Subqueries

44. What product that makes us the most money (qty*price) across all orders for that product? Returns 1.

```
SELECT productName

FROM Products NATURAL JOIN OrderDetails

GROUP BY products.productName

HAVING SUM(priceeach*quantityOrdered) =

(SELECT MAX(productTotals.productTotal) FROM

(SELECT  productCode, sum(quantityOrdered*priceEach) AS productTotal FROM

OrderDetails

GROUP BY productCode

 ) AS productTotals);
```

#	PRODUCTNAME
1	1992 Ferrari 360 Spider red

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 products.PRODUCTLINE AS productLines FROM products

GROUP BY products.PRODUCTLINE

HAVING COUNT(products.PRODUCTVENDOR) < 5);
```

SELECT productLine, produ... ×

Max. rows: 500 Fetched Rows: 3

#	PRODUCTLINE	PRODUCTVENDOR
1	Trains	Carousel DieCast Legends
2	Trains	Gearbox Collectibles
3	Trains	Classic Metal Creations

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

```
SELECT products.productName, products.productLine

FROM Products

WHERE productLine = (SELECT products.productLine AS productLines

FROM Products

GROUP BY products.productLine

HAVING COUNT(products.productName) = (

SELECT MAX(counts.productLines) AS maxcount

FROM (SELECT productLine, COUNT(productName) AS productLines

FROM Products

GROUP BY products.productLine) AS counts));
```

Max. rows: 500 Fetched Rows: 38

#	PRODUCTNAME	PRODUCTLINE
1	1952 Alpine Renault 1300	Classic Cars
2	1972 Alfa Romeo GTA	Classic Cars
3	1962 LanciaA Delta 16V	Classic Cars
4	1968 Ford Mustang	Classic Cars
5	2001 Ferrari Enzo	Classic Cars
6	1969 Corvair Monza	Classic Cars
7	1968 Dodge Charger	Classic Cars
8	1969 Ford Falcon	Classic Cars

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 customers.contactfirstname, customers.contactlastname FROM Customers

WHERE state = (SELECT DISTINCT state FROM Customers WHERE customers.city = 'San Francisco');

Max. rows: 500 Fetched Rows: 11

#	CONTACTFIRSTNAME	CONTACTLASTNAME
1	Susan	Nelson
2	Julie	Murphy
3	Juri	Hashimoto
4	Julie	Young
5	Mary	Young
6	Valarie	Thompson
7	Julie	Brown
8	Brian	Chandler

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 customers.customerName, customers.salesRepEmployeeNumber FROM Customers

WHERE customers.customerNumber =(SELECT orders.customerNumber FROM Orders

WHERE orders.orderNumber = (SELECT orderdetails.orderNumber FROM OrderDetails

GROUP BY orderDetails.orderNumber HAVING
SUM(orderdetails.quantityOrdered*orderDetails.priceEach) =

(SELECT MAX(totals.total) FROM (SELECT
SUM(orderdetails.quantityOrdered*orderDetails.priceEach) AS total

FROM OrderDetails GROUP BY orderDetails.orderNumber) AS totals)));

Max. rows: 500 Fetched Rows: 1

#	CUSTOMERNAME	SALESREPEMPLOYEENUMBER
1	Dragon Souveniers, Ltd.	1621

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 "Order 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

);
```

SELECT orderNumber, sum(p...

Max. rows: 145 Fetched Rows: 1

#	ORDERNUMBER	Order Total
1	10165	67392.85

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

```
SELECT customerName, orderNumber, sum(priceEach*quantityOrdered) AS "Order Total"

FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails

WHERE customerNumber =

(

    SELECT customerNumber

    FROM Orders

    WHERE orderNumber =

    (

        SELECT orderNumber

        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
    )
  )
)
AND orderNumber =
(
  SELECT orderNumber
  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
  )
)
GROUP BY customerName, orderNumber;
```

SELECT customerName, orde...

Max. rows: 145 | Fetched Rows: 1

#	CUSTOMERNAME	ORDERNUMBER	Order Total
1	Dragon Souveniers, Ltd.	10165	67392.85

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 HighestOrderCustomerNumber AS (SELECT customerNumber
  FROM Orders
  WHERE orderNumber =
  (
    SELECT orderNumber
    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
```

```
        )
    ));
```

CREATE VIEW

```
SELECT customerName, orderNumber, sum(priceEach*quantityOrdered) AS "Order Total"
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails
WHERE customerNumber =
(
    SELECT customerNumber FROM HighestOrderCustomerNumber
)
AND orderNumber =
(
    SELECT orderNumber
    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

    )

)

GROUP BY customerName, orderNumber;
```

SELECT customerName, orde... ×

Max. rows: 500 Fetched Rows: 1

#	CUSTOMERNAME	ORDERNUMBER	Order Total
1	Dragon Souveniers, Ltd.	10165	67392.85

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;

SELECT customerName FROM ... ×

Max. rows: 500 Fetched Rows: 61

#	CUSTOMERNAME
43	Danish Wholesale Imports
44	Corrida Auto Replicas, Ltd
45	Corporate Gift Ideas Co.
46	Collectables For Less Inc.
47	Clover Collections, Co.
48	Classic Legends Inc.
49	Classic Gift Ideas, Inc
50	Canadian Gift Exchange Network
51	Cambridge Collectables Co.
52	CAF Imports
53	Blauer See Auto, Co.
54	Baane Mini Imports
55	Auto AssociÃ©s & Cie.
56	Australian Gift Network, Co
57	Australian Collectors, Co.

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.

Nto sure what he wants for the avg msrp

SELECT products.productname, products.msrp, (SELECT AVG(products.msrp) FROM Products) AS average

FROM products

where products.msrp >= .95*(SELECT AVG(products.msrp) FROM Products)

AND products.msrp <= 1.05*(SELECT AVG(products.msrp) FROM Products);

Max. rows: 500 Fetched Rows: 14

#	PRODUCTNAME	MSRP	AVERAGE
1	1969 Harley Davidson Ultimate Chopper	95.70	100.4387
2	1937 Lincoln Berline	102.74	100.4387
3	1913 Ford Model T Speedster	101.31	100.4387
4	18th Century Vintage Horse Carriage	104.72	100.4387
5	Collectable Wooden Train	100.84	100.4387
6	1917 Maxwell Touring Car	99.21	100.4387
7	1936 Chrysler Airflow	97.39	100.4387
8	1980â€™s GM Manhattan Express	96.31	100.4387
9	1997 BMW F650 ST	99.89	100.4387
10	1974 Ducati 350 Mk3 Desmo	102.05	100.4387