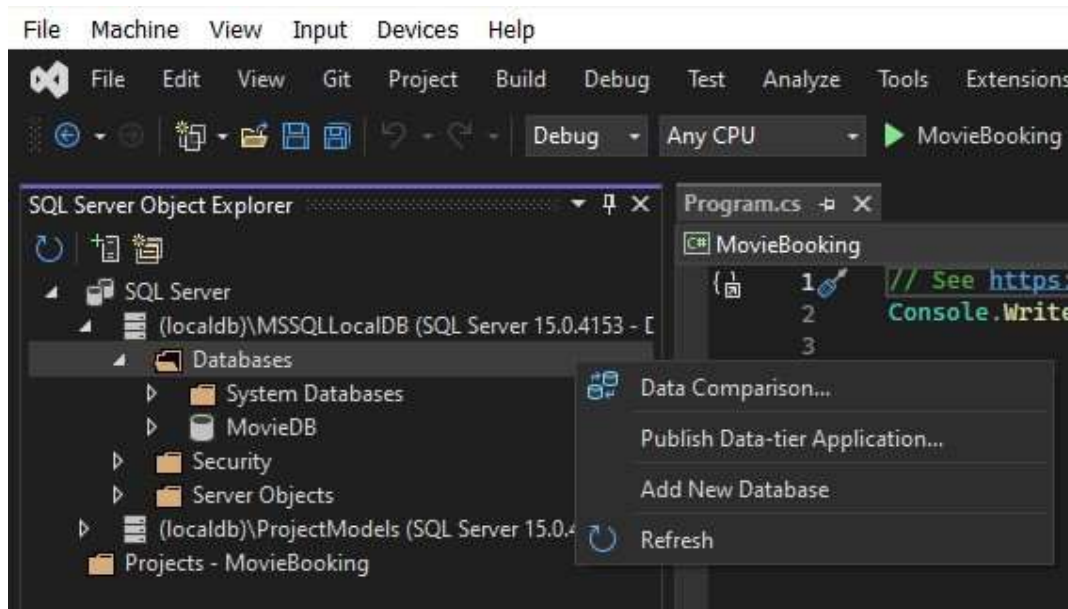


## TASK

Open visual studio 2022 in the vm/system

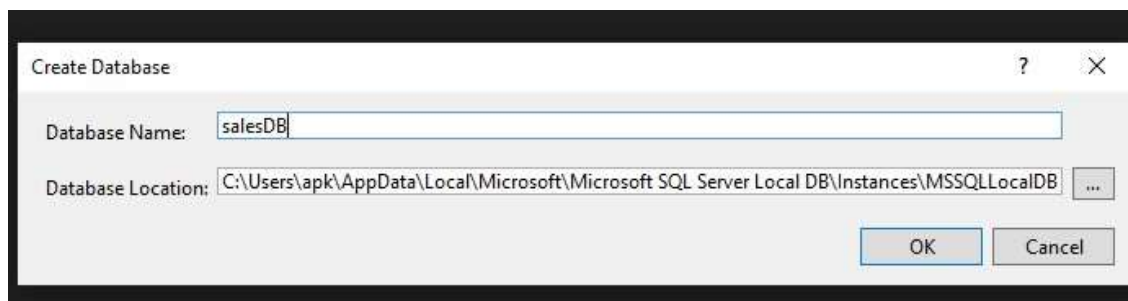
Give create a new project and enter the project name. Now using view gives sql server object explorer where the server lists there. Add the server to connect with the localdb. And connect it

Now open localdb we can see the databases click it and add a new database

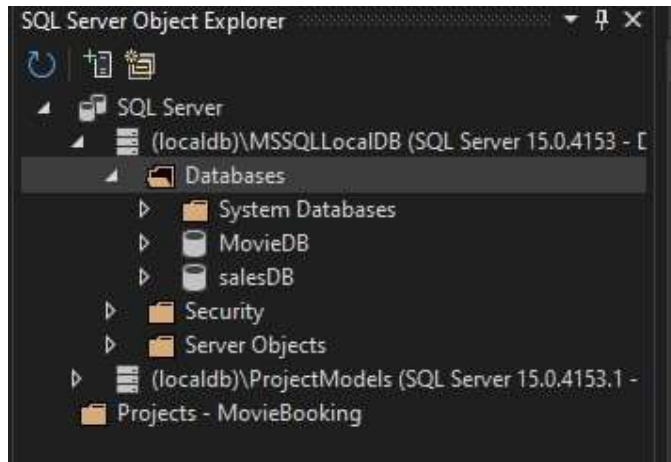


Give the database name

Now click OK

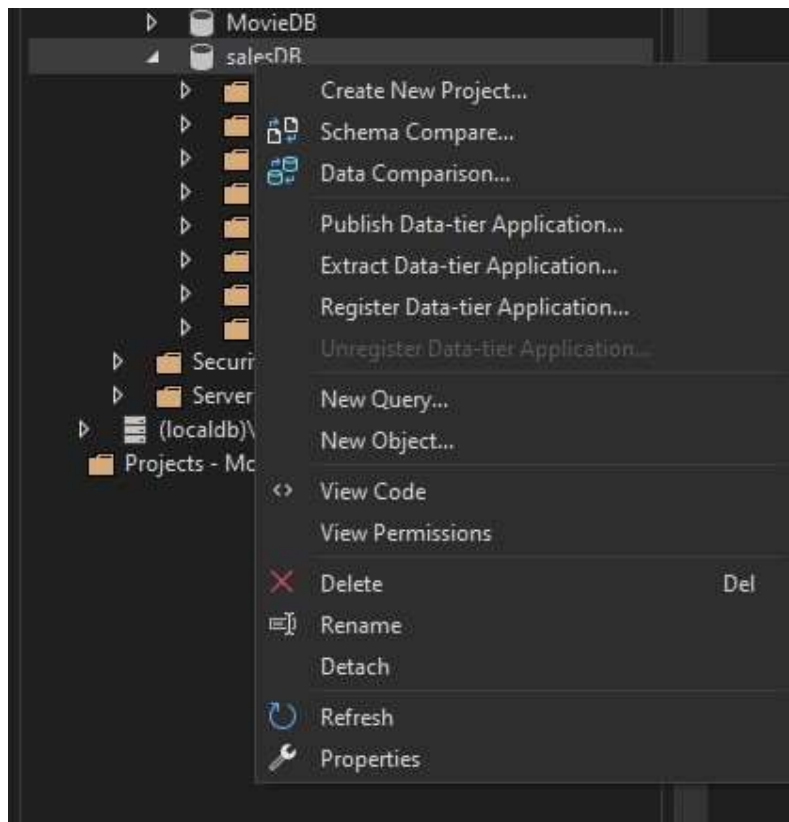


We can see the database created in the sql server object explorer side which gives a clear view of the database which will be used in the project

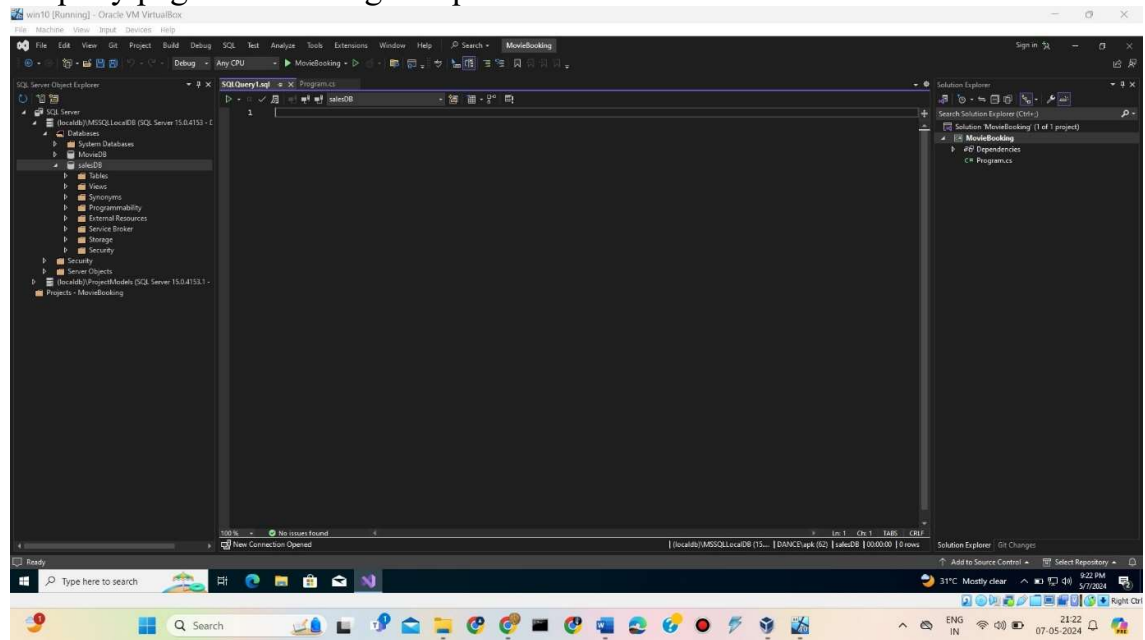


Now right-click the database we can see a new query. Click on it.

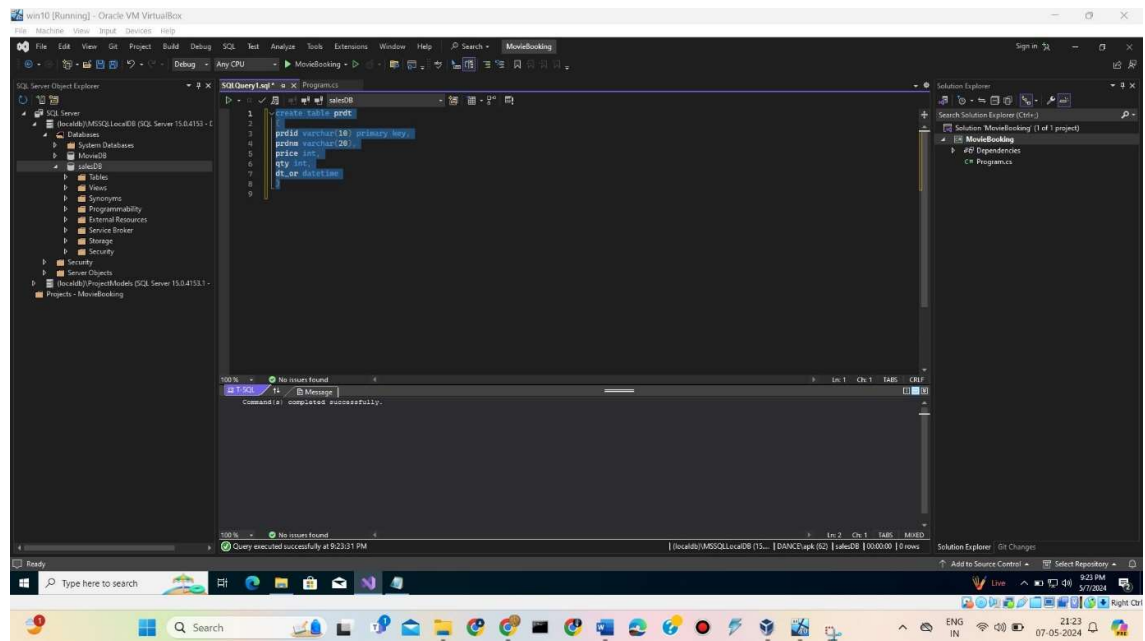
A new query page will be opened where the sql server queries are written.



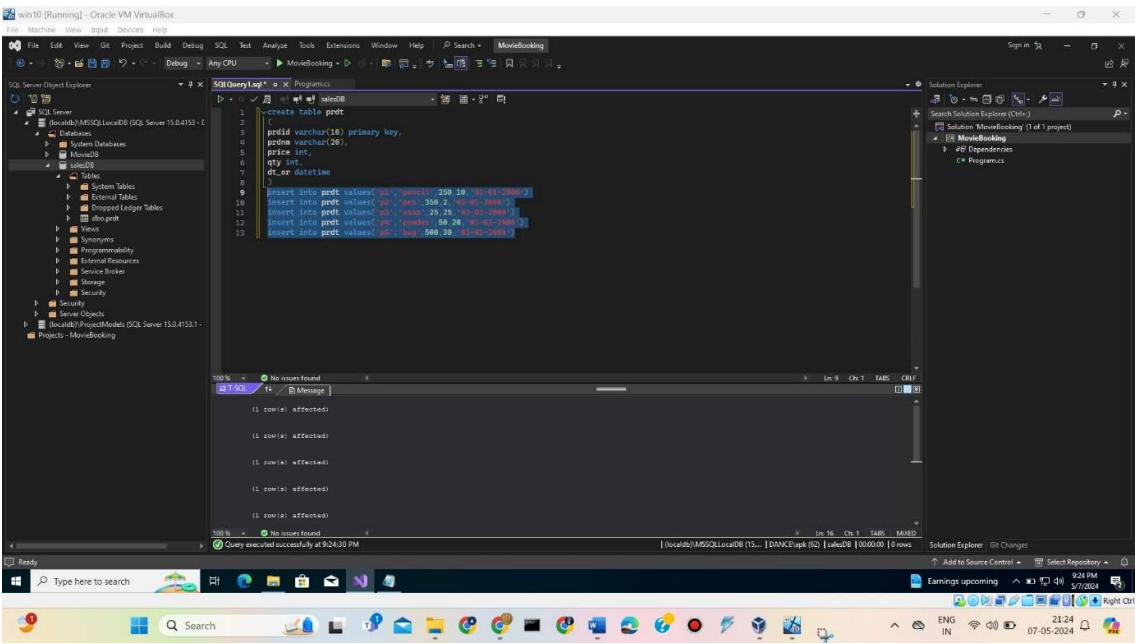
In query page start writing the queries.



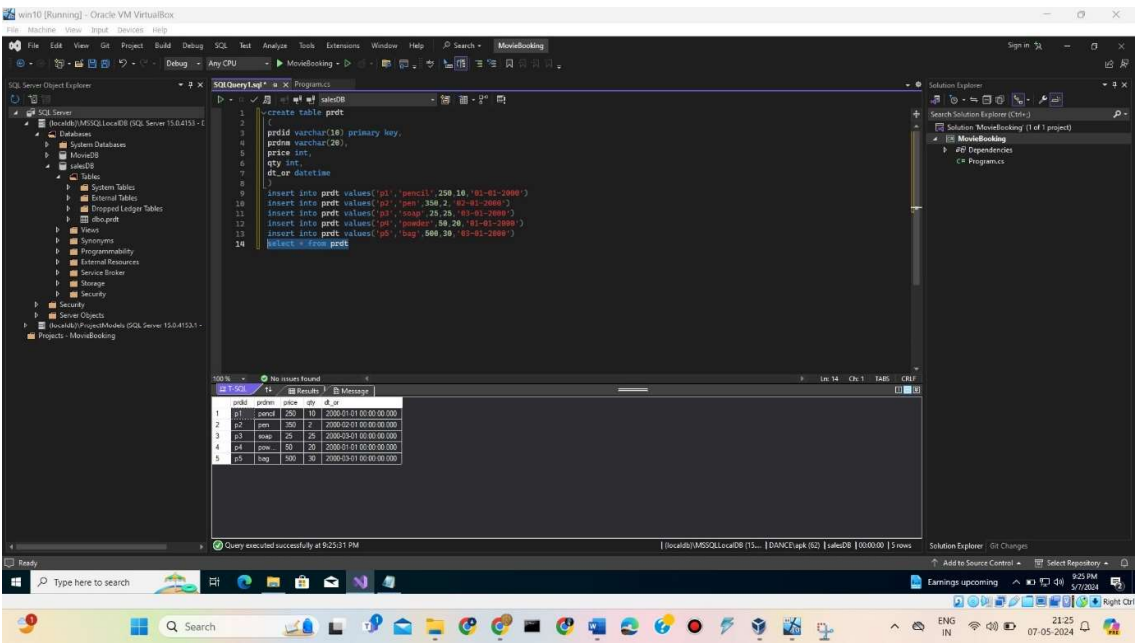
Create table for the database. First table product table



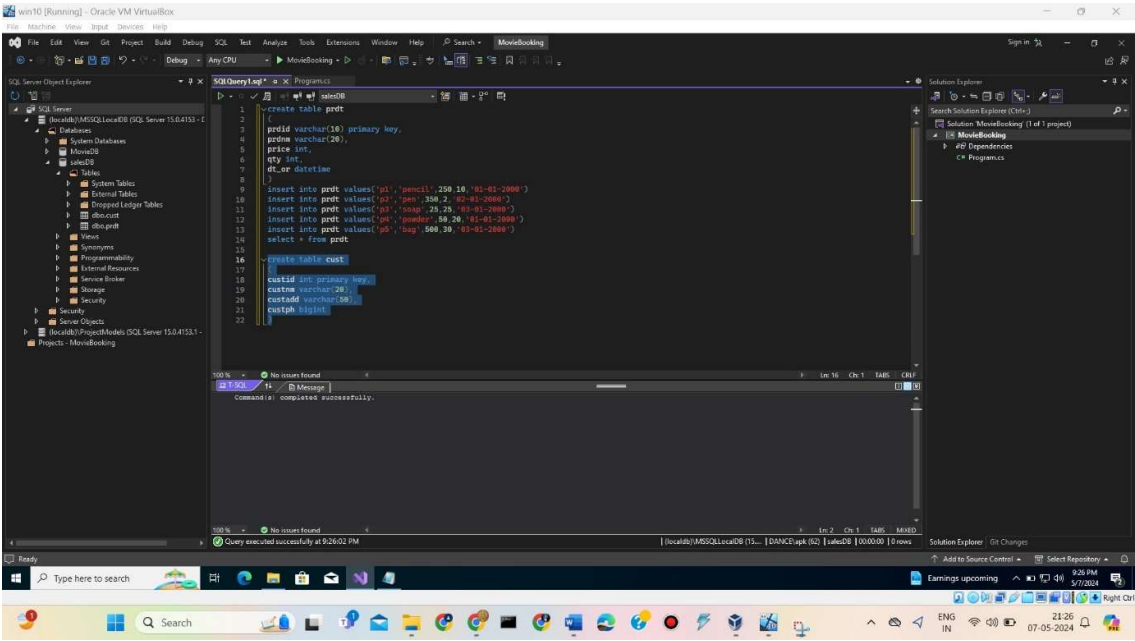
# Insert values into the table



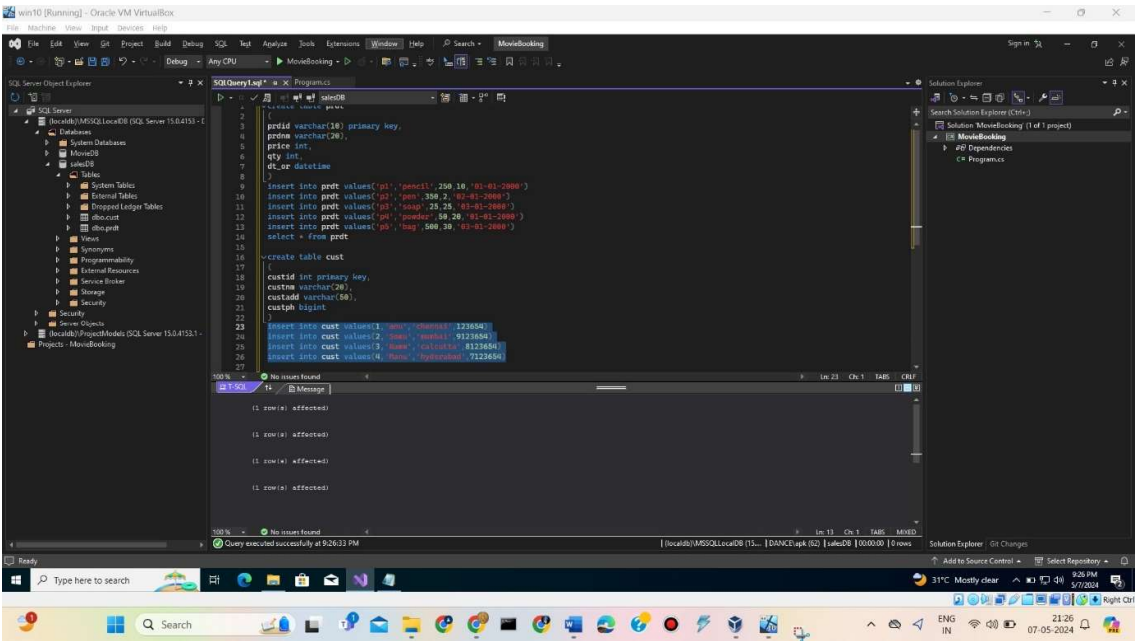
# Using select we can see the table with values



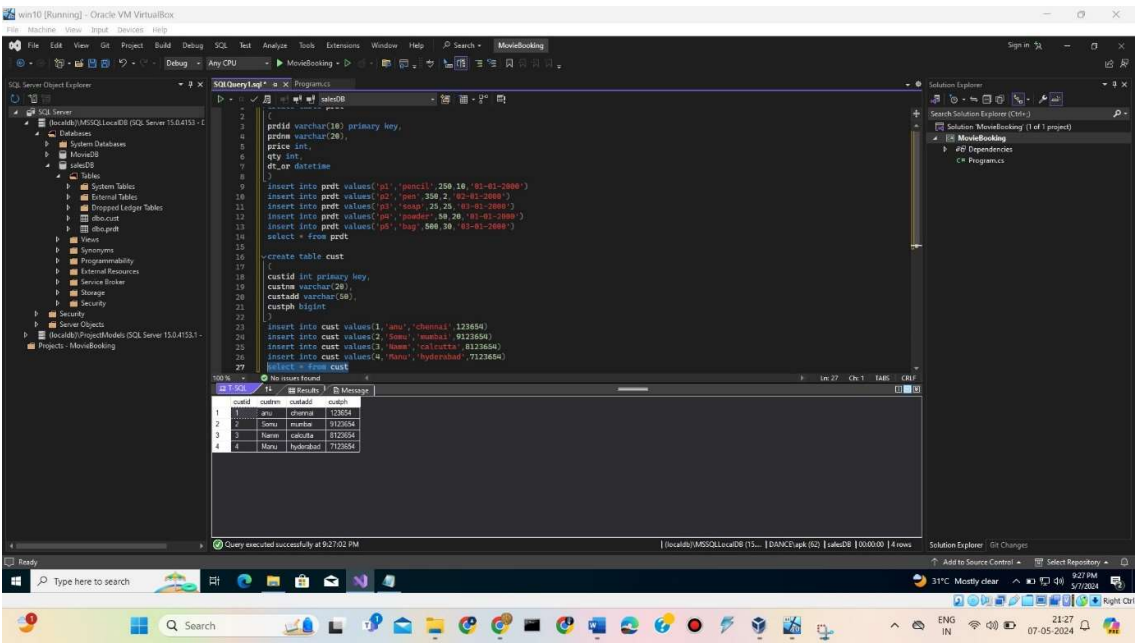
# Create table customer table



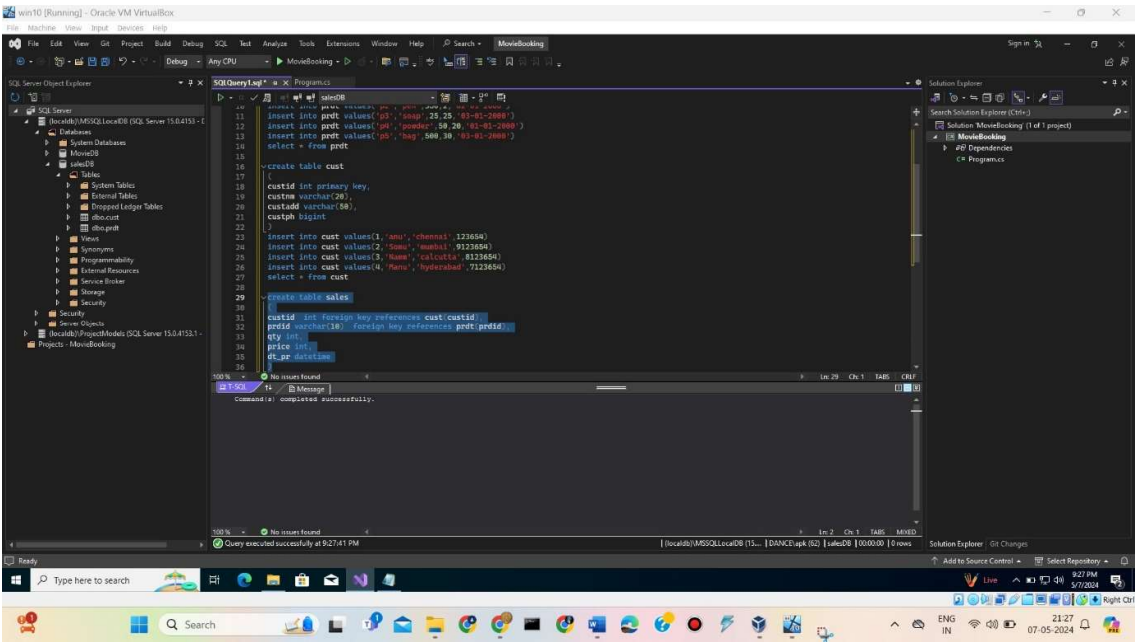
# Insert values into it



Using select we can see it

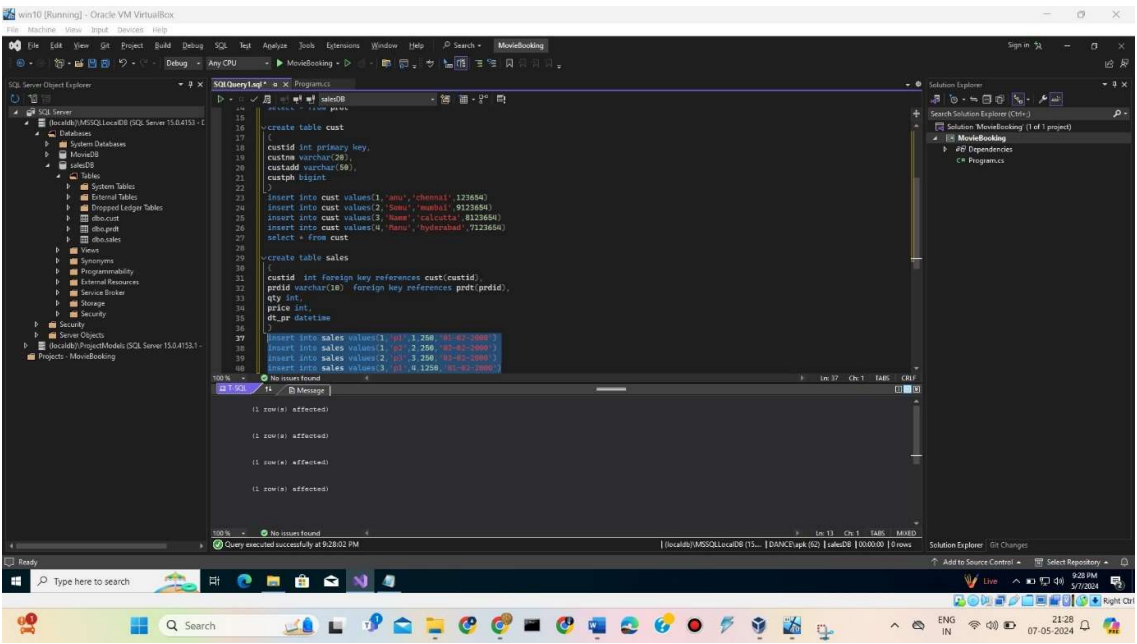


Create sales table

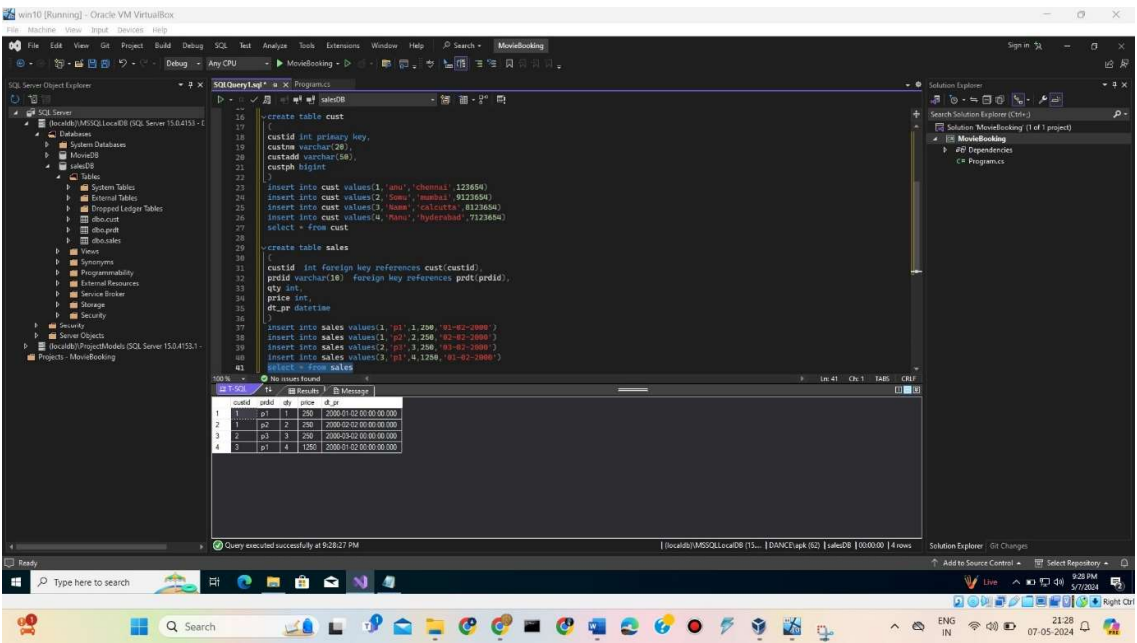




Insert values to it



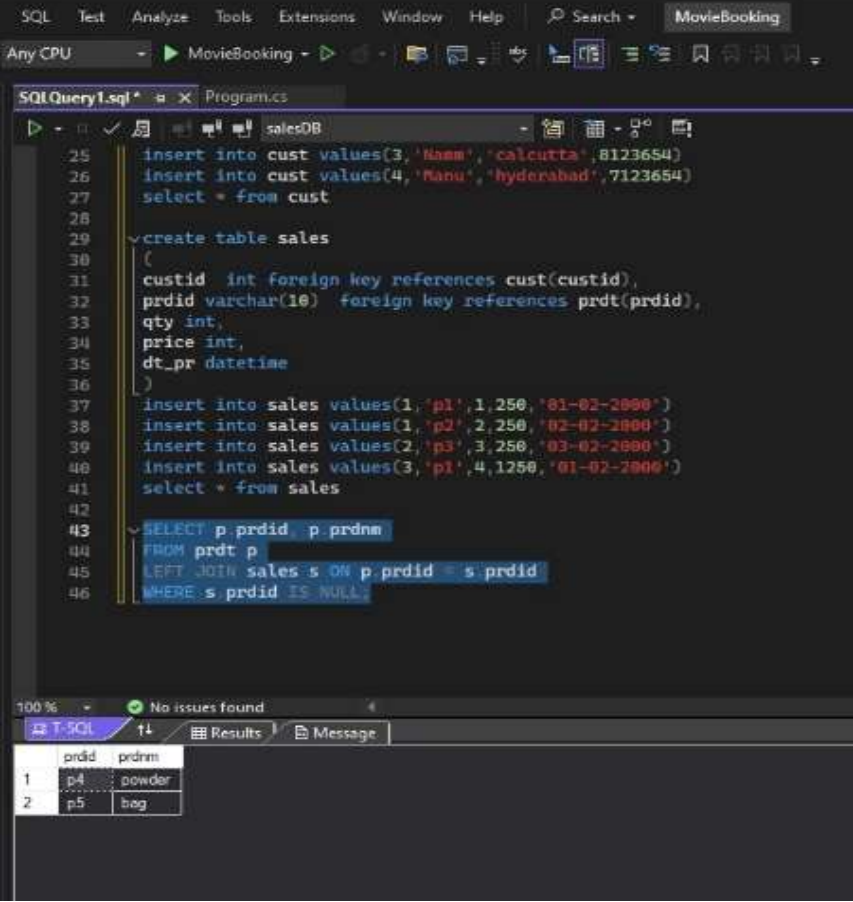
Using select we can see it



**Write a query to find which product has not at all been sold**

Using join we can implement it, where the product details need to be displayed like product id and name from product table and compare it with the sales table which is not sold, this is attained by left join of product table and sales table

**Select column-names from table1 left join table2 on common-column of both tables**



```
SQLQuery1.sql * x Program.cs
25 insert into cust values(3, 'Nann', 'calcutta', 8123654)
26 insert into cust values(4, 'Manu', 'hyderabad', 7123654)
27 select * from cust
28
29 create table sales
30 (
31     custid int foreign key references cust(custid),
32     prdid varchar(10) foreign key references prdt(prdid),
33     qty int,
34     price int,
35     dt_pr datetime
36 )
37 insert into sales values(1, 'p1', 1, 250, '01-02-2000')
38 insert into sales values(1, 'p2', 2, 250, '02-02-2000')
39 insert into sales values(2, 'p3', 3, 250, '03-02-2000')
40 insert into sales values(3, 'p1', 4, 1250, '01-02-2000')
41 select * from sales
42
43 SELECT p.prdid, p.prdnm
44 FROM prdt p
45 LEFT JOIN sales s ON p.prdid = s.prdid
46 WHERE s.prdid IS NULL
```

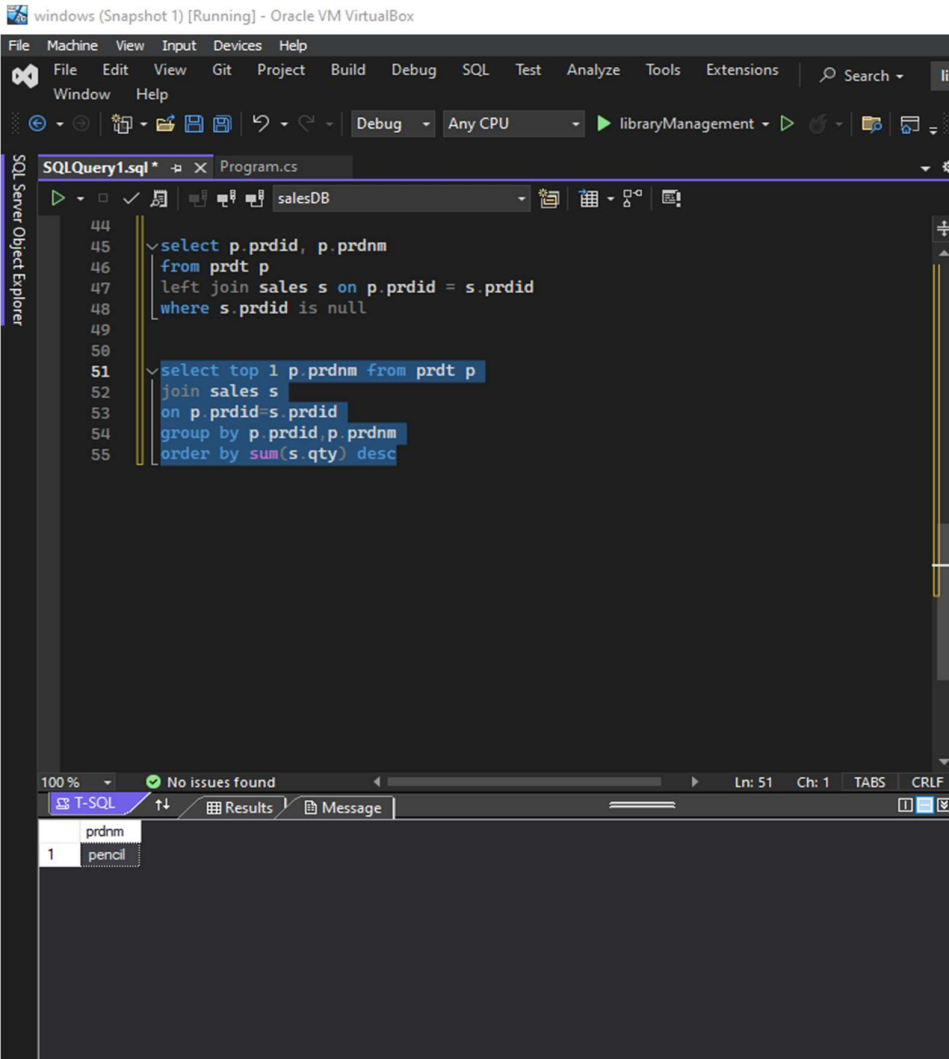
100% No issues found

	prdid	prdnm
1	p4	powder
2	p5	bag



## Write a query to find the most sold product and display it's name

this is also obtained by joining product and sales tables and use groupby and order by funtions



The screenshot shows the SQL Server Enterprise Manager interface. The main window displays a T-SQL query in a dark-themed editor. The query is as follows:

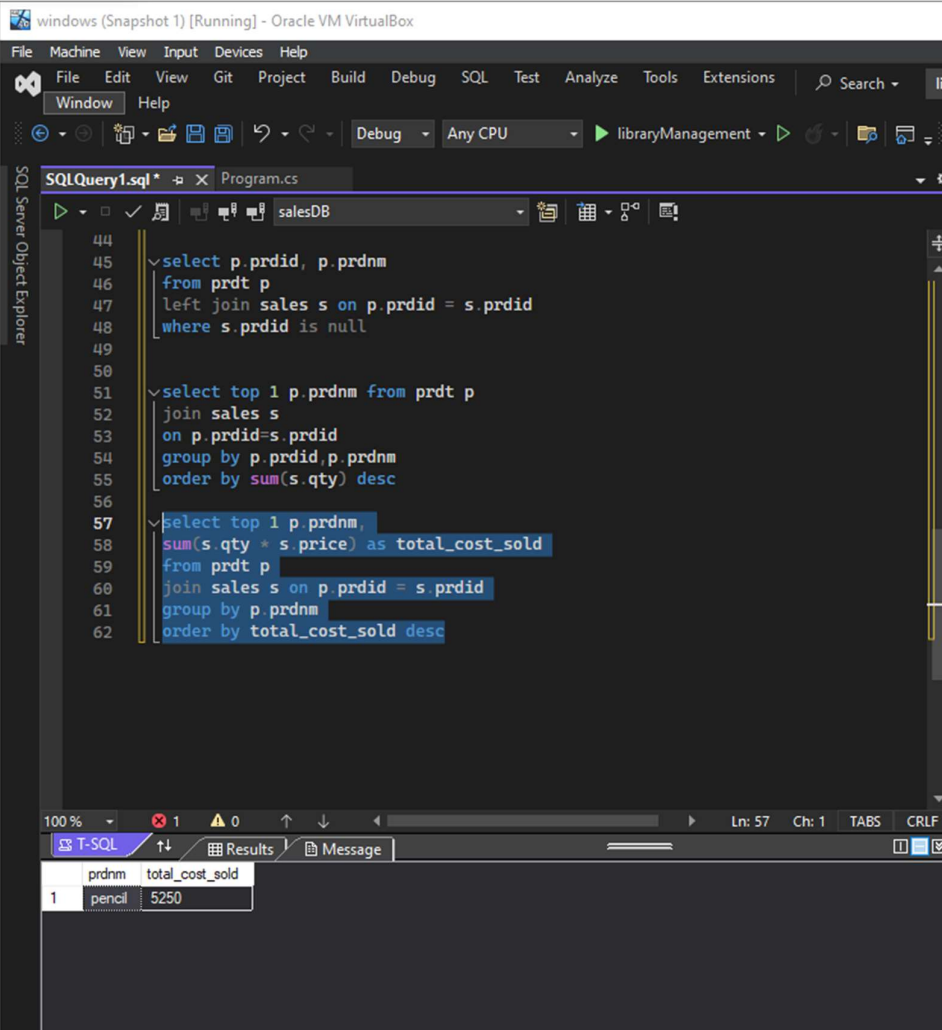
```
44  
45 select p.prdid, p.prdsn  
46 from prdt p  
47 left join sales s on p.prdid = s.prdid  
48 where s.prdid is null  
49  
50  
51 select top 1 p.prdsn from prdt p  
52 join sales s  
53 on p.prdid=s.prdid  
54 group by p.prdid, p.prdsn  
55 order by sum(s.qty) desc
```

The query is executed, and the results are displayed in the Results pane at the bottom. The results show a single row with the product name 'pencil'.

prdsn
1 pencil

**Write a query to find the most sold product and display it's name and total cost sold.**

Here the total cost is calculated and that has been ordered in descending where the total cost is calculated by two tables which is product and sales table by using join this has been achieved



The screenshot shows the SQL Server Enterprise Manager interface. The 'SQLQuery1.sql' file is open, displaying a T-SQL query. The query is as follows:

```
44  
45 select p.prdid, p.prdnm  
46 from prdt p  
47 left join sales s on p.prdid = s.prdid  
48 where s.prdid is null  
49  
50  
51 select top 1 p.prdnm from prdt p  
52 join sales s  
53 on p.prdid=s.prdid  
54 group by p.prdid,p.prdnm  
55 order by sum(s.qty) desc  
56  
57 select top 1 p.prdnm,  
58 sum(s.qty * s.price) as total_cost_sold  
59 from prdt p  
60 join sales s on p.prdid = s.prdid  
61 group by p.prdnm  
62 order by total_cost_sold desc
```

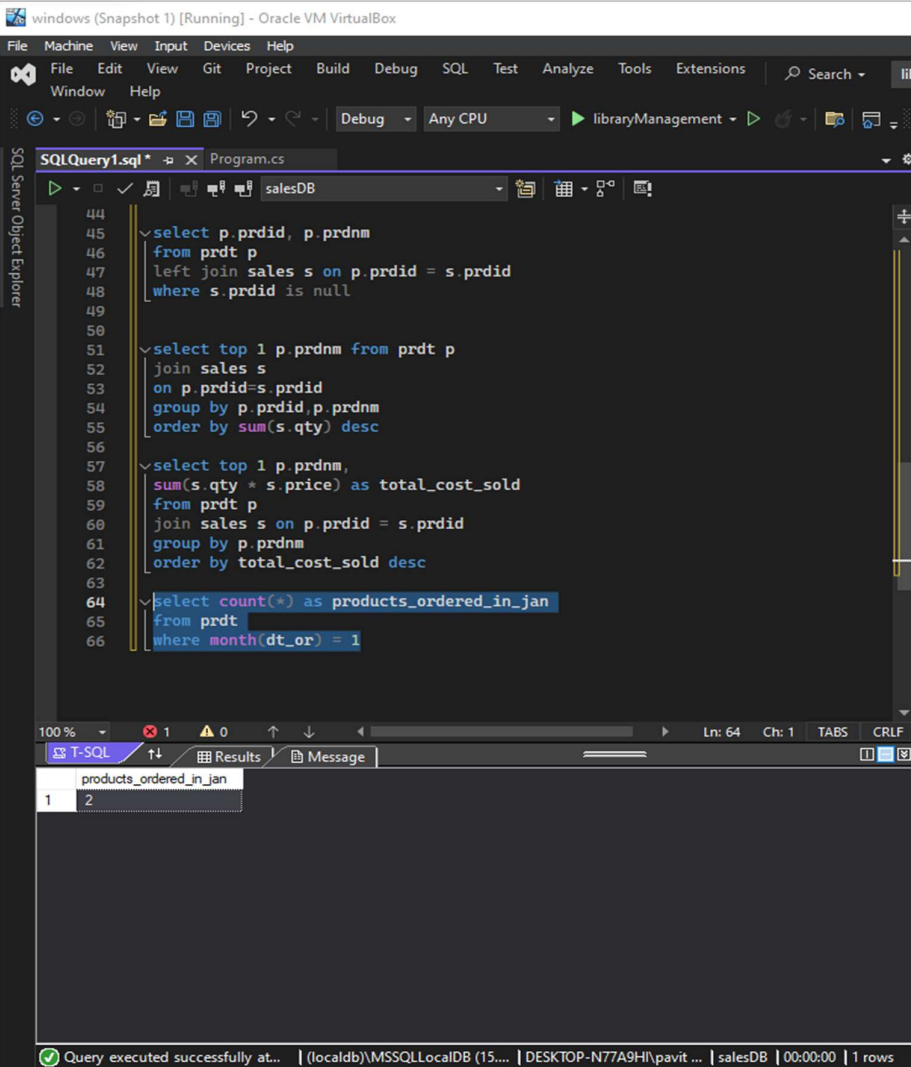
The results pane at the bottom shows the output of the query:

prdnm	total_cost_sold
1 pencil	5250

## How many products has been ordered in the month of Jan

Count aggregation function is used to count the no.of products ordered from product table

Using condition in where clause we can get the count in jan month **where month(date\_orderd)**



The screenshot shows the SQL Server Enterprise Manager interface. The 'SQL Query1.sql' file is open, displaying a T-SQL query. The query is as follows:

```
44
45
46 select p.prdid, p.prdnm
47 from prdt p
48 left join sales s on p.prdid = s.prdid
49 where s.prdid is null
50
51
52 select top 1 p.prdnm from prdt p
53 join sales s
54 on p.prdid=s.prdid
55 group by p.prdid,p.prdnm
56 order by sum(s.qty) desc
57
58 select top 1 p.prdnm,
59 sum(s.qty * s.price) as total_cost_sold
60 from prdt p
61 join sales s on p.prdid = s.prdid
62 group by p.prdnm
63 order by total_cost_sold desc
64
65 select count(*) as products_ordered_in_jan
66 from prdt
67 where month(dt_or) = 1
```

The results pane at the bottom shows the output of the query:

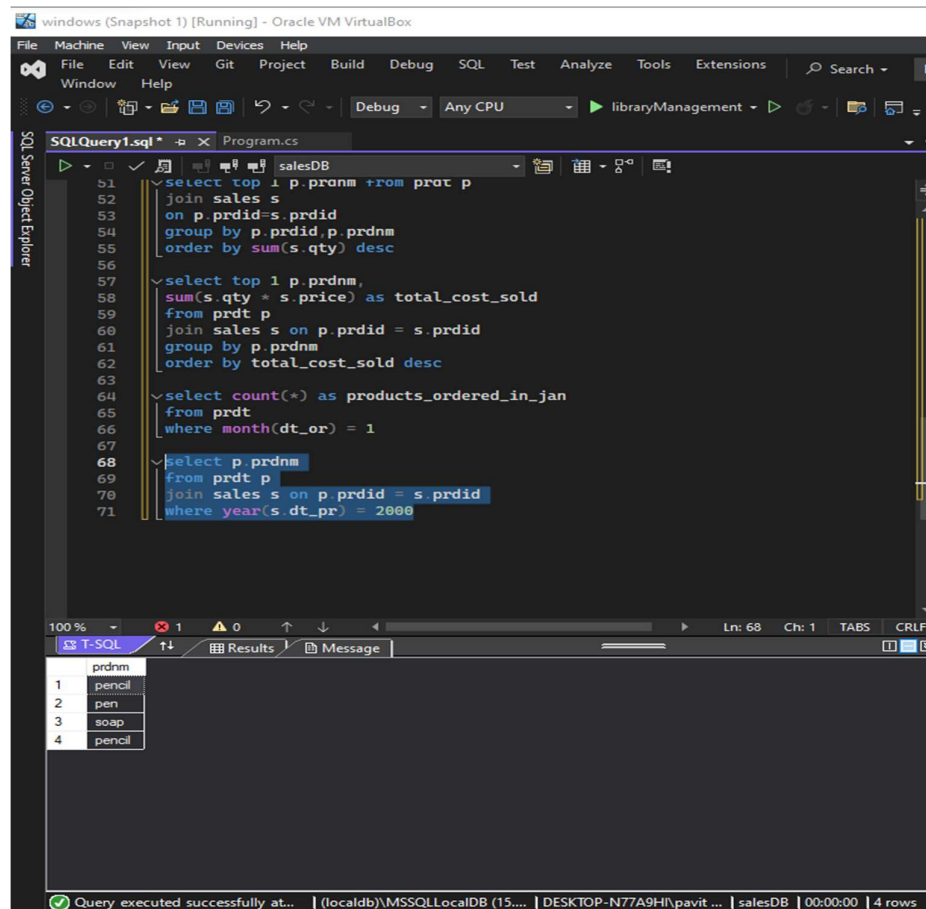
products_ordered_in_jan
2

The status bar at the bottom indicates: Query executed successfully at... | (localdb)\MSSQLLocalDB (15... | DESKTOP-N77A9HI\pavit ... | salesDB | 00:00:00 | 1 rows

## Display the names of the products that has been sold in the year 2000

By joining product and sales table we can attain it. Where there should be a condition the year of the date of purchase need to be 2000

where  $\text{year}(\text{date\_pur})=2000$



The screenshot shows the SQL Server Enterprise Edition interface. The main window displays a T-SQL query in the 'SQLQuery1.sql' editor. The query is as follows:

```
51 select top 1 p.prdsn + 'rom' as prdt p
52 join sales s
53 on p.prdsn=s.prdsn
54 group by p.prdsn,p.prdsn
55 order by sum(s.qty) desc
56
57 select top 1 p.prdsn,
58 sum(s.qty * s.price) as total_cost_sold
59 from prdt p
60 join sales s on p.prdsn = s.prdsn
61 group by p.prdsn
62 order by total_cost_sold desc
63
64 select count(*) as products_ordered_in_jan
65 from prdt
66 where month(dt_or) = 1
67
68 select p.prdsn
69 from prdt p
70 join sales s on p.prdsn = s.prdsn
71 where year(s.dt_pr) = 2000
```

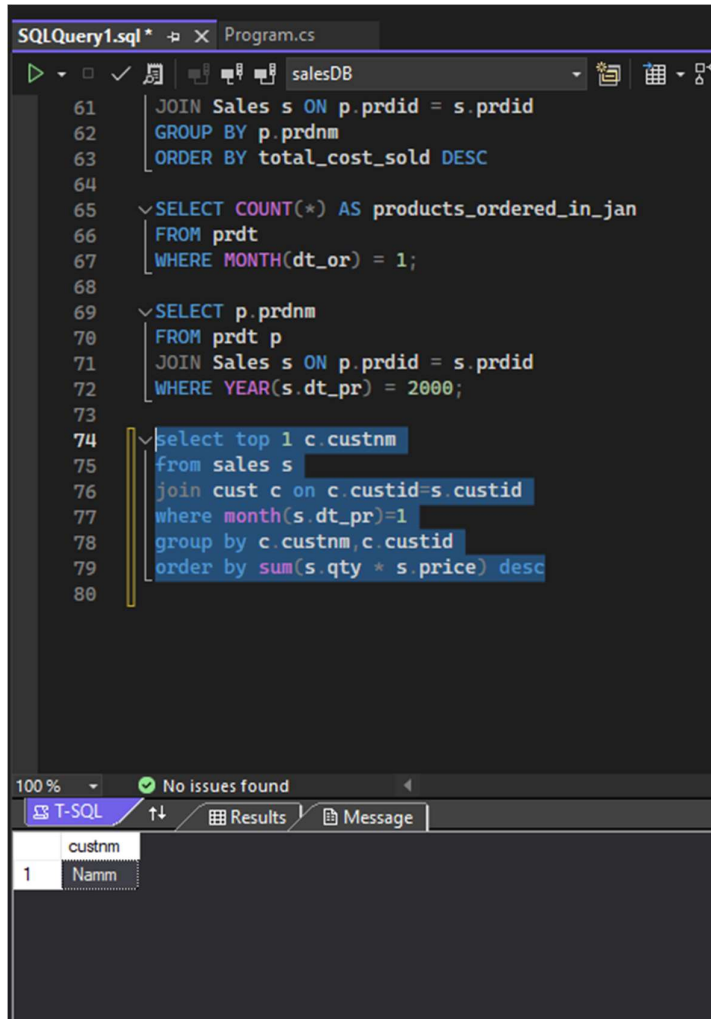
The results pane at the bottom shows the output of the query, which is a list of product names (prdsn) that were sold in the year 2000. The results are as follows:

prdsn
1 pencil
2 pen
3 soap
4 pencil

The status bar at the bottom indicates that the query was executed successfully at 15:00:00 on 11/11/2019, and it returned 4 rows.

## Display the name the customer who has made highest purchase in Jan month

Joining sales and customer table we can obtain it. Using where condition for month and groupby and orderby in descending is used.



The screenshot shows a SQL Server Enterprise Manager window with a T-SQL query editor. The query is as follows:

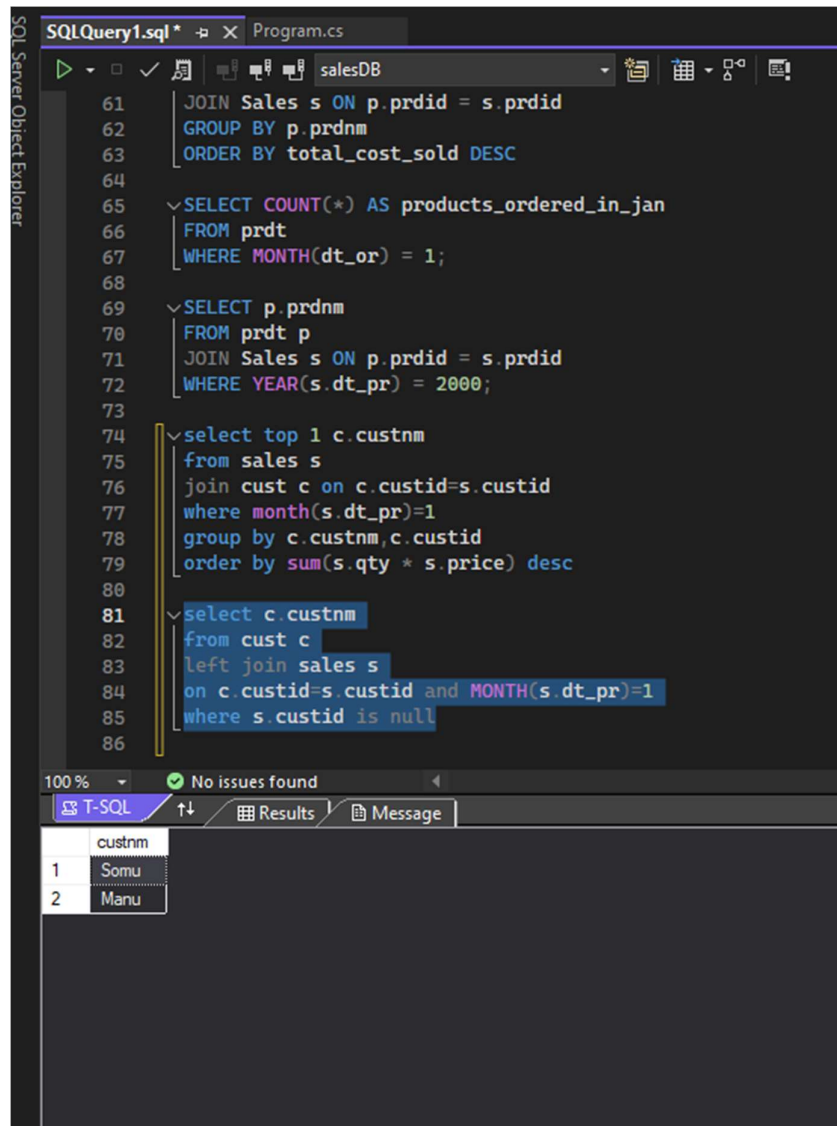
```
61 JOIN Sales s ON p.prdid = s.prdid
62 GROUP BY p.prdnm
63 ORDER BY total_cost_sold DESC
64
65 SELECT COUNT(*) AS products_ordered_in_jan
66 FROM prdt
67 WHERE MONTH(dt_or) = 1;
68
69 SELECT p.prdnm
70 FROM prdt p
71 JOIN Sales s ON p.prdid = s.prdid
72 WHERE YEAR(s.dt_pr) = 2000;
73
74 select top 1 c.custnm
75 from sales s
76 join cust c on c.custid=s.custid
77 where month(s.dt_pr)=1
78 group by c.custnm,c.custid
79 order by sum(s.qty * s.price) desc
80
```

The results pane at the bottom shows a single row with the customer name 'Namm'.

	custnm
1	Namm

## Display the name of the customers who haven't purchased any product in Jan month

Using left join we can establish it on customer and sales table in the condition where it should be jan month and the custid should not be null



```
61 JOIN Sales s ON p.prdid = s.prdid
62 GROUP BY p.prdnm
63 ORDER BY total_cost_sold DESC
64
65 SELECT COUNT(*) AS products_ordered_in_jan
66 FROM prdt
67 WHERE MONTH(dt_or) = 1;
68
69 SELECT p.prdnm
70 FROM prdt p
71 JOIN Sales s ON p.prdid = s.prdid
72 WHERE YEAR(s.dt_pr) = 2000;
73
74 select top 1 c.custnm
75 from sales s
76 join cust c on c.custid=s.custid
77 where month(s.dt_pr)=1
78 group by c.custnm,c.custid
79 order by sum(s.qty * s.price) desc
80
81 select c.custnm
82 from cust c
83 left join sales s
84 on c.custid=s.custid and MONTH(s.dt_pr)=1
85 where s.custid is null
86
```

100 % No issues found

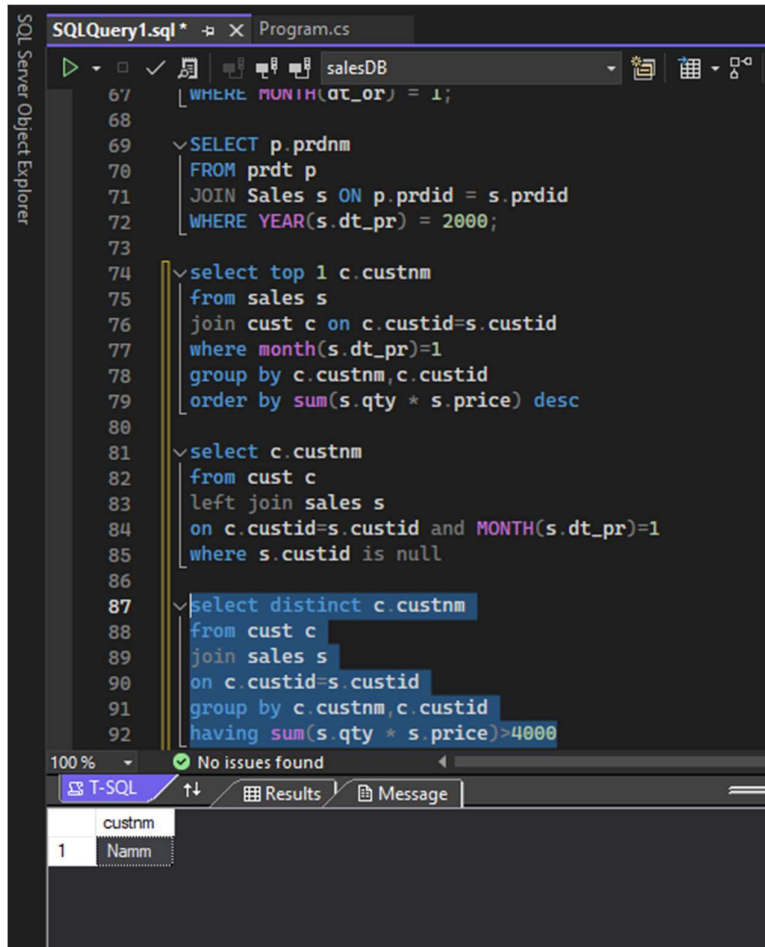
T-SQL Results Message

	custnm
1	Somu
2	Manu



## Display the name of the customers who have purchased above Rs.4000

By joining customer and sales table we can achieve it where the customer need to be purchased more than 4000 rupess which can be calculated by **qty\*price** in sales table



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SQL Server Object Explorer' with the 'salesDB' database selected. The right pane shows a T-SQL query editor with the following code:

```
67 WHERE MONTH(dt_pr) = 1;
68
69 SELECT p.prnm
70 FROM prdt p
71 JOIN Sales s ON p.prdid = s.prdid
72 WHERE YEAR(s.dt_pr) = 2000;
73
74 select top 1 c.custnm
75 from sales s
76 join cust c on c.custid=s.custid
77 where month(s.dt_pr)=1
78 group by c.custnm,c.custid
79 order by sum(s.qty * s.price) desc
80
81 select c.custnm
82 from cust c
83 left join sales s
84 on c.custid=s.custid and MONTH(s.dt_pr)=1
85 where s.custid is null
86
87 select distinct c.custnm
88 from cust c
89 join sales s
90 on c.custid=s.custid
91 group by c.custnm,c.custid
92 having sum(s.qty * s.price)>4000
```

Below the query editor, the 'Results' tab is active, displaying a table with the following data:

custnm
1 Namm