**DBMS LAB ASSIGNMENT 5**

**K.V.L.AMRUTHA**

**19BCS048**

**1. Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the**

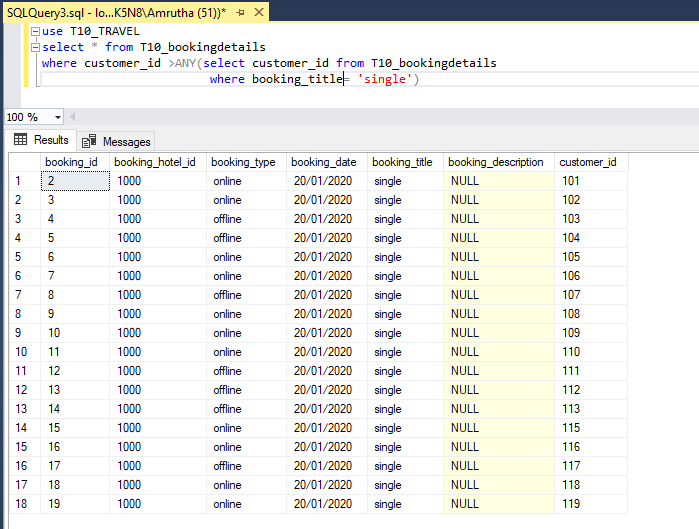
**difference between ANY and ALL**

use T10\_TRAVEL

select \* from T10\_bookingdetails

where customer\_id >ANY(select customer\_id from T10\_bookingdetails

where booking\_title= 'single')

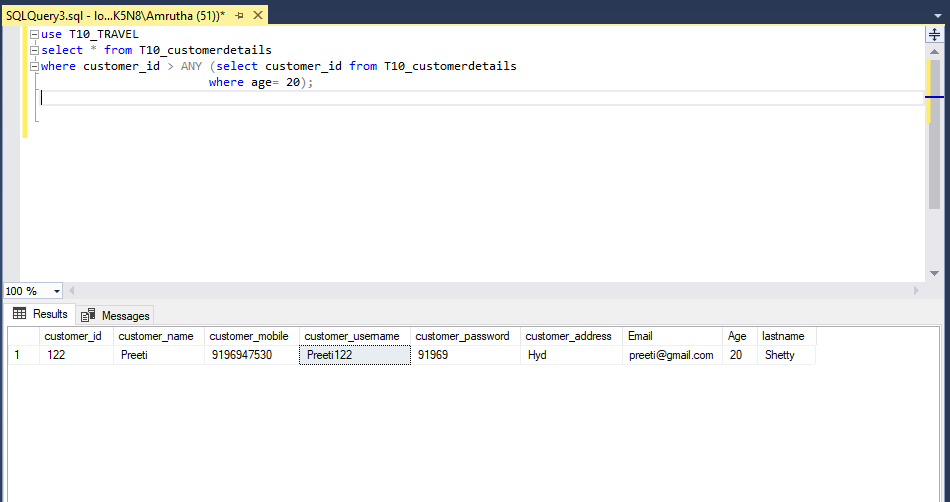


use T10\_TRAVEL

select \* from T10\_customerdetails

where customer\_id > ANY (select customer\_id from T10\_customerdetails

where age= 20);

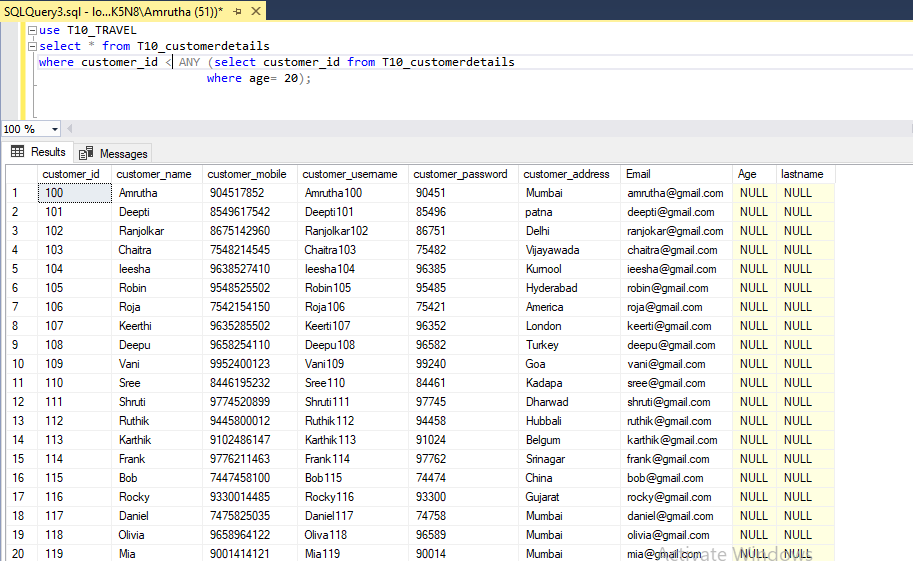


use T10\_TRAVEL

select \* from T10\_customerdetails

where customer\_id < ANY (select customer\_id from T10\_customerdetails

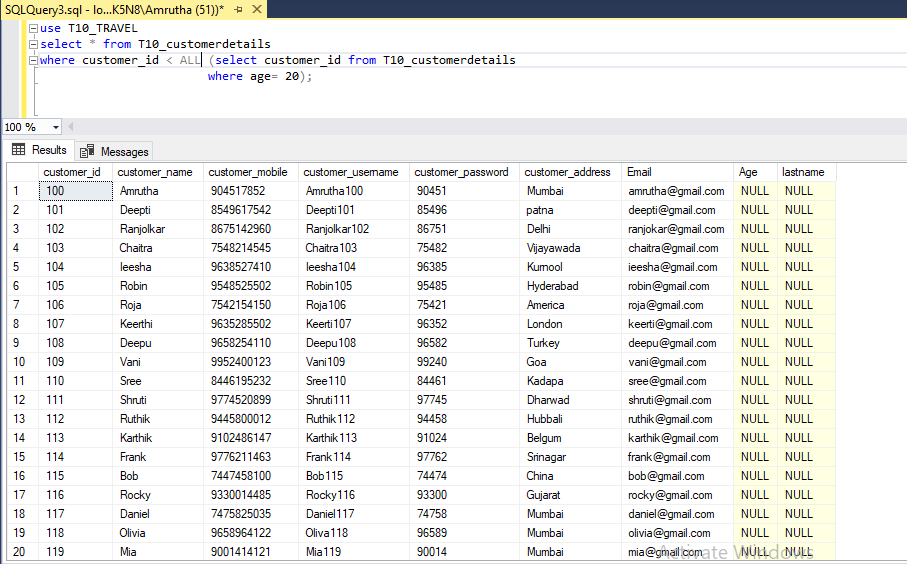
where age= 20);



use T10\_TRAVEL

select \* from T10\_customerdetails

where customer\_id < ALL (select customer\_id from T10\_customerdetails

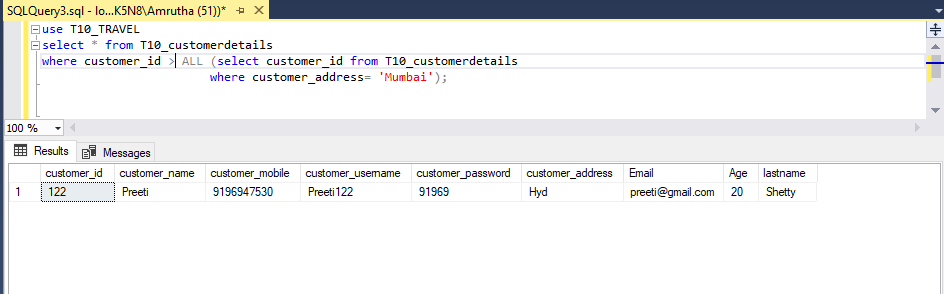
where age= 20); 

use T10\_TRAVEL

select \* from T10\_customerdetails

where customer\_id > ALL (select customer\_id from T10\_customerdetails

where customer\_address= 'Mumbai');

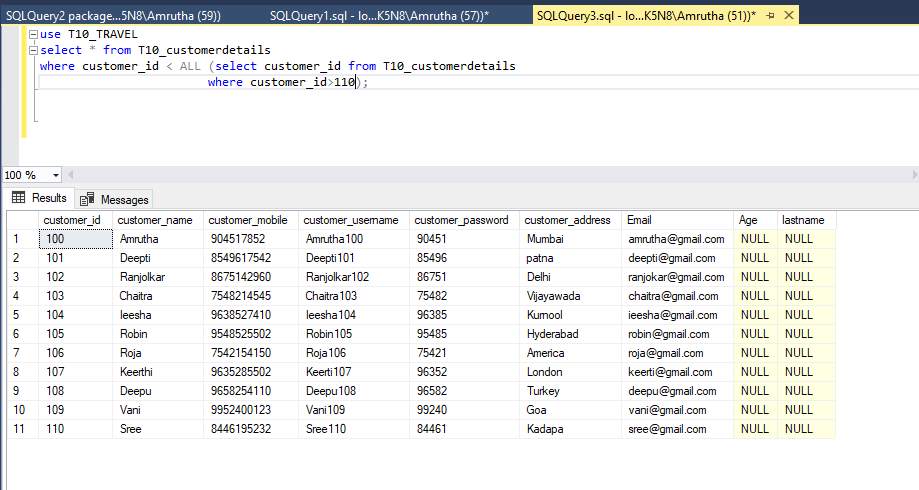


use T10\_TRAVEL

select \* from T10\_customerdetails

where customer\_id < ALL (select customer\_id from T10\_customerdetails

where customer\_id>110);



SELECT customer\_name, customer\_address FROM T10\_customerdetails

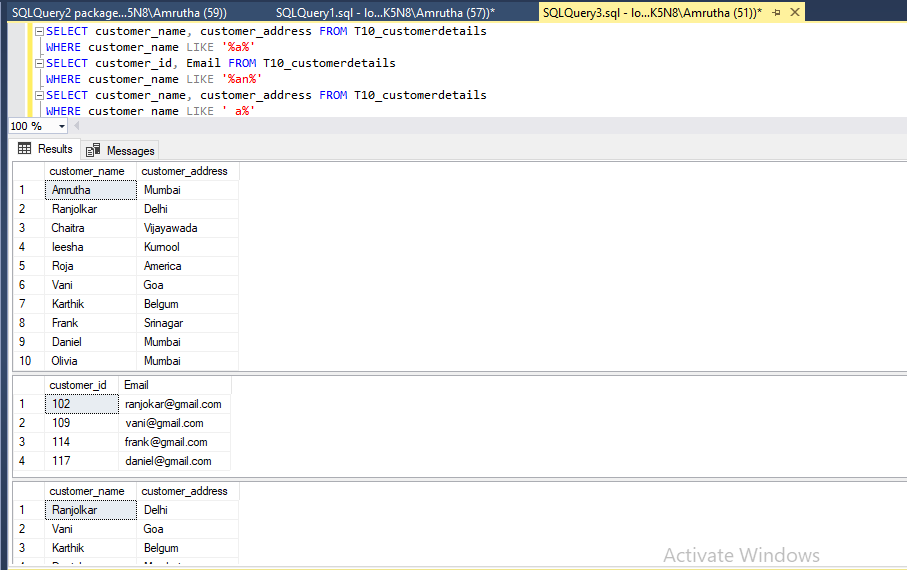
WHERE customer\_name LIKE '%a%'

SELECT customer\_id, Email FROM T10\_customerdetails

WHERE customer\_name LIKE '%an%'

SELECT customer\_name, customer\_address FROM T10\_customerdetails

WHERE customer\_name LIKE '\_a%'



**2. One query for each Aggregate function.**

SELECT AVG(package\_amount)

FROM T10\_packagedetails;

SELECT COUNT(customer\_mobile)

FROM T10\_customerdetails;

SELECT MIN(customer\_mobile)

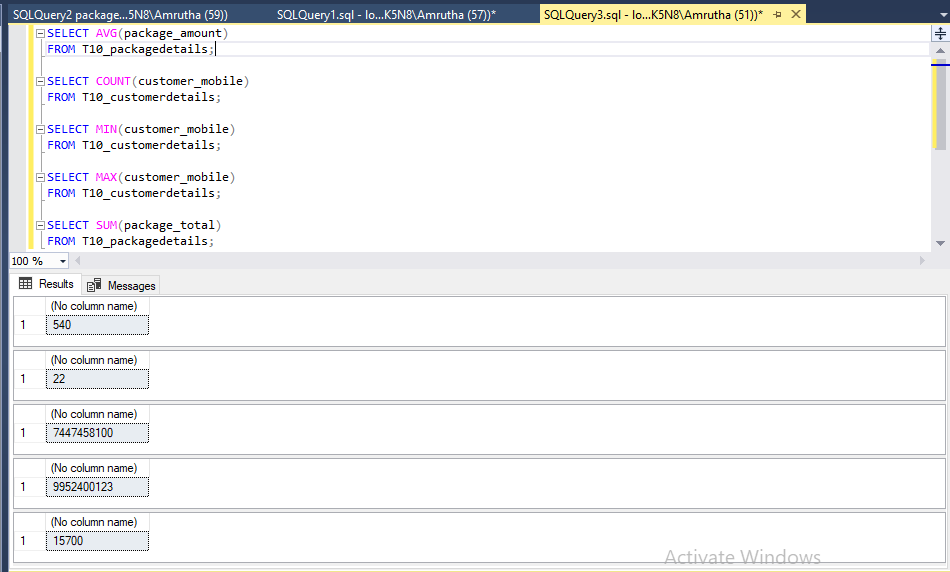
FROM T10\_customerdetails;

SELECT MAX(customer\_mobile)

FROM T10\_customerdetails;

SELECT SUM(package\_total)

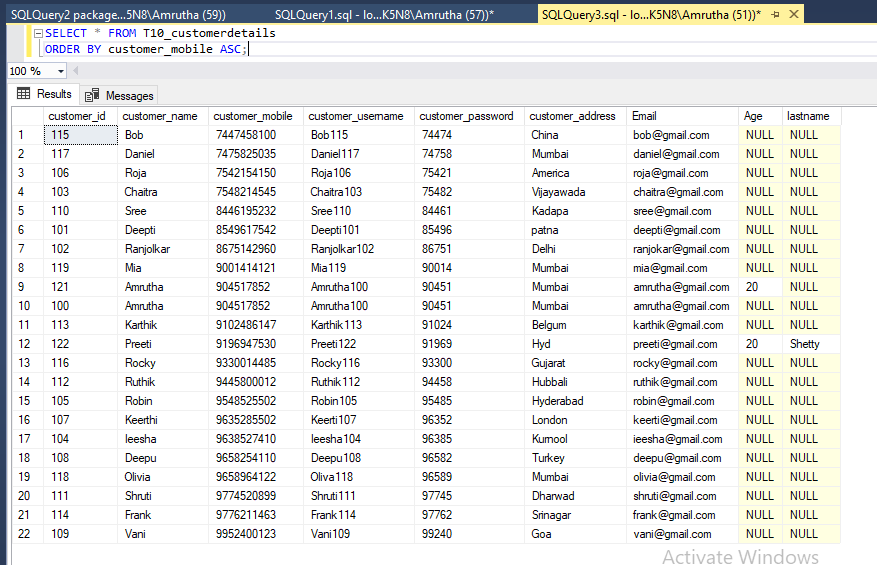
FROM T10\_packagedetails;



**3. Illustrate the usage of order by, group by and having clause (2 queries for each case)**

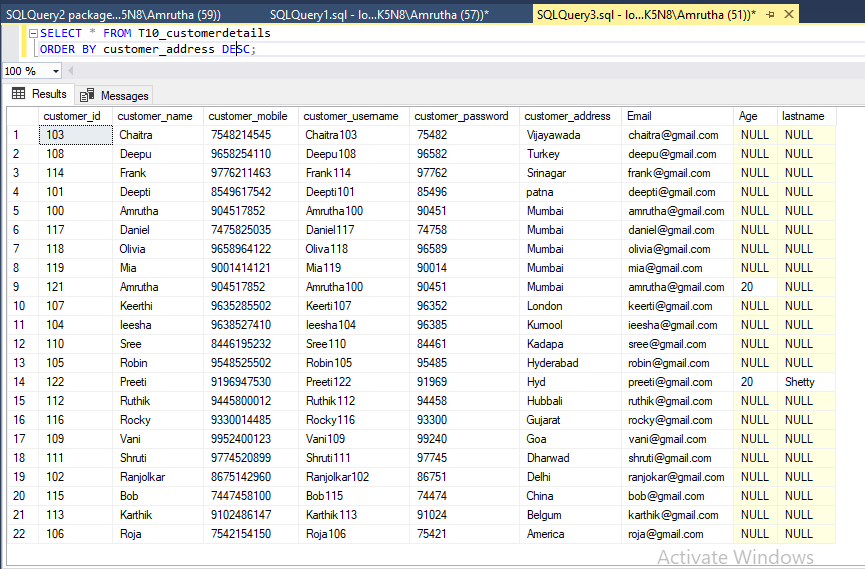
SELECT \* FROM T10\_customerdetails

ORDER BY customer\_mobile ASC;



SELECT \* FROM T10\_customerdetails

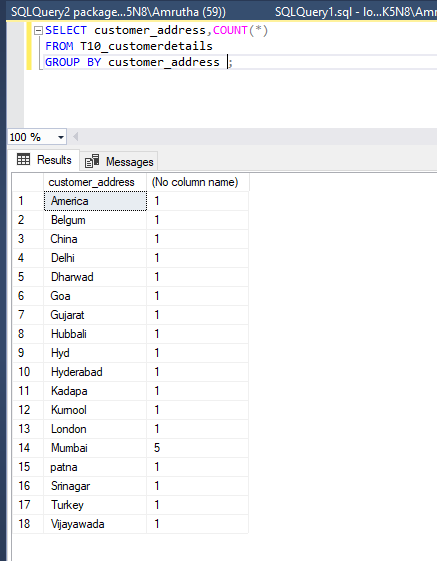
ORDER BY customer\_address DESC;



SELECT customer\_address,COUNT(\*)

FROM T10\_customerdetails

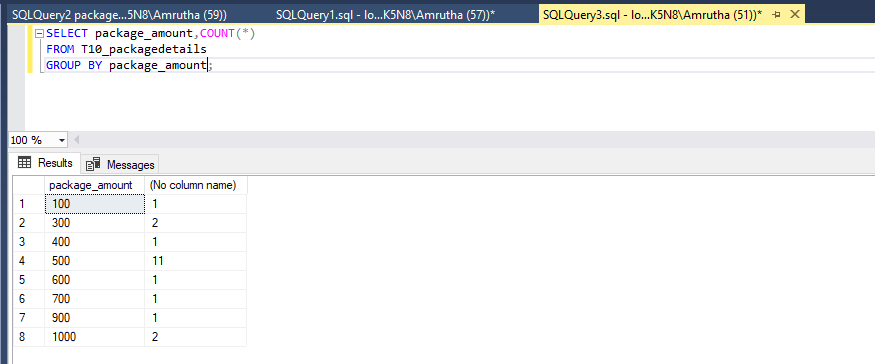
GROUP BY customer\_address ;



SELECT package\_amount,COUNT(\*)

FROM T10\_packagedetails

GROUP BY package\_amount;

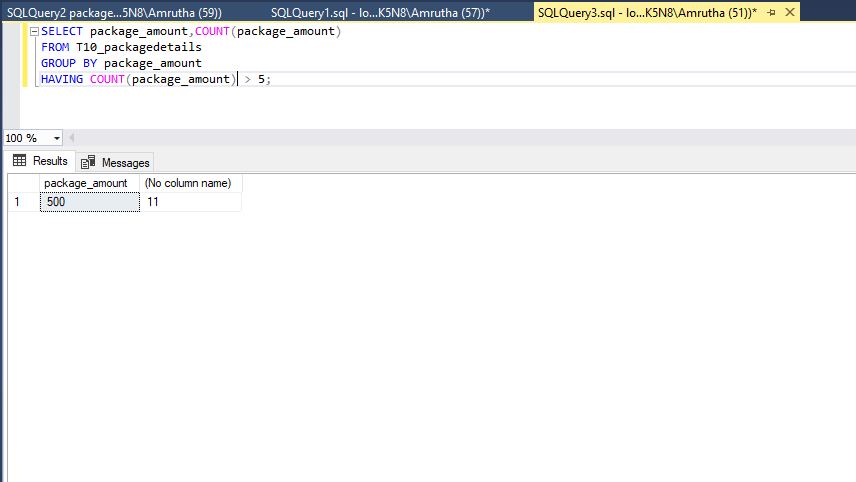


SELECT package\_amount,COUNT(package\_amount)

FROM T10\_packagedetails

GROUP BY package\_amount

HAVING COUNT(package\_amount) > 5;

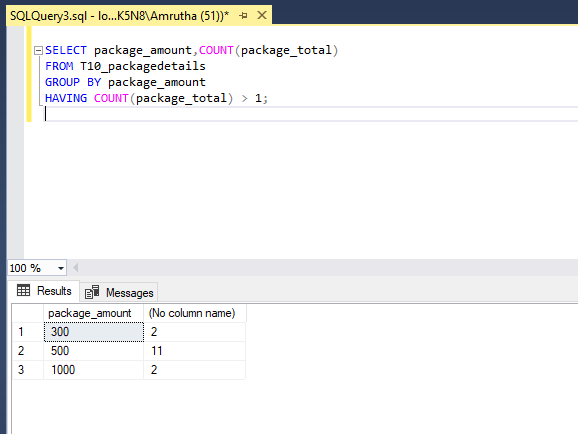


SELECT package\_amount,COUNT(package\_total)

FROM T10\_packagedetails

GROUP BY package\_amount

HAVING COUNT(package\_total) > 1;



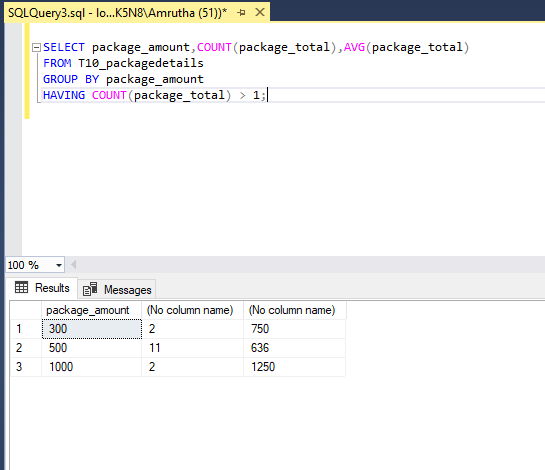
**4. Use Aggregate function with group by and having**

SELECT package\_amount,COUNT(package\_total),AVG(package\_total)

FROM T10\_packagedetails

GROUP BY package\_amount

HAVING COUNT(package\_total) > 1;

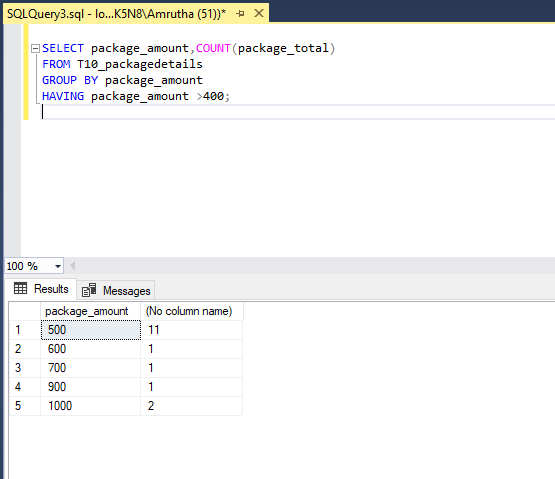


SELECT package\_amount,COUNT(package\_total)

FROM T10\_packagedetails

GROUP BY package\_amount

HAVING package\_amount >400;



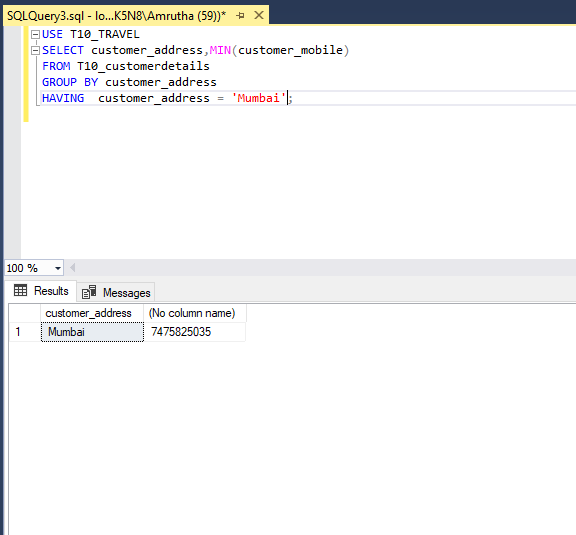
USE T10\_TRAVEL

SELECT customer\_address,MIN(customer\_mobile)

FROM T10\_customerdetails

GROUP BY customer\_address

HAVING customer\_address = 'Mumbai';



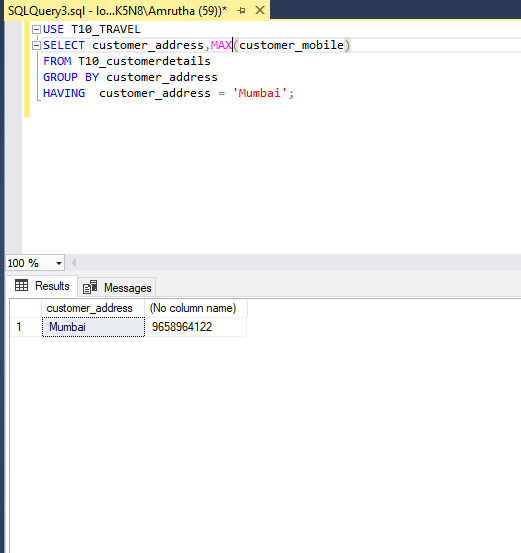
USE T10\_TRAVEL

SELECT customer\_address,MAX(customer\_mobile)

FROM T10\_customerdetails

GROUP BY customer\_address

HAVING customer\_address = 'Mumbai';



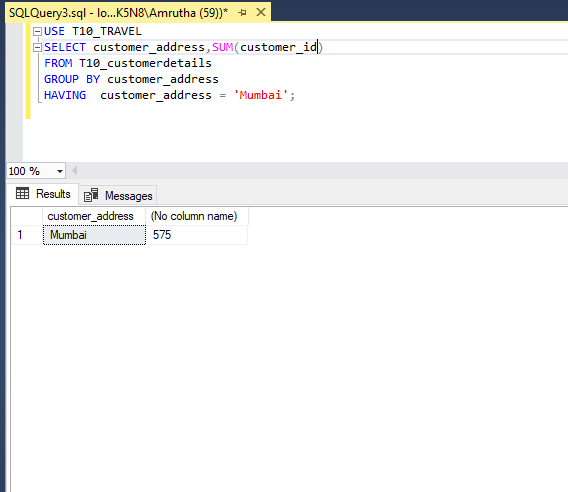
USE T10\_TRAVEL

SELECT customer\_address,SUM(customer\_id)

FROM T10\_customerdetails

GROUP BY customer\_address

HAVING customer\_address = 'Mumbai';



**5. Write at least 3 nested queries using order by, group by and having clause.**

SELECT package\_amount,SUM(package\_total)

FROM T10\_packagedetails

WHERE customer\_id = ANY

(

SELECT customer\_id

FROM T10\_bookingdetails

where customer\_id = ANY

(

SELECT customer\_id

FROM T10\_packagedetails

Where package\_amount>500

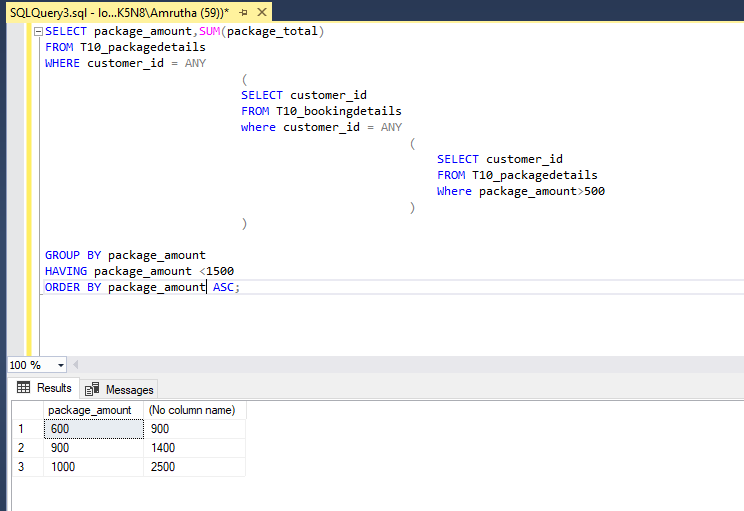
)

)

GROUP BY package\_amount

HAVING package\_amount <1500

ORDER BY package\_amount ASC;



**6. Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection**

USE T10\_TRAVEL

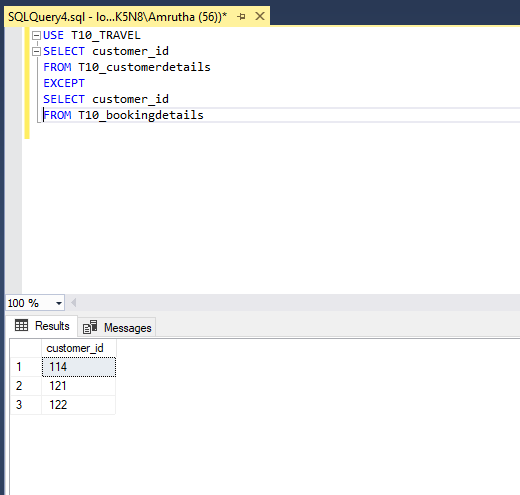
SELECT customer\_id

FROM T10\_customerdetails

EXCEPT

SELECT customer\_id

FROM T10\_bookingdetails



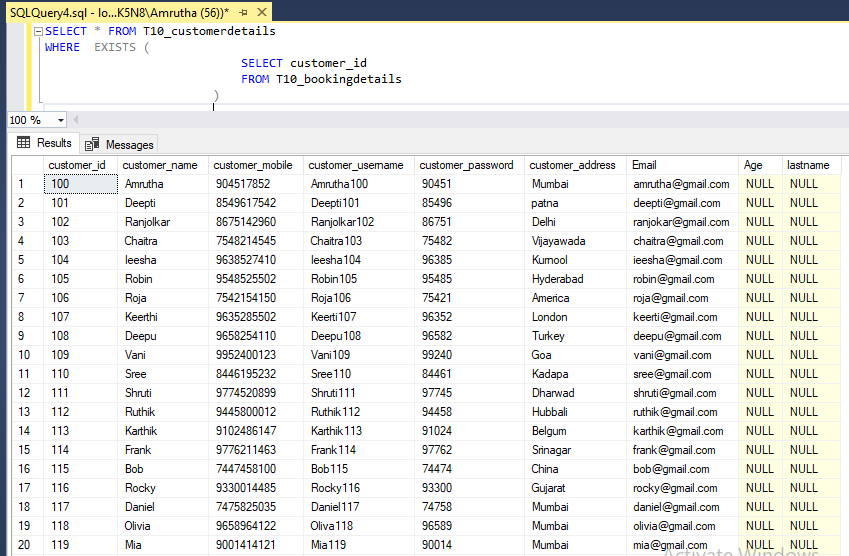
SELECT \* FROM T10\_customerdetails

WHERE EXISTS (

SELECT customer\_id

FROM T10\_bookingdetails

)



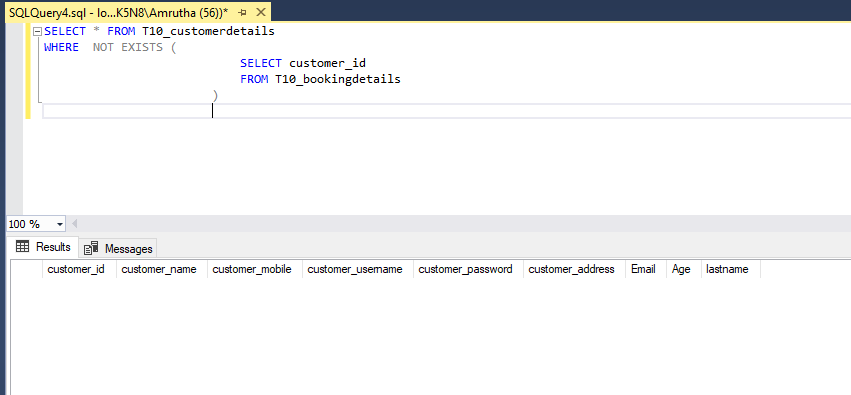
SELECT \* FROM T10\_customerdetails

WHERE NOT EXISTS (

SELECT customer\_id

FROM T10\_bookingdetails

)



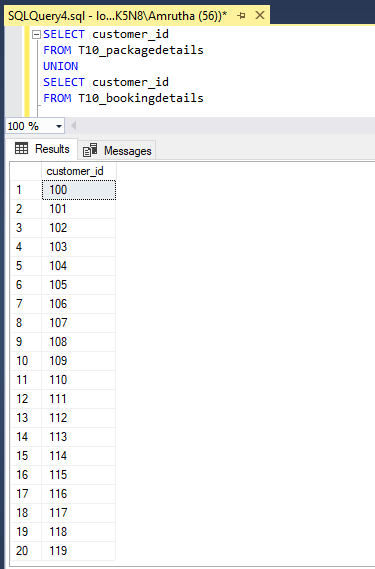
SELECT customer\_id

FROM T10\_packagedetails

UNION

SELECT customer\_id

FROM T10\_bookingdetails



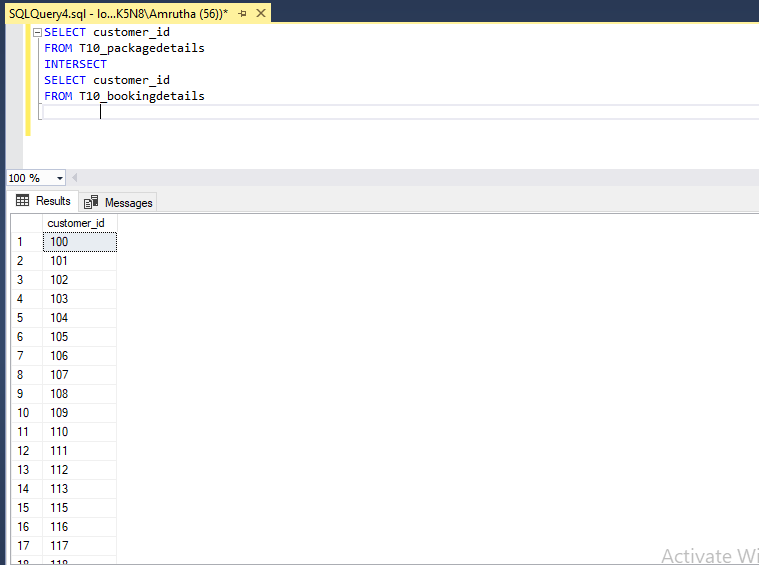
SELECT customer\_id

FROM T10\_packagedetails

INTERSECT

SELECT customer\_id

FROM T10\_bookingdetails



**7. INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance**

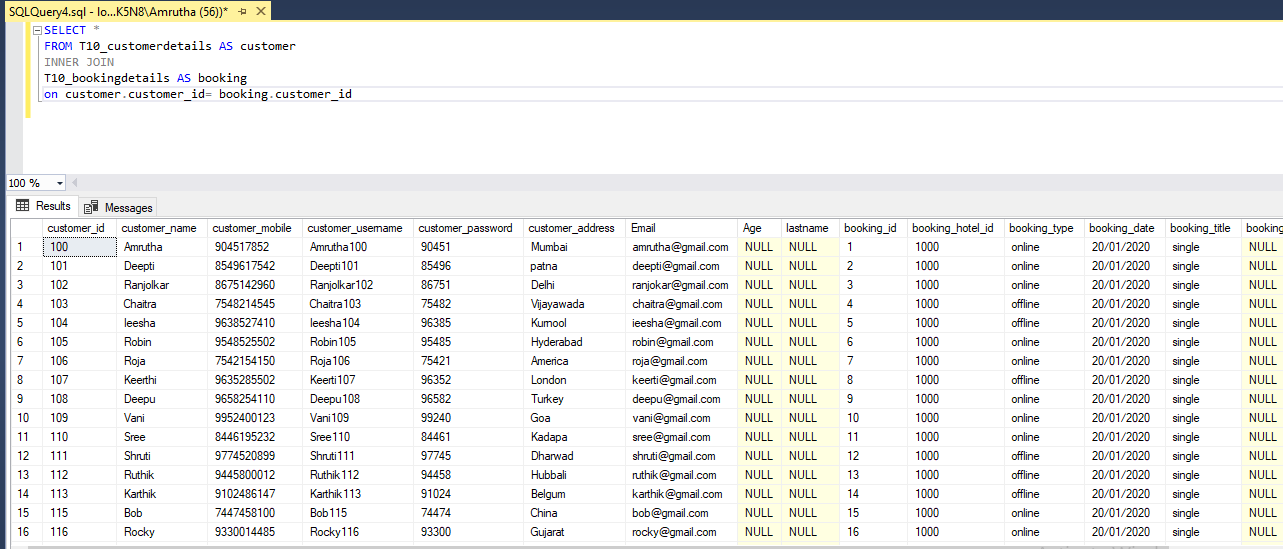
SELECT \*

FROM T10\_customerdetails AS customer

INNER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



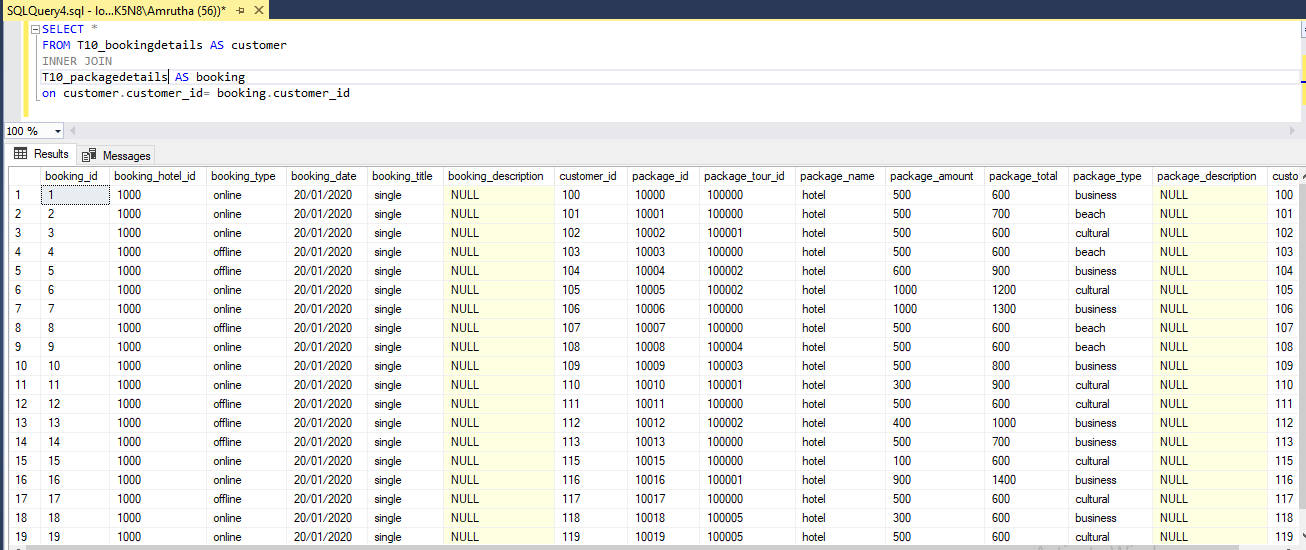
SELECT \*

FROM T10\_bookingdetails AS customer

INNER JOIN

T10\_packagedetails AS booking

on customer.customer\_id= booking.customer\_id



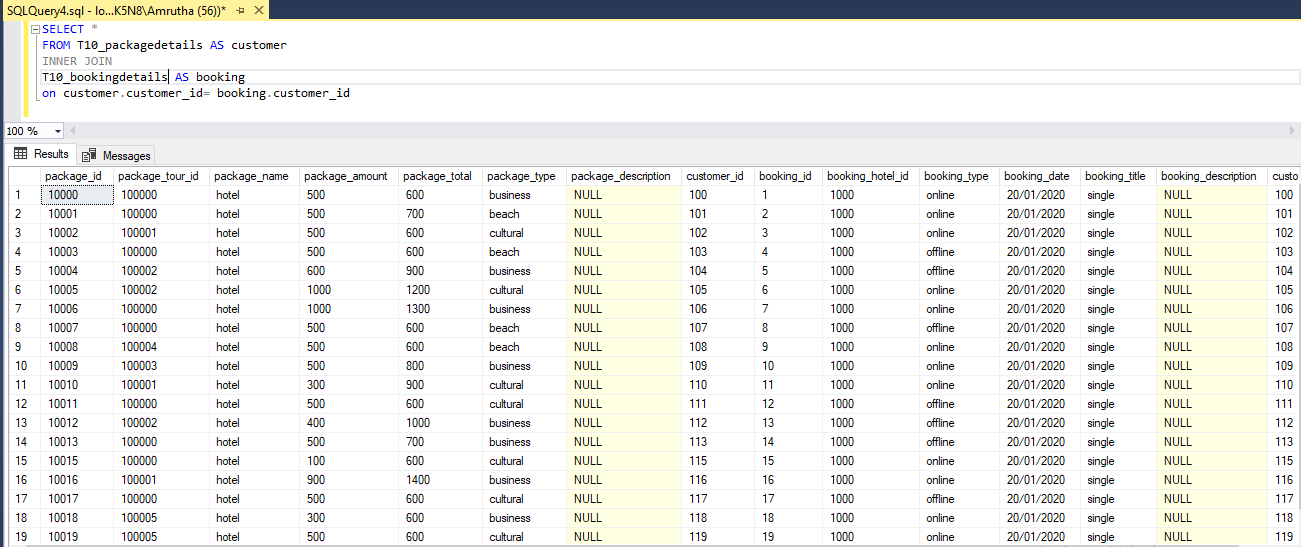
SELECT \*

FROM T10\_packagedetails AS customer

INNER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



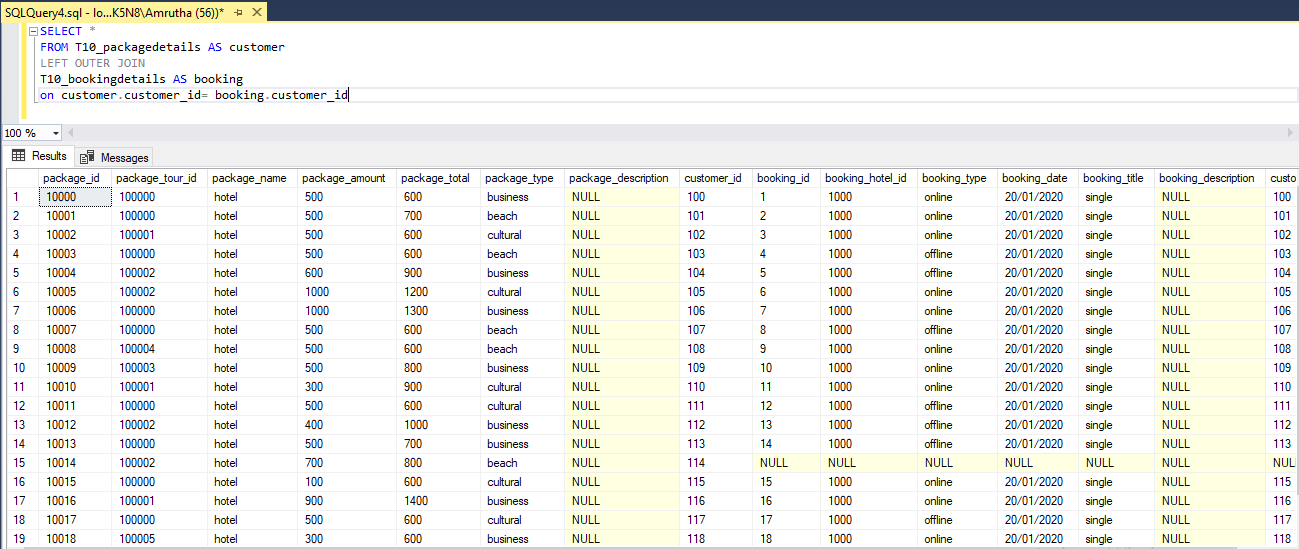
SELECT \*

FROM T10\_packagedetails AS customer

LEFT OUTER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



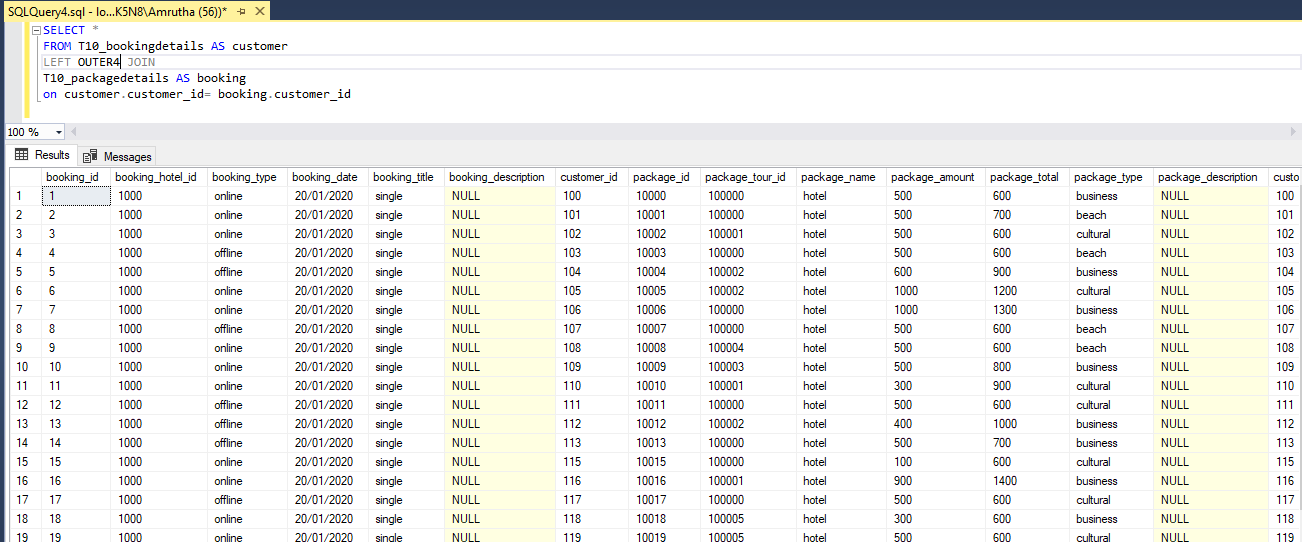
SELECT \*

FROM T10\_bookingdetails AS customer

LEFT OUTER JOIN

T10\_packagedetails AS booking

on customer.customer\_id= booking.customer\_id



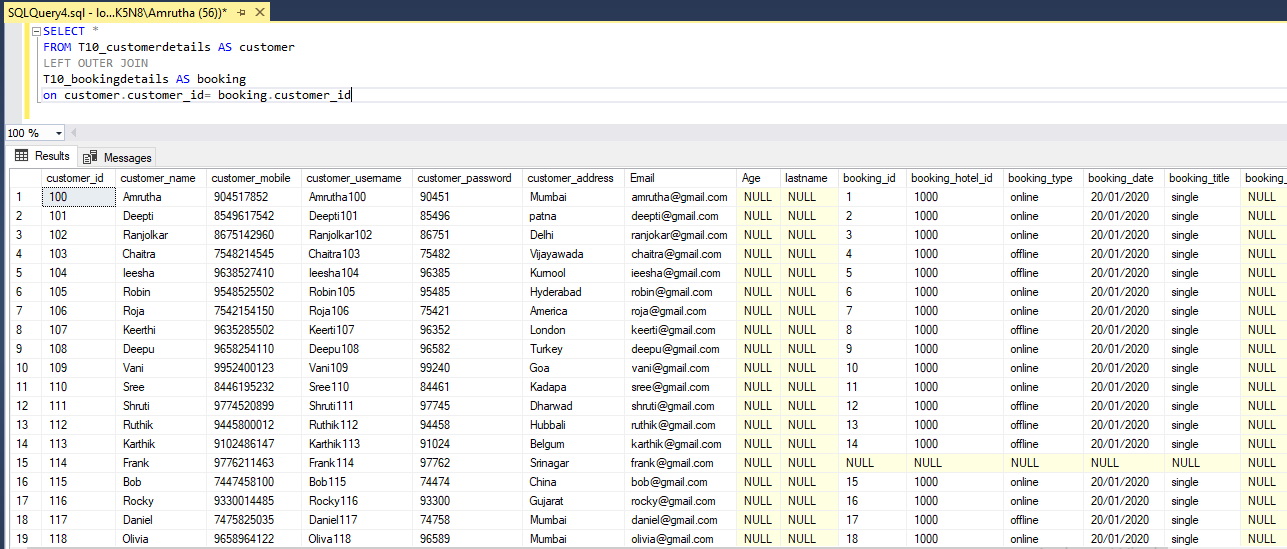
SELECT \*

FROM T10\_customerdetails AS customer

LEFT OUTER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



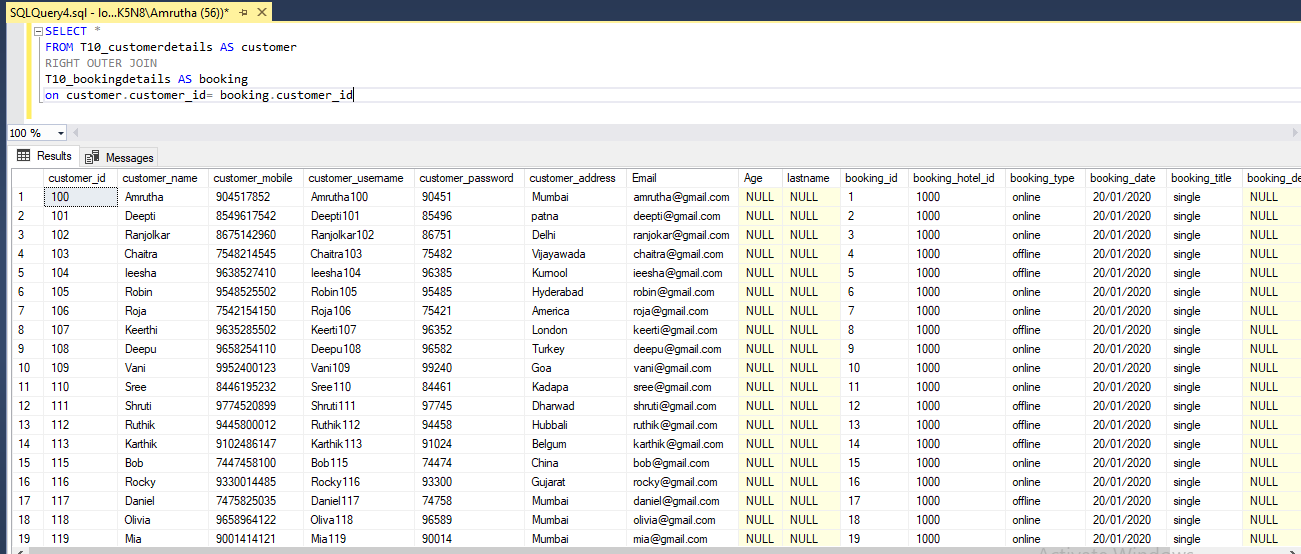
SELECT \*

FROM T10\_customerdetails AS customer

RIGHT OUTER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



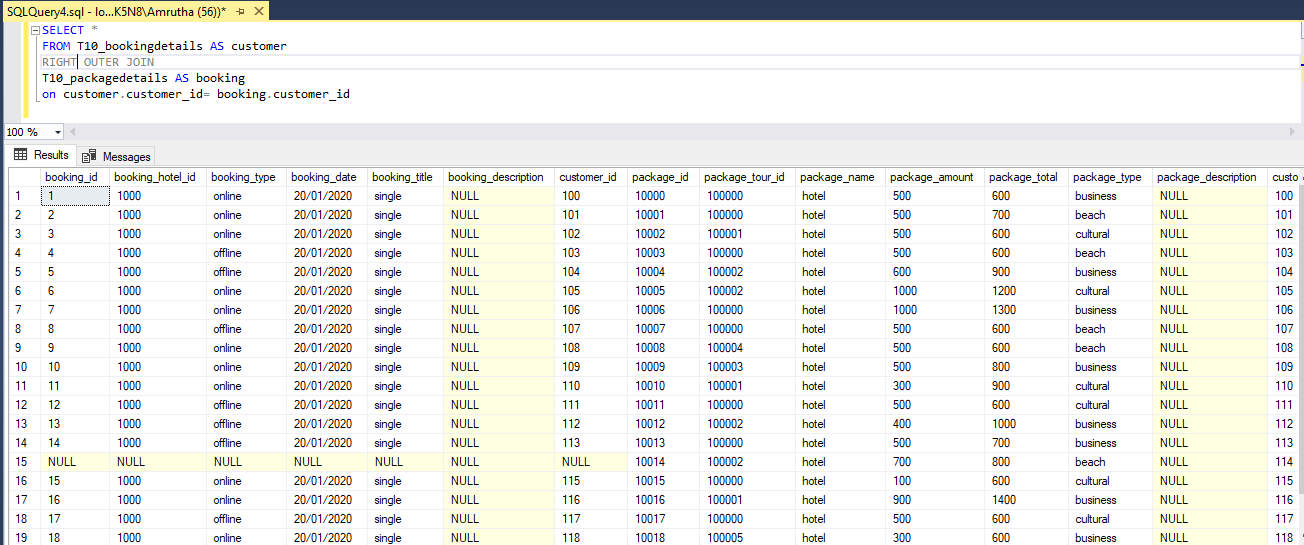
SELECT \*

FROM T10\_bookingdetails AS customer

RIGHT OUTER JOIN

T10\_packagedetails AS booking

on customer.customer\_id= booking.customer\_id



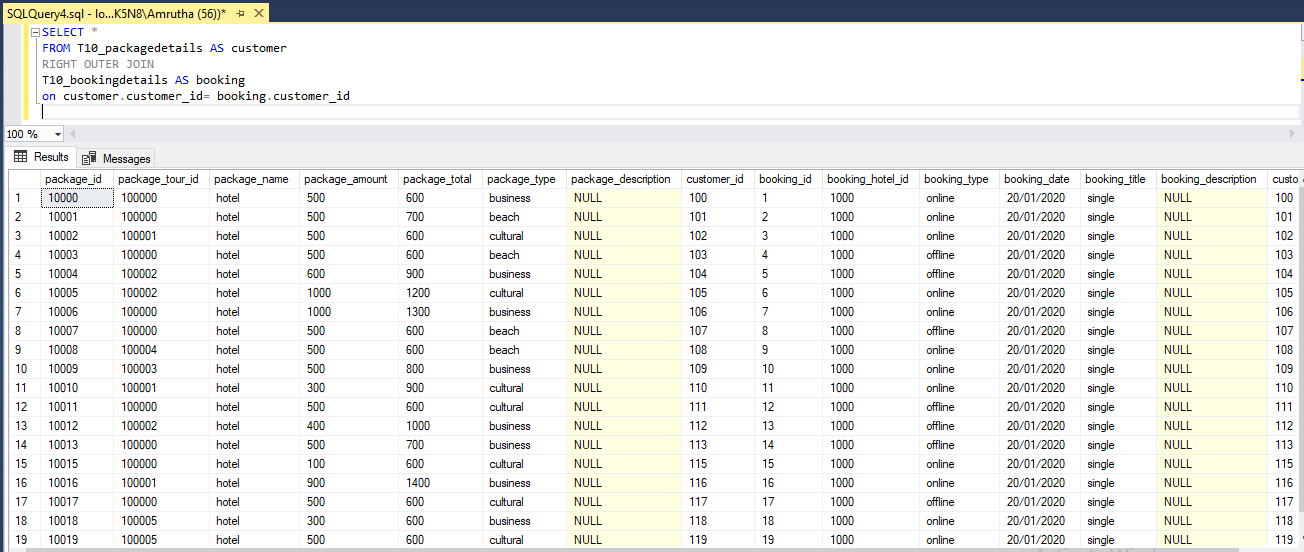
SELECT \*

FROM T10\_packagedetails AS customer

RIGHT OUTER JOIN

T10\_bookingdetails AS booking

on customer.customer\_id= booking.customer\_id



**8. Use all the above condition in JOIN as well.**

use T10\_TRAVEL

SELECT package\_type, SUM(package\_amount) AS Amount ,MIN(package\_total) AS Total\_Min,AVG(package\_tour\_id)AS Tour\_id,MAX(package\_total) AS Total\_Max

FROM T10\_packagedetails As customer

JOIN

T10\_bookingdetails AS booking

on customer.customer\_id=booking.customer\_id

GROUP BY package\_type

HAVING package\_type LIKE '\_\_%'

ORDER BY package\_type ASC;

