

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

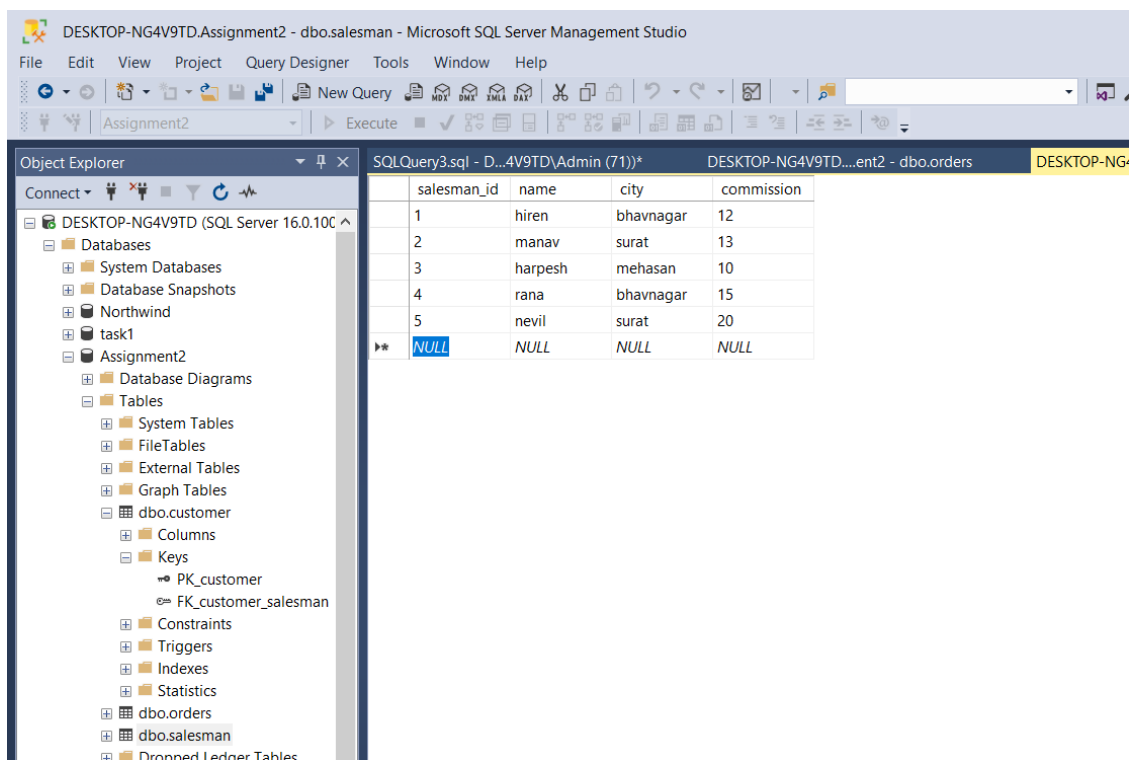
Sample table1: salesman

-salesman_id

-name

-city

-commission



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-NG4V9TD (SQL Server 16.0.100)'. The 'Tables' folder is expanded, showing 'dbo.customer' and 'dbo.orders'. The 'dbo.customer' table has a primary key 'PK_customer' and a foreign key 'FK_customer_salesman' that references the 'salesman' table. The 'dbo.orders' table is also visible. The main query window shows the results of a query, which is a table with four columns: 'salesman_id', 'name', 'city', and 'commission'. The results are as follows:

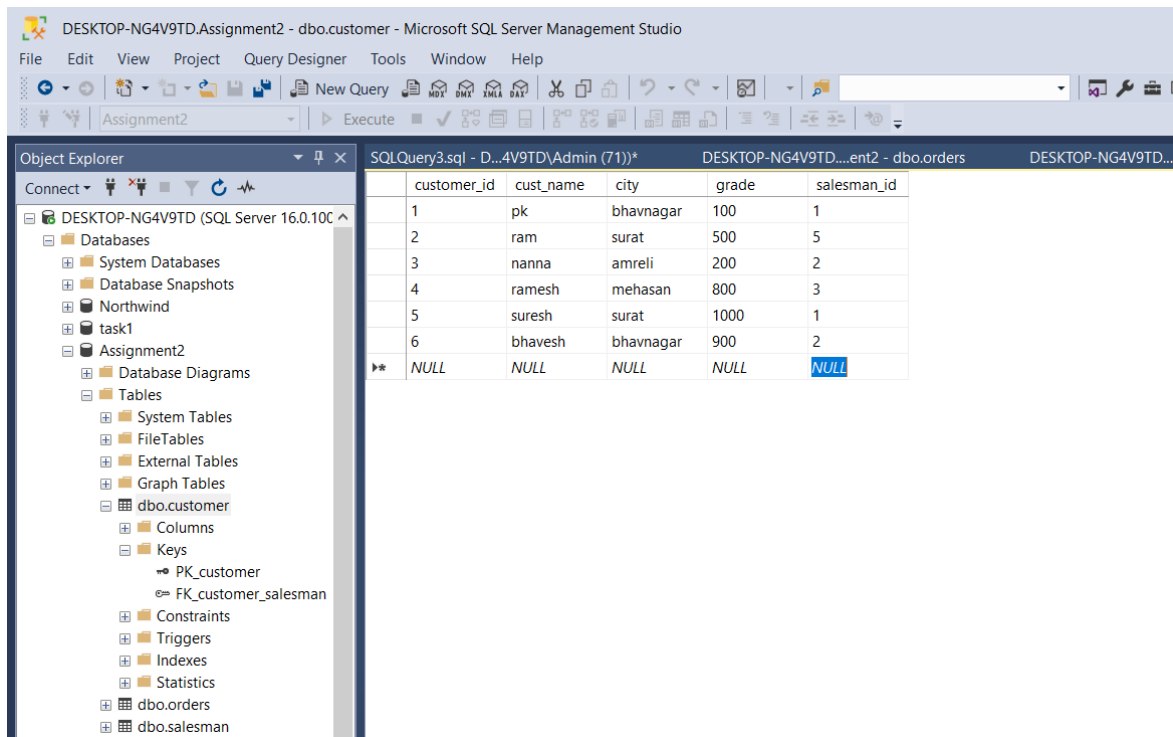
salesman_id	name	city	commission
1	hiren	bhavnagar	12
2	manav	surat	13
3	harpesh	mehasan	10
4	rana	bhavnagar	15
5	nevil	surat	20
NULL	NULL	NULL	NULL

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

Sample table2: customer

- customer_id
- cust_name
- city
- grade
- salesman_id



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the database structure for 'DESKTOP-NG4V9TD (SQL Server 16.0.100)'. The 'Tables' folder is expanded, showing 'dbo.customer'. The 'Query Designer' tab is active, displaying a table with 6 columns: 'customer_id', 'cust_name', 'city', 'grade', and 'salesman_id'. The table contains 6 rows of data, with the last row showing NULL values for all columns.

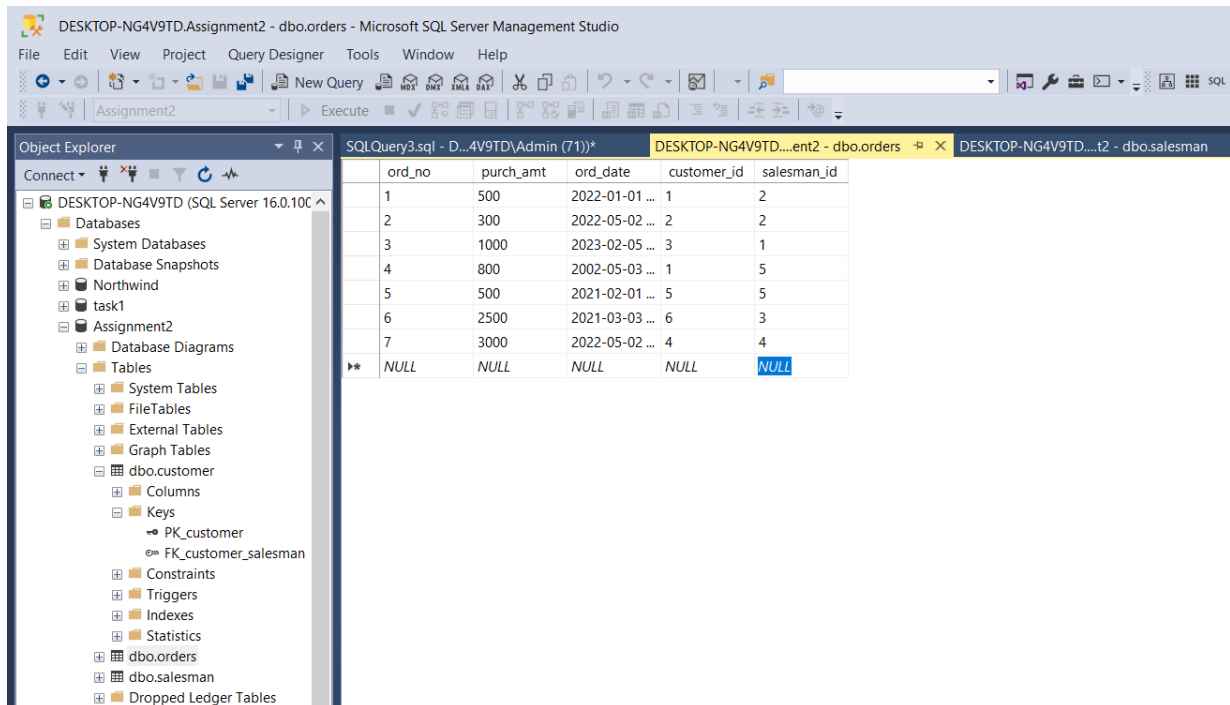
customer_id	cust_name	city	grade	salesman_id
1	pk	bhavnagar	100	1
2	ram	surat	500	5
3	nanna	amreli	200	2
4	ramesh	mehasan	800	3
5	suresh	surat	1000	1
6	bhavesh	bhavnagar	900	2
NULL	NULL	NULL	NULL	NULL

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

Sample table3: orders

- ord_no
- purch_amt
- ord_date
- customer_id
- salesman_id



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the database structure, including the 'Assignment2' database and its tables. The 'SQLQuery3.sql' window in the center shows the data from the 'orders' table. The data is as follows:

ord_no	purch_amt	ord_date	customer_id	salesman_id
1	500	2022-01-01 ...	1	2
2	300	2022-05-02 ...	2	2
3	1000	2023-02-05 ...	3	1
4	800	2002-05-03 ...	1	5
5	500	2021-02-01 ...	5	5
6	2500	2021-03-03 ...	6	3
7	3000	2022-05-02 ...	4	4
NULL	NULL	NULL	NULL	NULL

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

All tables display whit sql query

```
select * from salesman
select * from customer
select * from orders
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
select * from salesman
select * from customer
select * from orders
```

The query has been executed successfully, and the results are displayed in three separate result grids. The first grid shows the 'salesman' table with 5 rows. The second grid shows the 'customer' table with 6 rows. The third grid shows the 'orders' table with 7 rows.

salesman_id	name	city	commission
1	hiren	bhavnagar	12
2	manav	surat	13
3	harshesh	mahasan	10
4	ranu	bhavnagar	15
5	nevil	surat	20

customer_id	cust_name	city	grade	salesman_id
1	pk	bhavnagar	100	1
2	ram	surat	500	5
3	nanna	amwali	200	2
4	ramesh	mahasan	800	3
5	suresh	surat	1000	1
6	bhaves	bhavnagar	900	2

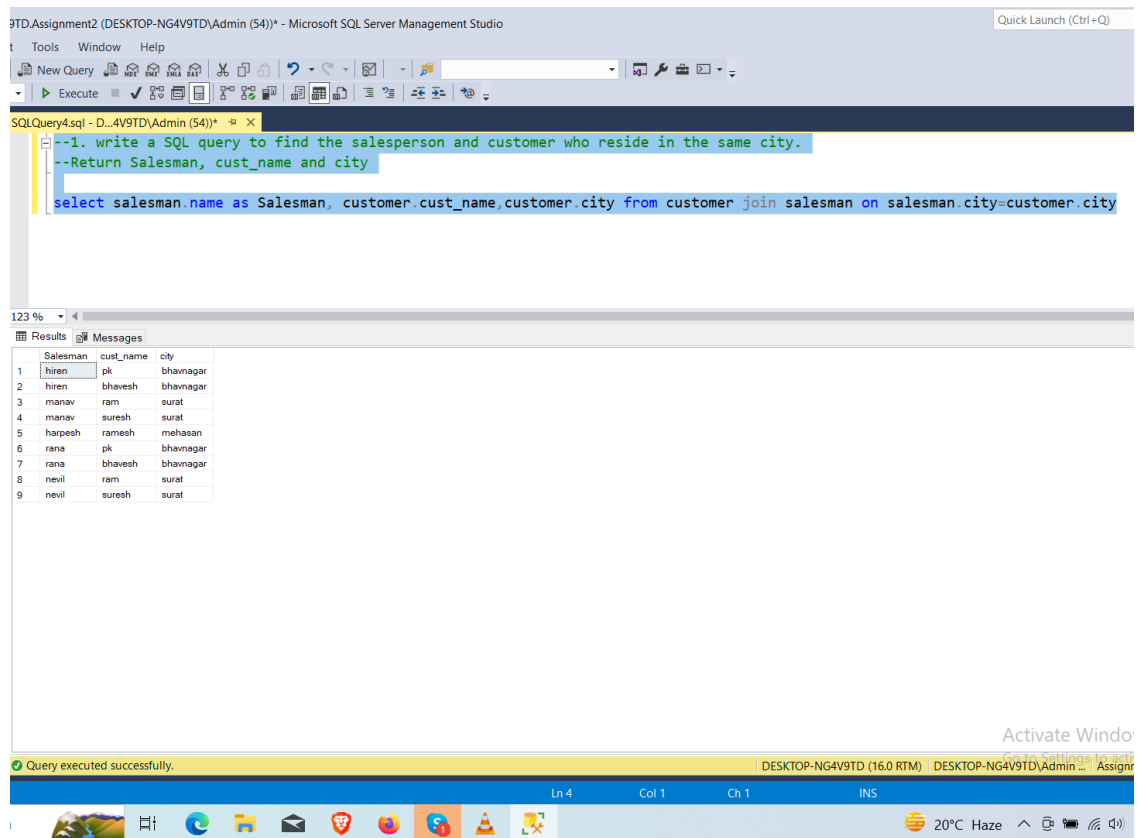
ord_no	purch_amt	ord_date	customer_id	salesman_id
1	500	2022-01-01 00:00:00.000	1	2
2	300	2022-05-02 00:00:00.000	2	2
3	1000	2023-02-05 00:00:00.000	3	1
4	800	2002-05-03 00:00:00.000	1	5
5	500	2021-02-01 00:00:00.000	5	5
6	2500	2021-03-03 00:00:00.000	6	3
7	3000	2022-05-02 00:00:00.000	4	4

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

1. write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city.

```
select salesman.name as Salesman,  
customer.cust_name, customer.city  
from customer join salesman  
on salesman.city=customer.city
```



The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a SQL query in a text editor. Below the editor, the 'Results' pane displays the output of the query as a table. The table has three columns: 'Salesman', 'cust_name', and 'city'. The data is as follows:

Salesman	cust_name	city
hiren	pk	bhavnagar
hiren	bhavesh	bhavnagar
manav	ram	surat
manav	suresh	surat
harpesh	ramesh	mehasan
rana	pk	bhavnagar
rana	bhavesh	bhavnagar
nevil	ram	surat
nevil	suresh	surat

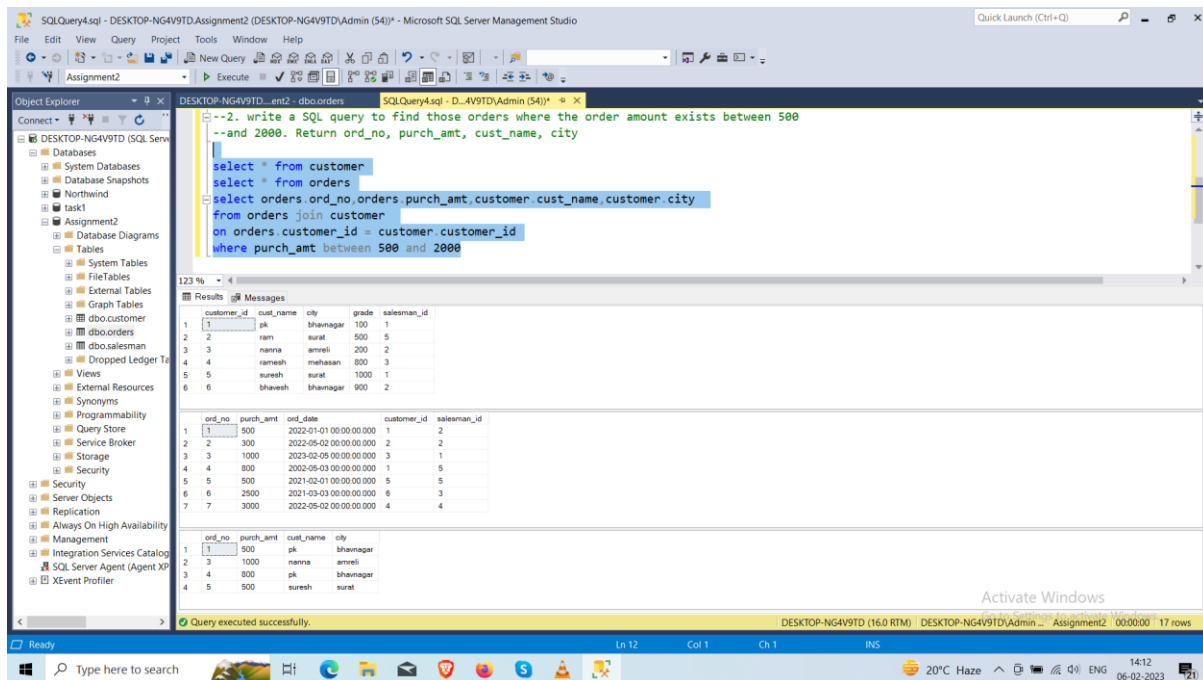
The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-NG4V9TD (16.0 RTM) | DESKTOP-NG4V9TD\Admin ... | Assignn'.

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

```
select * from customer
select * from orders
select
orders.ord_no,orders.purch_amt,customer.cust_name,customer.city
from orders join customer
on orders.customer_id = customer.customer_id
where purch_amt between 500 and 2000
```



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The SQL query editor displays the following query:

```
--2. write a SQL query to find those orders where the order amount exists between 500
--and 2000. Return ord_no, purch_amt, cust_name, city

select * from customer
select * from orders
select orders.ord_no,orders.purch_amt,customer.cust_name,customer.city
from orders join customer
on orders.customer_id = customer.customer_id
where purch_amt between 500 and 2000
```

The query results are displayed in three tables:

customer_id	cust_name	city	grade	salesman_id
1	pk	bhavanagar	100	1
2	ram	surat	500	5
3	nanna	amreli	200	2
4	ramesh	mehasan	800	3
5	suresh	surat	1000	1
6	bhawanagar	bhavanagar	900	2

ord_no	purch_amt	ord_date	customer_id	salesman_id
1	500	2022-01-01 00:00:00.000	1	2
2	300	2022-05-02 00:00:00.000	2	2
3	1000	2023-02-05 00:00:00.000	3	1
4	800	2002-05-03 00:00:00.000	1	5
5	500	2021-02-01 00:00:00.000	5	5
6	2500	2021-03-03 00:00:00.000	6	3
7	3000	2022-05-02 00:00:00.000	4	4

ord_no	purch_amt	cust_name	city
1	500	pk	bhavanagar
3	1000	nanna	amreli
4	800	pk	bhavanagar
5	500	suresh	surat

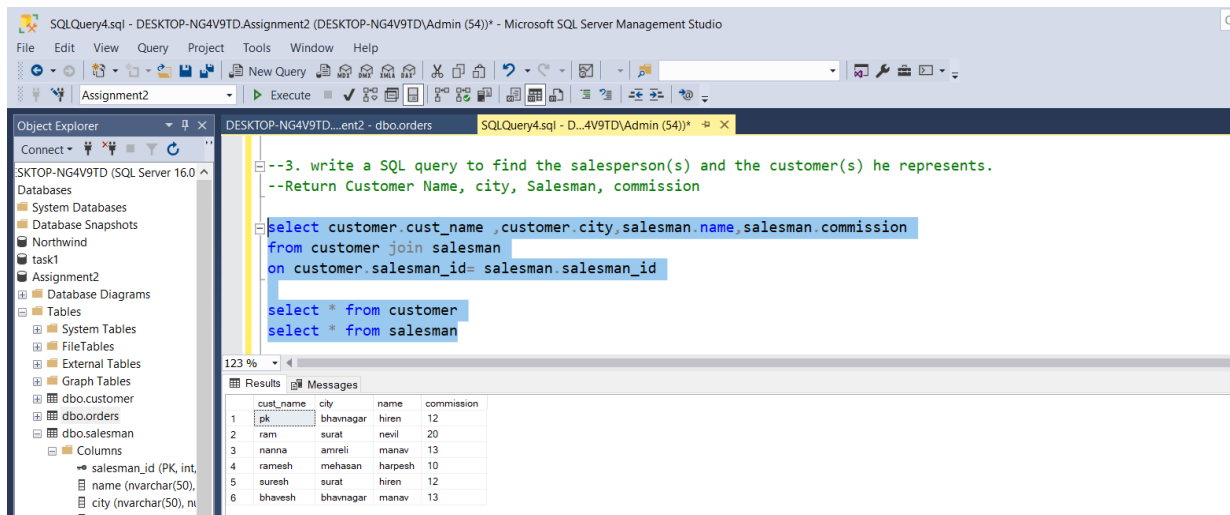
The status bar at the bottom indicates "Query executed successfully." and "17 rows".

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.

```
select customer.cust_name
,customer.city,salesman.name,salesman.commission
from customer join salesman
on customer.salesman_id= salesman.salesman_id
```



The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows the following SQL query:

```
--3. write a SQL query to find the salesperson(s) and the customer(s) he represents.
--Return Customer Name, city, Salesman, commission

select customer.cust_name ,customer.city,salesman.name,salesman.commission
from customer join salesman
on customer.salesman_id= salesman.salesman_id

select * from customer
select * from salesman
```

The query results are displayed in the Results pane, showing a table with the following data:

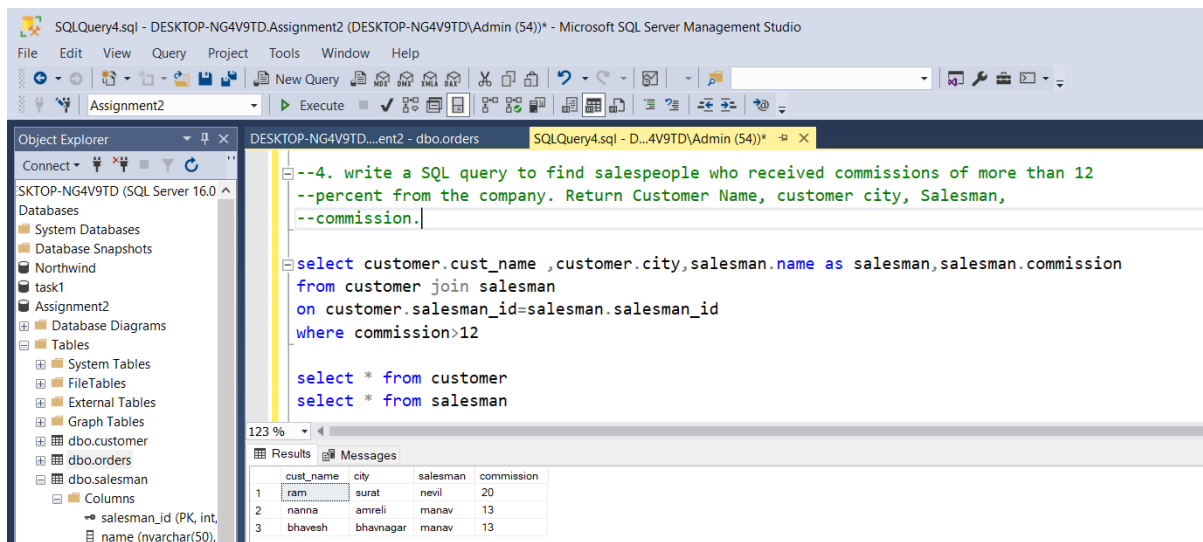
	cust_name	city	name	commission
1	pk	bhavnagar	hiren	12
2	ram	surat	nevil	20
3	nanna	amreli	manav	13
4	ramesh	mehasan	harshesh	10
5	suresh	surat	hiren	12
6	bhavesb	bhavnagar	manav	13

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman,commission.

```
select customer.cust_name ,customer.city,salesman.name as salesman,salesman.commission
from customer join salesman
on customer.salesman_id=salesman.salesman_id
where commission>12
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--4. write a SQL query to find salespeople who received commissions of more than 12
--percent from the company. Return Customer Name, customer city, Salesman,
--commission.

select customer.cust_name ,customer.city,salesman.name as salesman,salesman.commission
from customer join salesman
on customer.salesman_id=salesman.salesman_id
where commission>12

select * from customer
select * from salesman
```

The Results pane shows the output of the query:

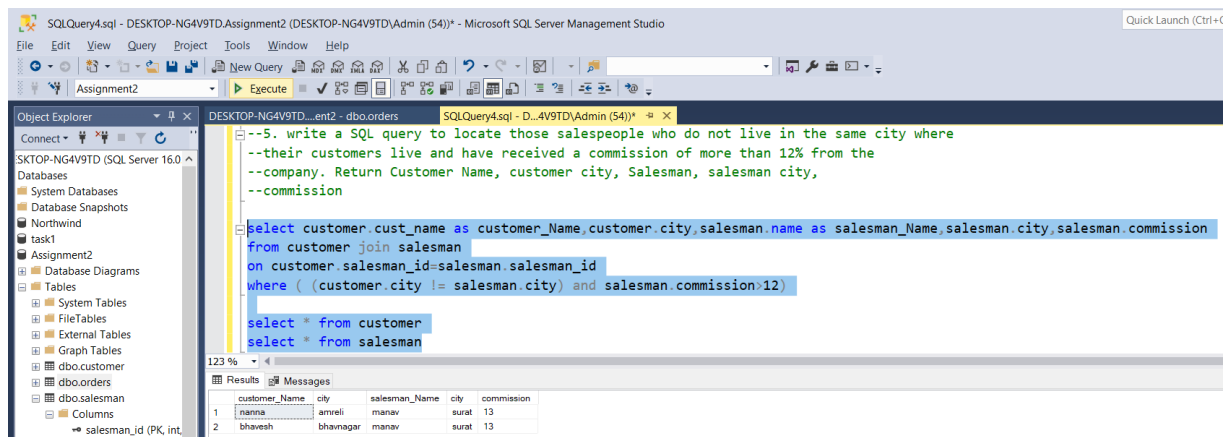
	cust_name	city	salesman	commission
1	ram	surat	nevil	20
2	nanna	amreli	manav	13
3	bhavesh	bhavnagar	manav	13

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

```
select customer.cust_name as  
customer_Name, customer.city, salesman.name as  
salesman_Name, salesman.city, salesman.commission  
from customer join salesman  
on customer.salesman_id=salesman.salesman_id  
where ( (customer.city != salesman.city) and  
salesman.commission>12)
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
--5. write a SQL query to locate those salespeople who do not live in the same city where  
--their customers live and have received a commission of more than 12% from the  
--company. Return Customer Name, customer city, Salesman, salesman city,  
--commission  
  
select customer.cust_name as customer_Name, customer.city, salesman.name as salesman_Name, salesman.city, salesman.commission  
from customer join salesman  
on customer.salesman_id=salesman.salesman_id  
where ( (customer.city != salesman.city) and salesman.commission>12)  
  
select * from customer  
select * from salesman
```

The Results pane shows the output of the query:

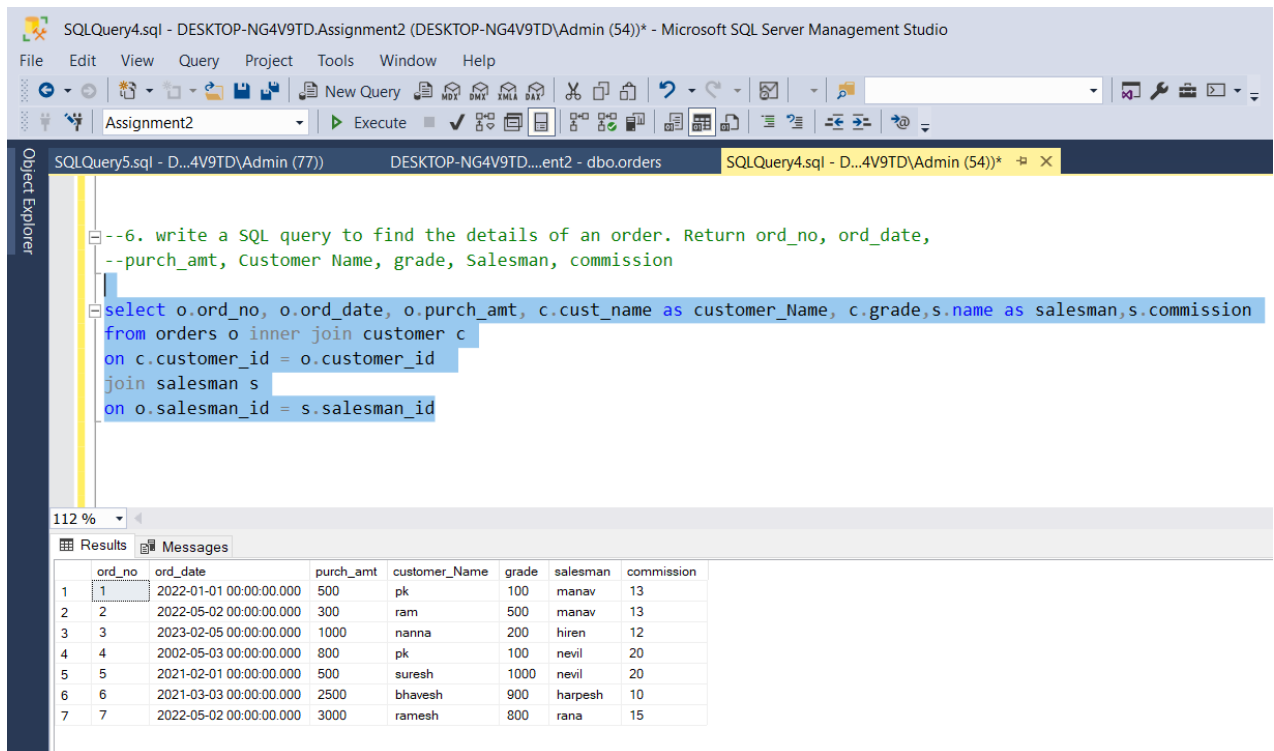
	customer_Name	city	salesman_Name	city	commission
1	nanna	amreli	manav	surat	13
2	bhavesh	bhavnagar	manav	surat	13

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

6. write a SQL query to find the details of an order. Return ord_no,ord_date,purch_amt, Customer Name, grade, Salesman, commission

```
select o.ord_no, o.ord_date, o.purch_amt, c.cust_name as
customer_Name, c.grade,s.name as salesman,s.commission
from orders o inner join customer c
on c.customer_id = o.customer_id
join salesman s
on o.salesman_id = s.salesman_id
```



The screenshot displays the Microsoft SQL Server Management Studio interface. The main query window shows a SQL query designed to retrieve order details by joining the orders, customer, and salesman tables. The query is as follows:

```
--6. write a SQL query to find the details of an order. Return ord_no, ord_date,
--purch_amt, Customer Name, grade, Salesman, commission

select o.ord_no, o.ord_date, o.purch_amt, c.cust_name as customer_Name, c.grade,s.name as salesman,s.commission
from orders o inner join customer c
on c.customer_id = o.customer_id
join salesman s
on o.salesman_id = s.salesman_id
```

Below the query editor, the 'Results' pane shows the output of the query, which consists of 7 rows of data. The columns are: ord_no, ord_date, purch_amt, customer_Name, grade, salesman, and commission.

ord_no	ord_date	purch_amt	customer_Name	grade	salesman	commission
1	2022-01-01 00:00:00.000	500	pk	100	manav	13
2	2022-05-02 00:00:00.000	300	ram	500	manav	13
3	2023-02-05 00:00:00.000	1000	nanna	200	hiren	12
4	2002-05-03 00:00:00.000	800	pk	100	nevil	20
5	2021-02-01 00:00:00.000	500	suresh	1000	nevil	20
6	2021-03-03 00:00:00.000	2500	bhavesh	900	harpesh	10
7	2022-05-02 00:00:00.000	3000	ramesh	800	rana	15

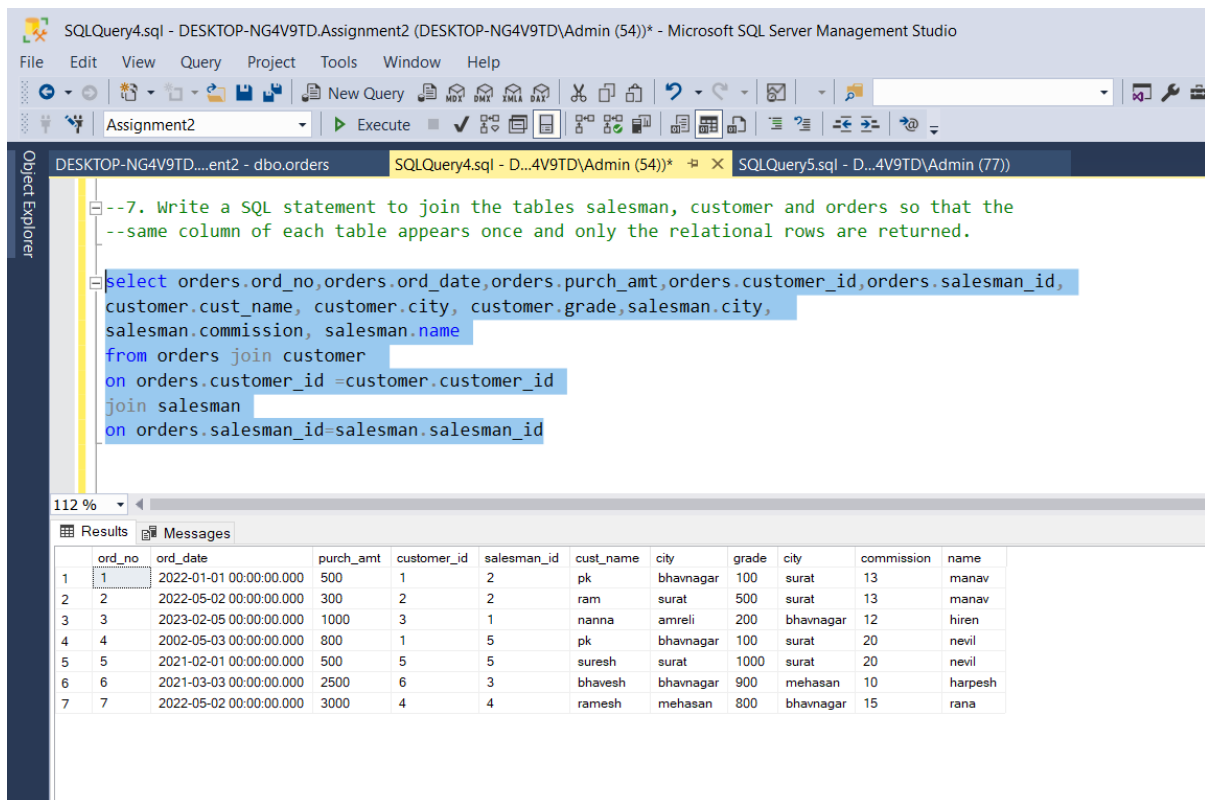
Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

Select

```
orders.ord_no,orders.ord_date,orders.purch_amt,orders.customer_id,orders.salesman_id,
customer.cust_name, customer.city, customer.grade,salesman.city,
salesman.commission, salesman.name
from orders join customer
on orders.customer_id =customer.customer_id
join salesman
on orders.salesman_id=salesman.salesman_id
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--7. Write a SQL statement to join the tables salesman, customer and orders so that the
--same column of each table appears once and only the relational rows are returned.

select orders.ord_no,orders.ord_date,orders.purch_amt,orders.customer_id,orders.salesman_id,
customer.cust_name, customer.city, customer.grade,salesman.city,
salesman.commission, salesman.name
from orders join customer
on orders.customer_id =customer.customer_id
join salesman
on orders.salesman_id=salesman.salesman_id
```

The Results pane shows the output of the query, displaying 7 rows of data with the following columns: ord_no, ord_date, purch_amt, customer_id, salesman_id, cust_name, city, grade, city, commission, and name.

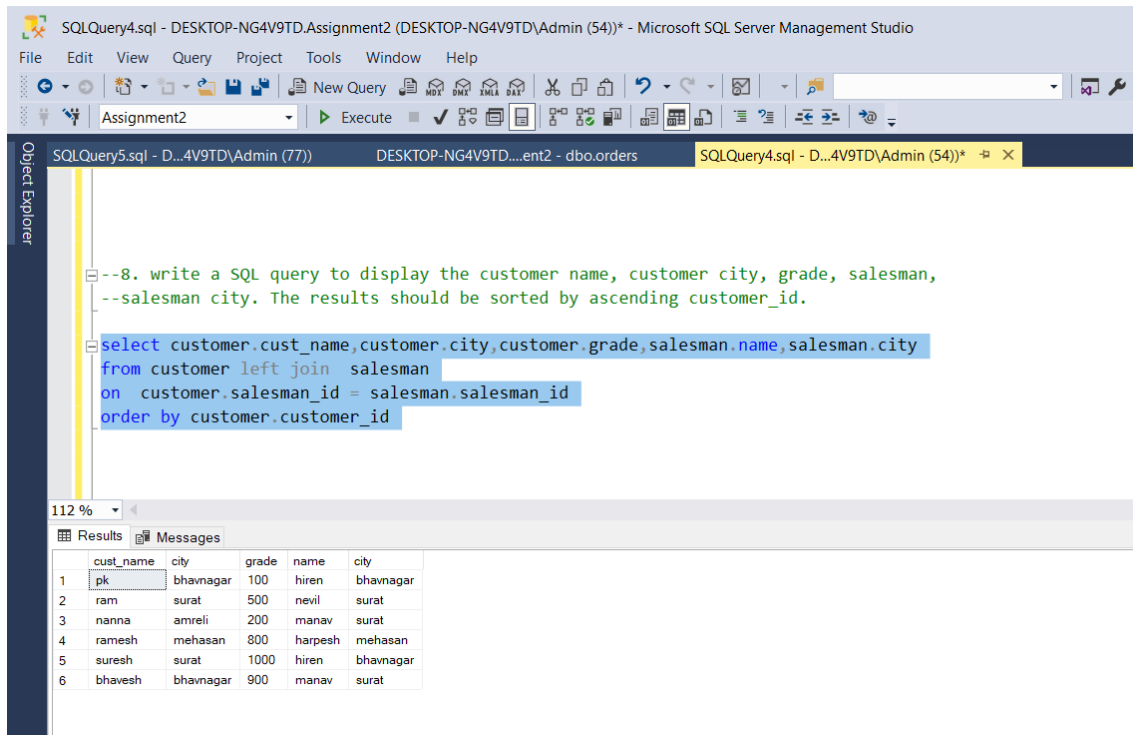
	ord_no	ord_date	purch_amt	customer_id	salesman_id	cust_name	city	grade	city	commission	name
1	1	2022-01-01 00:00:00.000	500	1	2	pk	bhavnagar	100	surat	13	manav
2	2	2022-05-02 00:00:00.000	300	2	2	ram	surat	500	surat	13	manav
3	3	2023-02-05 00:00:00.000	1000	3	1	nanna	amreli	200	bhavnagar	12	hiren
4	4	2002-05-03 00:00:00.000	800	1	5	pk	bhavnagar	100	surat	20	nevil
5	5	2021-02-01 00:00:00.000	500	5	5	suresh	surat	1000	surat	20	nevil
6	6	2021-03-03 00:00:00.000	2500	6	3	bhaves	bhavnagar	900	mehasan	10	harpesh
7	7	2022-05-02 00:00:00.000	3000	4	4	ramesh	mehasan	800	bhavnagar	15	rana

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

```
select
customer.cust_name,customer.city,customer.grade,salesman.name,salesman.city
from customer left join salesman
on customer.salesman_id = salesman.salesman_id
order by customer.customer_id
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
--8. write a SQL query to display the customer name, customer city, grade, salesman,
--salesman city. The results should be sorted by ascending customer_id.

select customer.cust_name,customer.city,customer.grade,salesman.name,salesman.city
from customer left join salesman
on customer.salesman_id = salesman.salesman_id
order by customer.customer_id
```

The Results pane at the bottom shows the output of the query, sorted by customer_id in ascending order. The results are as follows:

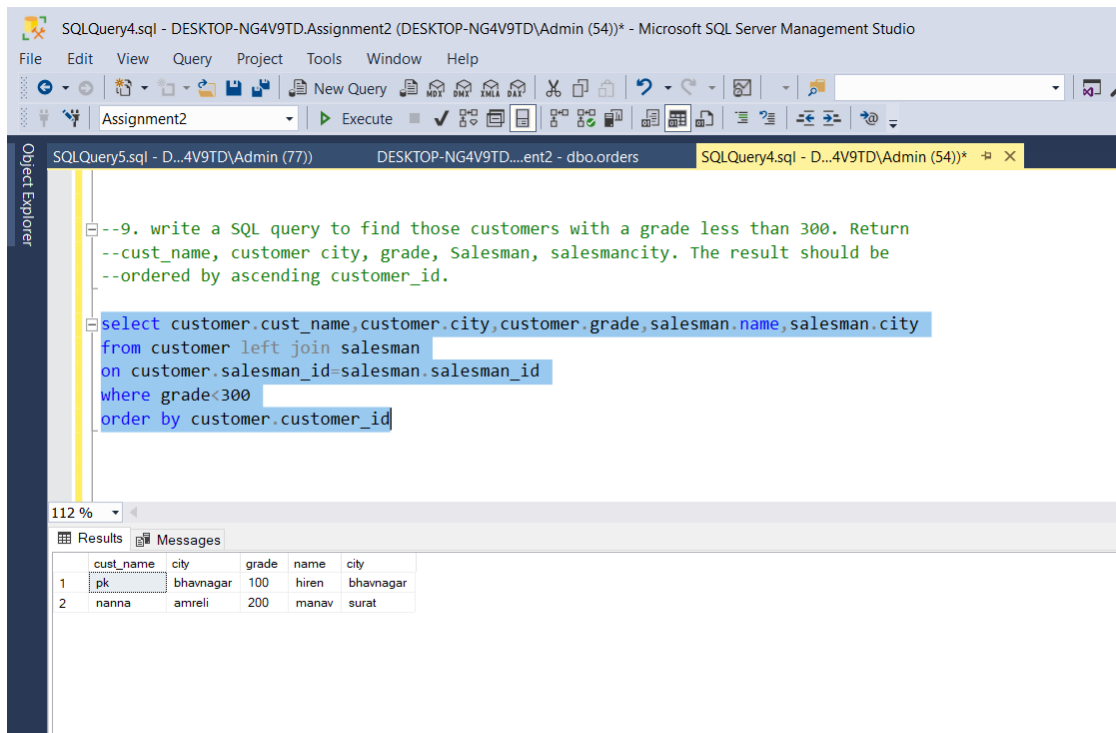
	cust_name	city	grade	name	city
1	pk	bhavnagar	100	hiren	bhavnagar
2	ram	surat	500	nevil	surat
3	nanna	amreli	200	manav	surat
4	ramesh	mehasan	800	harresh	mehasan
5	suresh	surat	1000	hiren	bhavnagar
6	bhavesh	bhavnagar	900	manav	surat

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

9. write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.

```
select customer.cust_name,customer.city,customer.grade,salesman.name,salesman.city
from customer left join salesman
on customer.salesman_id=salesman.salesman_id
where grade<300
order by customer.customer_id
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
--9. write a SQL query to find those customers with a grade less than 300. Return
--cust_name, customer city, grade, Salesman, salesmancity. The result should be
--ordered by ascending customer_id.

select customer.cust_name,customer.city,customer.grade,salesman.name,salesman.city
from customer left join salesman
on customer.salesman_id=salesman.salesman_id
where grade<300
order by customer.customer_id
```

The query is executed, and the results are displayed in the Results pane. The results show two rows of data:

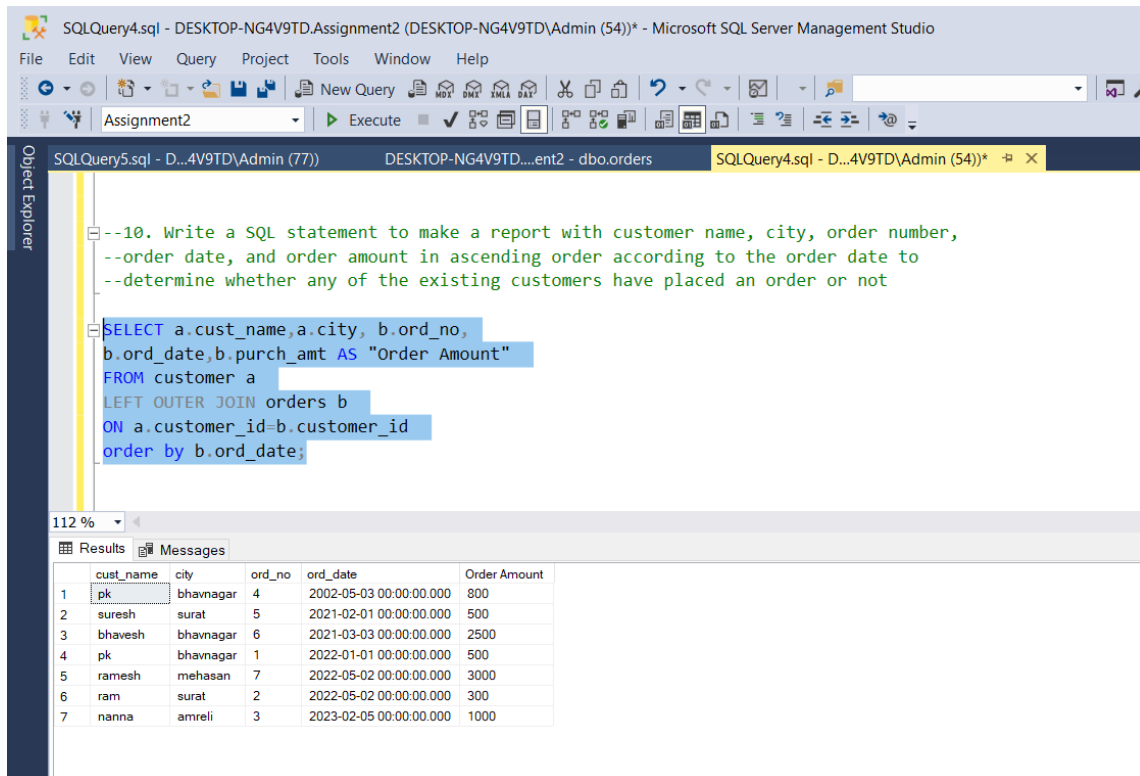
	cust_name	city	grade	name	city
1	pk	bhavnagar	100	hiren	bhavnagar
2	nanna	amreli	200	manav	surat

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not

```
SELECT a.cust_name, a.city, b.ord_no, b.ord_date, b.purch_amt AS  
"Order Amount"  
FROM customer a LEFT OUTER JOIN orders b  
ON a.customer_id=b.customer_id  
order by b.ord_date;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--10. Write a SQL statement to make a report with customer name, city, order number,  
--order date, and order amount in ascending order according to the order date to  
--determine whether any of the existing customers have placed an order or not  
  
SELECT a.cust_name, a.city, b.ord_no,  
b.ord_date, b.purch_amt AS "Order Amount"  
FROM customer a  
LEFT OUTER JOIN orders b  
ON a.customer_id=b.customer_id  
order by b.ord_date;
```

The Results pane shows the output of the query, displaying 7 rows of data. The columns are: cust_name, city, ord_no, ord_date, and Order Amount.

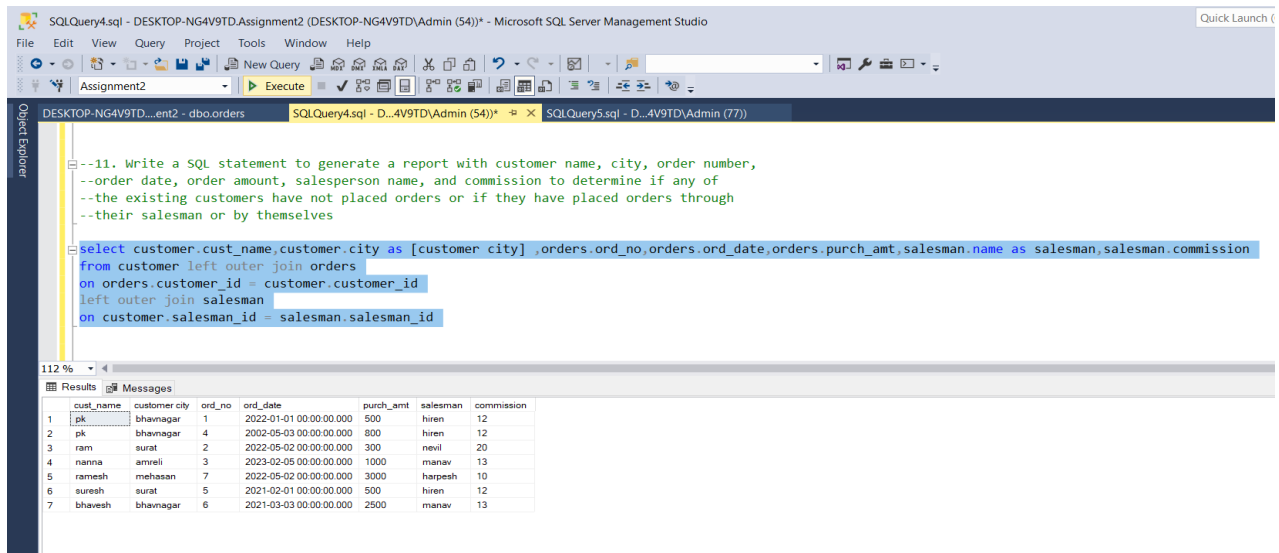
	cust_name	city	ord_no	ord_date	Order Amount
1	pk	bhavnagar	4	2002-05-03 00:00:00.000	800
2	suresh	surat	5	2021-02-01 00:00:00.000	500
3	bhavesh	bhavnagar	6	2021-03-03 00:00:00.000	2500
4	pk	bhavnagar	1	2022-01-01 00:00:00.000	500
5	ramesh	mehasan	7	2022-05-02 00:00:00.000	3000
6	ram	surat	2	2022-05-02 00:00:00.000	300
7	nanna	amreli	3	2023-02-05 00:00:00.000	1000

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

11. Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

```
select customer.cust_name, customer.city as [customer city], orders.ord_no, orders.ord_date, orders.purch_amt, salesman.name as salesman, salesman.commission
from customer left outer join orders
on orders.customer_id = customer.customer_id
left outer join salesman
on customer.salesman_id = salesman.salesman_id
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--11. Write a SQL statement to generate a report with customer name, city, order number,
--order date, order amount, salesperson name, and commission to determine if any of
--the existing customers have not placed orders or if they have placed orders through
--their salesman or by themselves

select customer.cust_name, customer.city as [customer city], orders.ord_no, orders.ord_date, orders.purch_amt, salesman.name as salesman, salesman.commission
from customer left outer join orders
on orders.customer_id = customer.customer_id
left outer join salesman
on customer.salesman_id = salesman.salesman_id
```

The Results pane shows the output of the query, which is a table with 7 columns: cust_name, customer city, ord_no, ord_date, purch_amt, salesman, and commission. The data is as follows:

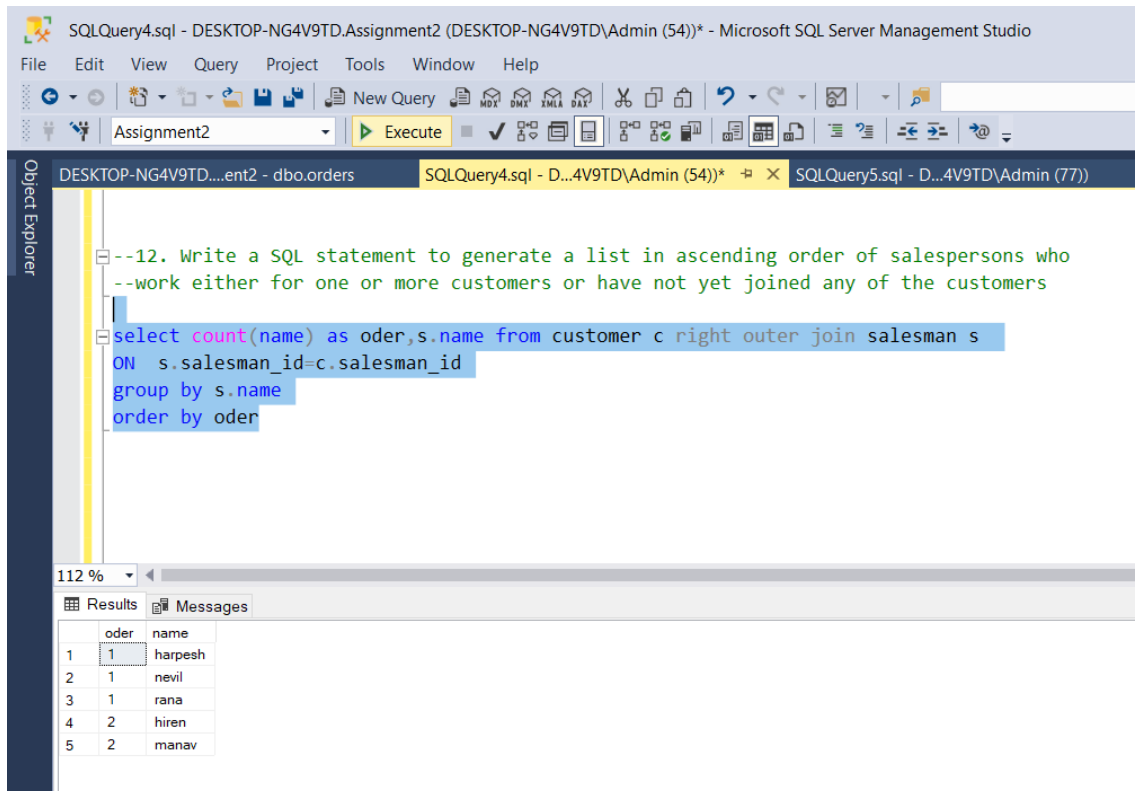
	cust_name	customer city	ord_no	ord_date	purch_amt	salesman	commission
1	pk	bhavnagar	1	2022-01-01 00:00:00.000	500	hiren	12
2	pk	bhavnagar	4	2022-05-03 00:00:00.000	800	hiren	12
3	ram	surat	2	2022-05-02 00:00:00.000	300	nevil	20
4	nanna	amreli	3	2022-02-05 00:00:00.000	1000	manav	13
5	ramesh	mehasan	7	2022-05-02 00:00:00.000	3000	harresh	10
6	suresh	surat	5	2021-02-01 00:00:00.000	500	hiren	12
7	bhavesh	bhavnagar	6	2021-03-03 00:00:00.000	2500	manav	13

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers

```
select count(name) as oder,s.name from customer c right outer join  
salesman s  
ON s.salesman_id=c.salesman_id  
group by s.name  
order by oder
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--12. Write a SQL statement to generate a list in ascending order of salespersons who  
--work either for one or more customers or have not yet joined any of the customers  
  
select count(name) as oder,s.name from customer c right outer join salesman s  
ON s.salesman_id=c.salesman_id  
group by s.name  
order by oder
```

The query has been executed, and the results are displayed in the Results pane. The results show a list of salespersons and the count of customers they are associated with (labeled as 'oder').

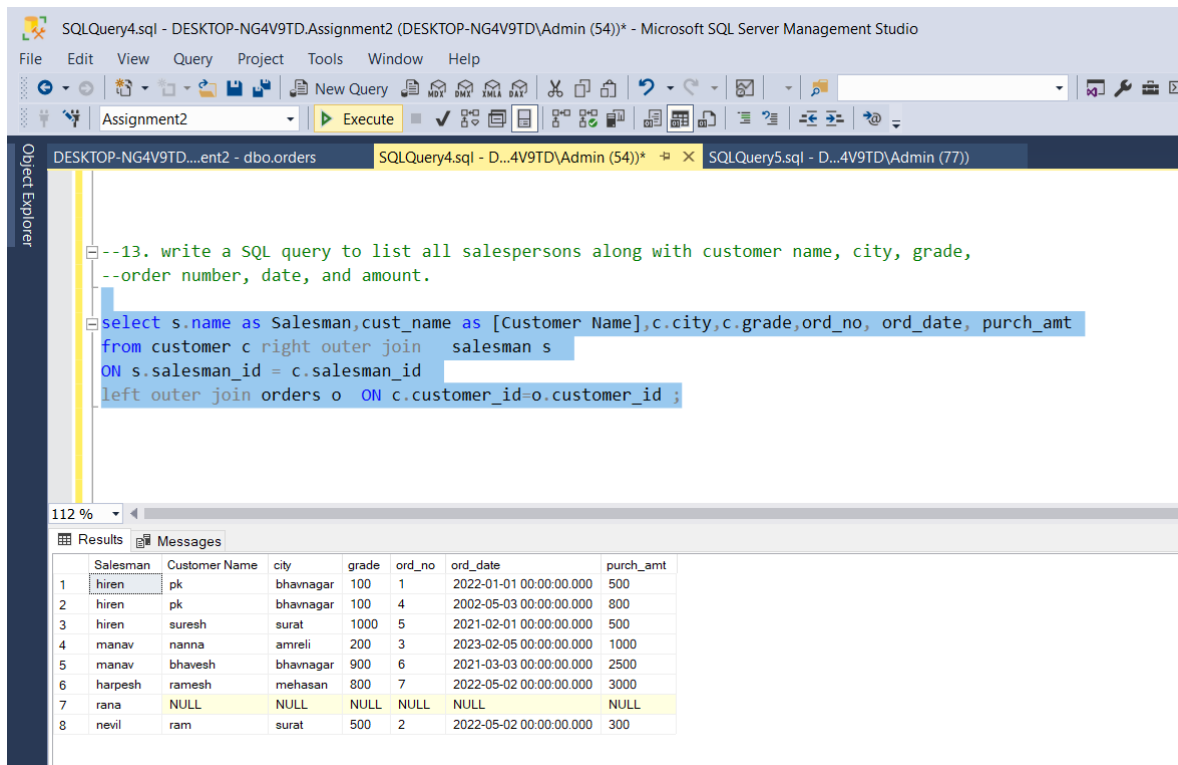
	oder	name
1	1	harpesh
2	1	nevil
3	1	rana
4	2	hiren
5	2	manav

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

```
select s.name as Salesman,cust_name as [Customer Name],c.city,c.grade,ord_no, ord_date, purch_amt
from customer c right outer join  salesman s
ON s.salesman_id = c.salesman_id
left outer join orders o ON c.customer_id=o.customer_id ;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
--13. write a SQL query to list all salespersons along with customer name, city, grade,
--order number, date, and amount.
select s.name as Salesman,cust_name as [Customer Name],c.city,c.grade,ord_no, ord_date, purch_amt
from customer c right outer join  salesman s
ON s.salesman_id = c.salesman_id
left outer join orders o ON c.customer_id=o.customer_id ;
```

The Results pane at the bottom shows the output of the query as a table with 8 rows and 7 columns:

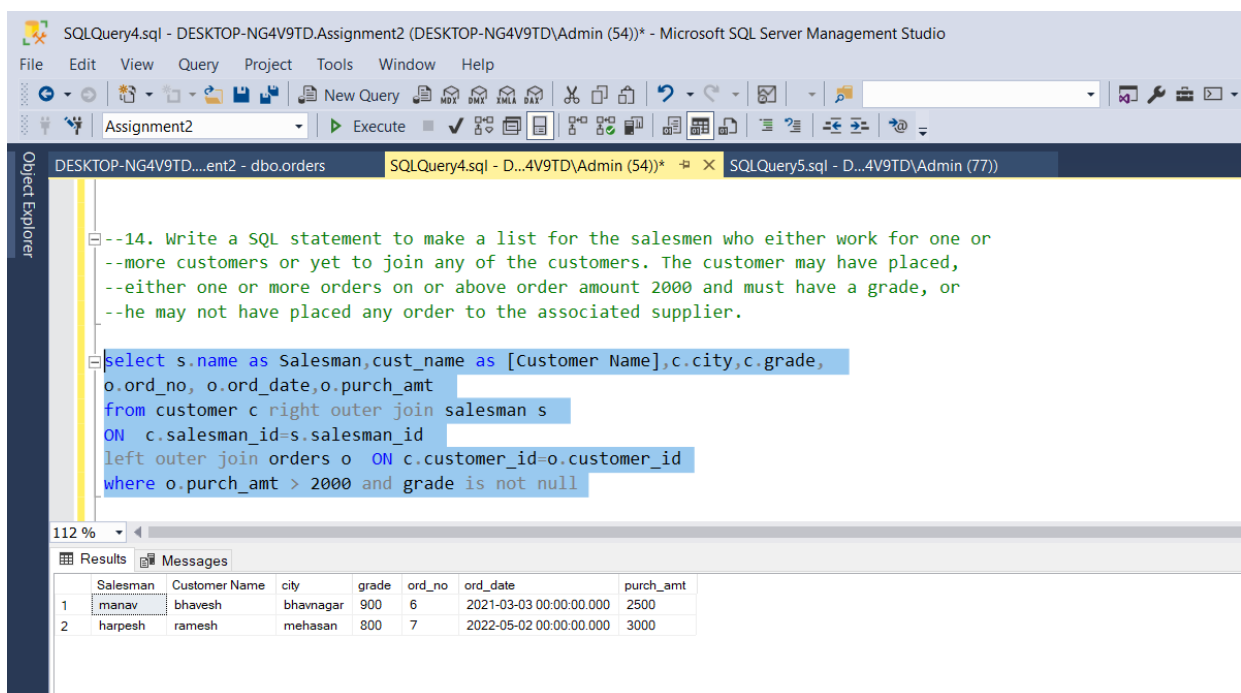
	Salesman	Customer Name	city	grade	ord_no	ord_date	purch_amt
1	hiren	pk	bhavnagar	100	1	2022-01-01 00:00:00.000	500
2	hiren	pk	bhavnagar	100	4	2002-05-03 00:00:00.000	800
3	hiren	suresh	surat	1000	5	2021-02-01 00:00:00.000	500
4	manav	nanna	amreli	200	3	2023-02-05 00:00:00.000	1000
5	manav	bhaves	bhavnagar	900	6	2021-03-03 00:00:00.000	2500
6	harpesh	ramesh	mehasan	800	7	2022-05-02 00:00:00.000	3000
7	rana	NULL	NULL	NULL	NULL	NULL	NULL
8	nevil	ram	surat	500	2	2022-05-02 00:00:00.000	300

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

```
select s.name as Salesman,cust_name as [Customer
Name],c.city,c.grade,
o.ord_no, o.ord_date,o.purch_amt
from customer c right outer join salesman s
ON c.salesman_id=s.salesman_id
left outer join orders o ON c.customer_id=o.customer_id
where o.purch_amt > 2000 and grade is not null
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the SQL statement for problem 14. The query is executed, and the results are shown in the Results pane at the bottom. The results table has 7 columns: Salesman, Customer Name, city, grade, ord_no, ord_date, and purch_amt. There are 2 rows of data.

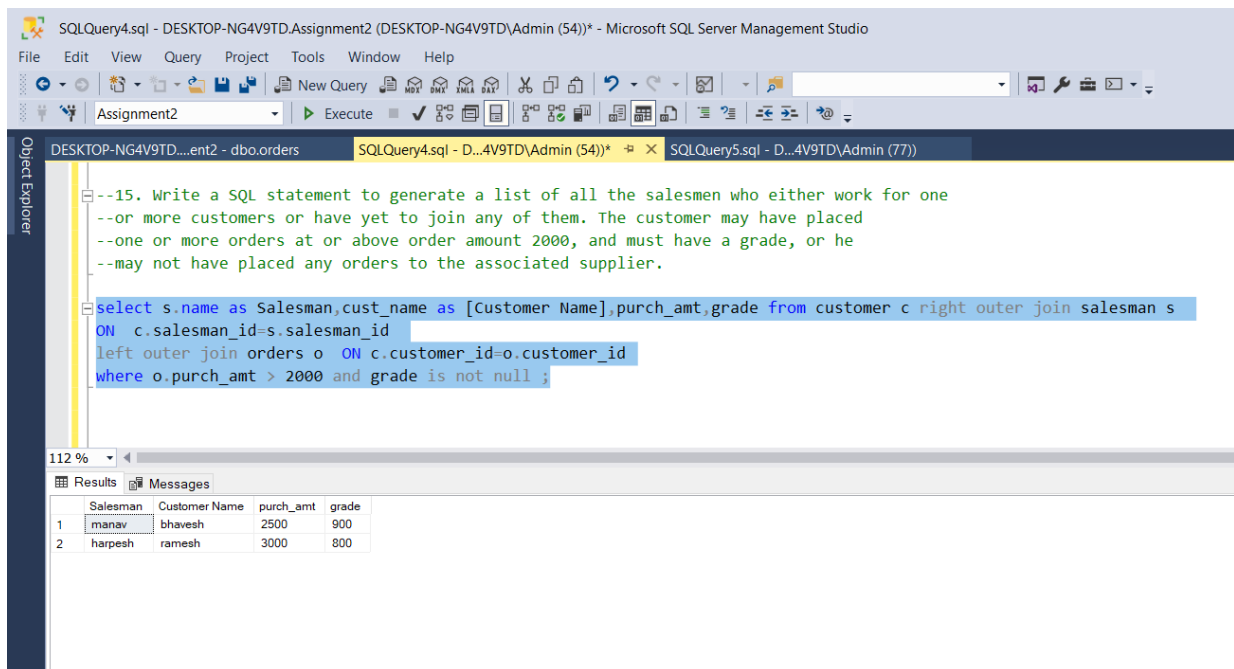
	Salesman	Customer Name	city	grade	ord_no	ord_date	purch_amt
1	manav	bhavesh	bhavnagar	900	6	2021-03-03 00:00:00.000	2500
2	harresh	ramesh	mehasan	800	7	2022-05-02 00:00:00.000	3000

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

15. Write a SQL statement to generate a list of all the salesmen who either work for one or more customers or have yet to join any of them. The customer may have placed one or more orders at or above order amount 2000, and must have a grade, or he may not have placed any orders to the associated supplier.

```
select s.name as Salesman,cust_name as  
[CustomerName],purch_amt,grade from customer c right outer join  
salesman s  
ON c.salesman_id=s.salesman_id  
left outer join orders o ON c.customer_id=o.customer_id  
where o.purch_amt > 2000 and grade is not null ;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--15. Write a SQL statement to generate a list of all the salesmen who either work for one  
--or more customers or have yet to join any of them. The customer may have placed  
--one or more orders at or above order amount 2000, and must have a grade, or he  
--may not have placed any orders to the associated supplier.  
  
select s.name as Salesman,cust_name as [Customer Name],purch_amt,grade from customer c right outer join salesman s  
ON c.salesman_id=s.salesman_id  
left outer join orders o ON c.customer_id=o.customer_id  
where o.purch_amt > 2000 and grade is not null ;
```

The query results are displayed in the Results pane, showing a table with the following data:

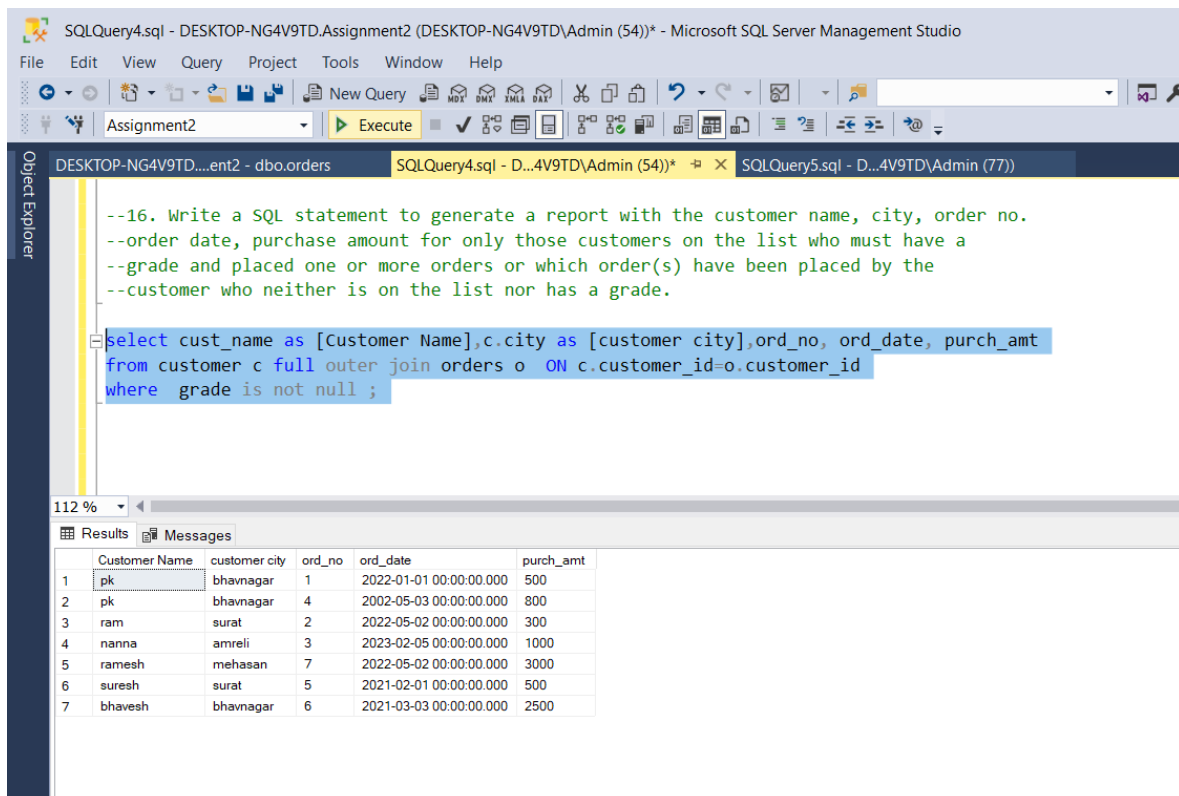
	Salesman	Customer Name	purch_amt	grade
1	manav	bhavesh	2500	900
2	harresh	ramesh	3000	800

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

```
select cust_name as [Customer Name],c.city as [customer  
city],ord_no, ord_date, purch_amt  
from customer c full outer join orders o  
ON c.customer_id=o.customer_id  
where grade is not null ;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--16. Write a SQL statement to generate a report with the customer name, city, order no.  
--order date, purchase amount for only those customers on the list who must have a  
--grade and placed one or more orders or which order(s) have been placed by the  
--customer who neither is on the list nor has a grade.  
  
select cust_name as [Customer Name],c.city as [customer city],ord_no, ord_date, purch_amt  
from customer c full outer join orders o ON c.customer_id=o.customer_id  
where grade is not null ;
```

The query results are displayed in a table with the following columns: Customer Name, customer city, ord_no, ord_date, and purch_amt. The results show 7 rows of data.

	Customer Name	customer city	ord_no	ord_date	purch_amt
1	pk	bhavnagar	1	2022-01-01 00:00:00.000	500
2	pk	bhavnagar	4	2002-05-03 00:00:00.000	800
3	ram	surat	2	2022-05-02 00:00:00.000	300
4	nanna	amreli	3	2023-02-05 00:00:00.000	1000
5	ramesh	mehasan	7	2022-05-02 00:00:00.000	3000
6	suresh	surat	5	2021-02-01 00:00:00.000	500
7	bhavesh	bhavnagar	6	2021-03-03 00:00:00.000	2500

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

17. Write a SQL query to combine each row of the salesman table with each row of the customer table

```
select * from customer cross join salesman
```

The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays the query: `--17. Write a SQL query to combine each row of the salesman table with each row of the customer table` and `select * from customer cross join salesman`. The bottom pane shows the results of the query, which is a cross join of the customer and salesman tables. The results table has 10 columns: customer_id, cust_name, city, grade, salesman_id, salesman_id, name, city, and commission. The data is organized into 30 rows, representing all possible combinations of the 6 customers and 5 salesmen.

	customer_id	cust_name	city	grade	salesman_id	salesman_id	name	city	commission
1	1	pk	bhavnagar	100	1	1	hiren	bhavnagar	12
2	2	ram	surat	500	5	1	hiren	bhavnagar	12
3	3	nanna	amreli	200	2	1	hiren	bhavnagar	12
4	4	ramesh	mehasan	800	3	1	hiren	bhavnagar	12
5	5	suresh	surat	1000	1	1	hiren	bhavnagar	12
6	6	bhaves	bhavnagar	900	2	1	hiren	bhavnagar	12
7	1	pk	bhavnagar	100	1	2	manav	surat	13
8	2	ram	surat	500	5	2	manav	surat	13
9	3	nanna	amreli	200	2	2	manav	surat	13
10	4	ramesh	mehasan	800	3	2	manav	surat	13
11	5	suresh	surat	1000	1	2	manav	surat	13
12	6	bhaves	bhavnagar	900	2	2	manav	surat	13
13	1	pk	bhavnagar	100	1	3	harpe	mehasan	10
14	2	ram	surat	500	5	3	harpe	mehasan	10
15	3	nanna	amreli	200	2	3	harpe	mehasan	10
16	4	ramesh	mehasan	800	3	3	harpe	mehasan	10
17	5	suresh	surat	1000	1	3	harpe	mehasan	10
18	6	bhaves	bhavnagar	900	2	3	harpe	mehasan	10
19	1	pk	bhavnagar	100	1	4	rana	bhavnagar	15
20	2	ram	surat	500	5	4	rana	bhavnagar	15
21	3	nanna	amreli	200	2	4	rana	bhavnagar	15
22	4	ramesh	mehasan	800	3	4	rana	bhavnagar	15
23	5	suresh	surat	1000	1	4	rana	bhavnagar	15
24	6	bhaves	bhavnagar	900	2	4	rana	bhavnagar	15
25	1	pk	bhavnagar	100	1	5	nevil	surat	20
26	2	ram	surat	500	5	5	nevil	surat	20
27	3	nanna	amreli	200	2	5	nevil	surat	20
28	4	ramesh	mehasan	800	3	5	nevil	surat	20
29	5	suresh	surat	1000	1	5	nevil	surat	20
30	6	bhaves	bhavnagar	900	2	5	nevil	surat	20

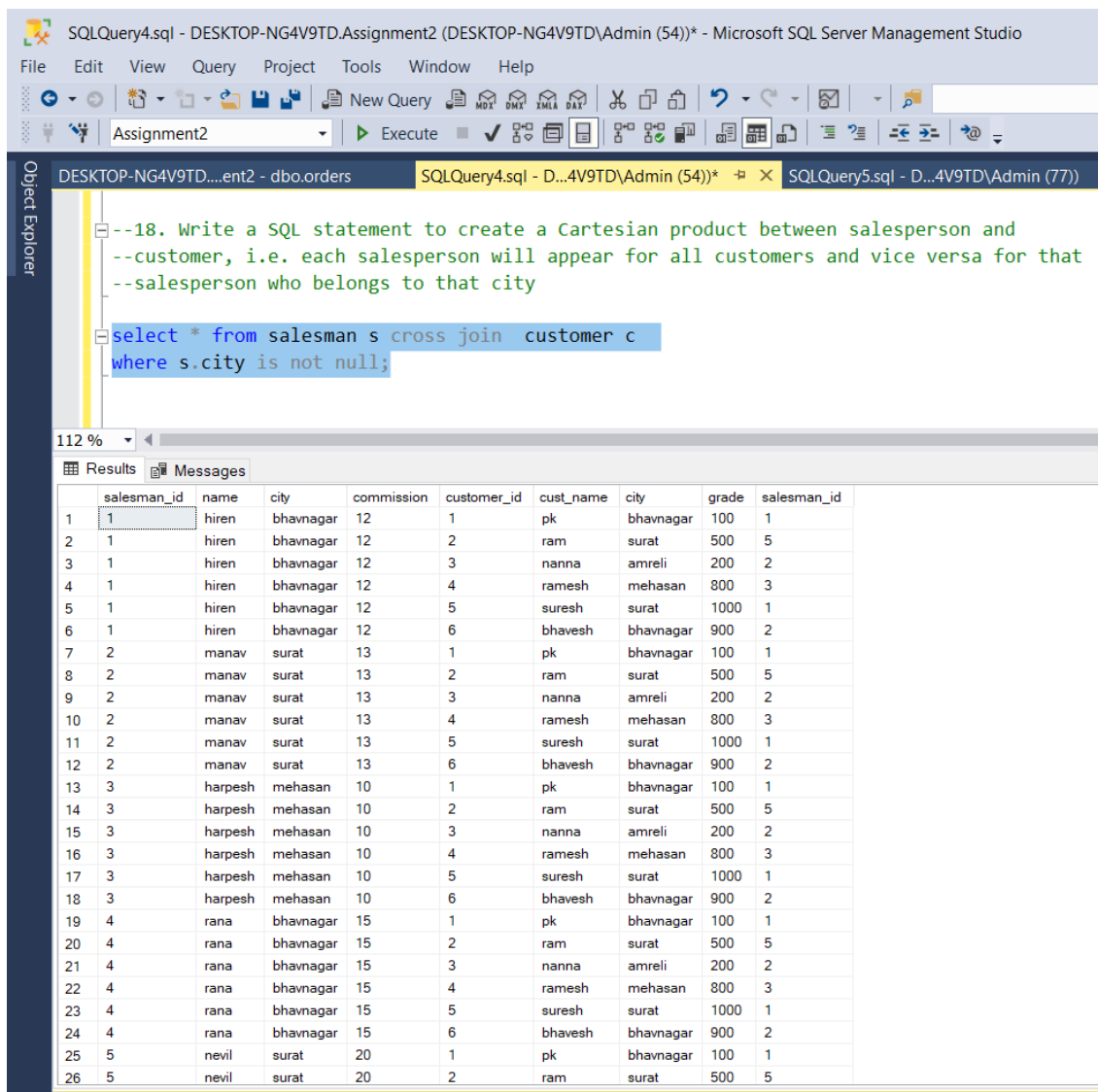
Query executed successfully

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

```
select * from salesman s cross join customer c
where s.city is not null;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--18. Write a SQL statement to create a Cartesian product between salesperson and
--customer, i.e. each salesperson will appear for all customers and vice versa for that
--salesperson who belongs to that city

select * from salesman s cross join customer c
where s.city is not null;
```

The Results pane shows the output of the query, which is a Cartesian product of the salesperson and customer tables, filtered by the condition where the salesperson's city is not null. The results are displayed in a table with the following columns: salesman_id, name, city, commission, customer_id, cust_name, city, grade, and salesman_id. The results are sorted by salesman_id.

salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	hiren	bhavnagar	12	1	pk	bhavnagar	100	1
2	hiren	bhavnagar	12	2	ram	surat	500	5
3	hiren	bhavnagar	12	3	nanna	amreli	200	2
4	hiren	bhavnagar	12	4	ramesh	mehasan	800	3
5	hiren	bhavnagar	12	5	suresh	surat	1000	1
6	hiren	bhavnagar	12	6	bhaves	bhavnagar	900	2
7	manav	surat	13	1	pk	bhavnagar	100	1
8	manav	surat	13	2	ram	surat	500	5
9	manav	surat	13	3	nanna	amreli	200	2
10	manav	surat	13	4	ramesh	mehasan	800	3
11	manav	surat	13	5	suresh	surat	1000	1
12	manav	surat	13	6	bhaves	bhavnagar	900	2
13	harpesh	mehasan	10	1	pk	bhavnagar	100	1
14	harpesh	mehasan	10	2	ram	surat	500	5
15	harpesh	mehasan	10	3	nanna	amreli	200	2
16	harpesh	mehasan	10	4	ramesh	mehasan	800	3
17	harpesh	mehasan	10	5	suresh	surat	1000	1
18	harpesh	mehasan	10	6	bhaves	bhavnagar	900	2
19	rana	bhavnagar	15	1	pk	bhavnagar	100	1
20	rana	bhavnagar	15	2	ram	surat	500	5
21	rana	bhavnagar	15	3	nanna	amreli	200	2
22	rana	bhavnagar	15	4	ramesh	mehasan	800	3
23	rana	bhavnagar	15	5	suresh	surat	1000	1
24	rana	bhavnagar	15	6	bhaves	bhavnagar	900	2
25	nevil	surat	20	1	pk	bhavnagar	100	1
26	nevil	surat	20	2	ram	surat	500	5

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade

```
select * from salesman s cross join customer c
where s.city is not null and c.grade is not null;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'SQLQuery4.sql - DESKTOP-NG4V9TD.Assignment2 (DESKTOP-NG4V9TD\Admin (54))* - Microsoft SQL Server Management Studio'. The menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains various icons for file operations, query execution, and formatting. The 'Object Explorer' on the left shows the database structure, including 'DESKTOP-NG4V9TD...ent2 - dbo.orders'. The 'Query Editor' displays the following SQL query:

```
--19. Write a SQL statement to create a Cartesian product between salesperson and
--customer, i.e. each salesperson will appear for every customer and vice versa for
--those salesmen who belong to a city and customers who require a grade

select * from salesman s cross join customer c
where s.city is not null and c.grade is not null;
```

The 'Results' tab shows the output of the query, which is a Cartesian product of the 'salesman' and 'customer' tables. The results are displayed in a table with 10 columns: salesman_id, name, city, commission, customer_id, cust_name, city, grade, and salesman_id. The table contains 26 rows of data, representing all combinations of salesmen and customers where both have non-null city and grade values.

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	1	hiren	bhavnagar	12	1	pk	bhavnagar	100	1
2	1	hiren	bhavnagar	12	2	ram	surat	500	5
3	1	hiren	bhavnagar	12	3	nanna	amreli	200	2
4	1	hiren	bhavnagar	12	4	ramesh	mehasan	800	3
5	1	hiren	bhavnagar	12	5	suresh	surat	1000	1
6	1	hiren	bhavnagar	12	6	bhaves	bhavnagar	900	2
7	2	manav	surat	13	1	pk	bhavnagar	100	1
8	2	manav	surat	13	2	ram	surat	500	5
9	2	manav	surat	13	3	nanna	amreli	200	2
10	2	manav	surat	13	4	ramesh	mehasan	800	3
11	2	manav	surat	13	5	suresh	surat	1000	1
12	2	manav	surat	13	6	bhaves	bhavnagar	900	2
13	3	harpesh	mehasan	10	1	pk	bhavnagar	100	1
14	3	harpesh	mehasan	10	2	ram	surat	500	5
15	3	harpesh	mehasan	10	3	nanna	amreli	200	2
16	3	harpesh	mehasan	10	4	ramesh	mehasan	800	3
17	3	harpesh	mehasan	10	5	suresh	surat	1000	1
18	3	harpesh	mehasan	10	6	bhaves	bhavnagar	900	2
19	4	rana	bhavnagar	15	1	pk	bhavnagar	100	1
20	4	rana	bhavnagar	15	2	ram	surat	500	5
21	4	rana	bhavnagar	15	3	nanna	amreli	200	2
22	4	rana	bhavnagar	15	4	ramesh	mehasan	800	3
23	4	rana	bhavnagar	15	5	suresh	surat	1000	1
24	4	rana	bhavnagar	15	6	bhaves	bhavnagar	900	2
25	5	nevil	surat	20	1	pk	bhavnagar	100	1
26	5	nevil	surat	20	2	ram	surat	500	5

The status bar at the bottom indicates 'Query executed successfully.'

Assignment:- 2

RETRIEVE DATA USING JOIN WITH WHERE CLAUSE

20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa for those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade

```
select * from salesman s cross join customer c
where s.city !=c.city and c.grade is not null and s.city is not null;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
--20. Write a SQL statement to make a Cartesian product between salesman and
--customer i.e. each salesman will appear for all customers and vice versa for those
--salesmen who must belong to a city which is not the same as his customer and the
--customers should have their own grade

select * from salesman s cross join customer c
where s.city !=c.city and c.grade is not null and s.city is not null;
```

The Results pane shows the output of the query, which is a Cartesian product of salesmen and customers where the city is different and the customer has a grade. The results are as follows:

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	1	hiren	bhavnagar	12	2	ram	surat	500	5
2	1	hiren	bhavnagar	12	3	nanna	amreli	200	2
3	1	hiren	bhavnagar	12	4	ramesh	mehasan	800	3
4	1	hiren	bhavnagar	12	5	suresh	surat	1000	1
5	2	manav	surat	13	1	pk	bhavnagar	100	1
6	2	manav	surat	13	3	nanna	amreli	200	2
7	2	manav	surat	13	4	ramesh	mehasan	800	3
8	2	manav	surat	13	6	bhaves	bhavnagar	900	2
9	3	harpesh	mehasan	10	1	pk	bhavnagar	100	1
10	3	harpesh	mehasan	10	2	ram	surat	500	5
11	3	harpesh	mehasan	10	3	nanna	amreli	200	2
12	3	harpesh	mehasan	10	5	suresh	surat	1000	1
13	3	harpesh	mehasan	10	6	bhaves	bhavnagar	900	2
14	4	rana	bhavnagar	15	2	ram	surat	500	5
15	4	rana	bhavnagar	15	3	nanna	amreli	200	2
16	4	rana	bhavnagar	15	4	ramesh	mehasan	800	3
17	4	rana	bhavnagar	15	5	suresh	surat	1000	1
18	5	nevil	surat	20	1	pk	bhavnagar	100	1
19	5	nevil	surat	20	3	nanna	amreli	200	2
20	5	nevil	surat	20	4	ramesh	mehasan	800	3
21	5	nevil	surat	20	6	bhaves	bhavnagar	900	2