SQL Practice 31-36

1. List the total number of products per product line where number of products > 3 (6)

SELECT productline, COUNT(productcode) AS numofproducts  
FROM Products  
GROUP BY productline  
HAVING COUNT(productcode) > 3;

A screenshot of a computer

Description automatically generated with medium confidence

1. List the orderNumber and order total for all orders that totaled more than $60,000.00. (3)

SELECT ordernumber, SUM(quantityordered\*priceeach) AS ordertotal  
FROM OrderDetails  
GROUP BY ordernumber  
HAVING SUM(quantityordered\*priceeach) > 60000;

A screenshot of a computer

Description automatically generated with medium confidence

1. List the products and the profit that we have made on them. The profit in each order for a given product is (priceEach – buyPrice) \* quantityOrdered. List the product’s name and code with the total profit that we have earned selling that product. Order the rows descending by profit. Only show those products whose profit is greater than $60,000.00. (11)

SELECT productname, productcode, SUM((priceeach-buyprice) \* quantityordered) AS profit  
FROM Products NATURAL JOIN OrderDetails  
GROUP BY productname, productcode  
HAVING SUM((priceeach-buyprice) \* quantityordered) > 60000  
ORDER BY profit DESC;

Text

Description automatically generated

1. List the average of the money spent on each product across all orders where that product appears when the customer is based in Japan. Show these products in descending order by the average expenditure (45)

SELECT productcode, avg(quantityordered\*priceeach) "average value"  
FROM OrderDetails inner join orders using (ordernumber)  
inner join customers cust using (customernumber)  
where cust.country = 'Japan'  
group by productcode  
order by avg(quantityordered\*priceeach) desc;

A picture containing graphical user interface

Description automatically generated

1. What is the profit per product (MSRP-buyprice). Order by productName. (110)

SELECT productname, (msrp-buyprice)  
FROM products;

Text

Description automatically generated

1. List the Customer Name and their total orders (quantity \* priceEach) across all orders that the customer has ever placed with us, in descending order by order total for those customers who have ordered more than $100,000.00 from us ordered by Customer total in descending order. (32)

SELECT customername, SUM(quantityordered\*priceeach) AS allorderstotal  
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails  
GROUP BY customername  
HAVING sum(quantityordered\*priceeach) > 100000  
ORDER BY allorderstotal DESC;

Graphical user interface, text

Description automatically generated