```
--**Easy:**

--1. Show the first name and the email address of customer with CompanyName 'Bike World'
--Anca: let's look through the views:
select FirstName, EmailAddress, ContactType, Name
from Sales.vStoreWithContacts
where Name= 'Bike World'

--2. Show the CompanyName for all customers with an address in City 'Dallas'.
select Name, AddressLine1, City, StateProvinceName
from Sales.vStoreWithAddresses
where City = 'Dallas'

--3. How many items with ListPrice more than $1000 have been sold?
--ANCA: Here is a count of all the products sold with a unitprice > 1000 - as in "all the product types sold":
--select Production.Product.Name, production.Product.ListPrice, count(*) as countOfItems
select DISTINCT [Name],
                [Color],
                [StandardCost],
                [ListPrice],
                [Class],
                [DaysToManufacture],
                [SizeUnitMeasureCode],
                [Style]


from Sales.SalesOrderDetail AS A
    join Production.Product AS B
        on A.ProductID = B.ProductID
where ListPrice > 1000
ORDER BY ListPrice

SELECT [Name]  FROM [Production].[Product] WHERE Name LIKE 'HL Touring Frame - Yellow, 60';
--group by Production.Product.ProductID, Production.Product.Name, Production.Product.ListPrice
--ANCA: and here is a count of all the individual units sold with that unit price condition - as in "counting each
  individual unit once - NOT just each product type once like I did above":
select sum(totalNumberOfUnitsSold)
```

```
from (
    select production.product.ProductID, sum(orderQty) as totalNumberOfUnitsSold
    from Sales.SalesOrderDetail
        join Production.Product
            on Sales.SalesOrderDetail.ProductID = Production.Product.ProductID
    where ListPrice > 1000
    group by production.product.ProductID
    ) x

--for the details supporting the second query above / aka the query inserted above: here is a list of all the products and how many were sold (based on the orderQty for each sales order item listed):
select production.product.ProductID, Production.Product.Name, Production.Product.ListPrice, sum (sales.salesorderdetail.orderQty) as totalNumberOfUnitsSold
from Sales.SalesOrderDetail
    join Production.Product
        on Sales.SalesOrderDetail.ProductID = Production.Product.ProductID
where ListPrice > 1000
group by production.product.ProductID, production.product.name, production.product.ListPrice


--4. Give the CompanyName of those customers with orders over $100,000. Include the subtotal plus tax plus freight.
--Anca: looking at the primary and foreign key descriptions, I found the relationship between the id stored as businessentityId for the store as being the same as the storeId stored in the customer record
--Also used HAVING to filter the data
select Sales.Store.Name as CompanyName, Sales.Customer.CustomerID, SalesOrderID, sum(SubTotal + TaxAmt + Freight) as orderTotal
from Sales.SalesOrderHeader
    join Sales.Customer
        on Sales.SalesOrderHeader.CustomerID = Sales.Customer.CustomerID
            join Sales.Store
                on Sales.Customer.StoreID = Sales.Store.BusinessEntityID
group by SalesOrderID, Sales.Customer.CustomerID, Sales.Store.Name
having sum(SubTotal + TaxAmt + Freight)  > 100000

--5. Find the number of left racing socks ('Racing Socks, L') ordered by CompanyName 'Riding Cycles'
select Production.Product.Name as itemName, Sales.Store.Name as companyName, sum(Sales.SalesOrderDetail.OrderQty) as totalCountOfItemsSold
```

```
from Sales.SalesOrderDetail
    join Sales.SalesOrderHeader
        on Sales.SalesOrderDetail.SalesOrderID = Sales.SalesOrderHeader.SalesOrderID
            join Sales.Customer
                on Sales.SalesOrderHeader.CustomerID = Sales.Customer.CustomerID
                    join Sales.Store
                        on Sales.Customer.StoreID = Sales.Store.BusinessEntityID
                            join Production.Product
                                on Sales.SalesOrderDetail.ProductID = Production.Product.ProductID
where Sales.Store.Name = 'Riding Cycles'
AND Production.Product.Name = 'Racing Socks, L'
group by Production.Product.Name, Sales.Store.Name


--**Medium**

--1. A "Single Item Order" is a customer order where only one item is ordered. Show the SalesOrderID and the UnitPrice
  for every Single Item Order.
select *
from Sales.SalesOrderDetail

--select ssod.salesorderid, ssod.UnitPrice, count(*) as countOfItemsInOrder
    --row_number() over
    --(partition by ssod.salesorderid order by ssod.orderqty desc) as rownum
select ssod.SalesOrderID, count(ssod.SalesOrderID) as count, ssod.unitprice
from Sales.SalesOrderDetail ssod
group by ssod.SalesOrderID, ssod.unitprice
order by ssod.SalesOrderID, count
--order by countOfItemsInOrder

--option without unitprices!!! That's why I was seeing more rows for the same order ID!!:
select ssod.salesorderid, count(*) as countOfItemsInOrder
from Sales.SalesOrderDetail ssod
group by ssod.SalesOrderID
--order by countOfItemsInOrder
order by ssod.SalesOrderID
```

```sql
--ANCA: FINAL ANSWER:
select *
from (
    select ssod.salesorderid, count(*) as countOfItemsInOrder
    from Sales.SalesOrderDetail ssod
    group by ssod.SalesOrderID
    ) tableOfSalesOrdersWithOneItem
where tableOfSalesOrdersWithOneItem.countOfItemsInOrder = 1
order by tableOfSalesOrdersWithOneItem.salesorderid


--2. Where did the racing socks go? List the product name and the CompanyName for all Customers who ordered ProductModel
   'Racing Socks'.
select pp.Name as ProductName, ss.Name
from Sales.SalesOrderDetail ssod
    join Production.Product pp
        on ssod.ProductID = pp.ProductID
            join Production.ProductModel ppm
                on pp.ProductModelID = ppm.ProductModelID
                    join Sales.SalesOrderHeader ssoh
                        on ssod.SalesOrderID = ssoh.SalesOrderID
                            join Sales.Customer sc
                                on ssoh.CustomerID = sc.CustomerID
                                    join Sales.Store ss
                                        on sc.StoreID = ss.BusinessEntityID
where ppm.Name = 'Racing Socks'
group by sc.CustomerID, ss.Name, pp.Name


--3. Show the product description for culture 'fr' for product with ProductID 736.
select *
from Production.Product pp
where pp.ProductID = 736

select *
from Production.Culture
```

```sql
select *
from Production.ProductDescription

--Anca: using the view!!
select *
from Production.vProductAndDescription pvpad
where pvpad.ProductID = 736 AND pvpad.CultureID = 'fr'

--4. Use the SubTotal value in SaleOrderHeader to list orders from the largest to the smallest.
--For each order show the CompanyName and the SubTotal and the total weight of the order.
select *
from Sales.SalesOrderDetail

--get company name and subtotal for each order:
select ss.Name as CompanyName, ssoh.SalesOrderID, ssoh.CustomerID, ssoh.SubTotal
from Sales.SalesOrderHeader ssoh
    join Sales.Customer sc
        on ssoh.CustomerID = sc.CustomerID
            join sales.Store ss
                on sc.StoreID = ss.BusinessEntityID
order by ssoh.SubTotal desc

--get weight for each product in each order:
select ssoh.SalesOrderID, pp.name as ProductName, ssod.OrderQty, pp.Weight, (
    case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
    else 0
    end) as weightPerProductType
from Sales.SalesOrderHeader ssoh
    join Sales.SalesOrderDetail ssod
        on ssoh.SalesOrderID = ssod.SalesOrderID
            join Production.Product pp
                on ssod.ProductID = pp.ProductID
group by ssoh.SalesOrderID, pp.name, ssod.OrderQty, pp.Weight,
    (
    case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
    else 0
    end)
```

```
order by ssoh.SalesOrderID

--get weight for entire order:
select tableWithProductWeights.SalesOrderID, sum(tableWithProductWeights.weightPerProductType) as totalOrderWeight
from (
    select ssoh.SalesOrderID, pp.name as ProductName, ssod.OrderQty, pp.Weight, (
        case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
        else 0
        end) as weightPerProductType
    from Sales.SalesOrderHeader ssoh
        join Sales.SalesOrderDetail ssod
            on ssoh.SalesOrderID = ssod.SalesOrderID
                join Production.Product pp
                    on ssod.ProductID = pp.ProductID
    group by ssoh.SalesOrderID, pp.name, ssod.OrderQty, pp.Weight,
        (
        case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
        else 0
        end)
--order by ssoh.SalesOrderID
        ) tableWithProductWeights
group by tableWithProductWeights.SalesOrderID
order by tableWithProductWeights.SalesOrderID

--join customer data and order weight data - FINAL ANSWER:
select ss.Name as CompanyName, ssoh.SalesOrderID, ssoh.CustomerID, ssoh.SubTotal, tableWithOrderWeights.totalOrderWeight
from Sales.SalesOrderHeader ssoh
    join Sales.Customer sc
        on ssoh.CustomerID = sc.CustomerID
            join sales.Store ss
                on sc.StoreID = ss.BusinessEntityID
                    join
                    (
                    select tableWithProductWeights.SalesOrderID, sum(tableWithProductWeights.weightPerProductType) as
                      totalOrderWeight
                    from (
                        select ssoh.SalesOrderID, pp.name as ProductName, ssod.OrderQty, (
```

```
                        case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
                        else 0
                        end) as weightPerProductType
                        from Sales.SalesOrderHeader ssoh
                            join Sales.SalesOrderDetail ssod
                                on ssoh.SalesOrderID = ssod.SalesOrderID
                                    join Production.Product pp
                                        on ssod.ProductID = pp.ProductID
                        group by ssoh.SalesOrderID, pp.name, ssod.OrderQty,
                        (
                        case when pp.weight is not null then (ssod.OrderQty * pp.Weight)
                        else 0
                        end)
--order by ssoh.SalesOrderID
                        ) tableWithProductWeights
                group by tableWithProductWeights.SalesOrderID
--order by tableWithProductWeights.SalesOrderID
                ) tableWithOrderWeights
                    on ssoh.SalesOrderID = tableWithOrderWeights.SalesOrderID
order by ssoh.SubTotal desc
--order by ssoh.SalesOrderID


--5. How many products in ProductCategory 'Cranksets' have been sold to an address in 'London'?
select *
from Production.ProductCategory
--Anca: I don't see a prod categ for cranksets ...

select *
from Production.ProductModel ppm

select *
from Production.ProductInventory

select *
from Production.vProductAndDescription
```

```
--**Hard**

--1. For each order show the SalesOrderID and SubTotal calculated three ways:
--    1.  From the SalesOrderHeader
--    2. Sum of OrderQty*UnitPrice
--    3. Sum of OrderQty*ListPrice


--include discounts??
select *
from Sales.SalesOrderDetail
where UnitPriceDiscount != 0

--get subtotal from sales order header:
select ssod.SalesOrderID, ssoh.SubTotal as SubTotalFromHeader
--select *
from Sales.SalesOrderDetail ssod
    join Sales.SalesOrderHeader ssoh
        on ssod.SalesOrderID = ssoh.SalesOrderID
group by ssod.SalesOrderID, ssoh.SubTotal
order by ssod.SalesOrderID

--get subtotal as orderqty * unit price:
select tableWithSubTotalBasedOnUnitPrice.SalesOrderID, sum(tableWithSubTotalBasedOnUnitPrice.SubTotalBasedOnUnitPrice)
  as OrderSubTotalBasedOnUnitPrice
from
    (
    select ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice) as SubTotalBasedOnUnitPrice
    from Sales.SalesOrderDetail ssod
        join Sales.SalesOrderHeader ssoh
            on ssod.SalesOrderID = ssoh.SalesOrderID
    group by ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice)
    --order by ssod.SalesOrderID
    ) tableWithSubtotalBasedOnUnitPrice
group by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
order by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
```

```sql
--get subtotal based on list price:
select tableWithProductSubTotalBasedOnListPrice.SalesOrderID, sum
  (tableWithProductSubTotalBasedOnListPrice.ProductSubTotalBasedOnListPrice) as SubTotalBasedOnListPrice
from (
    select ssod.SalesOrderID, ssod.OrderQty, ssod.ProductID, ssod.UnitPrice, pp.ListPrice, (ssod.OrderQty *
      pp.ListPrice) as ProductSubTotalBasedOnListPrice
    --select *
    from Sales.SalesOrderDetail ssod
        join Production.Product pp
            on ssod.ProductID = pp.ProductID
    ) tableWithProductSubTotalBasedOnListPrice
group by tableWithProductSubTotalBasedOnListPrice.SalesOrderID

--join all the tables: FINAL ANSWER for #1 above:
select ssod.SalesOrderID, tableWithOrderSubTotalFromHeader.SubTotalFromHeader,
  tableWithOrderSubTotalBasedOnUnitPrice.OrderSubTotalBasedOnUnitPrice,
  tableWithOrderSubTotalBasedOnListPrice.SubTotalBasedOnListPrice
from Sales.SalesOrderDetail ssod
    join (
    select ssod.SalesOrderID, ssoh.SubTotal as SubTotalFromHeader
    from Sales.SalesOrderDetail ssod
        join Sales.SalesOrderHeader ssoh
            on ssod.SalesOrderID = ssoh.SalesOrderID
    group by ssod.SalesOrderID, ssoh.SubTotal
    ) tableWithOrderSubTotalFromHeader
        on ssod.SalesOrderID = tableWithOrderSubTotalFromHeader.SalesOrderID
            join (
            select tableWithProductSubTotalBasedOnListPrice.SalesOrderID, sum
              (tableWithProductSubTotalBasedOnListPrice.ProductSubTotalBasedOnListPrice) as SubTotalBasedOnListPrice
            from (
                select ssod.SalesOrderID, ssod.OrderQty, ssod.ProductID, ssod.UnitPrice, pp.ListPrice, (ssod.OrderQty *
                  pp.ListPrice) as ProductSubTotalBasedOnListPrice
                from Sales.SalesOrderDetail ssod
                    join Production.Product pp
                        on ssod.ProductID = pp.ProductID
                ) tableWithProductSubTotalBasedOnListPrice
```

```
            group by tableWithProductSubTotalBasedOnListPrice.SalesOrderID
            ) tableWithOrderSubTotalBasedOnListPrice
                on ssod.SalesOrderID = tableWithOrderSubTotalBasedOnListPrice.SalesOrderID
                    join (
                    select tableWithSubTotalBasedOnUnitPrice.SalesOrderID, sum
                      (tableWithSubTotalBasedOnUnitPrice.SubTotalBasedOnUnitPrice) as OrderSubTotalBasedOnUnitPrice
                    from (
                    select ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice) as SubTotalBasedOnUnitPrice
                    from Sales.SalesOrderDetail ssod
                        join Sales.SalesOrderHeader ssoh
                            on ssod.SalesOrderID = ssoh.SalesOrderID
                    group by ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice)
--order by ssod.SalesOrderID
                        ) tableWithSubtotalBasedOnUnitPrice
                    group by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
--order by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
                    ) tableWithOrderSubTotalBasedOnUnitPrice
                        on ssod.SalesOrderID = tableWithOrderSubTotalBasedOnUnitPrice.SalesOrderID
                    group by ssod.SalesOrderID, tableWithOrderSubTotalFromHeader.SubTotalFromHeader,
                      tableWithOrderSubTotalBasedOnUnitPrice.OrderSubTotalBasedOnUnitPrice,
                      tableWithOrderSubTotalBasedOnListPrice.SubTotalBasedOnListPrice
                    order by ssod.SalesOrderID

--trimmed down:
select tableWithOrderSubTotalFromHeader.SalesOrderID, tableWithOrderSubTotalFromHeader.SubTotalFromHeader,
  tableWithOrderSubTotalBasedOnUnitPrice.OrderSubTotalBasedOnUnitPrice,
  tableWithOrderSubTotalBasedOnListPrice.SubTotalBasedOnListPrice
from (
    select ssod.SalesOrderID, ssoh.SubTotal as SubTotalFromHeader
    from Sales.SalesOrderDetail ssod
        join Sales.SalesOrderHeader ssoh
            on ssod.SalesOrderID = ssoh.SalesOrderID
    group by ssod.SalesOrderID, ssoh.SubTotal
    ) tableWithOrderSubTotalFromHeader
        join (
        select tableWithProductSubTotalBasedOnListPrice.SalesOrderID, sum
          (tableWithProductSubTotalBasedOnListPrice.ProductSubTotalBasedOnListPrice) as SubTotalBasedOnListPrice
```

```
        from (
            select ssod.SalesOrderID, ssod.OrderQty, ssod.ProductID, ssod.UnitPrice, pp.ListPrice, (ssod.OrderQty *
              pp.ListPrice) as ProductSubTotalBasedOnListPrice
            from Sales.SalesOrderDetail ssod
                join Production.Product pp
                    on ssod.ProductID = pp.ProductID
            ) tableWithProductSubTotalBasedOnListPrice
        group by tableWithProductSubTotalBasedOnListPrice.SalesOrderID
        ) tableWithOrderSubTotalBasedOnListPrice
            on tableWithOrderSubTotalFromHeader.SalesOrderID = tableWithOrderSubTotalBasedOnListPrice.SalesOrderID
                join (
                select tableWithSubTotalBasedOnUnitPrice.SalesOrderID, sum
                  (tableWithSubTotalBasedOnUnitPrice.SubTotalBasedOnUnitPrice) as OrderSubTotalBasedOnUnitPrice
                    from (
                    select ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice) as SubTotalBasedOnUnitPrice
                        from Sales.SalesOrderDetail ssod
                            join Sales.SalesOrderHeader ssoh
                                on ssod.SalesOrderID = ssoh.SalesOrderID
                        group by ssod.SalesOrderID, (ssod.OrderQty * ssod.UnitPrice)
--order by ssod.SalesOrderID
                    ) tableWithSubtotalBasedOnUnitPrice
                group by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
--order by tableWithSubTotalBasedOnUnitPrice.SalesOrderID
                ) tableWithOrderSubTotalBasedOnUnitPrice
                    on tableWithOrderSubTotalFromHeader.SalesOrderID =
                      tableWithOrderSubTotalBasedOnUnitPrice.SalesOrderID
                group by tableWithOrderSubTotalFromHeader.SalesOrderID,
                  tableWithOrderSubTotalFromHeader.SubTotalFromHeader,
                  tableWithOrderSubTotalBasedOnUnitPrice.OrderSubTotalBasedOnUnitPrice,
                  tableWithOrderSubTotalBasedOnListPrice.SubTotalBasedOnListPrice
                order by tableWithOrderSubTotalFromHeader.SalesOrderID



--2. Show how many orders are in the following ranges (in $): --ANCA: Based on total due?? Or just subtotal? QUESTION

--```
```

```
--    RANGE      Num Orders      Total Value
--    0-  99
--  100- 999
-- 1000-9999
--10000-

--```

select *
from Sales.SalesOrderHeader

select tableWithRanges.range as Range, count(*) as [Num Orders], sum(tableWithRanges.subtotal) as [Total Value]
from (
select case
    when ssoh.SubTotal between 0 and 99 then '        0-    99'
    when ssoh.SubTotal between 100 and 999 then '    100-  999'
    when ssoh.SubTotal between 1000 and 9999 then '  1000-9999'
    else '10000- '
    end as range,
    ssoh.subtotal
from Sales.SalesOrderHeader ssoh) tableWithRanges
group by tableWithRanges.range
order by tableWithRanges.range
```