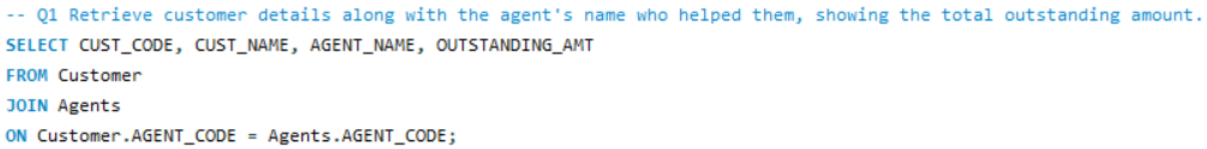
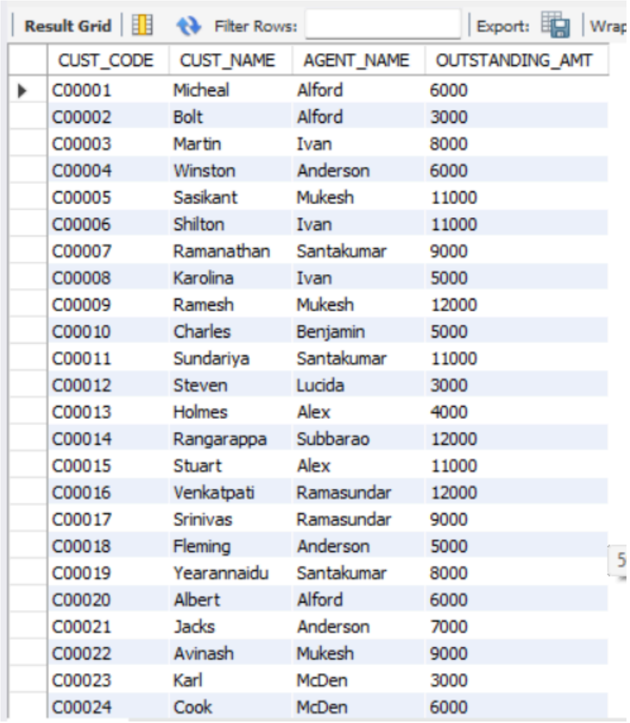
1)

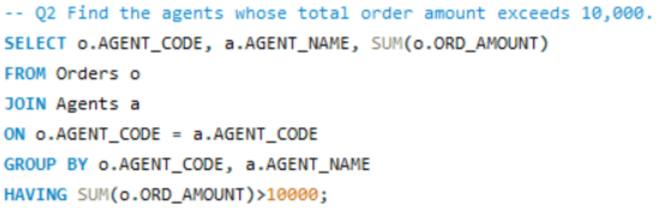


```
-- Q1 Retrieve customer details along with the agent's name who helped them, showing the total outstanding amount.
SELECT CUST_CODE, CUST_NAME, AGENT_NAME, OUTSTANDING_AMT
FROM Customer
JOIN Agents
ON Customer.AGENT_CODE = Agents.AGENT_CODE;
```

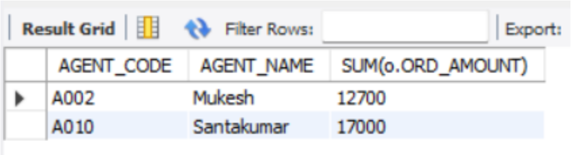


CUST_CODE	CUST_NAME	AGENT_NAME	OUTSTANDING_AMT
C00001	Micheal	Alford	6000
C00002	Bolt	Alford	3000
C00003	Martin	Ivan	8000
C00004	Winston	Anderson	6000
C00005	Sasikant	Mukesh	11000
C00006	Shilton	Ivan	11000
C00007	Ramanathan	Santakumar	9000
C00008	Karolina	Ivan	5000
C00009	Ramesh	Mukesh	12000
C00010	Charles	Benjamin	5000
C00011	Sundariya	Santakumar	11000
C00012	Steven	Lucida	3000
C00013	Holmes	Alex	4000
C00014	Rangarappa	Subbarao	12000
C00015	Stuart	Alex	11000
C00016	Venkatpati	Ramasundar	12000
C00017	Srinivas	Ramasundar	9000
C00018	Fleming	Anderson	5000
C00019	Yearannaidu	Santakumar	8000
C00020	Albert	Alford	6000
C00021	Jacks	Anderson	7000
C00022	Avinash	Mukesh	9000
C00023	Karl	McDen	3000
C00024	Cook	McDen	6000

2)

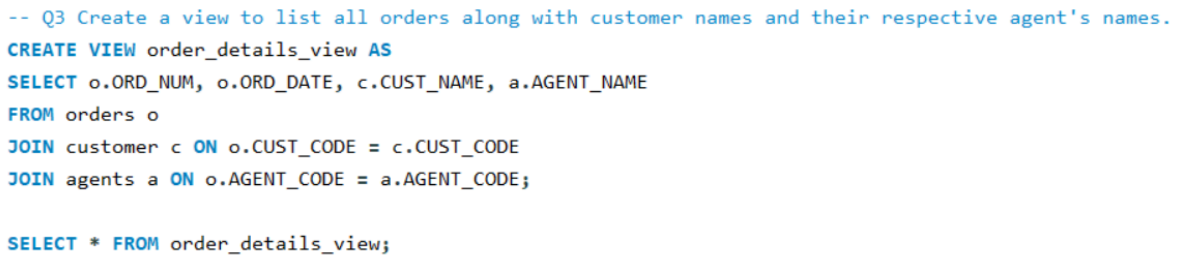


```
-- Q2 Find the agents whose total order amount exceeds 10,000.
SELECT o.AGENT_CODE, a.AGENT_NAME, SUM(o.ORD_AMOUNT)
FROM Orders o
JOIN Agents a
ON o.AGENT_CODE = a.AGENT_CODE
GROUP BY o.AGENT_CODE, a.AGENT_NAME
HAVING SUM(o.ORD_AMOUNT)>10000;
```



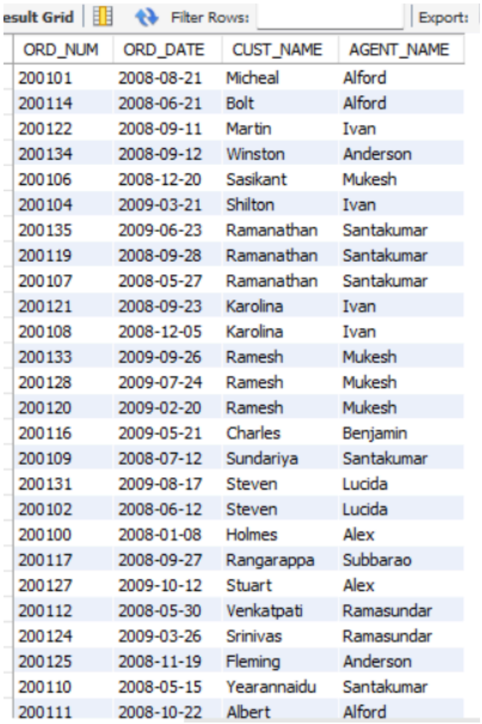
AGENT_CODE	AGENT_NAME	SUM(o.ORD_AMOUNT)
A002	Mukesh	12700
A010	Santakumar	17000

3)



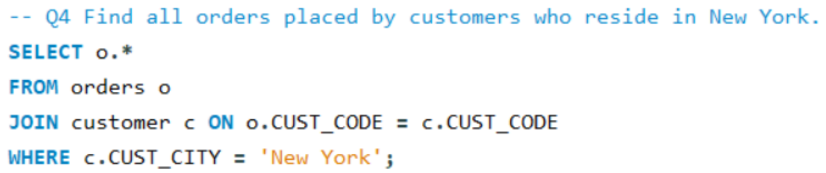
```sql
-- Q3 Create a view to list all orders along with customer names and their respective agent's names.
CREATE VIEW order_details_view AS
SELECT o.ORD_NUM, o.ORD_DATE, c.CUST_NAME, a.AGENT_NAME
FROM orders o
JOIN customer c ON o.CUST_CODE = c.CUST_CODE
JOIN agents a ON o.AGENT_CODE = a.AGENT_CODE;

SELECT * FROM order_details_view;
```

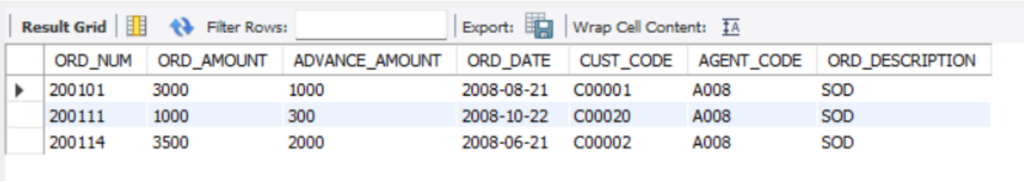


ORD_NUM	ORD_DATE	CUST_NAME	AGENT_NAME
200101	2008-08-21	Micheal	Alford
200114	2008-06-21	Bolt	Alford
200122	2008-09-11	Martin	Ivan
200134	2008-09-12	Winston	Anderson
200106	2008-12-20	Sasikant	Mukesh
200104	2009-03-21	Shilton	Ivan
200135	2009-06-23	Ramanathan	Santakumar
200119	2008-09-28	Ramanathan	Santakumar
200107	2008-05-27	Ramanathan	Santakumar
200121	2008-09-23	Karolina	Ivan
200108	2008-12-05	Karolina	Ivan
200133	2009-09-26	Ramesh	Mukesh
200128	2009-07-24	Ramesh	Mukesh
200120	2009-02-20	Ramesh	Mukesh
200116	2009-05-21	Charles	Benjamin
200109	2008-07-12	Sundariya	Santakumar
200131	2009-08-17	Steven	Lucida
200102	2008-06-12	Steven	Lucida
200100	2008-01-08	Holmes	Alex
200117	2008-09-27	Rangarappa	Subbarao
200127	2009-10-12	Stuart	Alex
200112	2008-05-30	Venkatpati	Ramasundar
200124	2009-03-26	Srinivas	Ramasundar
200125	2008-11-19	Fleming	Anderson
200110	2008-05-15	Yearannaidu	Santakumar
200111	2008-10-22	Albert	Alford

4)

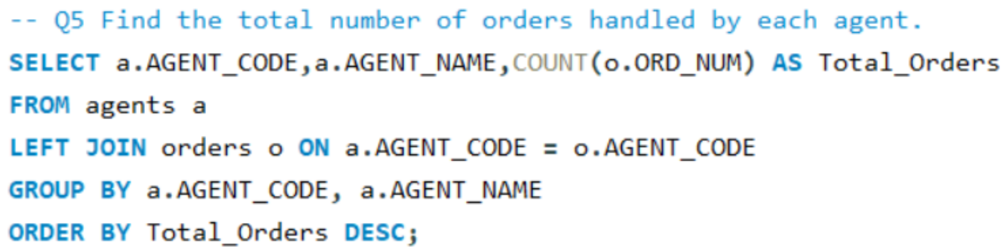


```sql
-- Q4 Find all orders placed by customers who reside in New York.
SELECT o.*
FROM orders o
JOIN customer c ON o.CUST_CODE = c.CUST_CODE
WHERE c.CUST_CITY = 'New York';
```

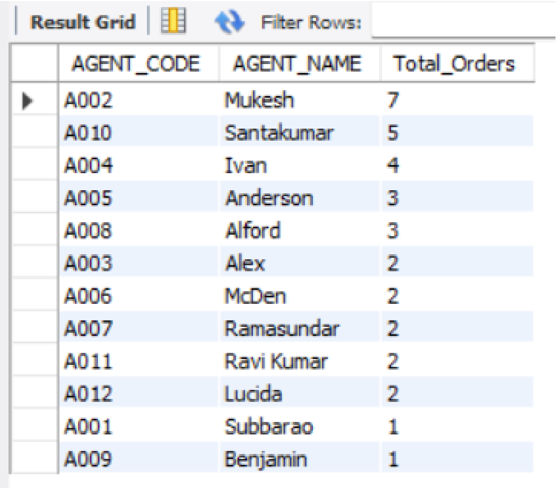


	ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
▶	200101	3000	1000	2008-08-21	C00001	A008	SOD
	200111	1000	300	2008-10-22	C00020	A008	SOD
	200114	3500	2000	2008-06-21	C00002	A008	SOD

5)

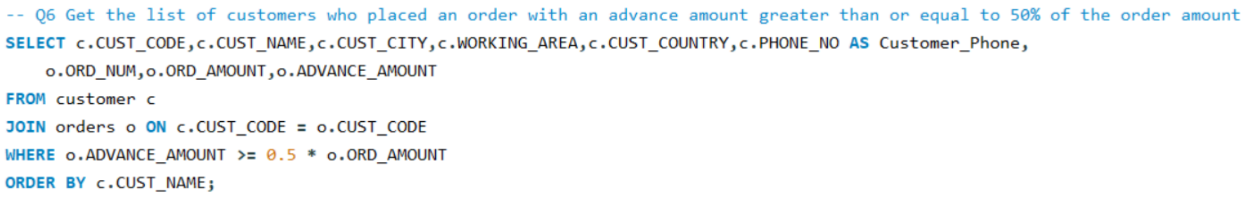


```
-- Q5 Find the total number of orders handled by each agent.
SELECT a.AGENT_CODE,a.AGENT_NAME,COUNT(o.ORD_NUM) AS Total_Orders
FROM agents a
LEFT JOIN orders o ON a.AGENT_CODE = o.AGENT_CODE
GROUP BY a.AGENT_CODE, a.AGENT_NAME
ORDER BY Total_Orders DESC;
```

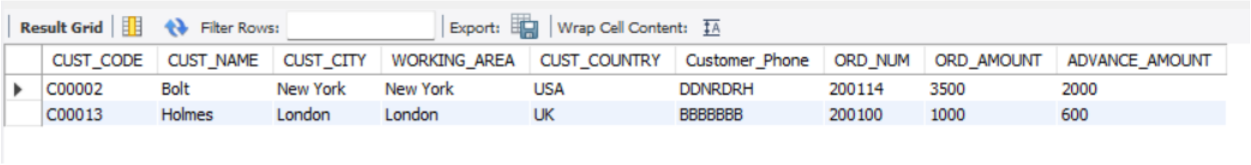


	AGENT_CODE	AGENT_NAME	Total_Orders
▶	A002	Mukesh	7
	A010	Santakumar	5
	A004	Ivan	4
	A005	Anderson	3
	A008	Alford	3
	A003	Alex	2
	A006	McDen	2
	A007	Ramasundar	2
	A011	Ravi Kumar	2
	A012	Lucida	2
	A001	Subbarao	1
	A009	Benjamin	1

6)

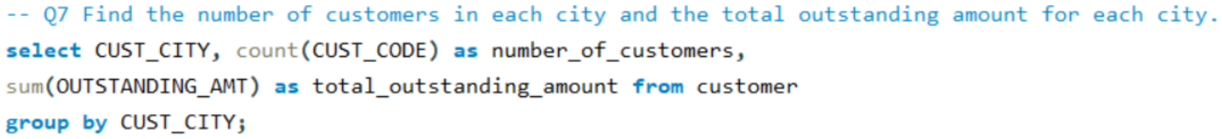


```
-- Q6 Get the list of customers who placed an order with an advance amount greater than or equal to 50% of the order amount
SELECT c.CUST_CODE,c.CUST_NAME,c.CUST_CITY,c.WORKING_AREA,c.CUST_COUNTRY,c.PHONE_NO AS Customer_Phone,
    o.ORD_NUM,o.ORD_AMOUNT,o.ADVANCE_AMOUNT
FROM customer c
JOIN orders o ON c.CUST_CODE = o.CUST_CODE
WHERE o.ADVANCE_AMOUNT >= 0.5 * o.ORD_AMOUNT
ORDER BY c.CUST_NAME;
```

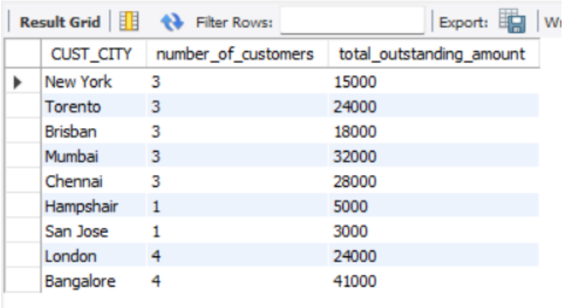


CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	Customer_Phone	ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT
C00002	Bolt	New York	New York	USA	DDNRDRH	200114	3500	2000
C00013	Holmes	London	London	UK	BBBBBBB	200100	1000	600

7)

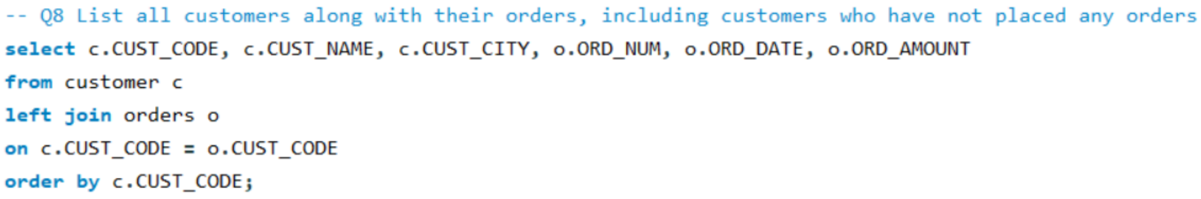


```
-- Q7 Find the number of customers in each city and the total outstanding amount for each city.
select CUST_CITY, count(CUST_CODE) as number_of_customers,
sum(OUTSTANDING_AMT) as total_outstanding_amount from customer
group by CUST_CITY;
```

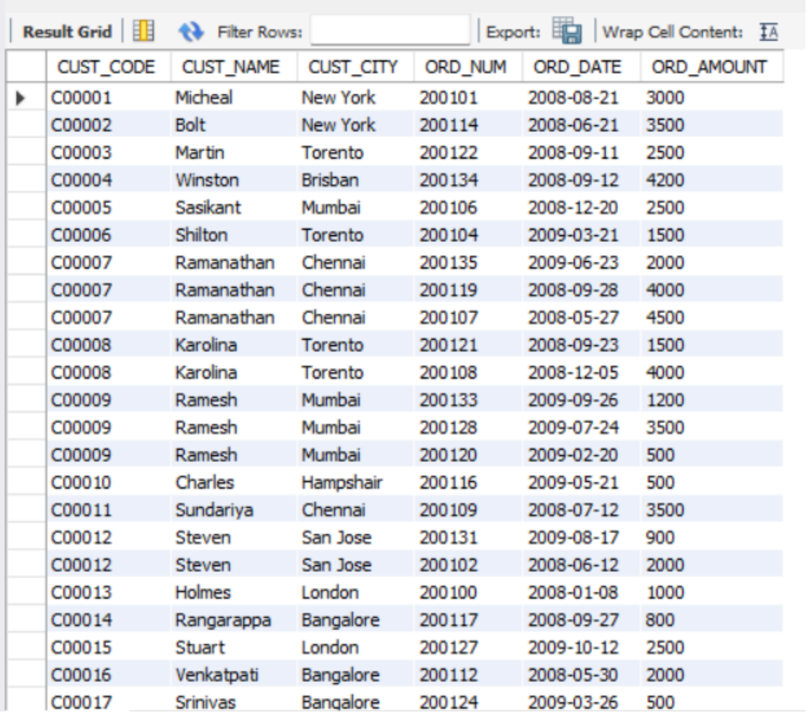


CUST_CITY	number_of_customers	total_outstanding_amount
New York	3	15000
Torento	3	24000
Brisban	3	18000
Mumbai	3	32000
Chennai	3	28000
Hampshair	1	5000
San Jose	1	3000
London	4	24000
Bangalore	4	41000

8)

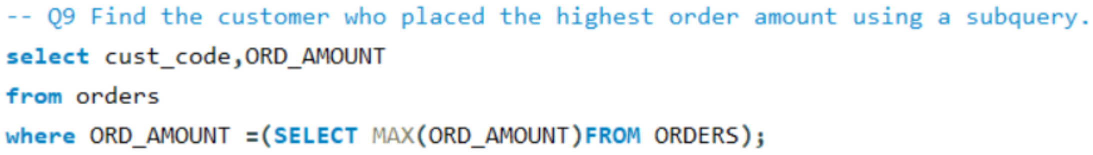


```
-- Q8 List all customers along with their orders, including customers who have not placed any orders
select c.CUST_CODE, c.CUST_NAME, c.CUST_CITY, o.ORD_NUM, o.ORD_DATE, o.ORD_AMOUNT
from customer c
left join orders o
on c.CUST_CODE = o.CUST_CODE
order by c.CUST_CODE;
```

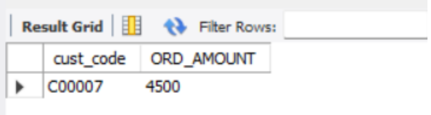


CUST_CODE	CUST_NAME	CUST_CITY	ORD_NUM	ORD_DATE	ORD_AMOUNT
C00001	Micheal	New York	200101	2008-08-21	3000
C00002	Bolt	New York	200114	2008-06-21	3500
C00003	Martin	Torento	200122	2008-09-11	2500
C00004	Winston	Brisban	200134	2008-09-12	4200
C00005	Sasikant	Mumbai	200106	2008-12-20	2500
C00006	Shilton	Torento	200104	2009-03-21	1500
C00007	Ramanathan	Chennai	200135	2009-06-23	2000
C00007	Ramanathan	Chennai	200119	2008-09-28	4000
C00007	Ramanathan	Chennai	200107	2008-05-27	4500
C00008	Karolina	Torento	200121	2008-09-23	1500
C00008	Karolina	Torento	200108	2008-12-05	4000
C00009	Ramesh	Mumbai	200133	2009-09-26	1200
C00009	Ramesh	Mumbai	200128	2009-07-24	3500
C00009	Ramesh	Mumbai	200120	2009-02-20	500
C00010	Charles	Hampshair	200116	2009-05-21	500
C00011	Sundariya	Chennai	200109	2008-07-12	3500
C00012	Steven	San Jose	200131	2009-08-17	900
C00012	Steven	San Jose	200102	2008-06-12	2000
C00013	Holmes	London	200100	2008-01-08	1000
C00014	Rangarappa	Bangalore	200117	2008-09-27	800
C00015	Stuart	London	200127	2009-10-12	2500
C00016	Venkatpati	Bangalore	200112	2008-05-30	2000
C00017	Srinivas	Bangalore	200124	2009-03-26	500

9)

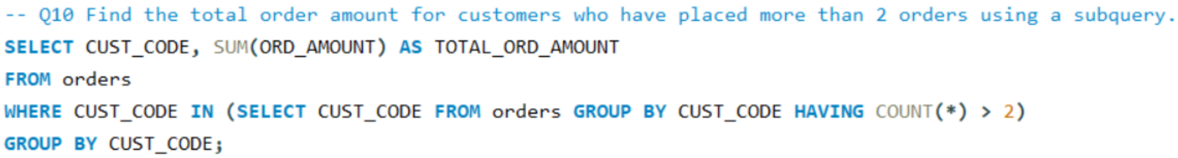


```
-- Q9 Find the customer who placed the highest order amount using a subquery.
select cust_code,ORD_AMOUNT
from orders
where ORD_AMOUNT =(SELECT MAX(ORD_AMOUNT)FROM ORDERS);
```

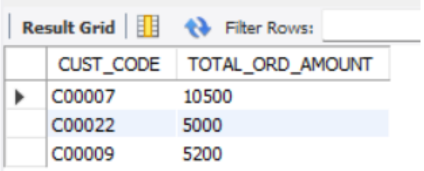


cust_code	ORD_AMOUNT
C00007	4500

10)

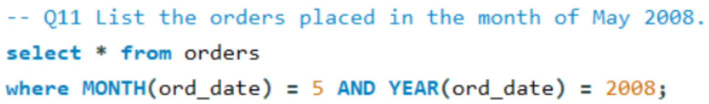


```
-- Q10 Find the total order amount for customers who have placed more than 2 orders using a subquery.
SELECT CUST_CODE, SUM(ORD_AMOUNT) AS TOTAL_ORD_AMOUNT
FROM orders
WHERE CUST_CODE IN (SELECT CUST_CODE FROM orders GROUP BY CUST_CODE HAVING COUNT(*) > 2)
GROUP BY CUST_CODE;
```

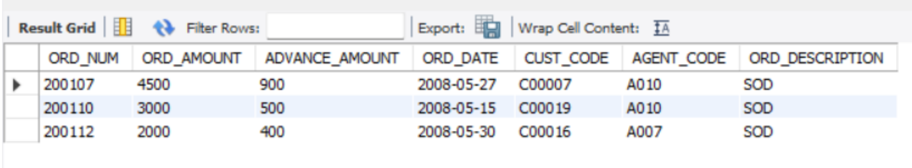


CUST_CODE	TOTAL_ORD_AMOUNT
C00007	10500
C00022	5000
C00009	5200

11)

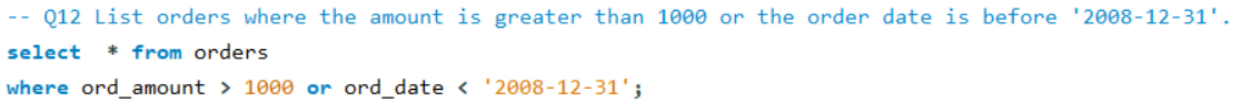


```sql
-- Q11 List the orders placed in the month of May 2008.
select * from orders
where MONTH(ord_date) = 5 AND YEAR(ord_date) = 2008;
```

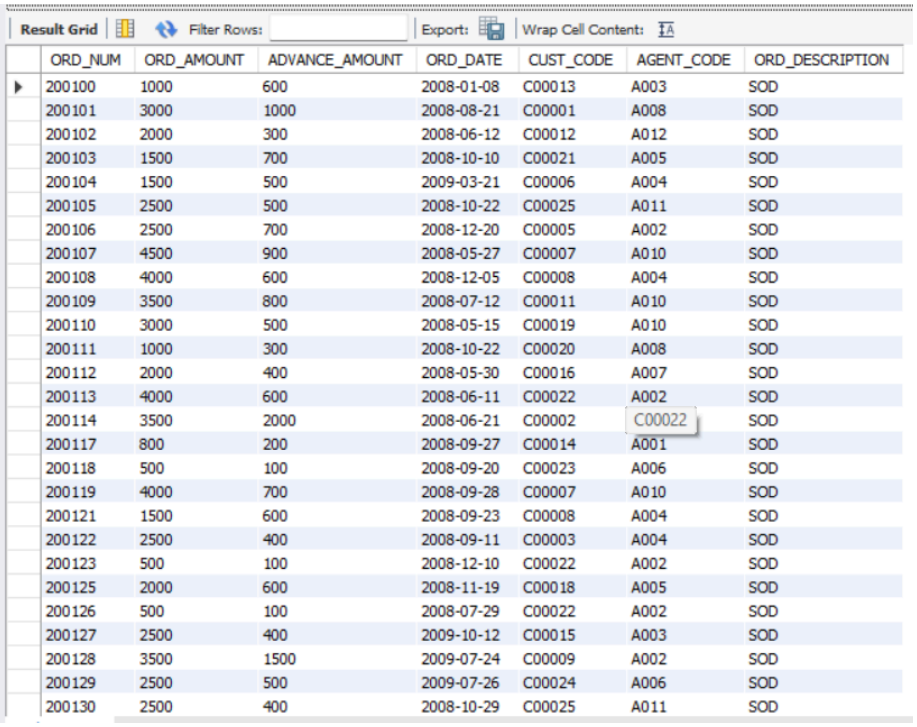


ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
200107	4500	900	2008-05-27	C00007	A010	SOD
200110	3000	500	2008-05-15	C00019	A010	SOD
200112	2000	400	2008-05-30	C00016	A007	SOD

12)

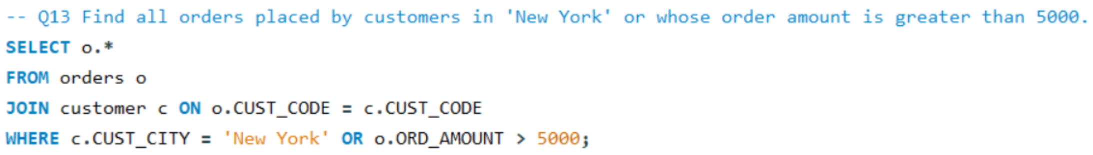


```sql
-- Q12 List orders where the amount is greater than 1000 or the order date is before '2008-12-31'.
select  * from orders
where ord_amount > 1000 or ord_date < '2008-12-31';
```

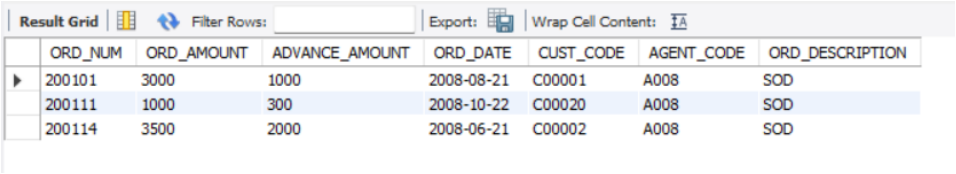


ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
200100	1000	600	2008-01-08	C00013	A003	SOD
200101	3000	1000	2008-08-21	C00001	A008	SOD
200102	2000	300	2008-06-12	C00012	A012	SOD
200103	1500	700	2008-10-10	C00021	A005	SOD
200104	1500	500	2009-03-21	C00006	A004	SOD
200105	2500	500	2008-10-22	C00025	A011	SOD
200106	2500	700	2008-12-20	C00005	A002	SOD
200107	4500	900	2008-05-27	C00007	A010	SOD
200108	4000	600	2008-12-05	C00008	A004	SOD
200109	3500	800	2008-07-12	C00011	A010	SOD
200110	3000	500	2008-05-15	C00019	A010	SOD
200111	1000	300	2008-10-22	C00020	A008	SOD
200112	2000	400	2008-05-30	C00016	A007	SOD
200113	4000	600	2008-06-11	C00022	A002	SOD
200114	3500	2000	2008-06-21	C00002	C00022	SOD
200117	800	200	2008-09-27	C00014	A001	SOD
200118	500	100	2008-09-20	C00023	A006	SOD
200119	4000	700	2008-09-28	C00007	A010	SOD
200121	1500	600	2008-09-23	C00008	A004	SOD
200122	2500	400	2008-09-11	C00003	A004	SOD
200123	500	100	2008-12-10	C00022	A002	SOD
200125	2000	600	2008-11-19	C00018	A005	SOD
200126	500	100	2008-07-29	C00022	A002	SOD
200127	2500	400	2009-10-12	C00015	A003	SOD
200128	3500	1500	2009-07-24	C00009	A002	SOD
200129	2500	500	2009-07-26	C00024	A006	SOD
200130	2500	400	2008-10-29	C00025	A011	SOD

13)



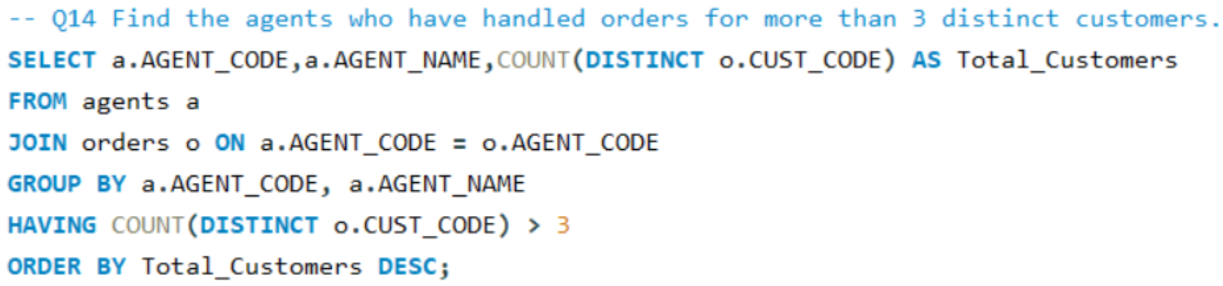
```sql
-- Q13 Find all orders placed by customers in 'New York' or whose order amount is greater than 5000.
SELECT o.*
FROM orders o
JOIN customer c ON o.CUST_CODE = c.CUST_CODE
WHERE c.CUST_CITY = 'New York' OR o.ORD_AMOUNT > 5000;
```



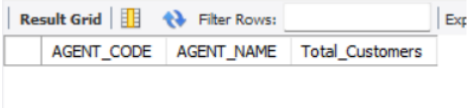
	ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
▶	200101	3000	1000	2008-08-21	C00001	A008	SOD
	200111	1000	300	2008-10-22	C00020	A008	SOD
	200114	3500	2000	2008-06-21	C00002	A008	SOD

---

14)



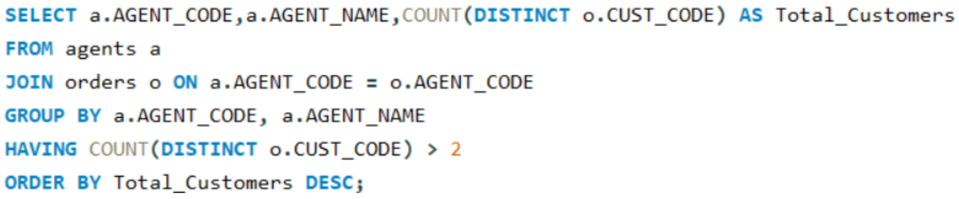
```sql
-- Q14 Find the agents who have handled orders for more than 3 distinct customers.
SELECT a.AGENT_CODE,a.AGENT_NAME,COUNT(DISTINCT o.CUST_CODE) AS Total_Customers
FROM agents a
JOIN orders o ON a.AGENT_CODE = o.AGENT_CODE
GROUP BY a.AGENT_CODE, a.AGENT_NAME
HAVING COUNT(DISTINCT o.CUST_CODE) > 3
ORDER BY Total_Customers DESC;
```



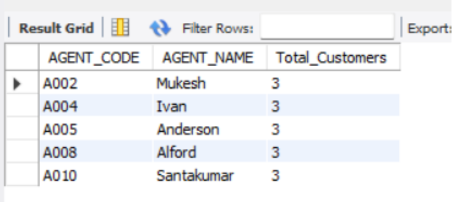
	AGENT_CODE	AGENT_NAME	Total_Customers

Ay: The question specifically asked for the agents who had handled orders for **more than 3** distinct customers, but we observed the query returned an empty table.

If the same query is run for agents who had handled orders for **more than 2** distinct customers, we get the following results;



```sql
SELECT a.AGENT_CODE,a.AGENT_NAME,COUNT(DISTINCT o.CUST_CODE) AS Total_Customers
FROM agents a
JOIN orders o ON a.AGENT_CODE = o.AGENT_CODE
GROUP BY a.AGENT_CODE, a.AGENT_NAME
HAVING COUNT(DISTINCT o.CUST_CODE) > 2
ORDER BY Total_Customers DESC;
```

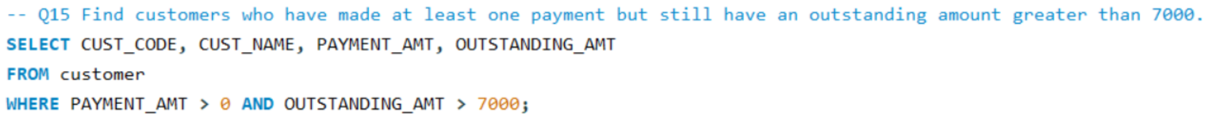


AGENT_CODE	AGENT_NAME	Total_Customers
A002	Mukesh	3
A004	Ivan	3
A005	Anderson	3
A008	Alford	3
A010	Santakumar	3

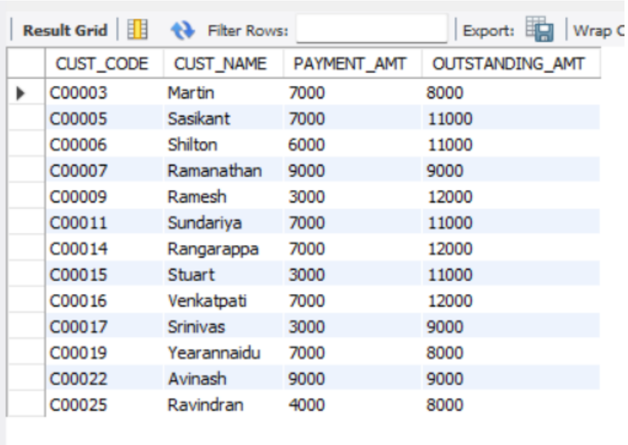
In conclusion, in the database the agents have handled orders for **at most 3** distinct customers.

---

15)



```
-- Q15 Find customers who have made at least one payment but still have an outstanding amount greater than 7000.
SELECT CUST_CODE, CUST_NAME, PAYMENT_AMT, OUTSTANDING_AMT
FROM customer
WHERE PAYMENT_AMT > 0 AND OUTSTANDING_AMT > 7000;
```



CUST_CODE	CUST_NAME	PAYMENT_AMT	OUTSTANDING_AMT
C00003	Martin	7000	8000
C00005	Sasikant	7000	11000
C00006	Shilton	6000	11000
C00007	Ramanathan	9000	9000
C00009	Ramesh	3000	12000
C00011	Sundariya	7000	11000
C00014	Rangarappa	7000	12000
C00015	Stuart	3000	11000
C00016	Venkatpati	7000	12000
C00017	Srinivas	3000	9000
C00019	Yearannaidu	7000	8000
C00022	Avinash	9000	9000
C00025	Ravindran	4000	8000