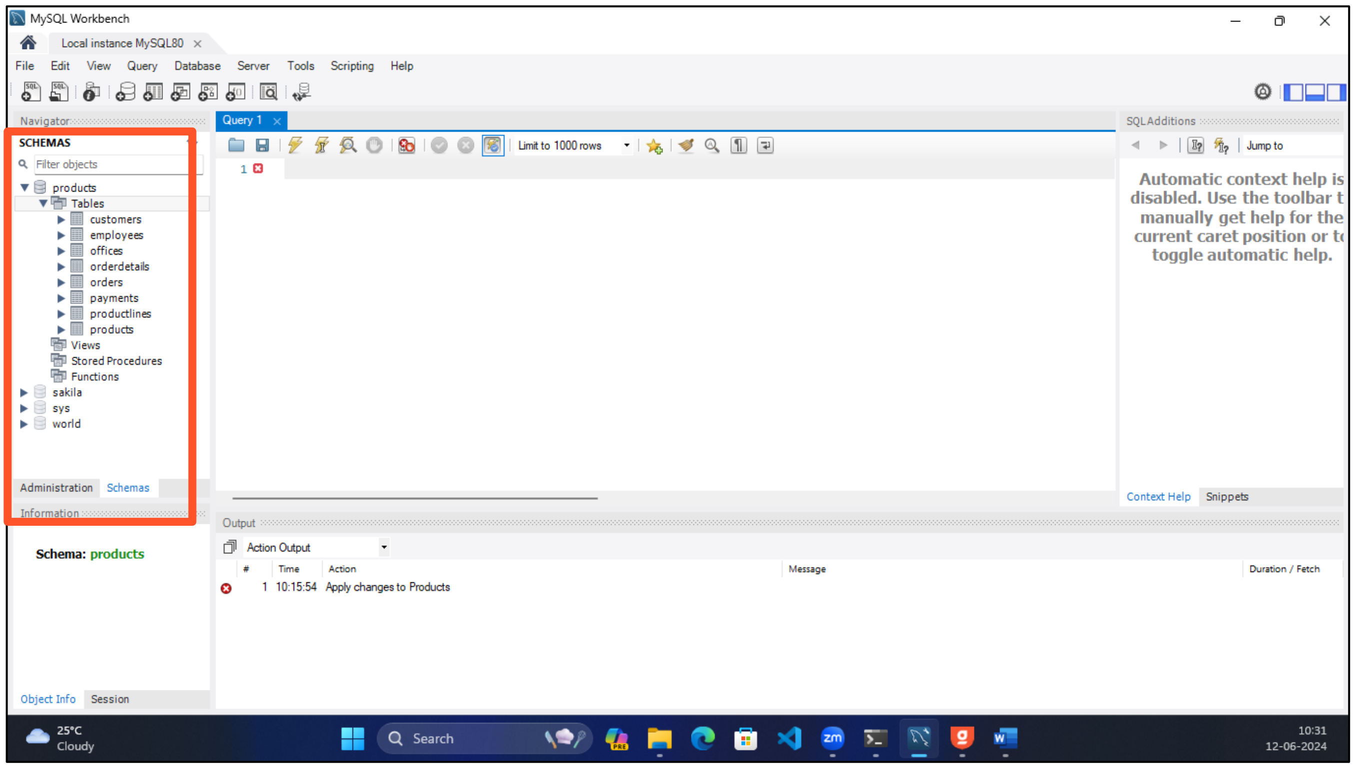
Name: RAUNAK RAJESH SHAH

MySQL Assignment 2

Using WINDOWS

TABLES CREATED SHOWN IN WORKBENCH

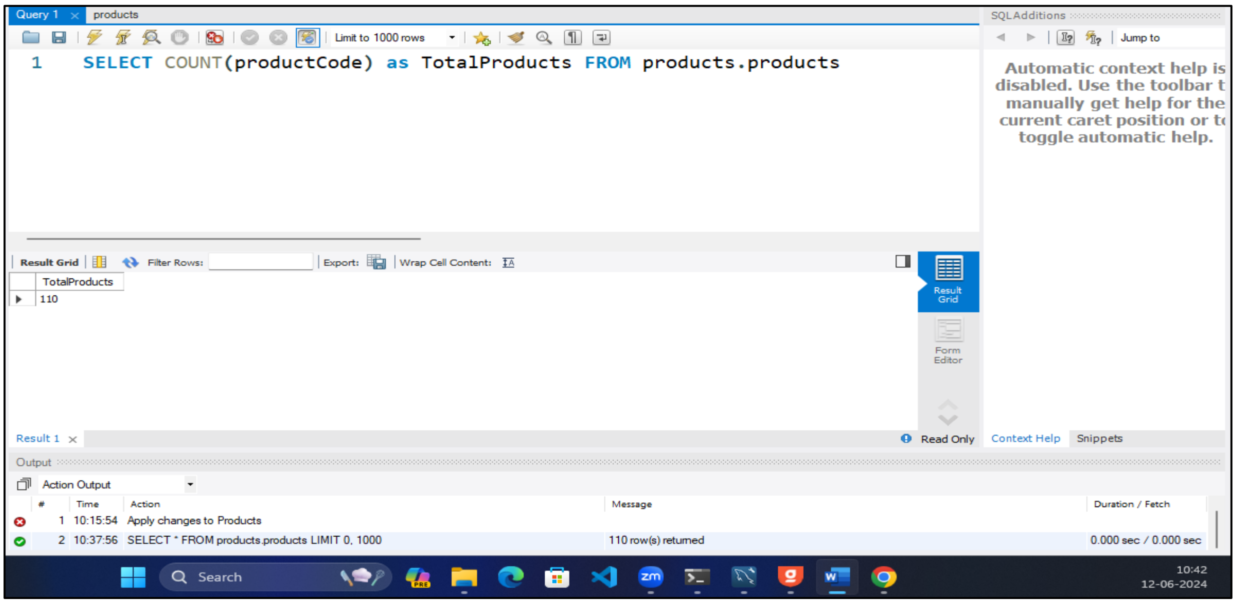


```
#  Time      Action                     Message   Duration / Fetch
1  10:15:54  Apply changes to Products
```

Execution of all the queries

Q1. Write a query to calculate the total number of products in the database.

```sql
SELECT COUNT(productCode) as TotalProducts FROM products.products
```



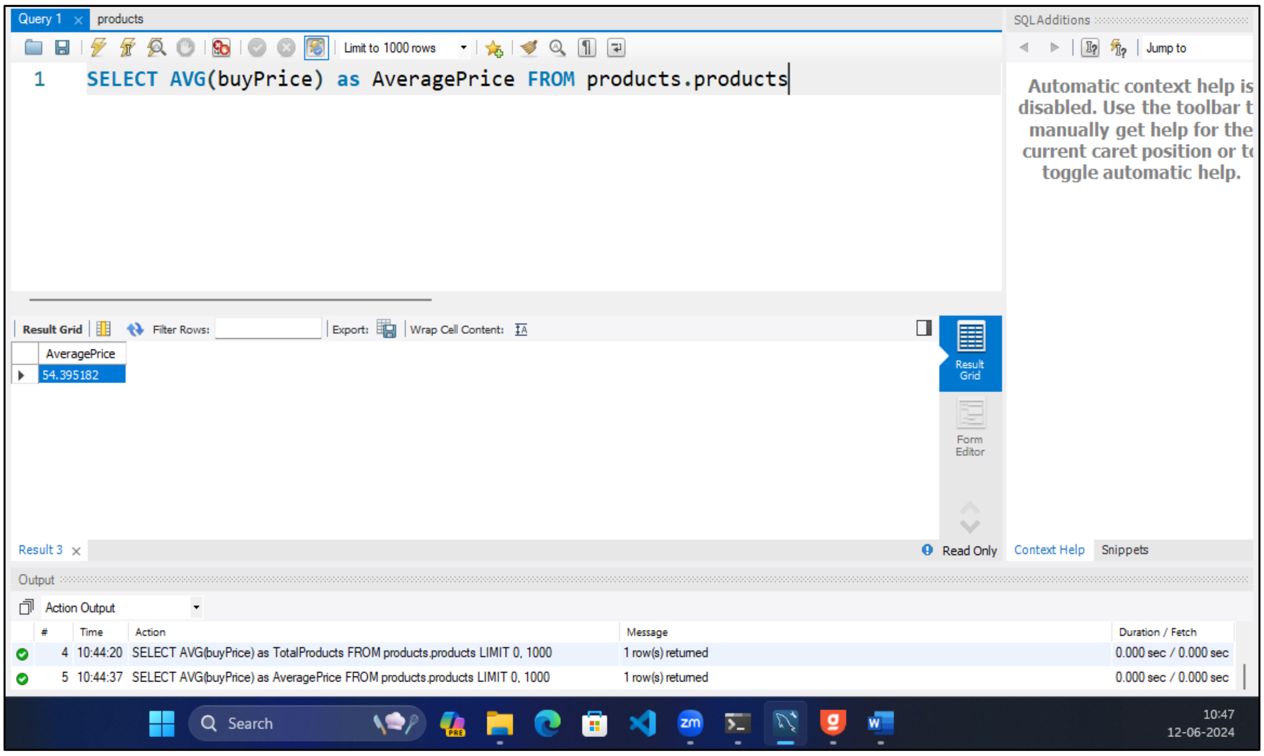
```sql
SELECT COUNT(productCode) as TotalProducts FROM products.products
```

TotalProducts
110

```
#  Time      Action                                       Message              Duration / Fetch
1  10:15:54  Apply changes to Products
2  10:37:56  SELECT * FROM products.products LIMIT 0, 1000  110 row(s) returned  0.000 sec / 0.000 sec
```

Q2. Write a query to find the average buy price of all products

```sql
SELECT AVG(buyPrice) as AveragePrice FROM products.products
```



```sql
SELECT AVG(buyPrice) as AveragePrice FROM products.products
```

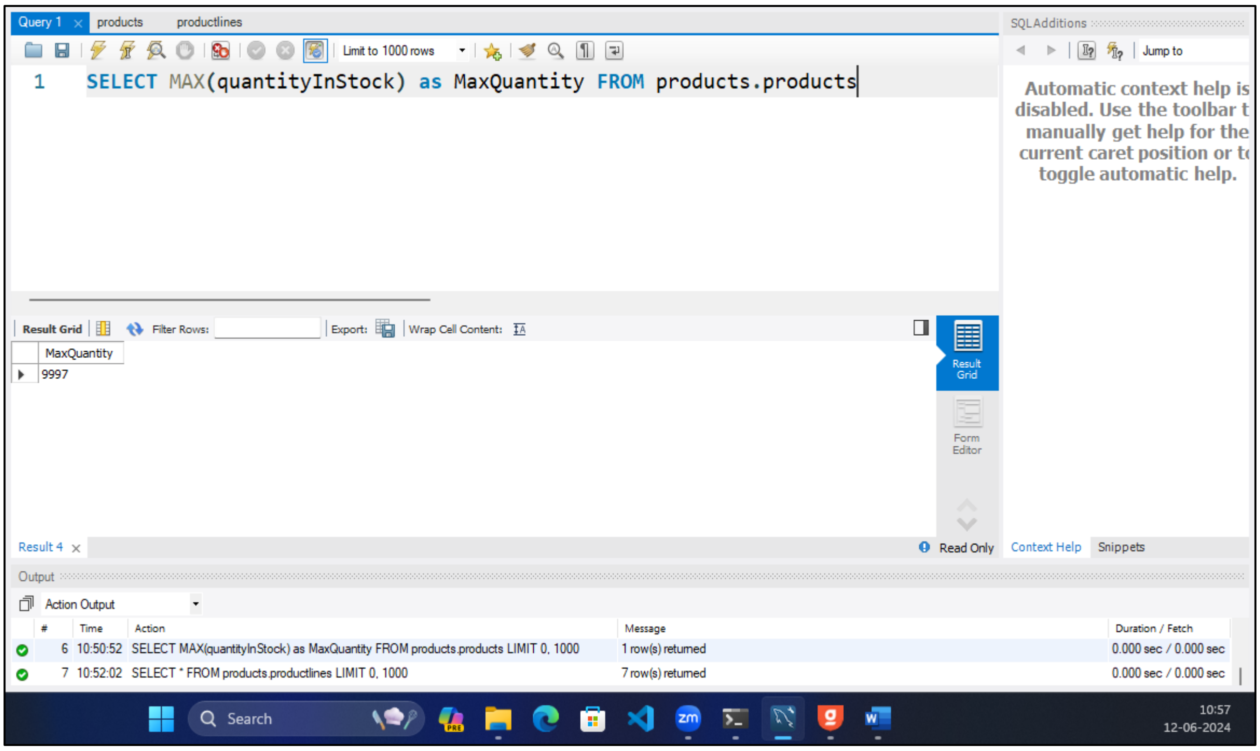
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

AveragePrice
54.395182

```
#  Time      Action                                                                     Message           Duration / Fetch
4  10:44:20  SELECT AVG(buyPrice) as TotalProducts FROM products.products LIMIT 0, 1000  1 row(s) returned  0.000 sec / 0.000 sec
5  10:44:37  SELECT AVG(buyPrice) as AveragePrice FROM products.products LIMIT 0, 1000   1 row(s) returned  0.000 sec / 0.000 sec
```

Q3. Write a query to determine the maximum quantity in stock across all products.

```sql
SELECT MAX(quantityInStock) as MaxQuantity FROM products.products
```



```sql
SELECT MAX(quantityInStock) as MaxQuantity FROM products.products
```

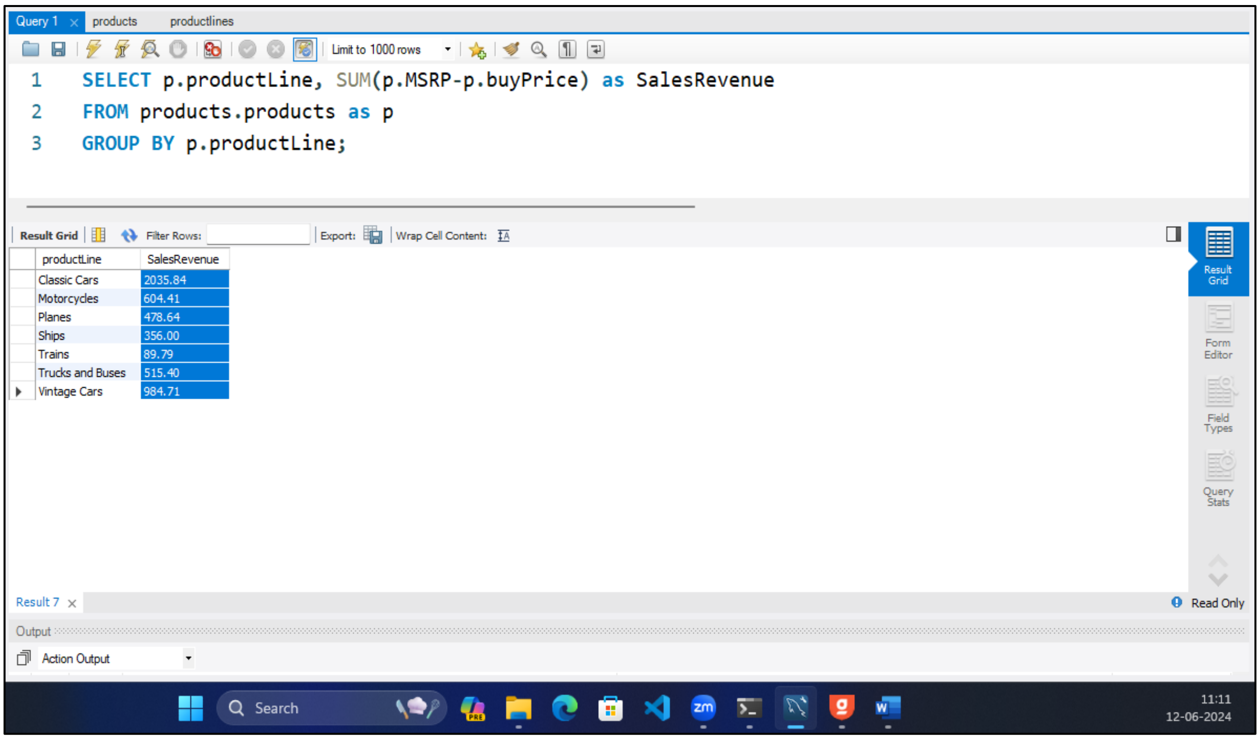
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

MaxQuantity
9997

```
#  Time      Action                                                                           Message            Duration / Fetch
6  10:50:52  SELECT MAX(quantityInStock) as MaxQuantity FROM products.products LIMIT 0, 1000  1 row(s) returned  0.000 sec / 0.000 sec
7  10:52:02  SELECT * FROM products.productlines LIMIT 0, 1000                               7 row(s) returned  0.000 sec / 0.000 sec
```

Q4. Write a query to calculate the total sales revenue for each product line.

```
SELECT p.productLine, SUM(p.MSRP-p.buyPrice) as SalesRevenue
FROM products.products as p
GROUP BY p.productLine;
```

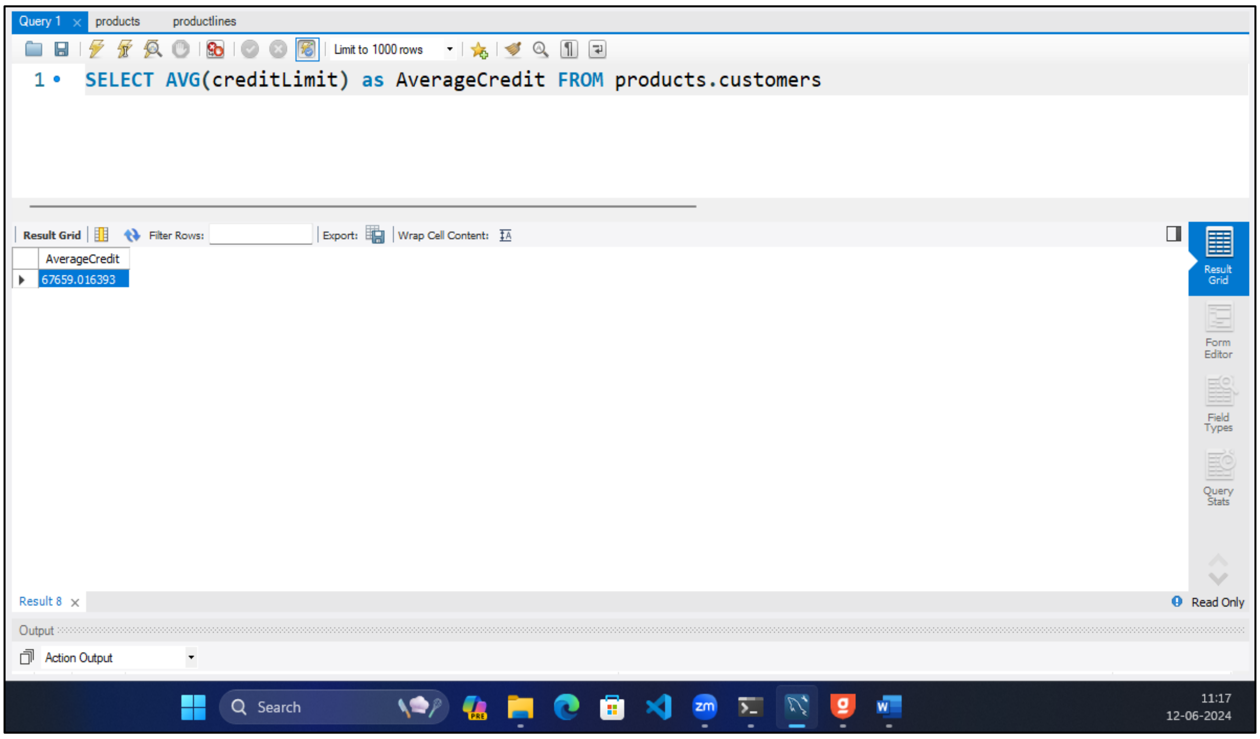


```
SELECT p.productLine, SUM(p.MSRP-p.buyPrice) as SalesRevenue
FROM products.products as p
GROUP BY p.productLine;
```

productLine	SalesRevenue
Classic Cars	2035.84
Motorcycles	604.41
Planes	478.64
Ships	356.00
Trains	89.79
Trucks and Buses	515.40
Vintage Cars	984.71

Q5. Write a query to determine the average credit limit for all customers.

```
SELECT AVG(creditLimit) as AverageCredit FROM products.customers
```

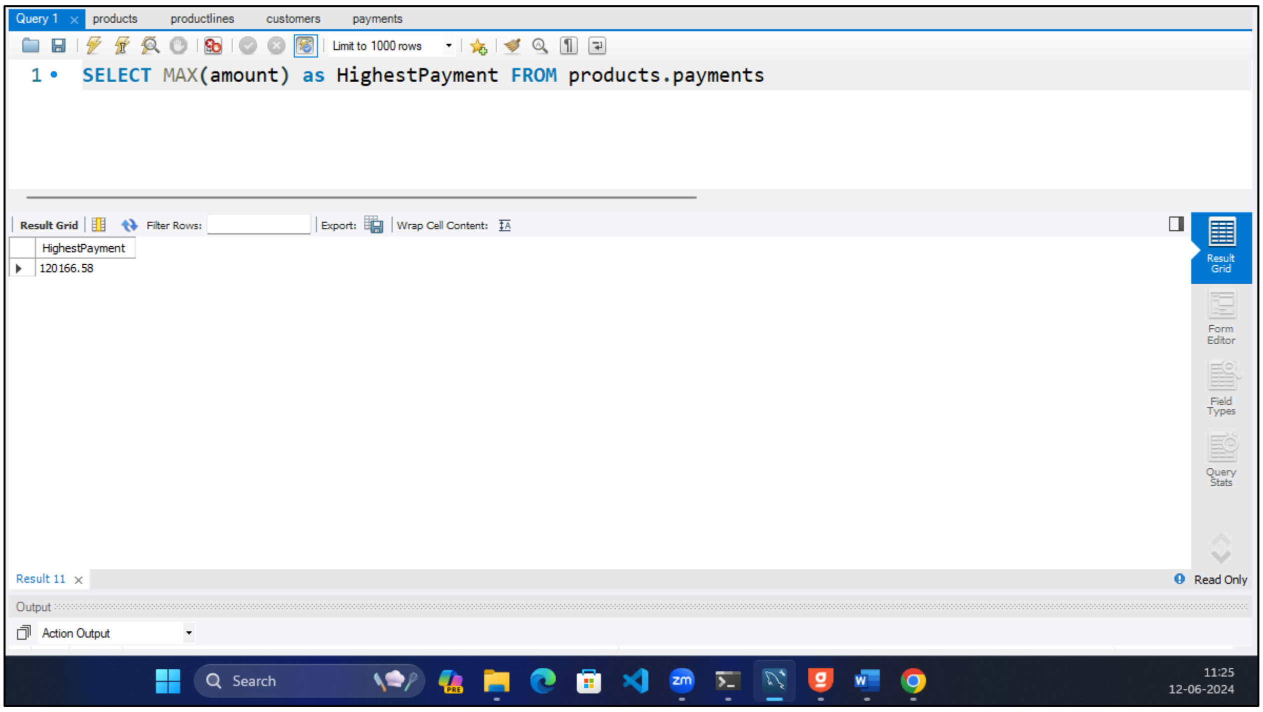


```
SELECT AVG(creditLimit) as AverageCredit FROM products.customers
```

AverageCredit
67659.016393

Q6. Write a query to find the highest payment amount made by a customer.

```
SELECT MAX(amount) as HighestPayment FROM products.payments
```

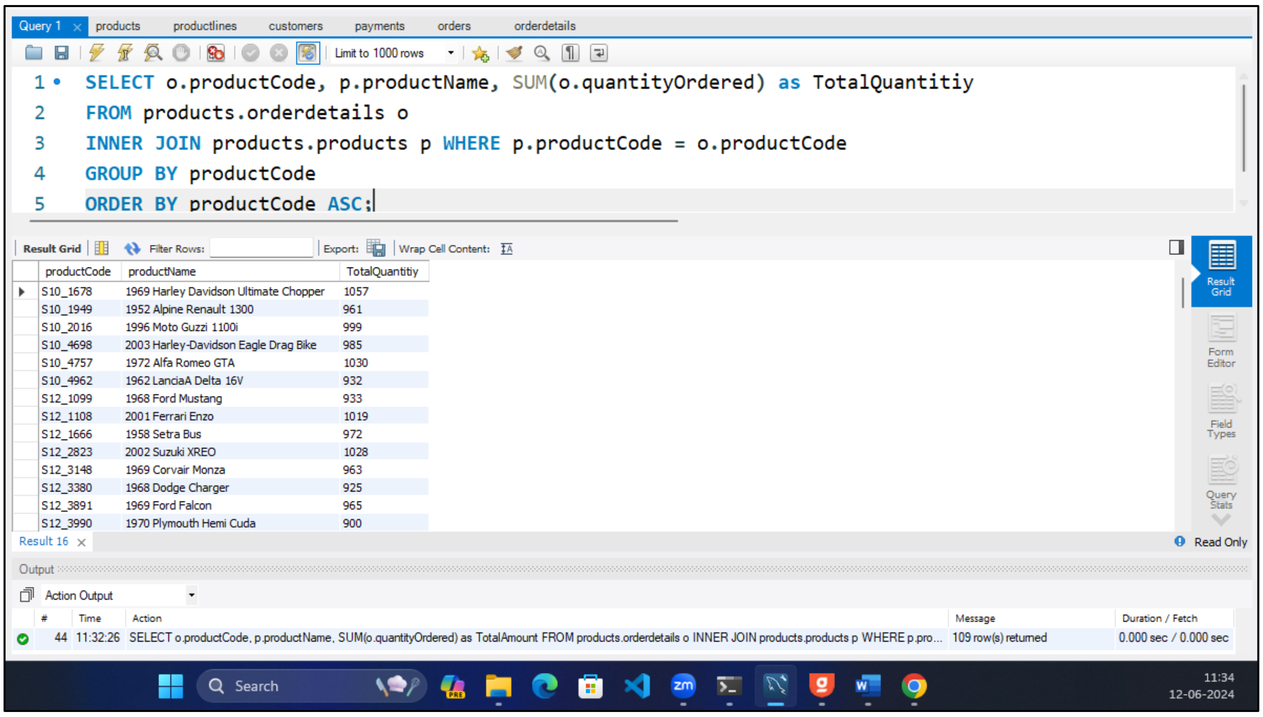


```
SELECT MAX(amount) as HighestPayment FROM products.payments
```

HighestPayment
120166.58

Q7. Write a query to calculate the total quantity ordered for each product.

```
SELECT o.productCode, p.productName, SUM(o.quantityOrdered) as To-
talQuantitiy
FROM products.orderdetails o
INNER JOIN products.products p WHERE p.productCode = o.productCode
GROUP BY productCode
ORDER BY productCode ASC;
```



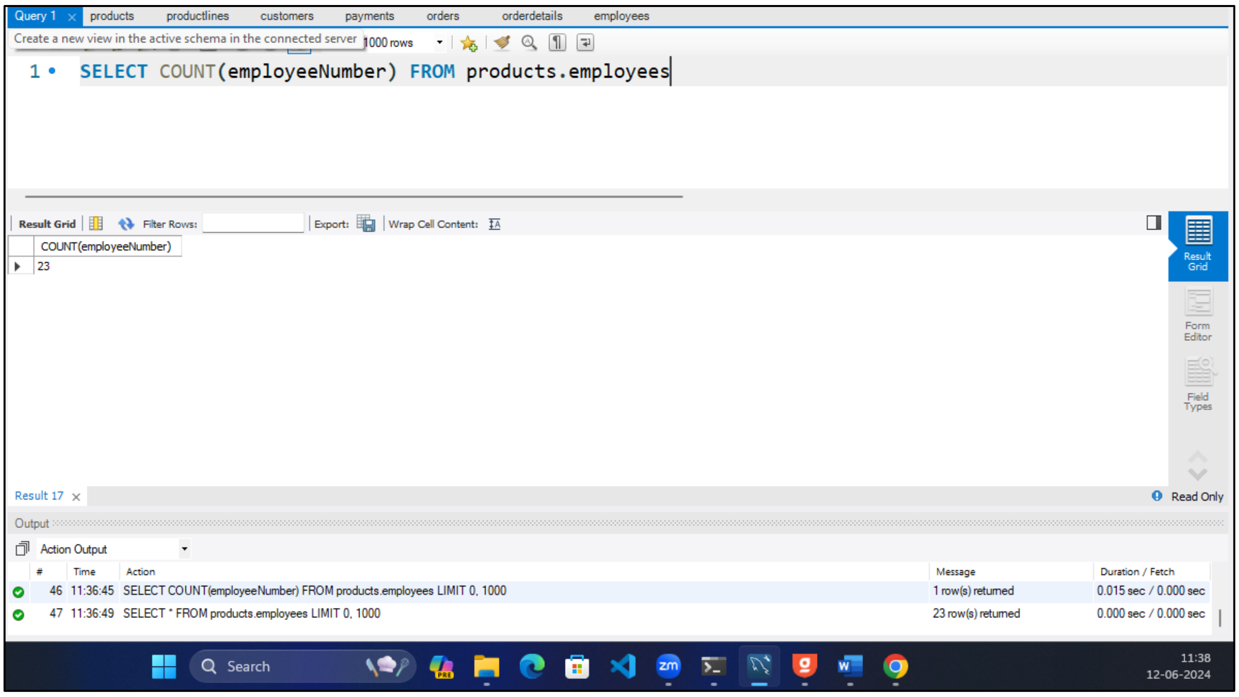
```
SELECT o.productCode, p.productName, SUM(o.quantityOrdered) as TotalQuantitiy
FROM products.orderdetails o
INNER JOIN products.products p WHERE p.productCode = o.productCode
GROUP BY productCode
ORDER BY productCode ASC;
```

productCode	productName	TotalQuantitiy
S10_1678	1969 Harley Davidson Ultimate Chopper	1057
S10_1949	1952 Alpine Renault 1300	961
S10_2016	1996 Moto Guzzi 1100i	999
S10_4698	2003 Harley-Davidson Eagle Drag Bike	985
S10_4757	1972 Alfa Romeo GTA	1030
S10_4962	1962 LanciaA Delta 16V	932
S12_1099	1968 Ford Mustang	933
S12_1108	2001 Ferrari Enzo	1019
S12_1666	1958 Setra Bus	972
S12_2823	2002 Suzuki XREO	1028
S12_3148	1969 Corvair Monza	963
S12_3380	1968 Dodge Charger	925
S12_3891	1969 Ford Falcon	965
S12_3990	1970 Plymouth Hemi Cuda	900

```
#   Time      Action                                                                                                                                   Message              Duration / Fetch
44  11:32:26  SELECT o.productCode, p.productName, SUM(o.quantityOrdered) as TotalAmount FROM products.orderdetails o INNER JOIN products.products p WHERE p.pro...  109 row(s) returned  0.000 sec / 0.000 sec
```

Q8. Write a query to determine the number of employees in each office.

```sql
SELECT COUNT(employeeNumber) FROM products.employees
```



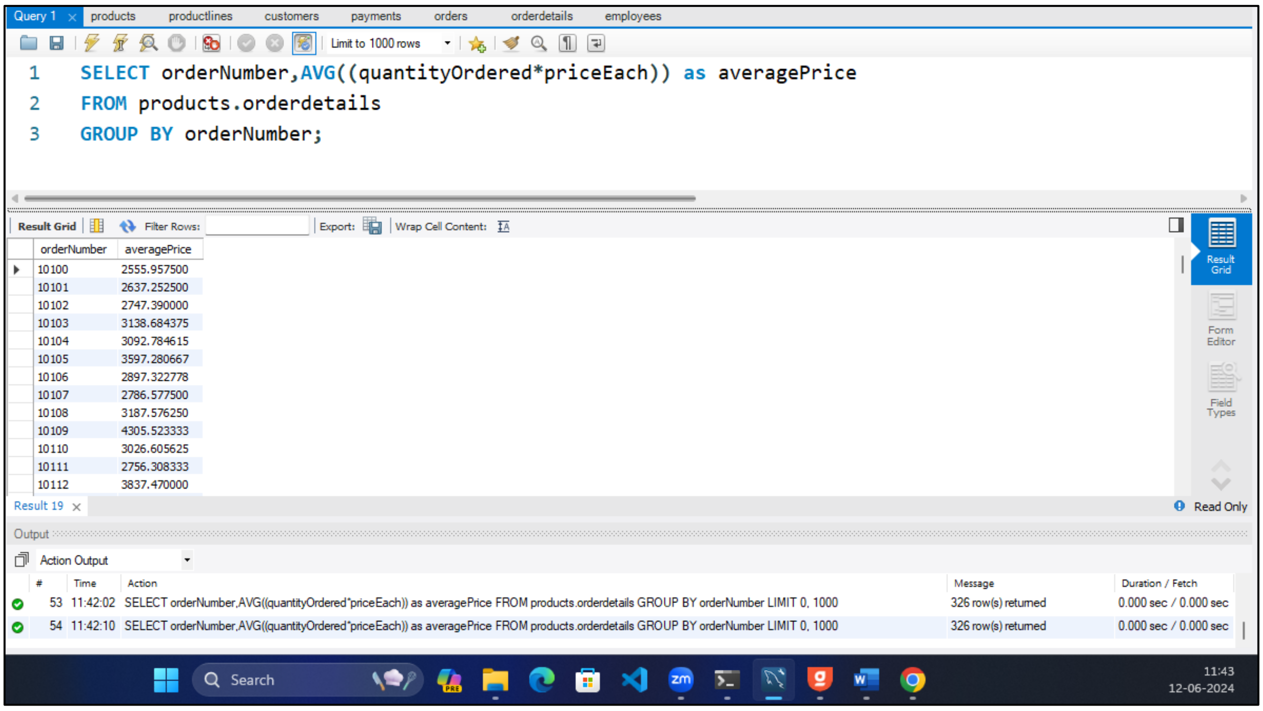
```sql
SELECT COUNT(employeeNumber) FROM products.employees
```

```
COUNT(employeeNumber)
23
```

```
#   Time      Action                                                             Message              Duration / Fetch
46  11:36:45  SELECT COUNT(employeeNumber) FROM products.employees LIMIT 0, 1000  1 row(s) returned    0.015 sec / 0.000 sec
47  11:36:49  SELECT * FROM products.employees LIMIT 0, 1000                     23 row(s) returned   0.000 sec / 0.000 sec
```

Q9. Write a query to calculate the average price for each order.

```sql
SELECT orderNumber,AVG((quantityOrdered*priceEach)) as averagePrice
FROM products.orderdetails
GROUP BY orderNumber;
```



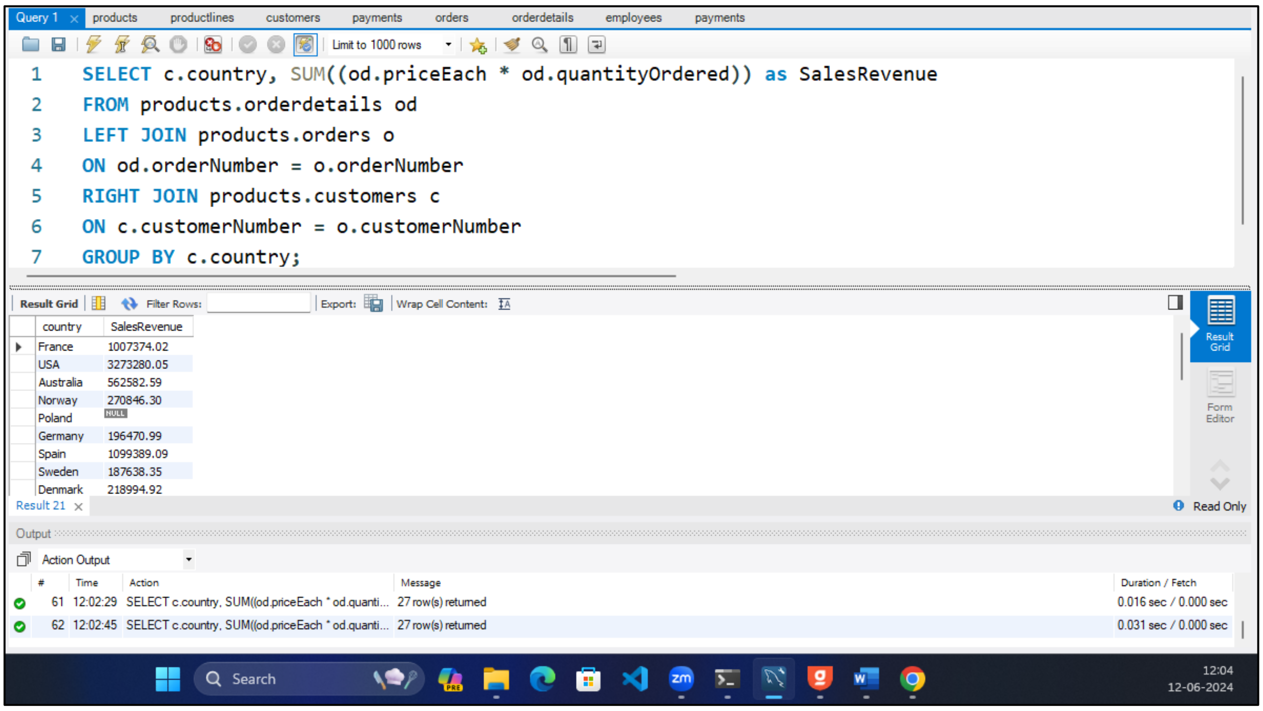
```sql
SELECT orderNumber,AVG((quantityOrdered*priceEach)) as averagePrice
FROM products.orderdetails
GROUP BY orderNumber;
```

```
orderNumber  averagePrice
10100        2555.957500
10101        2637.252500
10102        2747.390000
10103        3138.684375
10104        3092.784615
10105        3597.280667
10106        2897.322778
10107        2786.577500
10108        3187.576250
10109        4305.523333
10110        3026.605625
10111        2756.308333
10112        3837.470000
```

```
#   Time      Action                                                                                                                   Message              Duration / Fetch
53  11:42:02  SELECT orderNumber,AVG((quantityOrdered*priceEach)) as averagePrice FROM products.orderdetails GROUP BY orderNumber LIMIT 0, 1000  326 row(s) returned  0.000 sec / 0.000 sec
54  11:42:10  SELECT orderNumber,AVG((quantityOrdered*priceEach)) as averagePrice FROM products.orderdetails GROUP BY orderNumber LIMIT 0, 1000  326 row(s) returned  0.000 sec / 0.000 sec
```

Q10. Write a query to determine the total sales revenue for each country.

```sql
SELECT c.country, SUM((od.priceEach * od.quantityOrdered)) as Sales-
Revenue
FROM products.orderdetails od
LEFT JOIN products.orders o
ON od.orderNumber = o.orderNumber
RIGHT JOIN products.customers c
ON c.customerNumber = o.customerNumber
GROUP BY c.country;
```



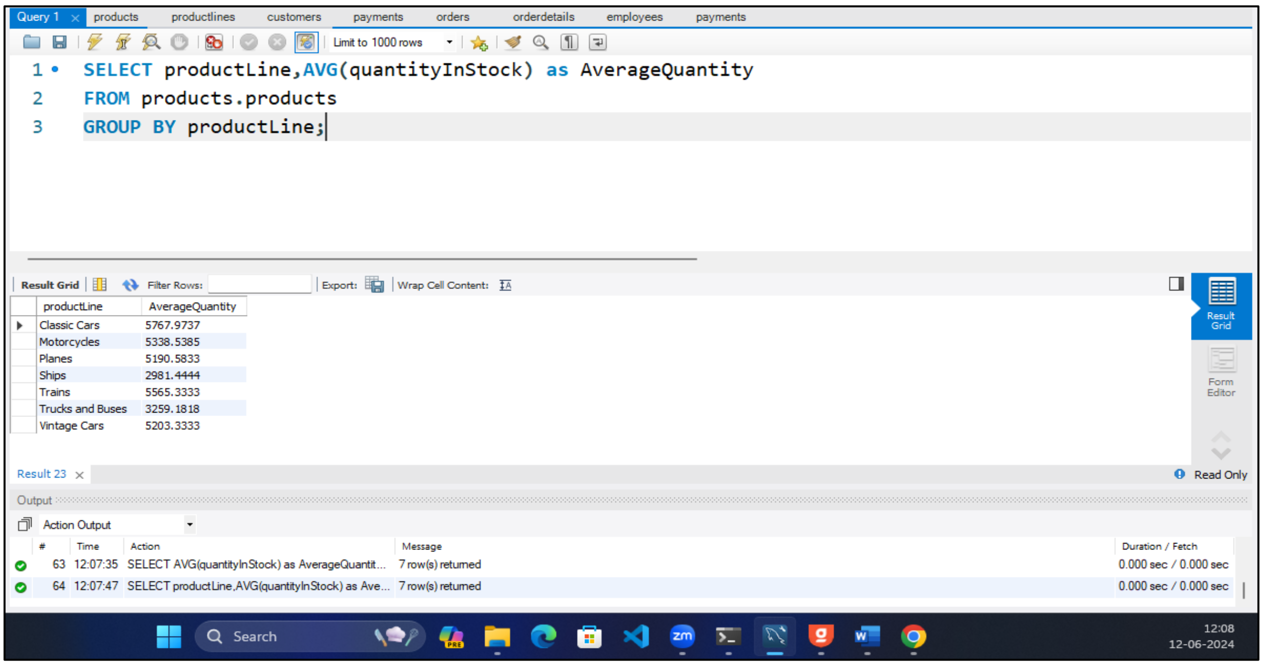
```sql
SELECT c.country, SUM((od.priceEach * od.quantityOrdered)) as SalesRevenue
FROM products.orderdetails od
LEFT JOIN products.orders o
ON od.orderNumber = o.orderNumber
RIGHT JOIN products.customers c
ON c.customerNumber = o.customerNumber
GROUP BY c.country;
```

country	SalesRevenue
France	1007374.02
USA	3273280.05
Australia	562582.59
Norway	270846.30
Poland	NULL
Germany	196470.99
Spain	1099389.09
Sweden	187638.35
Denmark	218994.92

```
#   Time      Action                                           Message              Duration / Fetch
61  12:02:29  SELECT c.country, SUM((od.priceEach * od.quanti...  27 row(s) returned  0.016 sec / 0.000 sec
62  12:02:45  SELECT c.country, SUM((od.priceEach * od.quanti...  27 row(s) returned  0.031 sec / 0.000 sec
```

Q11. Write a query to calculate the average quantity in stock for each product line.

```sql
SELECT productLine,AVG(quantityInStock) as AverageQuantity
FROM products.products
GROUP BY productLine;
```



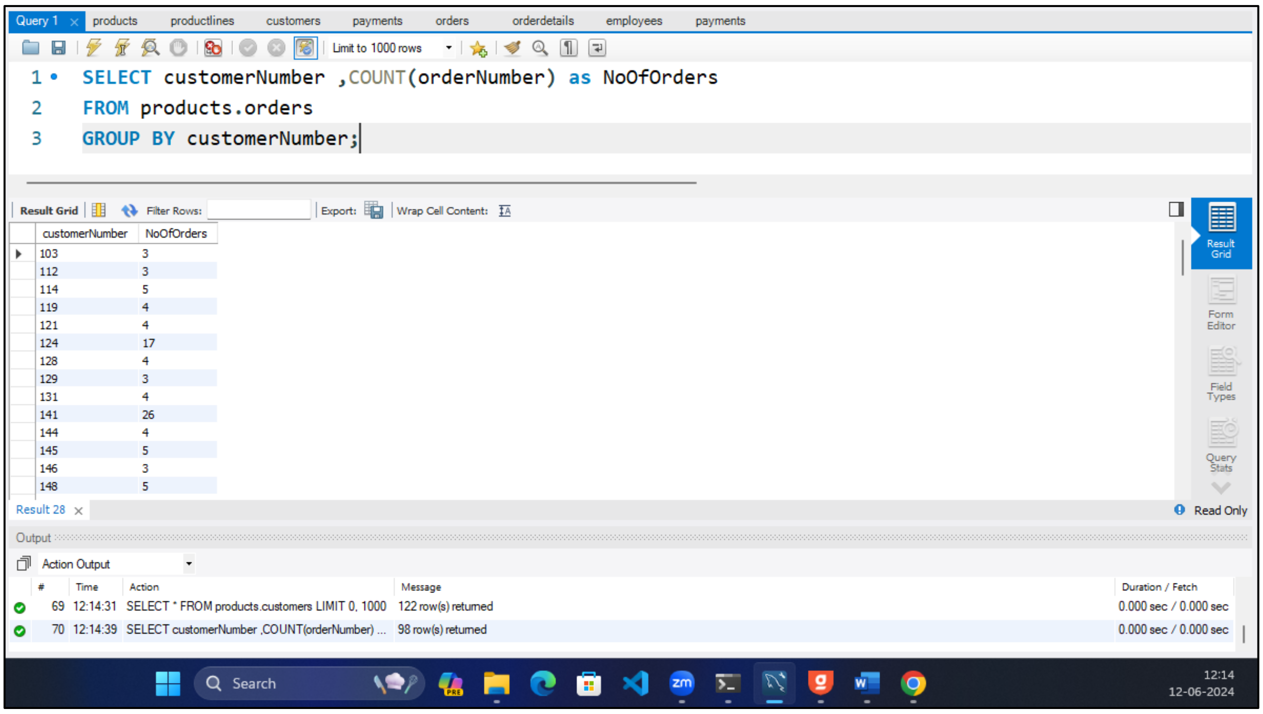
```sql
SELECT productLine,AVG(quantityInStock) as AverageQuantity
FROM products.products
GROUP BY productLine;
```

productLine	AverageQuantity
Classic Cars	5767.9737
Motorcycles	5338.5385
Planes	5190.5833
Ships	2981.4444
Trains	5565.3333
Trucks and Buses	3259.1818
Vintage Cars	5203.3333

```
#   Time      Action                                            Message             Duration / Fetch
63  12:07:35  SELECT AVG(quantityInStock) as AverageQuantit...   7 row(s) returned  0.000 sec / 0.000 sec
64  12:07:47  SELECT productLine,AVG(quantityInStock) as Ave...  7 row(s) returned  0.000 sec / 0.000 sec
```

Q12. Write a query to determine the total number of orders placed by each customer.

```sql
SELECT customerNumber ,COUNT(orderNumber) as NoOfOrders
FROM products.orders
GROUP BY customerNumber;
```



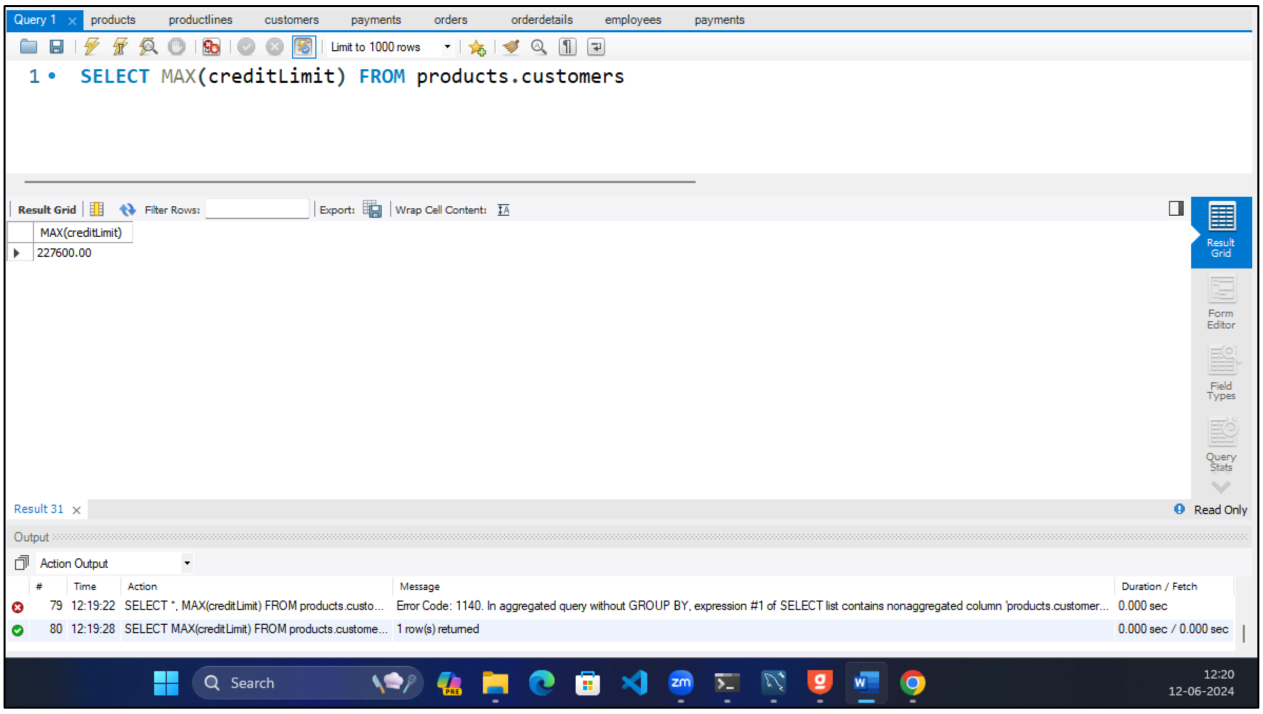
```sql
SELECT customerNumber ,COUNT(orderNumber) as NoOfOrders
FROM products.orders
GROUP BY customerNumber;
```

customerNumber	NoOfOrders
103	3
112	3
114	5
119	4
121	4
124	17
128	4
129	3
131	4
141	26
144	4
145	5
146	3
148	5

#	Time	Action	Message	Duration / Fetch
69	12:14:31	SELECT * FROM products.customers LIMIT 0, 1000	122 row(s) returned	0.000 sec / 0.000 sec
70	12:14:39	SELECT customerNumber ,COUNT(orderNumber) ...	98 row(s) returned	0.000 sec / 0.000 sec

Q13. Write a query to find the maximum credit limit among all customers.

```sql
SELECT MAX(creditLimit) FROM products.customers
```



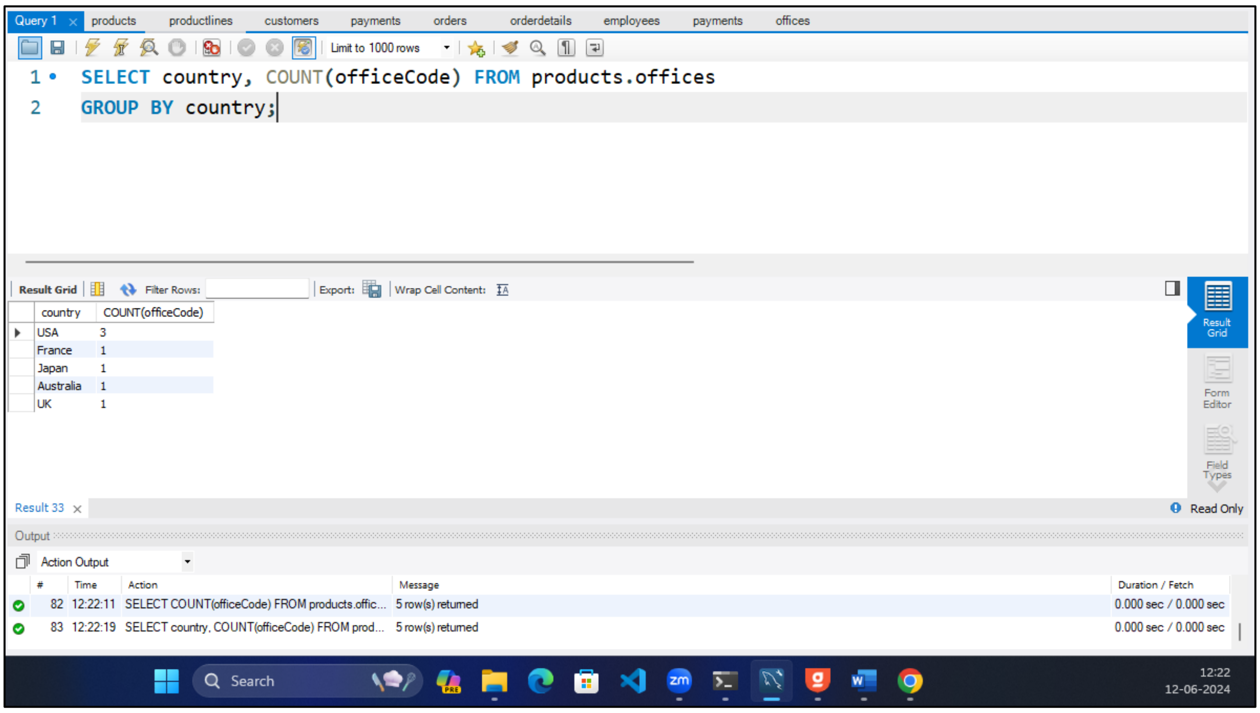
```sql
SELECT MAX(creditLimit) FROM products.customers
```

MAX(creditLimit)
227600.00

#	Time	Action	Message	Duration / Fetch
79	12:19:22	SELECT *, MAX(creditLimit) FROM products.custo...	Error Code: 1140. In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'products.customer...	0.000 sec
80	12:19:28	SELECT MAX(creditLimit) FROM products.custome...	1 row(s) returned	0.000 sec / 0.000 sec

Q14. Write a query to count the number of offices in each country.

```sql
SELECT country, COUNT(officeCode) FROM products.offices
GROUP BY country;
```



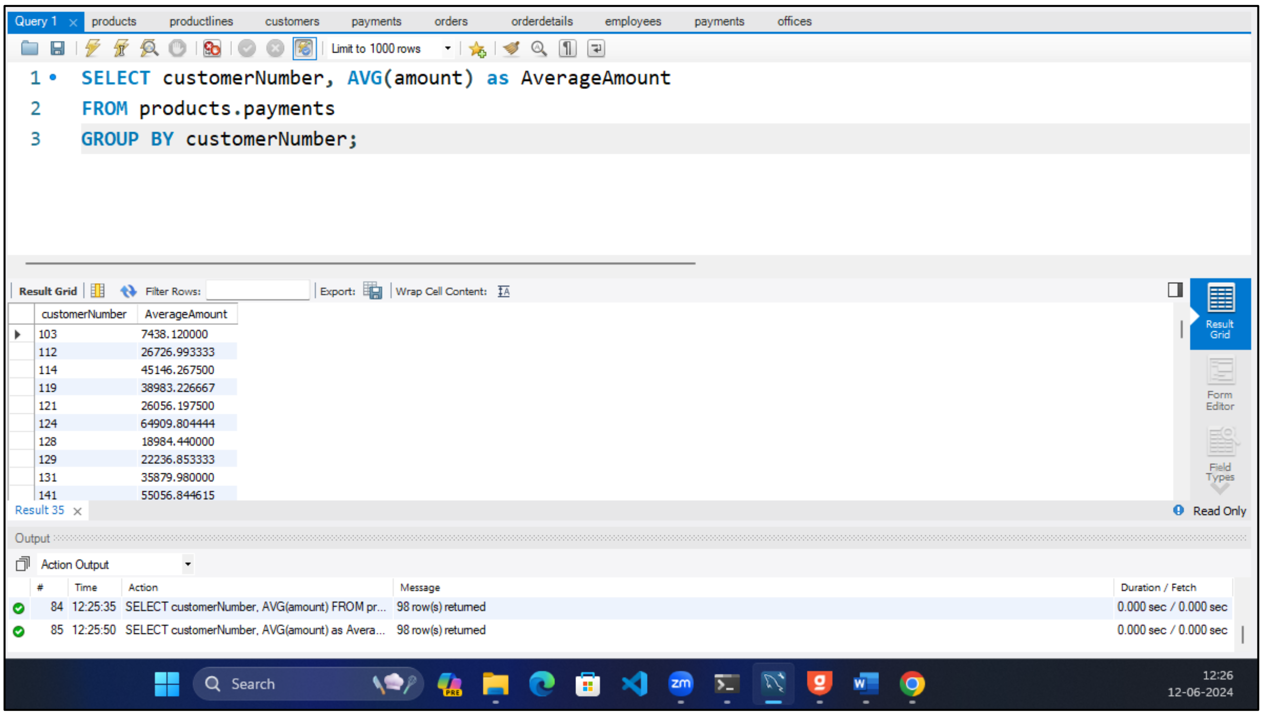
```sql
SELECT country, COUNT(officeCode) FROM products.offices
GROUP BY country;
```

country	COUNT(officeCode)
USA	3
France	1
Japan	1
Australia	1
UK	1

```
#   Time      Action                                                Message            Duration / Fetch
82  12:22:11  SELECT COUNT(officeCode) FROM products.offic...       5 row(s) returned  0.000 sec / 0.000 sec
83  12:22:19  SELECT country, COUNT(officeCode) FROM prod...        5 row(s) returned  0.000 sec / 0.000 sec
```

Q15. Write a query to calculate the average payment amount for each customer.

```sql
SELECT customerNumber, AVG(amount) as AverageAmount
FROM products.payments
GROUP BY customerNumber;
```



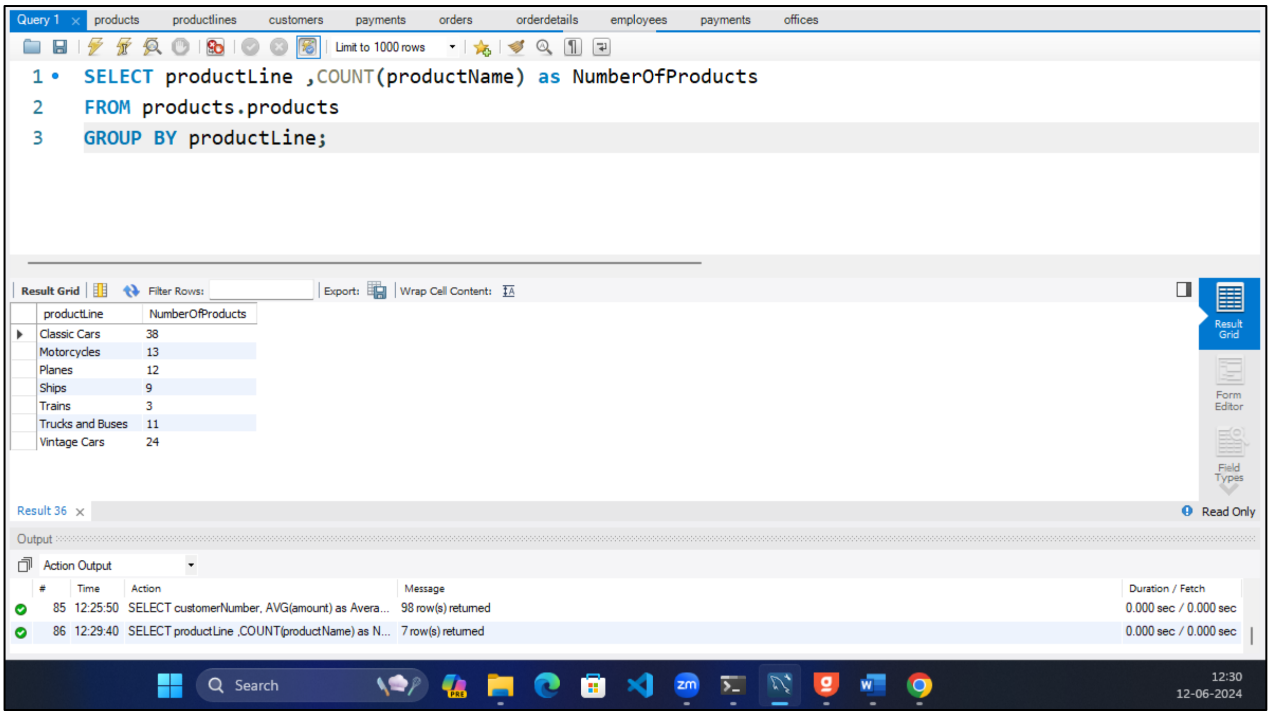
```sql
SELECT customerNumber, AVG(amount) as AverageAmount
FROM products.payments
GROUP BY customerNumber;
```

customerNumber	AverageAmount
103	7438.120000
112	26726.993333
114	45146.267500
119	38983.226667
121	26056.197500
124	64909.804444
128	18984.440000
129	22236.853333
131	35879.980000
141	55056.844615

```
#   Time      Action                                                Message             Duration / Fetch
84  12:25:35  SELECT customerNumber, AVG(amount) FROM pr...         98 row(s) returned  0.000 sec / 0.000 sec
85  12:25:50  SELECT customerNumber, AVG(amount) as Avera...        98 row(s) returned  0.000 sec / 0.000 sec
```

Q16. Write a query to determine the number of products in each product line.

```sql
SELECT productLine ,COUNT(productName) as NumberOfProducts
FROM products.products
GROUP BY productLine;
```



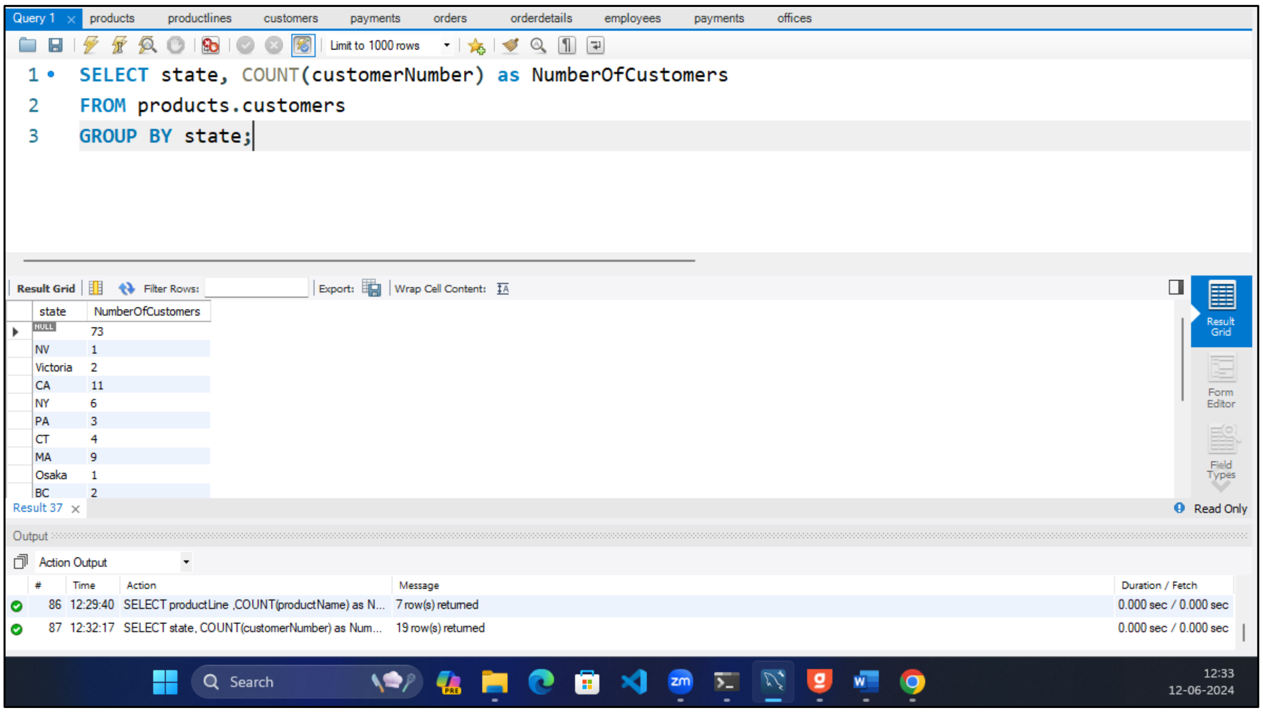
```sql
SELECT productLine ,COUNT(productName) as NumberOfProducts
FROM products.products
GROUP BY productLine;
```

productLine	NumberOfProducts
Classic Cars	38
Motorcycles	13
Planes	12
Ships	9
Trains	3
Trucks and Buses	11
Vintage Cars	24

#	Time	Action	Message	Duration / Fetch
85	12:25:50	SELECT customerNumber, AVG(amount) as Avera...	98 row(s) returned	0.000 sec / 0.000 sec
86	12:29:40	SELECT productLine ,COUNT(productName) as N...	7 row(s) returned	0.000 sec / 0.000 sec

Q17. Write a query to count the number of customers in each state.

```sql
SELECT state, COUNT(customerNumber) as NumberOfCustomers
FROM products.customers
GROUP BY state;
```



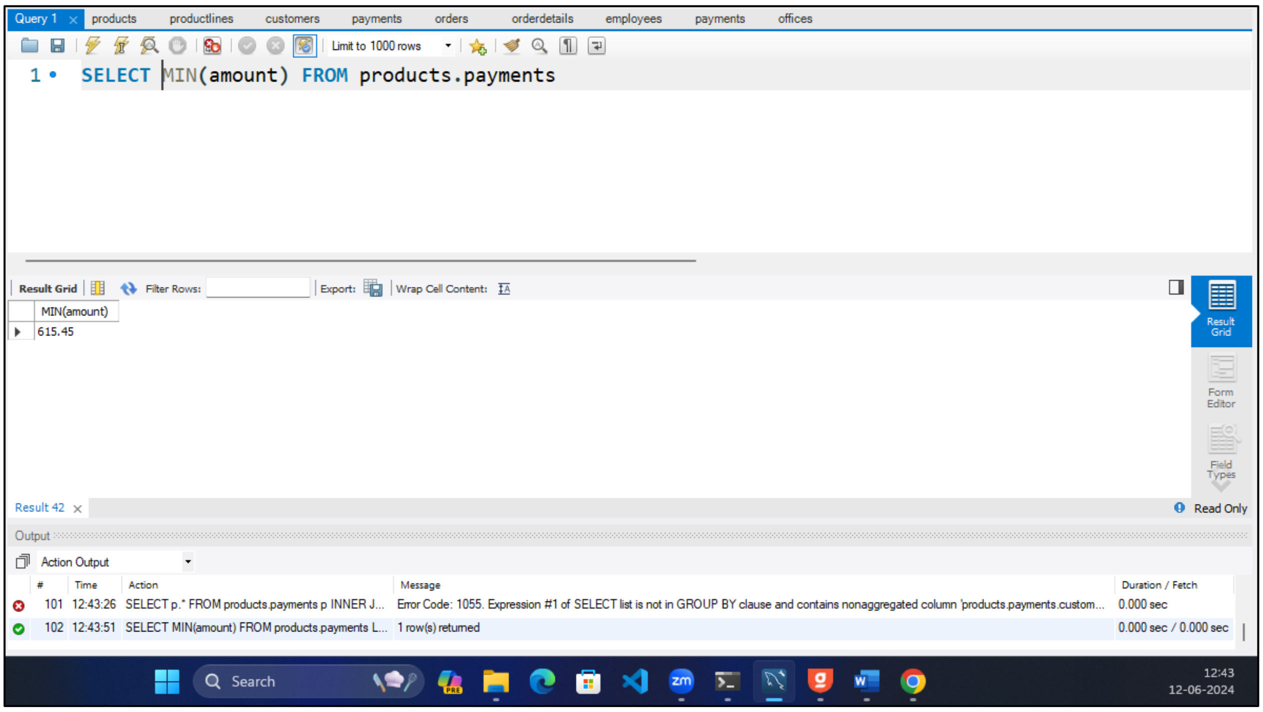
```sql
SELECT state, COUNT(customerNumber) as NumberOfCustomers
FROM products.customers
GROUP BY state;
```

state	NumberOfCustomers
NULL	73
NV	1
Victoria	2
CA	11
NY	6
PA	3
CT	4
MA	9
Osaka	1
BC	2

#	Time	Action	Message	Duration / Fetch
86	12:29:40	SELECT productLine ,COUNT(productName) as N...	7 row(s) returned	0.000 sec / 0.000 sec
87	12:32:17	SELECT state, COUNT(customerNumber) as Num...	19 row(s) returned	0.000 sec / 0.000 sec

Q18. Write a query to find the minimum payment amount among all customers.

```
SELECT MIN(amount) FROM products.payments
```



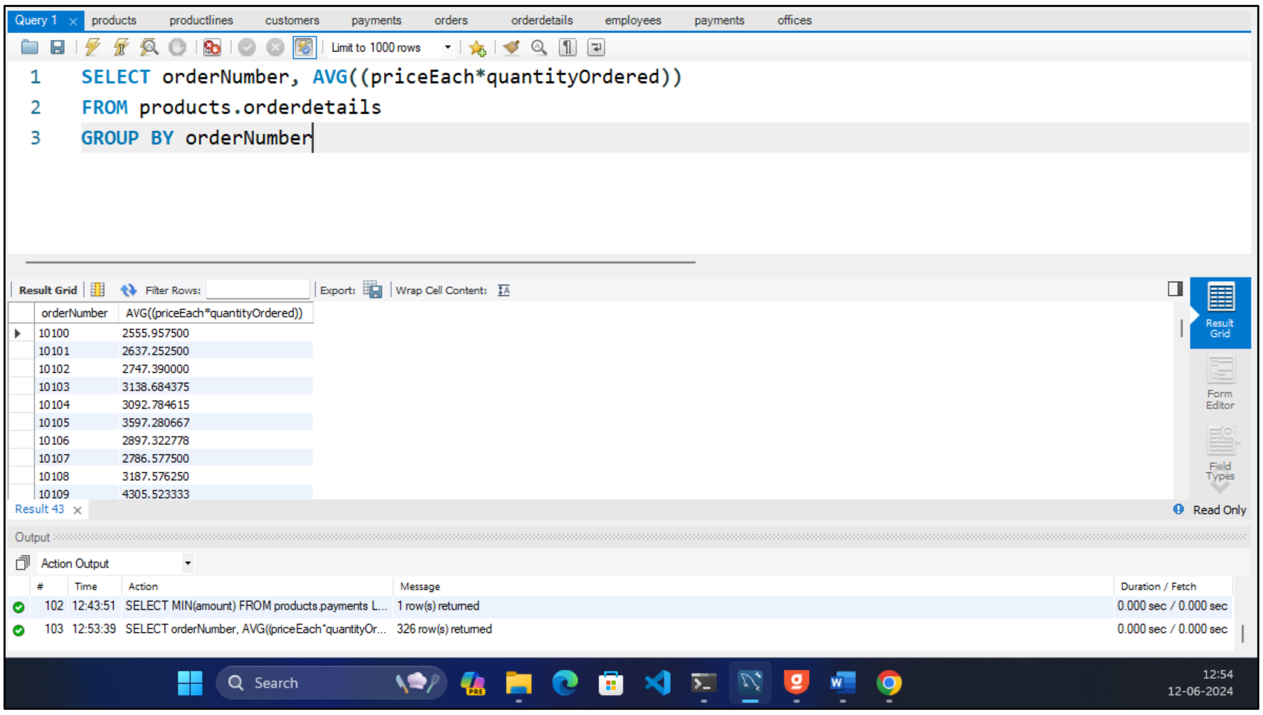
```
SELECT MIN(amount) FROM products.payments
```

MIN(amount)
615.45

```
#    Time      Action                                           Message                                                                                              Duration / Fetch
101  12:43:26  SELECT p.* FROM products.payments p INNER J...   Error Code: 1055. Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'products.payments.custom...   0.000 sec
102  12:43:51  SELECT MIN(amount) FROM products.payments L...   1 row(s) returned                                                                                    0.000 sec / 0.000 sec
```

Q19. Write a query to calculate the average sales revenue per order.

```
SELECT orderNumber, AVG((priceEach*quantityOrdered))
FROM products.orderdetails
GROUP BY orderNumber
```



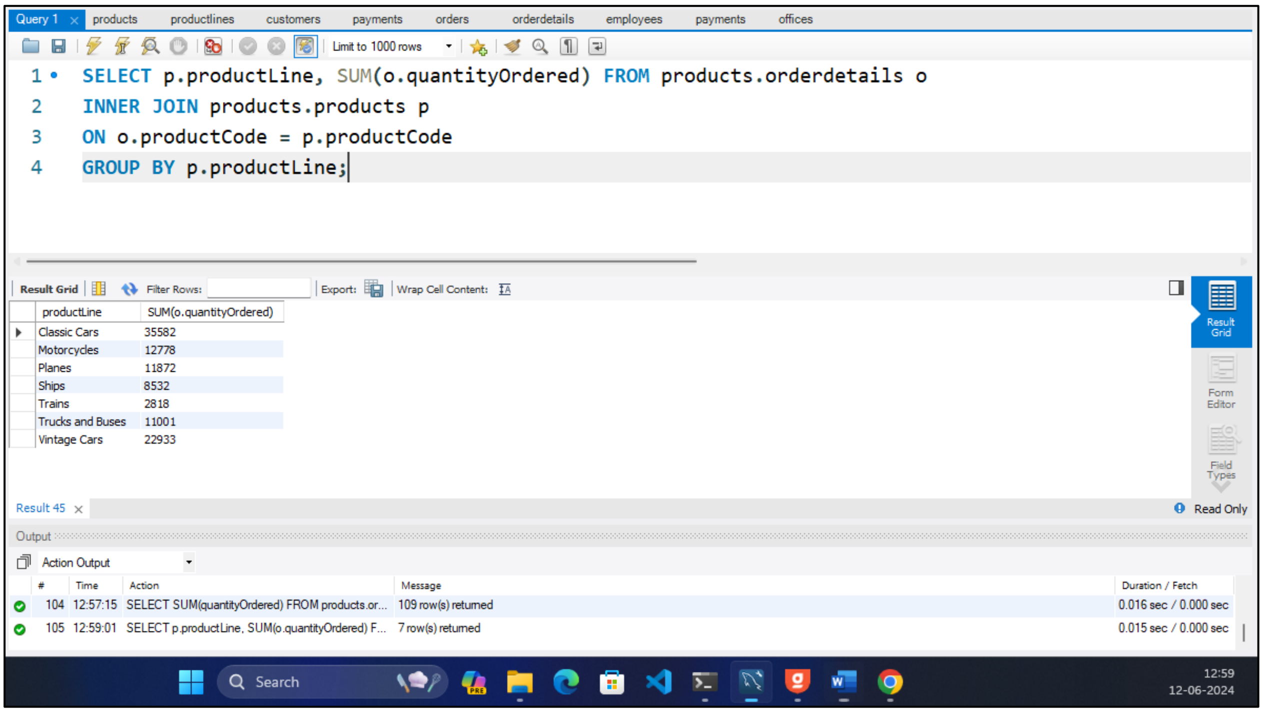
```
SELECT orderNumber, AVG((priceEach*quantityOrdered))
FROM products.orderdetails
GROUP BY orderNumber
```

orderNumber	AVG((priceEach*quantityOrdered))
10100	2555.957500
10101	2637.252500
10102	2747.390000
10103	3138.684375
10104	3092.784615
10105	3597.280667
10106	2897.322778
10107	2786.577500
10108	3187.576250
10109	4305.523333

```
#    Time      Action                                           Message              Duration / Fetch
102  12:43:51  SELECT MIN(amount) FROM products.payments L...   1 row(s) returned    0.000 sec / 0.000 sec
103  12:53:39  SELECT orderNumber, AVG((priceEach*quantityOr...  326 row(s) returned  0.000 sec / 0.000 sec
```

Q20. Write a query to determine the total quantity ordered for each product line.

```sql
SELECT p.productLine, SUM(o.quantityOrdered) FROM products.orderde-
tails o
INNER JOIN products.products p
ON o.productCode = p.productCode
GROUP BY p.productLine;
```



```sql
SELECT p.productLine, SUM(o.quantityOrdered) FROM products.orderdetails o
INNER JOIN products.products p
ON o.productCode = p.productCode
GROUP BY p.productLine;
```

productLine	SUM(o.quantityOrdered)
Classic Cars	35582
Motorcycles	12778
Planes	11872
Ships	8532
Trains	2818
Trucks and Buses	11001
Vintage Cars	22933

#	Time	Action	Message	Duration / Fetch
104	12:57:15	SELECT SUM(quantityOrdered) FROM products.or...	109 row(s) returned	0.016 sec / 0.000 sec
105	12:59:01	SELECT p.productLine, SUM(o.quantityOrdered) F...	7 row(s) returned	0.015 sec / 0.000 sec