Step-By-Step Instructions

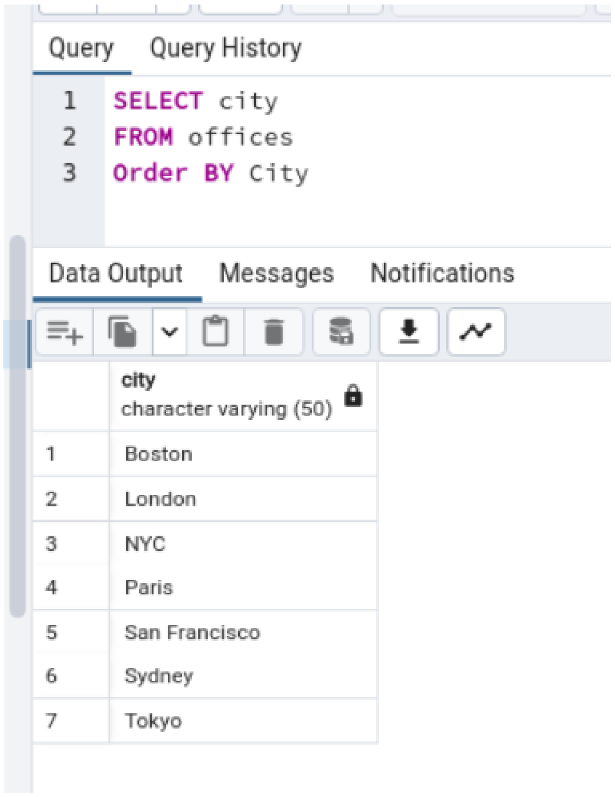
Results Submission

Copy and paste your SQL queries into a document. Number each query to match its problem. Save the document as a PDF and submit

Query SELECT Problems Using the Classic Models database

For this lab you must create and execute queries against the ClassicModels database to fulfill the requirements listed below. As a HINT, the expected number of rows in the answer set is shown in parentheses.

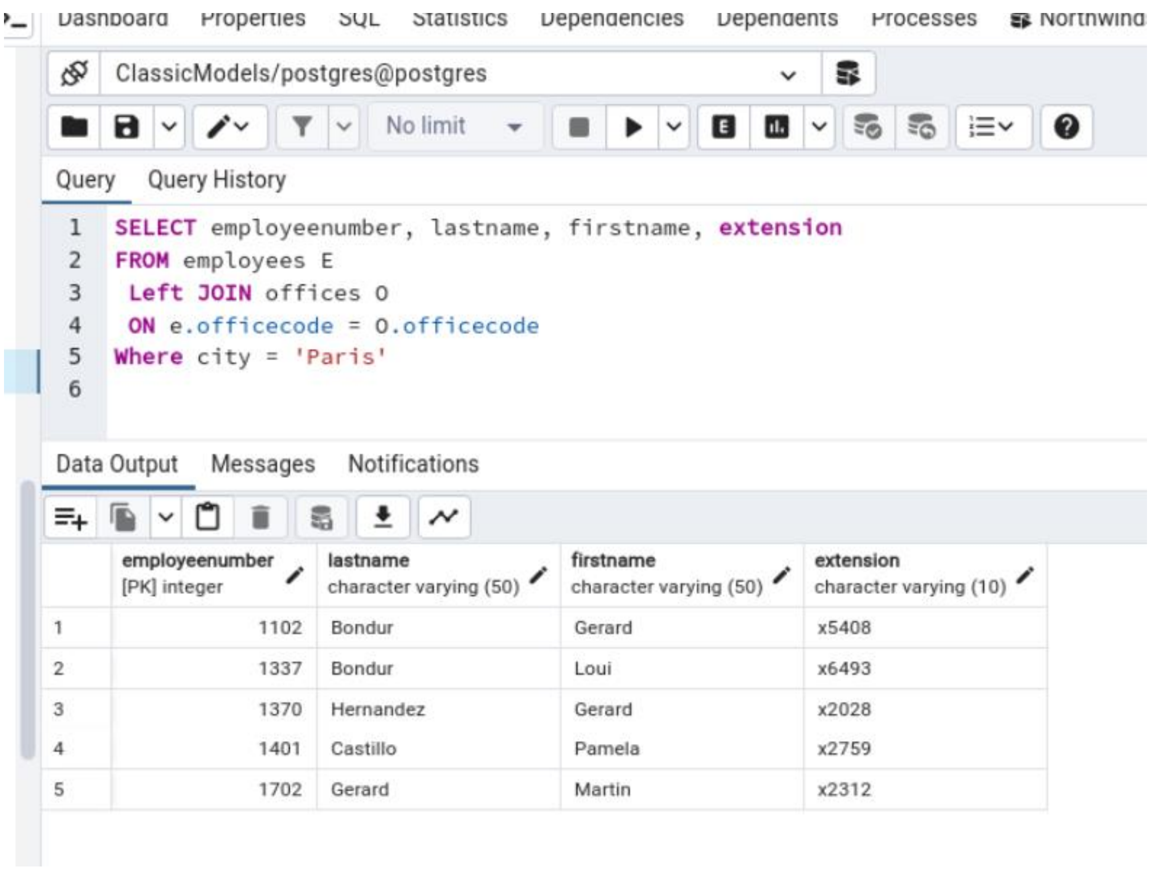
1. List the names of the cities in alphabetical order where Classic Models has offices. (7)



```
SELECT city
FROM offices
Order BY City
```

	city character varying (50)
1	Boston
2	London
3	NYC
4	Paris
5	San Francisco
6	Sydney
7	Tokyo

2. List the EmployeeNumber, LastName, FirstName, Extension for all employees working out of the Paris office. (5)

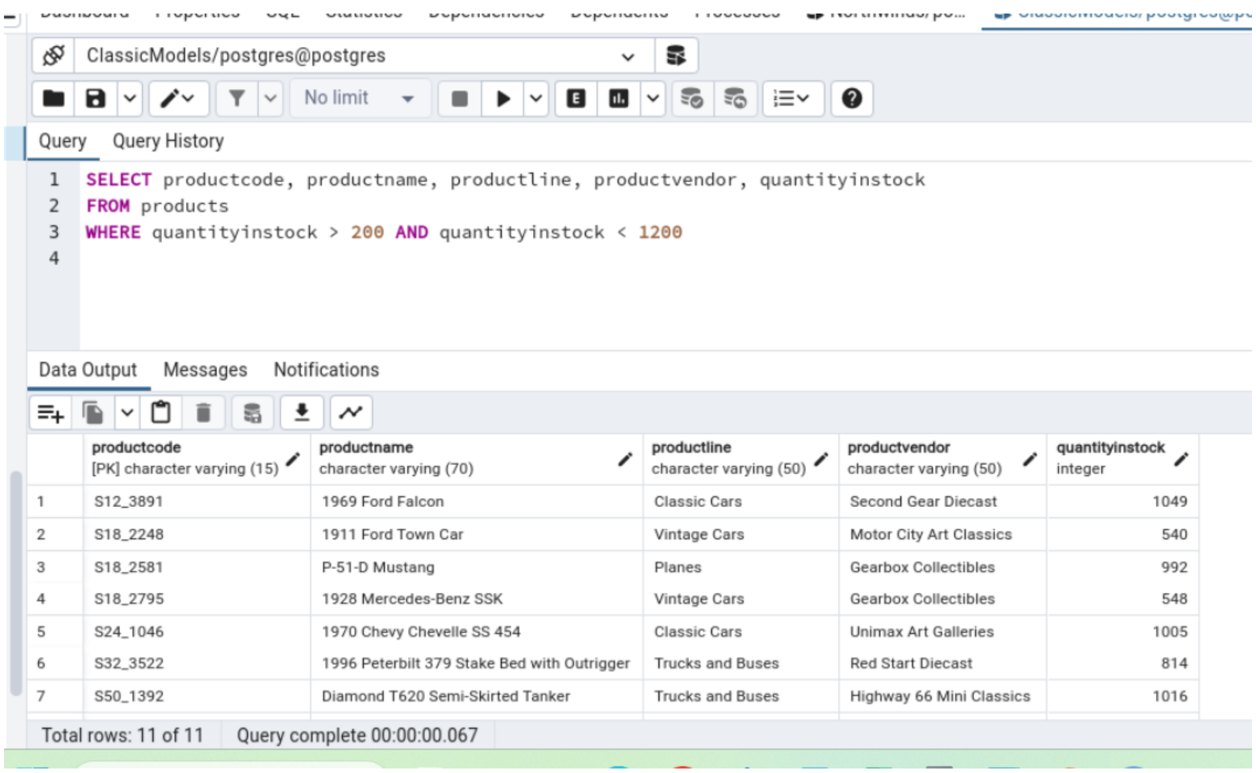


ClassicModels/postgres@postgres

```
SELECT employeenumber, lastname, firstname, extension
FROM employees E
 Left JOIN offices O
 ON e.officecode = O.officecode
Where city = 'Paris'
```

	employeenumber [PK] integer	lastname character varying (50)	firstname character varying (50)	extension character varying (10)
1	1102	Bondur	Gerard	x5408
2	1337	Bondur	Loui	x6493
3	1370	Hernandez	Gerard	x2028
4	1401	Castillo	Pamela	x2759
5	1702	Gerard	Martin	x2312

3. List the ProductCode, ProductName, ProductVendor, QuantityInStock and ProductLine for all products with a QuantityInStock between 200 and 1200. (11)



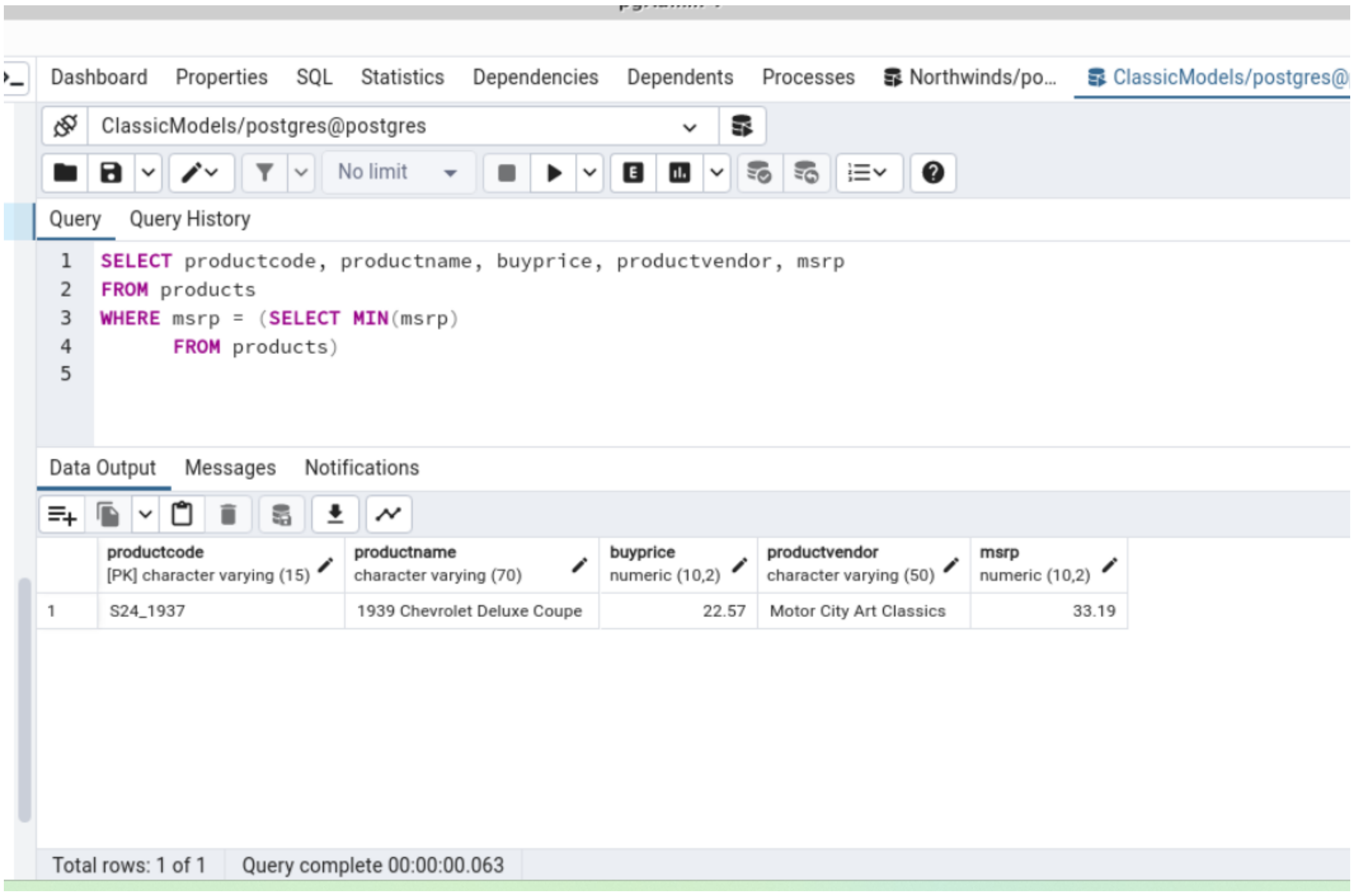
ClassicModels/postgres@postgres

```
SELECT productcode, productname, productline, productvendor, quantityinstock
FROM products
WHERE quantityinstock > 200 AND quantityinstock < 1200
```

	productcode [PK] character varying (15)	productname character varying (70)	productline character varying (50)	productvendor character varying (50)	quantityinstock integer
1	S12_3891	1969 Ford Falcon	Classic Cars	Second Gear Diecast	1049
2	S18_2248	1911 Ford Town Car	Vintage Cars	Motor City Art Classics	540
3	S18_2581	P-51-D Mustang	Planes	Gearbox Collectibles	992
4	S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	Gearbox Collectibles	548
5	S24_1046	1970 Chevy Chevelle SS 454	Classic Cars	Unimax Art Galleries	1005
6	S32_3522	1996 Peterbilt 379 Stake Bed with Outrigger	Trucks and Buses	Red Start Diecast	814
7	S50_1392	Diamond T620 Semi-Skirted Tanker	Trucks and Buses	Highway 66 Mini Classics	1016

Total rows: 11 of 11   Query complete 00:00:00.067

4. (Use a SUBQUERY) List the ProductCode, ProductName, ProductVendor, BuyPrice and MSRP for the least expensive (lowest MSRP) product sold by ClassicModels. (“MSRP” is the Manufacturer’s Suggested Retail Price.) (1)

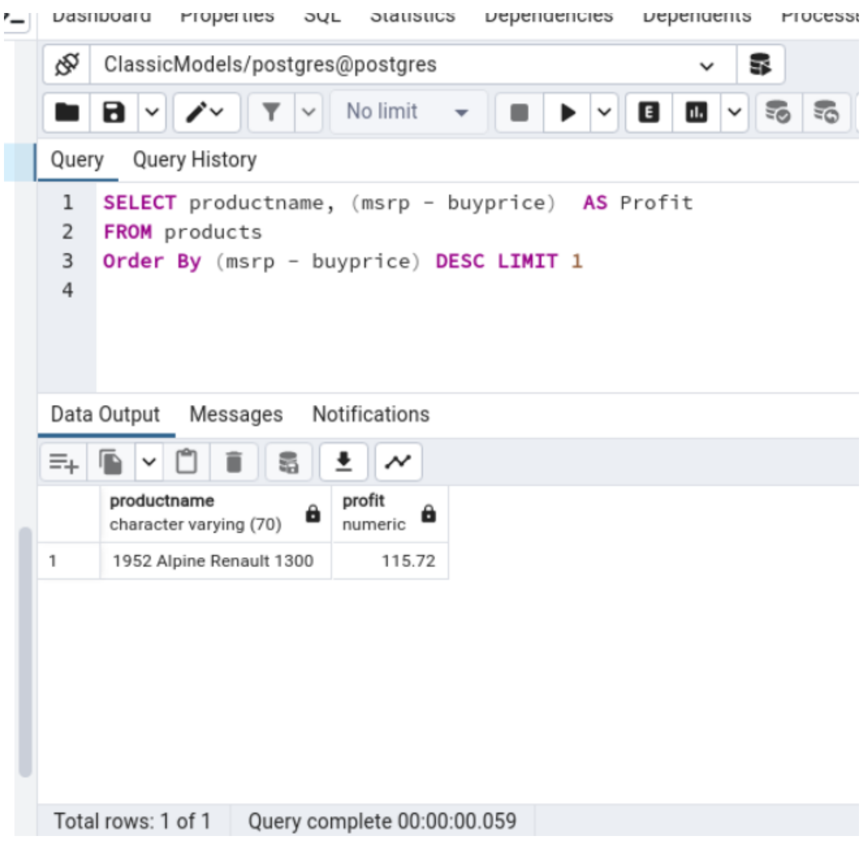


```
SELECT productcode, productname, buyprice, productvendor, msrp
FROM products
WHERE msrp = (SELECT MIN(msrp)
      FROM products)
```

	productcode [PK] character varying (15)	productname character varying (70)	buyprice numeric (10,2)	productvendor character varying (50)	msrp numeric (10,2)
1	S24_1937	1939 Chevrolet Deluxe Coupe	22.57	Motor City Art Classics	33.19

Total rows: 1 of 1 Query complete 00:00:00.063

5. What is the ProductName and Profit of the product that has the highest profit (profit = MSRP minus BuyPrice). (1)

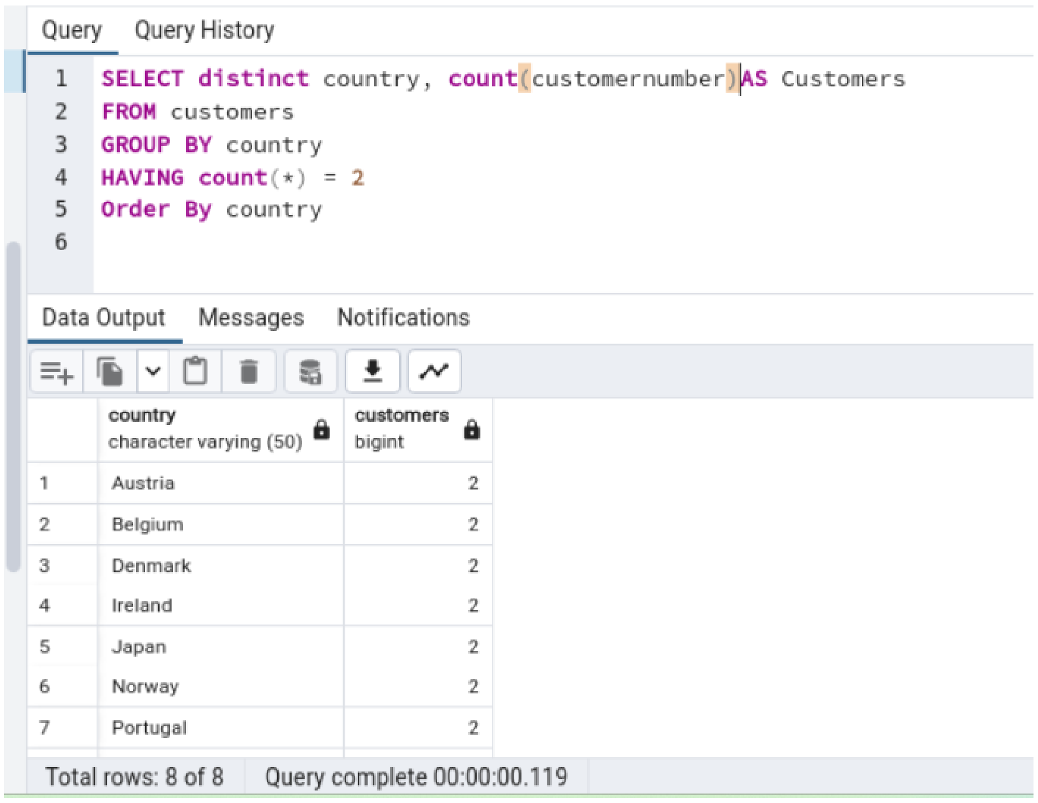


```sql
SELECT productname, (msrp - buyprice)  AS Profit
FROM products
Order By (msrp - buyprice) DESC LIMIT 1
```

```
   productname                    profit
   character varying (70)         numeric
1  1952 Alpine Renault 1300       115.72
```

Total rows: 1 of 1   Query complete 00:00:00.059

6. List the country and the number of customers from that country for all countries having just two customers. List the countries sorted in ascending alphabetical order. Title the column heading for the count of customers as "Customers".(8)

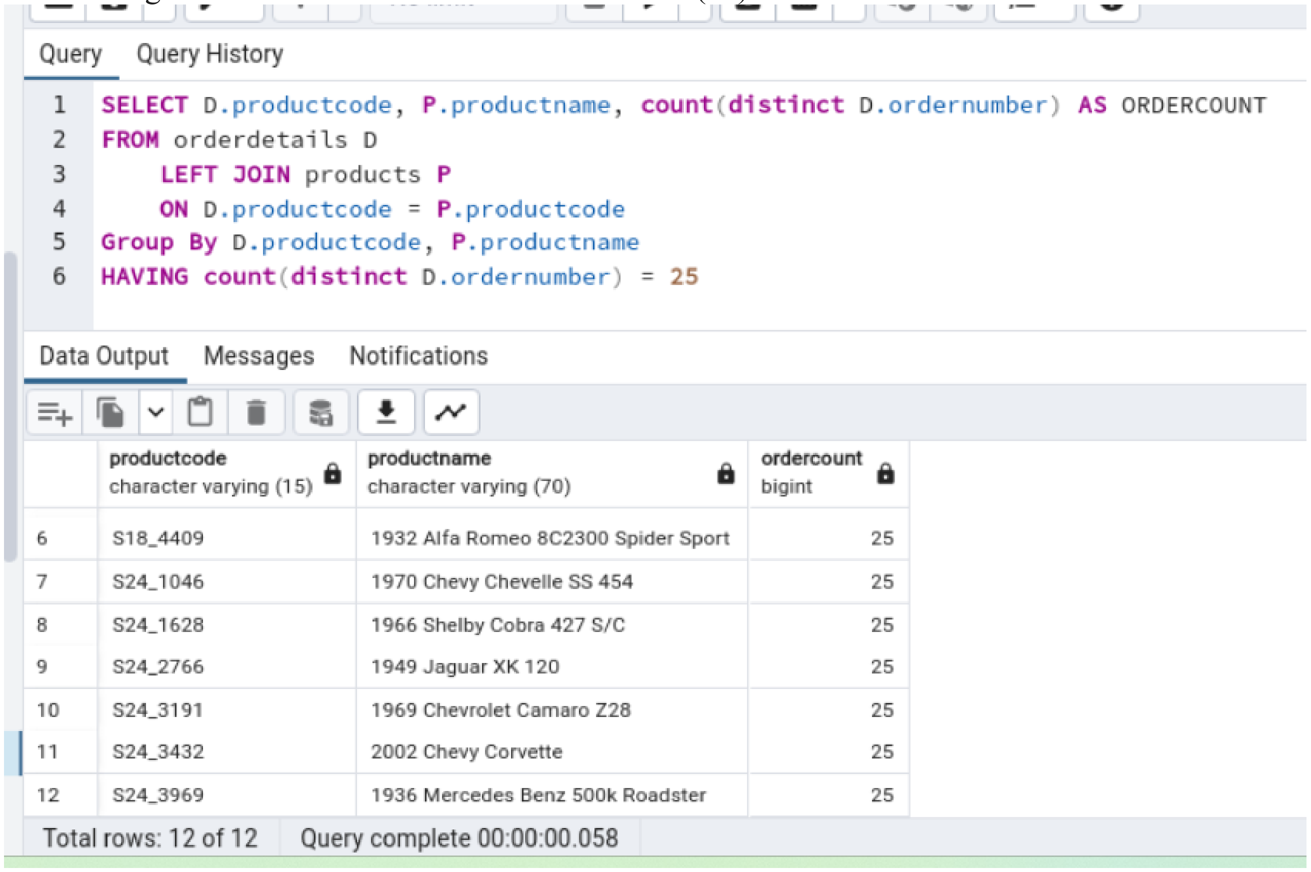


```sql
SELECT distinct country, count(customernumber)AS Customers
FROM customers
GROUP BY country
HAVING count(*) = 2
Order By country
```

```
   country                   customers
   character varying (50)    bigint
1  Austria                   2
2  Belgium                   2
3  Denmark                   2
4  Ireland                   2
5  Japan                     2
6  Norway                    2
7  Portugal                  2
```

Total rows: 8 of 8   Query complete 00:00:00.119

7. List the ProductCode, ProductName, and number of orders for the products with exactly 25 orders. Title the column heading for the count of orders as “OrderCount”. (12)



Query | Query History

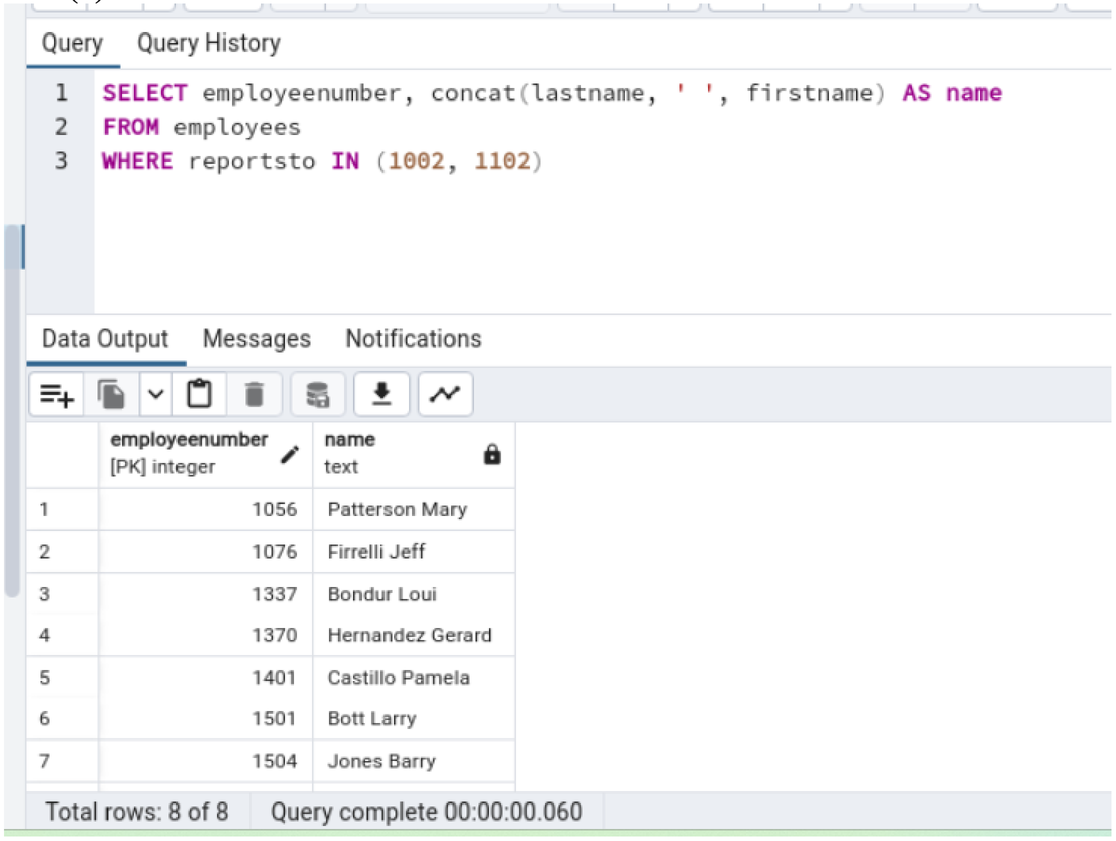
```
SELECT D.productcode, P.productname, count(distinct D.ordernumber) AS ORDERCOUNT
FROM orderdetails D
    LEFT JOIN products P
    ON D.productcode = P.productcode
Group By D.productcode, P.productname
HAVING count(distinct D.ordernumber) = 25
```

Data Output | Messages | Notifications

	productcode character varying (15)	productname character varying (70)	ordercount bigint
6	S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	25
7	S24_1046	1970 Chevy Chevelle SS 454	25
8	S24_1628	1966 Shelby Cobra 427 S/C	25
9	S24_2766	1949 Jaguar XK 120	25
10	S24_3191	1969 Chevrolet Camaro Z28	25
11	S24_3432	2002 Chevy Corvette	25
12	S24_3969	1936 Mercedes Benz 500k Roadster	25

Total rows: 12 of 12 Query complete 00:00:00.058

8. List the EmployeeNumber, Firstname + Lastname (concatenated into one column in the answer set, separated by a blank and referred to as ‘name’) for all the employees reporting to Diane Murphy or Gerard Bondur. (8)



Query | Query History

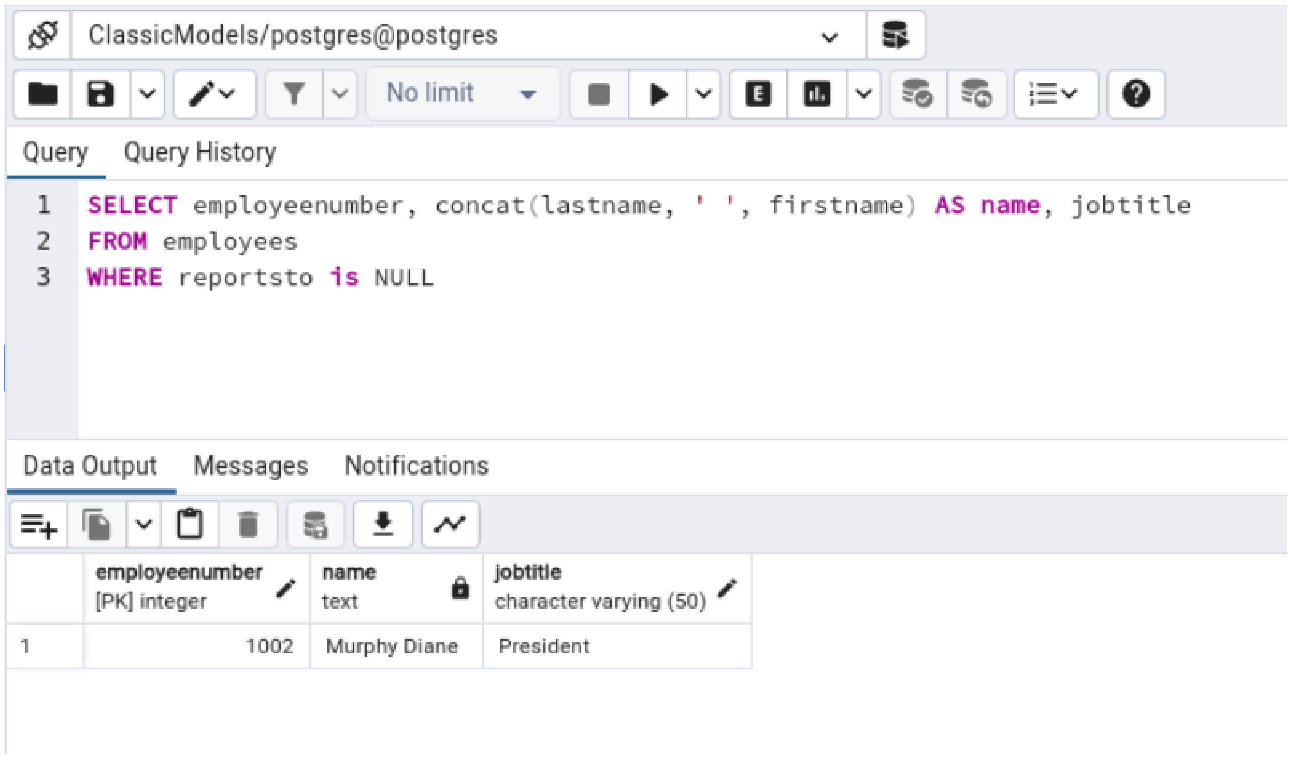
```
SELECT employeenumber, concat(lastname, ' ', firstname) AS name
FROM employees
WHERE reportsto IN (1002, 1102)
```

Data Output | Messages | Notifications

	employeenumber [PK] integer	name text
1	1056	Patterson Mary
2	1076	Firrelli Jeff
3	1337	Bondur Loui
4	1370	Hernandez Gerard
5	1401	Castillo Pamela
6	1501	Bott Larry
7	1504	Jones Barry

Total rows: 8 of 8 Query complete 00:00:00.060

9. List the EmployeeNumber, LastName, FirstName of the president of the company (the one employee with no boss.) (1)

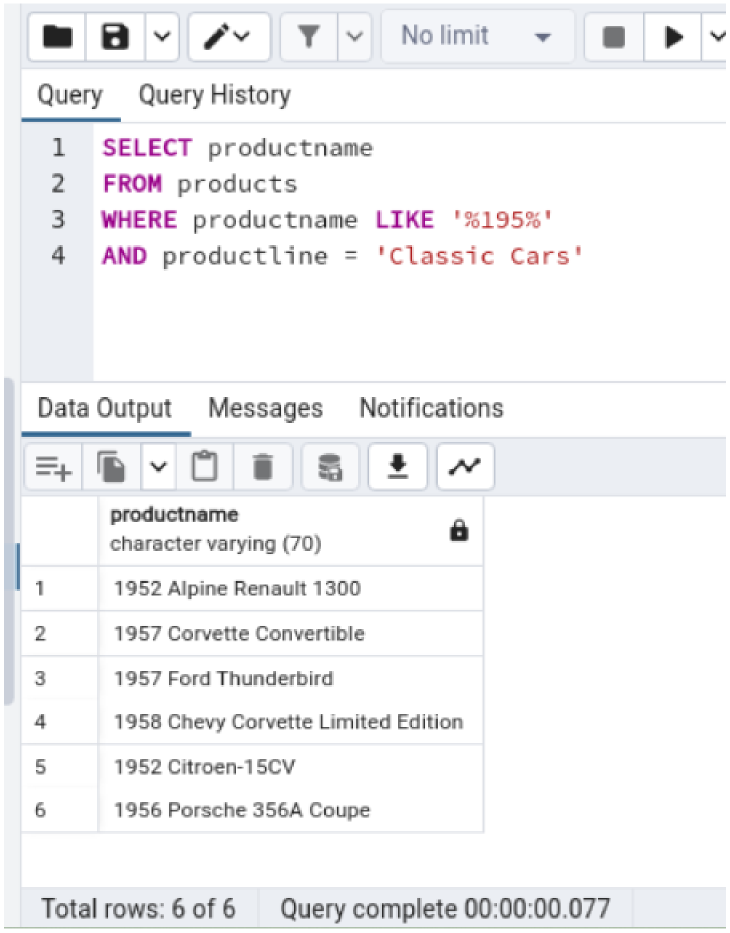


ClassicModels/postgres@postgres

```
SELECT employeenumber, concat(lastname, ' ', firstname) AS name, jobtitle
FROM employees
WHERE reportsto is NULL
```

	employeenumber [PK] integer	name text	jobtitle character varying (50)
1	1002	Murphy Diane	President

10. List the ProductName for all products in the “Classic Cars” product line from the 1950’s. (6)

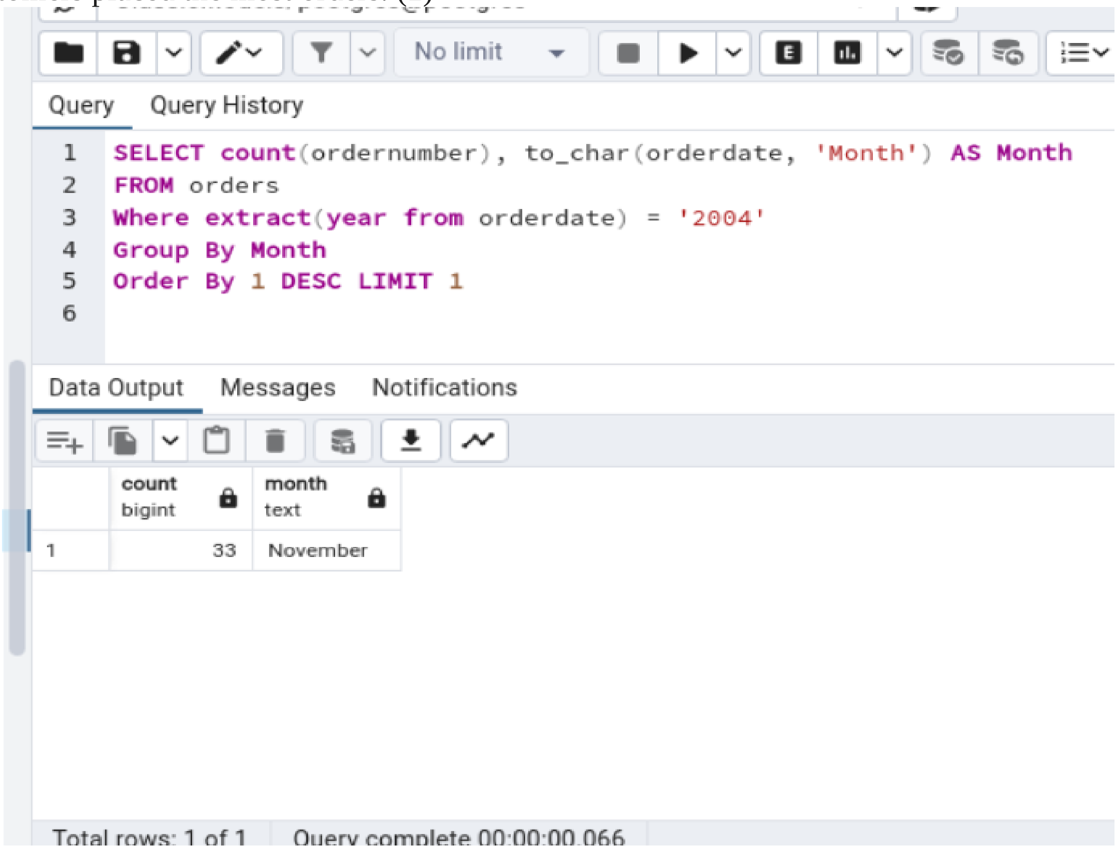


```
SELECT productname
FROM products
WHERE productname LIKE '%195%'
AND productline = 'Classic Cars'
```

	productname character varying (70)
1	1952 Alpine Renault 1300
2	1957 Corvette Convertible
3	1957 Ford Thunderbird
4	1958 Chevy Corvette Limited Edition
5	1952 Citroen-15CV
6	1956 Porsche 356A Coupe

Total rows: 6 of 6 Query complete 00:00:00.077

11. List the month name and the total number of orders for the month in 2004 in which ClassicModels customers placed the most orders. (1)

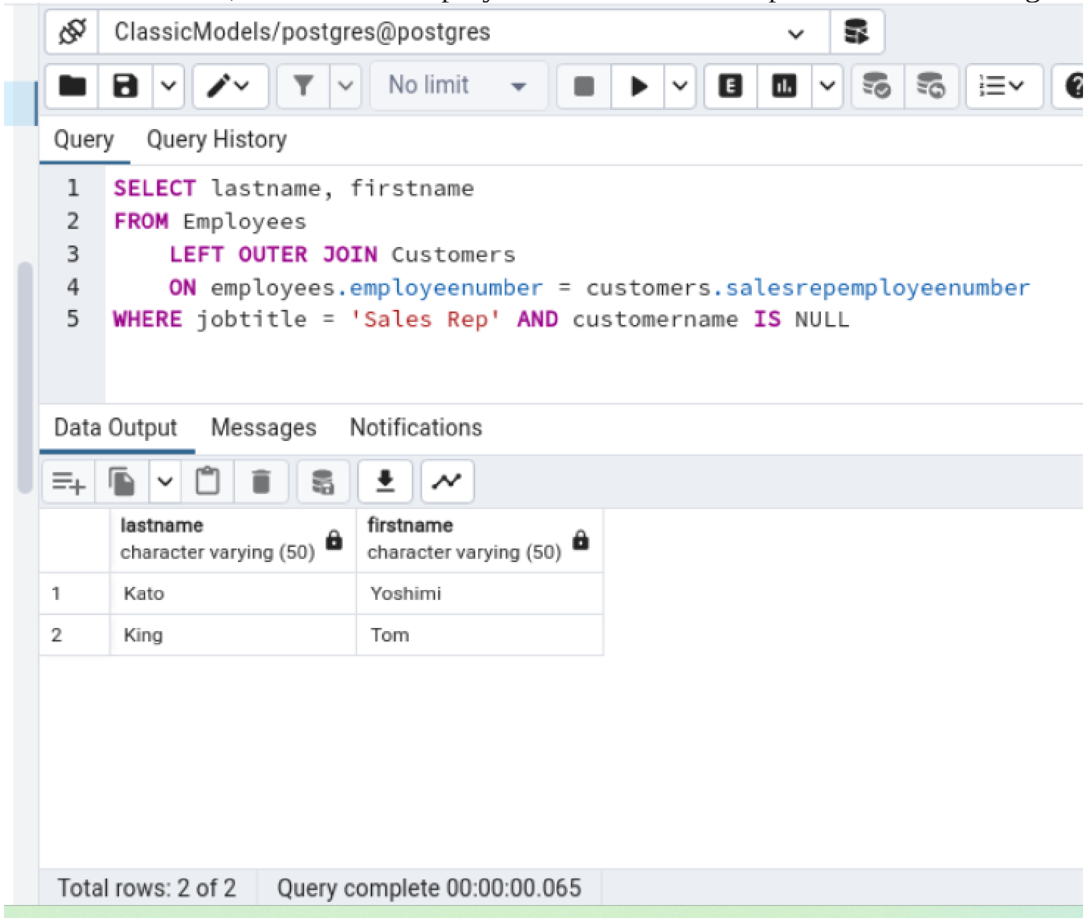


```sql
SELECT count(ordernumber), to_char(orderdate, 'Month') AS Month
FROM orders
Where extract(year from orderdate) = '2004'
Group By Month
Order By 1 DESC LIMIT 1
```

```
   count    month
   bigint   text
1      33   November

Total rows: 1 of 1    Query complete 00:00:00.066
```

12. List the firstname, lastname of employees who are Sales Reps who have no assigned customers. (2)

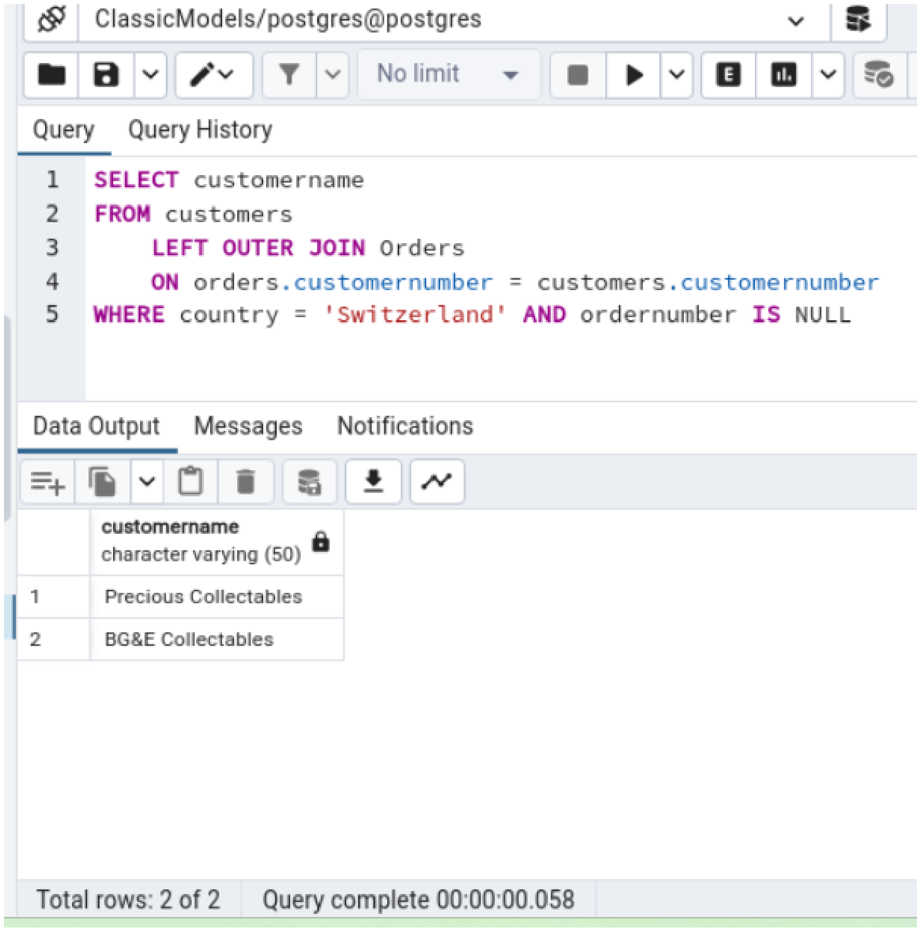


```sql
SELECT lastname, firstname
FROM Employees
    LEFT OUTER JOIN Customers
    ON employees.employeenumber = customers.salesrepemployeenumber
WHERE jobtitle = 'Sales Rep' AND customername IS NULL
```

```
   lastname                  firstname
   character varying (50)    character varying (50)
1  Kato                      Yoshimi
2  King                      Tom

Total rows: 2 of 2    Query complete 00:00:00.065
```

13. List the customername of customers from Switzerland with no orders. (2)

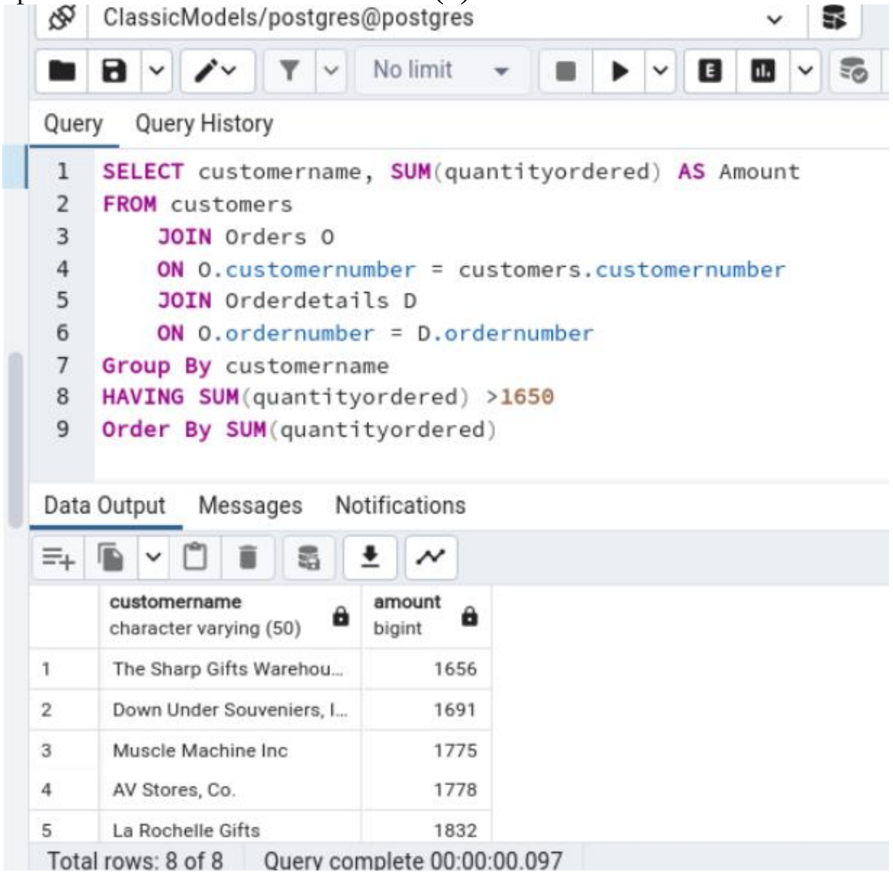


```
SELECT customername
FROM customers
    LEFT OUTER JOIN Orders
    ON orders.customernumber = customers.customernumber
WHERE country = 'Switzerland' AND ordernumber IS NULL
```

```
   customername
   character varying (50)
1  Precious Collectables
2  BG&E Collectables
```

Total rows: 2 of 2   Query complete 00:00:00.058

14. List the customername and total quantity of products ordered for customers who have ordered more than 1650 products across all their orders. (8)



```
SELECT customername, SUM(quantityordered) AS Amount
FROM customers
    JOIN Orders O
    ON O.customernumber = customers.customernumber
    JOIN Orderdetails D
    ON O.ordernumber = D.ordernumber
Group By customername
HAVING SUM(quantityordered) >1650
Order By SUM(quantityordered)
```

```
   customername              amount
   character varying (50)    bigint
1  The Sharp Gifts Warehou...  1656
2  Down Under Souveniers, I... 1691
3  Muscle Machine Inc          1775
4  AV Stores, Co.              1778
5  La Rochelle Gifts           1832
```

Total rows: 8 of 8   Query complete 00:00:00.097

Query DML/DDL Problems Using the Classic Models database

1. Create a NEW table named "TopCustomers" with three columns: CustomerNumber (integer), ContactDate (DATE) and OrderTotal (a real number.) None of these columns can be NULL.

```
create table if not exists TopCustomers (
    Customernumber int not null,
    ContactDate    DATE not null,
     OrderTotal     decimal(9,2) not null default 0,        constraint     PKTopCustome
rs primary key (CustomerNumber)
 );
```

2. Populate the new table "TopCustomers" with the CustomerNumber, today's date, and the total value of all their orders (PriceEach * quantityOrdered) for those customers whose order total value is greater than $140,000. (should insert 10 rows)

```
insert into TopCustomers
    select c.customernumber, CURRENT_date,
              SUM(priceEach * Quantityordered)
        from Customers c, Orders o,OrderDetails d
            where c.Customernumber = o.Customernumber
          and o.Ordernumber = d.Ordernumber
    group by c.Customernumber
    having SUM(priceEach * Quantityordered) > 140000;
```

3. List the contents of the TopCustomers table in descending OrderTotal sequence. (10)

```
select * from topcustomers order by 3 desc;
```

4. Add a new column to the TopCustomers table called OrderCount (integer).

```
alter table topcustomers
        add column OrderCount integer ;
```

5. Update the Top Customers table, setting the OrderCount to a random number between 1 and 10. Hint: use (RANDOM() *10)

```
update topcustomers
        set ordercount = floor((rand()*18));
```

6. List the contents of the TopCustomers table in descending OrderCount sequence. (10 rows)

```
select *
        from topcustomers
        order by 4 desc;
```

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

```
Drop table topcustomers;
```

Appendix “A” – Overview of the Classic Models database

Classic Models Company

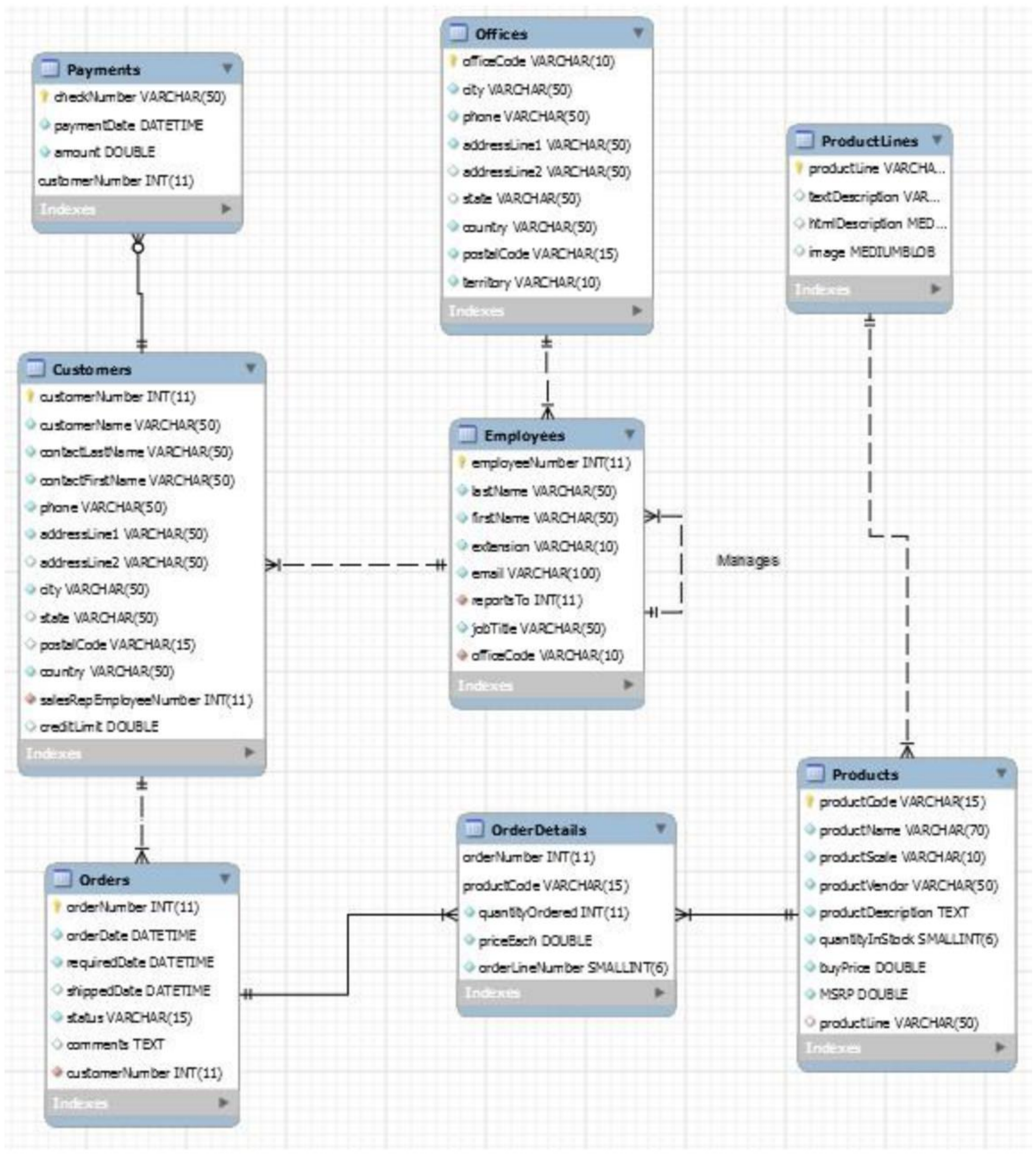
Classic Models is a retailer of diecast miniature collectible model cars, motorcycles, airplanes, trains and ships.

Classic Models Database

The sample “Classic Models” database consists of the following eight tables:

Customers	**stores customer data.**
Products	stores information about the scale models.
ProductLines	stores a list of Classic Models’ various product lines.
Orders	stores sales orders placed by customers.
OrderDetails	stores sales order line items for each sales order.
Payments	stores payments made by customers for purchases
Employees	stores all Classic Models employee information
Offices	stores sales office data.

Classic Models Data Model


Payments
checkNumber VARCHAR(50)
paymentDate DATETIME
amount DOUBLE
customerNumber INT(11)
Indexes
Offices
officeCode VARCHAR(10)
city VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
state VARCHAR(50)
country VARCHAR(50)
postalCode VARCHAR(15)
territory VARCHAR(10)
Indexes
ProductLines
productLine VARCHA...
textDescription VAR...
htmlDescription MED...
image MEDIUMBLOB
Indexes
Customers
customerNumber INT(11)
customerName VARCHAR(50)
contactLastName VARCHAR(50)
contactFirstName VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(50)
state VARCHAR(50)
postalCode VARCHAR(15)
country VARCHAR(50)
salesRepEmployeeNumber INT(11)
creditLimit DOUBLE
Indexes
Employees
employeeNumber INT(11)
lastName VARCHAR(50)
firstName VARCHAR(50)
extension VARCHAR(10)
email VARCHAR(100)
reportsTo INT(11)
jobTitle VARCHAR(50)
officeCode VARCHAR(10)
Indexes
Manages
Products
productCode VARCHAR(15)
productName VARCHAR(70)
productScale VARCHAR(10)
productVendor VARCHAR(50)
productDescription TEXT
quantityInStock SMALLINT(6)
buyPrice DOUBLE
MSRP DOUBLE
productLine VARCHAR(50)
Indexes
OrderDetails
orderNumber INT(11)
productCode VARCHAR(15)
quantityOrdered INT(11)
priceEach DOUBLE
orderLineNumber SMALLINT(6)
Indexes
Orders
orderNumber INT(11)
orderDate DATETIME
requiredDate DATETIME
shippedDate DATETIME
status VARCHAR(15)
comments TEXT
customerNumber INT(11)
Indexes