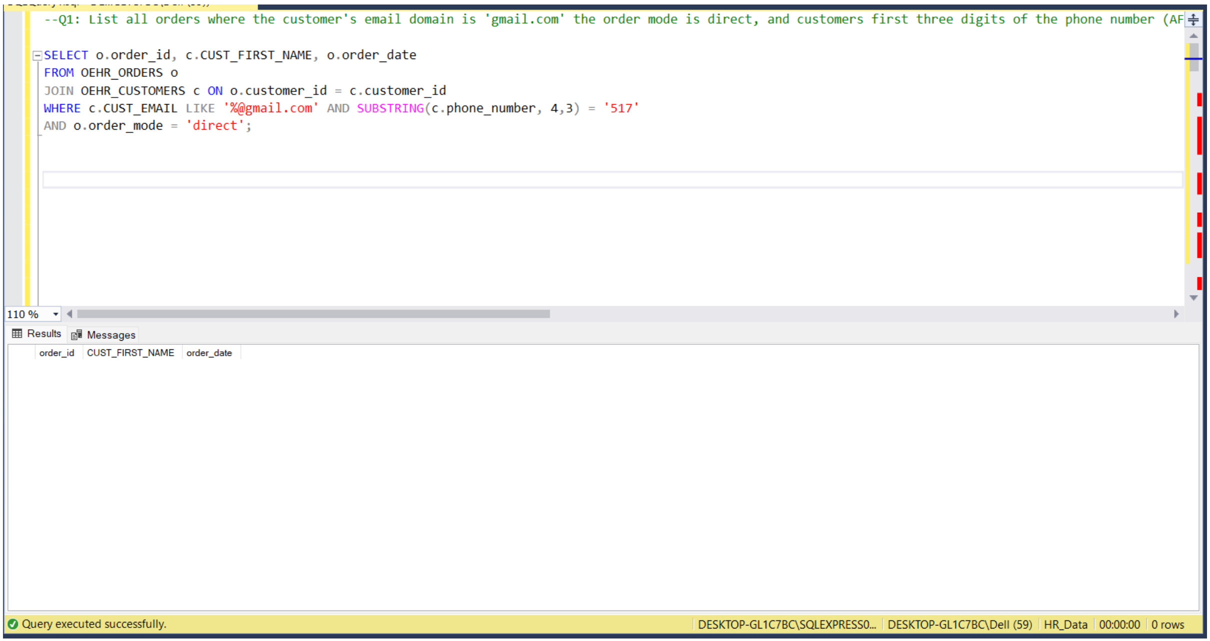
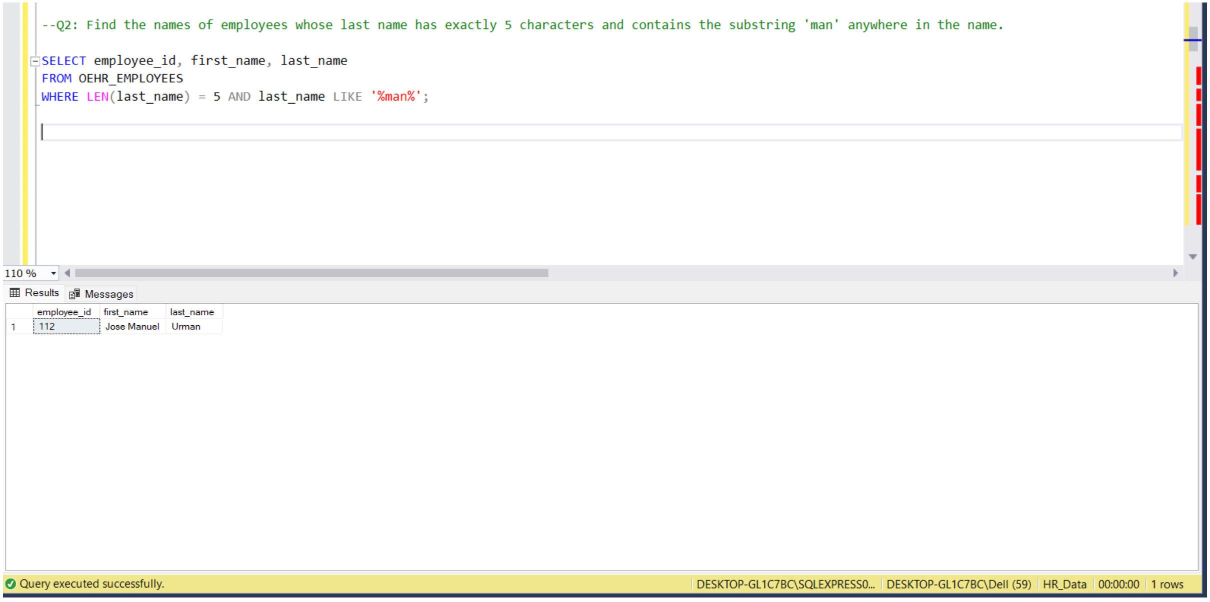


```
--Q1: List all orders where the customer's email domain is 'gmail.com' the order mode is direct, and customers first three digits of the phone number (AF

SELECT o.order_id, c.CUST_FIRST_NAME, o.order_date
FROM OEHR_ORDERS o
JOIN OEHR_CUSTOMERS c ON o.customer_id = c.customer_id
WHERE c.CUST_EMAIL LIKE '%@gmail.com' AND SUBSTRING(c.phone_number, 4,3) = '517'
AND o.order_mode = 'direct';
```

order_id	CUST_FIRST_NAME	order_date

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 0 rows

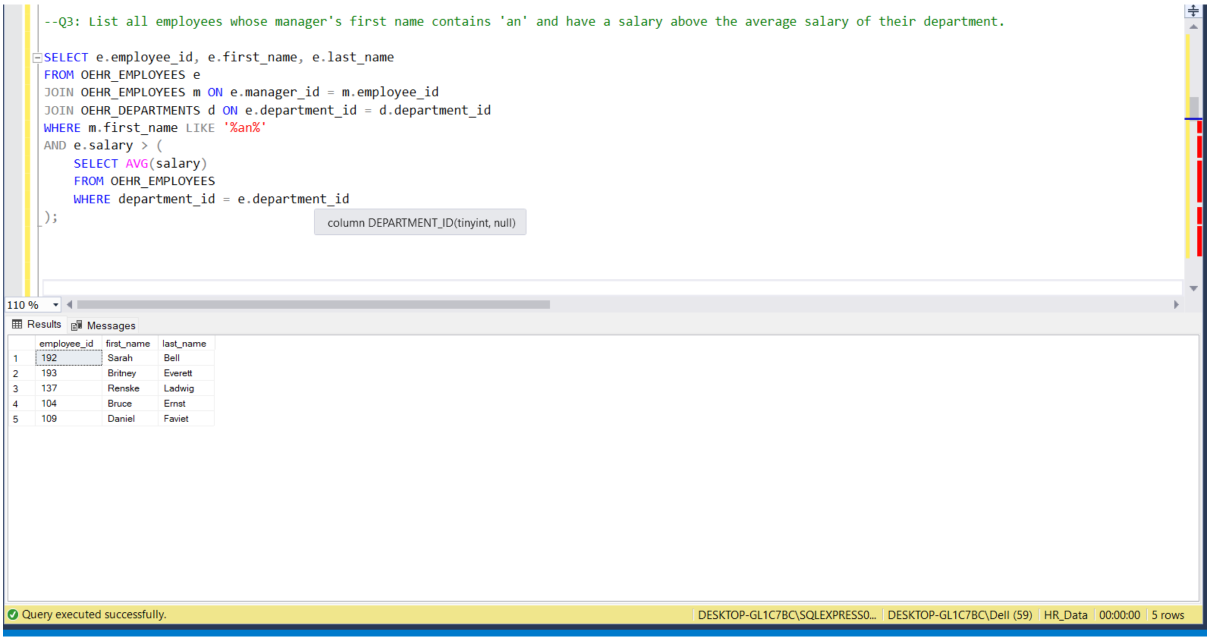


```
--Q2: Find the names of employees whose last name has exactly 5 characters and contains the substring 'man' anywhere in the name.

SELECT employee_id, first_name, last_name
FROM OEHR_EMPLOYEES
WHERE LEN(last_name) = 5 AND last_name LIKE '%man%';
```

	employee_id	first_name	last_name
1	112	Jose Manuel	Urman

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 1 rows

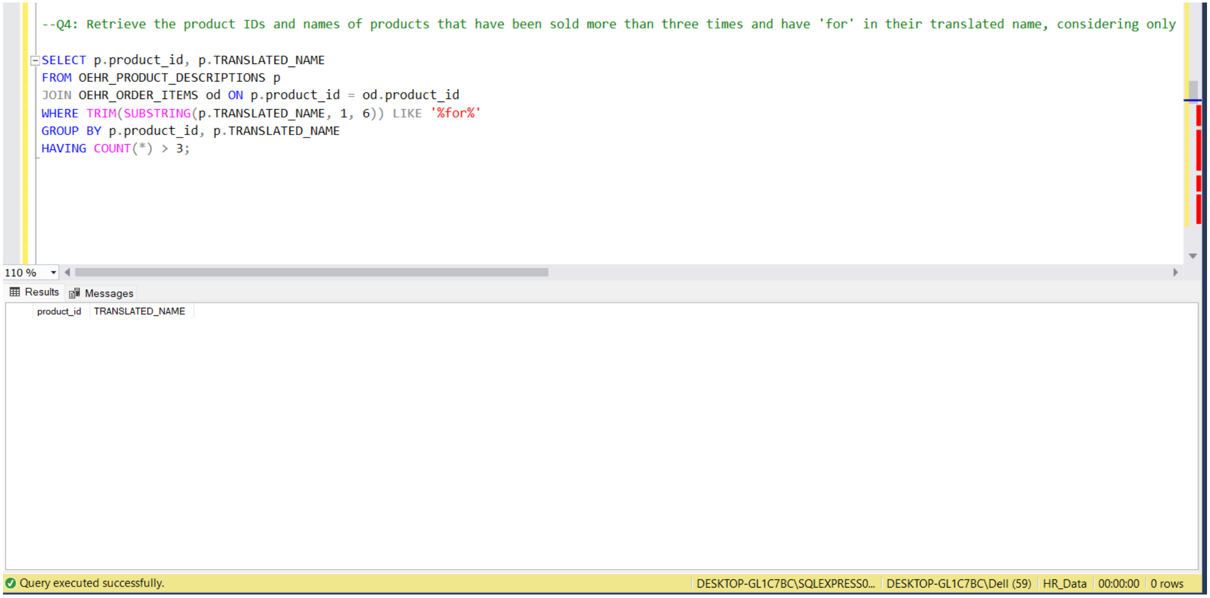


```sql
--Q3: List all employees whose manager's first name contains 'an' and have a salary above the average salary of their department.

SELECT e.employee_id, e.first_name, e.last_name
FROM OEHR_EMPLOYEES e
JOIN OEHR_EMPLOYEES m ON e.manager_id = m.employee_id
JOIN OEHR_DEPARTMENTS d ON e.department_id = d.department_id
WHERE m.first_name LIKE '%an%'
AND e.salary > (
    SELECT AVG(salary)
    FROM OEHR_EMPLOYEES
    WHERE department_id = e.department_id
);
```

	employee_id	first_name	last_name
1	192	Sarah	Bell
2	193	Britney	Everett
3	137	Renske	Ladwig
4	104	Bruce	Ernst
5	109	Daniel	Faviet

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 5 rows

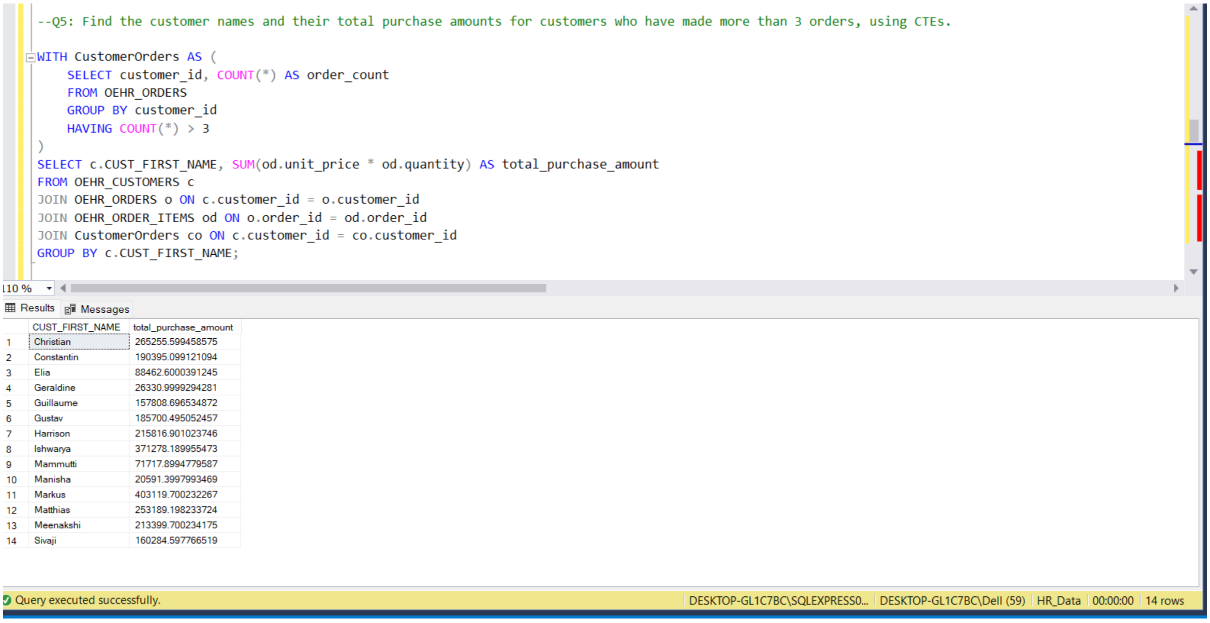


```sql
--Q4: Retrieve the product IDs and names of products that have been sold more than three times and have 'for' in their translated name, considering only

SELECT p.product_id, p.TRANSLATED_NAME
FROM OEHR_PRODUCT_DESCRIPTIONS p
JOIN OEHR_ORDER_ITEMS od ON p.product_id = od.product_id
WHERE TRIM(SUBSTRING(p.TRANSLATED_NAME, 1, 6)) LIKE '%for%'
GROUP BY p.product_id, p.TRANSLATED_NAME
HAVING COUNT(*) > 3;
```

	product_id	TRANSLATED_NAME

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 0 rows

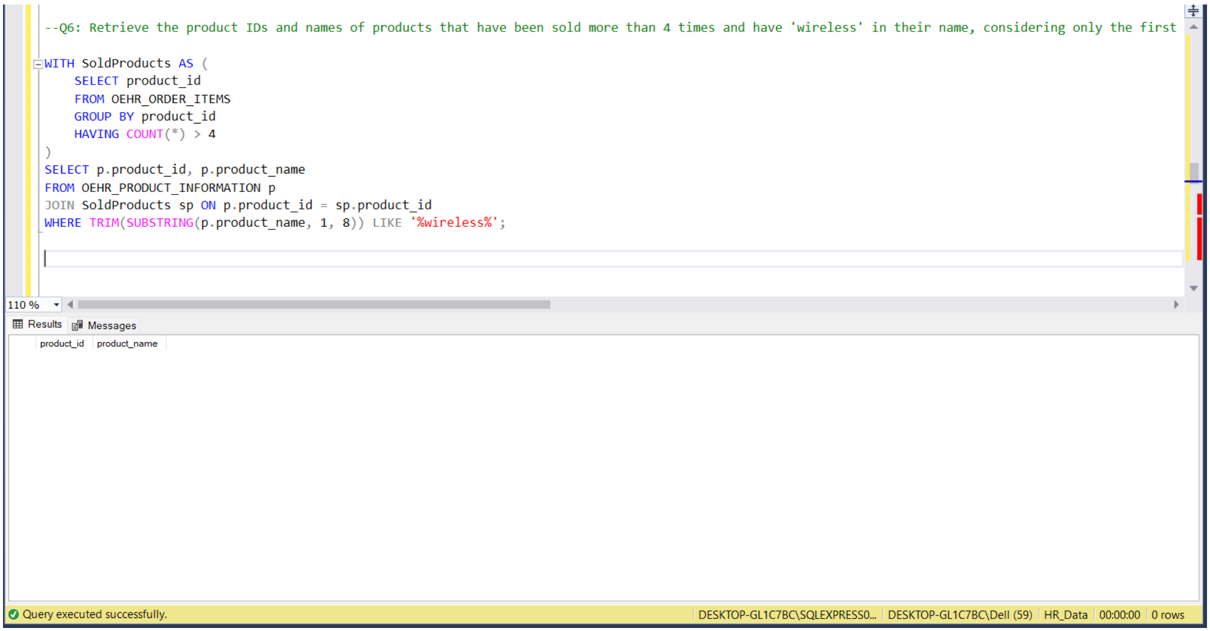


```sql
--Q5: Find the customer names and their total purchase amounts for customers who have made more than 3 orders, using CTEs.

WITH CustomerOrders AS (
    SELECT customer_id, COUNT(*) AS order_count
    FROM OEHR_ORDERS
    GROUP BY customer_id
    HAVING COUNT(*) > 3
)
SELECT c.CUST_FIRST_NAME, SUM(od.unit_price * od.quantity) AS total_purchase_amount
FROM OEHR_CUSTOMERS c
JOIN OEHR_ORDERS o ON c.customer_id = o.customer_id
JOIN OEHR_ORDER_ITEMS od ON o.order_id = od.order_id
JOIN CustomerOrders co ON c.customer_id = co.customer_id
GROUP BY c.CUST_FIRST_NAME;
```

	CUST_FIRST_NAME	total_purchase_amount
1	Christian	265255.599458575
2	Constantin	190395.099121094
3	Elia	88462.6000391245
4	Geraldine	26330.9999294281
5	Guillaume	157808.696534872
6	Gustav	185700.495052457
7	Harrison	215816.901023746
8	Ishwarya	371278.189955473
9	Mammutti	71717.8994779587
10	Manisha	20591.3997993469
11	Markus	403119.700232267
12	Matthias	253189.198233724
13	Meenakshi	213399.700234175
14	Sivaji	160284.597766519

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 14 rows

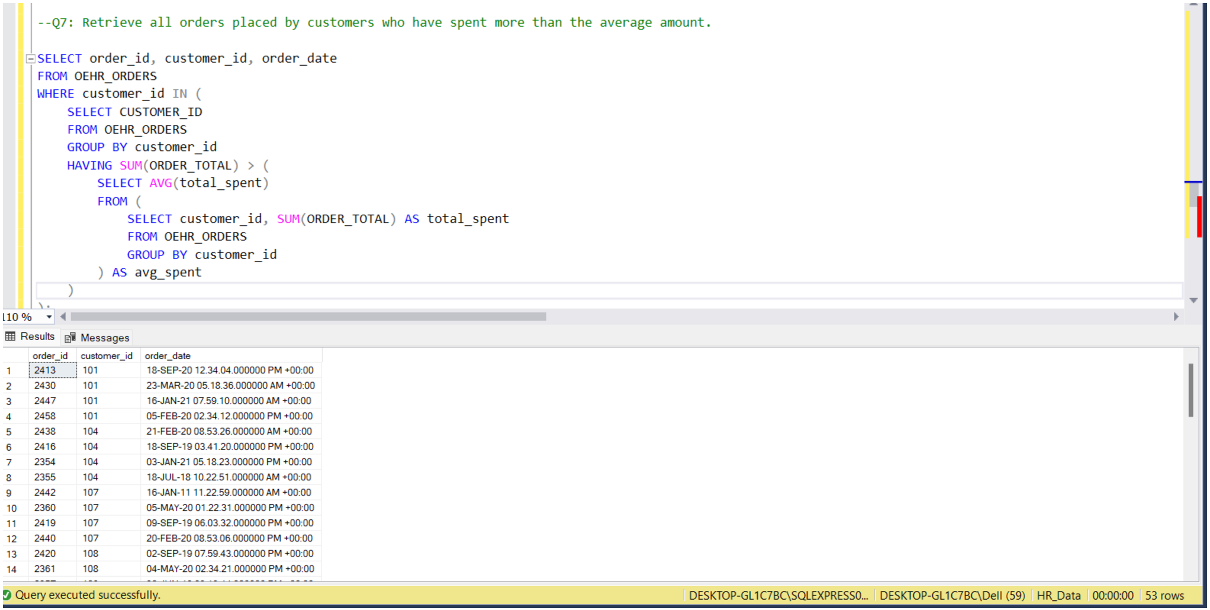


```
--Q6: Retrieve the product IDs and names of products that have been sold more than 4 times and have 'wireless' in their name, considering only the first

WITH SoldProducts AS (
    SELECT product_id
    FROM OEHR_ORDER_ITEMS
    GROUP BY product_id
    HAVING COUNT(*) > 4
)
SELECT p.product_id, p.product_name
FROM OEHR_PRODUCT_INFORMATION p
JOIN SoldProducts sp ON p.product_id = sp.product_id
WHERE TRIM(SUBSTRING(p.product_name, 1, 8)) LIKE '%wireless%';
```

product_id	product_name

```
Query executed successfully.    DESKTOP-GL1C7BC\SQLEXPRESS0...  DESKTOP-GL1C7BC\Dell (59)  HR_Data  00:00:00  0 rows
```

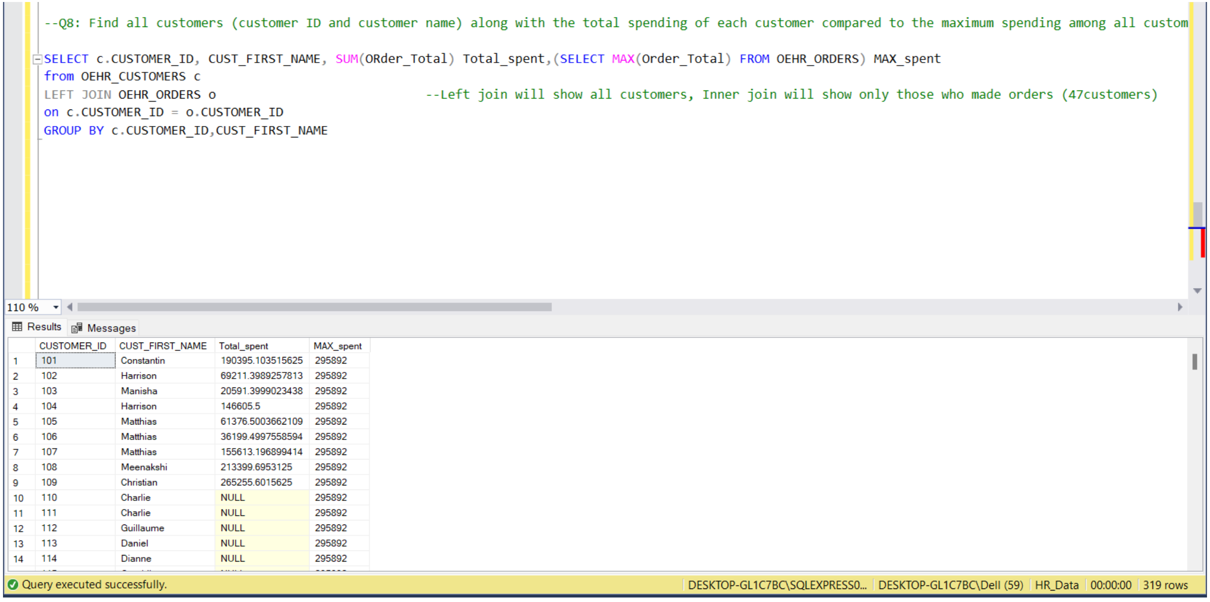


```
--Q7: Retrieve all orders placed by customers who have spent more than the average amount.

SELECT order_id, customer_id, order_date
FROM OEHR_ORDERS
WHERE customer_id IN (
    SELECT CUSTOMER_ID
    FROM OEHR_ORDERS
    GROUP BY customer_id
    HAVING SUM(ORDER_TOTAL) > (
        SELECT AVG(total_spent)
        FROM (
            SELECT customer_id, SUM(ORDER_TOTAL) AS total_spent
            FROM OEHR_ORDERS
            GROUP BY customer_id
        ) AS avg_spent
    )
```

	order_id	customer_id	order_date
1	2413	101	18-SEP-20 12.34.04.000000 PM +00:00
2	2430	101	23-MAR-20 05.18.36.000000 AM +00:00
3	2447	101	16-JAN-21 07.59.10.000000 AM +00:00
4	2458	101	05-FEB-20 02.34.12.000000 PM +00:00
5	2438	104	21-FEB-20 08.53.26.000000 AM +00:00
6	2416	104	18-SEP-19 03.41.20.000000 PM +00:00
7	2354	104	03-JAN-21 05.18.23.000000 PM +00:00
8	2355	104	18-JUL-18 10.22.51.000000 AM +00:00
9	2442	107	16-JAN-11 11.22.59.000000 AM +00:00
10	2360	107	05-MAY-20 01.22.31.000000 PM +00:00
11	2419	107	09-SEP-19 06.03.32.000000 PM +00:00
12	2440	107	20-FEB-20 08.53.06.000000 PM +00:00
13	2420	108	02-SEP-19 07.59.43.000000 PM +00:00
14	2361	108	04-MAY-20 02.34.21.000000 PM +00:00

```
Query executed successfully.    DESKTOP-GL1C7BC\SQLEXPRESS0...  DESKTOP-GL1C7BC\Dell (59)  HR_Data  00:00:00  53 rows
```

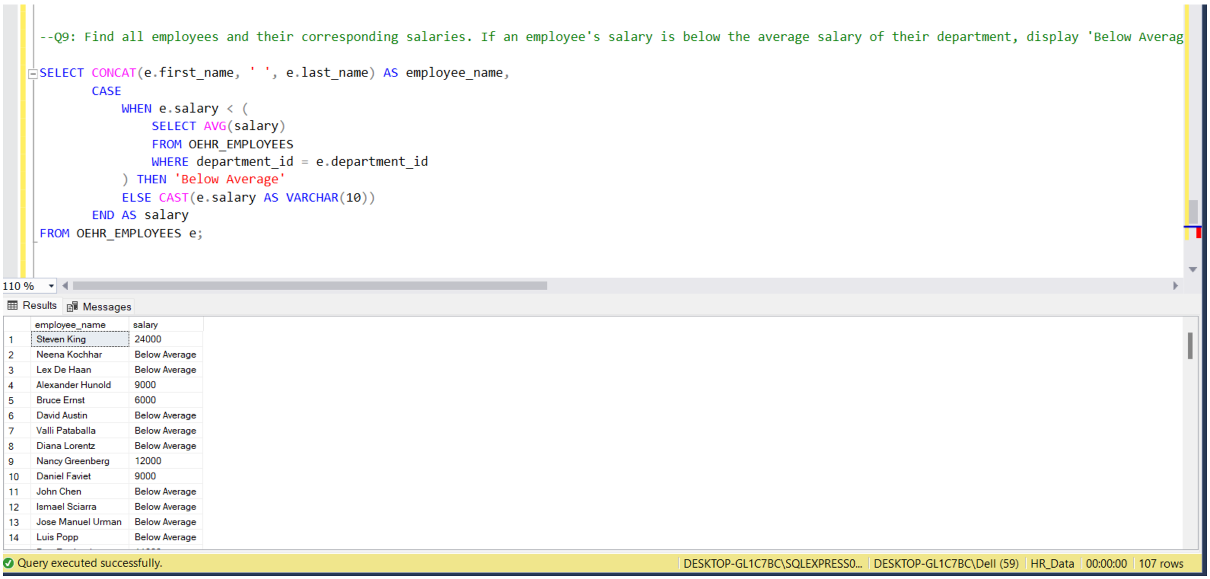


```
--Q8: Find all customers (customer ID and customer name) along with the total spending of each customer compared to the maximum spending among all custom

SELECT c.CUSTOMER_ID, CUST_FIRST_NAME, SUM(ORder_Total) Total_spent,(SELECT MAX(Order_Total) FROM OEHR_ORDERS) MAX_spent
from OEHR_CUSTOMERS c
LEFT JOIN OEHR_ORDERS o                    --Left join will show all customers, Inner join will show only those who made orders (47customers)
on c.CUSTOMER_ID = o.CUSTOMER_ID
GROUP BY c.CUSTOMER_ID,CUST_FIRST_NAME
```

	CUSTOMER_ID	CUST_FIRST_NAME	Total_spent	MAX_spent
1	101	Constantin	190395.103515625	295892
2	102	Harrison	69211.3989257813	295892
3	103	Manisha	20591.3999023438	295892
4	104	Harrison	146605.5	295892
5	105	Matthias	61376.5003662109	295892
6	106	Matthias	36199.4997558594	295892
7	107	Matthias	155613.196899414	295892
8	108	Meenakshi	213399.6953125	295892
9	109	Christian	265255.6015625	295892
10	110	Charlie	NULL	295892
11	111	Charlie	NULL	295892
12	112	Guillaume	NULL	295892
13	113	Daniel	NULL	295892
14	114	Dianne	NULL	295892

```
Query executed successfully.    DESKTOP-GL1C7BC\SQLEXPRESS0...  DESKTOP-GL1C7BC\Dell (59)  HR_Data  00:00:00  319 rows
```

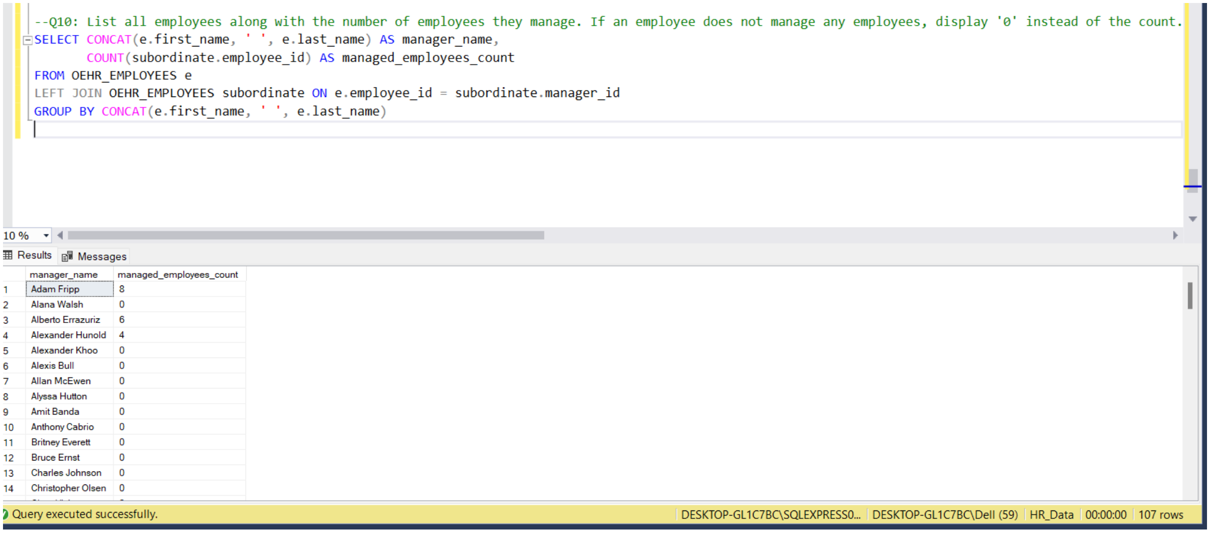


```sql
--Q9: Find all employees and their corresponding salaries. If an employee's salary is below the average salary of their department, display 'Below Averag

SELECT CONCAT(e.first_name, ' ', e.last_name) AS employee_name,
       CASE
           WHEN e.salary < (
               SELECT AVG(salary)
               FROM OEHR_EMPLOYEES
               WHERE department_id = e.department_id
           ) THEN 'Below Average'
           ELSE CAST(e.salary AS VARCHAR(10))
       END AS salary
FROM OEHR_EMPLOYEES e;
```

	employee_name	salary
1	Steven King	24000
2	Neena Kochhar	Below Average
3	Lex De Haan	Below Average
4	Alexander Hunold	9000
5	Bruce Ernst	6000
6	David Austin	Below Average
7	Valli Pataballa	Below Average
8	Diana Lorentz	Below Average
9	Nancy Greenberg	12000
10	Daniel Faviet	9000
11	John Chen	Below Average
12	Ismael Sciarra	Below Average
13	Jose Manuel Urman	Below Average
14	Luis Popp	Below Average

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 107 rows



```sql
--Q10: List all employees along with the number of employees they manage. If an employee does not manage any employees, display '0' instead of the count.
SELECT CONCAT(e.first_name, ' ', e.last_name) AS manager_name,
       COUNT(subordinate.employee_id) AS managed_employees_count
FROM OEHR_EMPLOYEES e
LEFT JOIN OEHR_EMPLOYEES subordinate ON e.employee_id = subordinate.manager_id
GROUP BY CONCAT(e.first_name, ' ', e.last_name)
```

	manager_name	managed_employees_count
1	Adam Fripp	8
2	Alana Walsh	0
3	Alberto Errazuriz	6
4	Alexander Hunold	4
5	Alexander Khoo	0
6	Alexis Bull	0
7	Allan McEwen	0
8	Alyssa Hutton	0
9	Amit Banda	0
10	Anthony Cabrio	0
11	Britney Everett	0
12	Bruce Ernst	0
13	Charles Johnson	0
14	Christopher Olsen	0

Query executed successfully. | DESKTOP-GL1C7BC\SQLEXPRESS0... | DESKTOP-GL1C7BC\Dell (59) | HR_Data | 00:00:00 | 107 rows