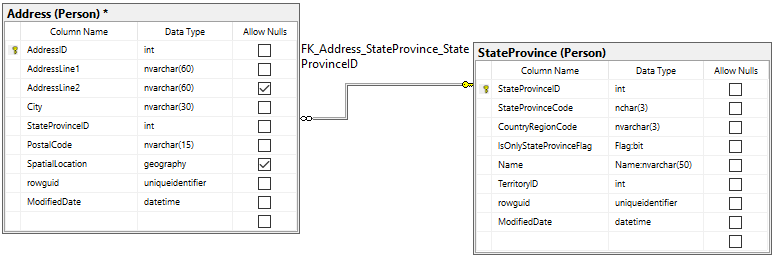
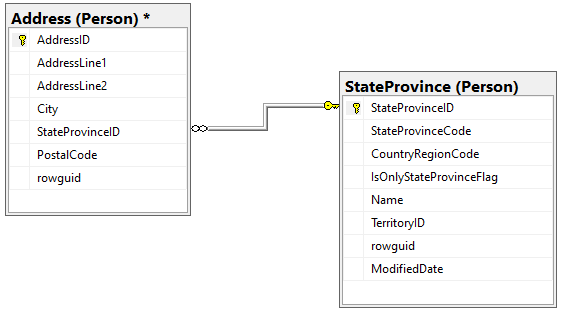
Proposition #1 (Simple)

Display for each distinct city, the name, ID, and code for the province it belongs to.



**Columns from Tables**

|  |  |
| --- | --- |
| Table Name | Column Name |
| StateProvince | StateProvinceCode, Name |
| Address | City, StateProvinceID |

**Order By**

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Address | StateProvinceID | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT DISTINCT PA.City**

**,PA.StateProvinceID**

**,PSP.StateProvinceCode**

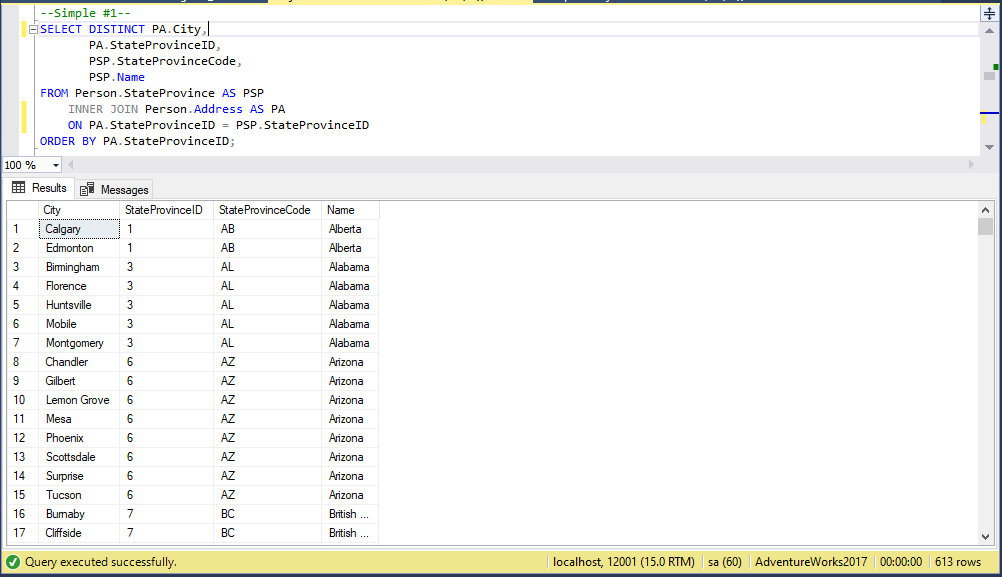
**,PSP.Name**

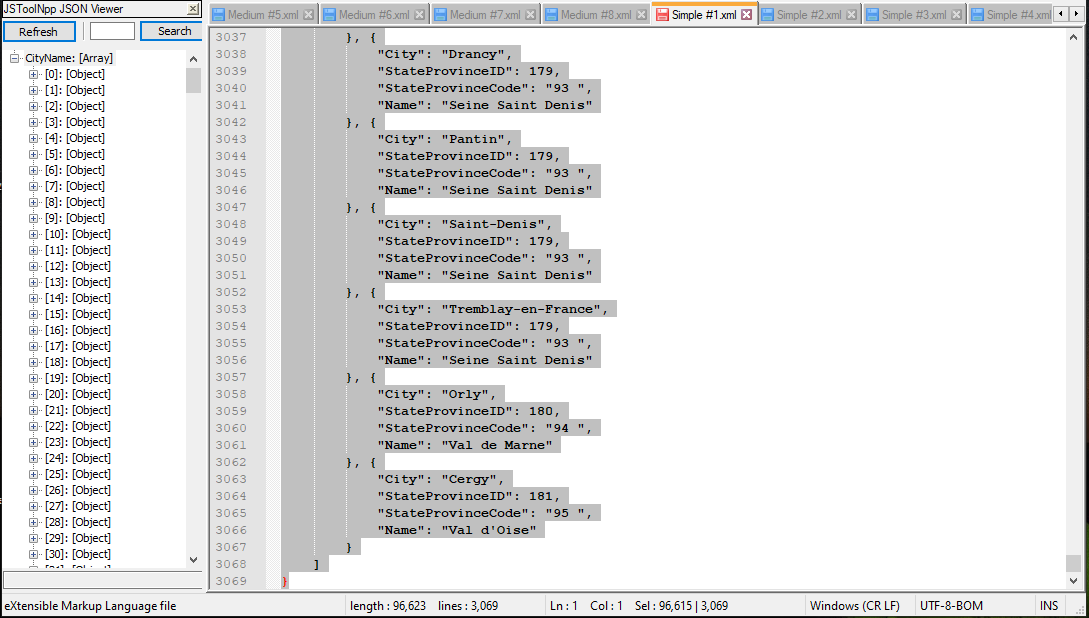
**FROM Person.StateProvince AS PSP**

**INNER JOIN Person.Address AS PA ON PA.StateProvinceID = PSP.StateProvinceID**

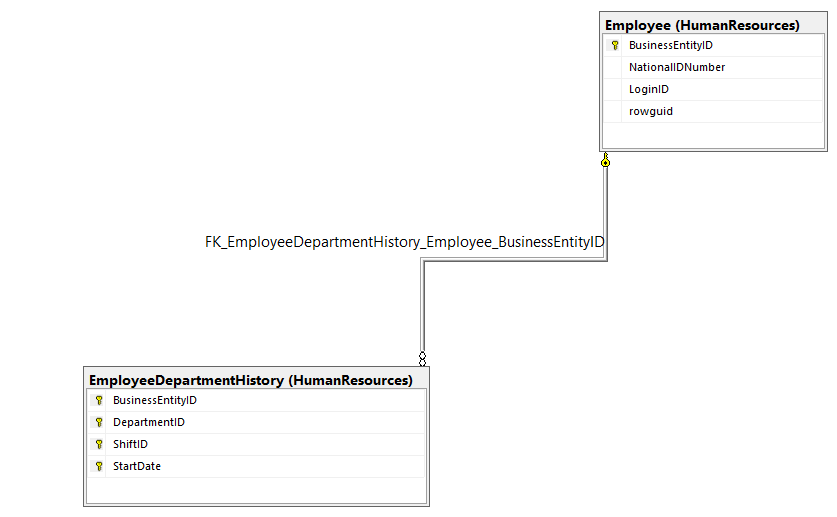
**ORDER BY PA.StateProvinceID;**

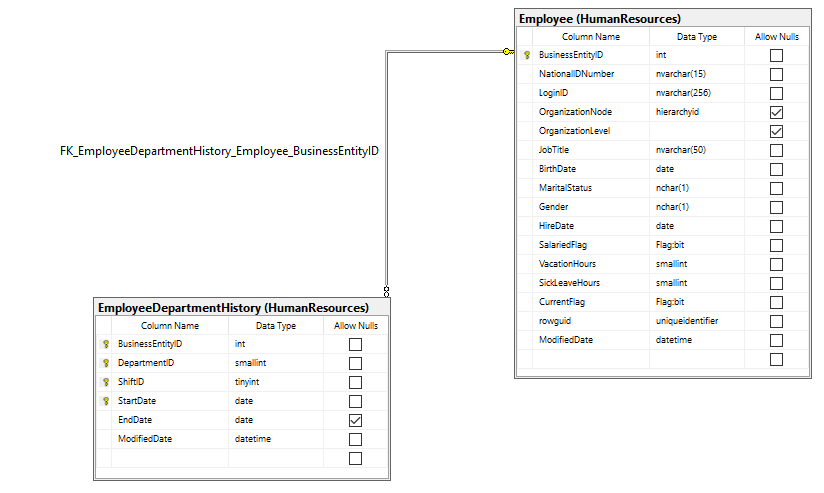
**Output**





Proposition #2 (Simple)

Display for each employee: their business entity ID, the ID of their department, shift ID, and job title



**Columns from Tables**

|  |  |
| --- | --- |
| Table Name | Column Name |
| EmployeeDepartmentHistory | BusinessEntityID, DepartmentID, ShiftID |
| Employee | JobTitle |

**Order By**

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| EmployeeDepartmentHistory | DepartmentID, ShiftID | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT HREDH.BusinessEntityID**

**,HREDH.DepartmentID**

**,HREDH.ShiftID**

**,HRE.JobTitle**

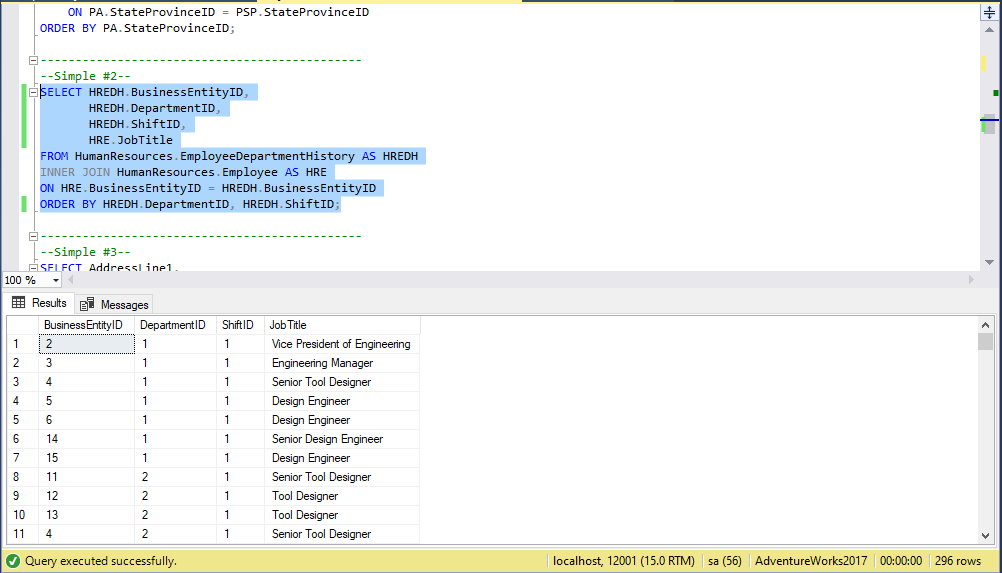
**FROM HumanResources.EmployeeDepartmentHistory AS HREDH**

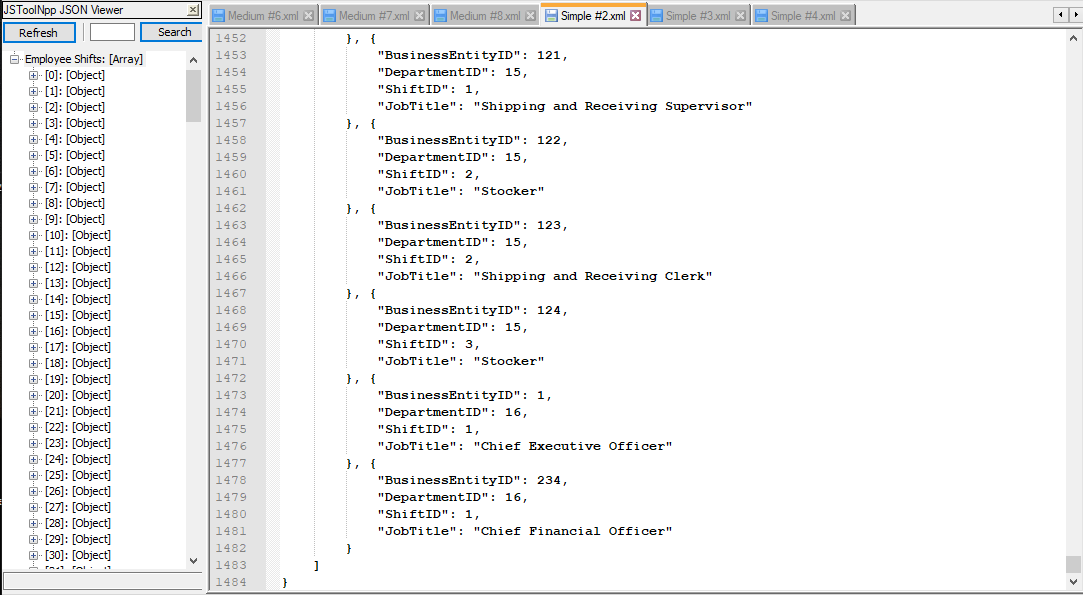
**INNER JOIN HumanResources.Employee AS HRE ON HRE.BusinessEntityID = HREDH.BusinessEntityID**

**ORDER BY HREDH.DepartmentID**

**,HREDH.ShiftID;**

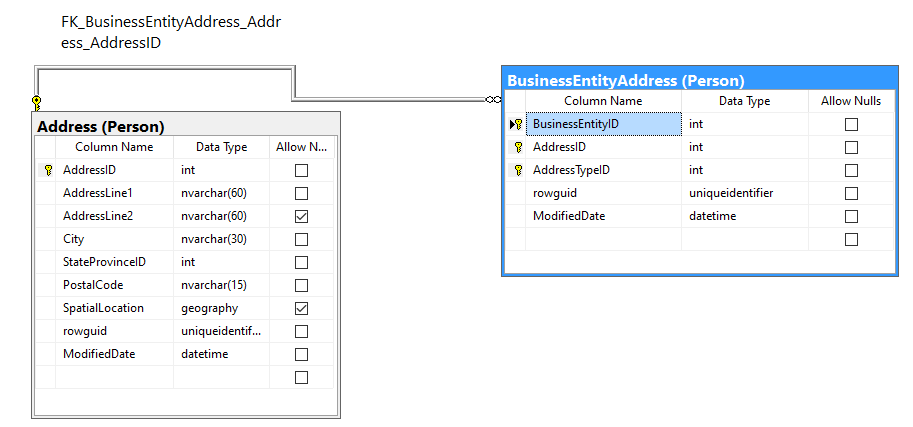
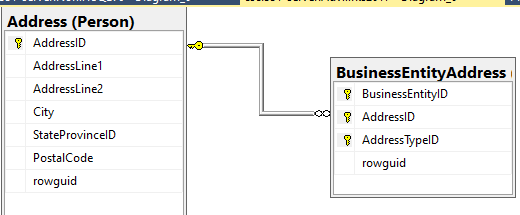
**Output**





Proposition #3 (Simple)

Display for all known addresses that do not use the 2nd address line: their address line 1, postal code, city, and address type ID



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Address | AddressLine1, PostalCode, City |
| BusinessEntityAddress | AddressTypeID |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| PersonAddress | PostalCode, City | DESC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT PA.AddressLine1**

**,PA.PostalCode**

**,PA.City**

**,PBEA.AddressTypeID**

**FROM Person.Address AS PA**

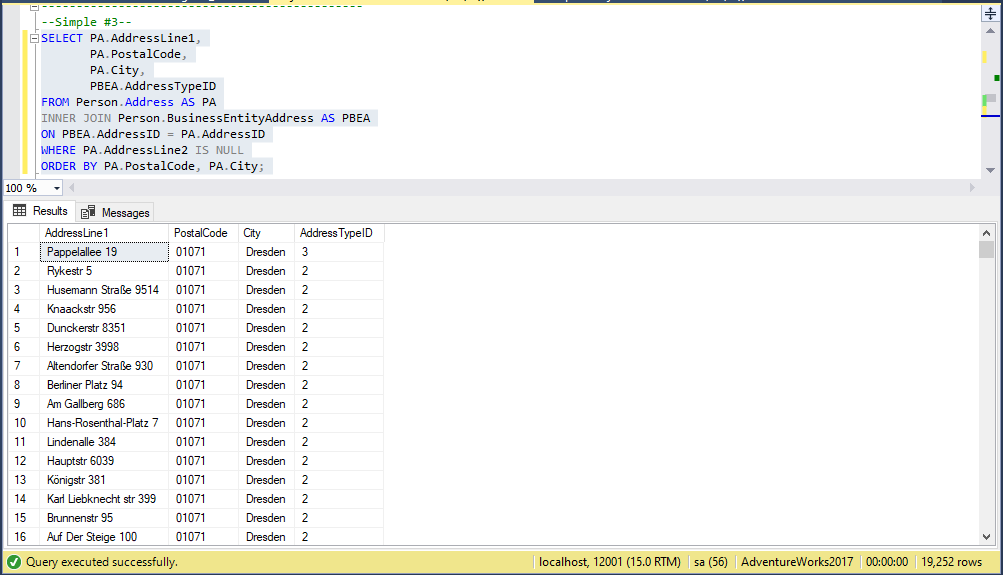
**INNER JOIN Person.BusinessEntityAddress AS PBEA ON PBEA.AddressID = PA.AddressID**

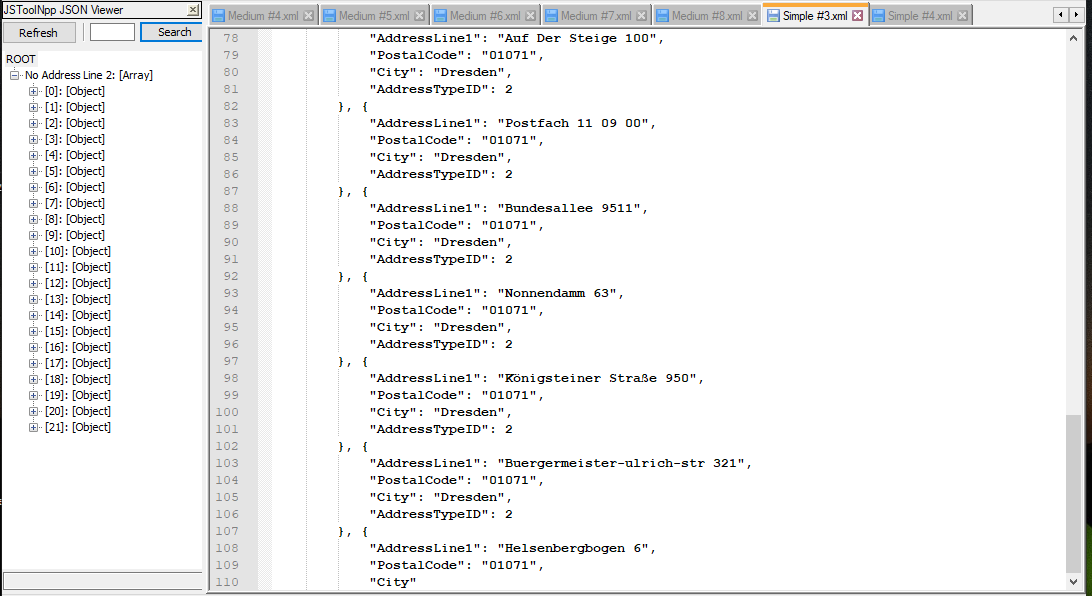
**WHERE PA.AddressLine2 IS NULL**

**ORDER BY PA.PostalCode**

**,PA.City;**

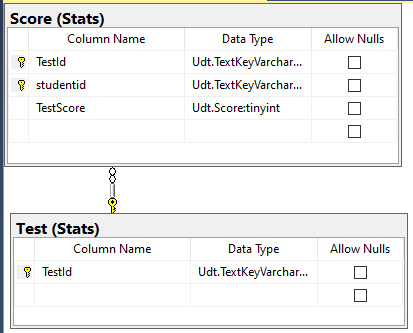
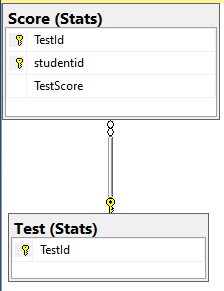
**Output**





Proposition #4 (Simple)

Display for each student: their student ID, the test, and their test score for that test



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Test | TestId, StudentId, TestScore |
| Score |  |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Test | StudentId, TestId | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT SS.studentid AS 'Student ID'**

**,SS.TestId AS 'Test ID'**

**,SS.TestScore AS 'Test Score'**

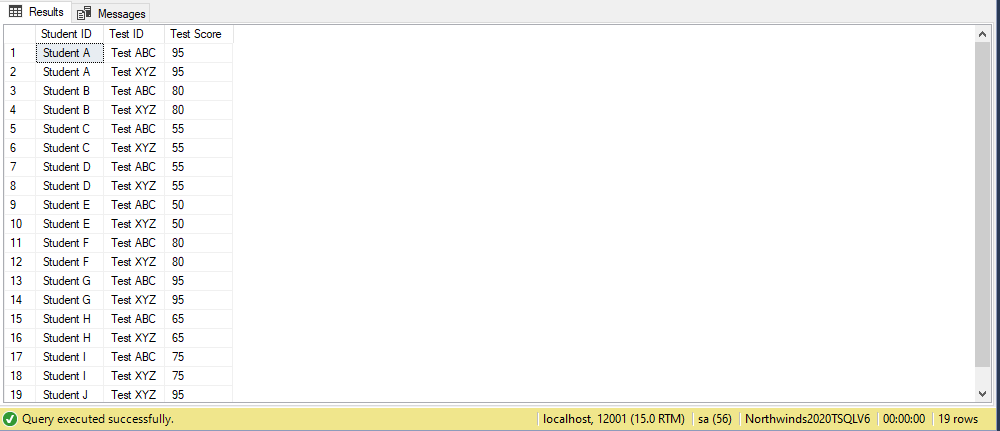
**FROM Stats.Test AS ST**

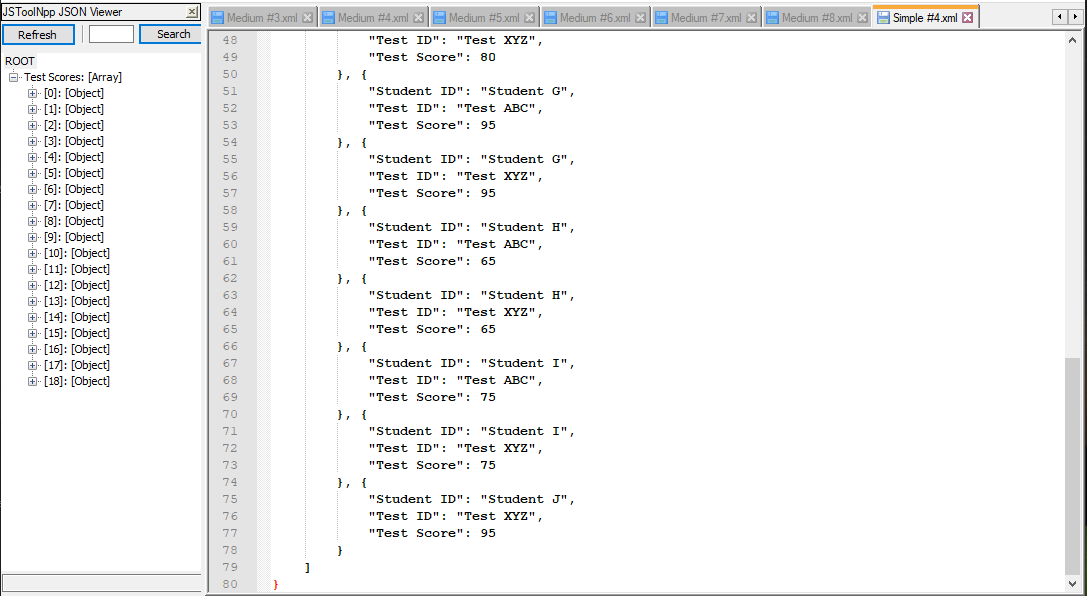
**INNER JOIN Stats.Score AS SS ON ST.TestId = SS.TestId**

**ORDER BY SS.studentid**

**,[Test ID];**

**Output**

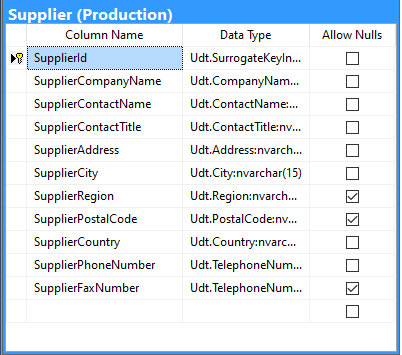




Proposition #5 (Simple)

Display for each supplier, their name, id, and phone number





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Supplier | SupplierCompanyName, SupplierId, SupplierPhoneNumber |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Supplier | SupplierCOmpnayName | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT SupplierId**

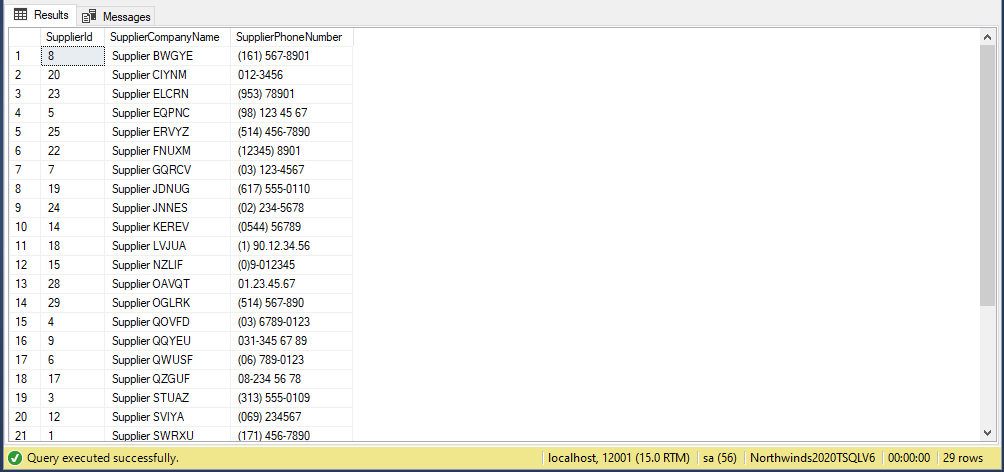
**,SupplierCompanyName**

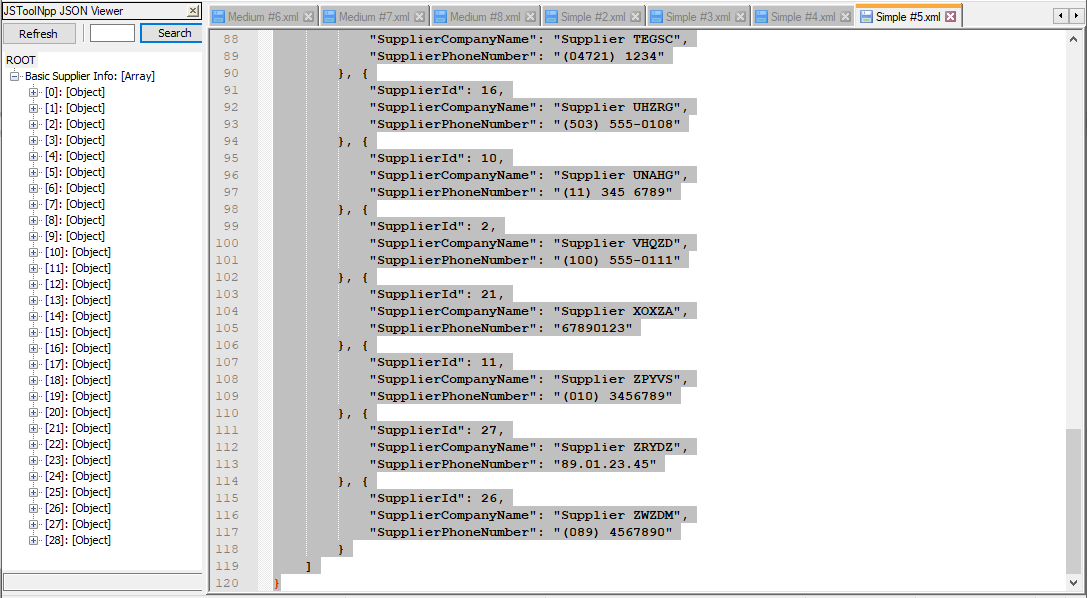
**,SupplierPhoneNumber**

**FROM Production.Supplier**

**ORDER BY SupplierCompanyName;**

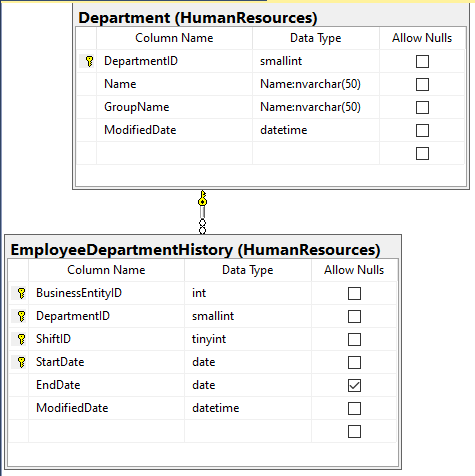
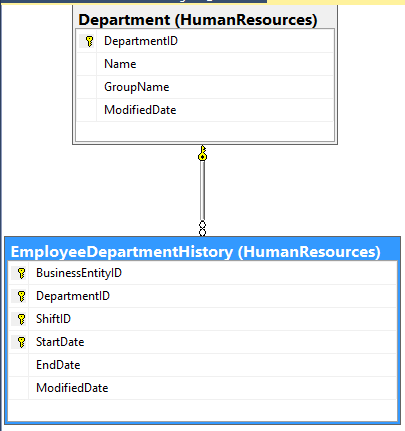
**Output**





Proposition #6 (Medium)

Display for each department: the department ID, the name of the department, the shift ID, and how many employees worked in that department during that shift



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Department | DepartmentName |
| EmployeeDepartmentHistory | DepartmentID, ShiftID |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| EmployeeDepartmentHistory | DepartmentID | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT HREDH.DepartmentID**

**,HRD.Name**

**,COUNT(HREDH.DepartmentID) AS 'Employee Number'**

**,HREDH.ShiftID**

**FROM HumanResources.Department AS HRD**

**INNER JOIN HumanResources.EmployeeDepartmentHistory AS HREDH ON HRD.DepartmentID = HREDH.DepartmentID**

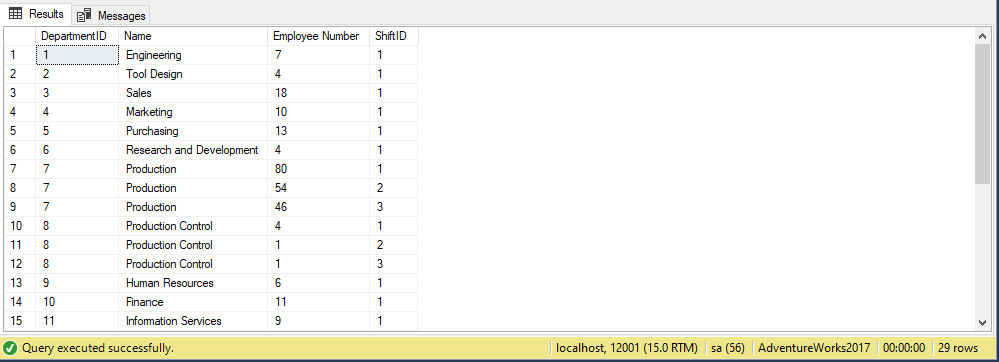
**GROUP BY HREDH.DepartmentID**

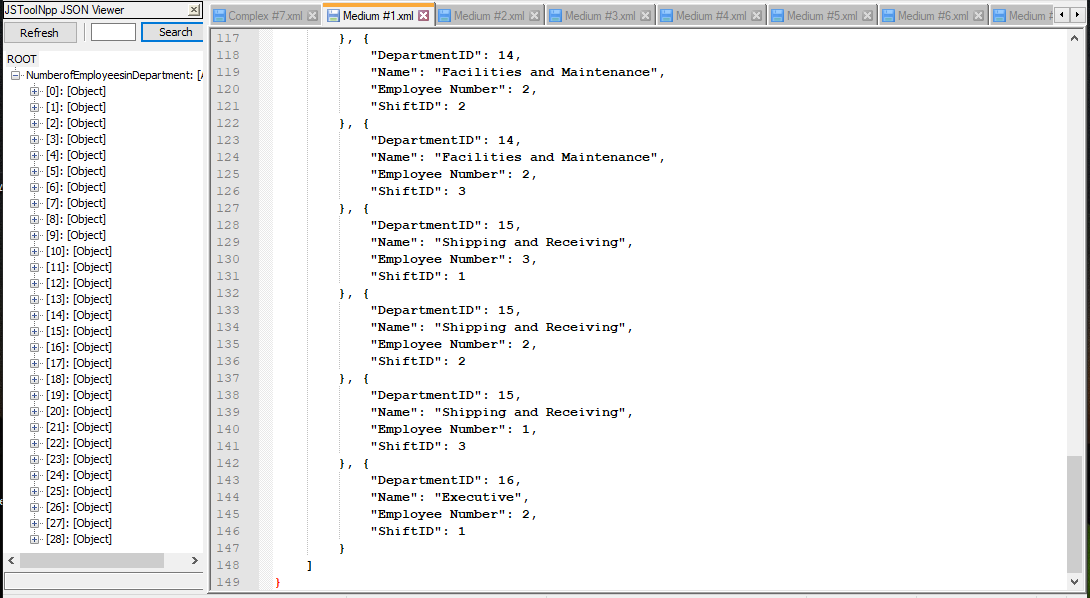
**,HREDH.ShiftID**

**,HRD.Name**

**ORDER BY HREDH.DepartmentID;**

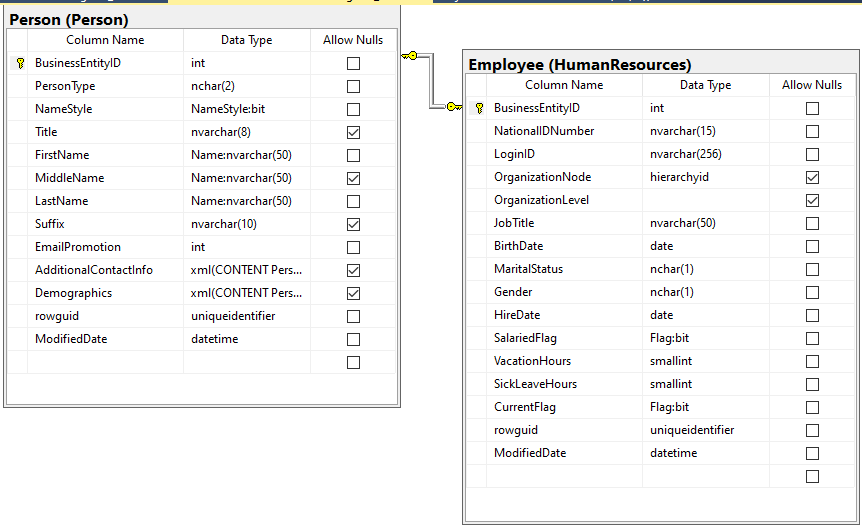
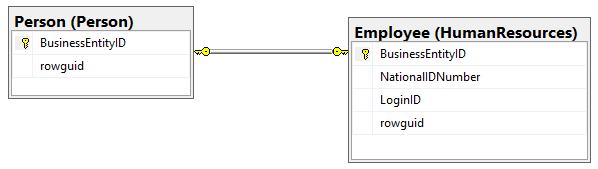
**Output**





Proposition #7 (Medium)

Display for each employee that is not a buyer: their name (formatted with last name 1st), and their job title



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Employee | JobTitle |
| Person | LastName, FirstName |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Employee | JobTitle | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT CONCAT (**

**PP.LastName**

**,' '**

**,PP.FirstName**

**) AS Name**

**,HRE.JobTitle**

**FROM HumanResources.Employee AS HRE**

**INNER JOIN Person.Person AS PP ON HRE.BusinessEntityID = PP.BusinessEntityID**

**WHERE HRE.JobTitle <> 'Buyer'**

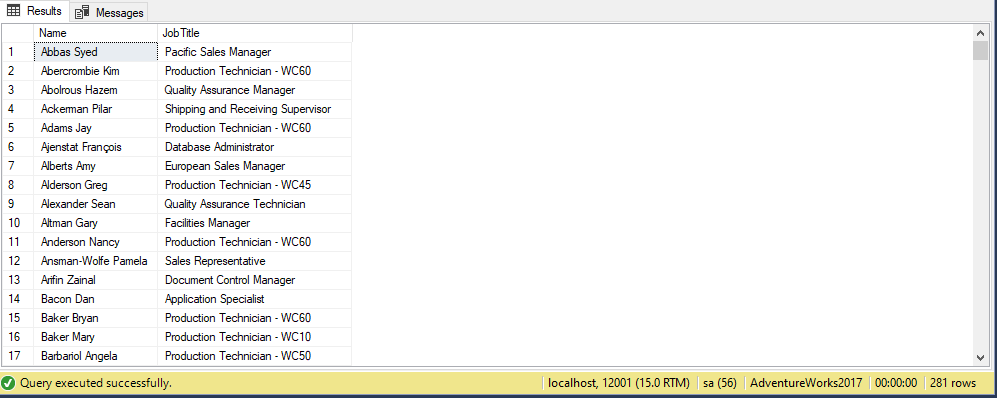
**GROUP BY PP.LastName**

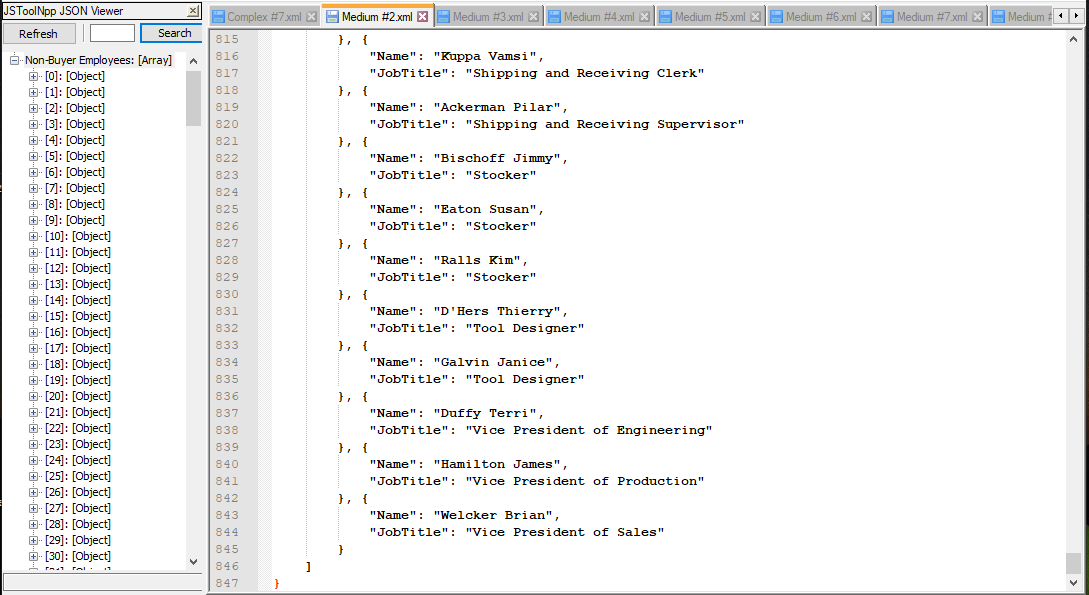
**,PP.FirstName**

**,HRE.JobTitle**

**ORDER BY HRE.JobTitle;**

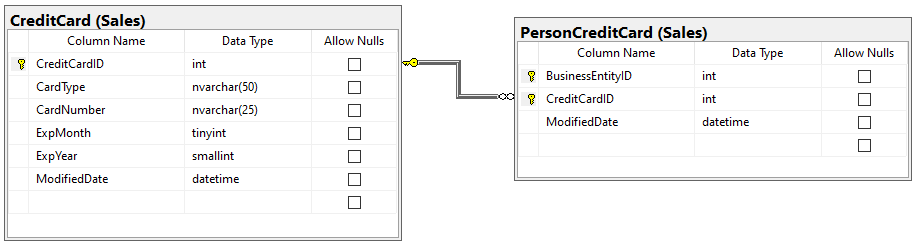
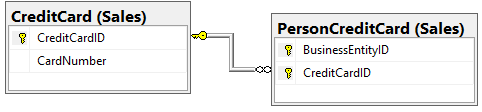
**Output**





Proposition #8 (Medium)

Display for each credit card company the number of cards that expired in 2008 with the last date it was modified not being in 2011.



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| CreditCard | CardType, COUNT(CardType) AS number of cards expired |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| CreditCard | CardType | ASC |

**Solution Query**

**USE AdventureWorks2017**

**SELECT SCC.CardType**

**,COUNT(SCC.CardType) AS 'Number of Credit Cards expired'**

**FROM Sales.CreditCard AS SCC**

**INNER JOIN Sales.PersonCreditCard AS SPCC ON SCC.CreditCardID = SPCC.CreditCardID**

**WHERE SCC.ExpYear = '2008'**

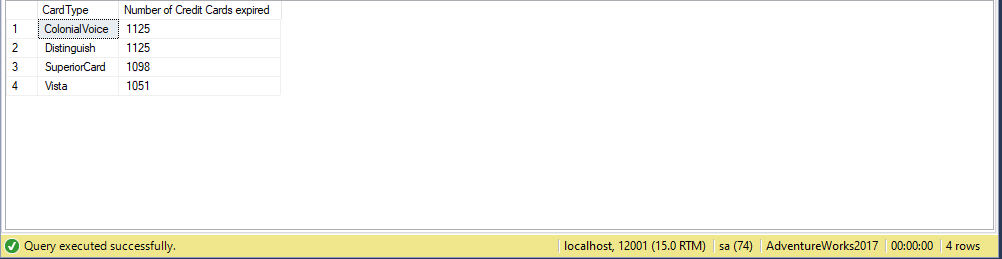
**AND YEAR(SPCC.ModifiedDate) <> '2011'**

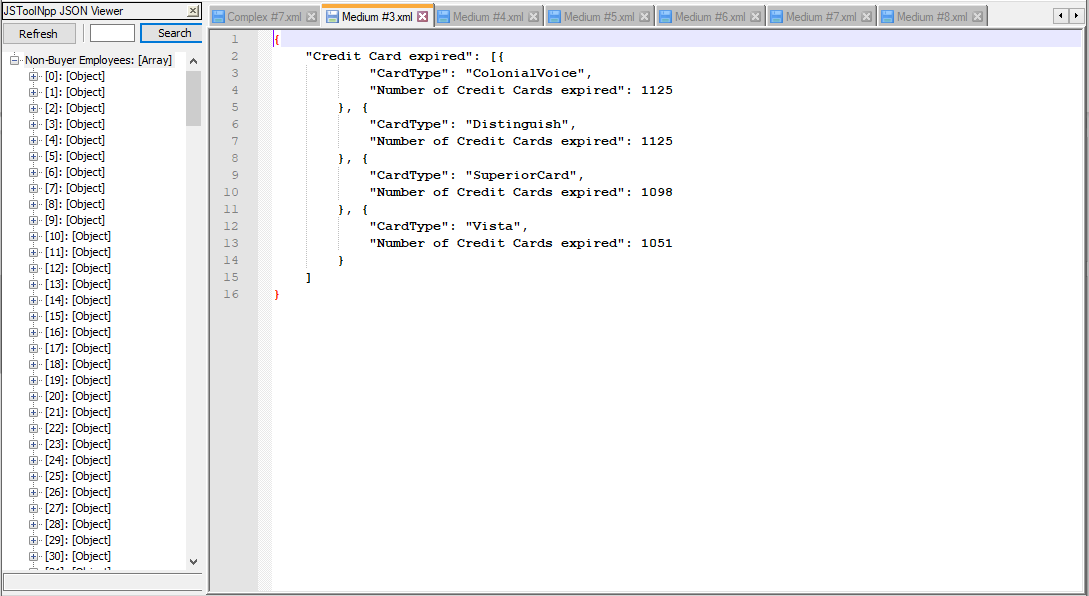
**GROUP BY SCC.CardType**

**,SCC.ExpYear**

**ORDER BY SCC.CardType ASC;**

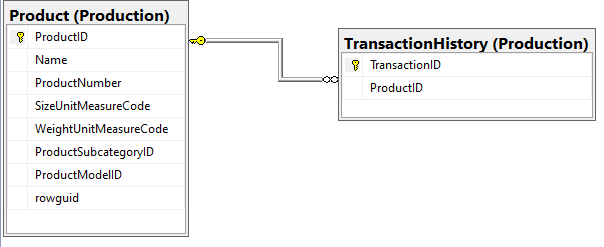
**Output**

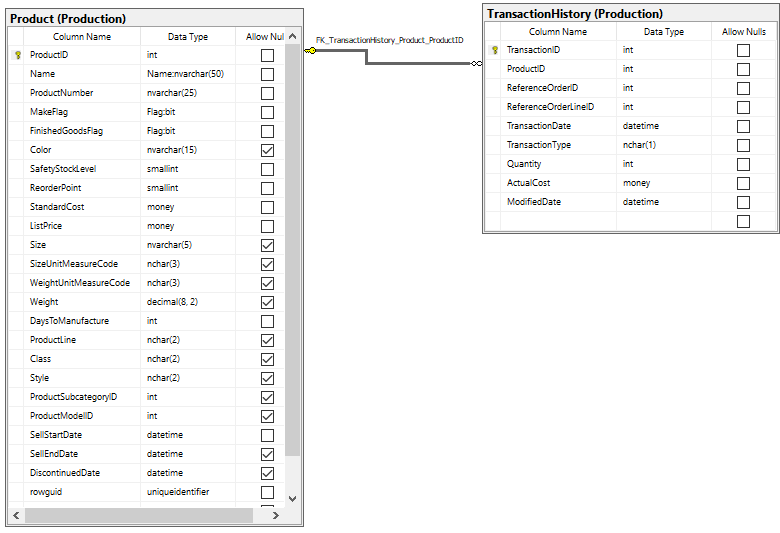




Proposition #9 (Medium)

Display for each product: the product ID, the total number of that product sold, and the name of that product





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| TransactionHistory | ProductID |
| Product | COUNT(PP.ProductID) AS Number of Items sold, ProductName |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Product | Number of Items Sold | ASC |

**Solution Query**

USE AdventureWorks2017

SELECT PTH.ProductID,

COUNT(PP.ProductID) AS 'Number of Items Sold',

PP.Name

FROM Production.Product AS PP

INNER JOIN Production.TransactionHistory AS PTH

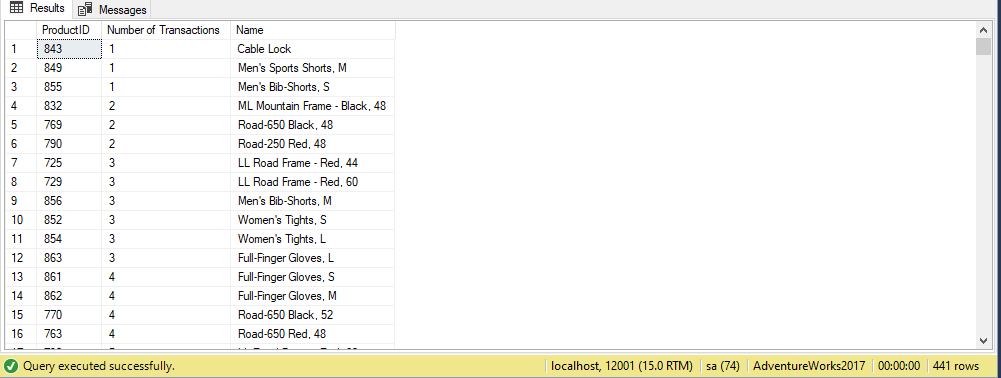
ON PP.ProductID = PTH.ProductID

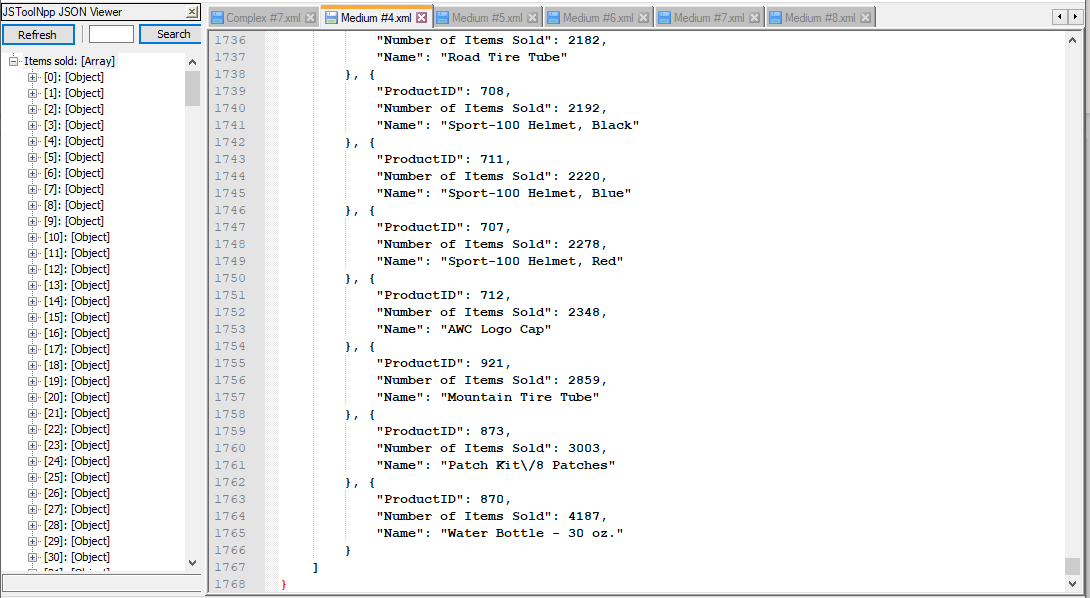
GROUP BY PTH.ProductID,

PP.Name

ORDER BY [Number of Items Sold];

**Output**





Proposition #10 (Medium)

Display for each category: the number of products it contains, the category name, and the description of the category



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Category | CategoryName, Description |
| Product | COUNT(ProductId) AS Number of Products in this category |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Category | CategoryName | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT COUNT(PP.CategoryId) AS 'Number of Products in this Category'**

**,PC.CategoryName**

**,PC.Description**

**FROM Production.Product AS PP**

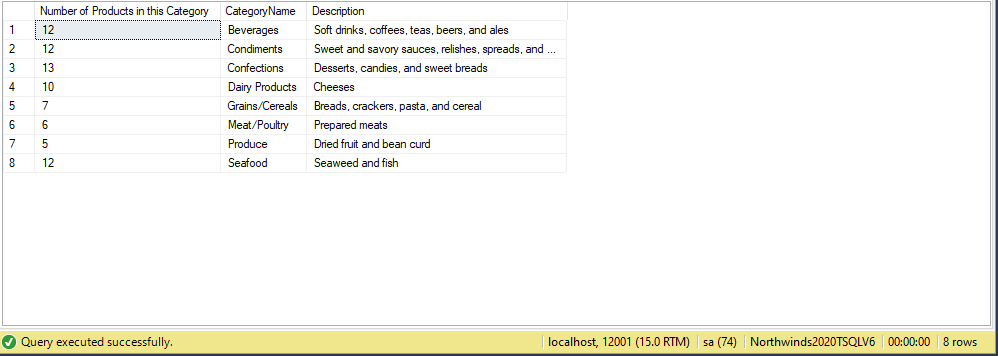
**INNER JOIN Production.Category AS PC ON PP.CategoryId = PC.CategoryId**

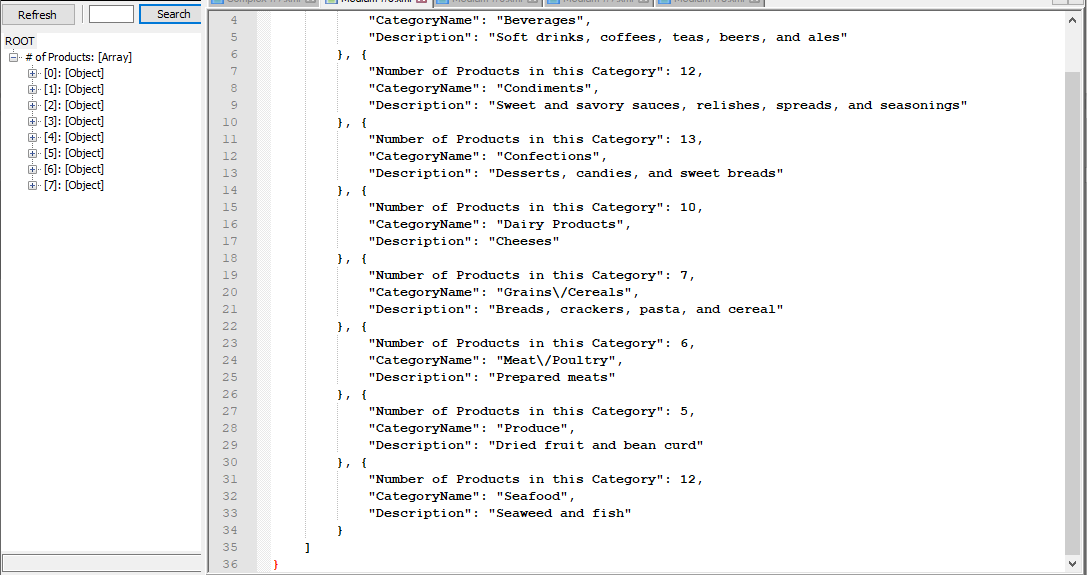
**GROUP BY PC.CategoryName**

**,PC.Description**

**ORDER BY PC.CategoryName;**

**Output**





Proposition #11 (Medium)

Display for each product: their ID, name, and price of that product.



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Product | ProductName, UnitPrice |
| OrderDetails | ProductID |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Product | ProductName | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT SOD.ProductId**

**,PP.ProductName**

**,PP.UnitPrice**

**FROM Sales.OrderDetail AS SOD**

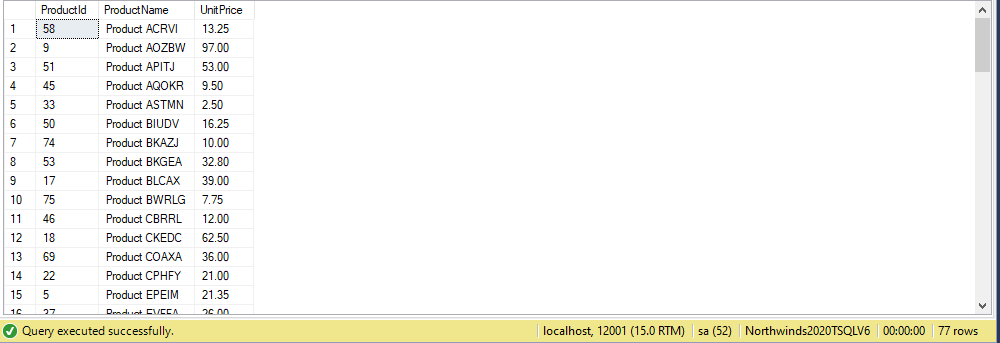
**INNER JOIN Production.Product AS PP ON SOD.ProductId = PP.ProductId**

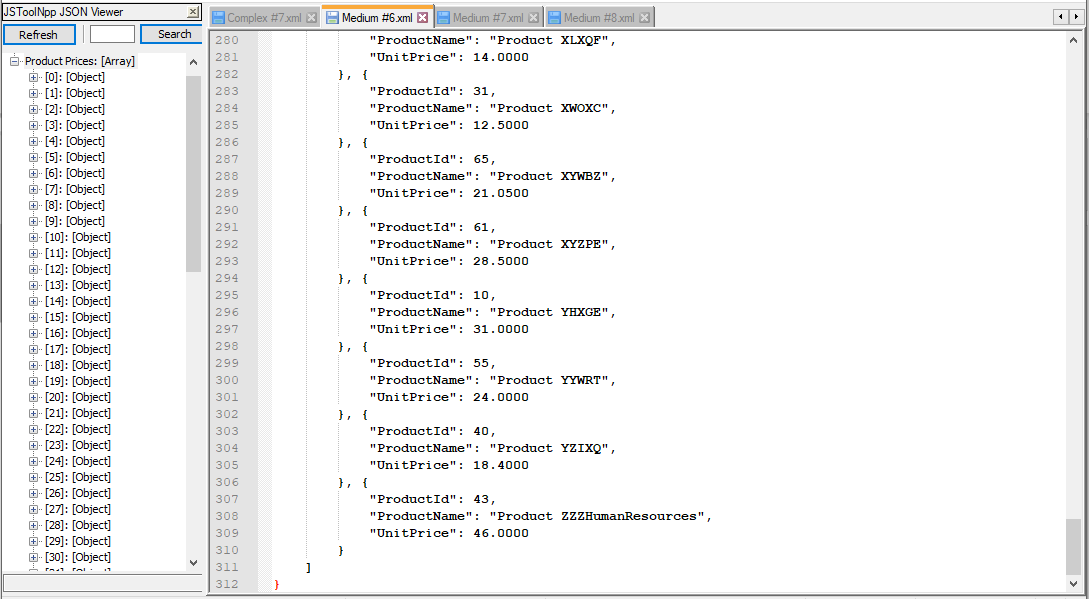
**GROUP BY SOD.ProductId**

**,PP.ProductName**

**,PP.UnitPrice;**

**Output**





Proposition #12 (Medium)

Display for each customer: their first and last name, and the date of their last survey response





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| DimCustomer | FirstName, LastName |
| FactSurveyResponse | Date |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| DimCustomer | LastName | ASC |

**Solution Query**

**USE AdventureWorksDW2017;**

**SELECT DBC.FirstName**

**,DBC.LastName**

**,MAX(DFSR.DATE) AS DATE**

**FROM dbo.DimCustomer AS DBC**

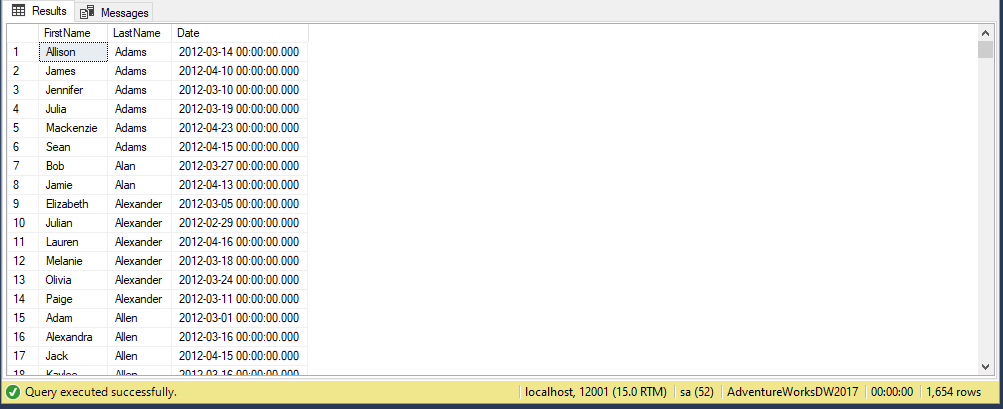
**INNER JOIN dbo.FactSurveyResponse AS DFSR ON DBC.CustomerKey = DFSR.CustomerKey**

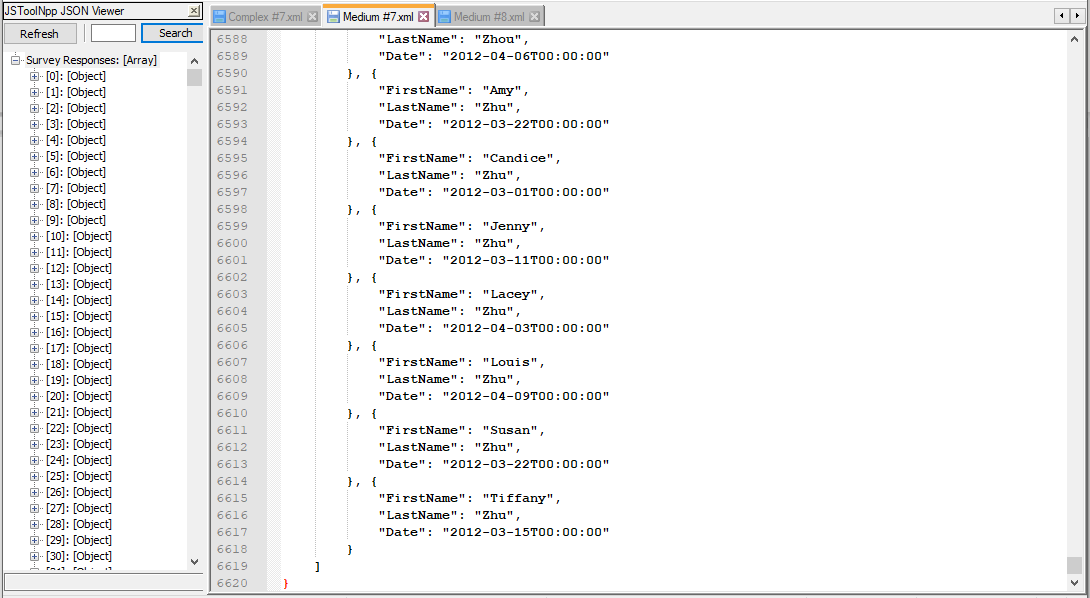
**GROUP BY DBC.FirstName**

**,DBC.LastName**

**ORDER BY DBC.LastName;**

**Output**





Proposition #13 (Medium)

Display for each employees’ first and last name, the calendar quarter, and the sales amount quota





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| DimEmployee | FirstName, LastName |
| FactsSalesQuota | CalendarQuarter, SalesAmountQuota |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| DimEmployee | Lastname | ASC |

**Solution Query**

**USE AdventureWorksDW2017;**

**SELECT CONCAT (**

**DBE.FirstName**

**,' '**

**,DBE.LastName**

**) AS EmployeeName**

**,FSQ.CalendarQuarter**

**,SUM(FSQ.SalesAmountQuota) AS Quota**

**FROM dbo.DimEmployee AS DBE**

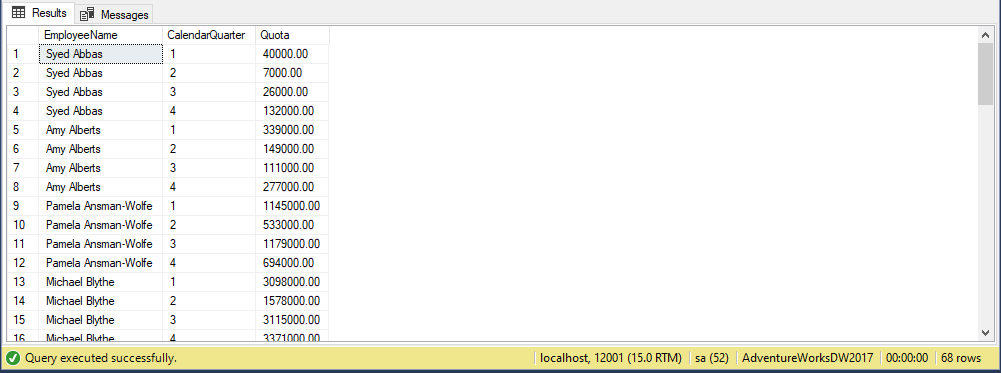
**INNER JOIN dbo.FactSalesQuota AS FSQ ON DBE.EmployeeKey = FSQ.EmployeeKey**

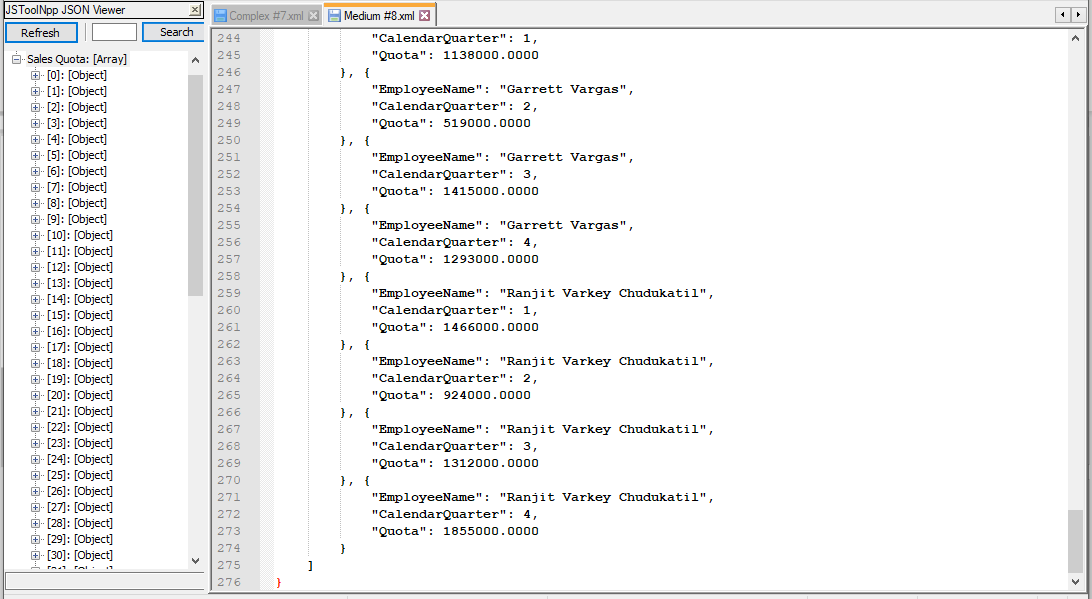
**GROUP BY DBE.LastName**

**,FSQ.CalendarQuarter**

**ORDER BY DBE.LastName;**

**Output**





Function that will be used for the next 2 queries:

USE AdventureWorks2017;

CREATE FUNCTION fnd\_name

(@BusinessEntityID int)

RETURNS varchar(15)

BEGIN

RETURN

(

SELECT LastName

FROM Person.Person

WHERE BusinessEntityID = @BusinessEntityID

);

END;

Proposition #14 (Complex)

Display for all customers whose last names come before the last name as a person with the inputted business entity ID: their first and last names, their city of origin and their business entity ID





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Person | FirstName, LastName, BusinessEntityID |
| PersonAddress | City |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Address | City, LastName | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT PA.City**

**,PP.FirstName**

**,PP.LastName**

**,PP.BusinessEntityID**

**FROM Person.BusinessEntityAddress AS PBEA**

**INNER JOIN Person.Address AS PA ON PA.AddressID = PBEA.AddressID**

**INNER JOIN Person.Person AS PP ON PP.BusinessEntityID = PBEA.BusinessEntityID**

**WHERE PP.LastName <= dbo.fnd\_name(273)**

**GROUP BY PA.City**

**,PP.LastName**

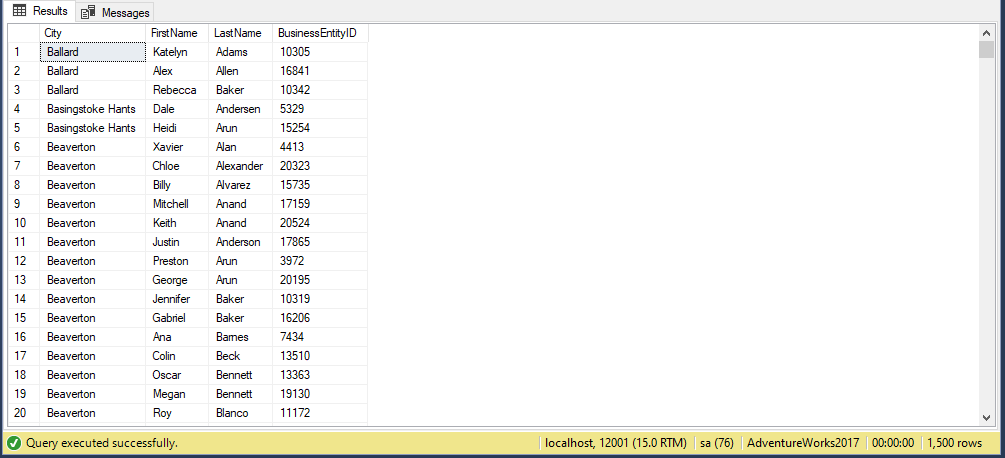
**,PP.FirstName**

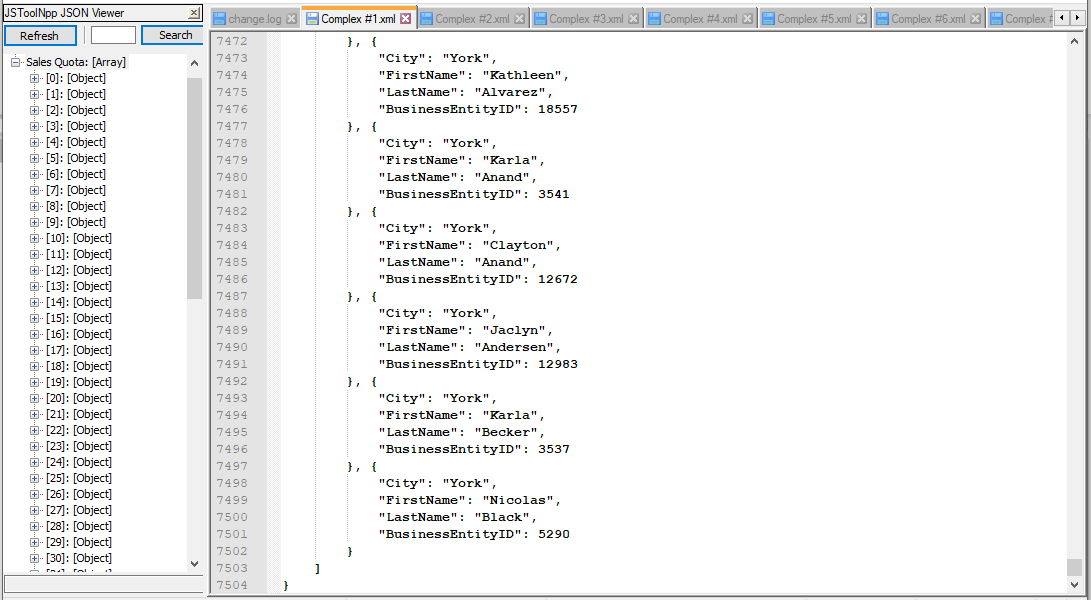
**,PP.BusinessEntityID**

**ORDER BY PA.City**

**,PP.LastName;**

**Output**





Proposition #15 (Complex)

Display for each business entity with a last name that comes alphabetically before the last name of the person with the inputted ID: their ID, name, their address ID, and the last date said address was modified

Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| BusinessEntityAddress | BusinessEntityId, AddressID, ModifiedDate |
| Person | CONCAT(FirstName, MiddleName, LastName) AS Name |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| PersonAddress | Name | ASC |

**Solution Query**

**USE AdventureWorks2017;**

**SELECT PBEA.BusinessEntityID**

**,CONCAT (**

**PP.FirstName**

**,' '**

**,PP.MiddleName**

**,' '**

**,PP.LastName**

**) AS Name**

**,PBEA.AddressID**

**,PBEA.ModifiedDate**

**FROM Person.BusinessEntityAddress AS PBEA**

**INNER JOIN Person.Person AS PP ON PP.BusinessEntityID = PBEA.BusinessEntityID**

**INNER JOIN HumanResources.EmployeeDepartmentHistory AS HREDH ON PP.BusinessEntityID = HREDH.BusinessEntityID**

**WHERE PP.LastName < dbo.fnd\_name(11)**

**GROUP BY CONCAT (**

**PP.FirstName**

**,' '**

**,PP.MiddleName**

**,' '**

**,PP.LastName**

**)**

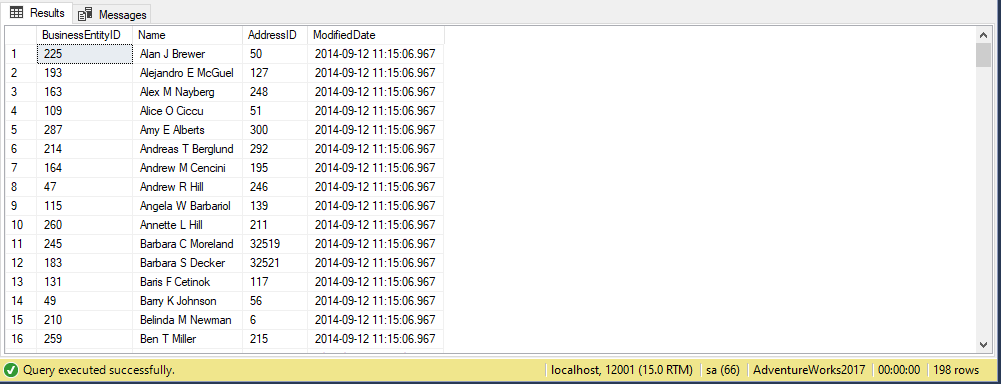
**,PBEA.AddressID**

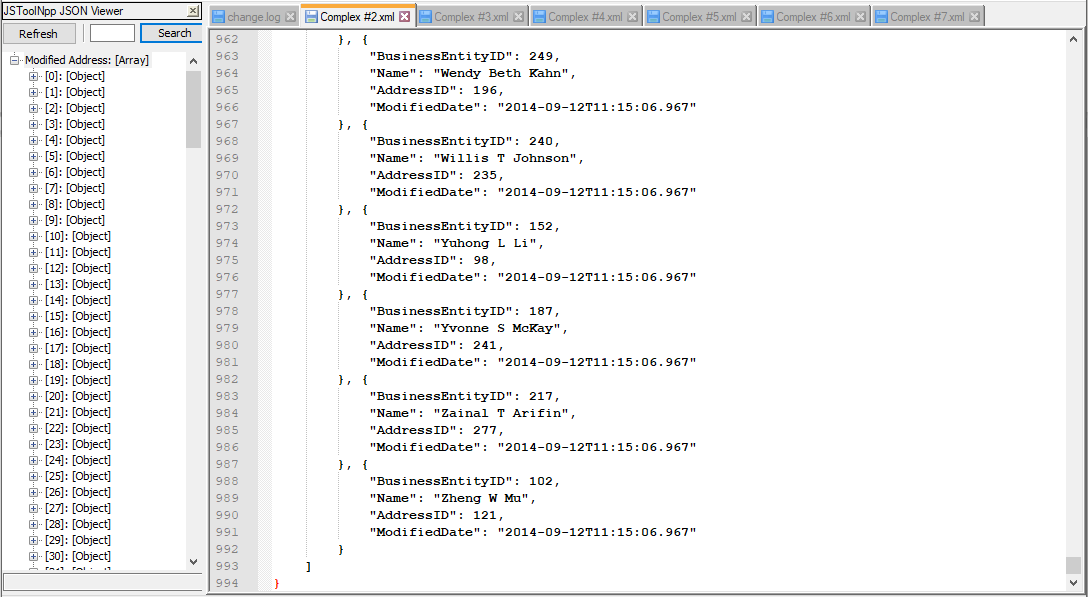
**,PBEA.ModifiedDate**

**,PBEA.BusinessEntityID**

**ORDER BY [Name] ASC;**

**Output**





Function that will be used for the next 3 queries:

USE Northwinds2020TSQLV6;

CREATE FUNCTION fnd\_suppliernm

(@SupplierID int)

RETURNS varchar(15)

BEGIN

RETURN

(

SELECT SupplierCompanyName

FROM Production.Supplier

WHERE SupplierId = @SupplierID

);

END;

Proposition #16 (Complex)

Display for all products that share the same supplier as the inputted ID: the product ID, name, and the ID of the category said product belongs to.





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Product | ProductId, ProductName |
| Category | CategoryId |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Category | CategoryID | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT PrPr.ProductId**

**,PrPr.ProductName**

**,PrC.CategoryId**

**FROM Production.Product AS PrPr**

**INNER JOIN Production.Category AS PrC ON PrPr.CategoryId = PrC.CategoryId**

**INNER JOIN Production.Supplier AS PrS ON PrS.SupplierId = PrPr.SupplierId**

**WHERE PrS.SupplierCompanyName = dbo.fnd\_suppliernm(11)**

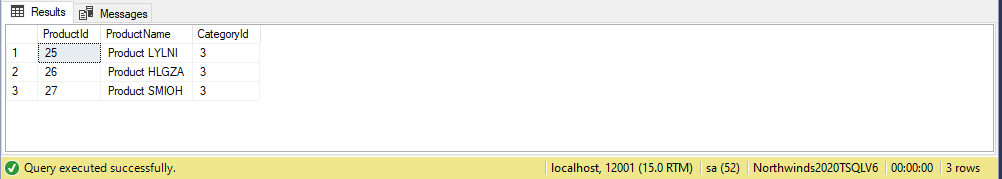
**GROUP BY PrC.CategoryId**

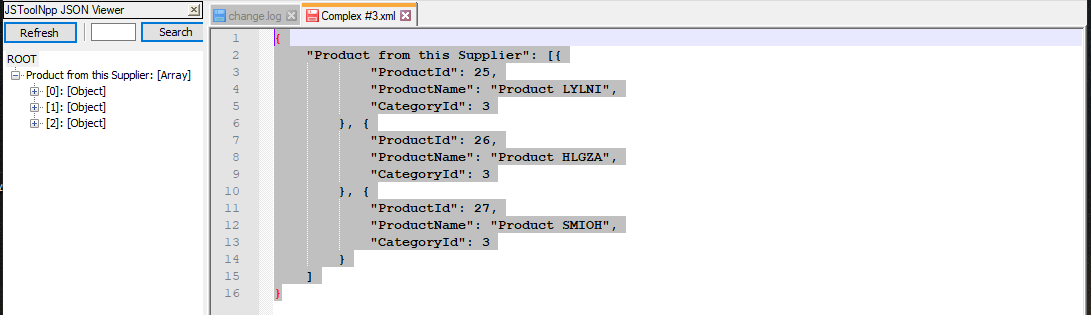
**,PrPr.ProductName**

**,PrPr.ProductId**

**ORDER BY PrC.CategoryId;**

**Output**





Proposition #17 (Complex)

Display for all products that have the same supplier as the inputted ID: the total undiscounted cost of the order of their product



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Order | OrderId, |
| OrderDetail | SUM(UnitPrice \* Quantity) AS Total UndiscountedPrice |
| Product | ProductName, SupplierId |
| Supplier | SupplierCompanyName |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| OrderDetail | TotalUndiscountedPrice | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT SO.OrderId**

**,SUM((SOD.Quantity \* SOD.UnitPrice)) AS 'Total Undiscounted Price'**

**,PrPr.ProductName**

**,PrPr.SupplierId**

**,PrS.SupplierCompanyName**

**FROM Sales.[Order] AS SO**

**INNER JOIN Sales.OrderDetail AS SOD ON SO.OrderId = SOD.OrderId**

**INNER JOIN Production.Product AS PrPr ON PrPr.ProductId = SOD.ProductId**

**INNER JOIN Production.Supplier AS PrS ON PrS.SupplierId = PrPr.SupplierId**

**WHERE PrS.SupplierCompanyName = dbo.fnd\_suppliernm(11)**

**GROUP BY PrPr.SupplierId**

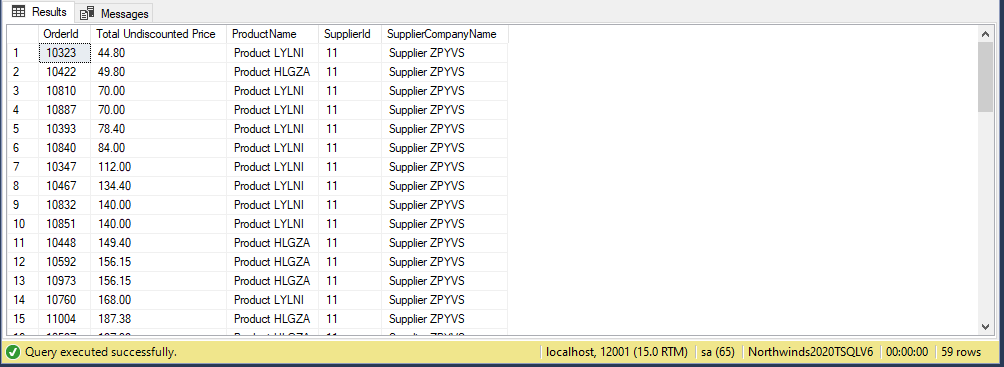
**,SO.OrderId**

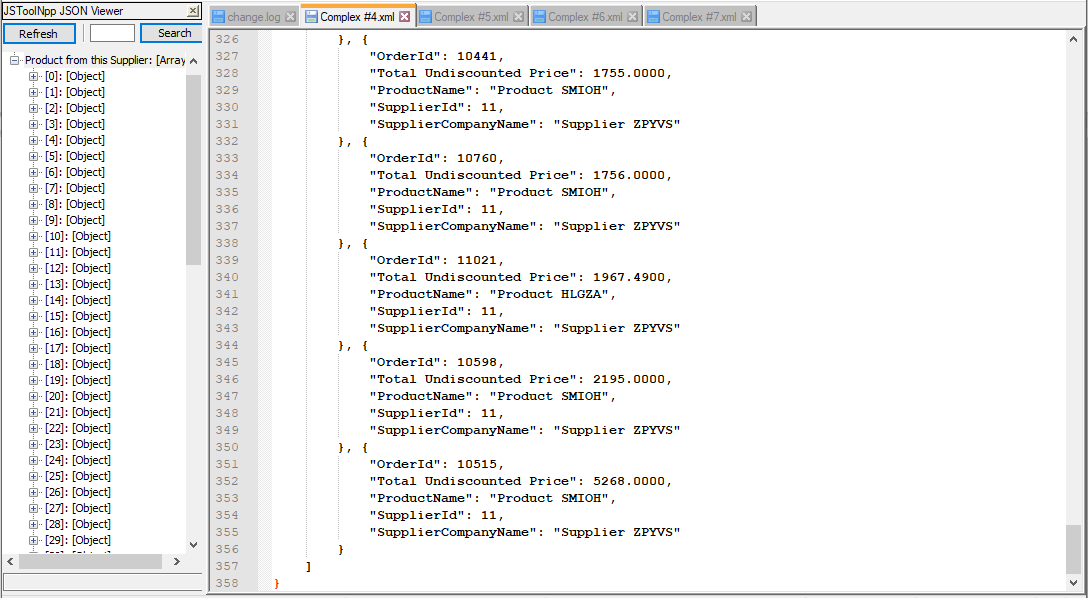
**,PrPr.ProductName**

**,PrS.SupplierCompanyName**

**ORDER BY [Total Undiscounted Price];**

**Output**





Proposition #18 (Complex)

Display for the supplier company that has ID number 11: the order ID of orders that bought products that this company supplies, the name of the products that were bought and the amount of that product bought.





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| Order | OrderId |
| OrderDetail | Quantity AS Number of Items |
| Product | ProductName |
| Supplier | SupplierCompanyName |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| OrderDetail | Quantity | ASC |
| Order | OrderId | ASC |
| Product | ProductName | ASC |

**Solution Query**

**USE Northwinds2020TSQLV6;**

**SELECT SO.OrderId**

**,SOD.Quantity AS 'Number of items'**

**,PP.ProductName**

**,PrS.SupplierCompanyName**

**FROM Sales.[Order] AS SO**

**INNER JOIN Sales.OrderDetail AS SOD ON SO.OrderId = SOD.OrderId**

**INNER JOIN Production.Product AS PP ON SOD.ProductId = PP.ProductId**

**INNER JOIN Production.Supplier AS PrS ON PrS.SupplierId = PP.SupplierId**

**WHERE PrS.SupplierCompanyName = dbo.fnd\_suppliernm(11)**

**GROUP BY PP.ProductName**

**,SO.OrderId**

**,PrS.SupplierCompanyName**

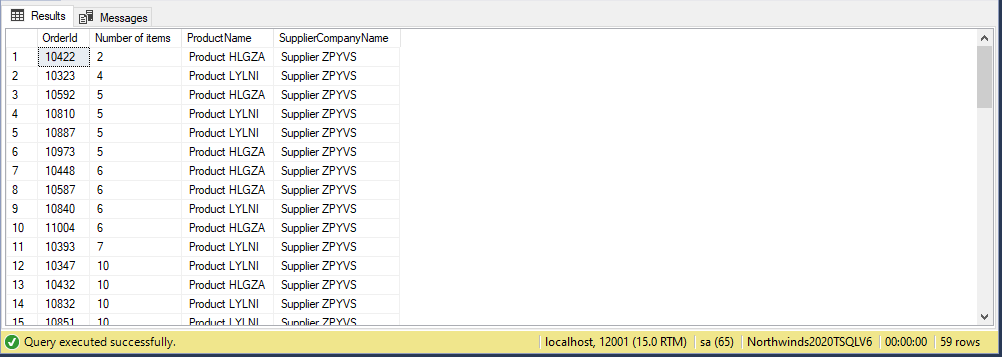
**,SOD.Quantity**

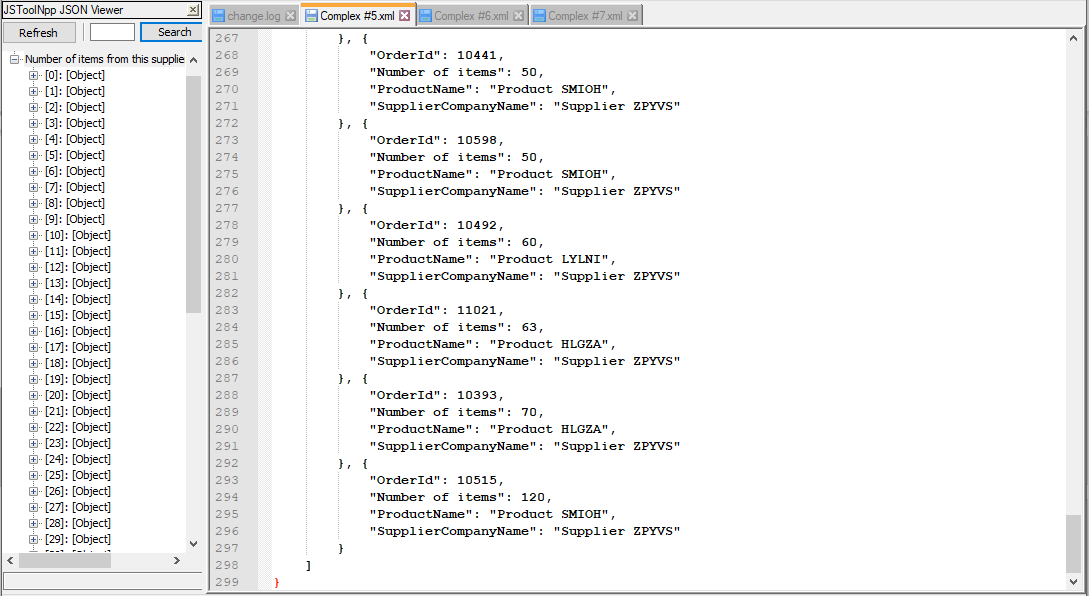
**ORDER BY SOD.Quantity ASC**

**,SO.OrderId**

**,PP.ProductName;**

**Output**





Function that will be used for the next 2 queries:

USE AdventureWorksDW2017

CREATE FUNCTION fnd\_city

(

@GeoKey int

)

RETURNS varchar(15)

BEGIN

RETURN

(

SELECT SalesTerritoryKey

FROM dbo.DimGeography

WHERE GeographyKey = @GeoKey

);

END;

Proposition #19 (Complex)

Display for the sales territory that has Geography Key #155: the city, first, last, and middle name of the employee in charge of that territory, the number of orders they handled, and the order date of those orders made after the start of 2013



Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| DimGeography | City |
| DimEmployee | CONCAT(LastName, FirstName, MiddleName) AS Name |
| FactsResellerSales | COUNT(SalesOrderNumber) AS # of Orders, OrderDate |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| DimGeography | City | ASC |

**Solution Query**

**USE AdventureWorksDW2017;**

**SELECT DG.City**

**,CONCAT (**

**DE.LastName**

**,' '**

**,DE.FirstName**

**,' '**

**,DE.MiddleName**

**) AS Name**

**,COUNT(FRS.SalesOrderNumber) AS '# of orders'**

**,FRS.OrderDate**

**FROM dbo.DimGeography AS DG**

**INNER JOIN dbo.DimEmployee AS DE ON DE.SalesTerritoryKey = DG.SalesTerritoryKey**

**INNER JOIN dbo.FactResellerSales AS FRS ON FRS.EmployeeKey = DE.EmployeeKey**

**WHERE FRS.SalesTerritoryKey = dbo.fnd\_city(155)**

**AND FRS.OrderDate > '20130101'**

**GROUP BY DG.City**

**,FRS.OrderDate**

**,CONCAT (**

**DE.LastName**

**,' '**

**,DE.FirstName**

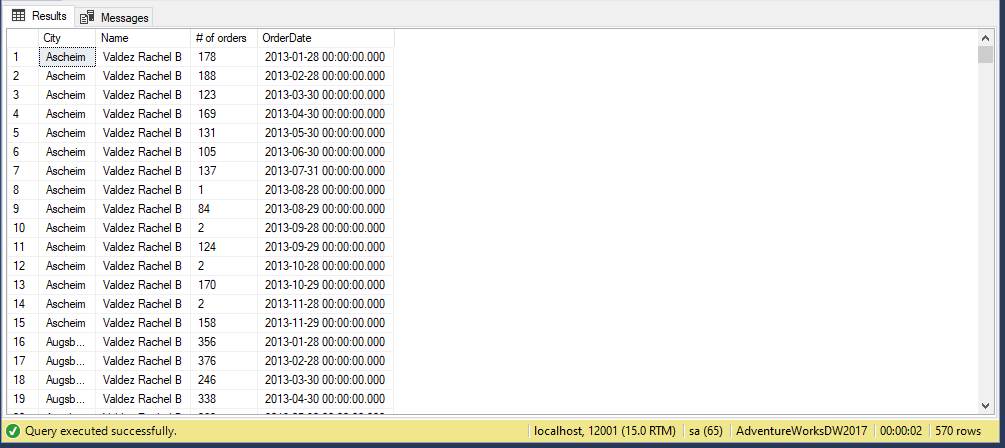
**,' '**

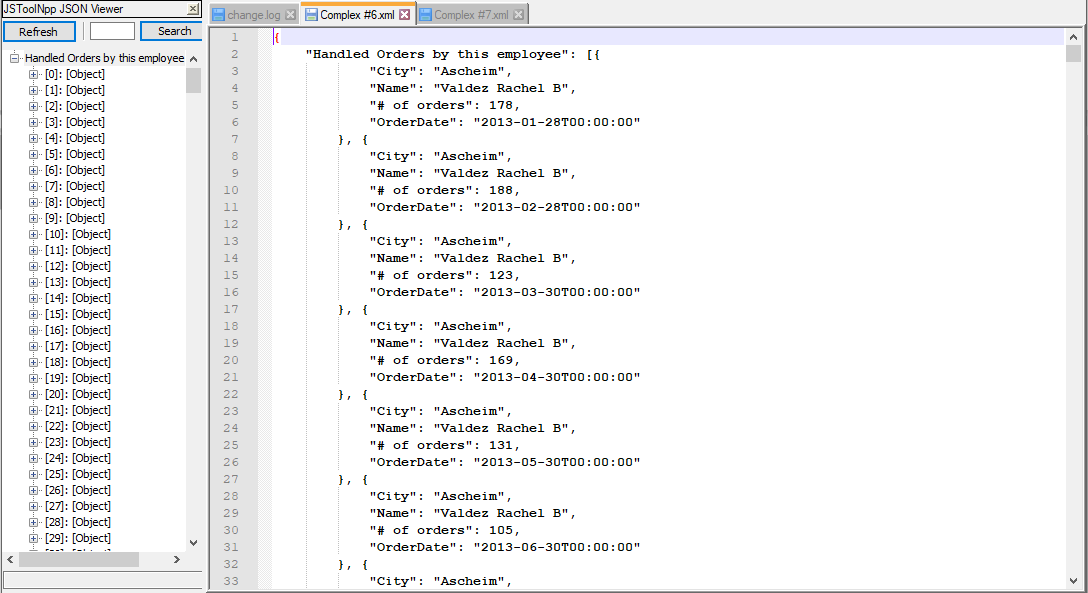
**,DE.MiddleName**

**)**

**ORDER BY DG.City;**

**Output**





Proposition #20 (Complex)

Display for each employee: their full name, their total sales quota, and the name of their sales territory country





Columns from Tables

|  |  |
| --- | --- |
| Table Name | Column Name |
| DimEmployee | CONCAT(LastName, FirstName, MiddleName) AS Name |
| FactsSalesQuota | Sum(SalesAmountQuota) AS Total Sales Quota |
| DimSalesTerritory | SalesTerritoryCountry |

Order By

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| DimSalesTerritory | SalesTerritoryCountry | ASC |
| FactsSalesQuota | Total Sales Quota | ASC |

**Solution Query**

**USE AdventureWorksDW2017;**

**SELECT CONCAT (**

**DE.LastName**

**,' '**

**,DE.FirstName**

**,' '**

**,DE.MiddleName**

**) AS Name**

**,SUM(FSQ.SalesAmountQuota) AS 'Total Sales Quota'**

**,DST.SalesTerritoryCountry**

**FROM dbo.DimEmployee AS DE**

**INNER JOIN dbo.FactSalesQuota AS FSQ ON FSQ.EmployeeKey = DE.EmployeeKey**

**INNER JOIN dbo.DimSalesTerritory AS DST ON DST.SalesTerritoryKey = DE.SalesTerritoryKey**

**WHERE DE.SalesTerritoryKey <= dbo.fnd\_city(155)**

**GROUP BY DST.SalesTerritoryCountry**

**,CONCAT (**

**DE.LastName**

**,' '**

**,DE.FirstName**

**,' '**

**,DE.MiddleName**

**)**

**ORDER BY DST.SalesTerritoryCountry**

**,[Total Sales Quota];**

**Output**

