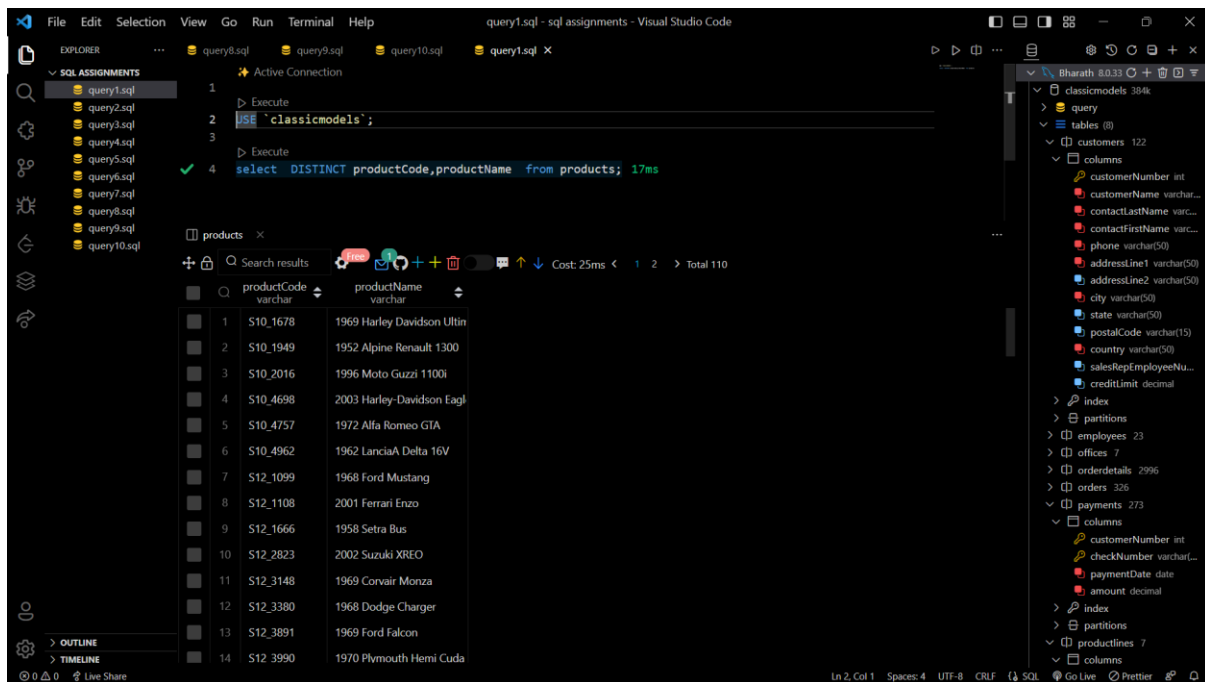


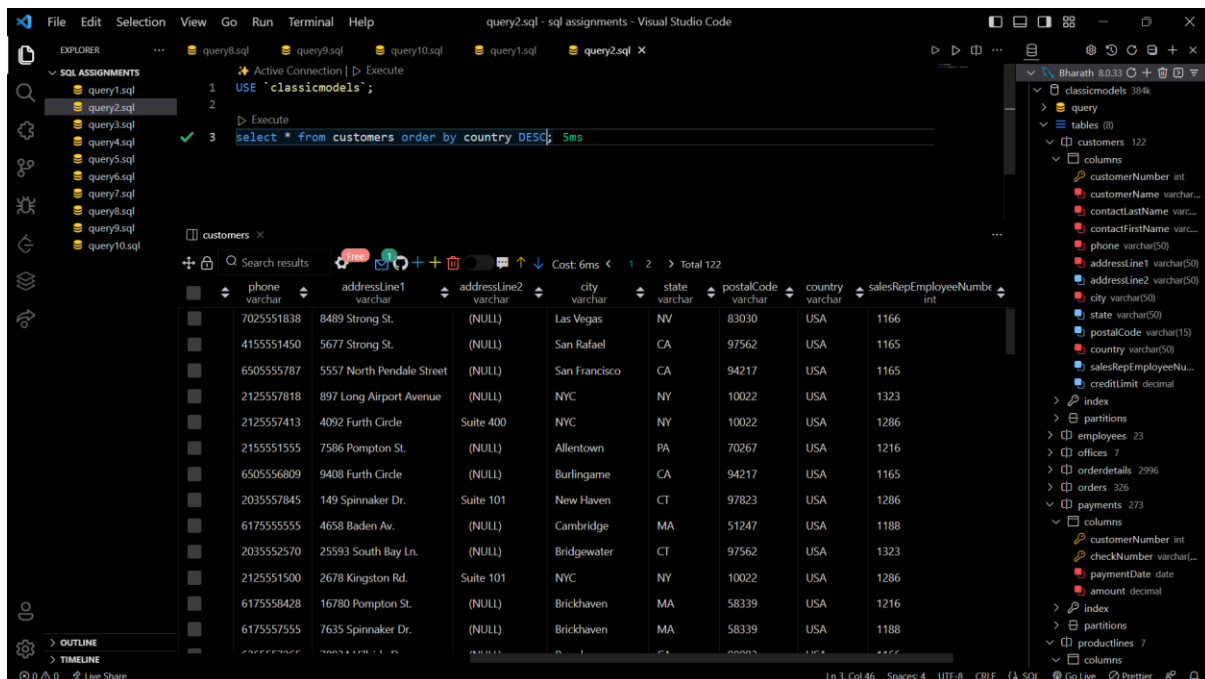
Write the query to get only the product Code and product name from the product code.



```
1 1 > Execute
2 2 USE `classicmodels`;
3 3
4 4 select DISTINCT productCode,productName from products; 17ms
```

	productCode	productName
1	S10_1678	1969 Harley Davidson Ultr
2	S10_1949	1952 Alpine Renault 1300
3	S10_2016	1996 Moto Guzzi 1100i
4	S10_4698	2003 Harley-Davidson Eagl
5	S10_4757	1972 Alfa Romeo GTA
6	S10_4962	1962 Lancia Delta 16V
7	S12_1099	1968 Ford Mustang
8	S12_1108	2001 Ferrari Enzo
9	S12_1666	1958 Setra Bus
10	S12_2823	2002 Suzuki XREO
11	S12_3148	1969 Corvair Monza
12	S12_3380	1968 Dodge Charger
13	S12_3891	1969 Ford Falcon
14	S12_3990	1970 Plymouth Hemi Cuda

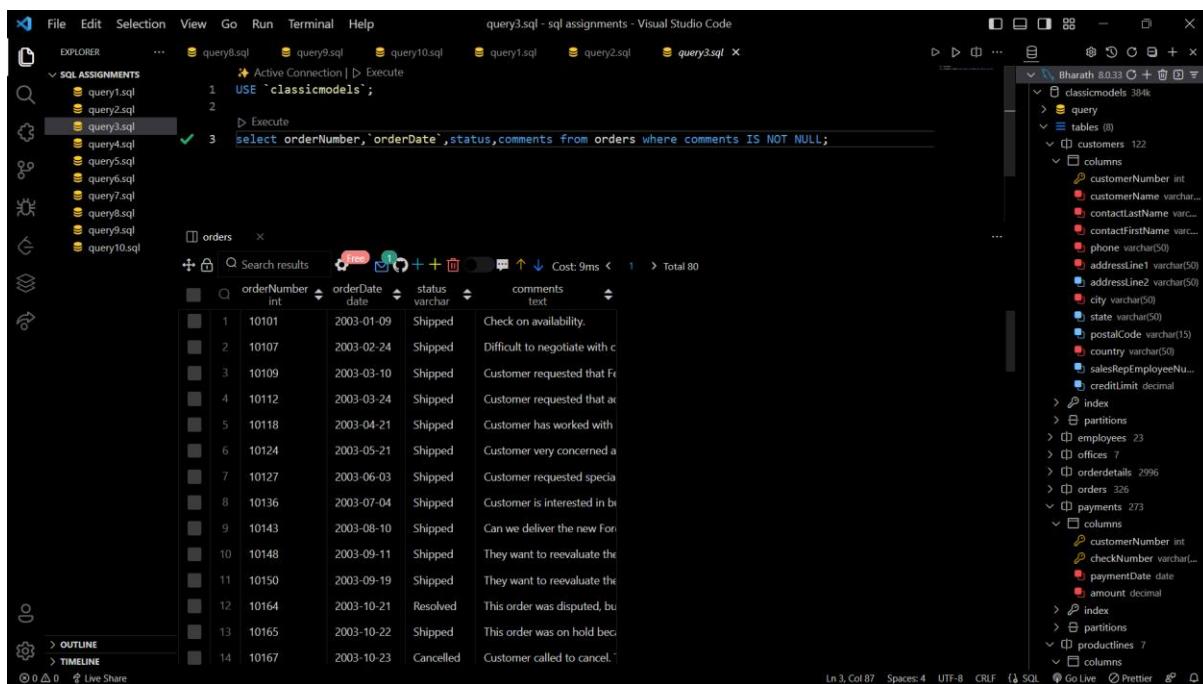
Write the query to arrange the customer details based on the country name in descending order.



```
1 1 USE `classicmodels`;
2 2
3 3 select * from customers order by country DESC; 5ms
```

phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumbe
7025551838	8489 Strong St.	(NULL)	Las Vegas	NV	83030	USA	1166
4155551450	5677 Strong St.	(NULL)	San Rafael	CA	97562	USA	1165
6505555787	5557 North Pendale Street	(NULL)	San Francisco	CA	94217	USA	1165
2125557818	897 Long Airport Avenue	(NULL)	NYC	NY	10022	USA	1323
2125557413	4092 Furth Circle	Suite 400	NYC	NY	10022	USA	1286
2155551555	7586 Pompton St.	(NULL)	Allentown	PA	70267	USA	1216
6505556809	9408 Furth Circle	(NULL)	Burlingame	CA	94217	USA	1165
2035557845	149 Spinnaker Dr.	Suite 101	New Haven	CT	97823	USA	1286
6175555555	4658 Baden Av.	(NULL)	Cambridge	MA	51247	USA	1188
2035552570	25593 South Bay Ln.	(NULL)	Bridgewater	CT	97562	USA	1323
2125551500	2678 Kingston Rd.	Suite 101	NYC	NY	10022	USA	1286
6175558428	16780 Pompton St.	(NULL)	Brickhaven	MA	58339	USA	1216
6175557555	7635 Spinnaker Dr.	(NULL)	Brickhaven	MA	58339	USA	1188

Write the query to find the order number, order date, and status for the customers having comments about shipment.



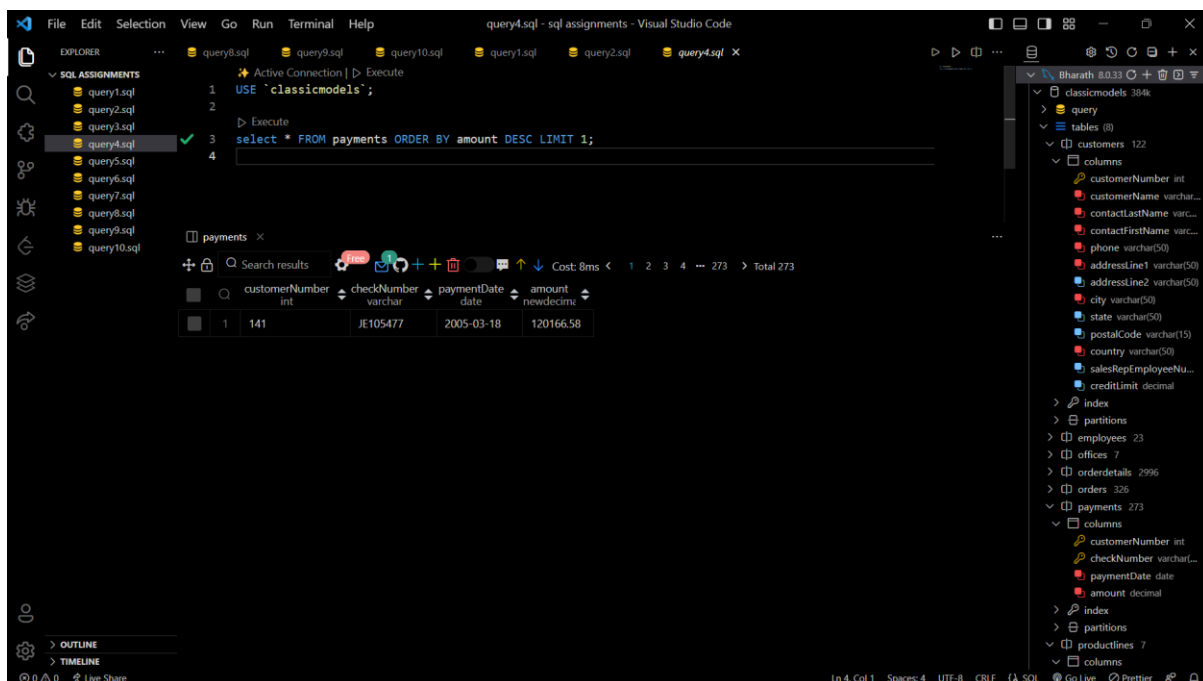
The screenshot shows the Visual Studio Code interface with a SQL query in `query3.sql`. The query is:

```
USE `classicmodels`;
select orderNumber,`orderDate`,status,comments from orders where comments IS NOT NULL;
```

The results are displayed in the `orders` table, showing 14 rows of data. The columns are `orderNumber`, `orderDate`, `status`, and `comments`.

	orderNumber	orderDate	status	comments
1	10101	2003-01-09	Shipped	Check on availability.
2	10107	2003-02-24	Shipped	Difficult to negotiate with c
3	10109	2003-03-10	Shipped	Customer requested that Fe
4	10112	2003-03-24	Shipped	Customer requested that a
5	10118	2003-04-21	Shipped	Customer has worked with
6	10124	2003-05-21	Shipped	Customer very concerned a
7	10127	2003-06-03	Shipped	Customer requested specia
8	10136	2003-07-04	Shipped	Customer is interested in b
9	10143	2003-08-10	Shipped	Can we deliver the new For
10	10148	2003-09-11	Shipped	They want to reevaluate the
11	10150	2003-09-19	Shipped	They want to reevaluate the
12	10164	2003-10-21	Resolved	This order was disputed, bu
13	10165	2003-10-22	Shipped	This order was on hold bec
14	10167	2003-10-23	Cancelled	Customer called to cancel.

Write the query to find the customer who has made the highest payment along with the payment date.



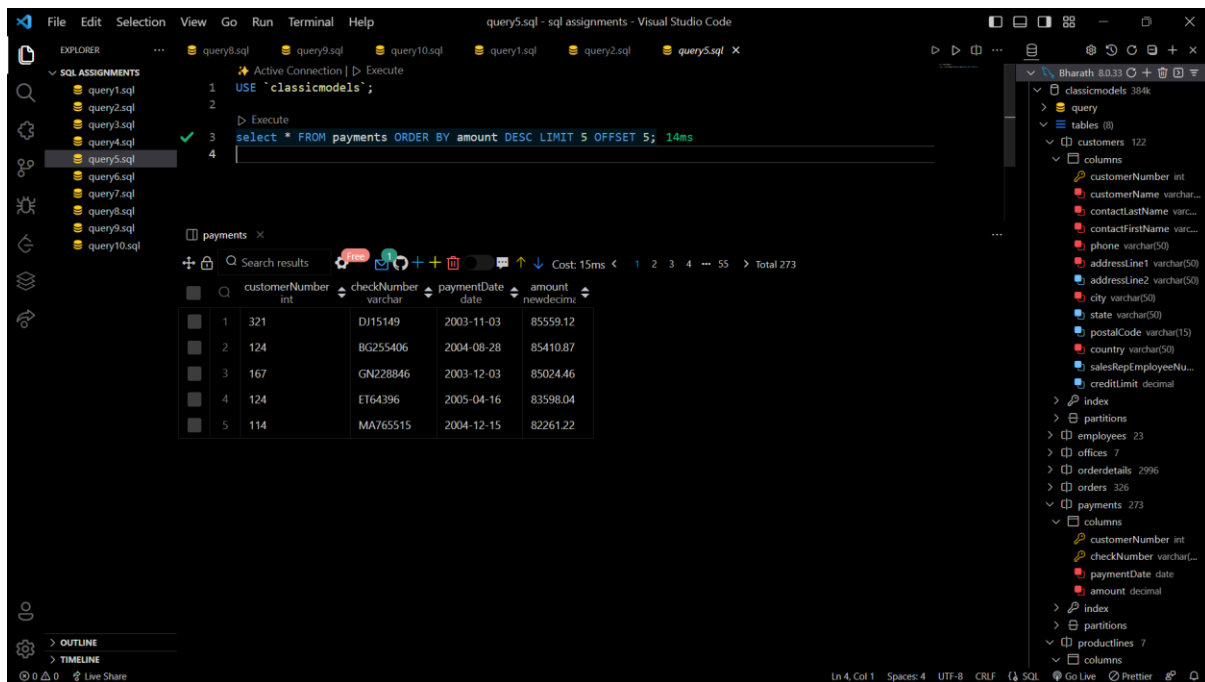
The screenshot shows the Visual Studio Code interface with a SQL query in `query4.sql`. The query is:

```
USE `classicmodels`;
select * FROM payments ORDER BY amount DESC LIMIT 1;
```

The results are displayed in the `payments` table, showing 1 row of data. The columns are `customerNumber`, `checkNumber`, `paymentDate`, and `amount`.

	customerNumber	checkNumber	paymentDate	amount
1	141	JE105477	2005-03-18	120166.58

I have a list of the top five customers who made the highest payment and write a query to find the next top five customers who made the highest payment.



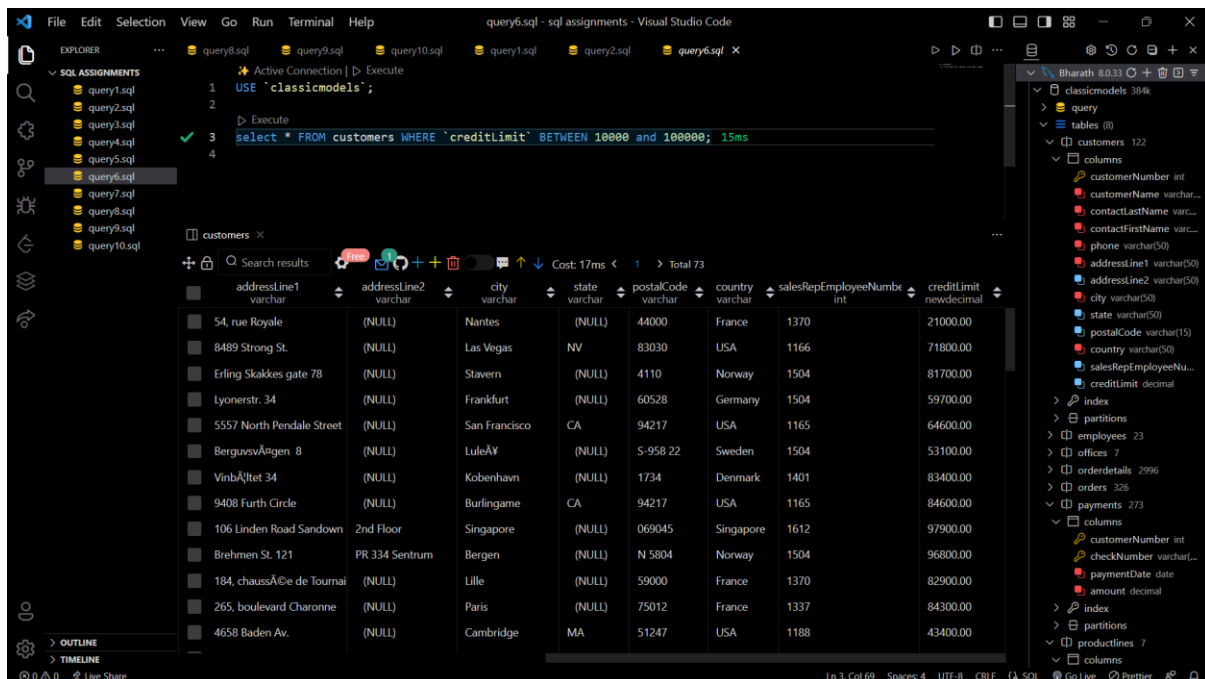
The screenshot shows the Visual Studio Code interface with a SQL query in the editor. The query is:

```
USE 'classicmodels';
select * FROM payments ORDER BY amount DESC LIMIT 5 OFFSET 5; 14ms
```

The results table displays the following data:

	customerNumber	checkNumber	paymentDate	amount
1	321	DJ15149	2003-11-03	85559.12
2	124	BG255406	2004-08-28	85410.87
3	167	GN228846	2003-12-03	85024.46
4	124	ET64396	2005-04-16	83598.04
5	114	MA765515	2004-12-15	82261.22

Write the query to find the customer details whose credit limit is between 10,000 to 1,00,000.



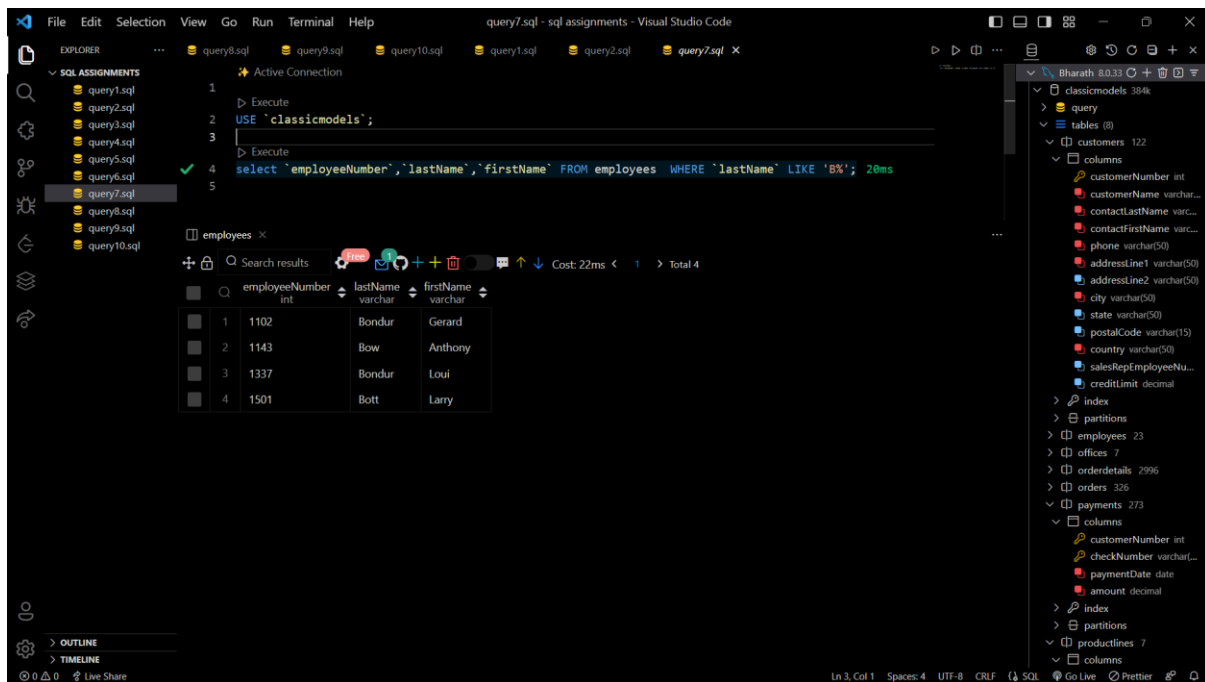
The screenshot shows the Visual Studio Code interface with a SQL query in the editor. The query is:

```
USE 'classicmodels';
select * FROM customers WHERE 'creditLimit' BETWEEN 10000 and 100000; 15ms
```

The results table displays the following data:

	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
54, rue Royale	(NULL)	Nantes	(NULL)	44000	France	1370	21000.00	
8489 Strong St.	(NULL)	Las Vegas	NV	83030	USA	1166	71800.00	
Erling Skakkes gate 78	(NULL)	Stavern	(NULL)	4110	Norway	1504	81700.00	
Lyonerstr. 34	(NULL)	Frankfurt	(NULL)	60528	Germany	1504	59700.00	
5557 North Pendale Street	(NULL)	San Francisco	CA	94217	USA	1165	64600.00	
Berguvsvägen 8	(NULL)	Luleå	(NULL)	S-958 22	Sweden	1504	53100.00	
Vinbållstet 34	(NULL)	København	(NULL)	1734	Denmark	1401	83400.00	
9408 Furth Circle	(NULL)	Burlingame	CA	94217	USA	1165	84600.00	
106 Linden Road Sandown	2nd Floor	Singapore	(NULL)	069045	Singapore	1612	97900.00	
Brehmen St. 121	PR 334 Sentrum	Bergen	(NULL)	N 5804	Norway	1504	96800.00	
184, chaussée de Tournai	(NULL)	Lille	(NULL)	59000	France	1370	82900.00	
265, boulevard Charonne	(NULL)	Paris	(NULL)	75012	France	1337	84300.00	
4658 Baden Av.	(NULL)	Cambridge	MA	51247	USA	1188	43400.00	

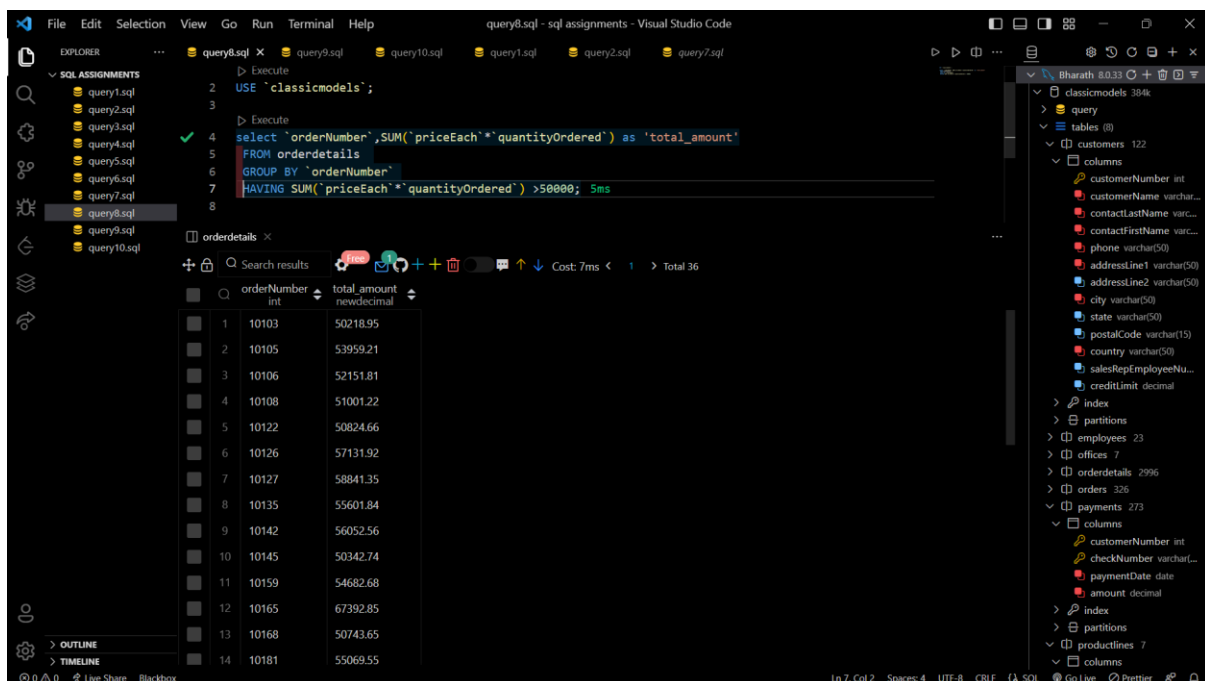
Write the query to get the Employee number, lastname, and first name from the employees' table whose last name starts with 'B'.



```
1 -- Active Connection
2 USE 'classicmodels';
3
4 -- Execute
5 select 'employeeNumber','lastName','firstName' FROM employees WHERE 'lastName' LIKE 'B%'; 20ms
```

	employeeNumber int	lastName varchar	firstName varchar
1	1102	Bondur	Gerard
2	1143	Bow	Anthony
3	1337	Bondur	Loui
4	1501	Bott	Larry

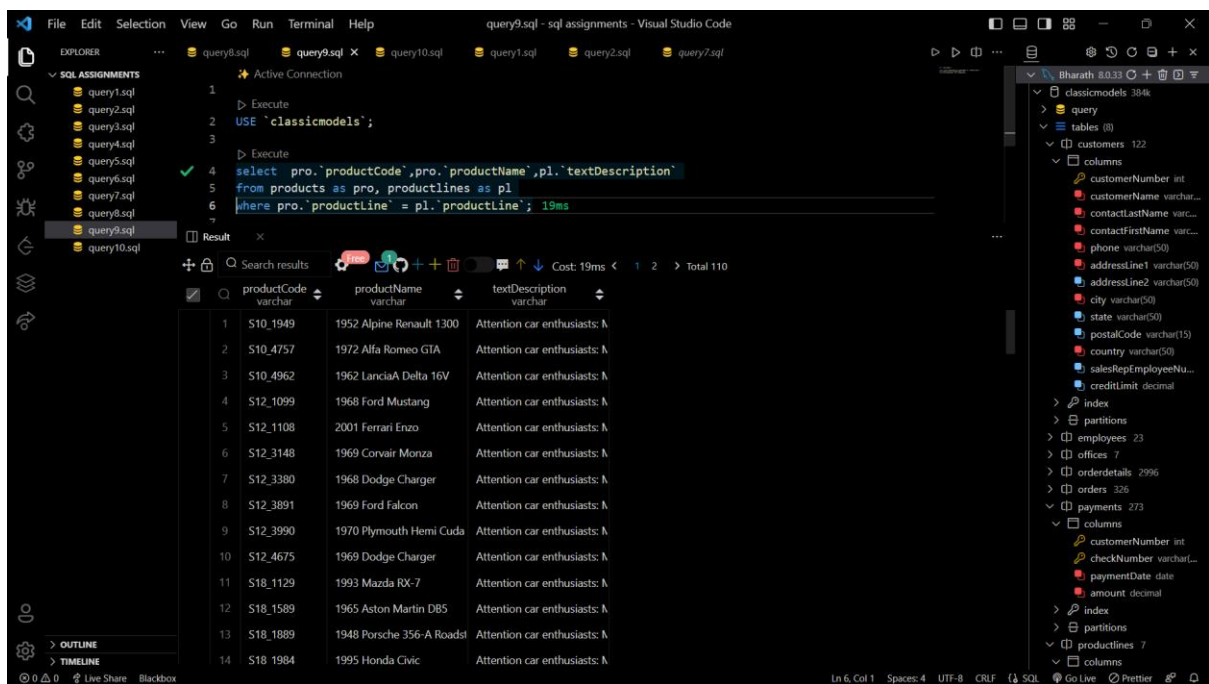
write a query to select the order whose total amount is greater than 50,000.



```
2 -- Execute
3 USE 'classicmodels';
4
5 -- Execute
6 select 'orderNumber',SUM('priceEach'*quantityOrdered') as 'total_amount'
7 FROM orderdetails
8 GROUP BY 'orderNumber'
9 HAVING SUM('priceEach'*quantityOrdered') >50000; 5ms
```

	orderNumber int	total_amount newdecimal
1	10103	50218.95
2	10105	53959.21
3	10106	52151.81
4	10108	51001.22
5	10122	50824.66
6	10126	57131.92
7	10127	58841.35
8	10135	55601.84
9	10142	56052.56
10	10145	50342.74
11	10159	54682.68
12	10165	67392.85
13	10168	50743.65
14	10181	55069.55

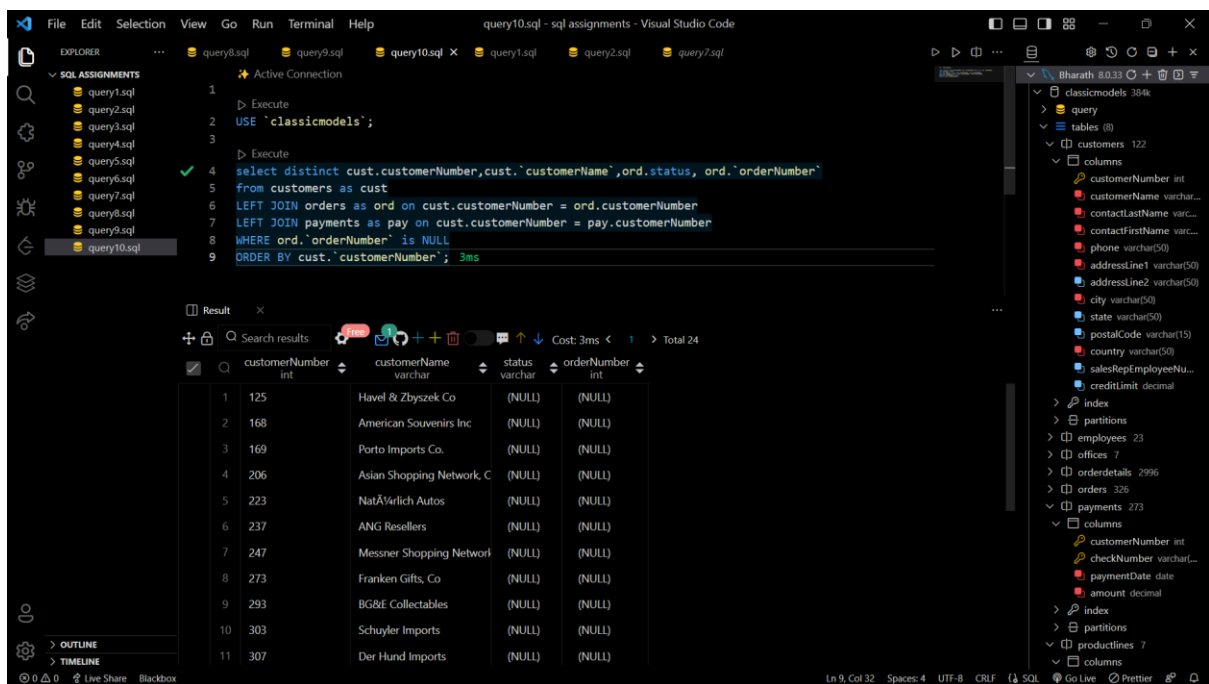
Write the query to find the productcode, product name & text description from the tables products and product lines.



```
1 1. Active Connection
2 2. USE 'classicmodels';
3 3.
4 4. select pro.`productCode`,pro.`productName`,pl.`textDescription`
5 5. from products as pro, productlines as pl
6 6. where pro.`productline` = pl.`productline`; 19ms
```

	productCode	productName	textDescription
1	S10_1949	1952 Alpine Renault 1300	Attention car enthusiasts: N
2	S10_4757	1972 Alfa Romeo GTA	Attention car enthusiasts: N
3	S10_4962	1962 Lancia Delta 16V	Attention car enthusiasts: N
4	S12_1099	1968 Ford Mustang	Attention car enthusiasts: N
5	S12_1108	2001 Ferrari Enzo	Attention car enthusiasts: N
6	S12_3148	1969 Corvair Monza	Attention car enthusiasts: N
7	S12_3380	1968 Dodge Charger	Attention car enthusiasts: N
8	S12_3891	1969 Ford Falcon	Attention car enthusiasts: N
9	S12_3990	1970 Plymouth Hemi Cuda	Attention car enthusiasts: N
10	S12_4675	1969 Dodge Charger	Attention car enthusiasts: N
11	S18_1129	1993 Mazda RX-7	Attention car enthusiasts: N
12	S18_1589	1965 Aston Martin DB5	Attention car enthusiasts: N
13	S18_1889	1948 Porsche 356-A Roadster	Attention car enthusiasts: N
14	S18_1984	1995 Honda Civic	Attention car enthusiasts: N

Write the query to get the customer number, customer name, order number, status from the tables order, payments, and customers who have no order.



```
1 1. Active Connection
2 2. USE 'classicmodels';
3 3.
4 4. select distinct cust.`customerNumber`,cust.`customerName`,ord.`status`,ord.`orderNumber`
5 5. from customers as cust
6 6. LEFT JOIN orders as ord on cust.`customerNumber` = ord.`customerNumber`
7 7. LEFT JOIN payments as pay on cust.`customerNumber` = pay.`customerNumber`
8 8. WHERE ord.`orderNumber` is NULL
9 9. ORDER BY cust.`customerNumber`; 3ms
```

	customerNumber	customerName	status	orderNumber
1	125	Havel & Zbyszek Co	(NULL)	(NULL)
2	168	American Souvenirs Inc	(NULL)	(NULL)
3	169	Porto Imports Co.	(NULL)	(NULL)
4	206	Asian Shopping Network, C	(NULL)	(NULL)
5	223	NatÄrlich Autos	(NULL)	(NULL)
6	237	ANG Resellers	(NULL)	(NULL)
7	247	Messner Shopping Network	(NULL)	(NULL)
8	273	Franken Gifts, Co	(NULL)	(NULL)
9	293	BG&E Collectables	(NULL)	(NULL)
10	303	Schuyler Imports	(NULL)	(NULL)
11	307	Der Hund Imports	(NULL)	(NULL)