SQL Practice 44-53

44. What product that makes us the most money (qty\*price) across all orders for that product?

Returns 1.

SELECT p.PRODUCTCODE, p.PRODUCTNAME, p.PRODUCTLINE,

(o.QUANTITYORDERED \* o.PRICEEACH) AS MostMoney

FROM ORDERDETAILS o

INNER JOIN

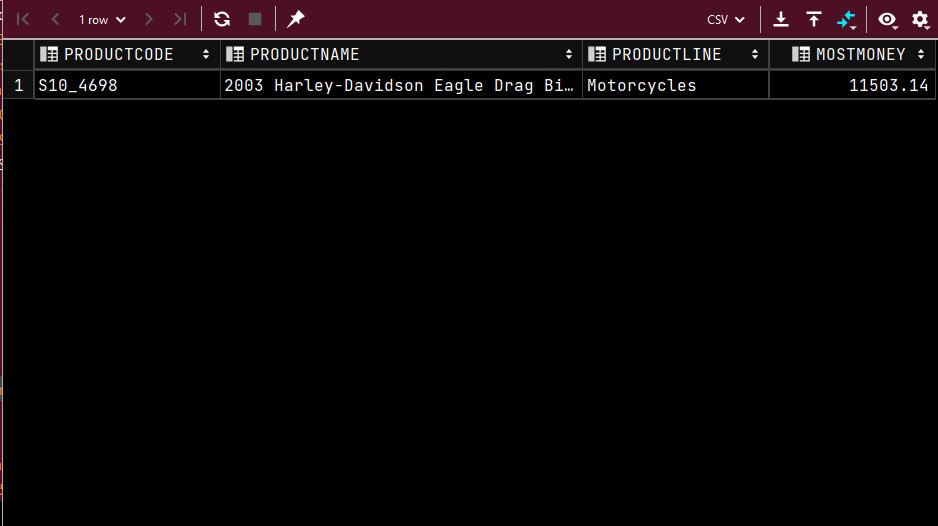
PRODUCTS p

ON o.PRODUCTCODE = p.PRODUCTCODE

WHERE (o.QUANTITYORDERED \* o.PRICEEACH) =

(SELECT MAX(QUANTITYORDERED \* PRICEEACH)

|  |  |
| --- | --- |
| FROM | ORDERDETAILS o); |



45.List the product lines and vendors for product lines which are supported by < 5 vendors. That is, there are < 5 vendors making products within that product line. Returns 3.

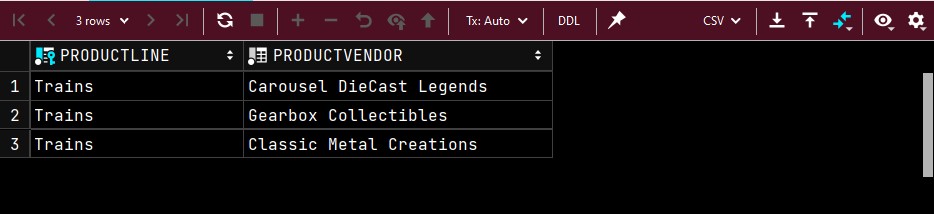
SELECT PRODUCTLINE, PRODUCTVENDOR FROM PRODUCTS

WHERE PRODUCTLINE =

(SELECT PRODUCTLINE FROM PRODUCTS

GROUP BY PRODUCTLINE

HAVING COUNT(\*) < 5);



46.List the products in the product line with the most number of products. Returns 38.

SELECT PRODUCTNAME, PRODUCTLINE

FROM PRODUCTS

WHERE PRODUCTLINE =

(

SELECT PRODUCTLINE AS PL

FROM PRODUCTS

GROUP BY PRODUCTLINE

HAVING COUNT(PRODUCTNAME) =

(

SELECT MAX(PL) AS MP

FROM

(

SELECT PRODUCTLINE, COUNT(PRODUCTNAME) AS NOP

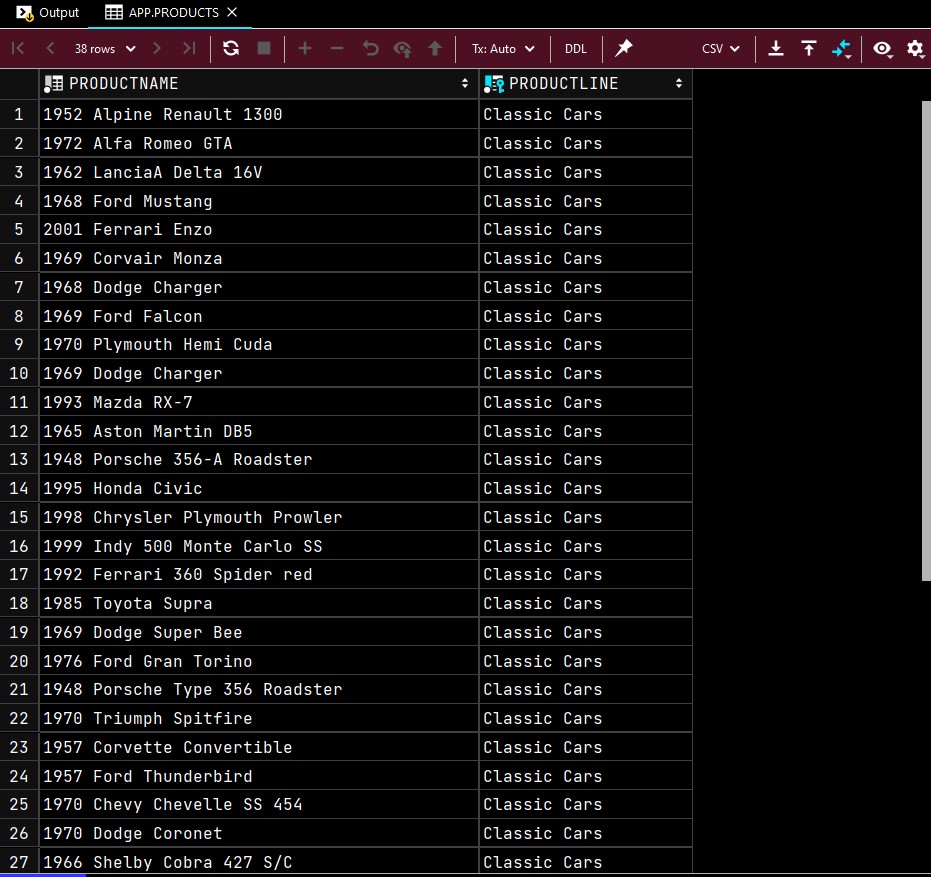
FROM PRODUCTS

GROUP BY PRODUCTLINE

) AS MNOP

)

);



47. Find the first name and last name of all customer contacts whose customer is located in the same state as the San Francisco office. Returns 11.

SELECT CONTACTFIRSTNAME, CONTACTLASTNAME

FROM CUSTOMERS

WHERE STATE = (SELECT STATE

FROM OFFICES

WHERE CITY = 'San Francisco');



48. What is the customer and salesperson of the highest priced order? The price of the order is the sum of the quantity ordered \* the price each for all the items within that order. Returns 1.

SELECT c.CUSTOMERNAME, e.FIRSTNAME AS "EmployeeFirstName", e.LASTNAME AS

"EmployeeLastName"

FROM CUSTOMERS c INNER JOIN EMPLOYEES e ON c.SALESREPEMPLOYEENUMBER = e.EMPLOYEENUMBER

INNER JOIN ORDERS o ON c.CUSTOMERNUMBER = o.CUSTOMERNUMBER

INNER JOIN ORDERDETAILS od ON o.ORDERNUMBER = od.ORDERNUMBER

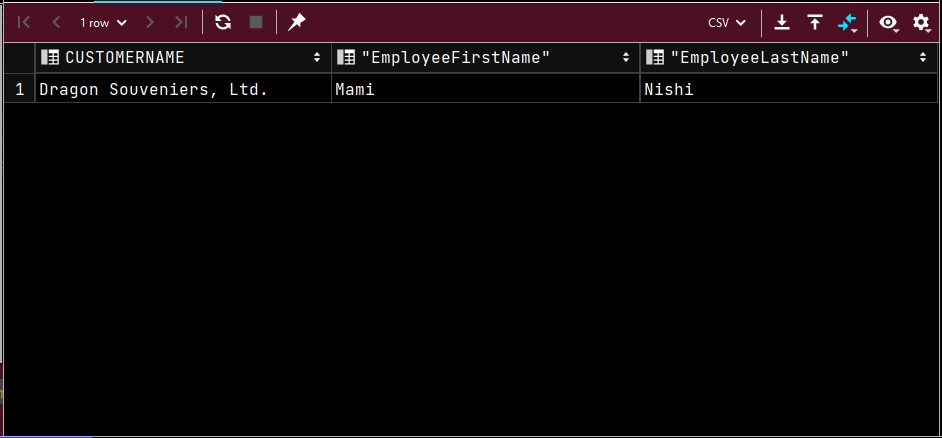
GROUP BY od.ORDERNUMBER, c.CUSTOMERNAME, e.FIRSTNAME, e.LASTNAME

HAVING SUM(QUANTITYORDERED \* PRICEEACH) =

(SELECT MAX(TOTALPRICE) FROM

(SELECT SUM(QUANTITYORDERED \* PRICEEACH) AS TOTALPRICE FROM ORDERDETAILS

GROUP BY ORDERNUMBER) piceTable);



49. What is the order number and the cost of the order for the most expensive orders? Note that there could be more than one order which all happen to add up to the same cost, and that same cost could be the highest cost among all orders. The cost of an order is the sum of the quantity ordered \* the price each for all the items within that order. Returns 1.

SELECT ORDERNUMBER, SUM(priceEach\*quantityOrdered) AS TOTAL

FROM ORDERDETAILS

GROUP BY ORDERNUMBER

HAVING SUM(priceEach\*quantityOrdered) = (

SELECT MAX(ORDERTOTALS.orderTotal)

FROM

(

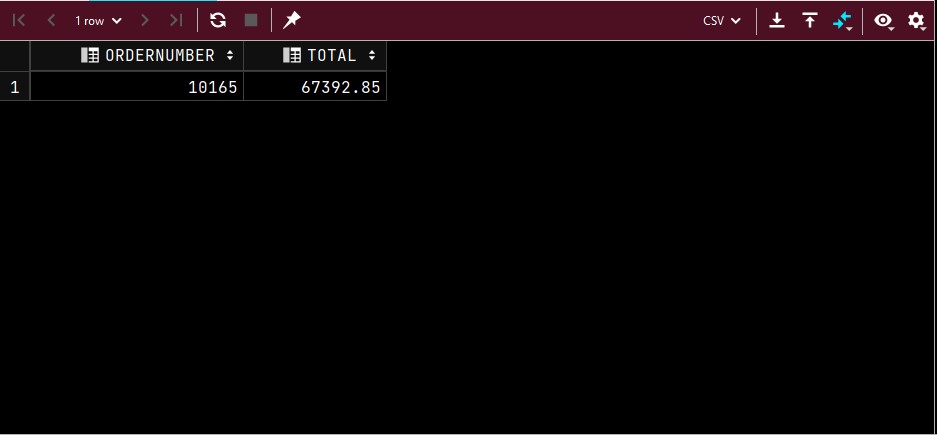
SELECT SUM(priceEach\*QUANTITYORDERED) AS ORDERTOTAL

FROM ORDERDETAILS

GROUP BY ORDERNUMBER

) AS ORDERTOTALS

);



50.What is the name of the customer, the order number, and the total cost of the most expensive orders? Returns 1.

SELECT c.CUSTOMERNAME, od.ORDERNUMBER, SUM(QUANTITYORDERED \* PRICEEACH) AS

TotalCost

FROM CUSTOMERS c INNER JOIN ORDERS o ON c.CUSTOMERNUMBER = o.CUSTOMERNUMBER

INNER JOIN ORDERDETAILS od ON o.ORDERNUMBER = od.ORDERNUMBER

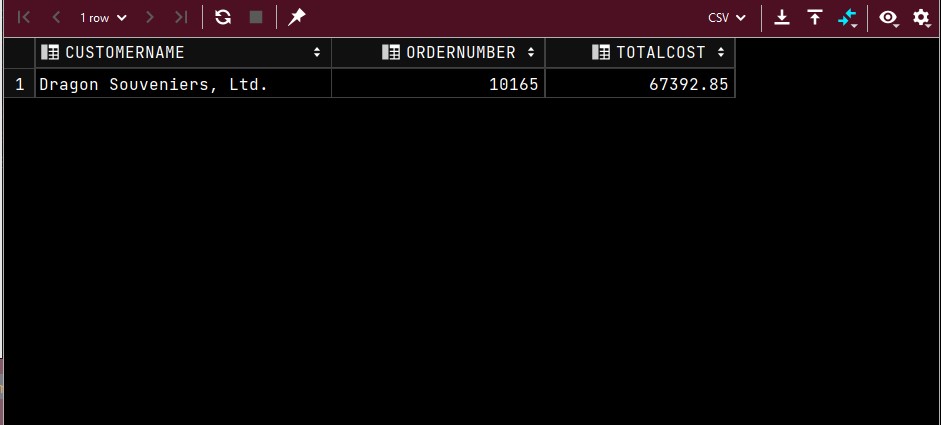
GROUP BY od.ORDERNUMBER, c.CUSTOMERNAME

HAVING SUM(QUANTITYORDERED \* PRICEEACH) =

(SELECT MAX(TotalPrice) FROM

(SELECT SUM(QUANTITYORDERED \* PRICEEACH) AS TotalPrice FROM ORDERDETAILS

GROUP BY orderNumber) piceTable);



51.Take some portion of the above query and put that into a view. Then rewrite the above query to use the view that you just created and consider how incorporating the view made the query easier to understand. If you do not know how many rows this returns, please come see me immediately.

CREATE VIEW TotalCosts AS

SELECT SUM(QUANTITYORDERED \* PRICEEACH) AS TotalPrice, orderNumber FROM

ORDERDETAILS

GROUP BY orderNumber;

SELECT c.CUSTOMERNAME, o.ORDERNUMBER, tc.TOTALPRICE

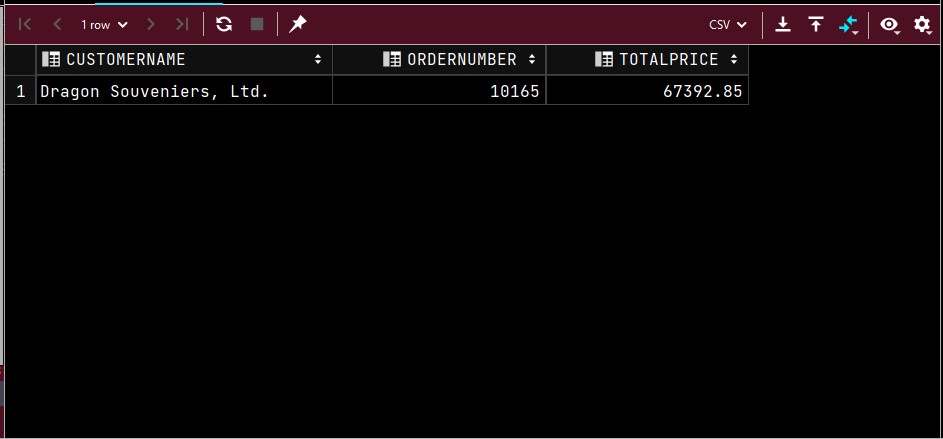
FROM CUSTOMERS c INNER JOIN Orders o ON c.CUSTOMERNUMBER = o.CUSTOMERNUMBER

INNER JOIN TotalCosts tc ON o.ORDERNUMBER = tc.ORDERNUMBER

GROUP BY o.ORDERNUMBER, c.CUSTOMERNAME, tc.TOTALPRICE

HAVING tc.TOTALPRICE =

(SELECT MAX(TotalPrice) FROM TotalCosts);



52.Show all of the customers who have ordered at least one product with the name “Ford” in it, that “Dragon Souveniers, Ltd.” has also ordered. List them in reverse alphabetical order, and do not consider the case of the letters in the customer name in the ordering. Show each customer no more than once. Returns 61.

SELECT customerName FROM CUSTOMERS

NATURAL JOIN ORDERS NATURAL JOIN ORDERDETAILS NATURAL JOIN PRODUCTS

WHERE productName IN

(SELECT productName FROM CUSTOMERS

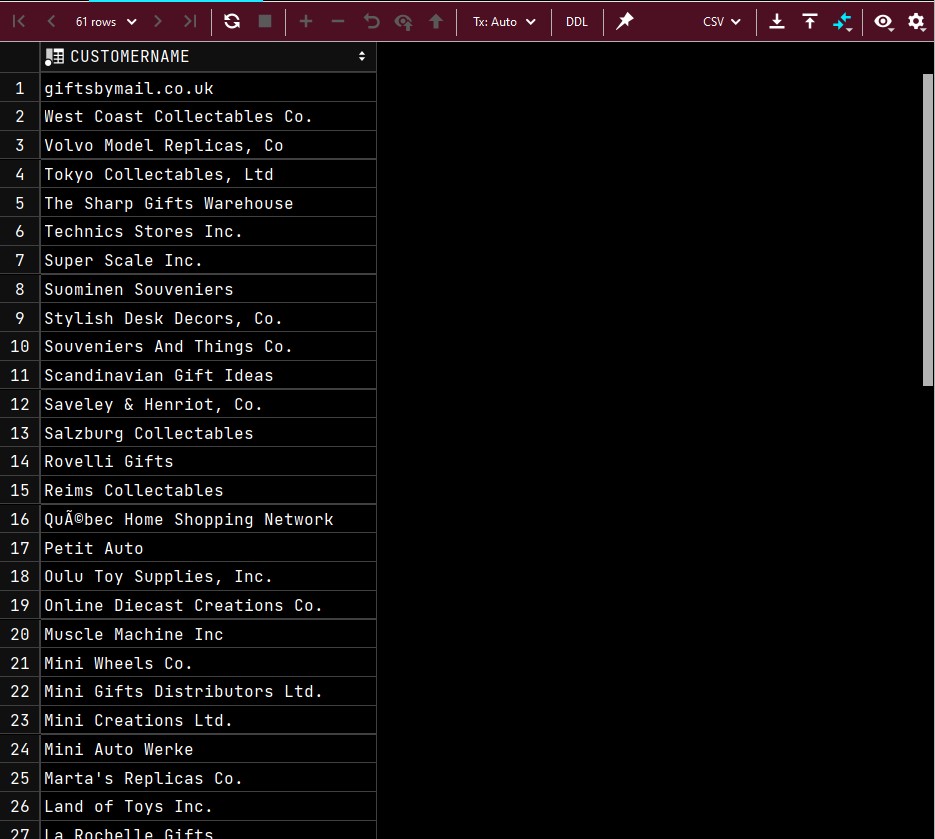
NATURAL JOIN ORDERS NATURAL JOIN ORDERDETAILS NATURAL JOIN PRODUCTS

WHERE customerName = 'Dragon Souveniers, Ltd.'

AND productname LIKE '%Ford%') GROUP BY customerName

HAVING COUNT(productName) >= 1

ORDER BY customerName DESC;



53.Which products have an MSRP within 5% of the average MSRP across all products? List the Product Name, the MSRP, and the average MSRP ordered by the product MSRP. If we denote the average MSRP as aMSRP, then the % difference between a particular MSRP and aMSRP is 100 \* (MSRP - aMSRP)/aMSRP. Returns 14.

SELECT productName, MSRP,

(SELECT AVG(MSRP)

FROM PRODUCTS) as aMSRP

FROM PRODUCTS

WHERE (100 \* ((MSRP - (SELECT AVG(MSRP)

FROM PRODUCTS)) / (SELECT AVG(MSRP)

FROM PRODUCTS))) > -5

AND

(100 \* ((MSRP - (SELECT AVG(MSRP)

FROM PRODUCTS)) /(SELECT AVG(MSRP)

FROM PRODUCTS))) < 5

ORDER BY MSRP;

