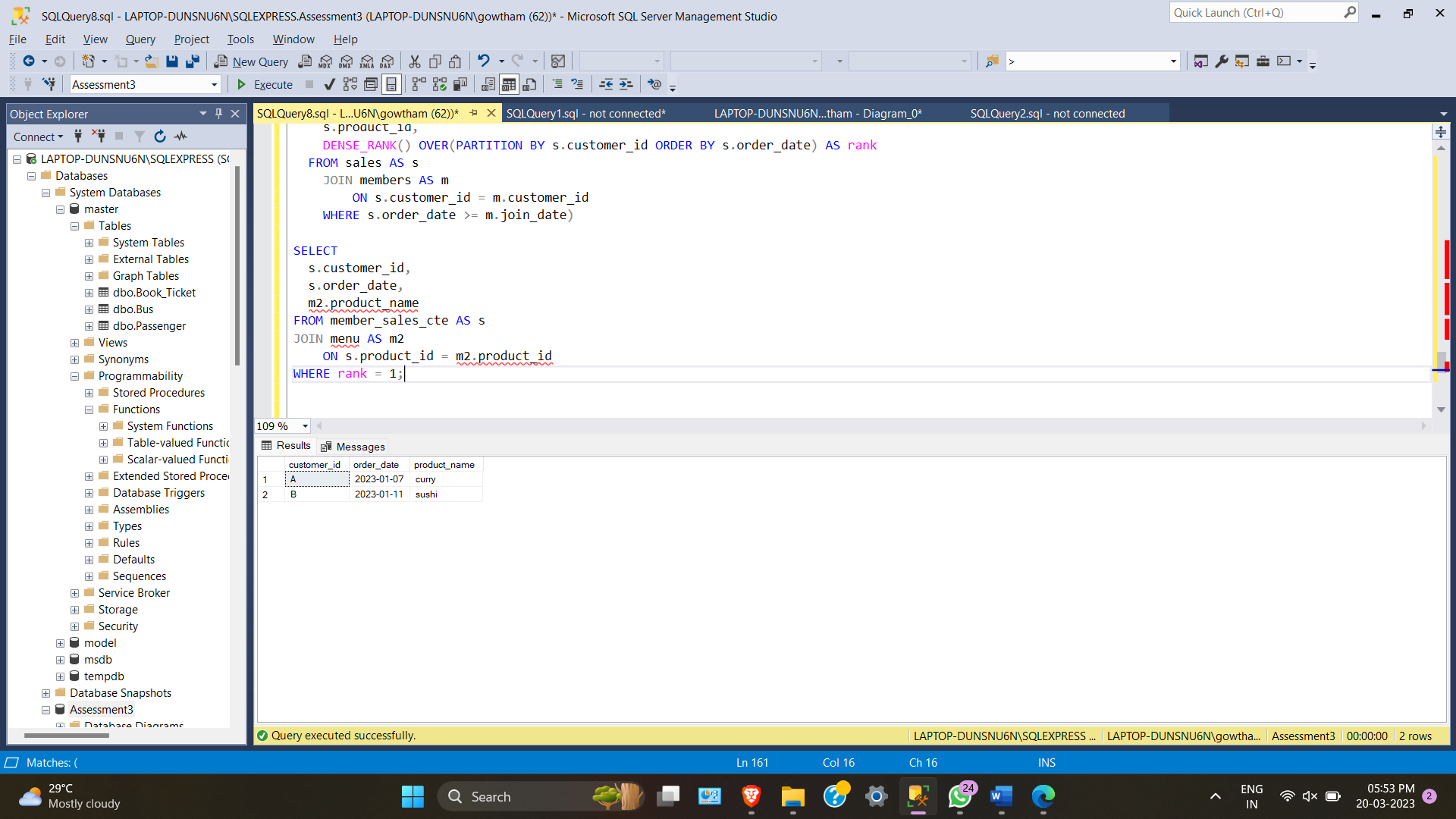
m2.product\_name

FROM member\_sales\_cte AS s

JOIN menu AS m2

ON s.product\_id = m2.product\_id

WHERE rank = 1;



7)Which item was purchased just before the customer became a member?

WITH prior\_member\_purchased\_cte AS

(

SELECT

s.customer\_id,

m.join\_date,

s.order\_date,

s.product\_id,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date DESC) AS rank

FROM sales AS s

JOIN members AS m

ON s.customer\_id = m.customer\_id

WHERE s.order\_date < m.join\_date

)

SELECT

s.customer\_id,

s.order\_date,

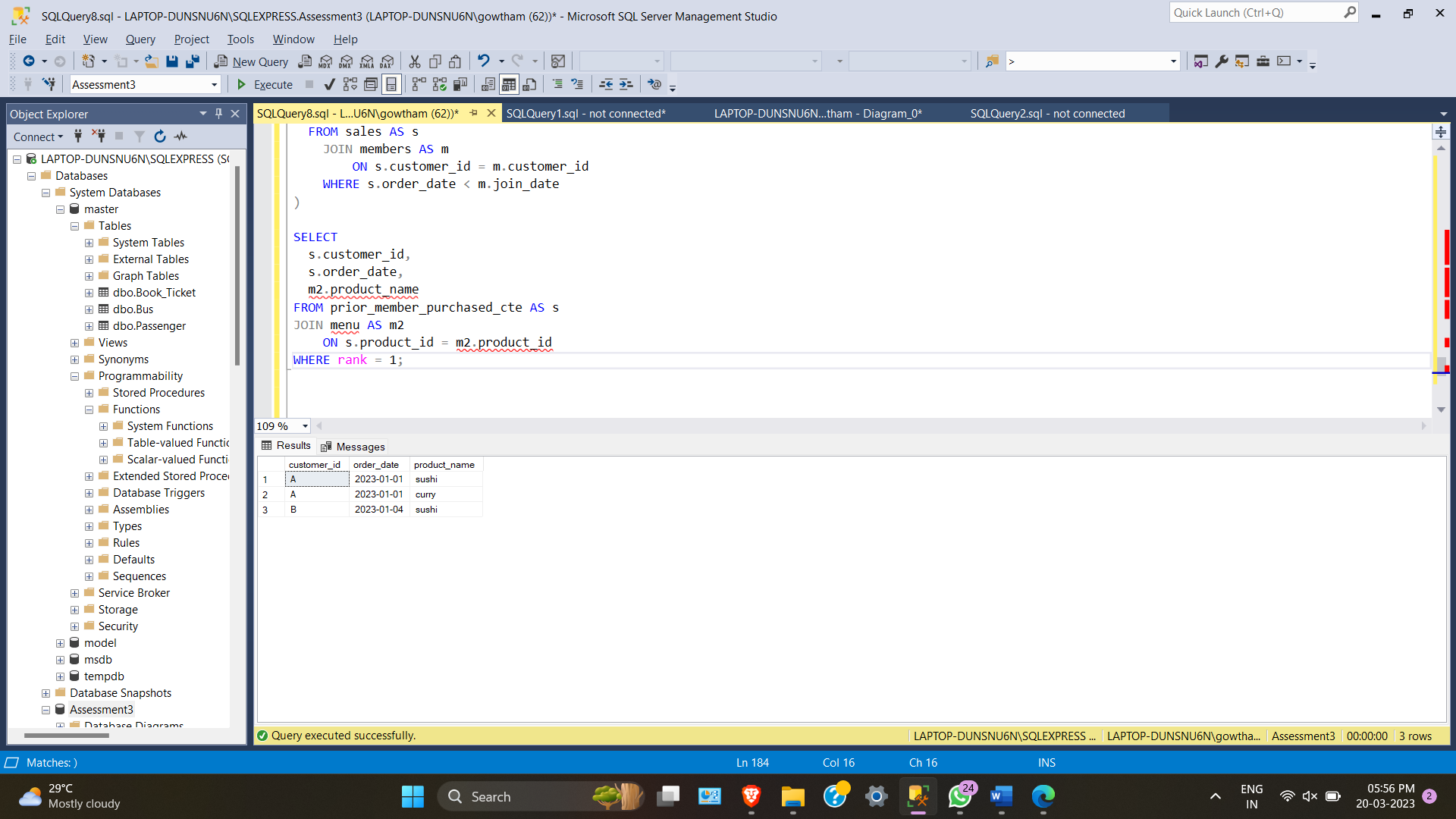
m2.product\_name

FROM prior\_member\_purchased\_cte AS s

JOIN menu AS m2

ON s.product\_id = m2.product\_id

WHERE rank = 1;



8)What is the total items and amount spent for each member before they became a member?

SELECT

s.customer\_id,

COUNT(DISTINCT s.product\_id) AS unique\_menu\_item,

SUM(mm.price) AS total\_sales

FROM sales AS s

JOIN members AS m

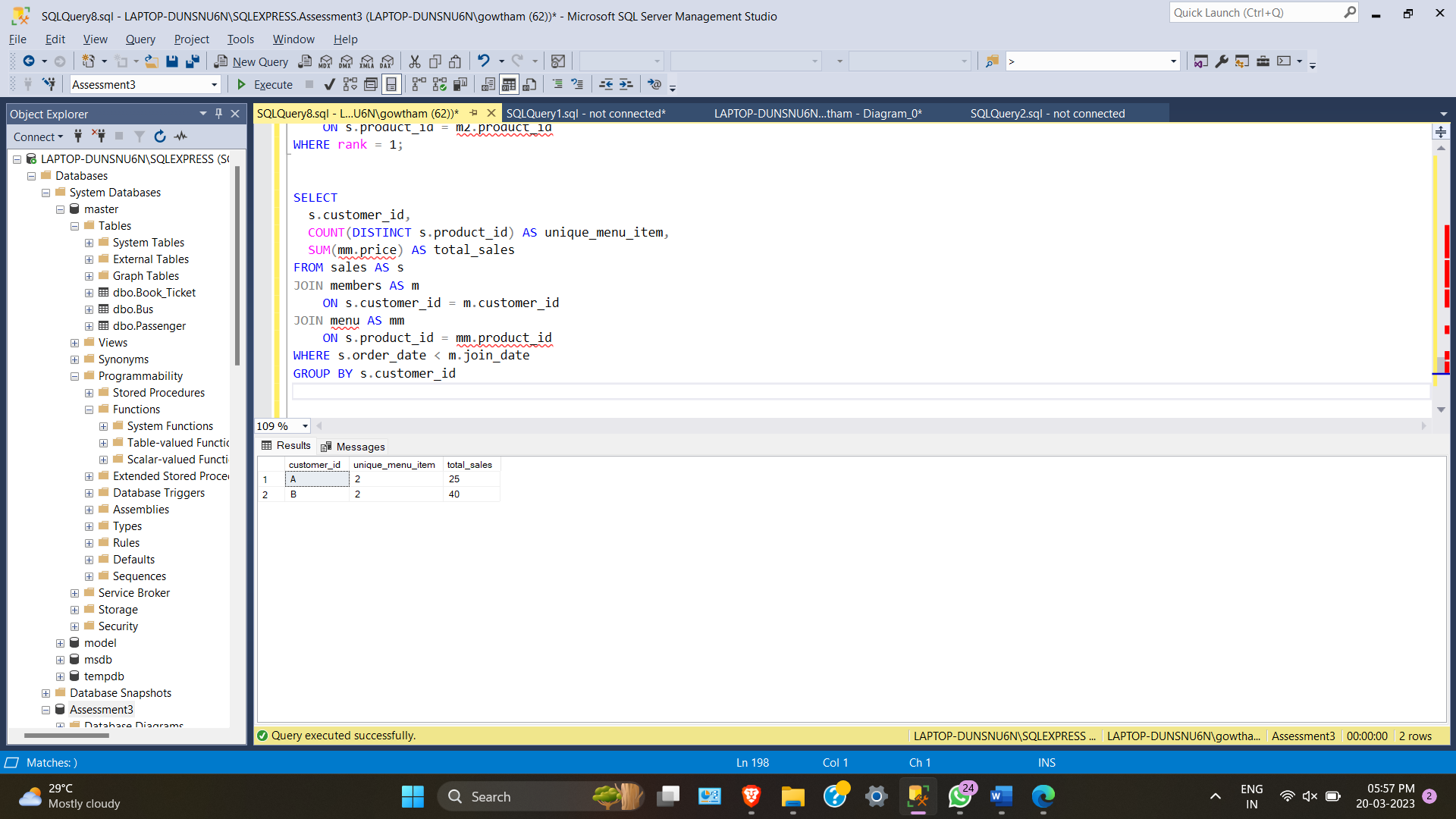
ON s.customer\_id = m.customer\_id

JOIN menu AS mm

ON s.product\_id = mm.product\_id

WHERE s.order\_date < m.join\_date

GROUP BY s.customer\_id



9)If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

WITH price\_points\_cte AS

(

SELECT \*,

CASE WHEN product\_name = 'sushi' THEN price \* 20

ELSE price \* 10 END AS points

FROM menu

)

SELECT

s.customer\_id,

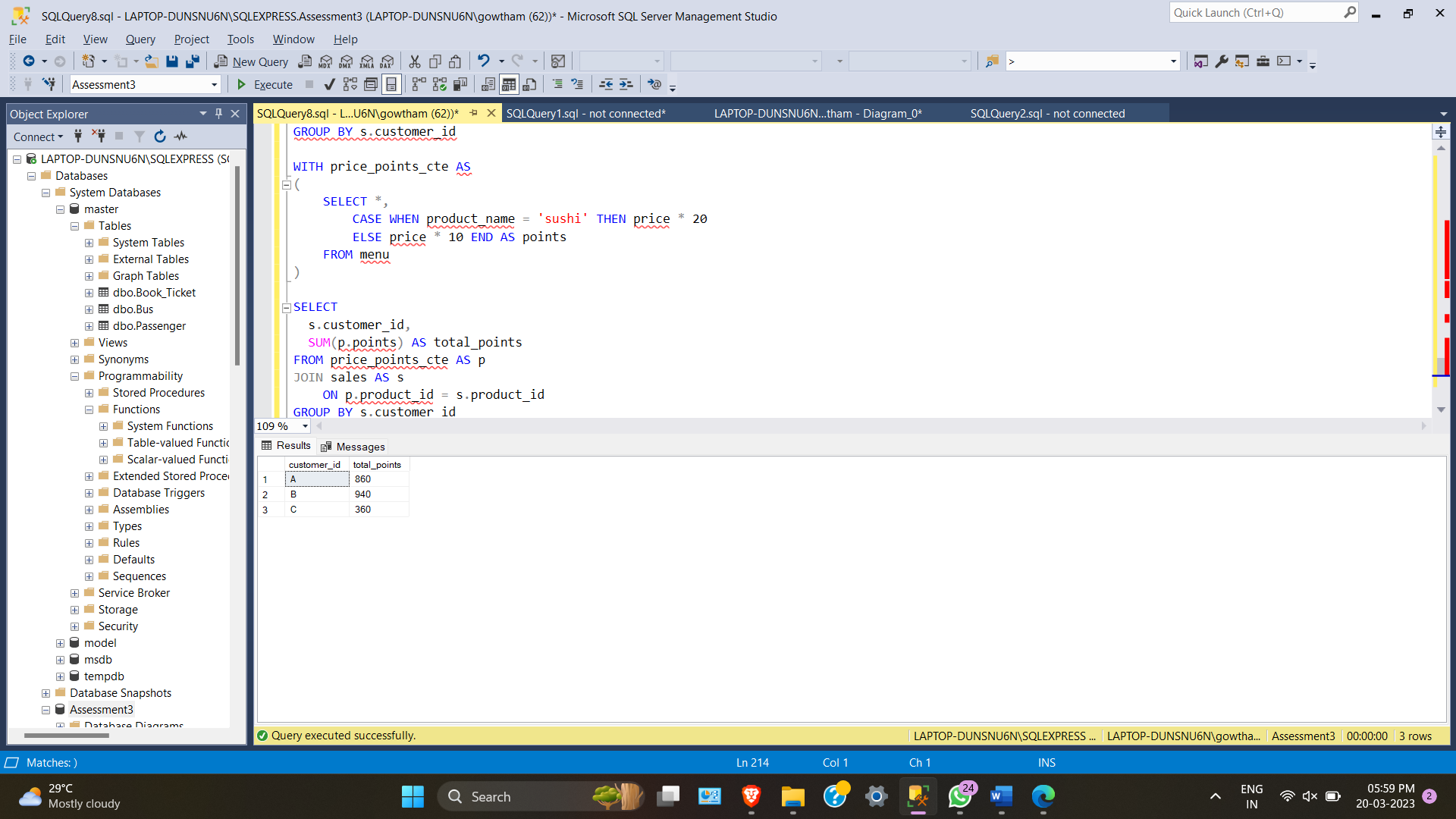
SUM(p.points) AS total\_points

FROM price\_points\_cte AS p

JOIN sales AS s

ON p.product\_id = s.product\_id

GROUP BY s.customer\_id



10)In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi - how many points do customer A and B have at the end of January?

-- 1. Find member validity date of each customer and get last date of January

-- 2. Use CASE WHEN to allocate points by date and product id

-- 3. SUM price and points

WITH dates\_cte AS

(

SELECT

\*,

DATEADD(DAY, 6, join\_date) AS valid\_date,

EOMONTH('2023/04/05') AS last\_date

FROM members AS m

)

SELECT

d.customer\_id, s.order\_date, d.join\_date, d.valid\_date, d.last\_date, m.product\_name,

m.price,

SUM(

CASE WHEN m.product\_name = 'sushi' THEN 2 \* 10 \* m.price

WHEN s.order\_date BETWEEN d.join\_date AND d.valid\_date THEN 2 \* 10 \* m.price

ELSE 10 \* m.price END) AS points

FROM dates\_cte AS d

JOIN sales AS s

ON d.customer\_id = s.customer\_id

JOIN menu AS m

ON s.product\_id = m.product\_id

WHERE s.order\_date < d.last\_date

GROUP BY d.customer\_id, s.order\_date, d.join\_date, d.valid\_date, d.last\_date, m.product\_name, m.price

