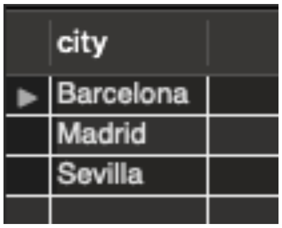
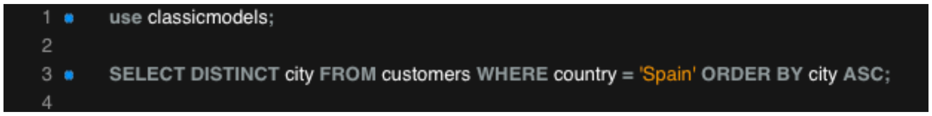
Jackson Lee
2/28/21
Database Systems

Homework 3:

1. List in alphabetical order without duplicates the names of the cities in Spain where ClassicModels has customers. (3)

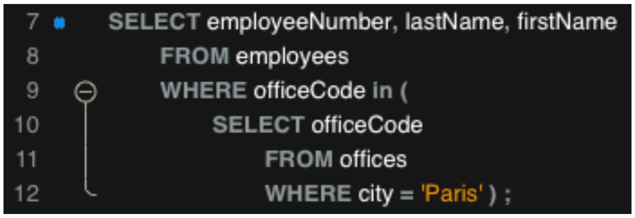


city
Barcelona
Madrid
Sevilla

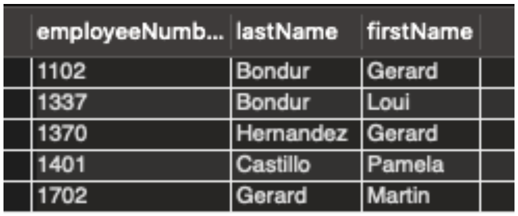


```sql
use classicmodels;

SELECT DISTINCT city FROM customers WHERE country = 'Spain' ORDER BY city ASC;

```

2. List the employee id, last name, first name of each employee who works in Paris (use a subquery). (5)

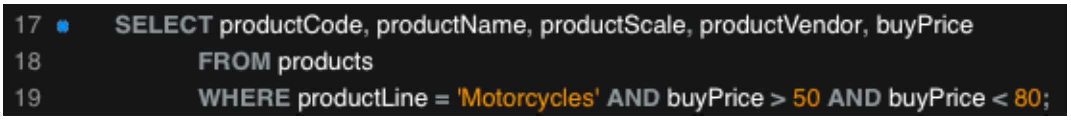


```sql
SELECT employeeNumber, lastName, firstName
    FROM employees
    WHERE officeCode in (
        SELECT officeCode
            FROM offices
            WHERE city = 'Paris' ) ;
```

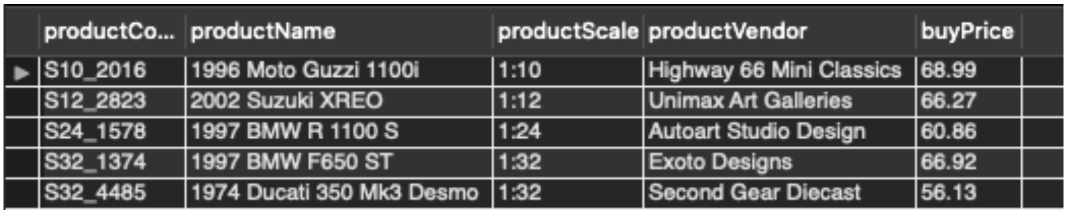


employeeNumb...	lastName	firstName
1102	Bondur	Gerard
1337	Bondur	Loui
1370	Hernandez	Gerard
1401	Castillo	Pamela
1702	Gerard	Martin

3) List the ProductCode, ProductName, ProductScale, ProductVendor, buyPrice for all products that are in the Motorcycles product line and have a buyPrice greater than 50 and less than 80. (5)

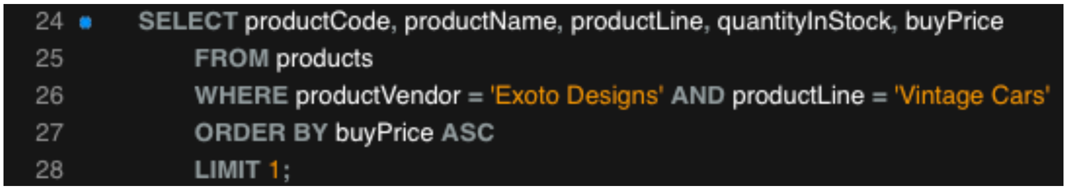


```sql
SELECT productCode, productName, productScale, productVendor, buyPrice
    FROM products
    WHERE productLine = 'Motorcycles' AND buyPrice > 50 AND buyPrice < 80;
```

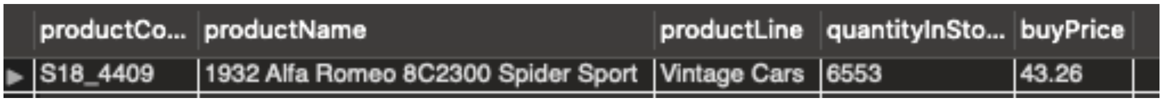


productCo...	productName	productScale	productVendor	buyPrice
S10_2016	1996 Moto Guzzi 1100i	1:10	Highway 66 Mini Classics	68.99
S12_2823	2002 Suzuki XREO	1:12	Unimax Art Galleries	66.27
S24_1578	1997 BMW R 1100 S	1:24	Autoart Studio Design	60.86
S32_1374	1997 BMW F650 ST	1:32	Exoto Designs	66.92
S32_4485	1974 Ducati 350 Mk3 Desmo	1:32	Second Gear Diecast	56.13

4. List the productCode, productName, productLine, quantityInStock, buyPrice for the least expensive Vintage Cars from ExotoDesigns. (1)

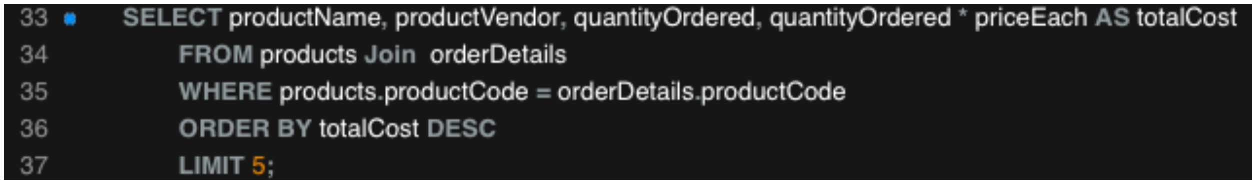


```sql
SELECT productCode, productName, productLine, quantityInStock, buyPrice
    FROM products
    WHERE productVendor = 'Exoto Designs' AND productLine = 'Vintage Cars'
    ORDER BY buyPrice ASC
    LIMIT 1;
```

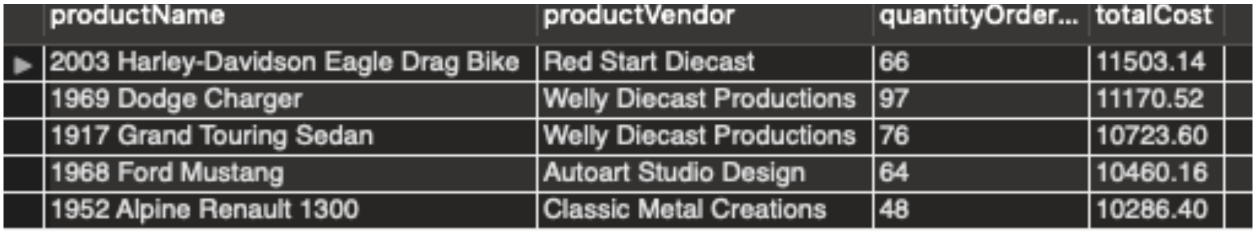


productCo...	productName	productLine	quantityInSto...	buyPrice
S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars	6553	43.26

5. List the top 5 most expensive Order Items (by total cost) listing the product name, the vendor, the quantity and the total cost for each ordered item. (5) HINT: total cost = (quantityOrdered * priceEach)

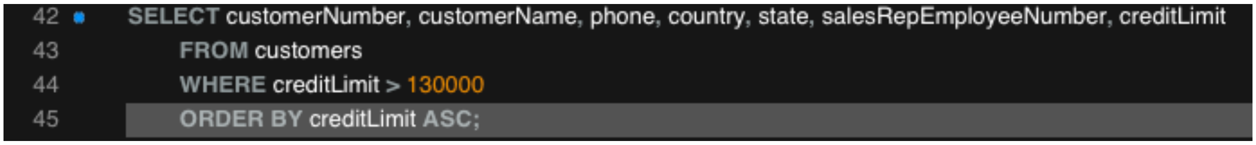


```
SELECT productName, productVendor, quantityOrdered, quantityOrdered * priceEach AS totalCost
	FROM products Join  orderDetails
	WHERE products.productCode = orderDetails.productCode
	ORDER BY totalCost DESC
	LIMIT 5;
```

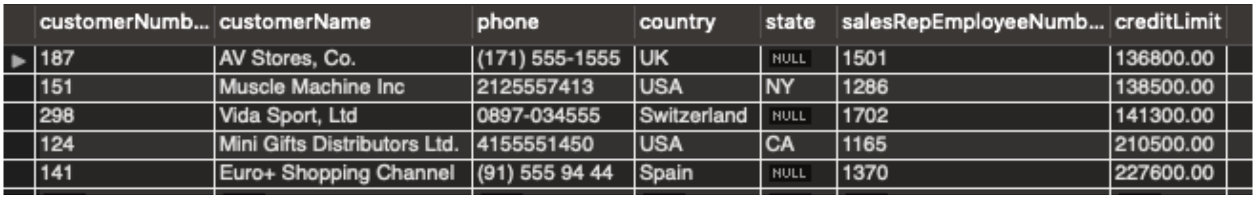


productName	productVendor	quantityOrder...	totalCost
2003 Harley-Davidson Eagle Drag Bike	Red Start Diecast	66	11503.14
1969 Dodge Charger	Welly Diecast Productions	97	11170.52
1917 Grand Touring Sedan	Welly Diecast Productions	76	10723.60
1968 Ford Mustang	Autoart Studio Design	64	10460.16
1952 Alpine Renault 1300	Classic Metal Creations	48	10286.40

6. List the customerNumber, customerName, phone, country, state, corresponding RepEmployee's Number, and creditLimit for customers with a credit limit of more than 130,000. List them in order from lowest to highest creditLimit. (5)

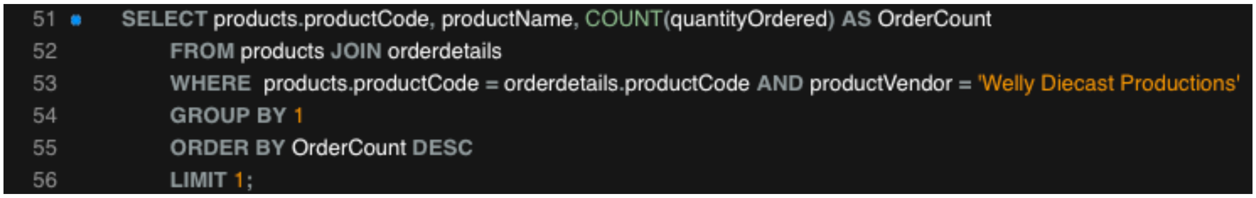


```
SELECT customerNumber, customerName, phone, country, state, salesRepEmployeeNumber, creditLimit
	FROM customers
	WHERE creditLimit > 130000
	ORDER BY creditLimit ASC;
```

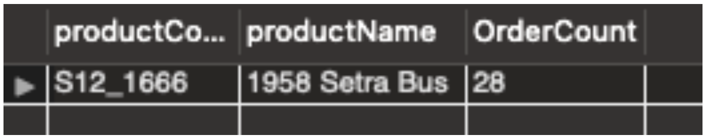


customerNumb...	customerName	phone	country	state	salesRepEmployeeNumb...	creditLimit
187	AV Stores, Co.	(171) 555-1555	UK	NULL	1501	136800.00
151	Muscle Machine Inc	2125557413	USA	NY	1286	138500.00
298	Vida Sport, Ltd	0897-034555	Switzerland	NULL	1702	141300.00
124	Mini Gifts Distributors Ltd.	4155551450	USA	CA	1165	210500.00
141	Euro+ Shopping Channel	(91) 555 94 44	Spain	NULL	1370	227600.00

7. List the productCode, productName, and count of orders for the product with the most orders where the productVendor is Welly Diecast Productions. Make sure to title the column heading for the count of orders as "OrderCount". (1)

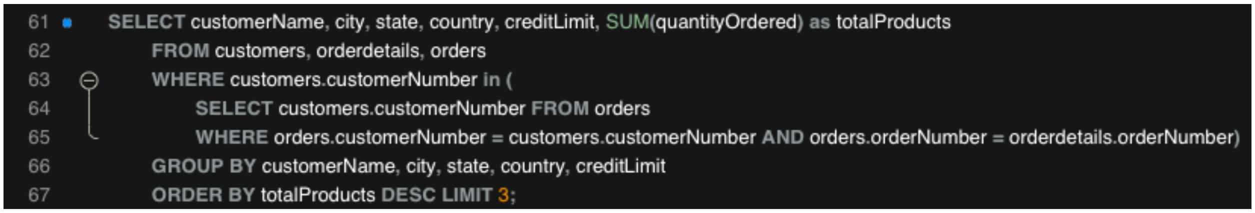


```
SELECT products.productCode, productName, COUNT(quantityOrdered) AS OrderCount
	FROM products JOIN orderdetails
	WHERE  products.productCode = orderdetails.productCode AND productVendor = 'Welly Diecast Productions'
	GROUP BY 1
	ORDER BY OrderCount DESC
	LIMIT 1;
```

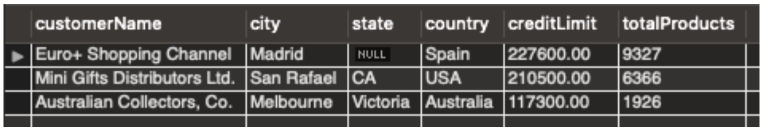


productCo...	productName	OrderCount
S12_1666	1958 Setra Bus	28

8. (use SUBQUERY)
List the name, city, state, country, credit limit, and total products ordered for the top 3 customers who ordered the most products. (3)

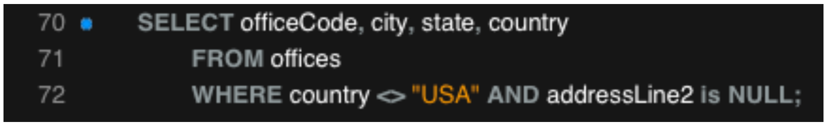


```sql
SELECT customerName, city, state, country, creditLimit, SUM(quantityOrdered) as totalProducts
    FROM customers, orderdetails, orders
    WHERE customers.customerNumber in (
        SELECT customers.customerNumber FROM orders
        WHERE orders.customerNumber = customers.customerNumber AND orders.orderNumber = orderdetails.orderNumber)
    GROUP BY customerName, city, state, country, creditLimit
    ORDER BY totalProducts DESC LIMIT 3;
```

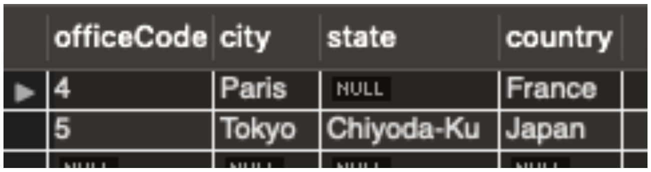


customerName	city	state	country	creditLimit	totalProducts
Euro+ Shopping Channel	Madrid	NULL	Spain	227600.00	9327
Mini Gifts Distributors Ltd.	San Rafael	CA	USA	210500.00	6366
Australian Collectors, Co.	Melbourne	Victoria	Australia	117300.00	1926

9. List the OfficeCode, city, state, country of all the offices that are not in USA and occupying the entire building (the office has no addressLine2 recorded).(2)

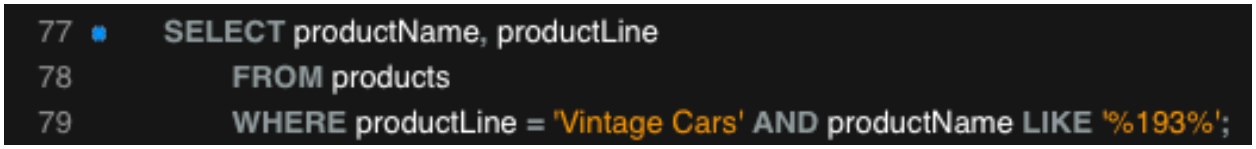


```sql
SELECT officeCode, city, state, country
    FROM offices
    WHERE country <> "USA" AND addressLine2 is NULL;
```

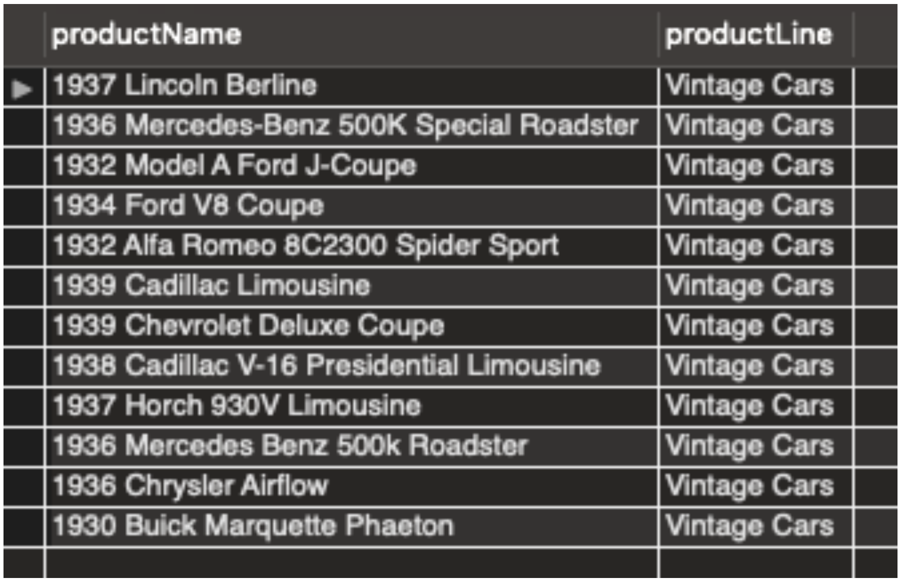


officeCode	city	state	country
4	Paris	NULL	France
5	Tokyo	Chiyoda-Ku	Japan
NULL	NULL	NULL	NULL

10. List the productName and productLine for all Vintage Cars made in the 1930s (productName contains the string “193…”). (12)

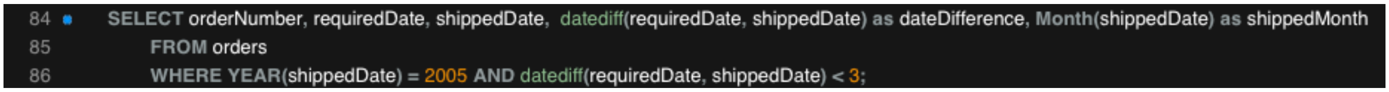


```sql
SELECT productName, productLine
    FROM products
    WHERE productLine = 'Vintage Cars' AND productName LIKE '%193%';
```

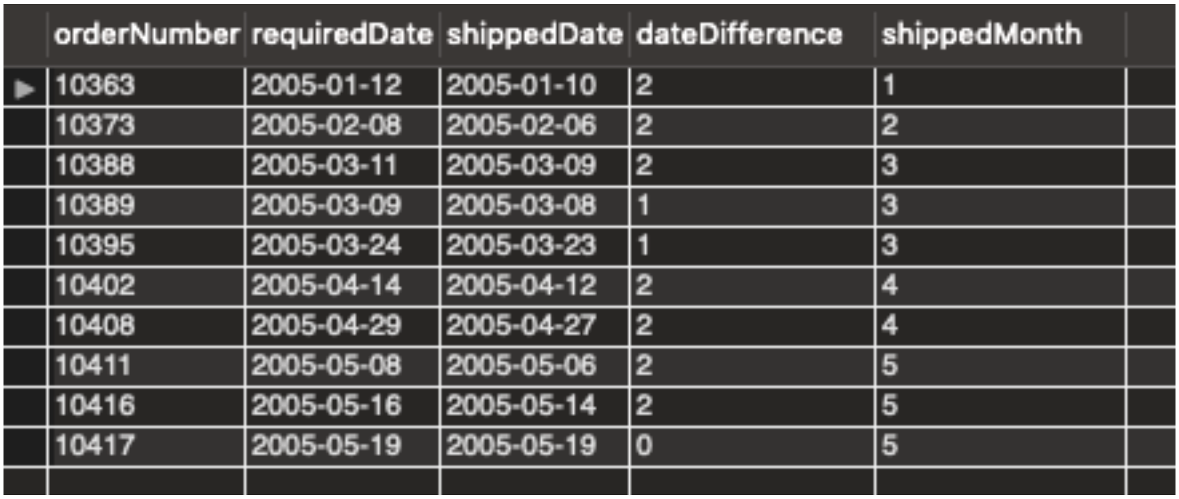


productName	productLine
1937 Lincoln Berline	Vintage Cars
1936 Mercedes-Benz 500K Special Roadster	Vintage Cars
1932 Model A Ford J-Coupe	Vintage Cars
1934 Ford V8 Coupe	Vintage Cars
1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars
1939 Cadillac Limousine	Vintage Cars
1939 Chevrolet Deluxe Coupe	Vintage Cars
1938 Cadillac V-16 Presidential Limousine	Vintage Cars
1937 Horch 930V Limousine	Vintage Cars
1936 Mercedes Benz 500k Roadster	Vintage Cars
1936 Chrysler Airflow	Vintage Cars
1930 Buick Marquette Phaeton	Vintage Cars

11. Select the order number, required date, shipped date, date difference, and shipped month for orders which were shipped less than 3 days before they were due (required date - shipping date < 3) for orders shipped in 2005 (10).

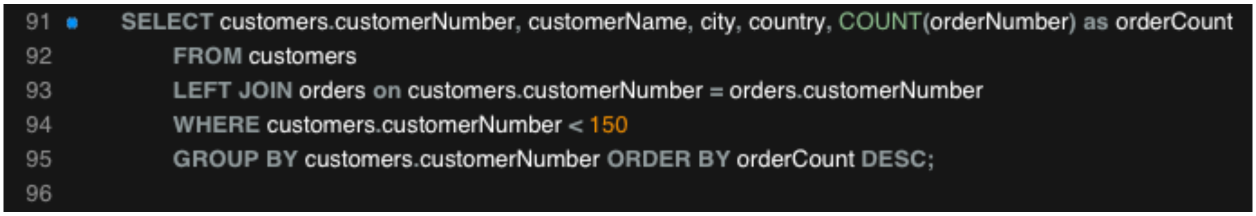


```
SELECT orderNumber, requiredDate, shippedDate,  datediff(requiredDate, shippedDate) as dateDifference, Month(shippedDate) as shippedMonth
        FROM orders
        WHERE YEAR(shippedDate) = 2005 AND datediff(requiredDate, shippedDate) < 3;
```

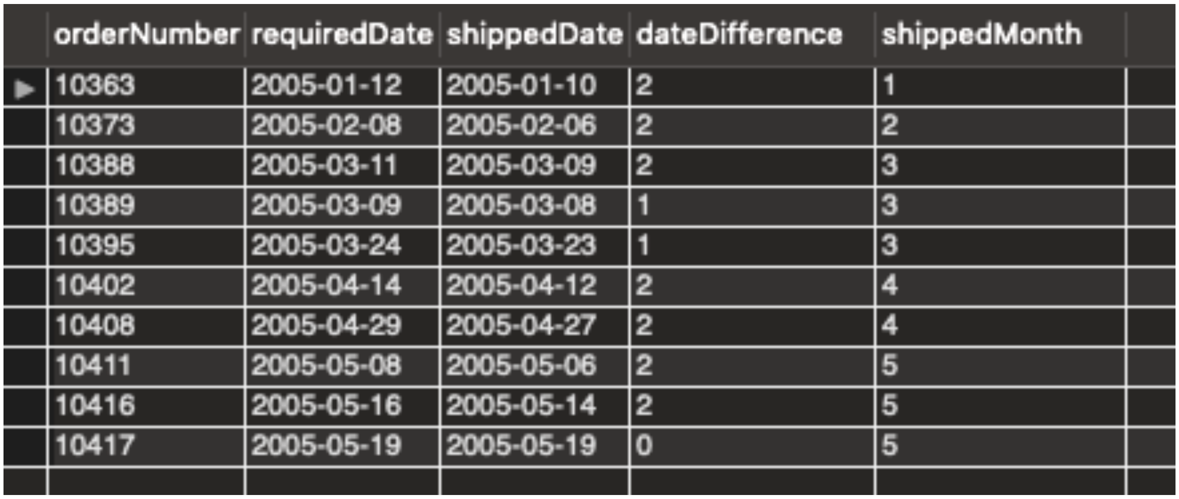


	orderNumber	requiredDate	shippedDate	dateDifference	shippedMonth
▶	10363	2005-01-12	2005-01-10	2	1
	10373	2005-02-08	2005-02-06	2	2
	10388	2005-03-11	2005-03-09	2	3
	10389	2005-03-09	2005-03-08	1	3
	10395	2005-03-24	2005-03-23	1	3
	10402	2005-04-14	2005-04-12	2	4
	10408	2005-04-29	2005-04-27	2	4
	10411	2005-05-08	2005-05-06	2	5
	10416	2005-05-16	2005-05-14	2	5
	10417	2005-05-19	2005-05-19	0	5

12. List the customerNumber, customerName, city, country, and count of their orders for all customers whose customer number is lower than 150. List them in descending order from highest to least number of orders. Title the column heading for the count of orders “Orders”. (15)

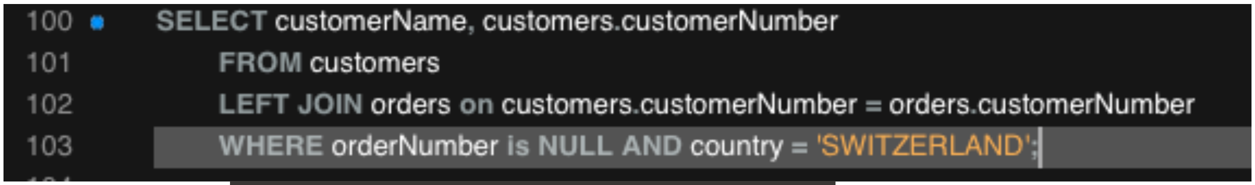


```
SELECT customers.customerNumber, customerName, city, country, COUNT(orderNumber) as orderCount
        FROM customers
        LEFT JOIN orders on customers.customerNumber = orders.customerNumber
        WHERE customers.customerNumber < 150
        GROUP BY customers.customerNumber ORDER BY orderCount DESC;

```

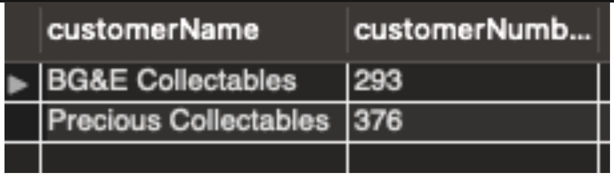


	orderNumber	requiredDate	shippedDate	dateDifference	shippedMonth
▶	10363	2005-01-12	2005-01-10	2	1
	10373	2005-02-08	2005-02-06	2	2
	10388	2005-03-11	2005-03-09	2	3
	10389	2005-03-09	2005-03-08	1	3
	10395	2005-03-24	2005-03-23	1	3
	10402	2005-04-14	2005-04-12	2	4
	10408	2005-04-29	2005-04-27	2	4
	10411	2005-05-08	2005-05-06	2	5
	10416	2005-05-16	2005-05-14	2	5
	10417	2005-05-19	2005-05-19	0	5

13. List the customerName and customerNumber for customers in Switzerland that have no orders. (2)

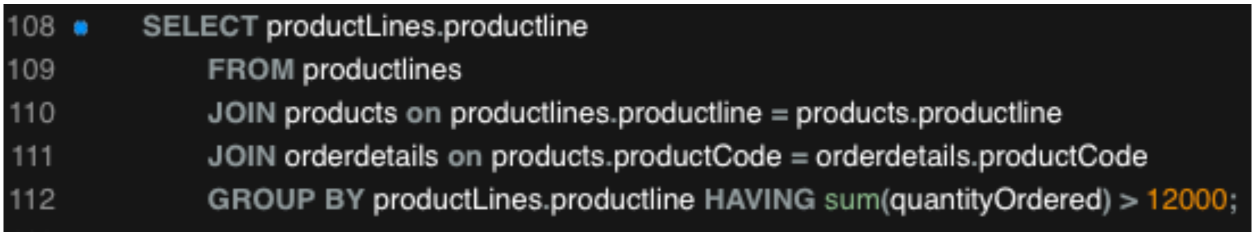


```
SELECT customerName, customers.customerNumber
     FROM customers
     LEFT JOIN orders on customers.customerNumber = orders.customerNumber
     WHERE orderNumber is NULL AND country = 'SWITZERLAND';
```

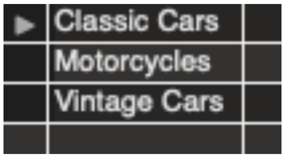


customerName	customerNumb...
BG&E Collectables	293
Precious Collectables	376

14. Select the product lines with over 12,000 orders (3)
In other words, if you tally up all the orders for classic cars, ships, trains, planes, etc., which categories have over 12,000 orders?

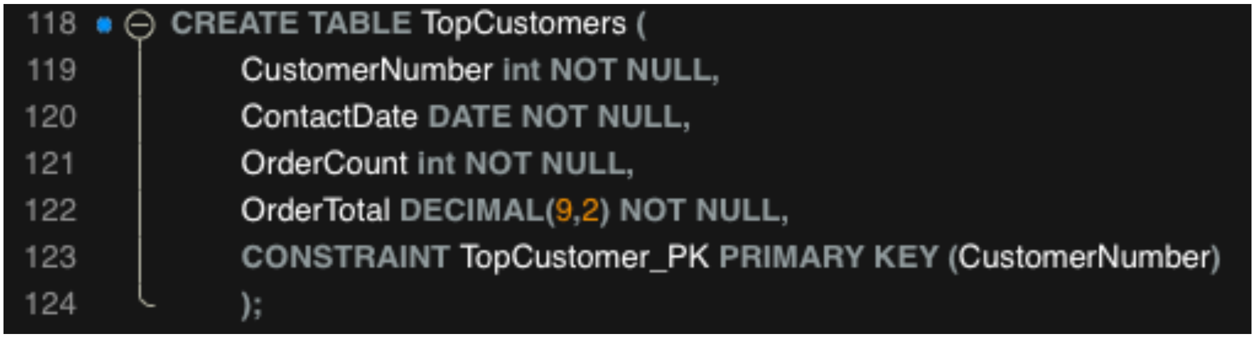


```
SELECT productLines.productline
     FROM productlines
     JOIN products on productlines.productline = products.productline
     JOIN orderdetails on products.productCode = orderdetails.productCode
     GROUP BY productLines.productline HAVING sum(quantityOrdered) > 12000;
```



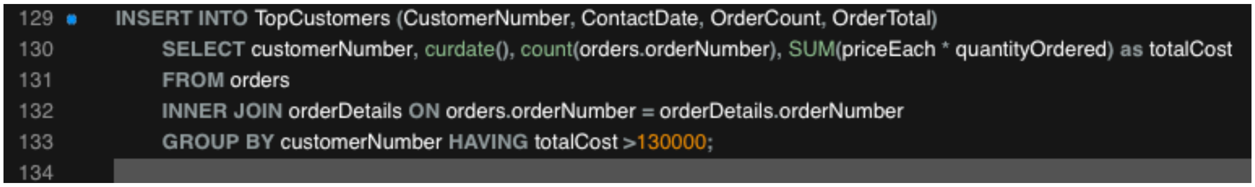
Classic Cars
Motorcycles
Vintage Cars

15. Create a NEW table named "TopCustomers" with four columns: CustomerNumber (integer), ContactDate (DATE), OrderCount(integer), and OrderTotal (a decimal number with 9 digits in total having two decimal places). None of these columns can be NULL. Include a PRIMARY KEY constraint named "TopCustomer_PK" on CustomerNumber. (no answer set)



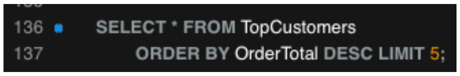
```
CREATE TABLE TopCustomers (
     CustomerNumber int NOT NULL,
     ContactDate DATE NOT NULL,
     OrderCount int NOT NULL,
     OrderTotal DECIMAL(9,2) NOT NULL,
     CONSTRAINT TopCustomer_PK PRIMARY KEY (CustomerNumber)
     );
```

16. Populate the new table “TopCustomers” with the CustomerNumber, today’s date, total number of orders (quantity), and the total value of all their orders (PriceEach * quantityOrdered) for those customers whose order total value is greater than $130,000. (inserted 16 rows, no answer set)

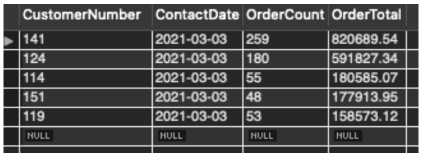


```
INSERT INTO TopCustomers (CustomerNumber, ContactDate, OrderCount, OrderTotal)
    SELECT customerNumber, curdate(), count(orders.orderNumber), SUM(priceEach * quantityOrdered) as totalCost
    FROM orders
    INNER JOIN orderDetails ON orders.orderNumber = orderDetails.orderNumber
    GROUP BY customerNumber HAVING totalCost >130000;
```

17. List the customerNumber, contactDate, orderCount, and orderTotal with the top five highest order totals from “TopCustomers” in descending orderTotal amount. (5)

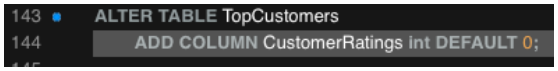


```
SELECT * FROM TopCustomers
    ORDER BY OrderTotal DESC LIMIT 5;
```



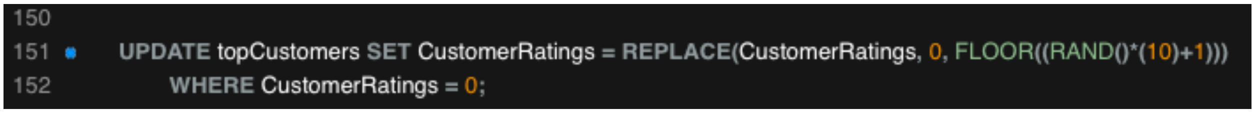
CustomerNumber	ContactDate	OrderCount	OrderTotal
141	2021-03-03	259	820689.54
124	2021-03-03	180	591827.34
114	2021-03-03	55	180585.07
151	2021-03-03	48	177913.95
119	2021-03-03	53	158573.12
NULL	NULL	NULL	NULL

18. Add a new column to the TopCustomers table called CustomerRatings (integer) set to zero by default. (No answer set)



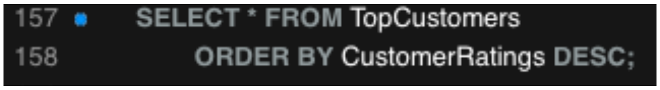
```
ALTER TABLE TopCustomers
    ADD COLUMN CustomerRatings int DEFAULT 0;
```

19. Update the TopCustomers table, setting the CustomerRatings column to a random number from 0 to 10. This will tell us a scale of customer satisfaction: “0” being “terrible”, and “10” being “great service!” HINT: use the RAND() and/or FLOOR() functions as needed. (16 rows affected)

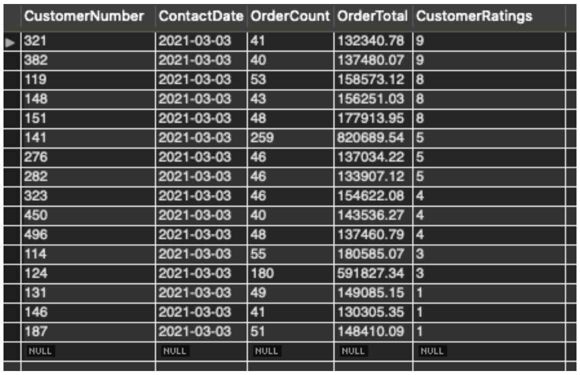


```
UPDATE topCustomers SET CustomerRatings = REPLACE(CustomerRatings, 0, FLOOR((RAND()*(10)+1)))
    WHERE CustomerRatings = 0;
```

20. List the contents of the TopCustomers table in descending CustomerRatings sequence. (16)

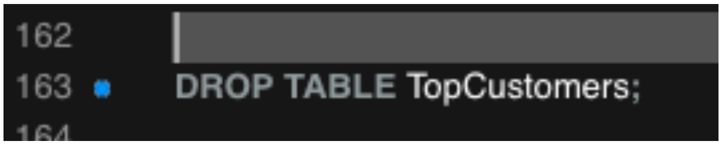


```
SELECT * FROM TopCustomers
    ORDER BY CustomerRatings DESC;
```



CustomerNumber	ContactDate	OrderCount	OrderTotal	CustomerRatings
321	2021-03-03	41	132340.78	9
382	2021-03-03	40	137480.07	9
119	2021-03-03	53	158573.12	8
148	2021-03-03	43	156251.03	8
151	2021-03-03	48	177913.95	8
141	2021-03-03	259	820689.54	5
276	2021-03-03	46	137034.22	5
282	2021-03-03	46	133907.12	5
323	2021-03-03	46	154622.08	4
450	2021-03-03	40	143536.27	4
496	2021-03-03	48	137460.79	4
114	2021-03-03	55	180585.07	3
124	2021-03-03	180	591827.34	3
131	2021-03-03	49	149085.15	1
146	2021-03-03	41	130305.35	1
187	2021-03-03	51	148410.09	1
NULL	NULL	NULL	NULL	NULL

21. Drop the TopCustomers table. (no answer set)



```sql

DROP TABLE TopCustomers;
```