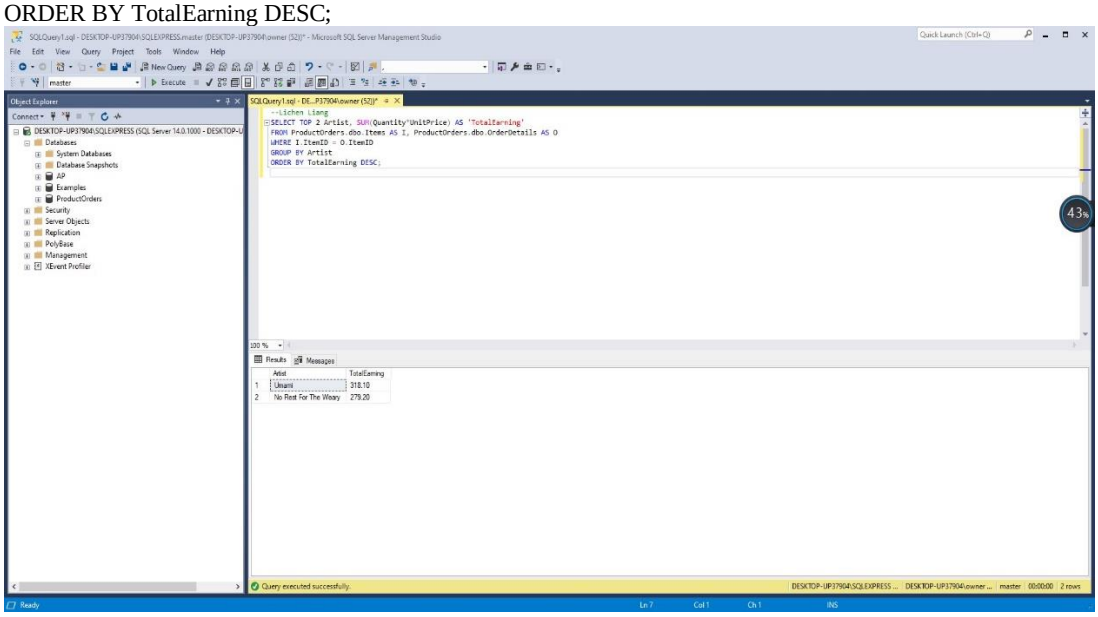
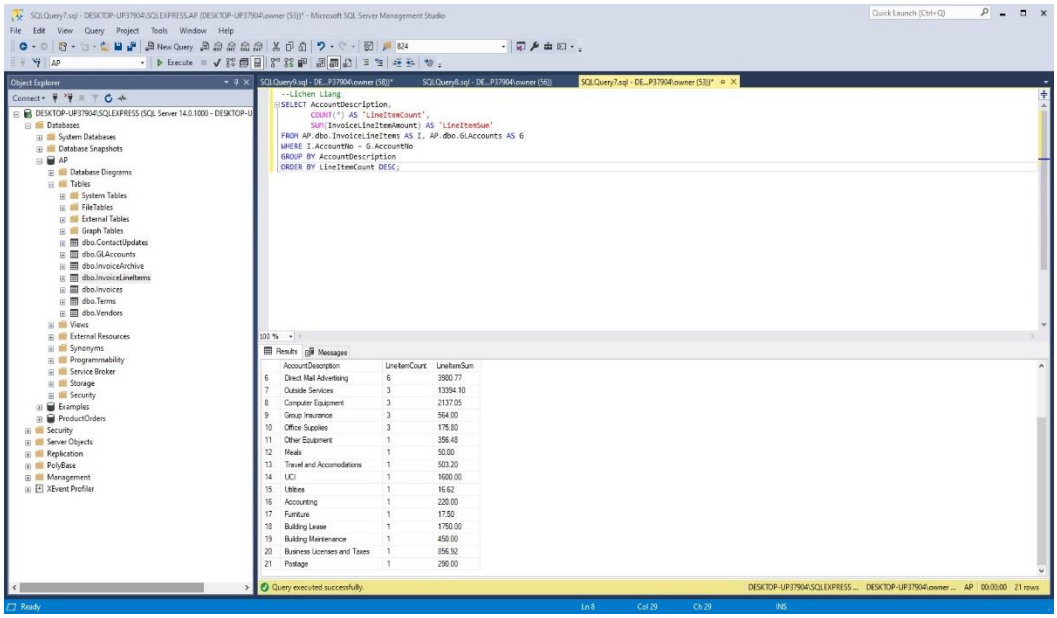
1.
```
SELECT TOP 2 Artist, SUM(Quantity*UnitPrice) AS 'TotalEarning'
FROM ProductOrders.dbo.Items AS I, ProductOrders.dbo.OrderDetails AS O
WHERE I.ItemID = O.ItemID
GROUP BY Artist
ORDER BY TotalEarning DESC;
```



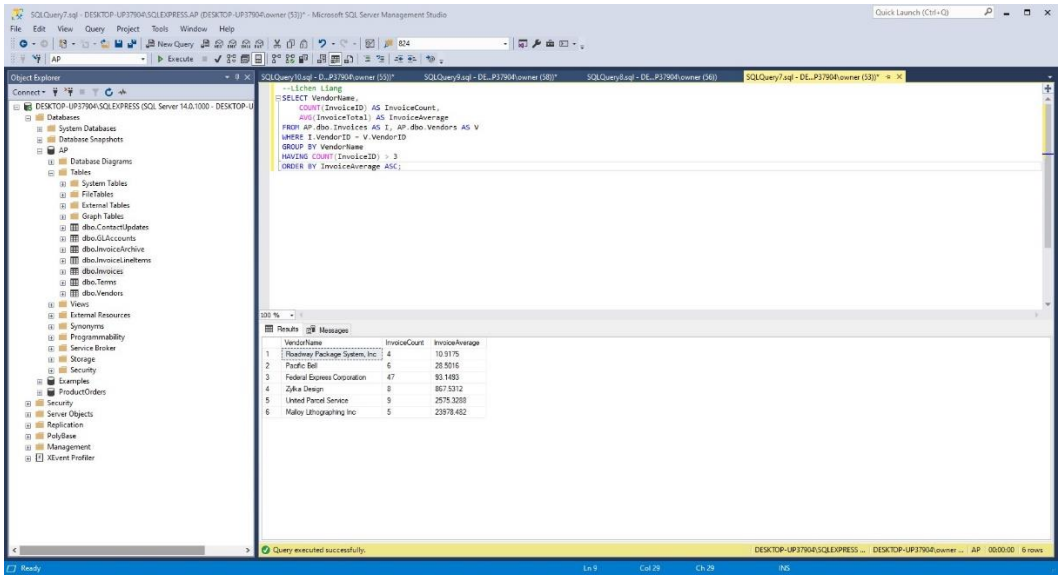
Listing the top 2 Authors with their total earnings.

2.
```
SELECT AccountDescription,
  COUNT(*) AS 'LineItemCount',
  SUM(InvoiceLineItemAmount) AS 'LineItemSum'
FROM AP.dbo.InvoiceLineItems AS I, AP.dbo.GLAccounts AS G
WHERE I.AccountNo = G.AccountNo
GROUP BY AccountDescription
ORDER BY LineItemCount DESC;
```



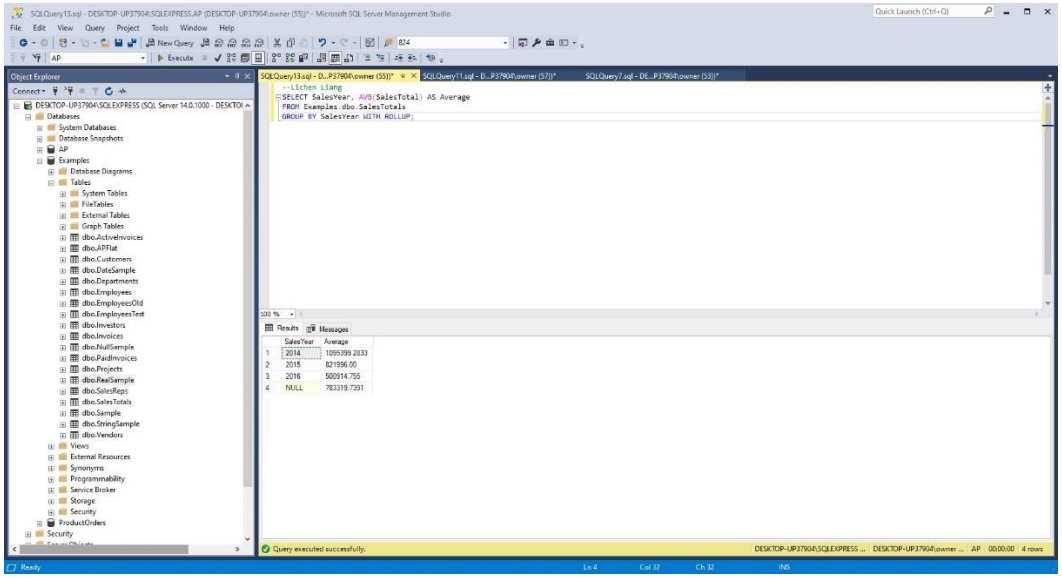
Listing account description with their line item count and the corresponding sum.

3. SELECT VendorName,
   COUNT(InvoiceID) AS InvoiceCount,
   AVG(InvoiceTotal) AS InvoiceAverage
   FROM AP.dbo.Invoices AS I, AP.dbo.Vendors AS V
   WHERE I.VendorID = V.VendorID
   GROUP BY VendorName
   HAVING COUNT(InvoiceID) > 3
   ORDER BY InvoiceAverage ASC;



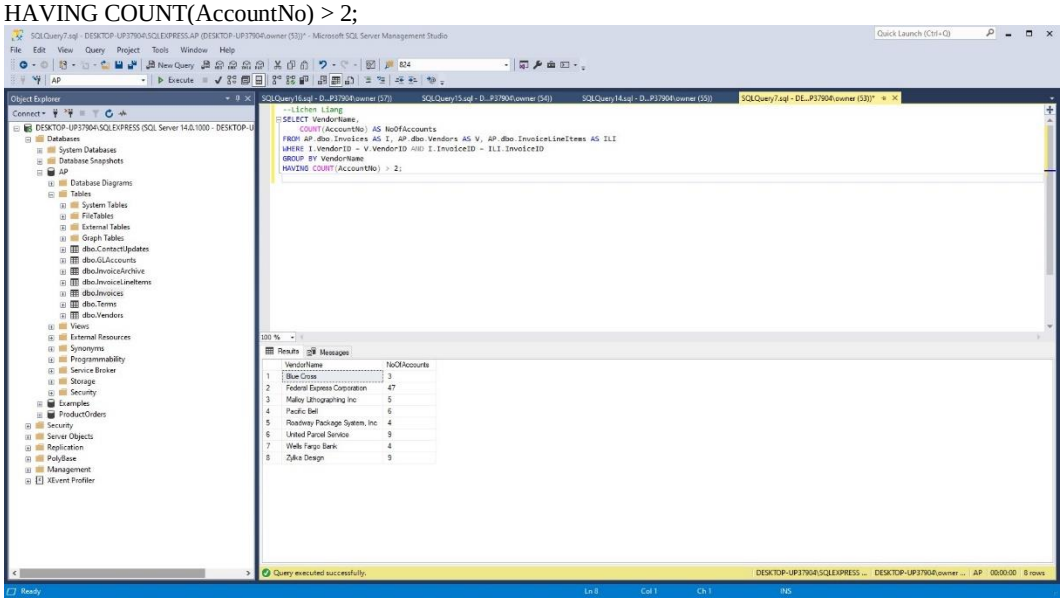
Getting the vendors with their invoice count and the average, with count more than three

4. SELECT SalesYear, AVG(SalesTotal) AS Average
   FROM Examples.dbo.SalesTotals
   GROUP BY SalesYear WITH ROLLUP;



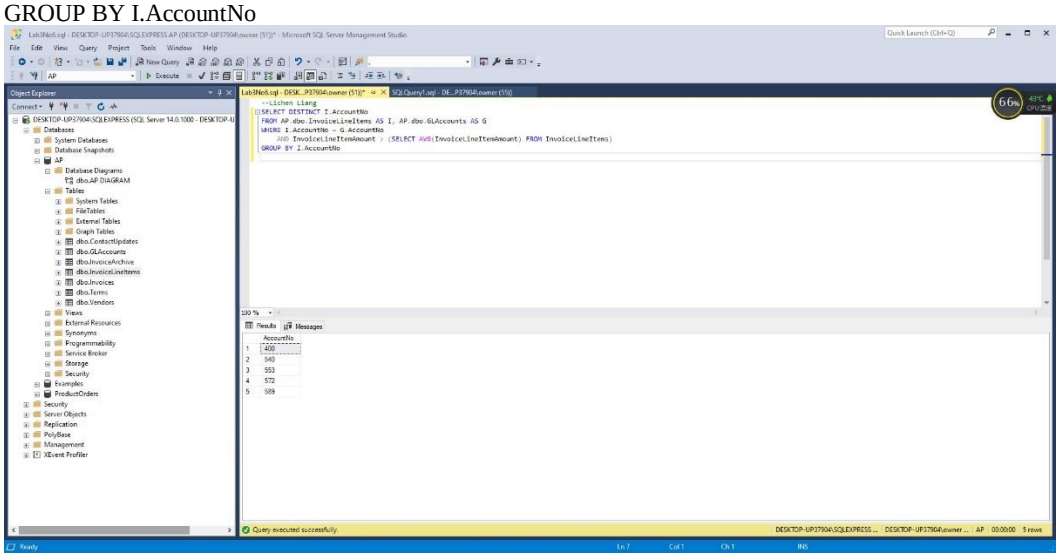
The average for 2014, 2015, 2016 are 1095399.28, 821996, 500914.755 respectfully with grand average of 783319.7391

5. SELECT VendorName,
COUNT(AccountNo) AS NoOfAccounts
FROM AP.dbo.Invoices AS I,
AP.dbo.Vendors AS V,
AP.dbo.InvoiceLineItems AS ILI
WHERE I.VendorID = V.VendorID AND I.InvoiceID = ILI.InvoiceID
GROUP BY VendorName
HAVING COUNT(AccountNo) > 2;



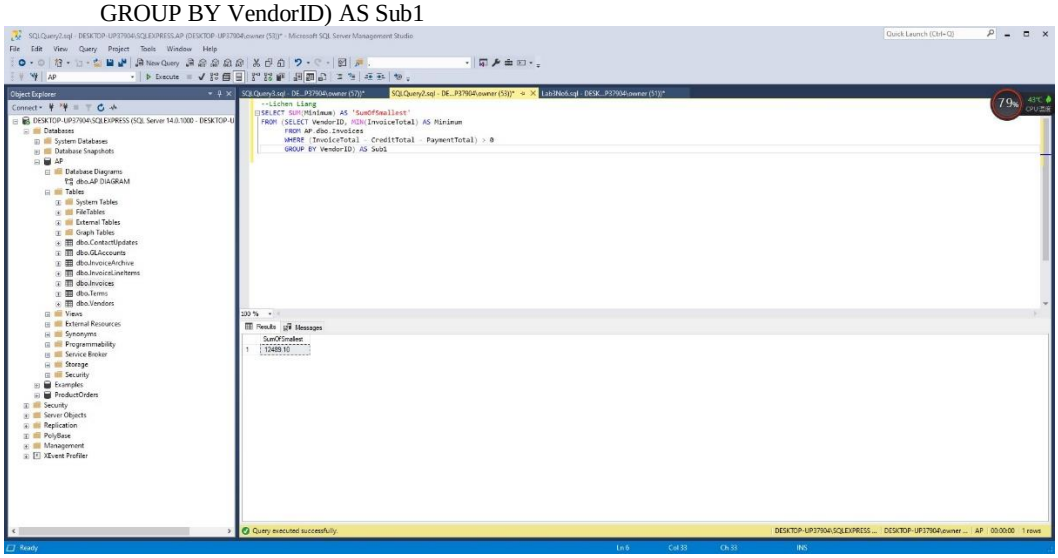
Vendors with more than 2 or more accounts.

6. SELECT DISTINCT I.AccountNo
FROM AP.dbo.InvoiceLineItems AS I, AP.dbo.GLAccounts AS G
WHERE I.AccountNo = G.AccountNo
AND InvoiceLineItemAmount >
(SELECT AVG(InvoiceLineItemAmount)
FROM InvoiceLineItems)
GROUP BY I.AccountNo



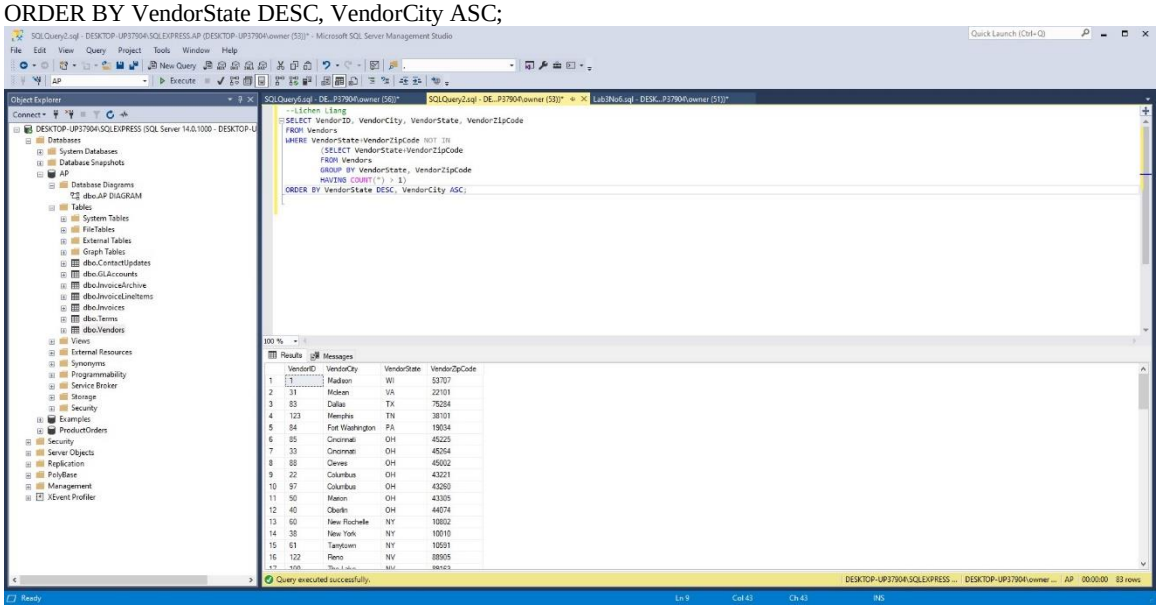
Account numbers that have Invoice Line Item Amount greater than the average of all Invoice Line Item Amount

7. SELECT SUM(Minimum) AS 'SumOfSmallest'
FROM (SELECT VendorID, MIN(InvoiceTotal) AS Minimum
FROM AP.dbo.Invoices
WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0
GROUP BY VendorID) AS Sub1



The sum of the smallest unpaid invoices.

8. SELECT VendorID, VendorCity, VendorState, VendorZipCode
FROM Vendors
WHERE VendorState+VendorZipCode NOT IN
(SELECT VendorState+Vendor ZipCode
FROM Vendors
GROUP BY VendorState, VendorZipCode
HAVING COUNT(*) > 1)
ORDER BY VendorState DESC, VendorCity ASC;



Vendors that do not share the same state and zip with another vendor

Remarks

In this lab we practiced query and subqueries. We learned how to use GROUP BY, HAVING along with multiple aggregate functions, and subqueries that are necessary for a comparison. Using the knowledge that we learned from class, we set up conditions for a query that meets the lab requirement. I think this lab is very efficient practice for lectures as they are somewhat similar. The challenge would be more specific requirements and more complicated queries.