Developing the soft skills needed in the business work environment such as teamwork, documentation and creating workflows.

Learning the structure of a newly acquired database systems by your company without any documentation.

Each Problem should follow the format in Problem 01.

Contents

[Proposition 1: What is the most recent order for each customer using Northwinds2020TSQLV6? 9](#_Toc67605789)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue 9](#_Toc67605790)

[Database 9](#_Toc67605791)

[Diagram of tables 9](#_Toc67605792)

[Columns from Standard view 10](#_Toc67605793)

[Project following columns from their respective tables in the select clause 10](#_Toc67605794)

[Order by 11](#_Toc67605795)

[Problem solving Query 11](#_Toc67605796)

[Sample Relational Output with total number of rows returned (830) 11](#_Toc67605797)

[Sample JSON Output with total number of rows returned (830) 11](#_Toc67605798)

[Proposition 01 Fixed: What is the most recent order for each customer using Northwinds2020TSQLV6? 12](#_Toc67605799)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 13](#_Toc67605800)

[Database 13](#_Toc67605801)

[Diagram(s) of tables 13](#_Toc67605802)

[Columns from Standard view 14](#_Toc67605803)

[Project following columns from their respective tables in the select clause 14](#_Toc67605804)

[Order by 15](#_Toc67605805)

[Problem solving Query 15](#_Toc67605806)

[Relational Output with total number of rows returned (830) 15](#_Toc67605807)

[Sample JSON Output with total number of rows returned (830) 15](#_Toc67605808)

[Proposition 02: Where are there customers but no employees using Northwinds2020TSQLV6? 16](#_Toc67605809)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 16](#_Toc67605810)

[Database 17](#_Toc67605811)

[Diagram of tables 17](#_Toc67605812)

[Columns from Standard view 17](#_Toc67605813)

[Project following columns from their respective tables in the select clause 18](#_Toc67605814)

[Problem solving Query 18](#_Toc67605815)

[Relational Output with total number of rows returned (66) 18](#_Toc67605816)

[Sample JSON Output with total number of rows returned (66) 19](#_Toc67605817)

[Proposition 03: What are the details for the items in the shopping cart using AdventureWorks2017? 21](#_Toc67605818)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 21](#_Toc67605819)

[Database 21](#_Toc67605820)

[Diagram(s) of tables 21](#_Toc67605821)

[Columns from Standard view 22](#_Toc67605822)

[Project following columns from their respective tables in the select clause 22](#_Toc67605823)

[Problem solving Query 23](#_Toc67605824)

[Relational Output with total number of rows returned (3) 23](#_Toc67605825)

[Sample JSON Output with total number of rows returned (3) 23](#_Toc67605826)

[Proposition 04: Where are there suppliers and employees using Northwinds2020TSQLV6? 24](#_Toc67605827)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 24](#_Toc67605828)

[Database 24](#_Toc67605829)

[Diagram(s) of tables 24](#_Toc67605830)

[Columns from Standard view 25](#_Toc67605831)

[Project following columns from their respective tables in the select clause 25](#_Toc67605832)

[Order by 25](#_Toc67605833)

[Problem solving Query 25](#_Toc67605834)

[Relational Output with total number of rows returned (1) 26](#_Toc67605835)

[Sample JSON Output with total number of rows returned (1) 27](#_Toc67605836)

[Proposition 05: Where have orders been shipped to that have no suppliers using Northwinds2020TSQLV6? 27](#_Toc67605837)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 27](#_Toc67605838)

[Database 27](#_Toc67605839)

[Diagram(s) of tables 27](#_Toc67605840)

[Columns from Standard view 28](#_Toc67605841)

[Project following columns from their respective tables in the select clause 28](#_Toc67605842)

[Problem solving Query 28](#_Toc67605843)

[Relational Output with total number of rows returned (4) 29](#_Toc67605844)

[Sample JSON Output with total number of rows returned (4) 29](#_Toc67605845)

[Proposition 06: How many employees have worked in each department, and how many have stopped working in that department using AdventureWorks2017? 30](#_Toc67605846)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 30](#_Toc67605847)

[Database 30](#_Toc67605848)

[Diagram(s) of tables 30](#_Toc67605849)

[Columns from Standard view 31](#_Toc67605850)

[Project following columns from their respective tables in the select clause 31](#_Toc67605851)

[Problem solving Query 32](#_Toc67605852)

[Relational Output with total number of rows returned (16) 32](#_Toc67605853)

[Sample JSON Output with total number of rows returned (16) 32](#_Toc67605854)

[Proposition 07: For each order, how many types of products were ordered and what was the quantity using AdventureWorks2017? 34](#_Toc67605855)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 34](#_Toc67605856)

[Database 34](#_Toc67605857)

[Diagram(s) of tables 34](#_Toc67605858)

[Columns from Standard view 35](#_Toc67605859)

[Project following columns from their respective tables in the select clause 35](#_Toc67605860)

[Problem solving Query 36](#_Toc67605861)

[Relational Output with total number of rows returned (4012) 36](#_Toc67605862)

[Sample JSON Output with total number of rows returned (4012) 36](#_Toc67605863)

[Proposition 08: What are the total sales the previous year in each country by their currency using AdventureWorks2017? 37](#_Toc67605864)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 37](#_Toc67605865)

[Database 37](#_Toc67605866)

[Diagram(s) of tables 37](#_Toc67605867)

[Columns from Standard view 38](#_Toc67605868)

[Project following columns from their respective tables in the select clause 38](#_Toc67605869)

[Problem solving Query 39](#_Toc67605870)

[Relational Output with total number of rows returned (8) 39](#_Toc67605871)

[Sample JSON Output with total number of rows returned (8) 39](#_Toc67605872)

[Proposition 08 Fixed: What are the total sales the previous year in each country by their currency using AdventureWorks2017? 40](#_Toc67605873)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 40](#_Toc67605874)

[Database 40](#_Toc67605875)

[Diagram(s) of tables 40](#_Toc67605876)

[Columns from Standard view 41](#_Toc67605877)

[Project following columns from their respective tables in the select clause 41](#_Toc67605878)

[Problem solving Query 42](#_Toc67605879)

[Relational Output with total number of rows returned (6) 42](#_Toc67605880)

[Sample JSON Output with total number of rows returned (6) 42](#_Toc67605881)

[Proposition 09: How many of products with a list price between 5 and 10 dollars were bought using AdventureWorks2017? 43](#_Toc67605882)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 43](#_Toc67605883)

[Database 43](#_Toc67605884)

[Diagram(s) of tables 43](#_Toc67605885)

[Columns from Standard view 45](#_Toc67605886)

[Project following columns from their respective tables in the select clause 45](#_Toc67605887)

[Problem solving Query 46](#_Toc67605888)

[Relational Output with total number of rows returned (8) 46](#_Toc67605889)

[Sample JSON Output with total number of rows returned (8) 46](#_Toc67605890)

[Proposition 10: For each item ordered, what was the quantity and list price cost difference using AdventureWorks2017? 48](#_Toc67605891)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 48](#_Toc67605892)

[Database 48](#_Toc67605893)

[Diagram(s) of tables 48](#_Toc67605894)

[Columns from Standard view 49](#_Toc67605895)

[Project following columns from their respective tables in the select clause 49](#_Toc67605896)

[Order by 49](#_Toc67605897)

[Problem solving Query 50](#_Toc67605898)

[Relational Output with total number of rows returned (266) 50](#_Toc67605899)

[Sample JSON Output with total number of rows returned (266) 50](#_Toc67605900)

[Proposition 11: Which customers had over 10 orders in 2016 using Northwinds2020TSQLV6? 52](#_Toc67605901)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 52](#_Toc67605902)

[Database 52](#_Toc67605903)

[Diagram(s) of tables 52](#_Toc67605904)

[Columns from Standard view 53](#_Toc67605905)

[Project following columns from their respective tables in the select clause 53](#_Toc67605906)

[Problem solving Query 54](#_Toc67605907)

[Relational Output with total number of rows returned (1) 54](#_Toc67605908)

[Sample JSON Output with total number of rows returned (1) 54](#_Toc67605909)

[Proposition 12: Which orders had more than 3 items in 2015 using Northwinds2020TSQLV6? 54](#_Toc67605910)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 54](#_Toc67605911)

[Database 55](#_Toc67605912)

[Diagram(s) of tables 55](#_Toc67605913)

[Columns from Standard view 56](#_Toc67605914)

[Project following columns from their respective tables in the select clause 56](#_Toc67605915)

[Order by 57](#_Toc67605916)

[Problem solving Query 57](#_Toc67605917)

[ORDER BY o.OrderId; 57](#_Toc67605918)

[Relational Output with total number of rows returned (80) 57](#_Toc67605919)

[Sample JSON Output with total number of rows returned (80) 57](#_Toc67605920)

[Proposition 13: How many orders did employees make in QI 2015 using Northwinds2020TSQLV6? 59](#_Toc67605921)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 59](#_Toc67605922)

[Database 59](#_Toc67605923)

[Diagram(s) of tables 59](#_Toc67605924)

[Columns from Standard view 60](#_Toc67605925)

[Project following columns from their respective tables in the select clause 60](#_Toc67605926)

[Problem solving Query 61](#_Toc67605927)

[Relational Output with total number of rows returned (9) 61](#_Toc67605928)

[Sample JSON Output with total number of rows returned (9) 61](#_Toc67605929)

[Proposition 14: What are the hexadecimal values of Customer Ids, and how many orders did each customer place, and how many products for those orders using Northwinds2020TSQLV6? 62](#_Toc67605930)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 62](#_Toc67605931)

[Database 62](#_Toc67605932)

[Diagram(s) of tables 62](#_Toc67605933)

[Columns from Standard view 64](#_Toc67605934)

[Project following columns from their respective tables in the select clause 64](#_Toc67605935)

[Order by 64](#_Toc67605936)

[Problem solving Query 65](#_Toc67605937)

[Relational Output with total number of rows returned (91) 66](#_Toc67605938)

[Sample JSON Output with total number of rows returned (91) 66](#_Toc67605939)

[Proposition 15: What is the name of each employee, and how many orders did they sell and with how many items using Northwinds2020TSQLV6? 69](#_Toc67605940)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 69](#_Toc67605941)

[Database 69](#_Toc67605942)

[Diagram(s) of tables 69](#_Toc67605943)

[Columns from Standard view 70](#_Toc67605944)

[Project following columns from their respective tables in the select clause 70](#_Toc67605945)

[Problem solving Query 70](#_Toc67605946)

[Relational Output with total number of rows returned (9) 71](#_Toc67605947)

[Sample JSON Output with total number of rows returned (9) 71](#_Toc67605948)

[Proposition 16: What are the octal values of geography keys, and how many employees are assigned to their area and how many customers are located there using AdventureWorksDW2017? 73](#_Toc67605949)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 73](#_Toc67605950)

[Database 74](#_Toc67605951)

[Diagram(s) of tables 74](#_Toc67605952)

[Columns from Standard view 75](#_Toc67605953)

[Project following columns from their respective tables in the select clause 75](#_Toc67605954)

[Order by 76](#_Toc67605955)

[Problem solving Query 76](#_Toc67605956)

[Relational Output with total number of rows returned (655) 76](#_Toc67605957)

[Sample JSON Output with total number of rows returned (655) 77](#_Toc67605958)

[Proposition 17: What are the binary values of the employee ids and their manager’s, and how many orders has each employee made using Northwinds2020TSQLV6? 78](#_Toc67605959)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 78](#_Toc67605960)

[Database 79](#_Toc67605961)

[Diagram(s) of tables 79](#_Toc67605962)

[Columns from Standard view 79](#_Toc67605963)

[Project following columns from their respective tables in the select clause 79](#_Toc67605964)

[Problem solving Query 80](#_Toc67605965)

[Relational Output with total number of rows returned (9) 81](#_Toc67605966)

[Sample JSON Output with total number of rows returned (9) 81](#_Toc67605967)

[Proposition 17 Fixed: What are the binary values of the employee ids and their manager’s, and how many orders has each employee made using Northwinds2020TSQLV6? 83](#_Toc67605968)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 83](#_Toc67605969)

[Database 83](#_Toc67605970)

[Diagram(s) of tables 83](#_Toc67605971)

[Columns from Standard view 84](#_Toc67605972)

[Project following columns from their respective tables in the select clause 84](#_Toc67605973)

[Problem solving Query 84](#_Toc67605974)

[Relational Output with total number of rows returned (9) 85](#_Toc67605975)

[Sample JSON Output with total number of rows returned (9) 85](#_Toc67605976)

[Proposition 18: How long ago was the newest order for each customer, and which employee placed the order with them using Northwinds2020TSQLV6? 88](#_Toc67605977)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 88](#_Toc67605978)

[Database 88](#_Toc67605979)

[Diagram(s) of tables 88](#_Toc67605980)

[Columns from Standard view 89](#_Toc67605981)

[Project following columns from their respective tables in the select clause 89](#_Toc67605982)

[Order by 89](#_Toc67605983)

[Problem solving Query 90](#_Toc67605984)

[Relational Output with total number of rows returned (90) 91](#_Toc67605985)

[Sample JSON Output with total number of rows returned (90) 91](#_Toc67605986)

[Proposition 19: What quart was each order place, and how many items were in the order, what was its cost, and who supplied the items using Northwinds2020TSQLV6? 92](#_Toc67605987)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 92](#_Toc67605988)

[Database 93](#_Toc67605989)

[Diagram(s) of tables 93](#_Toc67605990)

[Columns from Standard view 93](#_Toc67605991)

[Project following columns from their respective tables in the select clause 93](#_Toc67605992)

[Order by 94](#_Toc67605993)

[Problem solving Query 94](#_Toc67605994)

[Relational Output with total number of rows returned (830) 95](#_Toc67605995)

[Sample JSON Output with total number of rows returned (830) 95](#_Toc67605996)

[Proposition 20: What are the individual orders each customer made and how many items did they get with each order, and are they located where the employee who handled the order is using Northwinds2020TSQLV6? 98](#_Toc67605997)

[Detailed explanation of the problem that will help the developer to write the query to resolve the issue. 98](#_Toc67605998)

[Database 98](#_Toc67605999)

[Diagram(s) of tables 98](#_Toc67606000)

[Columns from Standard view 100](#_Toc67606001)

[Project following columns from their respective tables in the select clause 100](#_Toc67606002)

[Order by 101](#_Toc67606003)

[Problem solving Query 101](#_Toc67606004)

[Relational Output with total number of rows returned (832) 102](#_Toc67606005)

[Sample JSON Output with total number of rows returned (832) 102](#_Toc67606006)

# Proposition 1: What is the most recent order for each customer using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue

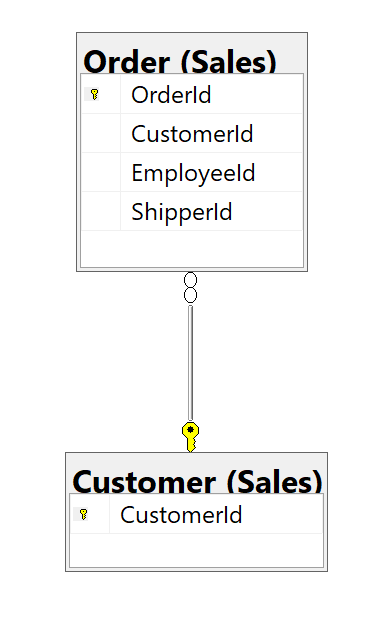
Find the orderdates for each customerid in the orders table by joining the customer and order tables.

## Database

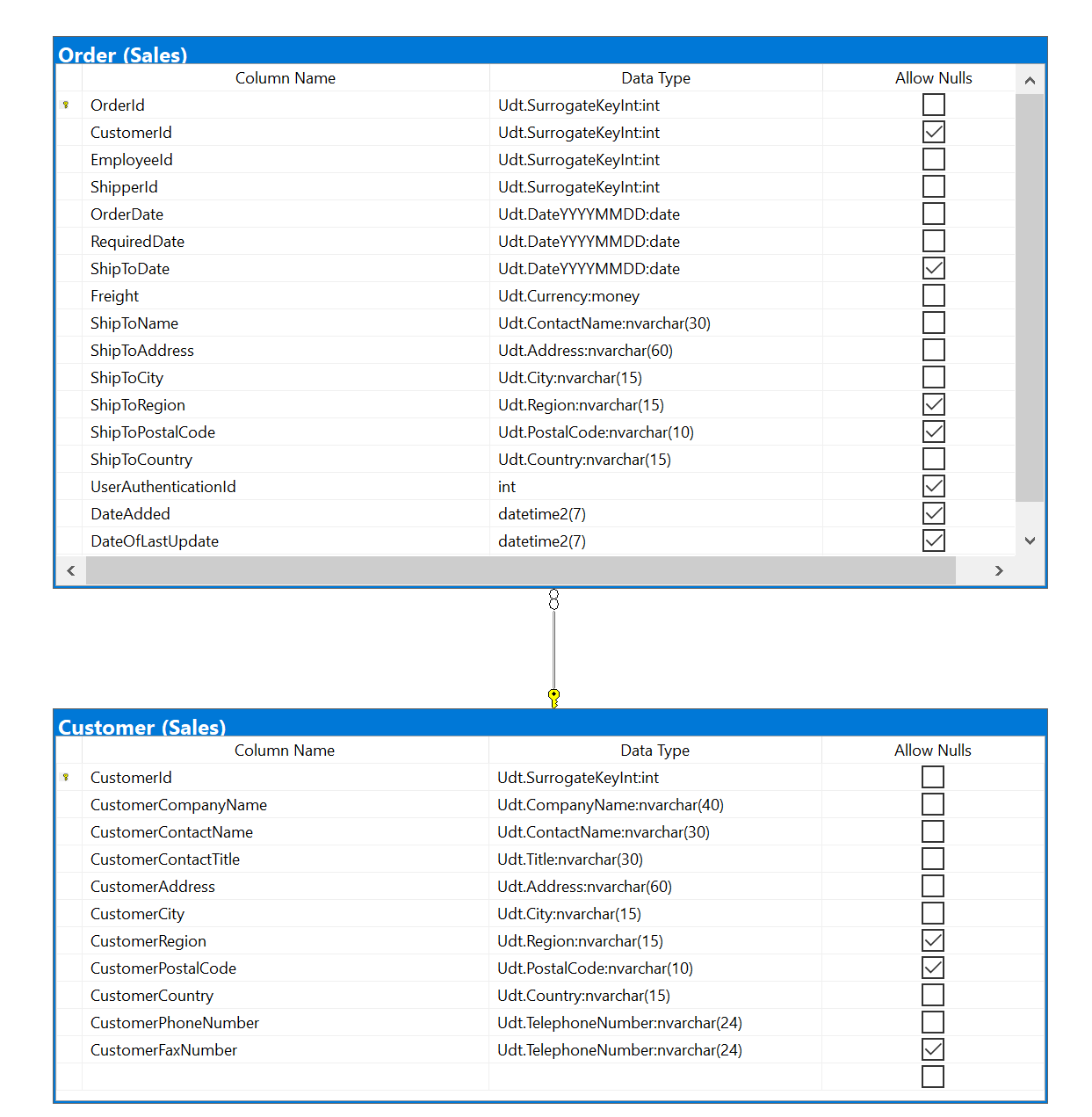
Northwinds2020TSQLV6

## Diagram of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Customers | CustomerId |
| Orders | OrderDate |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Customer | CustomerId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

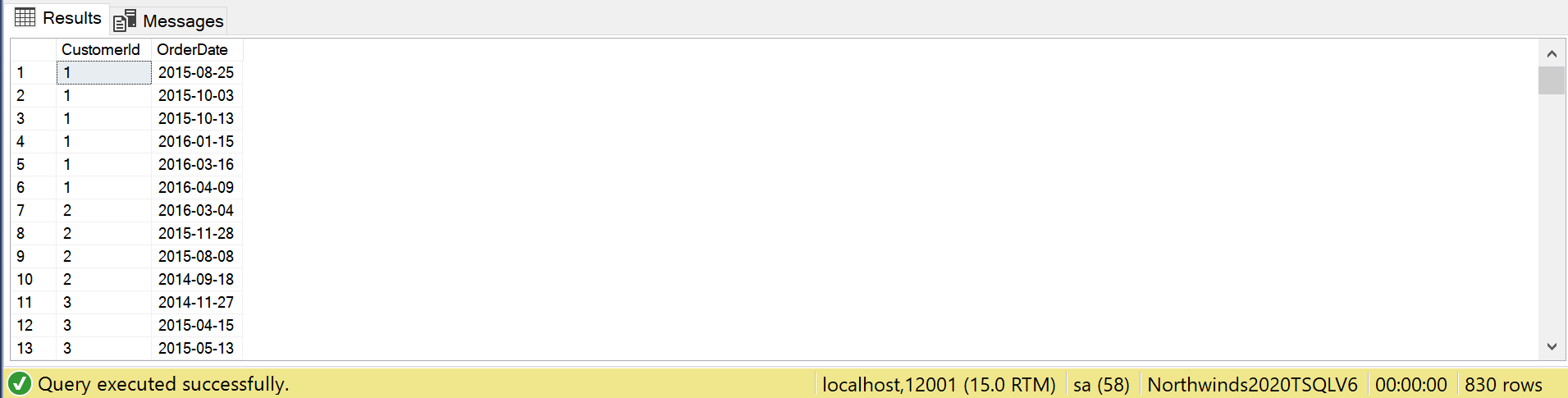
,o.OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

ORDER BY c.CustomerId;

## Sample Relational Output with total number of rows returned (830)



## Sample JSON Output with total number of rows returned (830)

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

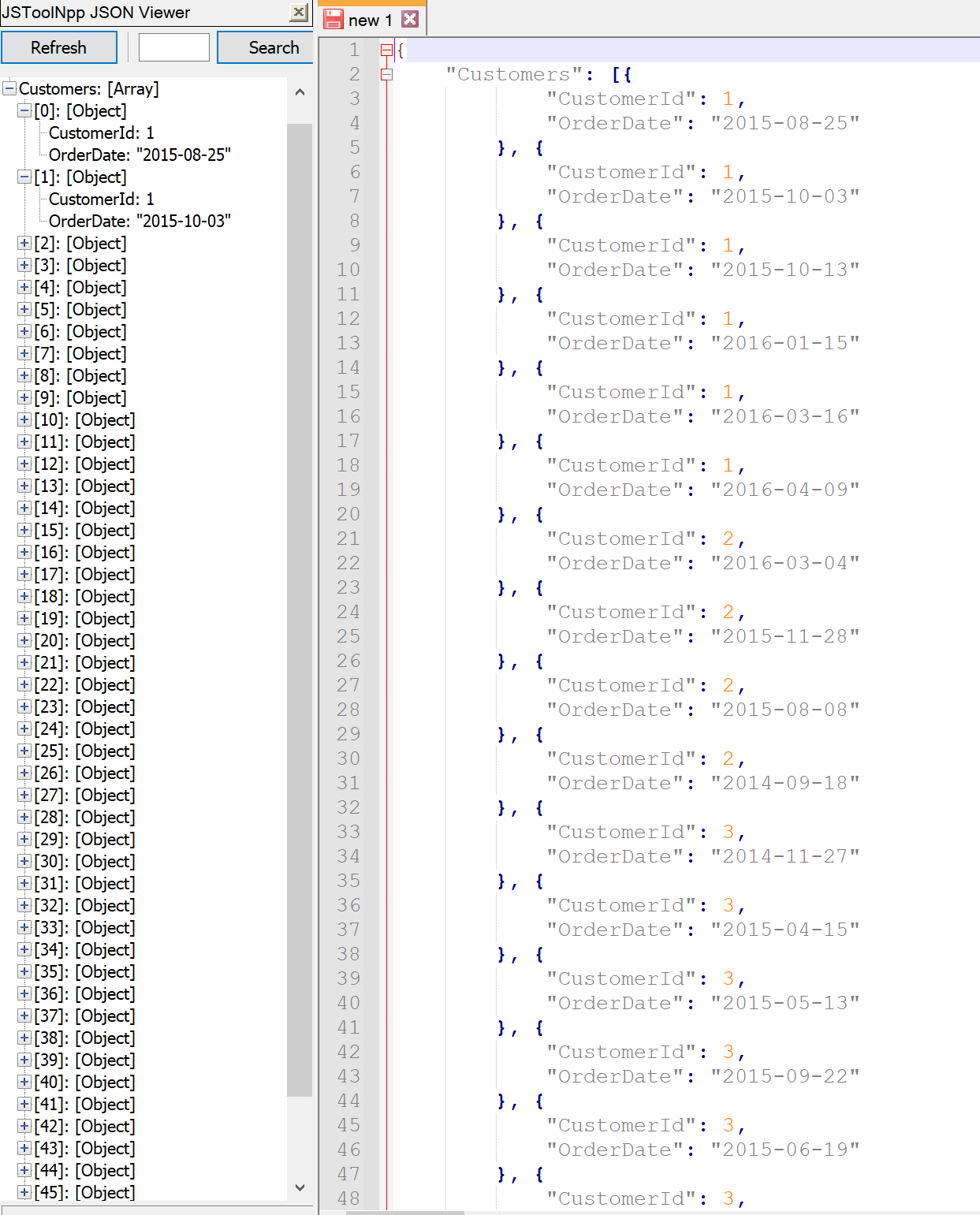
,o.OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

ORDER BY c.CustomerId

FOR JSON PATH, ROOT('Customers'), INCLUDE\_NULL\_VALUES;



# Proposition 01 Fixed: What is the most recent order for each customer using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

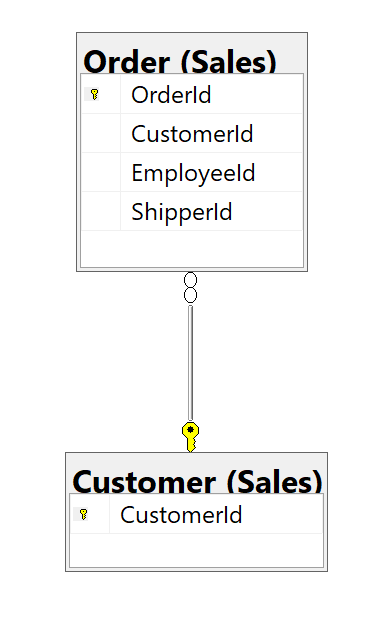
In the original query, there is a join between the Customer and Order table, but it is completely unnecessary. In the fixed version, the CustomerCompanyName is added since it is only in the Customer table, therefore justifying the join.

## Database

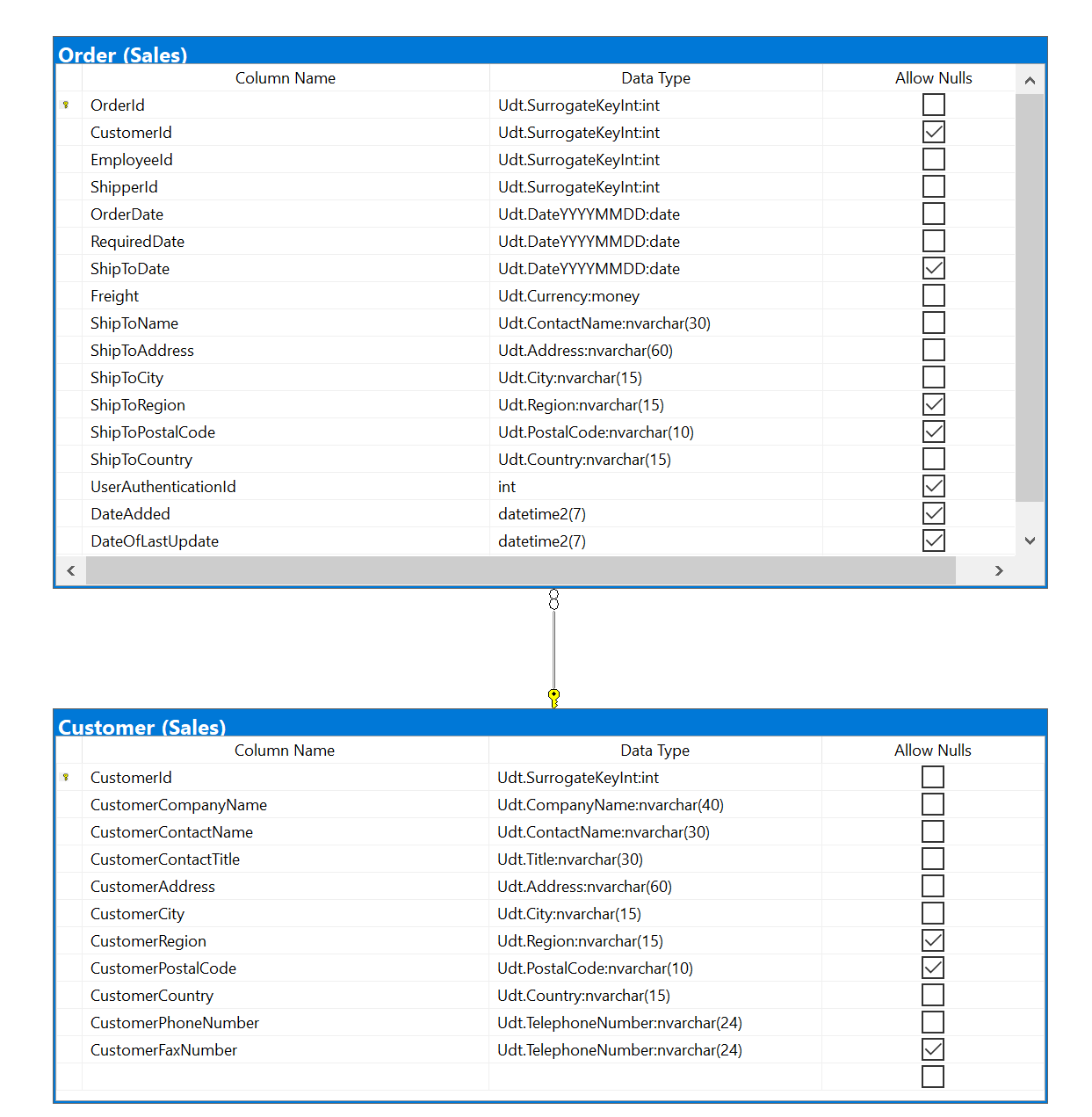
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Customer | CustomerId  CustomerCompanyName |
| Order | OrderDate |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Customer | CustomerId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

,c.CustomerCompanyName

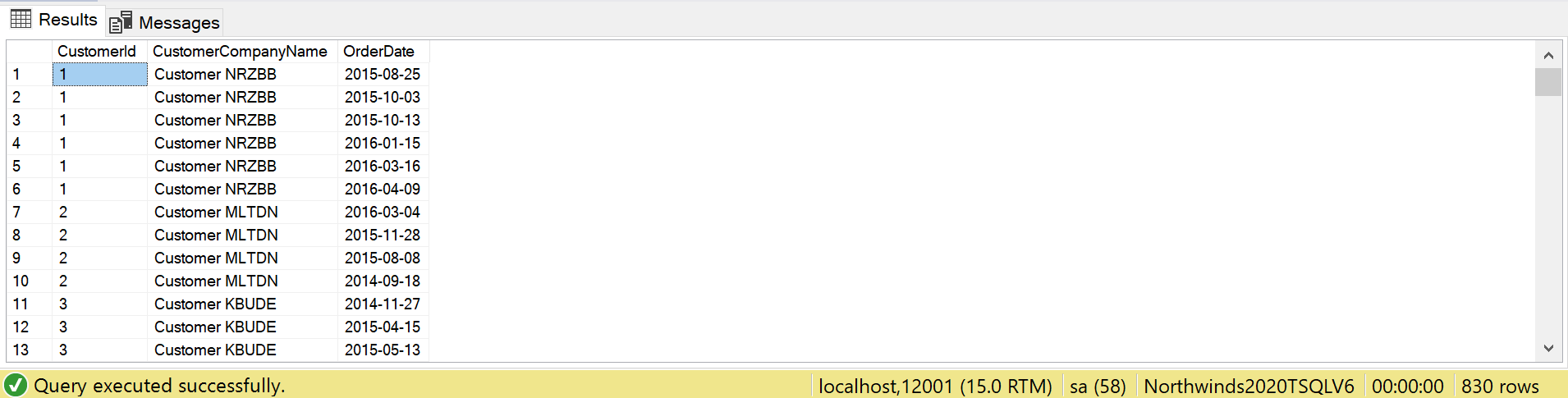
,o.OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

ORDER BY c.CustomerId;

## Relational Output with total number of rows returned (830)



## Sample JSON Output with total number of rows returned (830)

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

,c.CustomerCompanyName

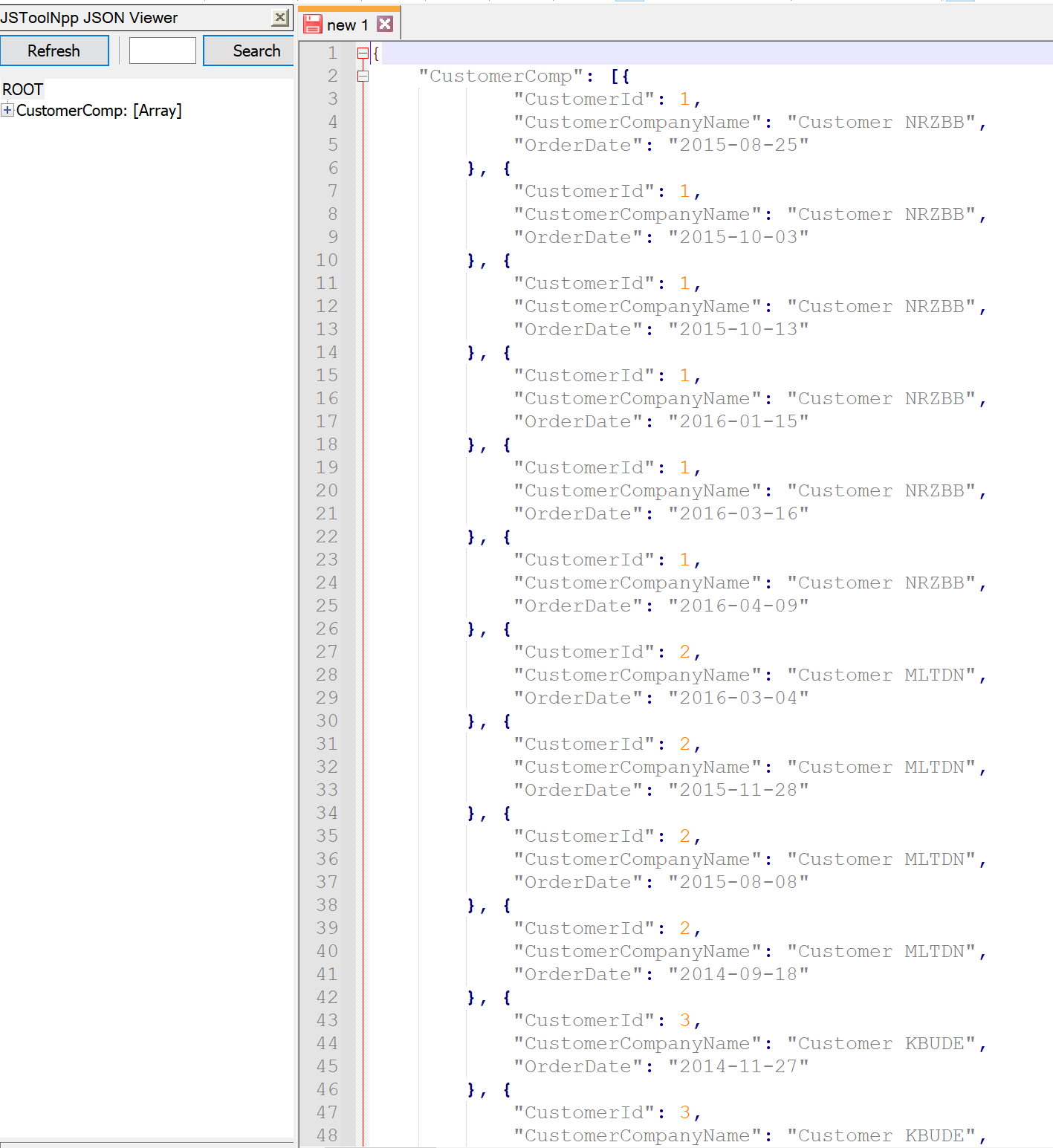
,o.OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

ORDER BY c.CustomerId

FOR JSON PATH, ROOT('CustomerComp'), INCLUDE\_NULL\_VALUES;



# Proposition 02: Where are there customers but no employees using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

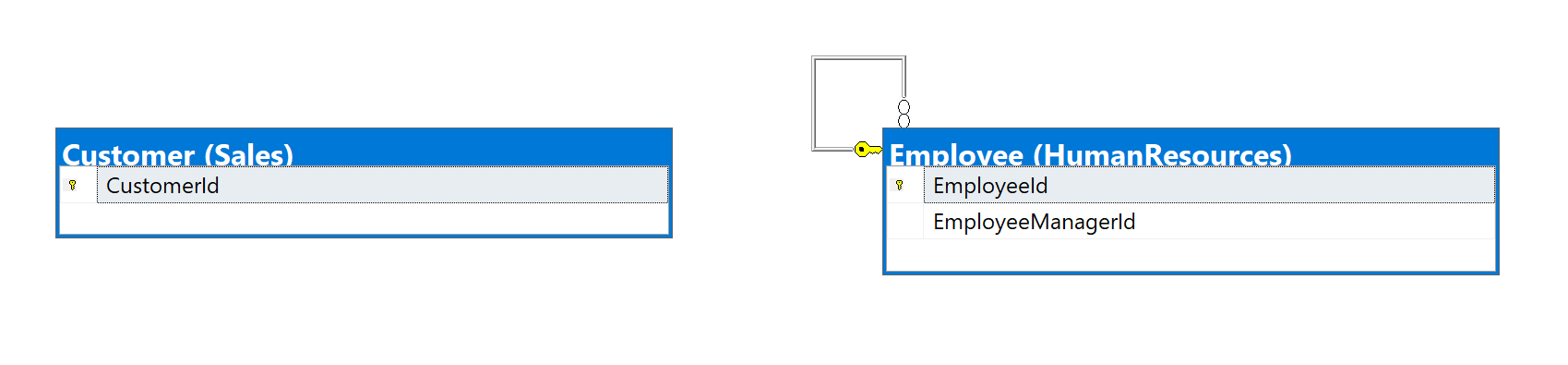
Display the country, region, and city where there are customers located, but no employees using the except clause.

## Database

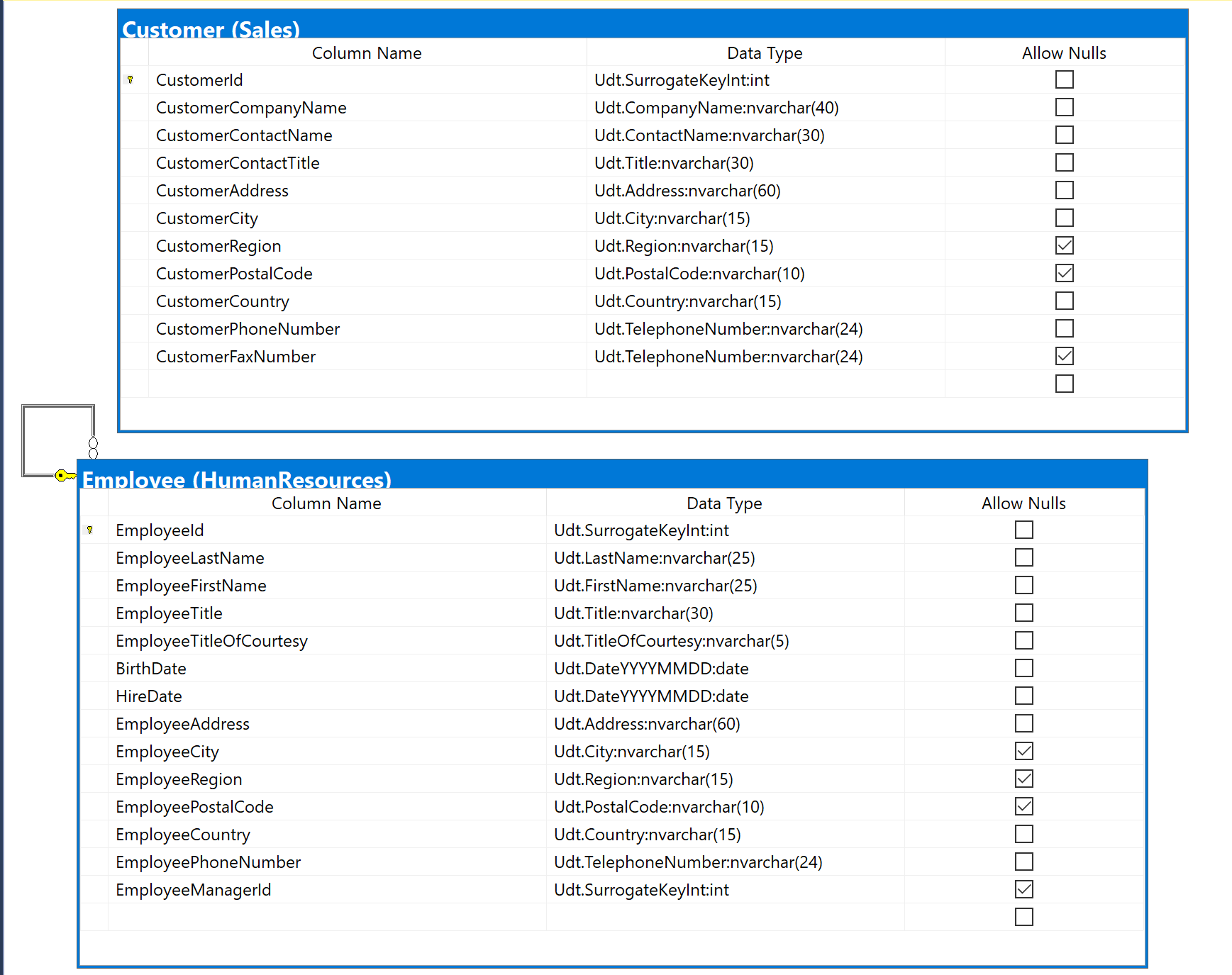
Northwinds2020TSQLV6

## Diagram of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Customer | CustomerCountry  CustomerRegion  CustomerCity |
| Employee | EmployeeCountry  EmployeeRegion  EmployeeCity |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerCountry AS Country

,c.CustomerRegion AS Region

,c.CustomerCity AS City

FROM Sales.Customer AS c

EXCEPT

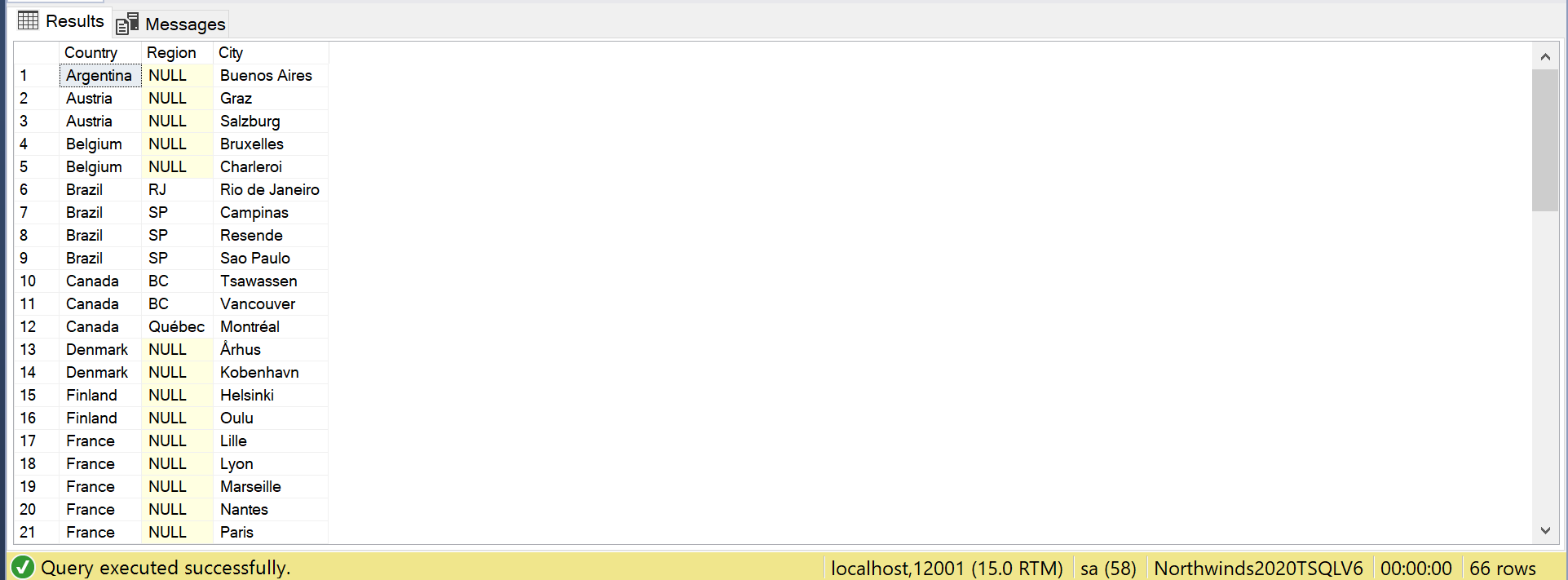
SELECT e.EmployeeCountry

,e.EmployeeRegion

,e.EmployeeCity

FROM HumanResources.Employee AS e;

## Relational Output with total number of rows returned (66)



## Sample JSON Output with total number of rows returned (66)

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerCountry AS Country

,c.CustomerRegion AS Region

,c.CustomerCity AS City

FROM Sales.Customer AS c

EXCEPT

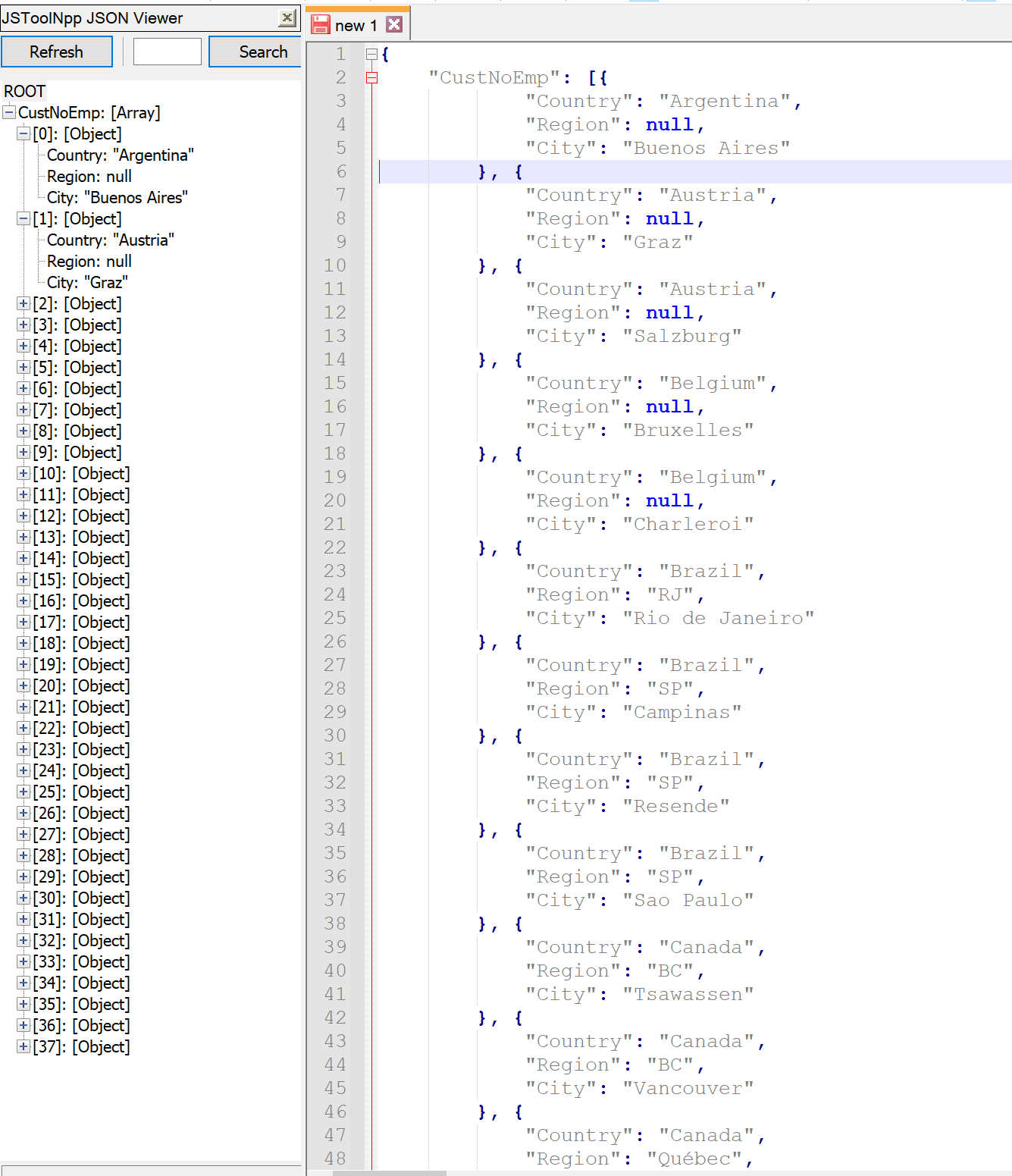
SELECT e.EmployeeCountry

,e.EmployeeRegion

,e.EmployeeCity

FROM HumanResources.Employee AS e

FOR JSON PATH, ROOT('CustNoEmp'), INCLUDE\_NULL\_VALUES;



# Proposition 03: What are the details for the items in the shopping cart using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

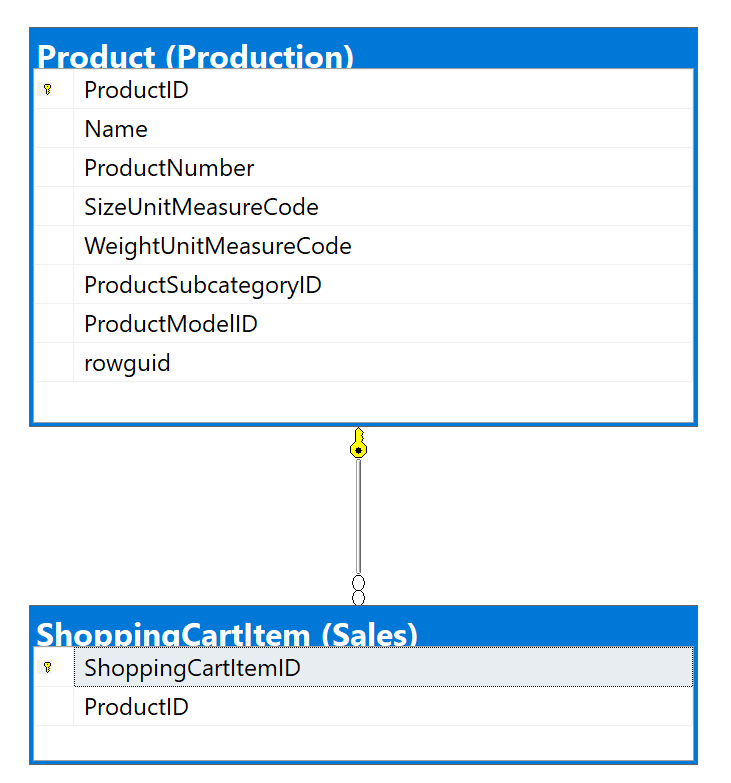
Show the productid, listprice, and name for all the products in shopping cart item, joining it with the product table.

## Database

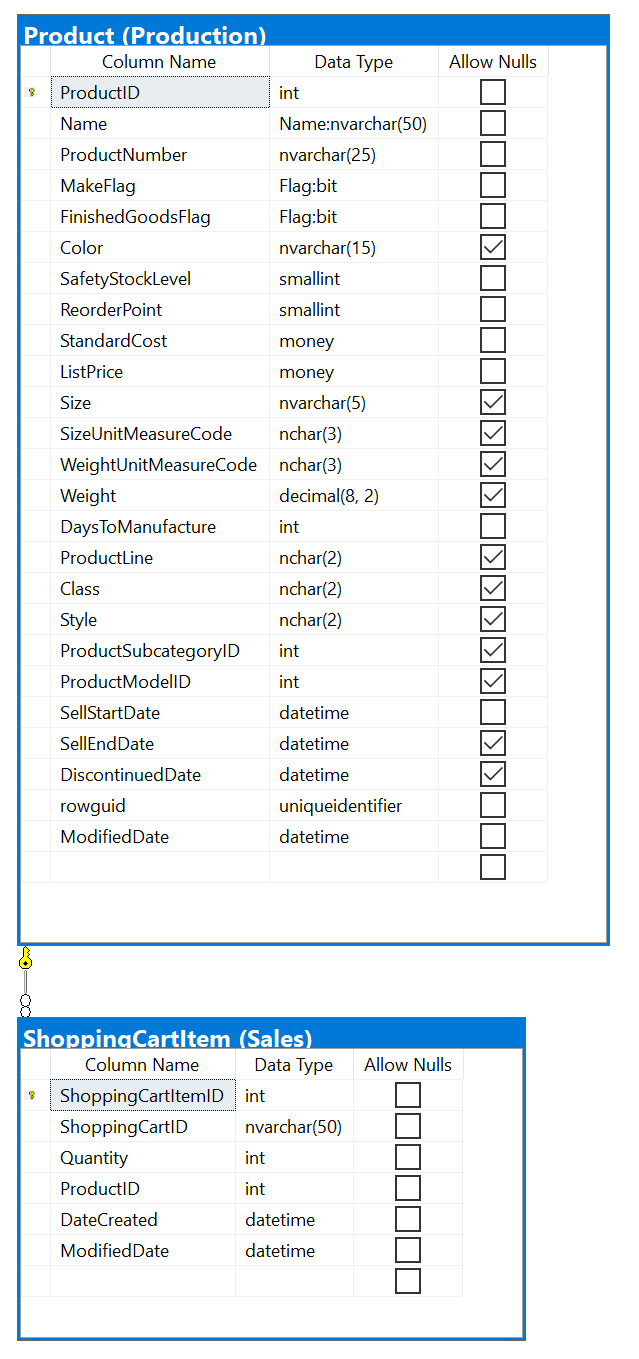
AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| ShoppingCartItem | ShoppingCartItemId  ProductId |
| Product | Name  ListPrice |

## Problem solving Query

**USE AdventureWorks2017;**

**GO**

**SELECT s.ShoppingCartItemId**

**,s.ProductId**

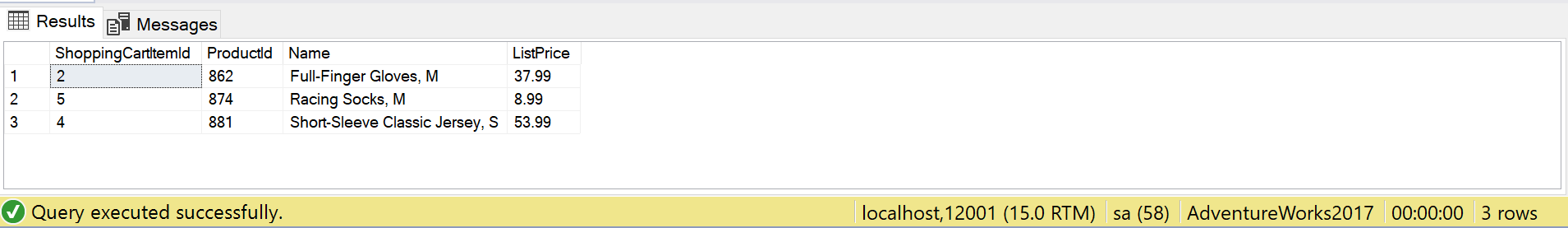
**,p.[Name]**

**,p.ListPrice**

**FROM Sales.ShoppingCartItem AS s**

**INNER JOIN Production.Product AS p ON s.ProductId = p.ProductId;**

## Relational Output with total number of rows returned (3)



## Sample JSON Output with total number of rows returned (3)

USE AdventureWorks2017;

GO

SELECT s.ShoppingCartItemId

,s.ProductId

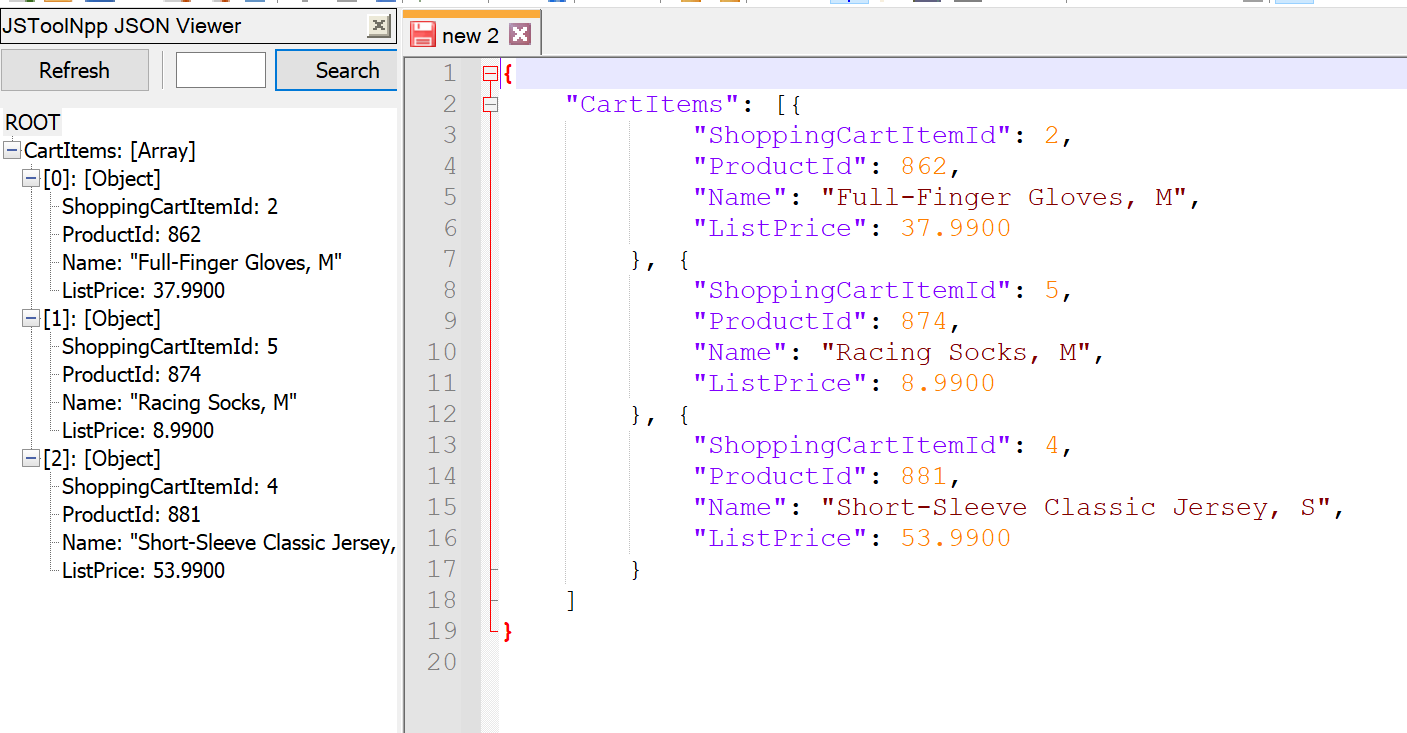
,p.[Name]

,p.ListPrice

FROM Sales.ShoppingCartItem AS s

INNER JOIN Production.Product AS p ON s.ProductId = p.ProductId

FOR JSON PATH, ROOT('CartItems'), INCLUDE\_NULL\_VALUES;



# Proposition 04: Where are there suppliers and employees using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

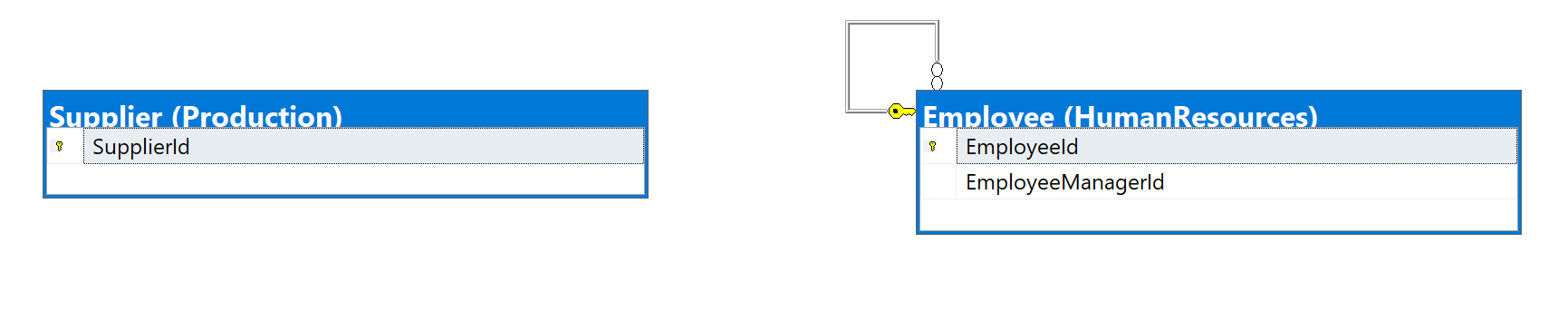
Show the country, region, and city for locations where there are both suppliers and employees by intersecting the two tables.

## Database

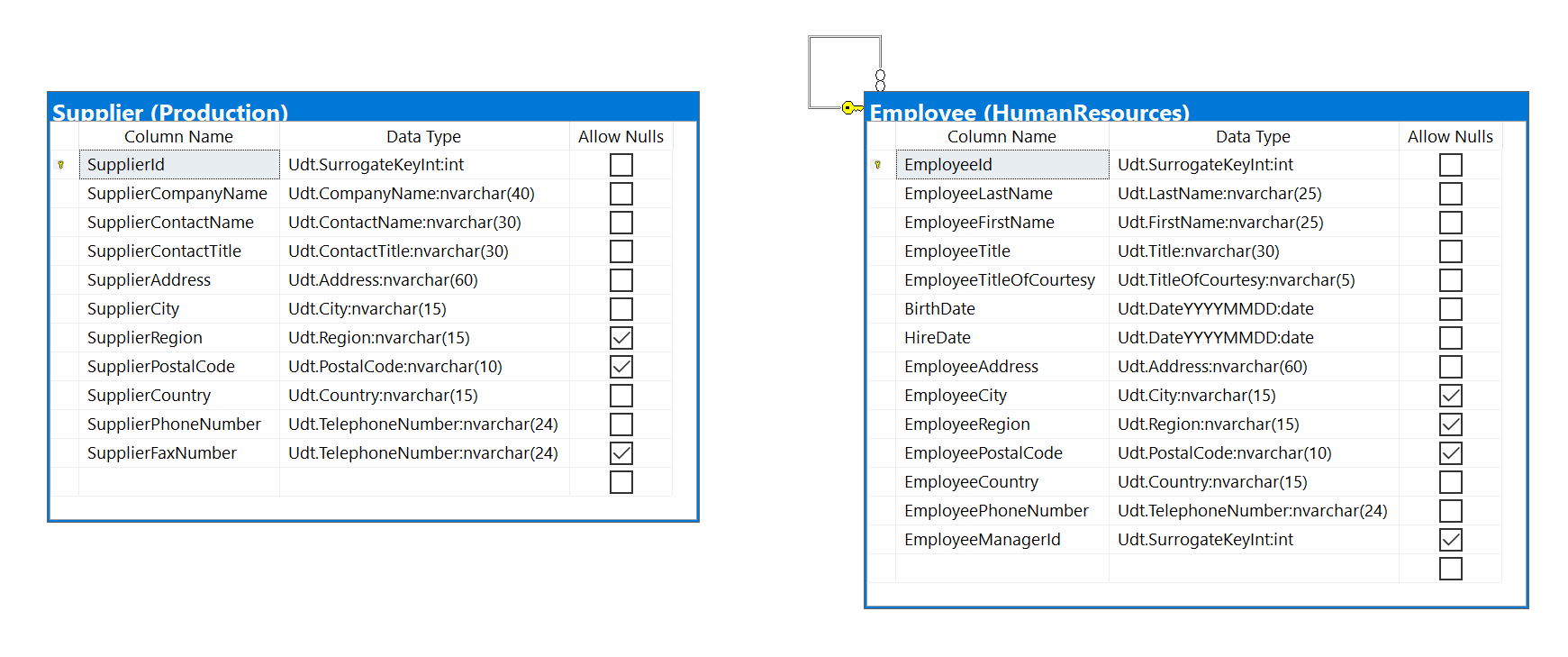
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Supplier | SupplierCountry  SupplierRegion  SupplierCity |
| Employee | EmployeeCountry  EmployeeRegion  EmployeeCity |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Supplier | SupplierCountry  SupplierRegion  SupplierCity | DESC  DESC  DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT s.SupplierCountry AS Country

,s.SupplierRegion AS Region

,s.SupplierCity AS City

FROM Production.Supplier AS s

INTERSECT

SELECT e.EmployeeCountry

,e.EmployeeRegion

,e.EmployeeCity

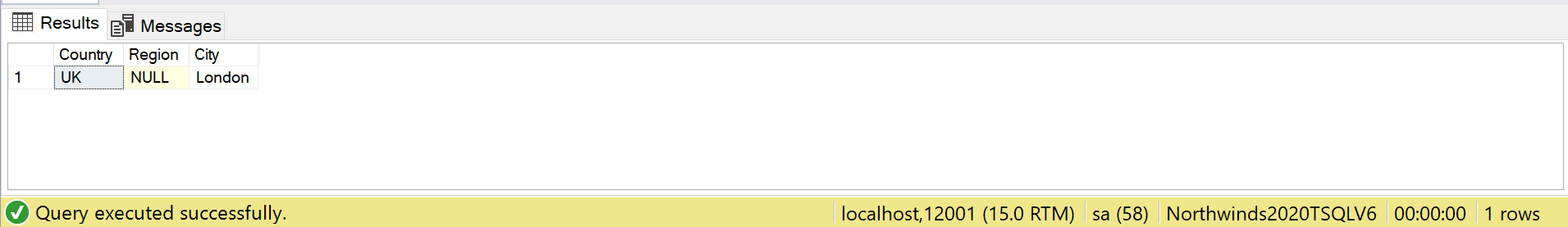
FROM HumanResources.Employee AS e

ORDER BY Country

,Region

,City;

## Relational Output with total number of rows returned (1)



## Sample JSON Output with total number of rows returned (1)

USE Northwinds2020TSQLV6;

GO

SELECT s.SupplierCountry AS Country

,s.SupplierRegion AS Region

,s.SupplierCity AS City

FROM Production.Supplier AS s

INTERSECT

SELECT e.EmployeeCountry

,e.EmployeeRegion

,e.EmployeeCity

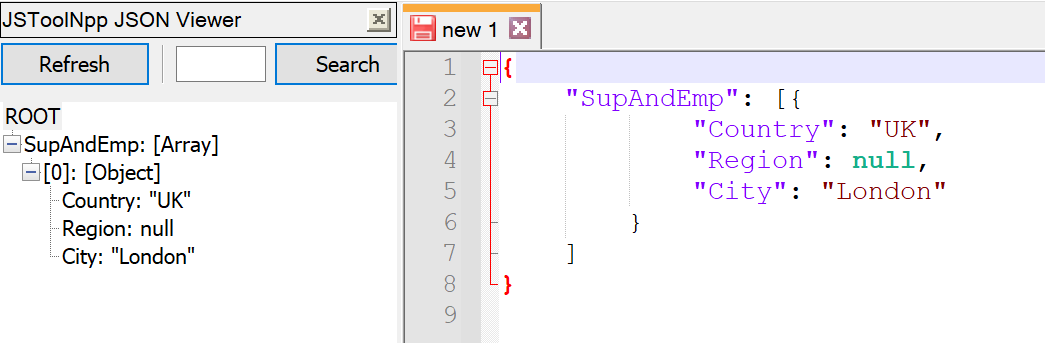
FROM HumanResources.Employee AS e

ORDER BY Country

,Region

,City

FOR JSON PATH, ROOT('SupAndEmp'), INCLUDE\_NULL\_VALUES;



# Proposition 05: Where have orders been shipped to that have no suppliers using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

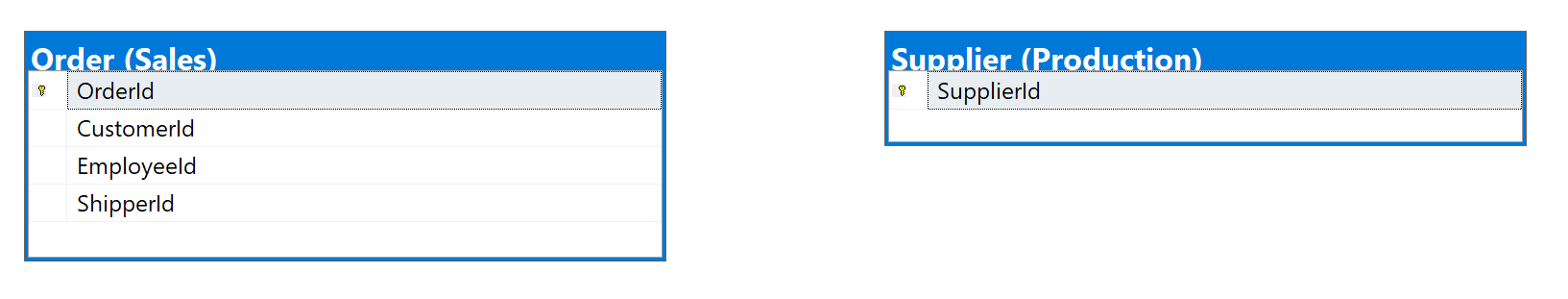
Find where the country, region, and city of the order and supplier tables intersect.

## Database

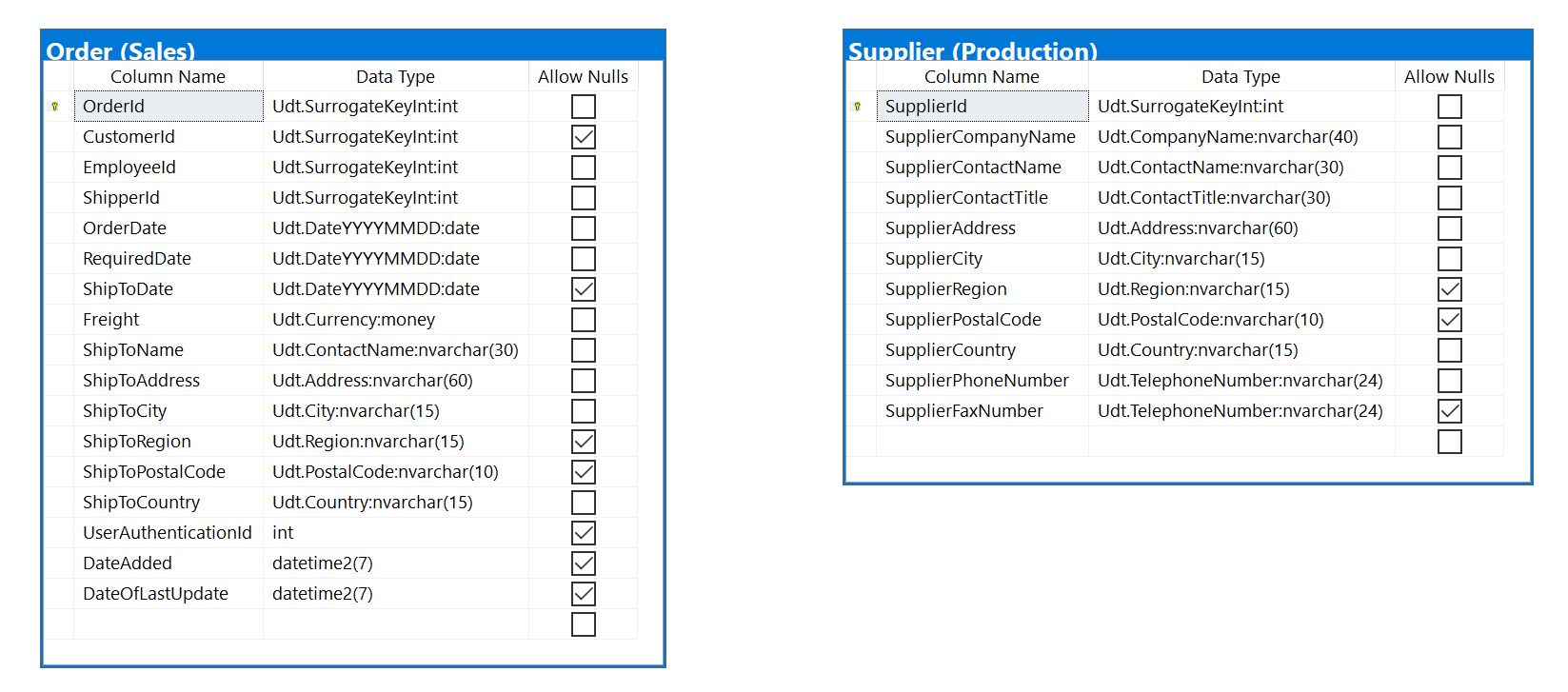
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Supplier | SupplierCountry  SupplierRegion  SupplierCity |
| Order | ShipToCountry  ShipToRegion  ShipToCity |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT o.ShipToCountry AS Country

,o.ShipToRegion AS Region

,o.ShipToCity AS City

FROM Sales.[Order] AS o

INTERSECT

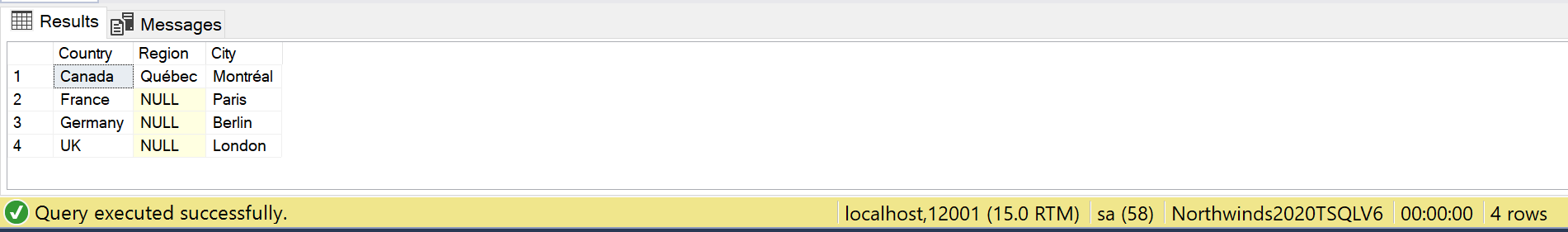
SELECT s.SupplierCountry

,s.SupplierRegion

,s.SupplierCity

FROM Production.Supplier AS s;

## Relational Output with total number of rows returned (4)



## Sample JSON Output with total number of rows returned (4)

USE Northwinds2020TSQLV6;

GO

SELECT o.ShipToCountry AS Country

,o.ShipToRegion AS Region

,o.ShipToCity AS City

FROM Sales.[Order] AS o

INTERSECT

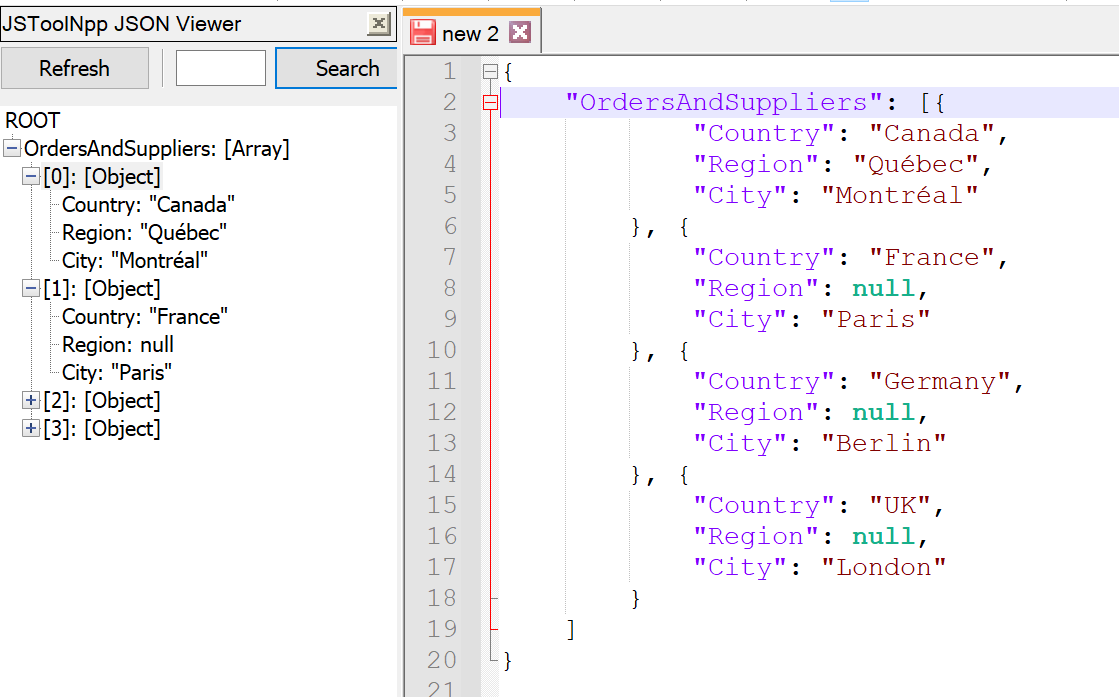
SELECT s.SupplierCountry

,s.SupplierRegion

,s.SupplierCity

FROM Production.Supplier AS s

FOR JSON PATH, ROOT('OrdersAndSuppliers'), INCLUDE\_NULL\_VALUES;



# Proposition 06: How many employees have worked in each department, and how many have stopped working in that department using AdventureWorks2017?

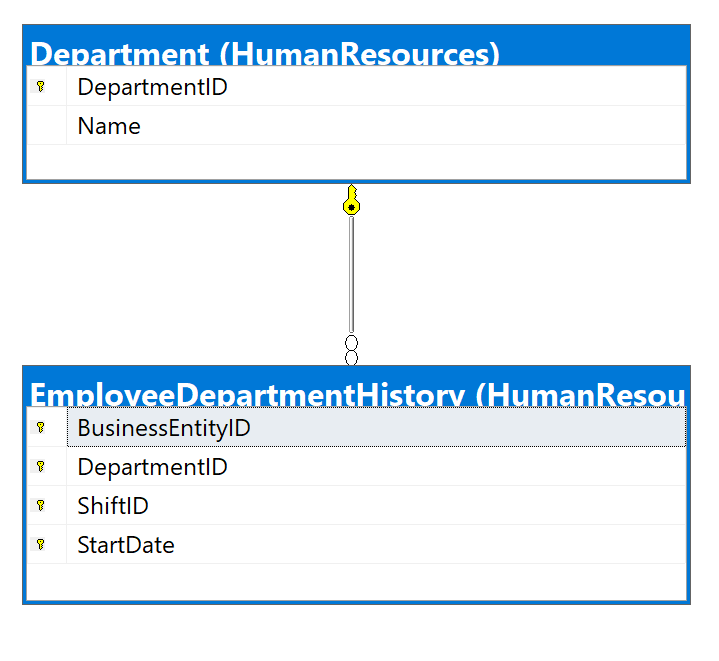
## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

For each department, the numer of BussinessEntityIds and EndDates need to be counted.

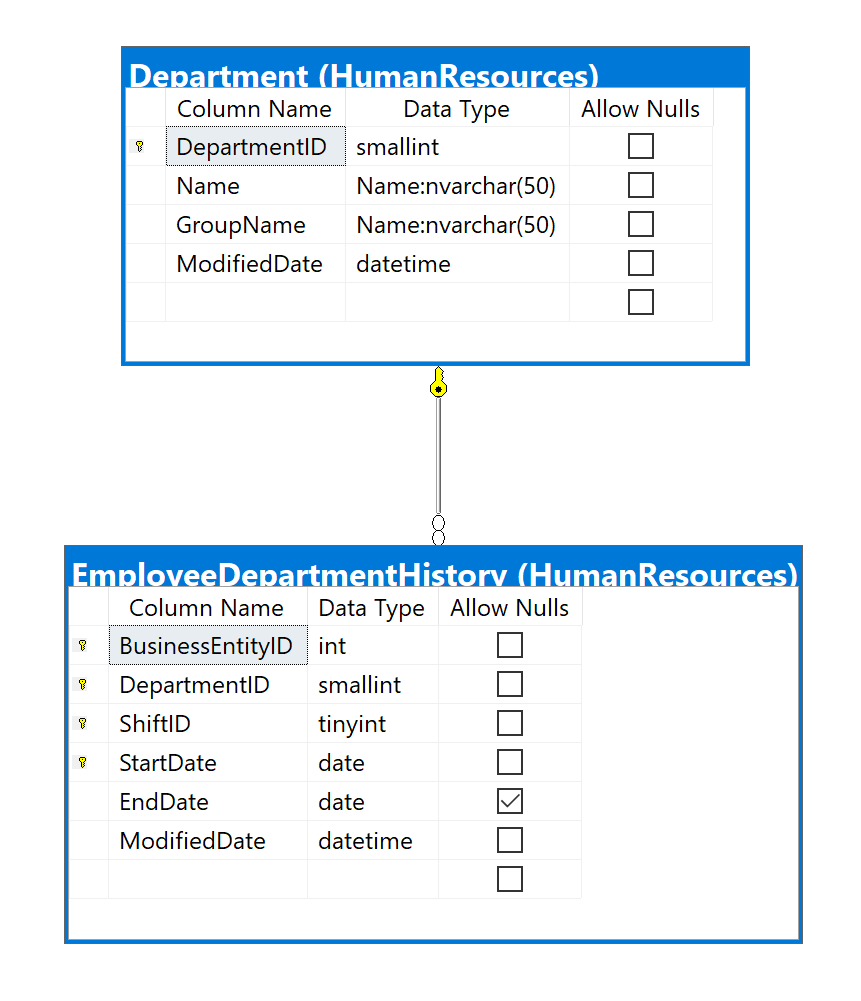
## Database

AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join

## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Department | DepartmentId  Name |
| Derived Column | Employees  NoLongerInDepartment |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT d.DepartmentId

,d.[Name]

,COUNT(dh.BusinessEntityId) AS Employees

,COUNT(dh.EndDate) AS NoLongerInDepartment

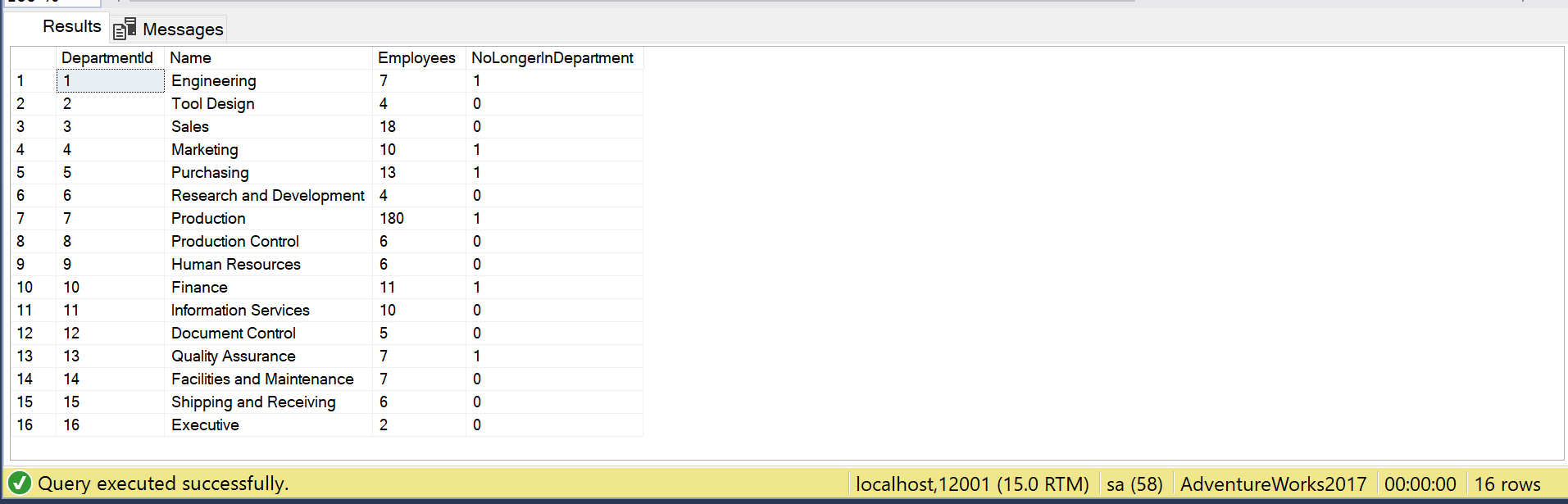
FROM HumanResources.Department AS d

INNER JOIN HumanResources.EmployeeDepartmentHistory AS dh ON d.DepartmentId = dh.DepartmentId

GROUP BY d.DepartmentId

,d.[Name];

## Relational Output with total number of rows returned (16)



## Sample JSON Output with total number of rows returned (16)

USE AdventureWorks2017;

GO

SELECT d.DepartmentId

,d.[Name]

,COUNT(dh.BusinessEntityId) AS Employees

,COUNT(dh.EndDate) AS NoLongerInDepartment

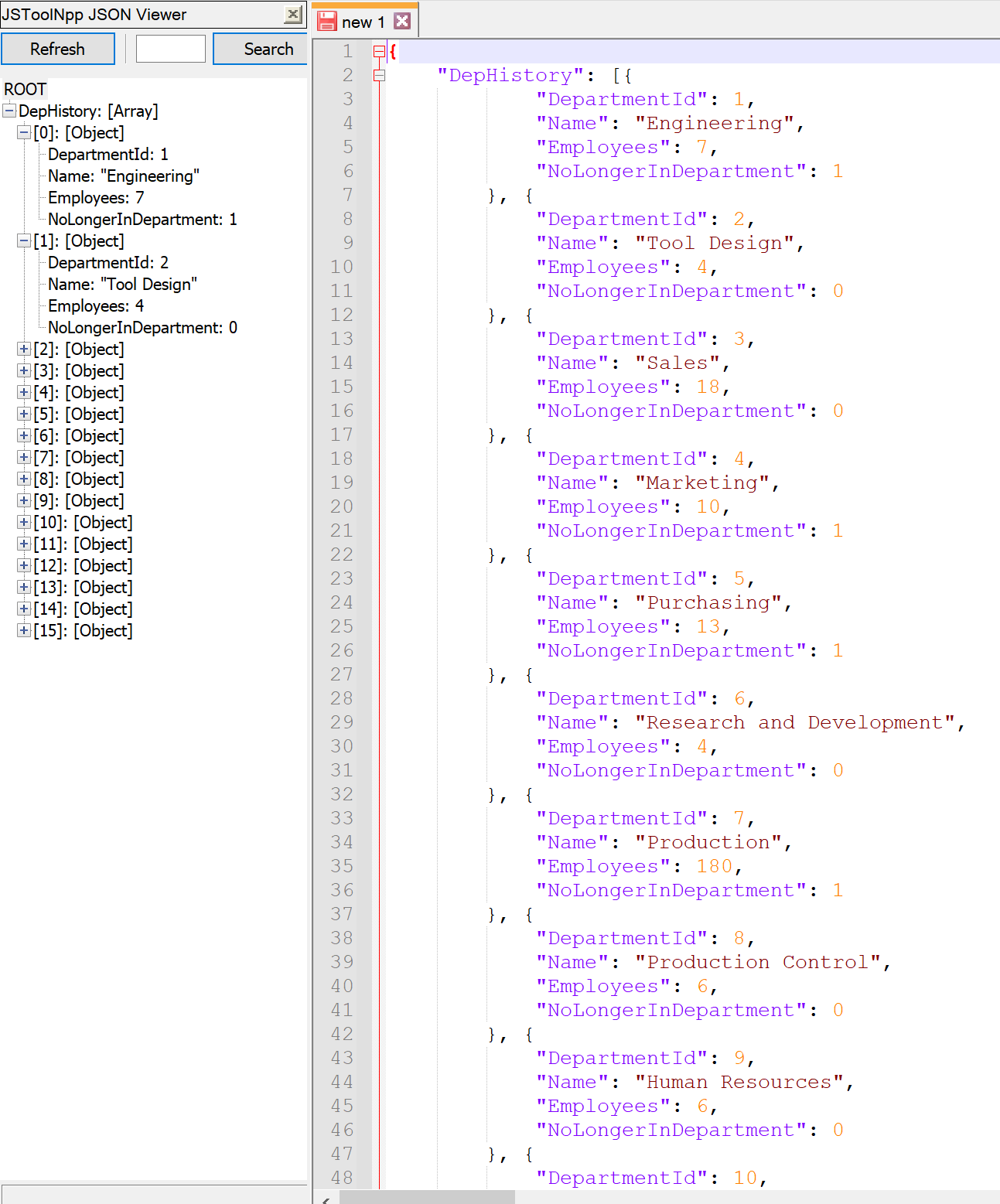
FROM HumanResources.Department AS d

INNER JOIN HumanResources.EmployeeDepartmentHistory AS dh ON d.DepartmentId = dh.DepartmentId

GROUP BY d.DepartmentId

,d.[Name]

FOR JSON PATH, ROOT('DepHistory'), INCLUDE\_NULL\_VALUES;



# Proposition 07: For each order, how many types of products were ordered and what was the quantity using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

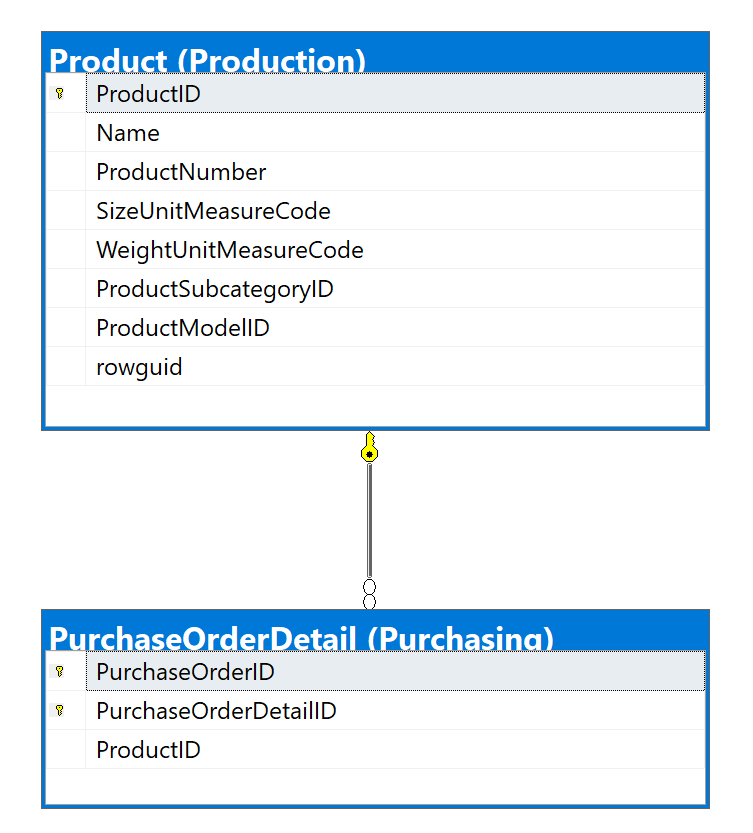
Based on the PurchasingOrderId, count the number of productids in each order and sum the quantity, listing out the names of the products by joining to the products table.

## Database

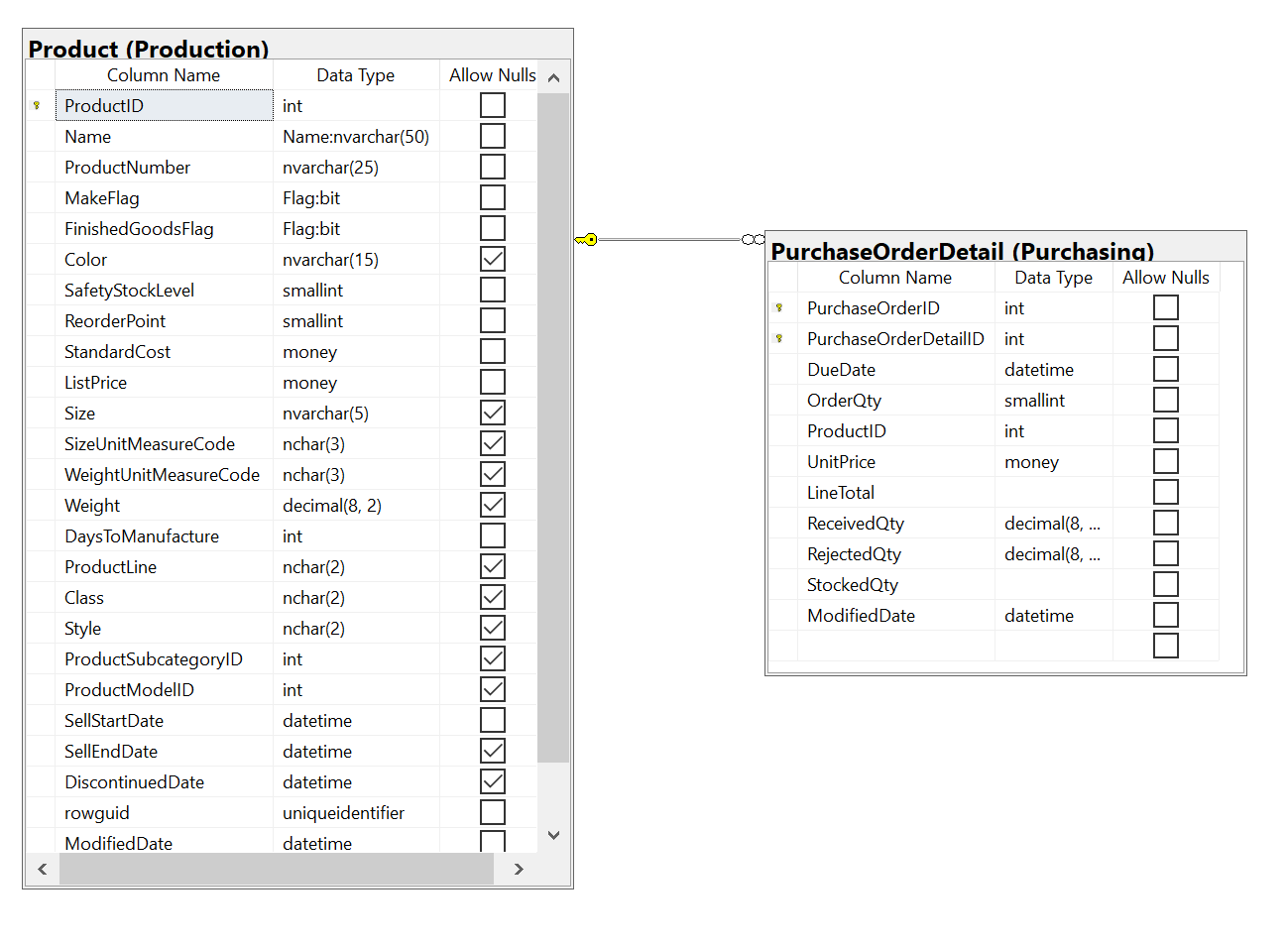
AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| PurchaseOrderDetails | PruchaseOrderId |
| Derived Column | NumOfProducts  Products  TotalQuantity |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT po.PurchaseOrderId

,COUNT(po.ProductId) AS NumOfProducts

,STRING\_AGG(p.[Name], ', ') AS Products

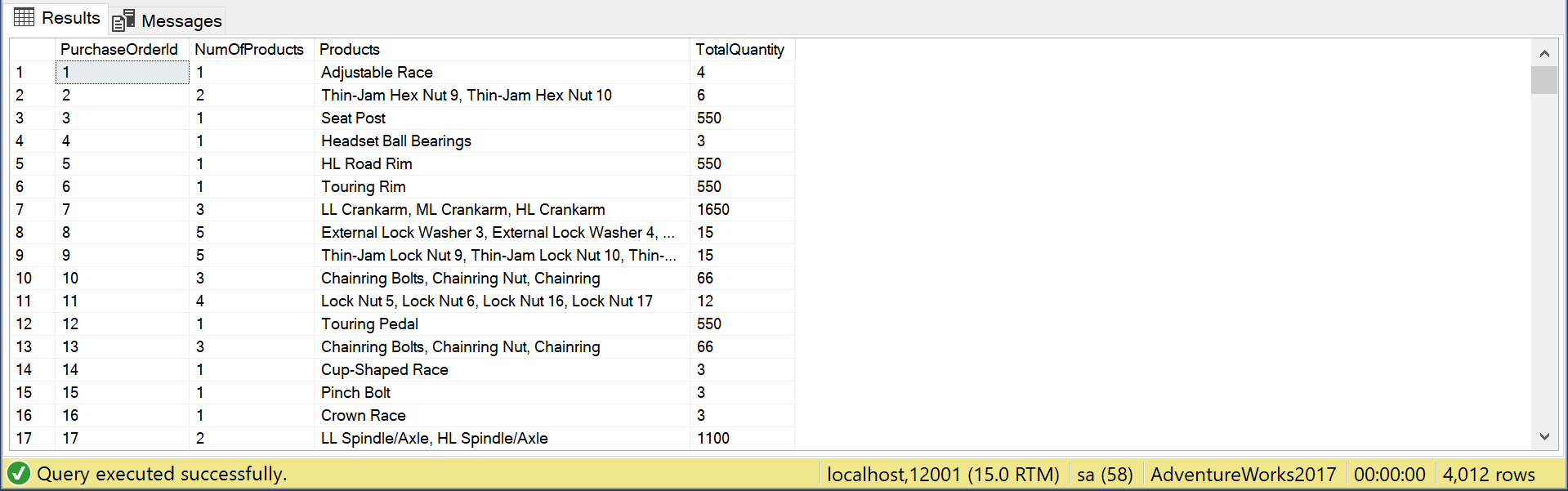
,SUM(po.OrderQty) AS TotalQuantity

FROM Purchasing.PurchaseOrderDetail AS po

INNER JOIN Production.Product AS p ON po.ProductId = p.ProductId

GROUP BY po.PurchaseOrderId;

## Relational Output with total number of rows returned (4012)



## Sample JSON Output with total number of rows returned (4012)

USE AdventureWorks2017;

GO

SELECT po.PurchaseOrderId

,COUNT(po.ProductId) AS NumOfProducts

,STRING\_AGG(p.[Name], ', ') AS Products

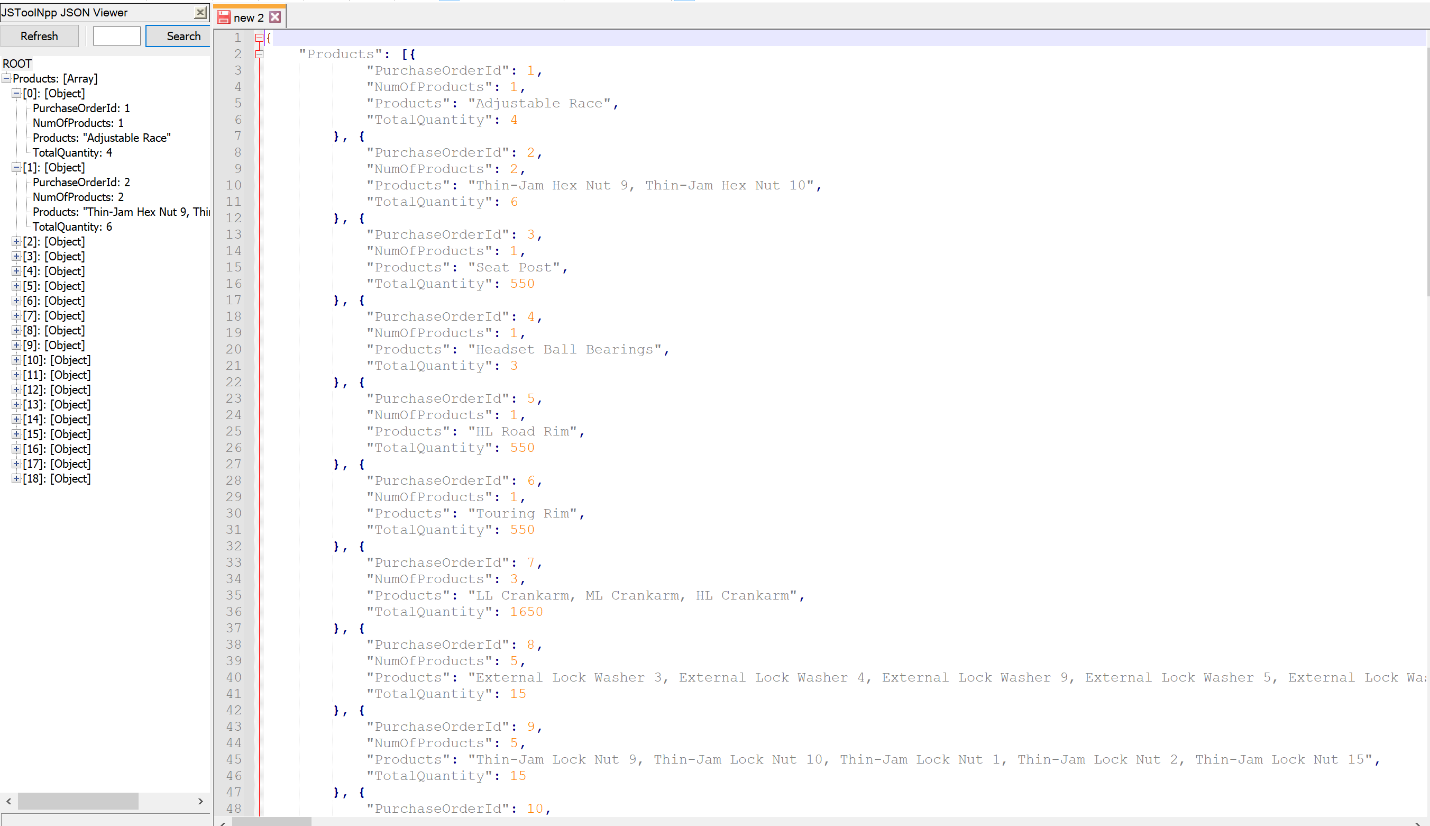
,SUM(po.OrderQty) AS TotalQuantity

FROM Purchasing.PurchaseOrderDetail AS po

INNER JOIN Production.Product AS p ON po.ProductId = p.ProductId

GROUP BY po.PurchaseOrderId

FOR JSON PATH, ROOT('Products'), INCLUDE\_NULL\_VALUES;



# Proposition 08: What are the total sales the previous year in each country by their currency using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

For each country region code, sum last year’s sales, and also give the currencies that are available in the country.

## Database

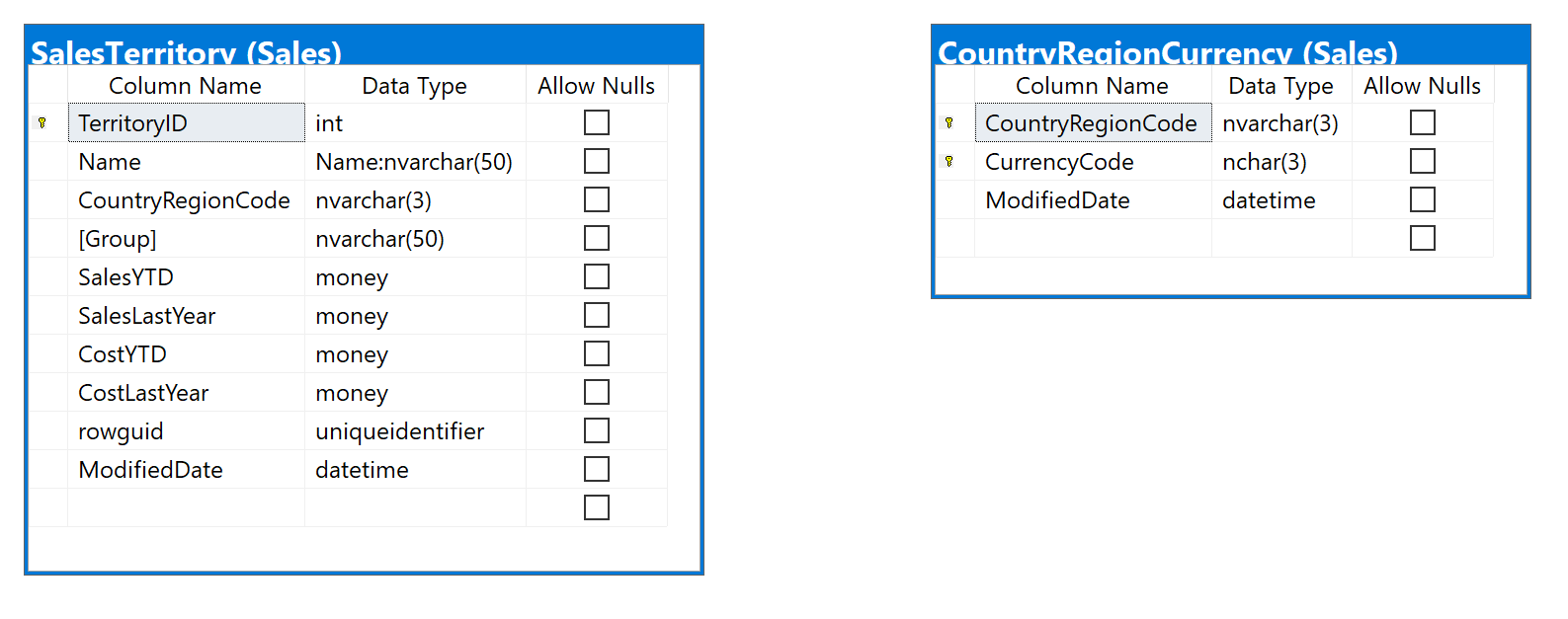
AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| SalesTerritory | CountryRegionCode |
| CountryRegionCurrency | CurrencyCode |
| Derived Column | TotalSalesLastYear |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT t.CountryRegionCode

,c.CurrencyCode

,SUM(t.SalesLastYear) AS TotalSalesLastYear

FROM Sales.SalesTerritory AS t

INNER JOIN Sales.CountryRegionCurrency AS c ON t.CountryRegionCode = c.CountryRegionCode

GROUP BY t.CountryRegionCode

,c.CurrencyCode;

## Relational Output with total number of rows returned (8)



## Sample JSON Output with total number of rows returned (8)

USE AdventureWorks2017;

GO

SELECT t.CountryRegionCode

,c.CurrencyCode

,SUM(t.SalesLastYear) AS TotalSalesLastYear

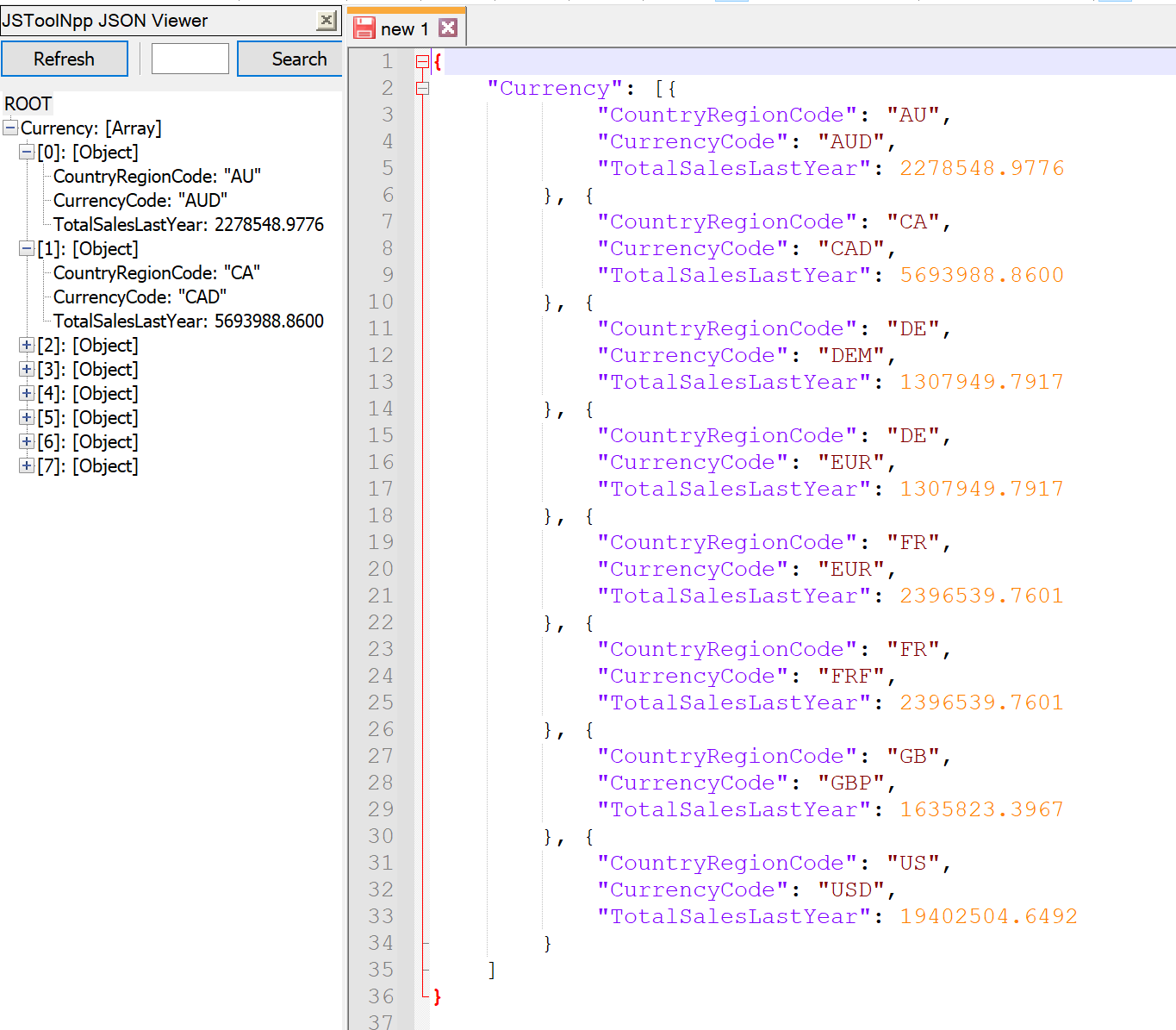
FROM Sales.SalesTerritory AS t

INNER JOIN Sales.CountryRegionCurrency AS c ON t.CountryRegionCode = c.CountryRegionCode

GROUP BY t.CountryRegionCode

,c.CurrencyCode

FOR JSON PATH, ROOT('Currency'), INCLUDE\_NULL\_VALUES;



# Proposition 08 Fixed: What are the total sales the previous year in each country by their currency using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

In the result set, the column TotalSalesLastYear is duplicated for countries that have more than one kind of currency. Using the String\_Agg function to list out the currencies makes sure that the TotalSalesLastYear show up once since each country will only show up once.

## Database

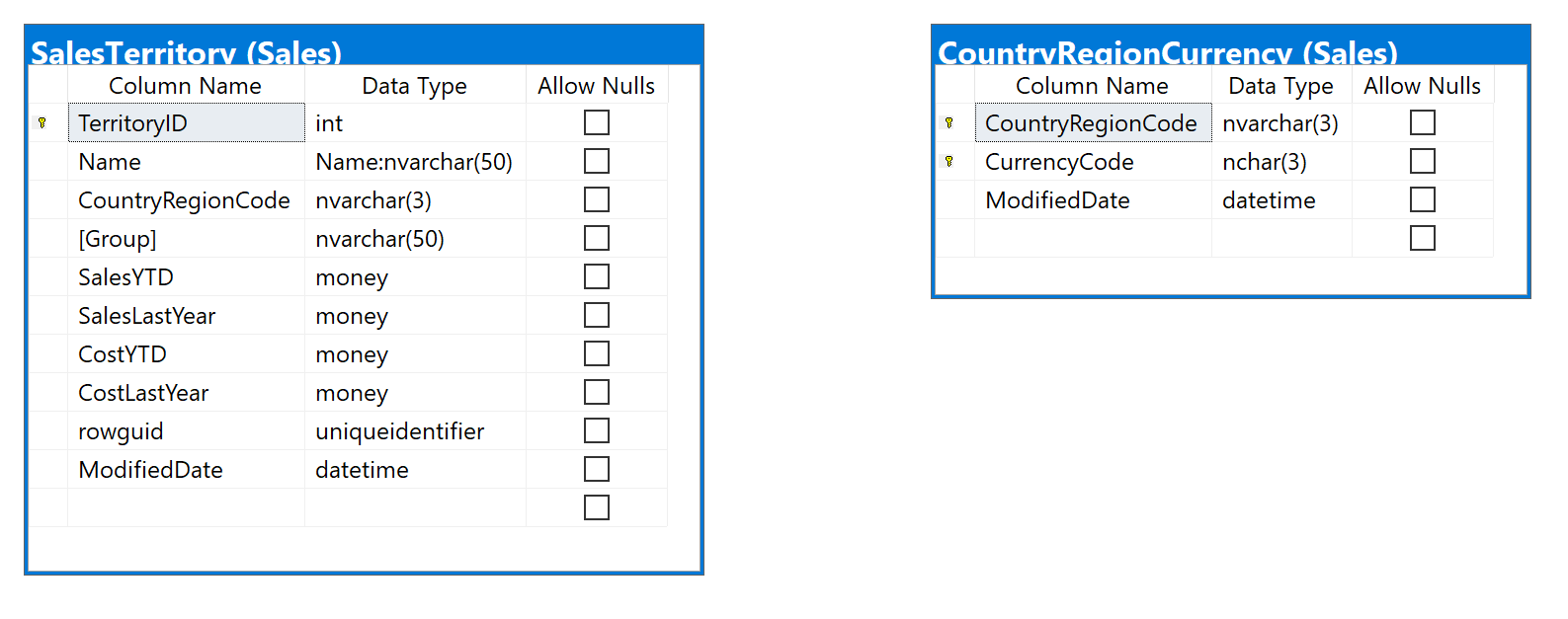
AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| SalesTerritory | CountryRegionCode |
| Derived Column | Currencies  TotalSalesLastYear |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT t.CountryRegionCode

,STRING\_AGG(c.CurrencyCode, ', ') AS Currencies

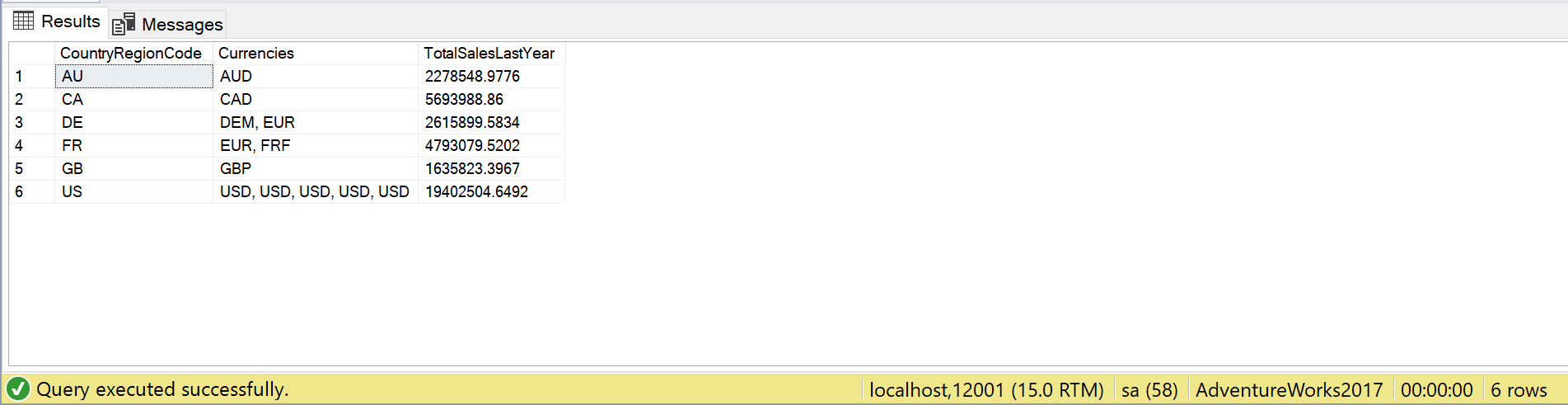
,SUM(t.SalesLastYear) AS TotalSalesLastYear

FROM Sales.SalesTerritory AS t

INNER JOIN Sales.CountryRegionCurrency AS c ON t.CountryRegionCode = c.CountryRegionCode

GROUP BY t.CountryRegionCode;

## Relational Output with total number of rows returned (6)



## Sample JSON Output with total number of rows returned (6)

USE AdventureWorks2017;

GO

SELECT t.CountryRegionCode

,STRING\_AGG(c.CurrencyCode, ', ') AS Currencies

,SUM(t.SalesLastYear) AS TotalSalesLastYear

FROM Sales.SalesTerritory AS t

INNER JOIN Sales.CountryRegionCurrency AS c ON t.CountryRegionCode = c.CountryRegionCode

GROUP BY t.CountryRegionCode

FOR JSON PATH, ROOT('Currency'), INCLUDE\_NULL\_VALUES;



# Proposition 09: How many of products with a list price between 5 and 10 dollars were bought using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

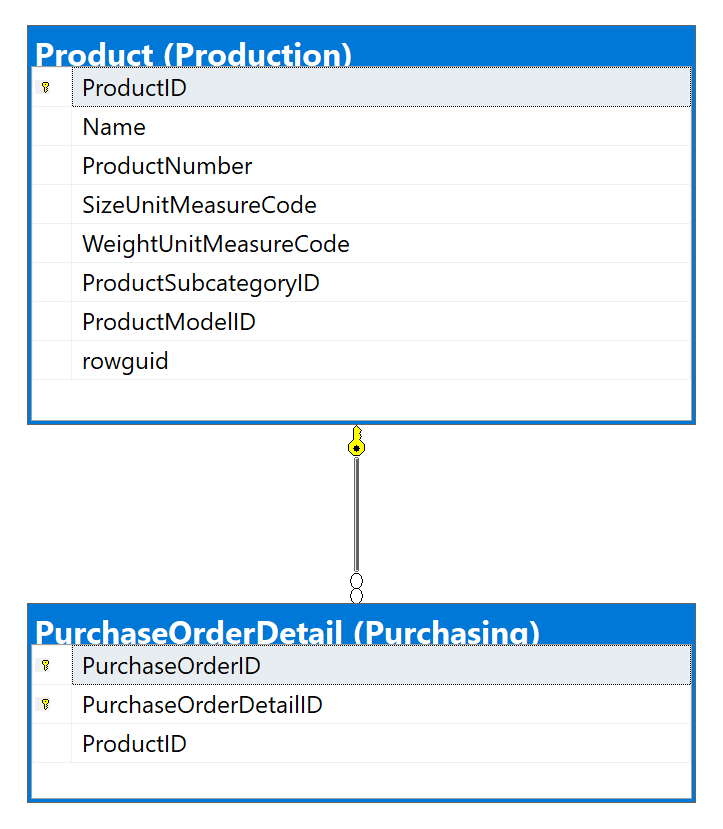
For each productid, sum the order quantities to find how many were bought for products with a list price between 5 and 10 dollars, joining the PurchaseOrderDetail and Product tables.

## Database

AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Product | ProductId  Name  ListPrice |
| Derived Column | TotalSold  Gross |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT p.ProductId

,p.[Name]

,p.ListPrice

,SUM(po.OrderQty) AS TotalSold

,(p.ListPrice \* SUM(po.OrderQty)) AS Gross

FROM Production.Product AS p

INNER JOIN Purchasing.PurchaseOrderDetail AS po ON p.ProductId = po.ProductId

WHERE p.ListPrice BETWEEN 5

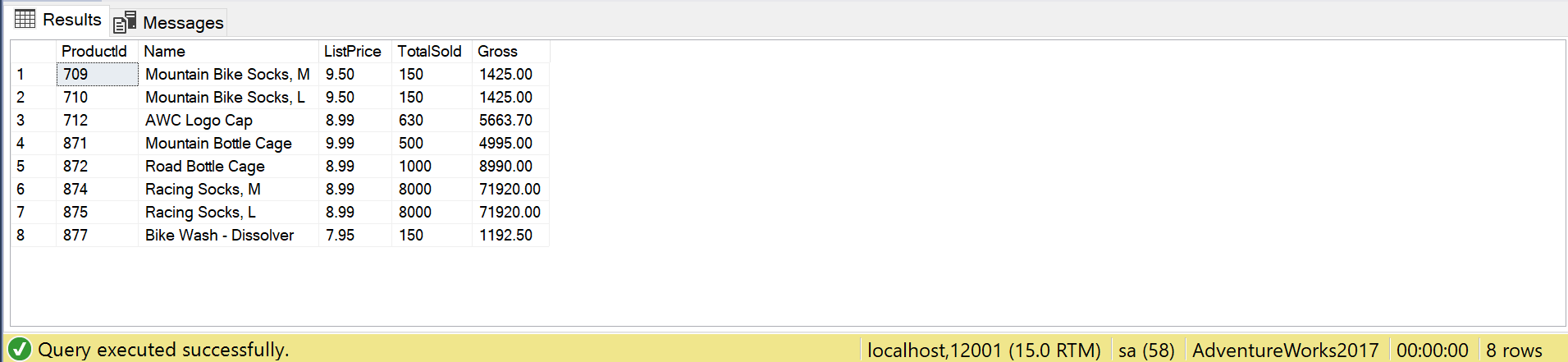
AND 10

GROUP BY p.ProductId

,p.[Name]

,p.ListPrice;

## Relational Output with total number of rows returned (8)



## Sample JSON Output with total number of rows returned (8)

USE AdventureWorks2017;

GO

SELECT p.ProductId

,p.[Name]

,p.ListPrice

,SUM(po.OrderQty) AS TotalSold

,(p.ListPrice \* SUM(po.OrderQty)) AS Gross

FROM Production.Product AS p

INNER JOIN Purchasing.PurchaseOrderDetail AS po ON p.ProductId = po.ProductId

WHERE p.ListPrice BETWEEN 5

AND 10

GROUP BY p.ProductId

,p.[Name]

,p.ListPrice

FOR JSON PATH, ROOT('Purchasing'), INCLUDE\_NULL\_VALUES;



# Proposition 10: For each item ordered, what was the quantity and list price cost difference using AdventureWorks2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

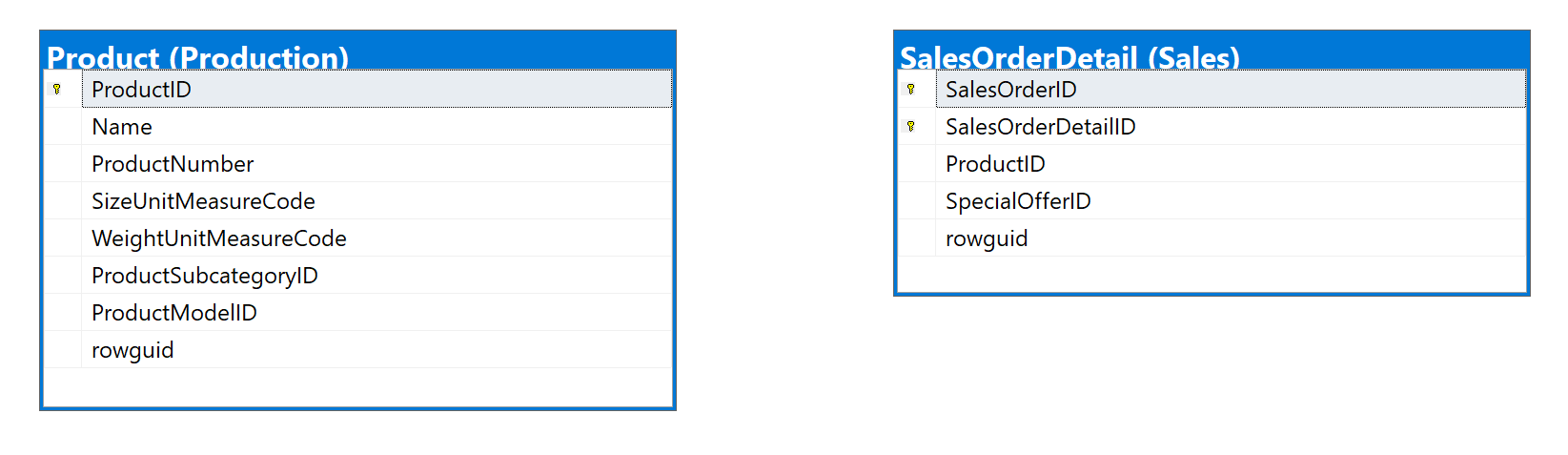
For each product in SalesOrderDetail, sum the quantity, list price, and standard cost to find the profit made on each product.

## Database

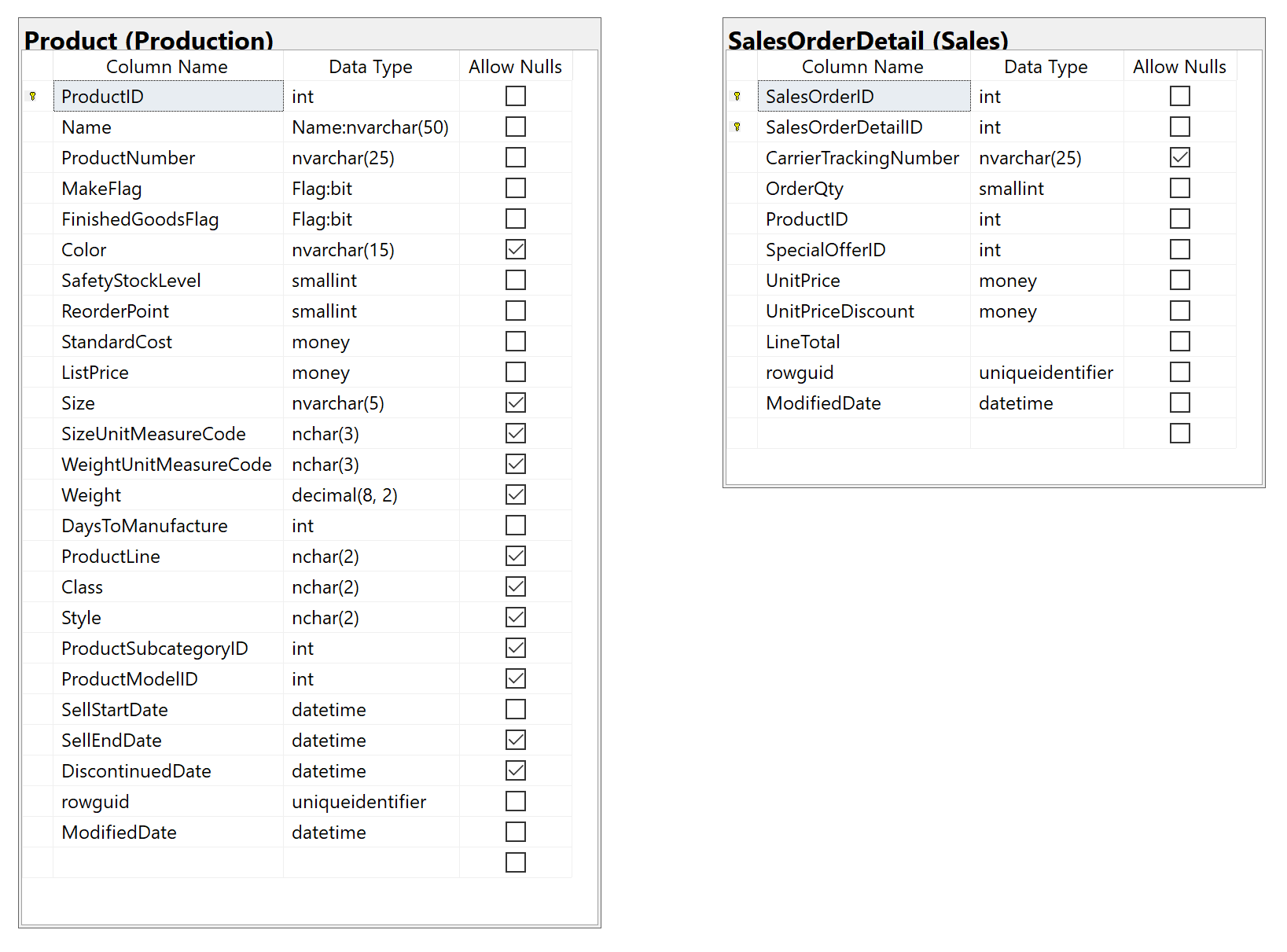
AdventureWorks2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| SalesOrderDetail | ProductId |
| Derived Column | Quantity  ListPrice  Cost  Profit |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| SalesOrderDetail | ProductId | DESC |

## Problem solving Query

USE AdventureWorks2017;

GO

SELECT s.ProductId

,SUM(s.OrderQty) AS Quantity

,SUM(p.ListPrice) AS ListPrice

,SUM(p.StandardCost) AS Cost

,(SUM(p.ListPrice) - SUM(p.StandardCost)) \* SUM(s.OrderQty) AS Profit

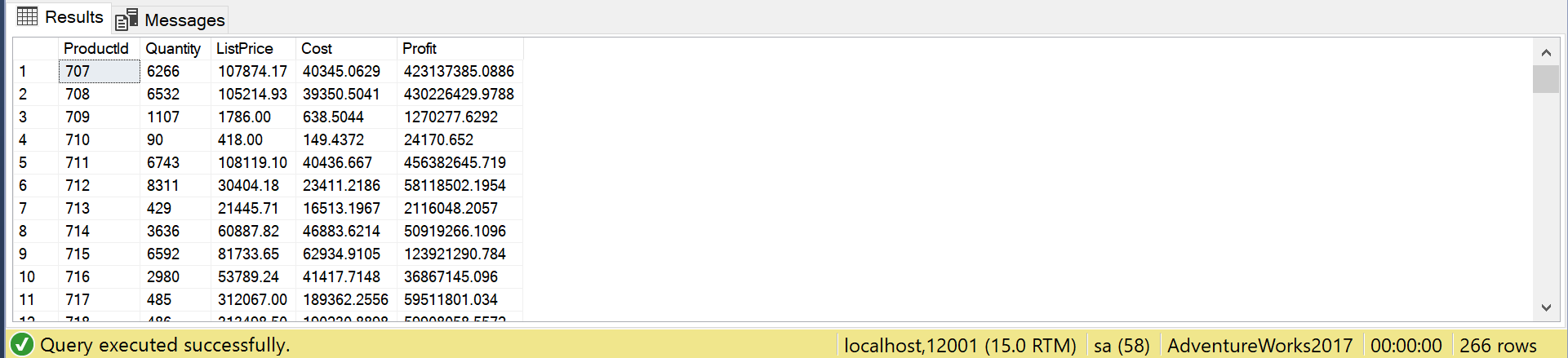
FROM Sales.SalesOrderDetail AS s

INNER JOIN Production.Product AS p ON s.ProductId = p.ProductId

GROUP BY s.ProductId

ORDER BY s.ProductId;

## Relational Output with total number of rows returned (266)



## Sample JSON Output with total number of rows returned (266)

USE AdventureWorks2017;

GO

SELECT s.ProductId

,SUM(s.OrderQty) AS Quantity

,SUM(p.ListPrice) AS ListPrice

,SUM(p.StandardCost) AS Cost

,(SUM(p.ListPrice) - SUM(p.StandardCost)) \* SUM(s.OrderQty) AS Profit

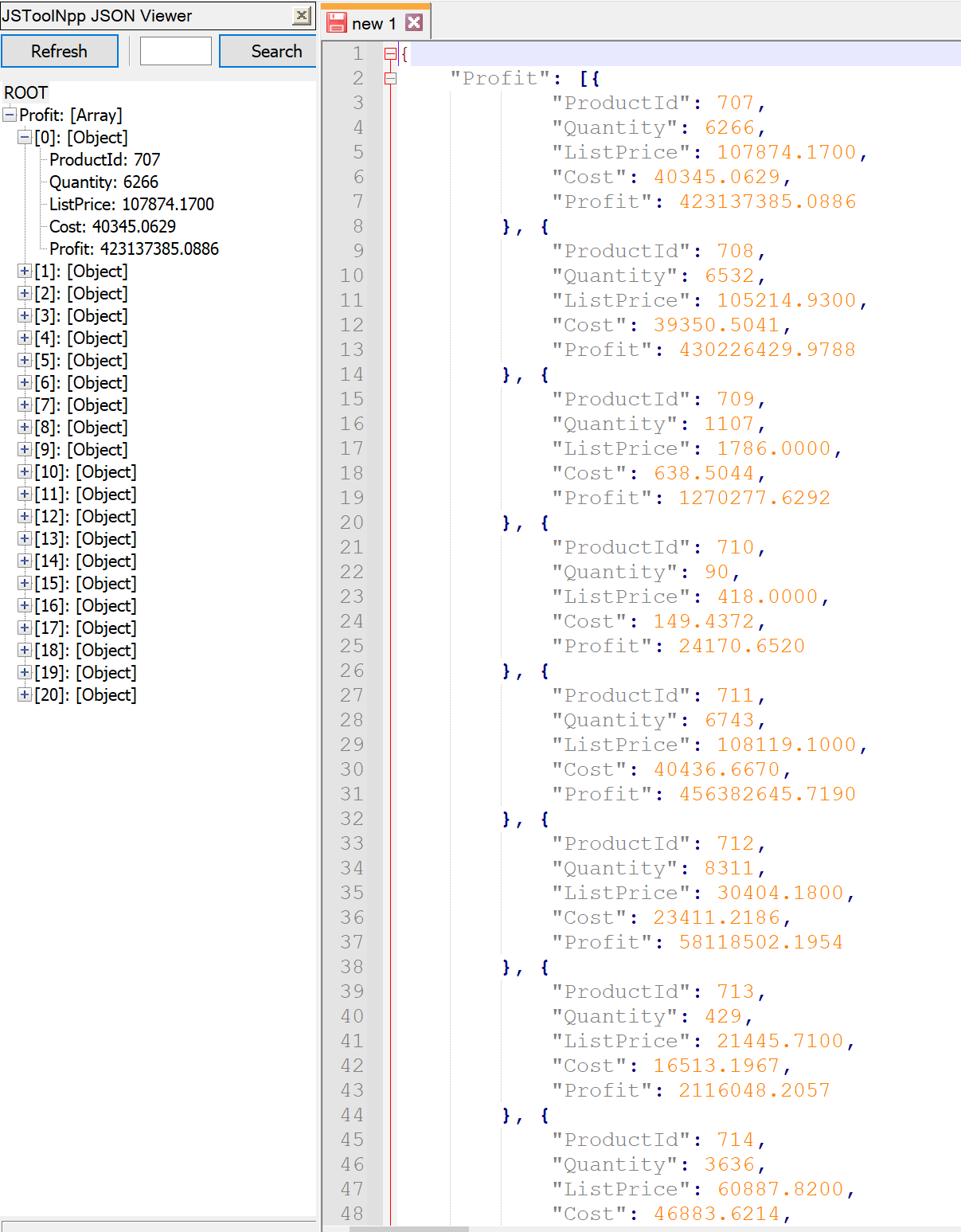
FROM Sales.SalesOrderDetail AS s

INNER JOIN Production.Product AS p ON s.ProductId = p.ProductId

GROUP BY s.ProductId

ORDER BY s.ProductId

FOR JSON PATH, ROOT('Profit'), INCLUDE\_NULL\_VALUES;



# Proposition 11: Which customers had over 10 orders in 2016 using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

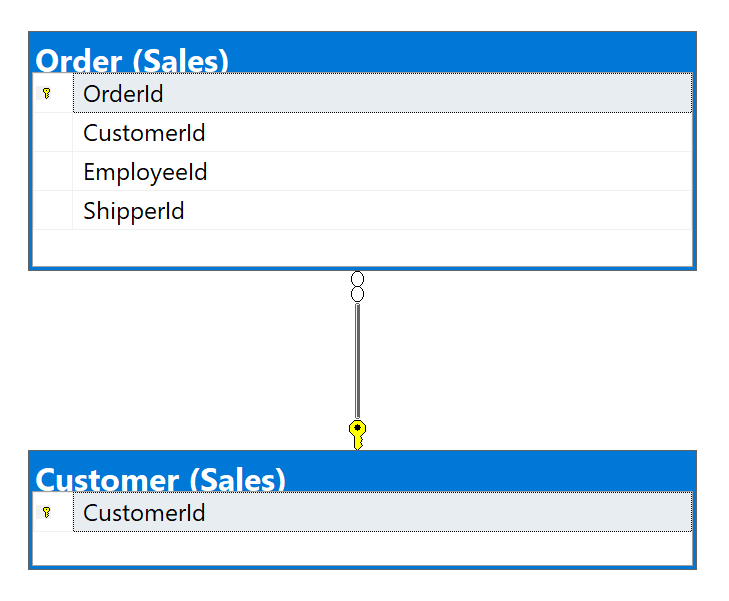
Based on CustomerId, count the number of orders and filter for those over 10

## Database

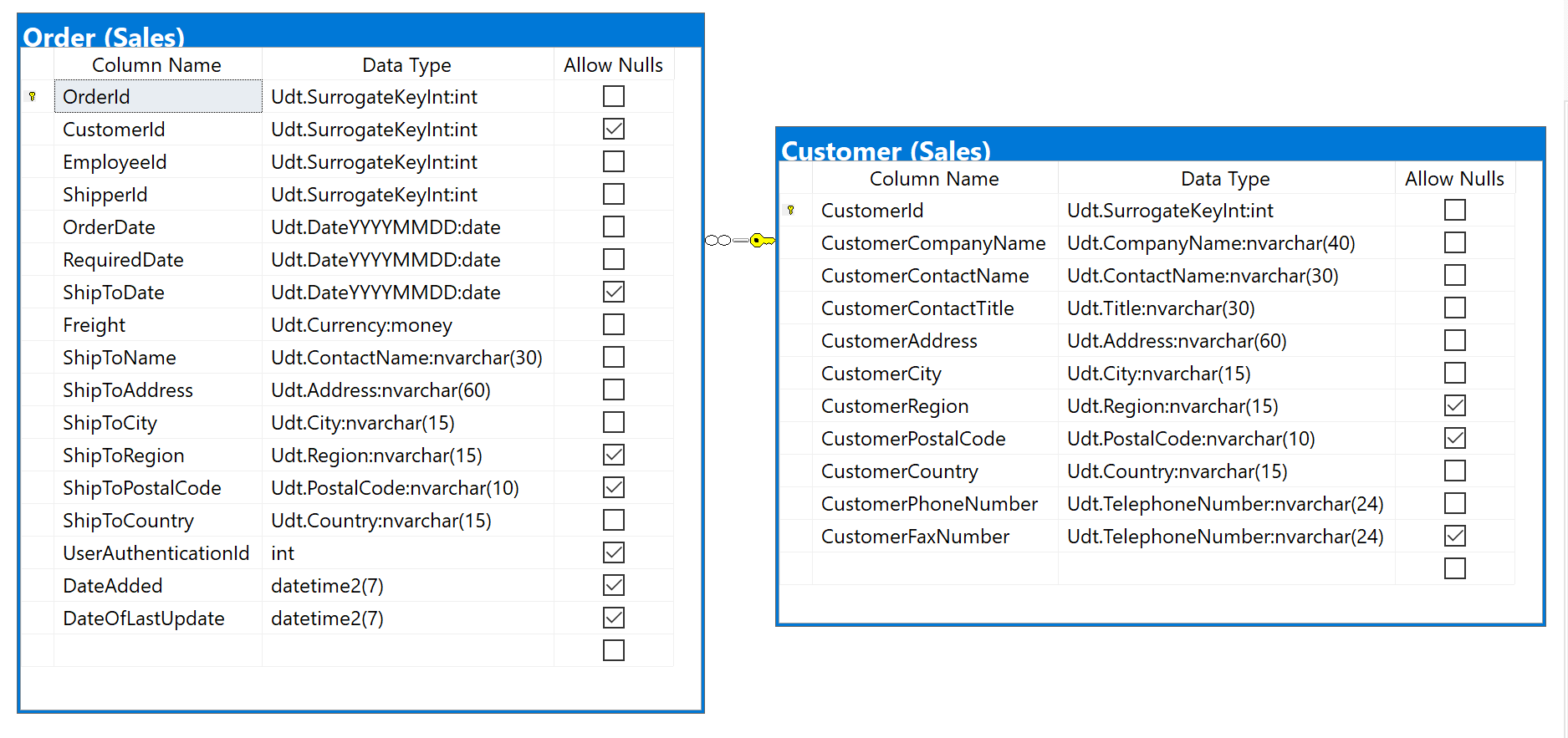
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Customer | CustomerId  CustomerCompanyName |
| Derived Column | NumOfOrders |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

,c.CustomerCompanyName

,COUNT(o.OrderId) AS NumOfOrders

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

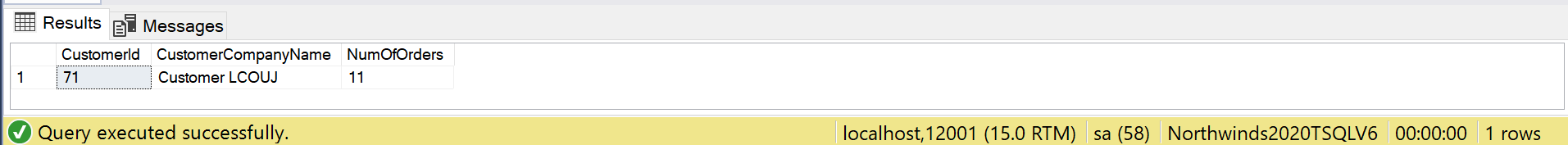
WHERE YEAR(o.OrderDate) = '2016'

GROUP BY c.CustomerId

,c.CustomerCompanyName

HAVING COUNT(o.OrderId) > 10;

## Relational Output with total number of rows returned (1)



## Sample JSON Output with total number of rows returned (1)

USE Northwinds2020TSQLV6;

GO

SELECT c.CustomerId

,c.CustomerCompanyName

,COUNT(o.OrderId) AS NumOfOrders

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

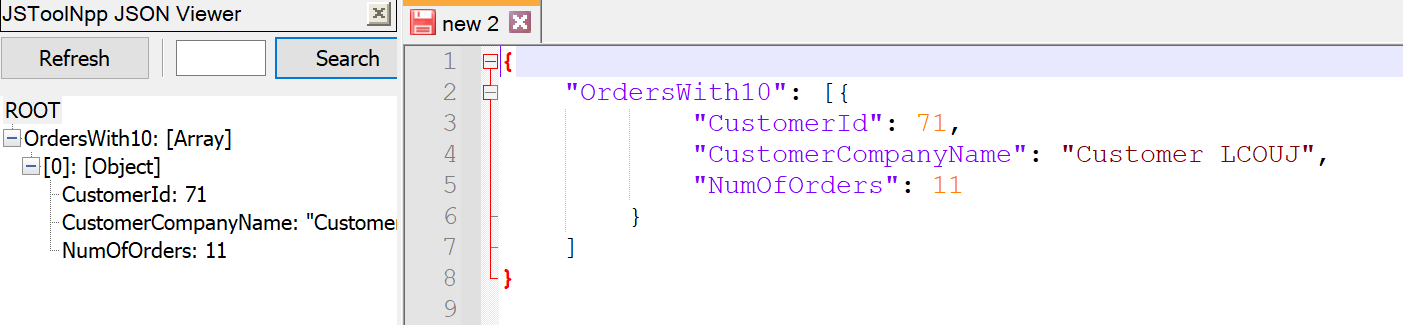
WHERE YEAR(o.OrderDate) = '2016'

GROUP BY c.CustomerId

,c.CustomerCompanyName

HAVING COUNT(o.OrderId) > 10

FOR JSON PATH, ROOT('OrdersWith10'), INCLUDE\_NULL\_VALUES;



# Proposition 12: Which orders had more than 3 items in 2015 using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

Group by the OrderId and count the number of productids, joining the order and orderdetail tables.

## Database

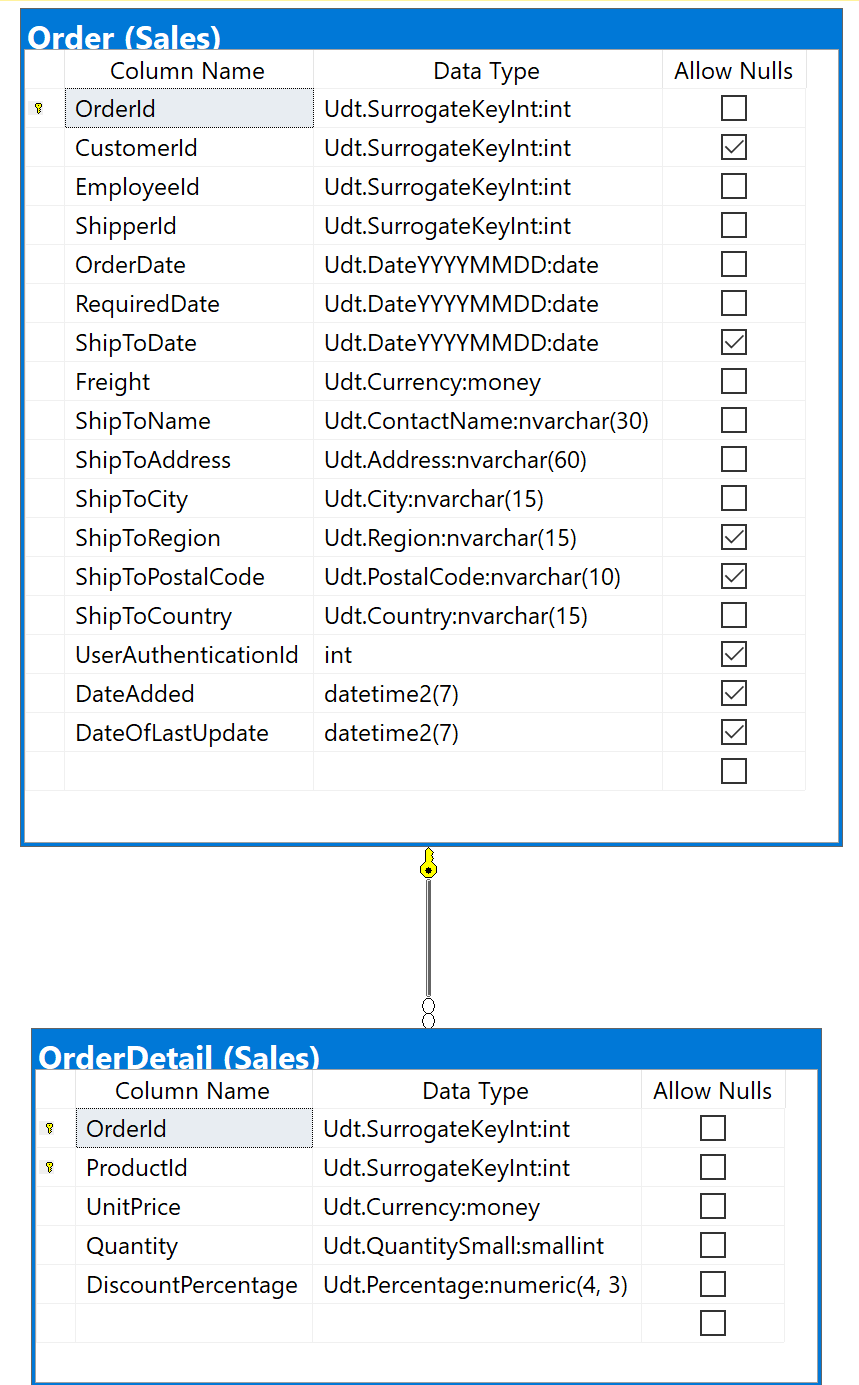
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Employee | OrderId |
| Dervied Column | NumOfItems |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Order | OrderId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT o.OrderId

,COUNT(od.ProductId) AS NumOfItems

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

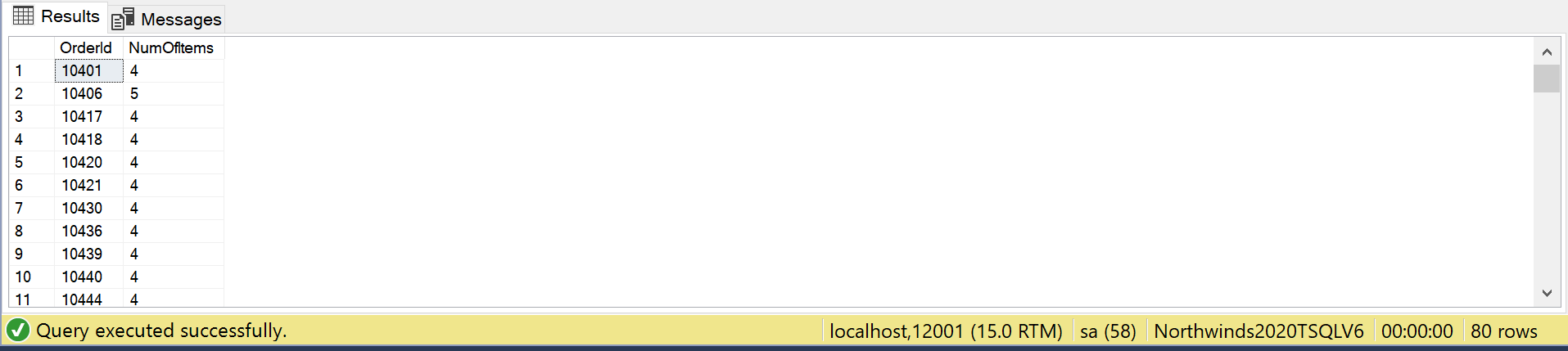
WHERE YEAR(o.OrderDate) = '2015'

GROUP BY o.OrderId

HAVING COUNT(DISTINCT od.ProductId) > 3

## ORDER BY o.OrderId;

## Relational Output with total number of rows returned (80)



## Sample JSON Output with total number of rows returned (80)

USE Northwinds2020TSQLV6;

GO

SELECT o.OrderId

,COUNT(od.ProductId) AS NumOfItems

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

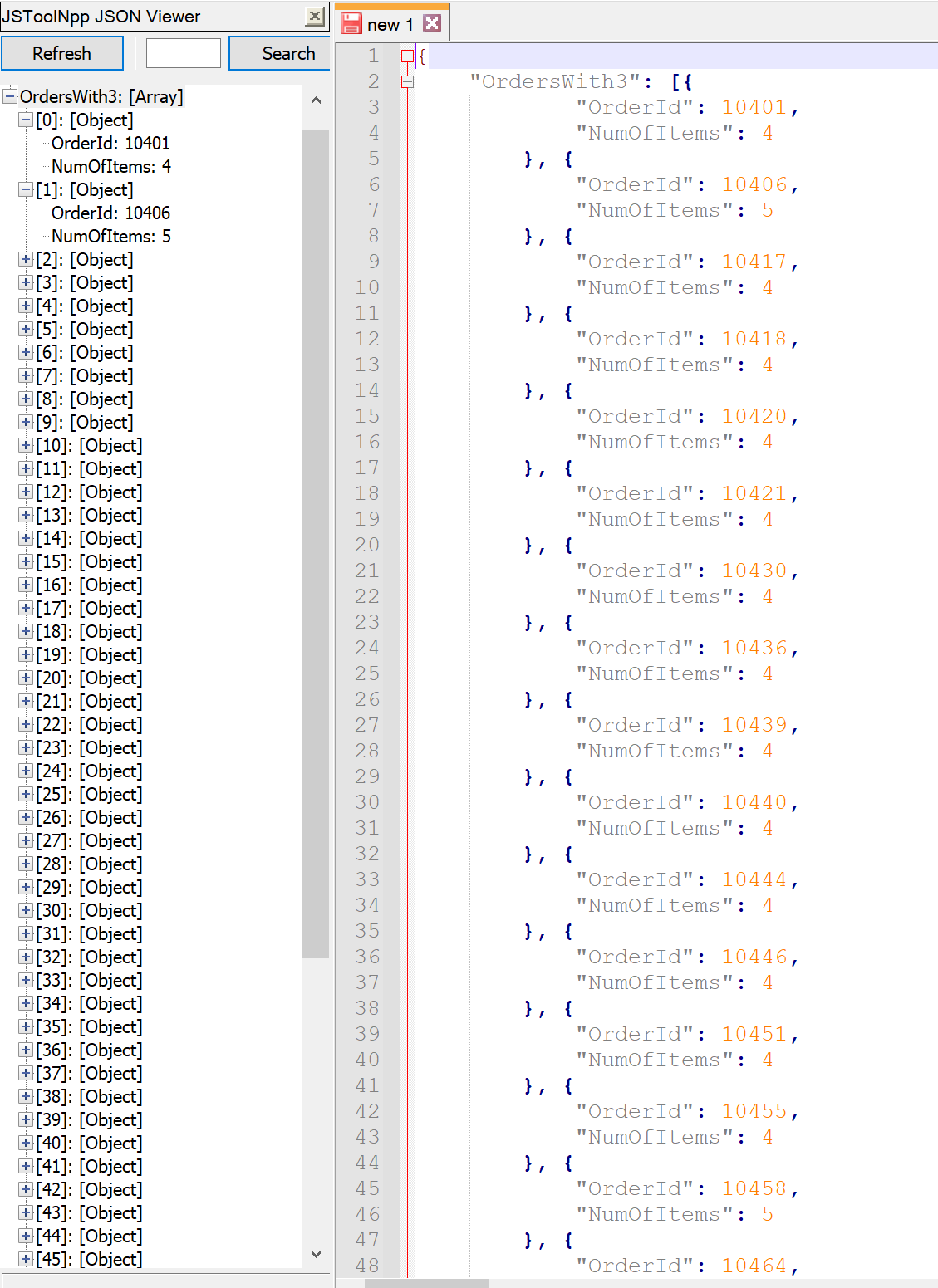
WHERE YEAR(o.OrderDate) = '2015'

GROUP BY o.OrderId

HAVING COUNT(DISTINCT od.ProductId) > 3

ORDER BY o.OrderId

FOR JSON PATH, ROOT('OrdersWith3'), INCLUDE\_NULL\_VALUES;



# Proposition 13: How many orders did employees make in QI 2015 using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

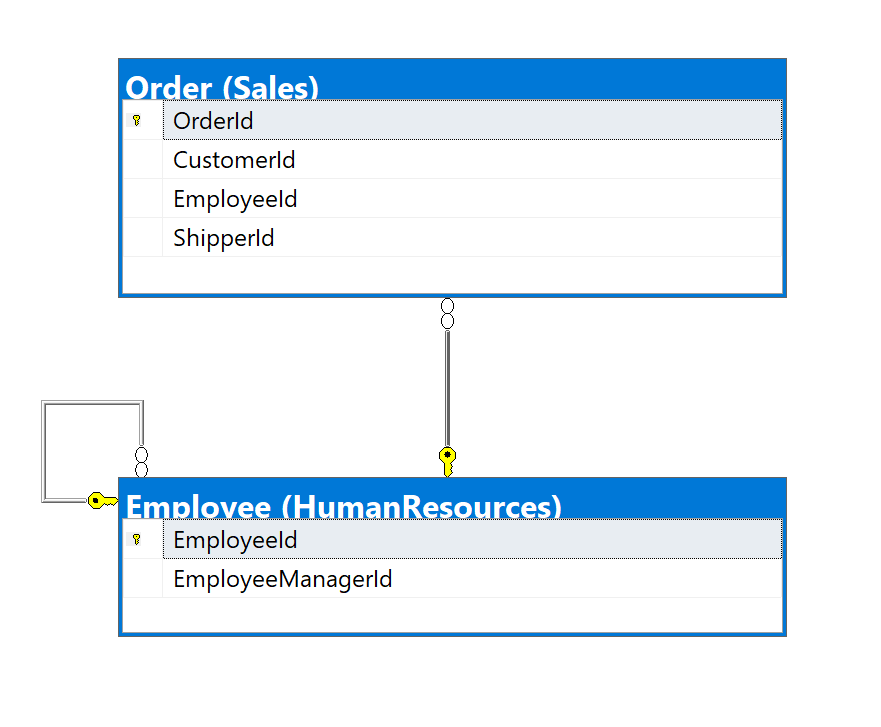
Count the number of OrderIds per employee that have an orderdate between January and March 2015.

## Database

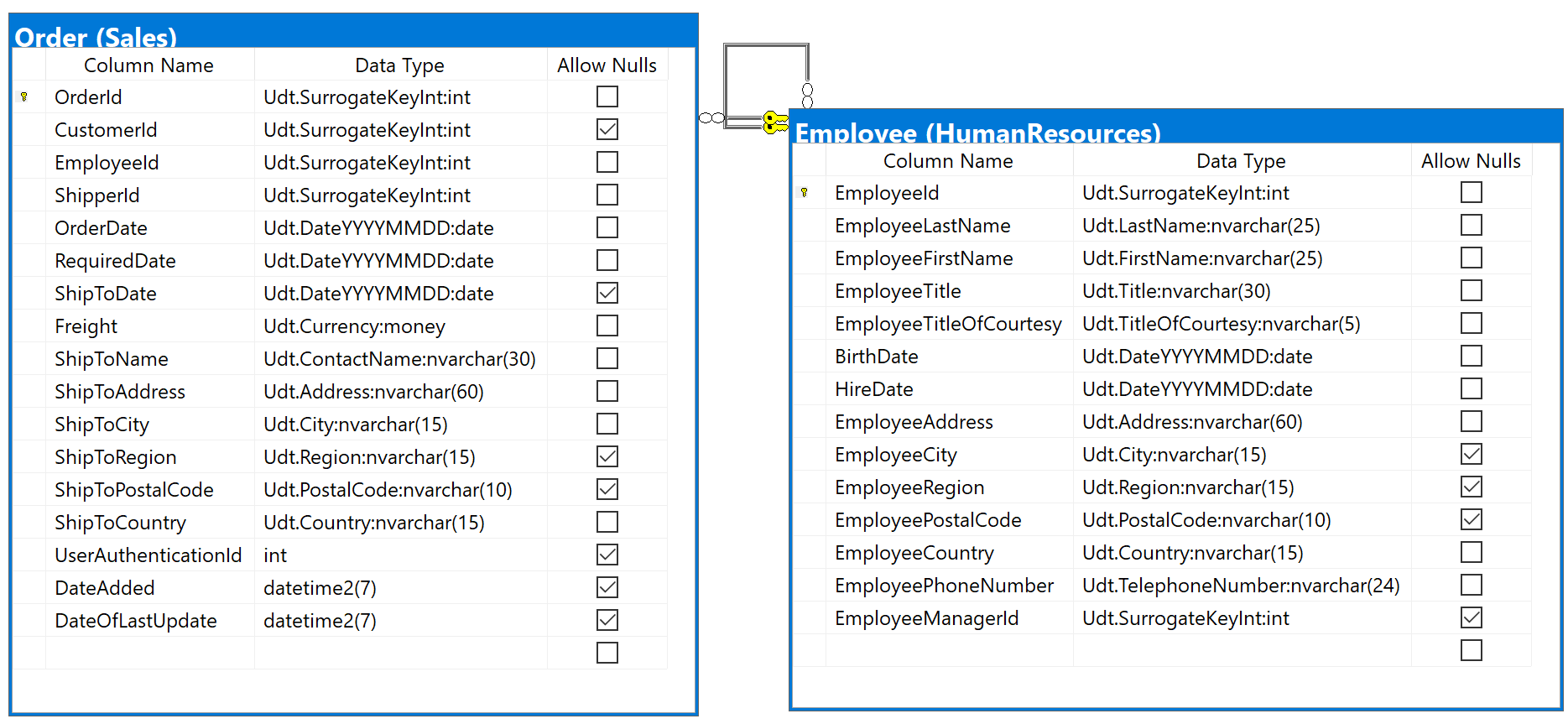
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Employee | EmployeeId |
| Derived Column | NumOfOrders |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

SELECT e.EmployeeId

,COUNT(o.OrderId) AS NumOfOrders

FROM HumanResources.Employee AS e

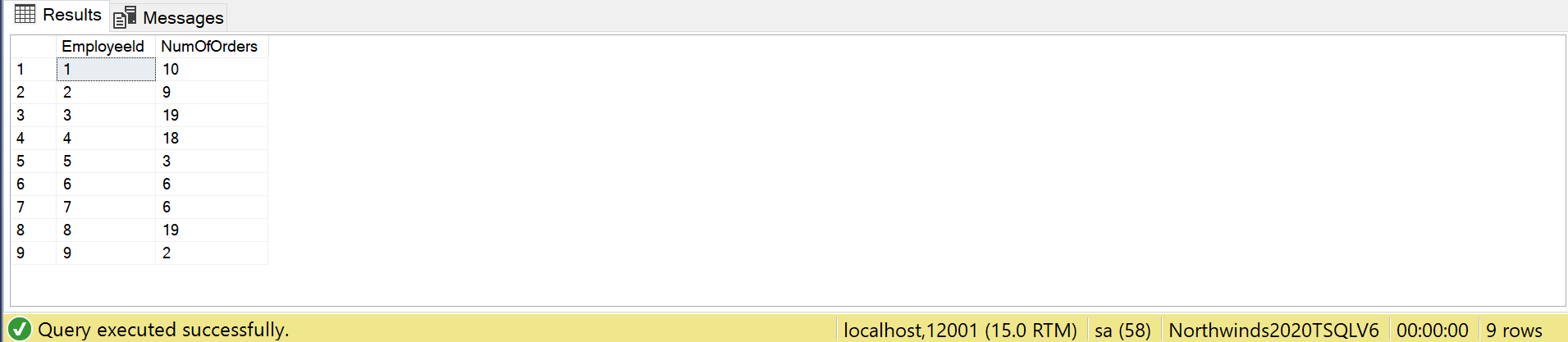
INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

WHERE o.OrderDate BETWEEN '20150101'

AND '20150331'

GROUP BY e.EmployeeId;

## Relational Output with total number of rows returned (9)



## Sample JSON Output with total number of rows returned (9)

USE Northwinds2020TSQLV6;

GO

SELECT e.EmployeeId

,COUNT(o.OrderId) AS NumOfOrders

FROM HumanResources.Employee AS e

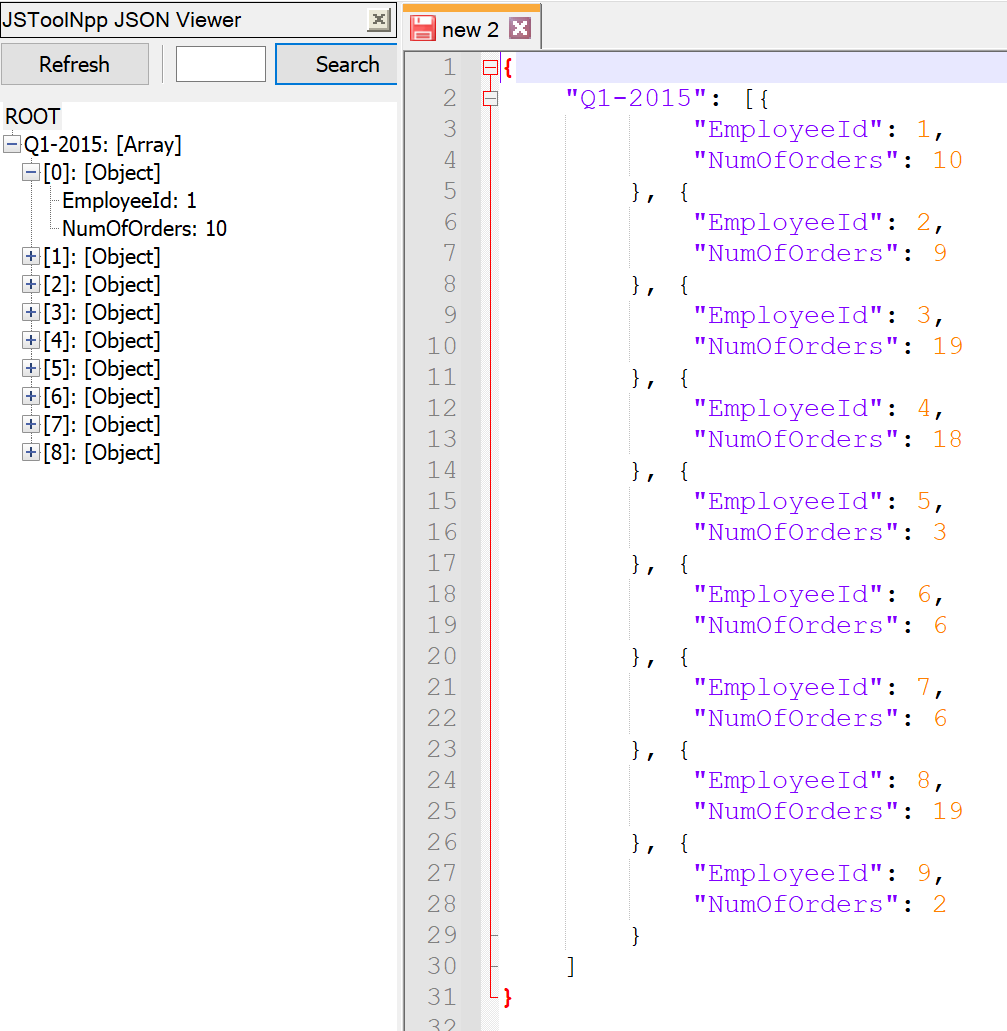
INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

WHERE o.OrderDate BETWEEN '20150101'

AND '20150331'

GROUP BY e.EmployeeId

FOR JSON PATH, ROOT('Q1-2015'), INCLUDE\_NULL\_VALUES;



# Proposition 14: What are the hexadecimal values of Customer Ids, and how many orders did each customer place, and how many products for those orders using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

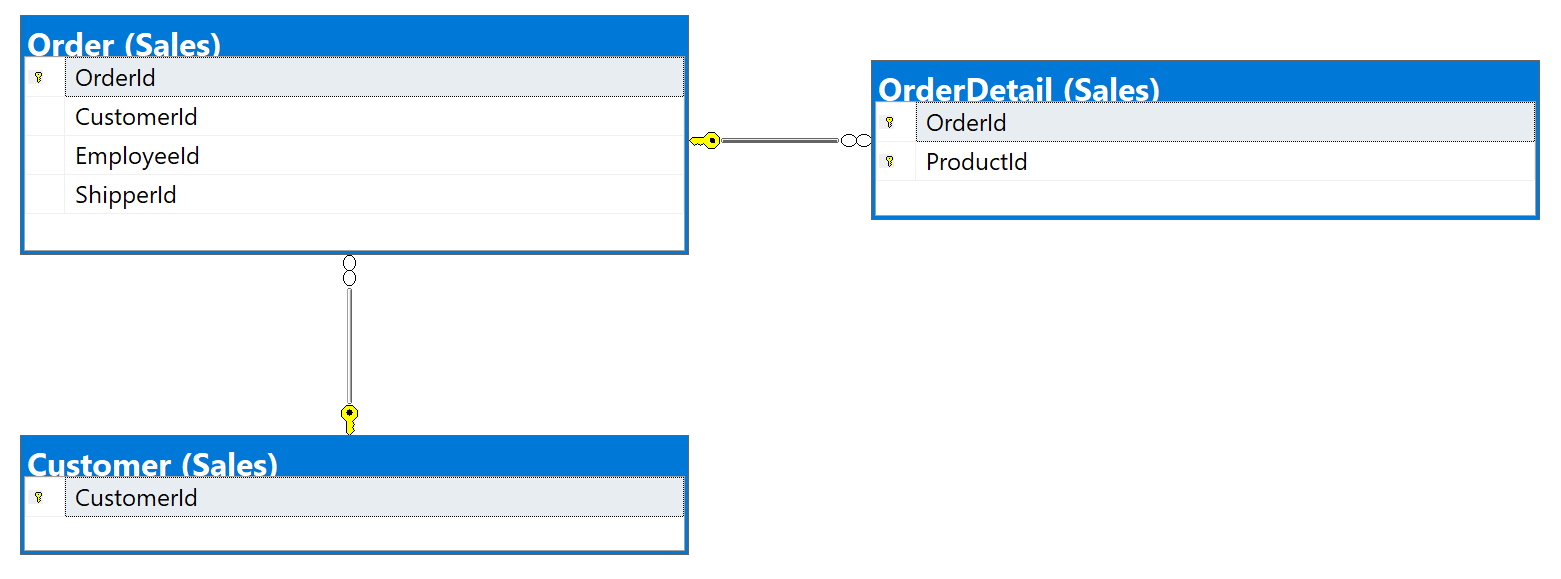
Using a function to convert integers to hexadecimals, convert CustomerIds to hex, and within a subquery count the productids, and then outside count the orderids and sum the counted productids.

## Database

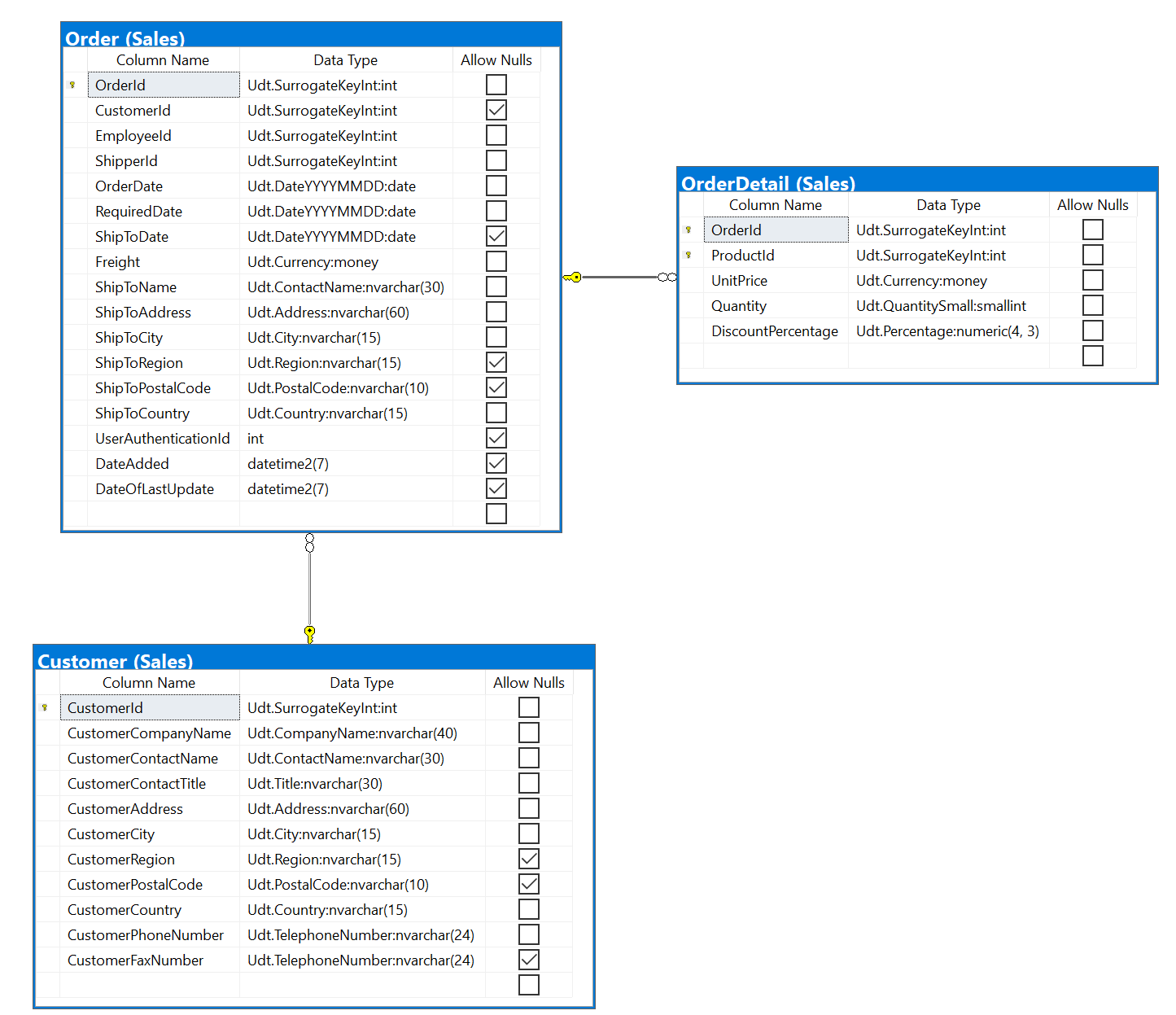
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Derived Column | CustomerId  NumberOfOrders  NumberOfProducts |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Customer (converted to hexadecimal using a scalar function) | CustomerId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToHex;

GO

CREATE FUNCTION dbo.IntToHex (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @hex VARCHAR(30);

WHILE @num > 0

BEGIN

SET @hex = CONCAT (

CASE

WHEN @num % 16 < 10

THEN CAST(@num % 16 AS VARCHAR)

WHEN @num % 16 = 10

THEN 'A'

WHEN @num % 16 = 11

THEN 'B'

WHEN @num % 16 = 12

THEN 'C'

WHEN @num % 16 = 13

THEN 'D'

WHEN @num % 16 = 14

THEN 'E'

WHEN @num % 16 = 15

THEN 'F'

END

,@hex

);

SET @num = @num / 16;

END;

RETURN @hex;

END;

GO

SELECT dbo.IntToHex(c.CustomerId) AS CustomerId

,COUNT(o.OrderId) AS NumberOfOrders

,SUM(o.OrderDetail) AS NumberOfProducts

FROM Sales.Customer AS c

LEFT OUTER JOIN (

SELECT o.OrderId

,o.CustomerId

,COUNT(od.ProductId) AS OrderDetail

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

GROUP BY o.OrderId

,o.CustomerId

) AS o ON c.CustomerId = o.CustomerId

GROUP BY c.CustomerId

ORDER BY CustomerId;

## Relational Output with total number of rows returned (91)



## Sample JSON Output with total number of rows returned (91)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToHex;

GO

CREATE FUNCTION dbo.IntToHex (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @hex VARCHAR(30);

WHILE @num > 0

BEGIN

SET @hex = CONCAT (

CASE

WHEN @num % 16 < 10

THEN CAST(@num % 16 AS VARCHAR)

WHEN @num % 16 = 10

THEN 'A'

WHEN @num % 16 = 11

THEN 'B'

WHEN @num % 16 = 12

THEN 'C'

WHEN @num % 16 = 13

THEN 'D'

WHEN @num % 16 = 14

THEN 'E'

WHEN @num % 16 = 15

THEN 'F'

END

,@hex

);

SET @num = @num / 16;

END;

RETURN @hex;

END;

GO

SELECT dbo.IntToHex(c.CustomerId) AS CustomerId

,COUNT(o.OrderId) AS NumberOfOrders

,SUM(o.OrderDetail) AS NumberOfProducts

FROM Sales.Customer AS c

LEFT OUTER JOIN (

SELECT o.OrderId

,o.CustomerId

,COUNT(od.ProductId) AS OrderDetail

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

GROUP BY o.OrderId

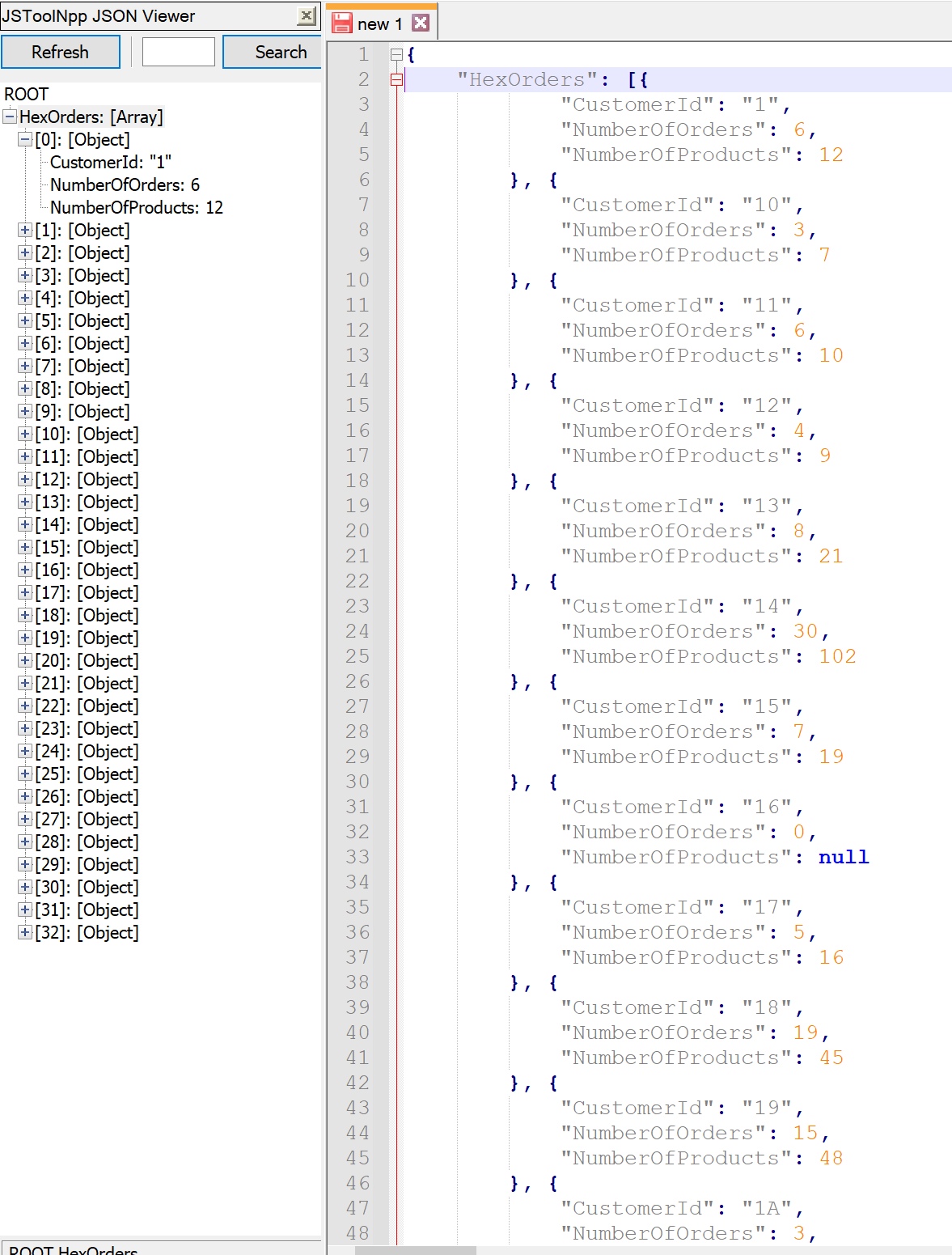
,o.CustomerId

) AS o ON c.CustomerId = o.CustomerId

GROUP BY c.CustomerId

ORDER BY CustomerId

FOR JSON PATH, ROOT('HexOrders'), INCLUDE\_NULL\_VALUES;



# Proposition 15: What is the name of each employee, and how many orders did they sell and with how many items using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

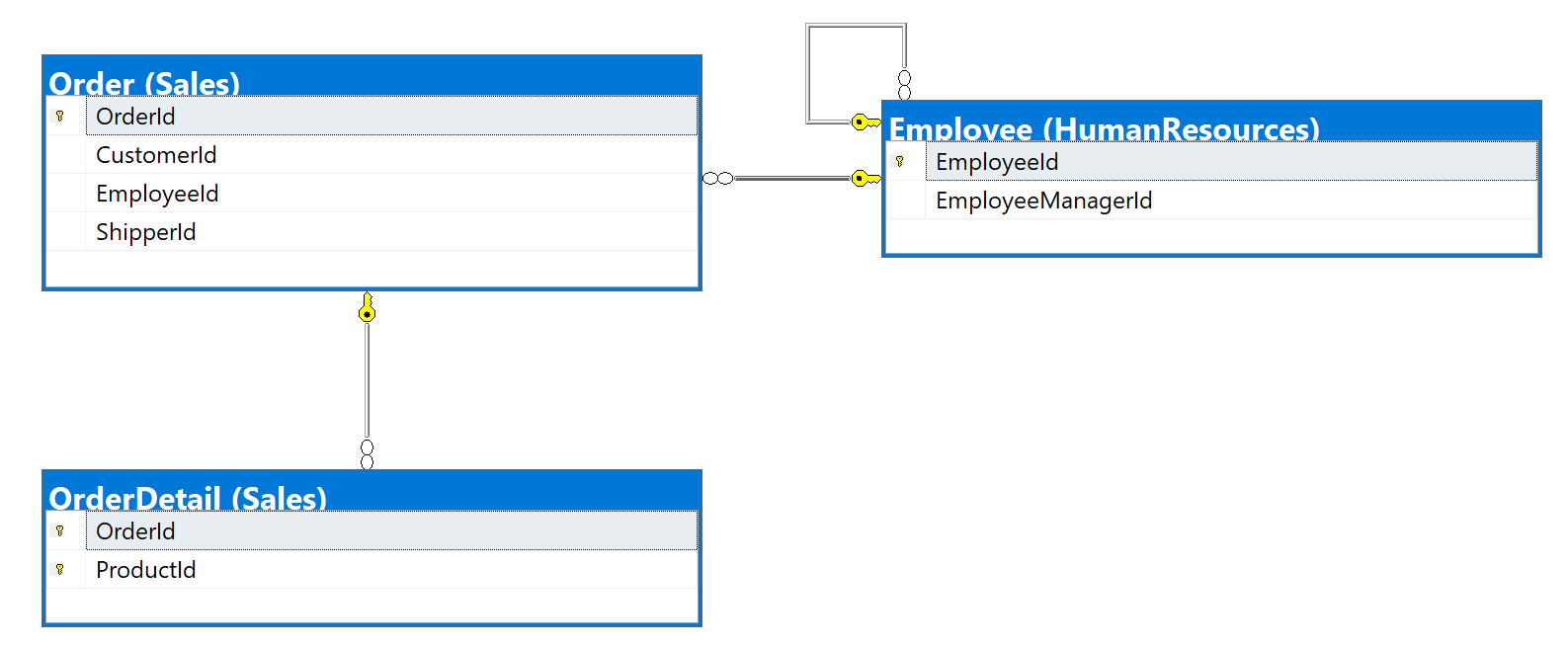
Using a function to concatenate each employee’s first and last name, count the number of productids for each order, and then count the orderids and sum the counted products.

## Database

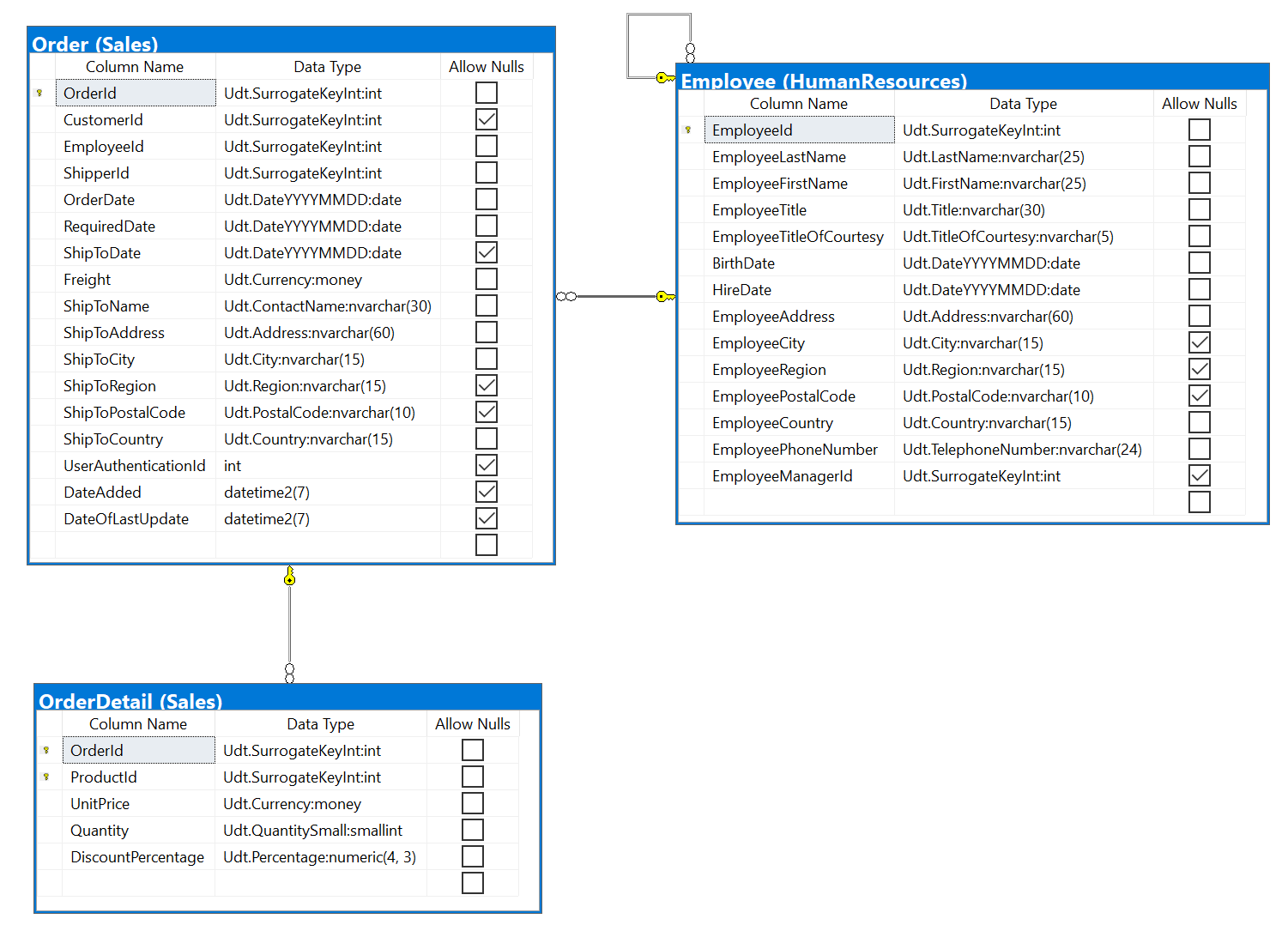
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Derived Column | Name  Orders  Products |
| Order | OrderId  OrderDate  ShipToDate  CustomerId  Freight |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.ConcatName;

GO

CREATE FUNCTION dbo.ConcatName (

@FirstName NVARCHAR(30)

,@LastName NVARCHAR(30)

)

RETURNS NVARCHAR(62)

AS

BEGIN

DECLARE @Result NVARCHAR(62);

SELECT @Result = CASE

WHEN @FirstName IS NULL

OR @LastName IS NULL

THEN 'Error in data'

ELSE CONCAT (

@LastName

,', '

,@FirstName

)

END;

RETURN @Result;

END;

GO

SELECT dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastName) AS [Name]

,COUNT(o.OrderId) AS Orders

,SUM(o.NumOfProducts) AS Products

FROM HumanResources.Employee AS e

INNER JOIN (

SELECT o.OrderId

,o.EmployeeId

,COUNT(od.ProductId) AS NumOfProducts

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

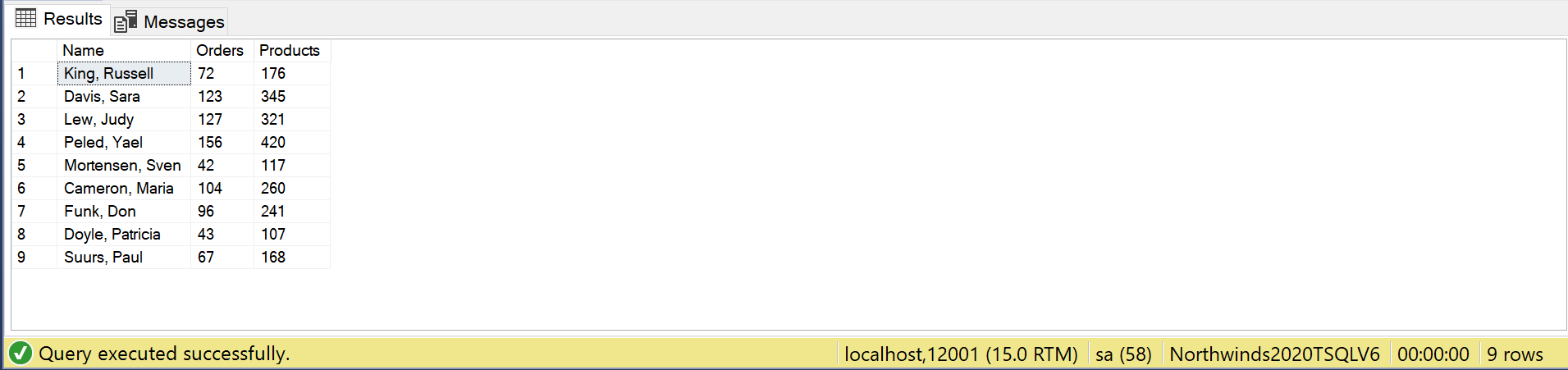
GROUP BY o.OrderId

,o.EmployeeId

) AS o ON e.EmployeeId = o.EmployeeId

GROUP BY dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastName);

## Relational Output with total number of rows returned (9)



## Sample JSON Output with total number of rows returned (9)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.ConcatName;

GO

CREATE FUNCTION dbo.ConcatName (

@FirstName NVARCHAR(30)

,@LastName NVARCHAR(30)

)

RETURNS NVARCHAR(62)

AS

BEGIN

DECLARE @Result NVARCHAR(62);

SELECT @Result = CASE

WHEN @FirstName IS NULL

OR @LastName IS NULL

THEN 'Error in data'

ELSE CONCAT (

@LastName

,', '

,@FirstName

)

END;

RETURN @Result;

END;

GO

SELECT dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastName) AS [Name]

,COUNT(o.OrderId) AS Orders

,SUM(o.NumOfProducts) AS Products

FROM HumanResources.Employee AS e

INNER JOIN (

SELECT o.OrderId

,o.EmployeeId

,COUNT(od.ProductId) AS NumOfProducts

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

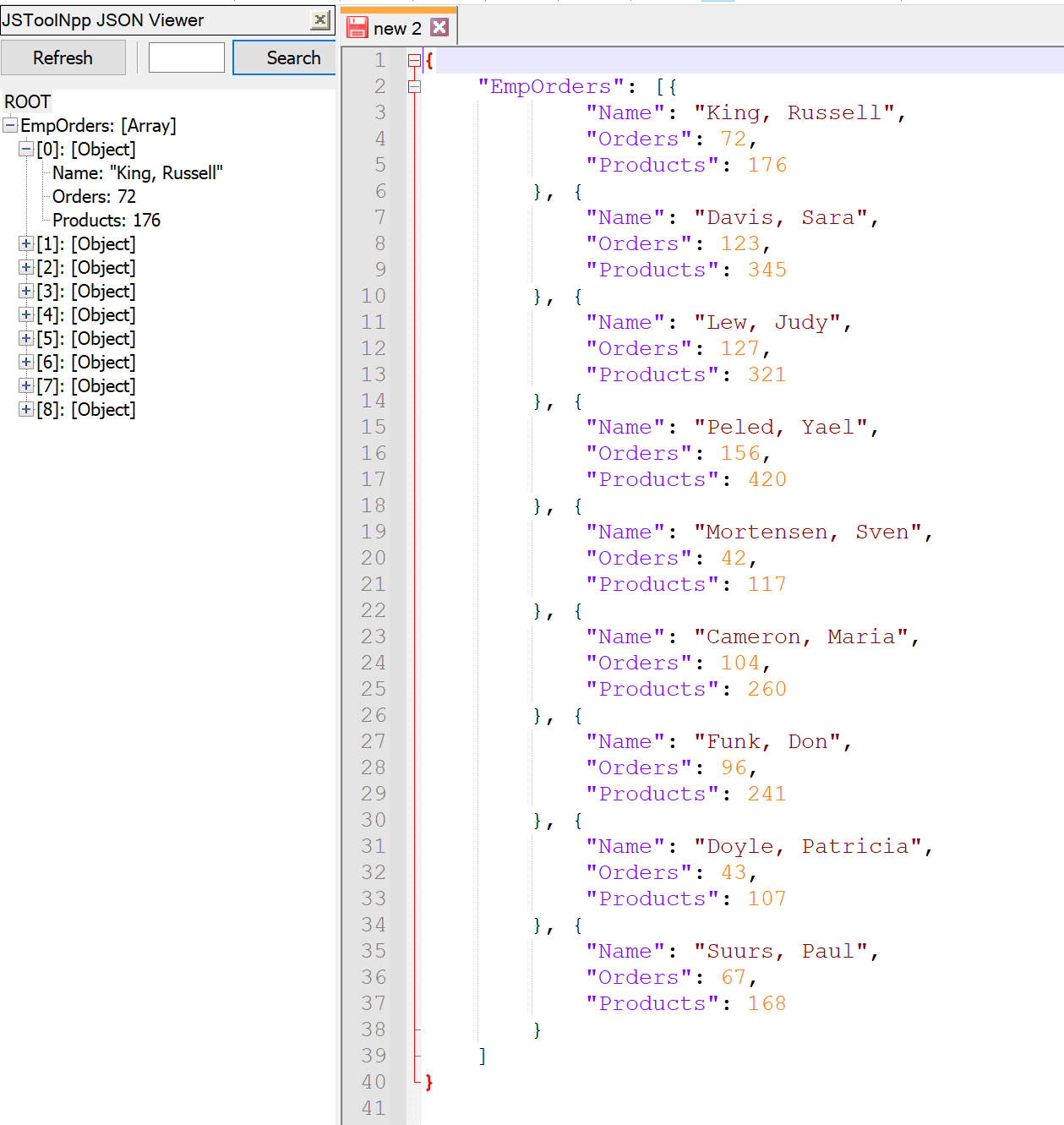
GROUP BY o.OrderId

,o.EmployeeId

) AS o ON e.EmployeeId = o.EmployeeId

GROUP BY dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastName)

FOR JSON PATH, ROOT('EmpOrders'), INCLUDE\_NULL\_VALUES;



# Proposition 16: What are the octal values of geography keys, and how many employees are assigned to their area and how many customers are located there using AdventureWorksDW2017?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

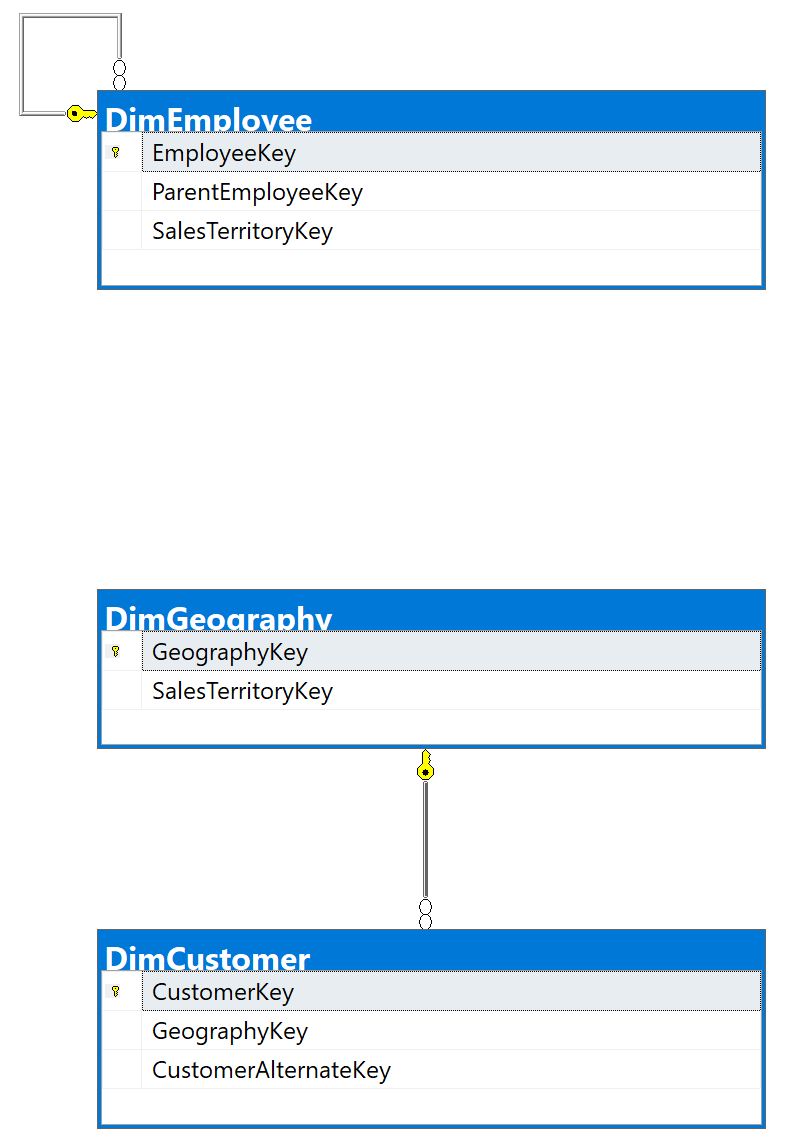
Using a function to convert int to octal, convert the geography keys, and then count the distinct customer and employee keys after joining those tables to the DimGeography table.

## Database

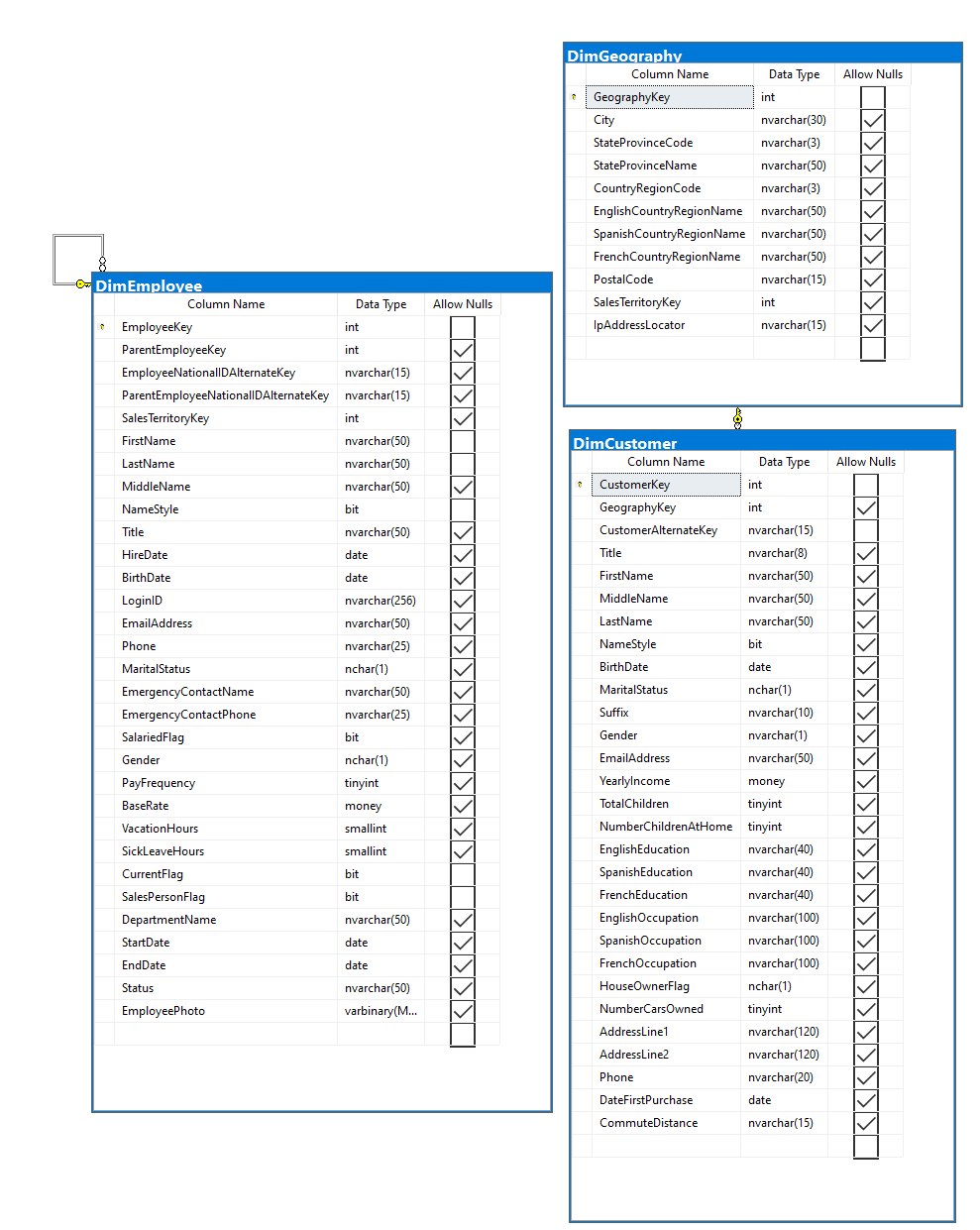
AdventureWorksDW2017

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| DimGeography | EnglishCountryRegionName  StateProvinceName  City  SalesTerritoryKey |
| Derived Column | GeographyKey  NumOfEmployees  NumOfCustomers |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| DimGeography (converted to octal using scalar function) | GeographyKey | DESC |

## Problem solving Query

USE AdventureWorksDW2017;

GO

DROP FUNCTION

IF EXISTS dbo.IntToOct;

GO

CREATE FUNCTION dbo.IntToOct (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @oct VARCHAR(30);

WHILE @num > 0

BEGIN

SET @oct = CONCAT (

@num % 8

,@oct

);

SET @num = @num / 8;

END;

RETURN @oct;

END;

GO

SELECT dbo.IntToOct(g.GeographyKey) AS GeographyKey

,g.EnglishCountryRegionName

,g.StateProvinceName

,g.City

,g.SalesTerritoryKey

,COUNT(DISTINCT c.CustomerKey) AS NumOfCustomers

,COUNT(DISTINCT e.EmployeeKey) AS NumOfEmployees

FROM (

dbo.DimGeography AS g LEFT OUTER JOIN dbo.DimCustomer AS c ON c.GeographyKey = g.GeographyKey

)

LEFT OUTER JOIN dbo.DimEmployee AS e ON g.SalesTerritoryKey = e.SalesTerritoryKey

GROUP BY dbo.IntToOct(g.GeographyKey)

,g.EnglishCountryRegionName

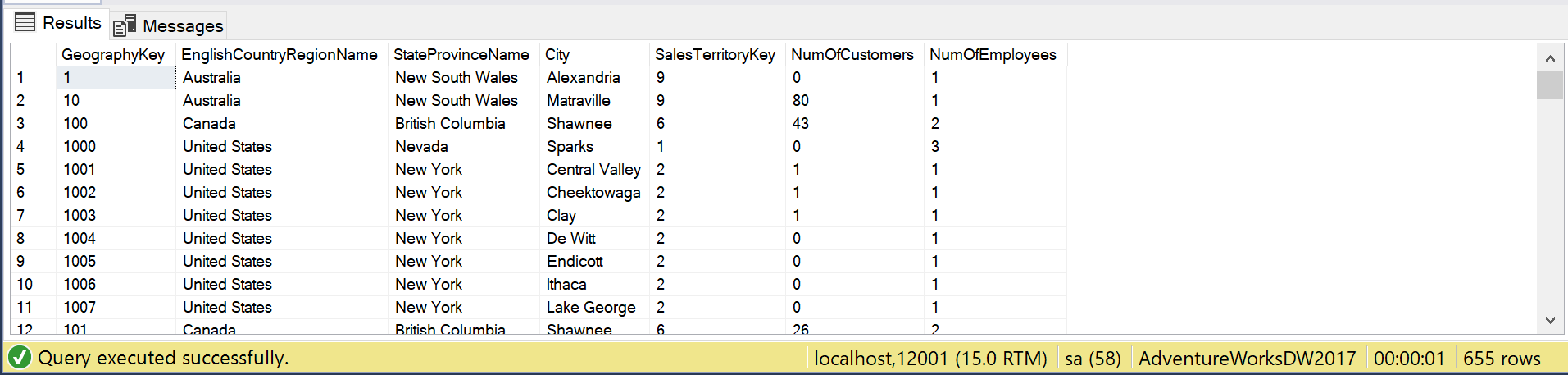
,g.StateProvinceName

,g.City

,g.SalesTerritoryKey

ORDER BY GeographyKey;

## Relational Output with total number of rows returned (655)



## Sample JSON Output with total number of rows returned (655)

USE AdventureWorksDW2017;

GO

DROP FUNCTION

IF EXISTS dbo.IntToOct;

GO

CREATE FUNCTION dbo.IntToOct (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @oct VARCHAR(30);

WHILE @num > 0

BEGIN

SET @oct = CONCAT (

@num % 8

,@oct

);

SET @num = @num / 8;

END;

RETURN @oct;

END;

GO

SELECT dbo.IntToOct(g.GeographyKey) AS GeographyKey

,g.EnglishCountryRegionName

,g.StateProvinceName

,g.City

,g.SalesTerritoryKey

,COUNT(DISTINCT c.CustomerKey) AS NumOfCustomers

,COUNT(DISTINCT e.EmployeeKey) AS NumOfEmployees

FROM (

dbo.DimGeography AS g LEFT OUTER JOIN dbo.DimCustomer AS c ON c.GeographyKey = g.GeographyKey

)

LEFT OUTER JOIN dbo.DimEmployee AS e ON g.SalesTerritoryKey = e.SalesTerritoryKey

GROUP BY dbo.IntToOct(g.GeographyKey)

,g.EnglishCountryRegionName

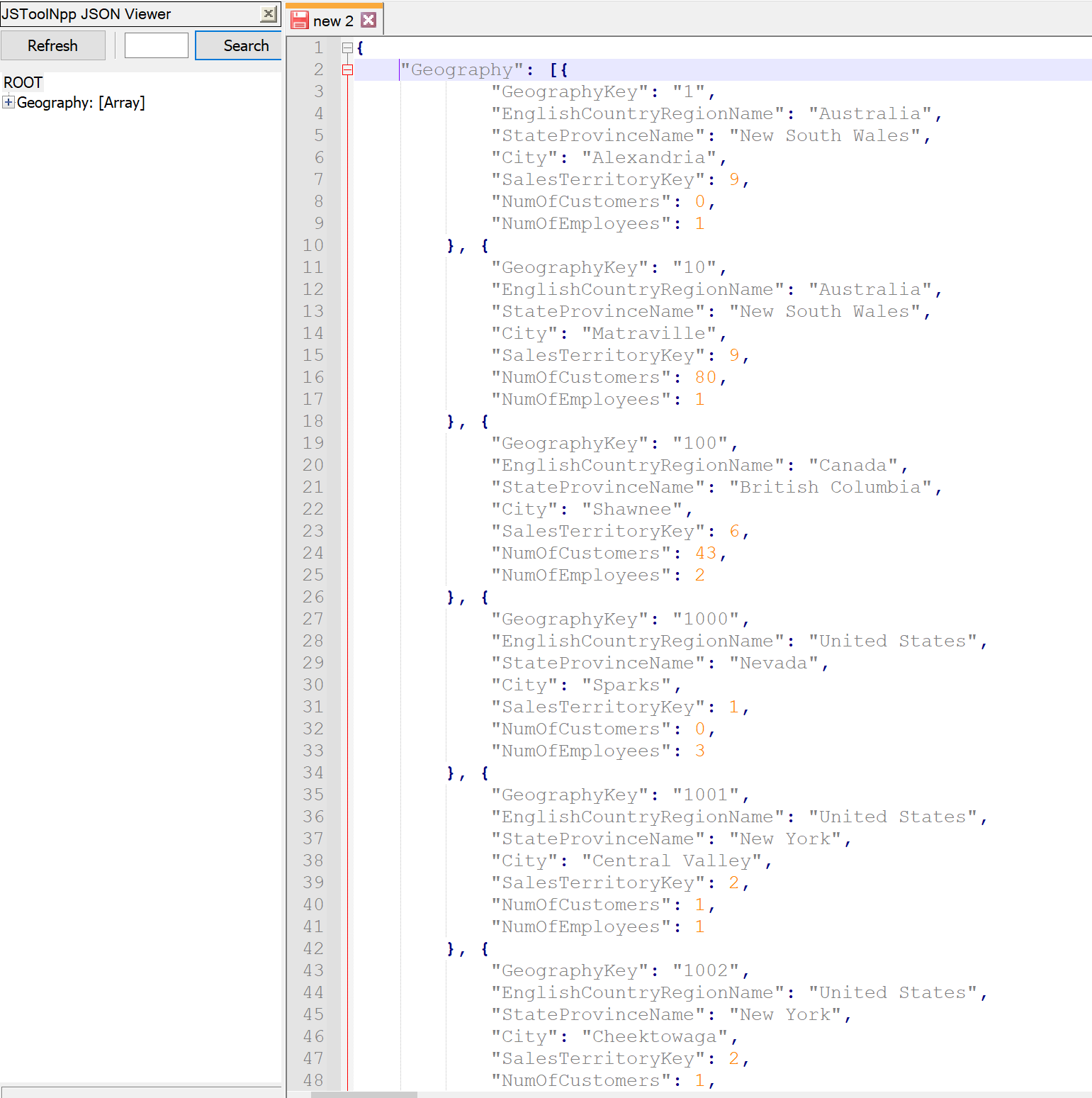
,g.StateProvinceName

,g.City

,g.SalesTerritoryKey

ORDER BY GeographyKey

FOR JSON PATH, ROOT('Geography'), INCLUDE\_NULL\_VALUES;



# Proposition 17: What are the binary values of the employee ids and their manager’s, and how many orders has each employee made using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

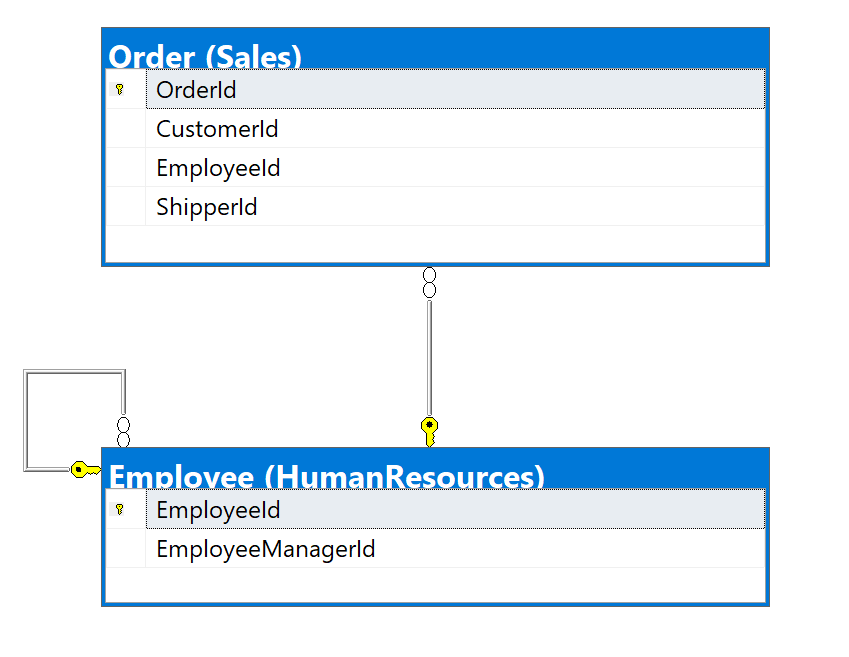
Using a function to convert the employee id to binary, join the employees table to itself to find the manager id, and then count the number of orderids by joining the Order table.

## Database

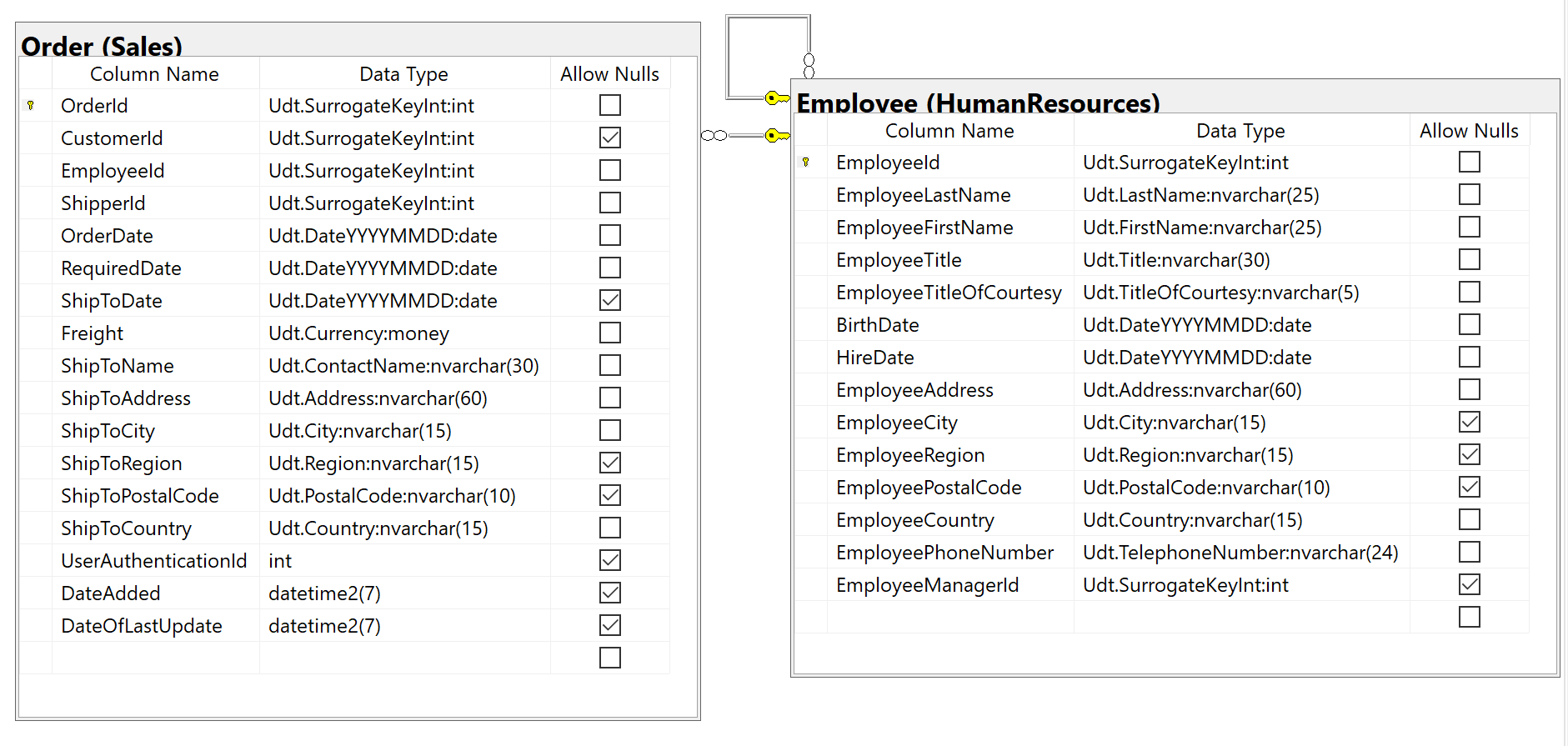
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Employee | EmployeeFirstName  EmployeeLastName |
| Derived Column | EmployeeId  ManagerId  NumOfOrders |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToBin;

GO

CREATE FUNCTION dbo.IntToBin (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @bin VARCHAR(30);

WHILE @num > 0

BEGIN

SET @bin = CONCAT (

@num % 2

,@bin

);

SET @num = @num / 2;

END;

RETURN @bin;

END;

GO

SELECT dbo.IntToBin(e.EmployeeId) AS EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,dbo.IntToBin(m.EmployeeId) AS ManagerId

,COUNT(DISTINCT o.OrderId) AS NumOfOrders

FROM (

HumanResources.Employee AS e LEFT OUTER JOIN HumanResources.Employee AS m ON e.EmployeeManagerId = m.EmployeeId

)

INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

GROUP BY dbo.IntToBin(e.EmployeeId)

,e.EmployeeFirstName

,e.EmployeeLastName

,m.EmployeeId;

## Relational Output with total number of rows returned (9)



## Sample JSON Output with total number of rows returned (9)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToBin;

GO

CREATE FUNCTION dbo.IntToBin (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @bin VARCHAR(30);

WHILE @num > 0

BEGIN

SET @bin = CONCAT (

@num % 2

,@bin

);

SET @num = @num / 2;

END;

RETURN @bin;

END;

GO

SELECT dbo.IntToBin(e.EmployeeId) AS EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,dbo.IntToBin(m.EmployeeId) AS ManagerId

,COUNT(DISTINCT o.OrderId) AS NumOfOrders

FROM (

HumanResources.Employee AS e LEFT OUTER JOIN HumanResources.Employee AS m ON e.EmployeeManagerId = m.EmployeeId

)

INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

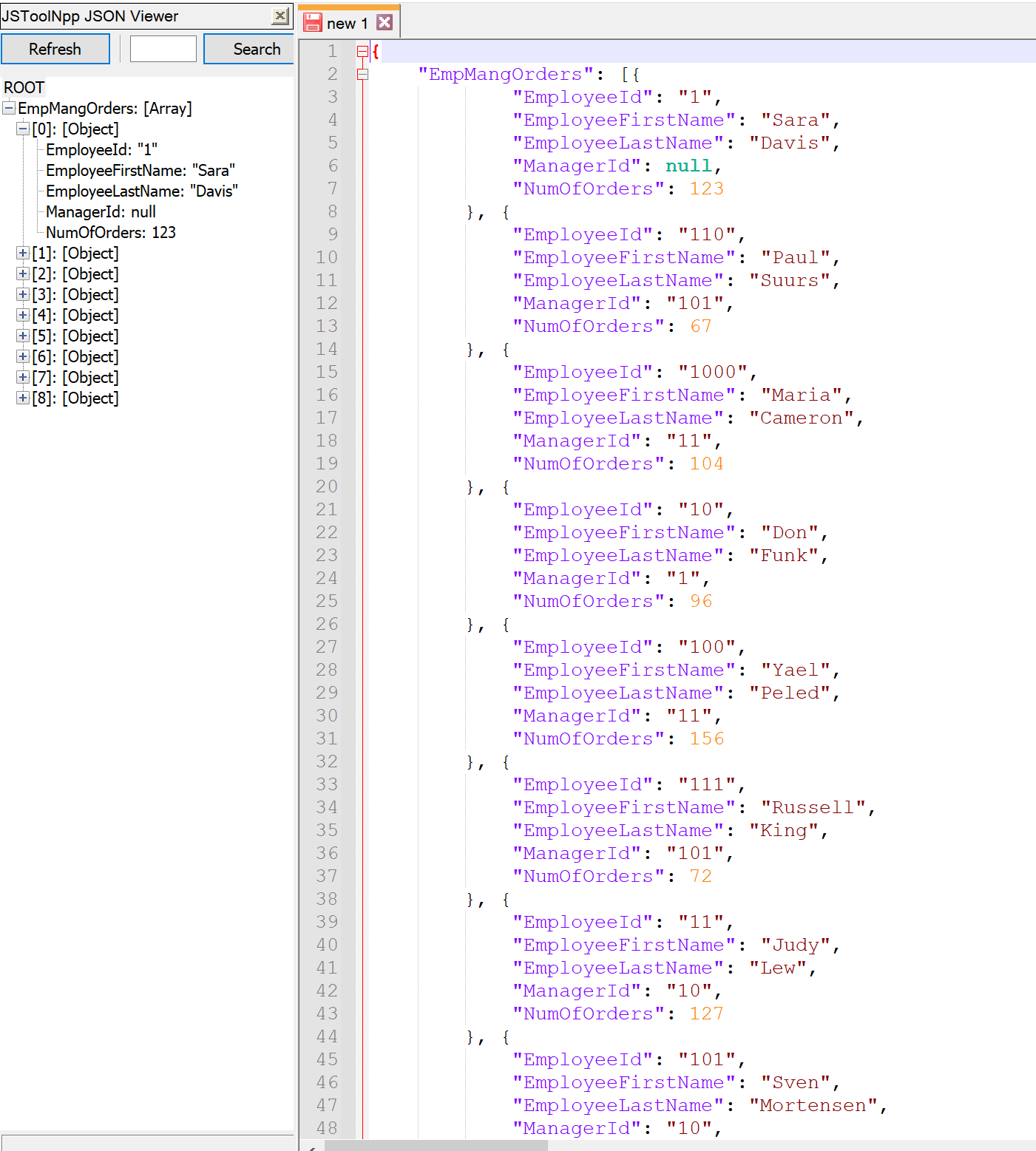
GROUP BY dbo.IntToBin(e.EmployeeId)

,e.EmployeeFirstName

,e.EmployeeLastName

,m.EmployeeId

FOR JSON PATH, ROOT('EmpMangOrders'), INCLUDE\_NULL\_VALUES;



# Proposition 17 Fixed: What are the binary values of the employee ids and their manager’s, and how many orders has each employee made using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

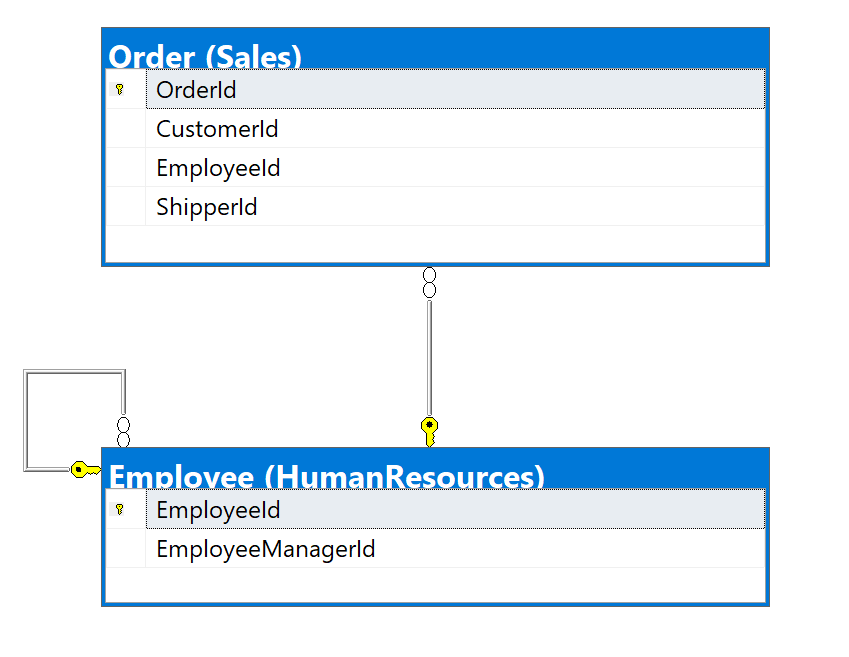
The fix gets rid of the useless self-join of the Employee table. Now the managerid is taken from one instance of the Employee table and the order table is joined twice to get the number of orders for the manager as well as the employee.

## Database

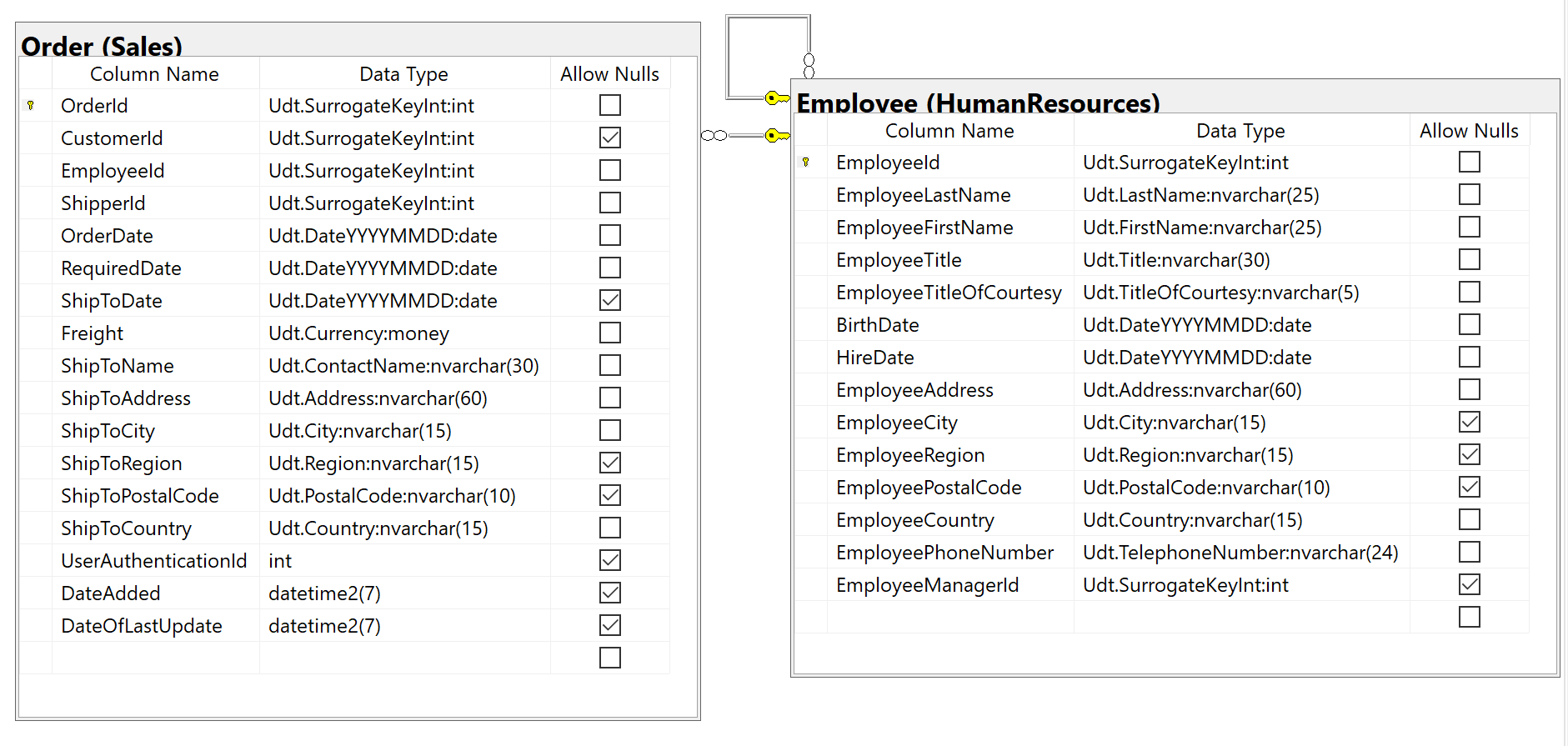
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Employee | EmployeeFirstName  EmployeeLastName |
| Derived Column | EmployeeId  ManagerId  NumOfOrders  ManagerOrders |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToBin;

GO

CREATE FUNCTION dbo.IntToBin (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @bin VARCHAR(30);

WHILE @num > 0

BEGIN

SET @bin = CONCAT (

@num % 2

,@bin

);

SET @num = @num / 2;

END;

RETURN @bin;

END;

GO

SELECT dbo.IntToBin(e.EmployeeId) AS EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,dbo.IntToBin(e.EmployeeManagerId) AS ManagerId

,COUNT(DISTINCT o.OrderId) AS NumOfOrders

,COUNT(DISTINCT om.OrderId) AS ManagerOrders

FROM HumanResources.Employee AS e

INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

LEFT OUTER JOIN Sales.[Order] AS om ON e.EmployeeManagerId = om.EmployeeId

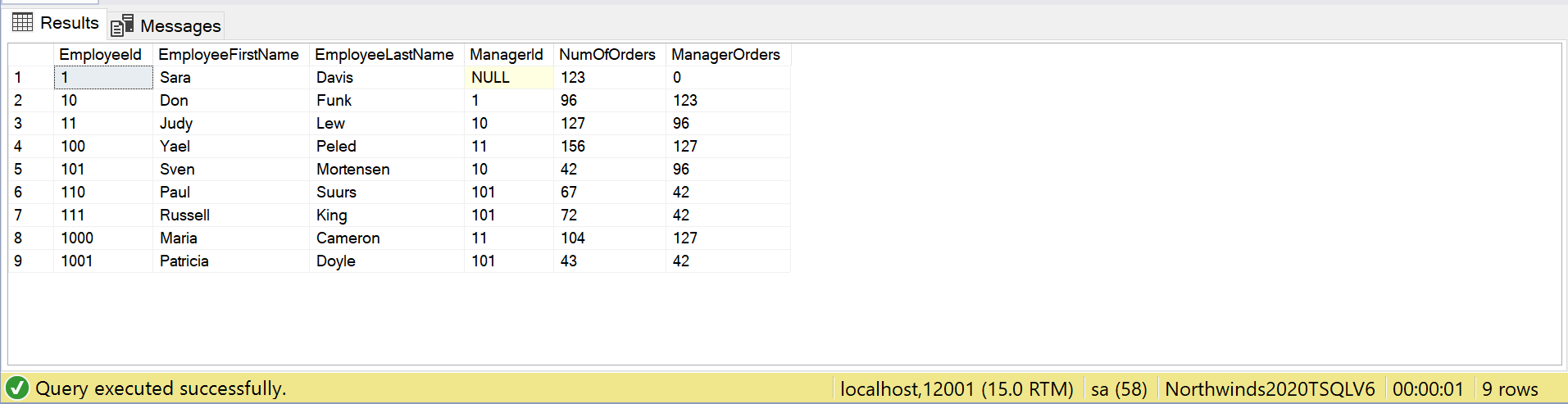
GROUP BY e.EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,e.EmployeeManagerId;

## Relational Output with total number of rows returned (9)



## Sample JSON Output with total number of rows returned (9)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS dbo.IntToBin;

GO

CREATE FUNCTION dbo.IntToBin (@num INT)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @bin VARCHAR(30);

WHILE @num > 0

BEGIN

SET @bin = CONCAT (

@num % 2

,@bin

);

SET @num = @num / 2;

END;

RETURN @bin;

END;

GO

SELECT dbo.IntToBin(e.EmployeeId) AS EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,dbo.IntToBin(e.EmployeeManagerId) AS ManagerId

,COUNT(DISTINCT o.OrderId) AS NumOfOrders

,COUNT(DISTINCT om.OrderId) AS ManagerOrders

FROM HumanResources.Employee AS e

INNER JOIN Sales.[Order] AS o ON e.EmployeeId = o.EmployeeId

LEFT OUTER JOIN Sales.[Order] AS om ON e.EmployeeManagerId = om.EmployeeId

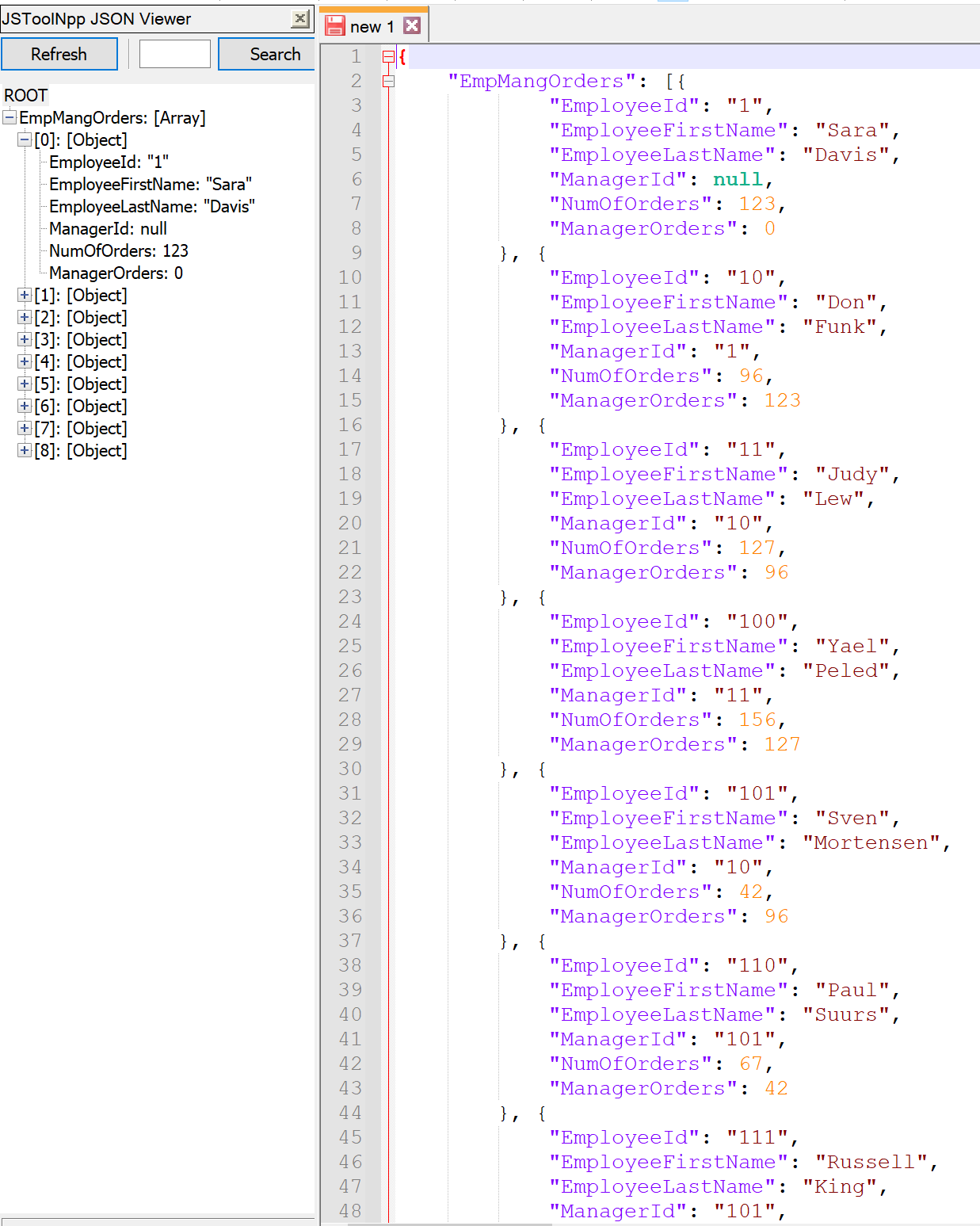
GROUP BY e.EmployeeId

,e.EmployeeFirstName

,e.EmployeeLastName

,e.EmployeeManagerId

FOR JSON PATH, ROOT('EmpMangOrders'), INCLUDE\_NULL\_VALUES;



# Proposition 18: How long ago was the newest order for each customer, and which employee placed the order with them using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

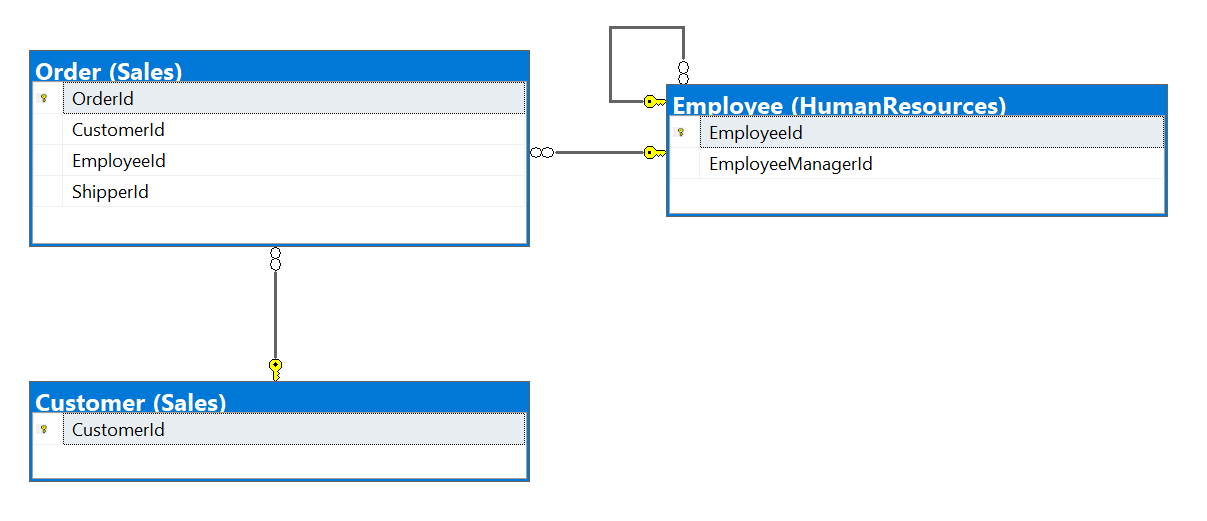
Cross apply the customer table with a subquery that joins the customer and order table in order to apply the function that states how long ago the last order was. Also find the employee associated with that order.

## Database

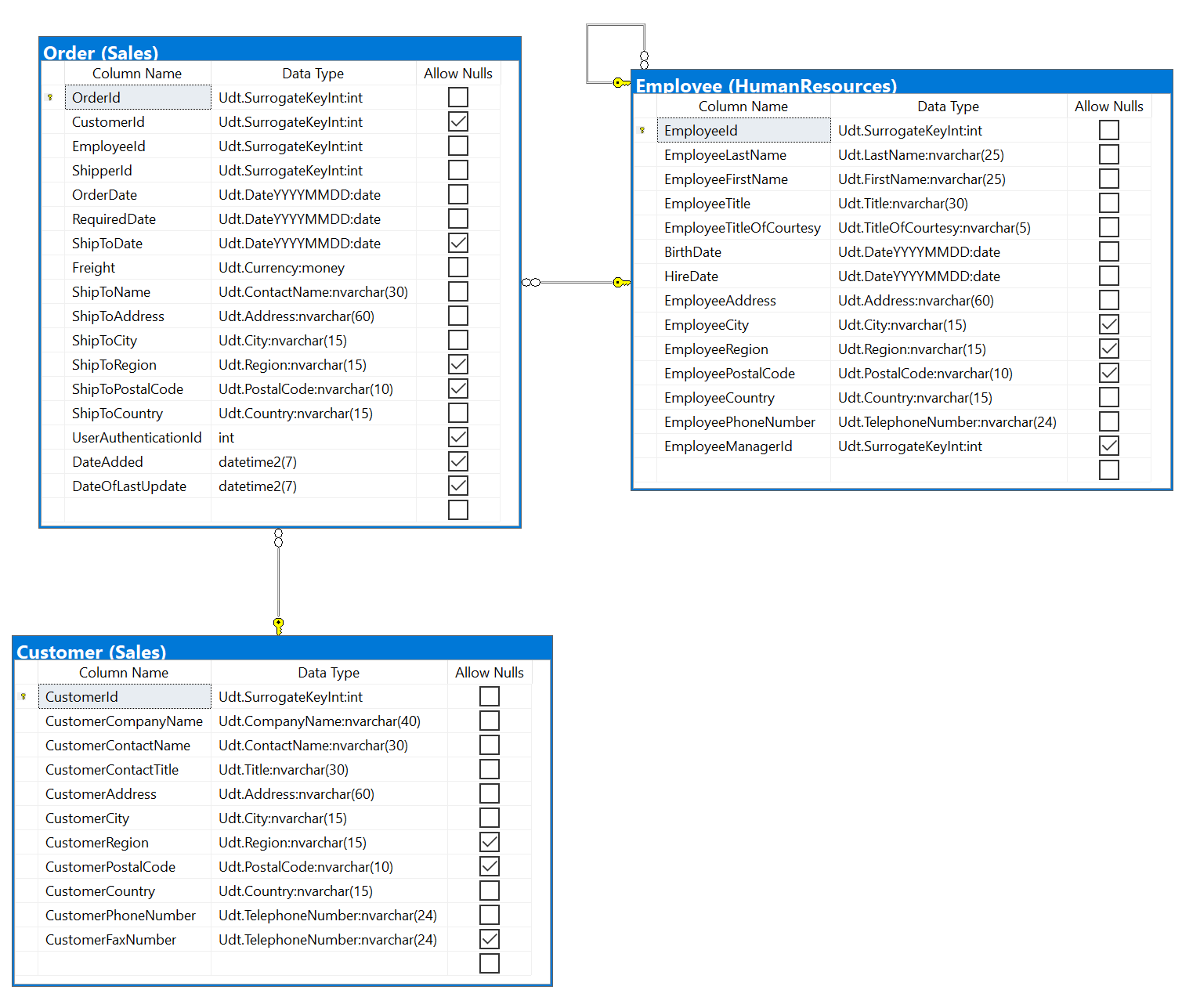
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Derived Column | CustomerId  OrderDate  LastOrder  EmployeeName |
| Order | OrderId |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Customer (from joined subquery) | CustomerId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.YearsAgo;

GO

CREATE FUNCTION Sales.YearsAgo (@date DATE)

RETURNS NVARCHAR(40)

AS

BEGIN

DECLARE @yeardiff INT = DATEDIFF(YEAR, @date, SYSDATETIME());

RETURN CASE

WHEN @yeardiff = 0

THEN 'The last order was this year'

WHEN @yeardiff = 1

THEN 'The last order was 1 year ago'

ELSE CONCAT (

'The last order was '

,@yeardiff

,' years ago'

)

END;

END;

GO

SELECT c.CustomerId

,c.OrderDate

,Sales.YearsAgo(c.OrderDate) AS LastOrder

,o.OrderId

,dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastname) AS EmployeeName

FROM Sales.Customer AS s

CROSS APPLY (

SELECT c.CustomerId

,MAX(o.OrderDate) AS OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = s.CustomerId

AND c.CustomerId = o.CustomerId

GROUP BY c.CustomerId

) AS c

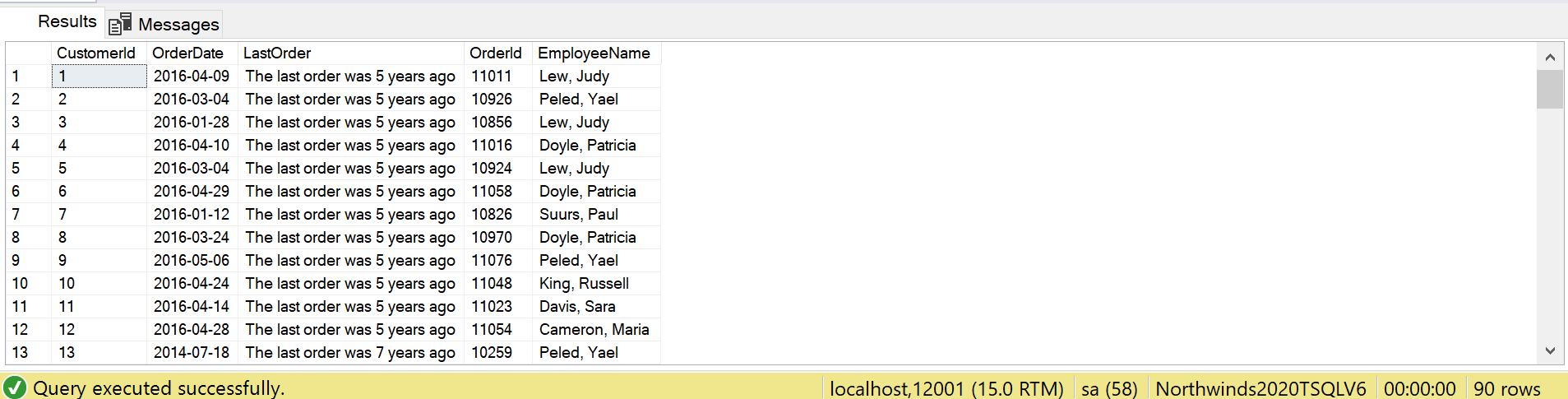
INNER JOIN Sales.[Order] AS o ON c.OrderDate = o.OrderDate

AND c.CustomerId = o.CustomerId

INNER JOIN HumanResources.Employee AS e ON o.EmployeeId = e.EmployeeId

ORDER BY c.CustomerId;

## Relational Output with total number of rows returned (90)



## Sample JSON Output with total number of rows returned (90)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.YearsAgo;

GO

CREATE FUNCTION Sales.YearsAgo (@date DATE)

RETURNS NVARCHAR(40)

AS

BEGIN

DECLARE @yeardiff INT = DATEDIFF(YEAR, @date, SYSDATETIME());

RETURN CASE

WHEN @yeardiff = 0

THEN 'The last order was this year'

WHEN @yeardiff = 1

THEN 'The last order was 1 year ago'

ELSE CONCAT (

'The last order was '

,@yeardiff

,' years ago'

)

END;

END;

GO

SELECT c.CustomerId

,c.OrderDate

,Sales.YearsAgo(c.OrderDate) AS LastOrder

,o.OrderId

,dbo.ConcatName(e.EmployeeFirstName, e.EmployeeLastname) AS EmployeeName

FROM Sales.Customer AS s

CROSS APPLY (

SELECT c.CustomerId

,MAX(o.OrderDate) AS OrderDate

FROM Sales.Customer AS c

INNER JOIN Sales.[Order] AS o ON c.CustomerId = s.CustomerId

AND c.CustomerId = o.CustomerId

GROUP BY c.CustomerId

) AS c

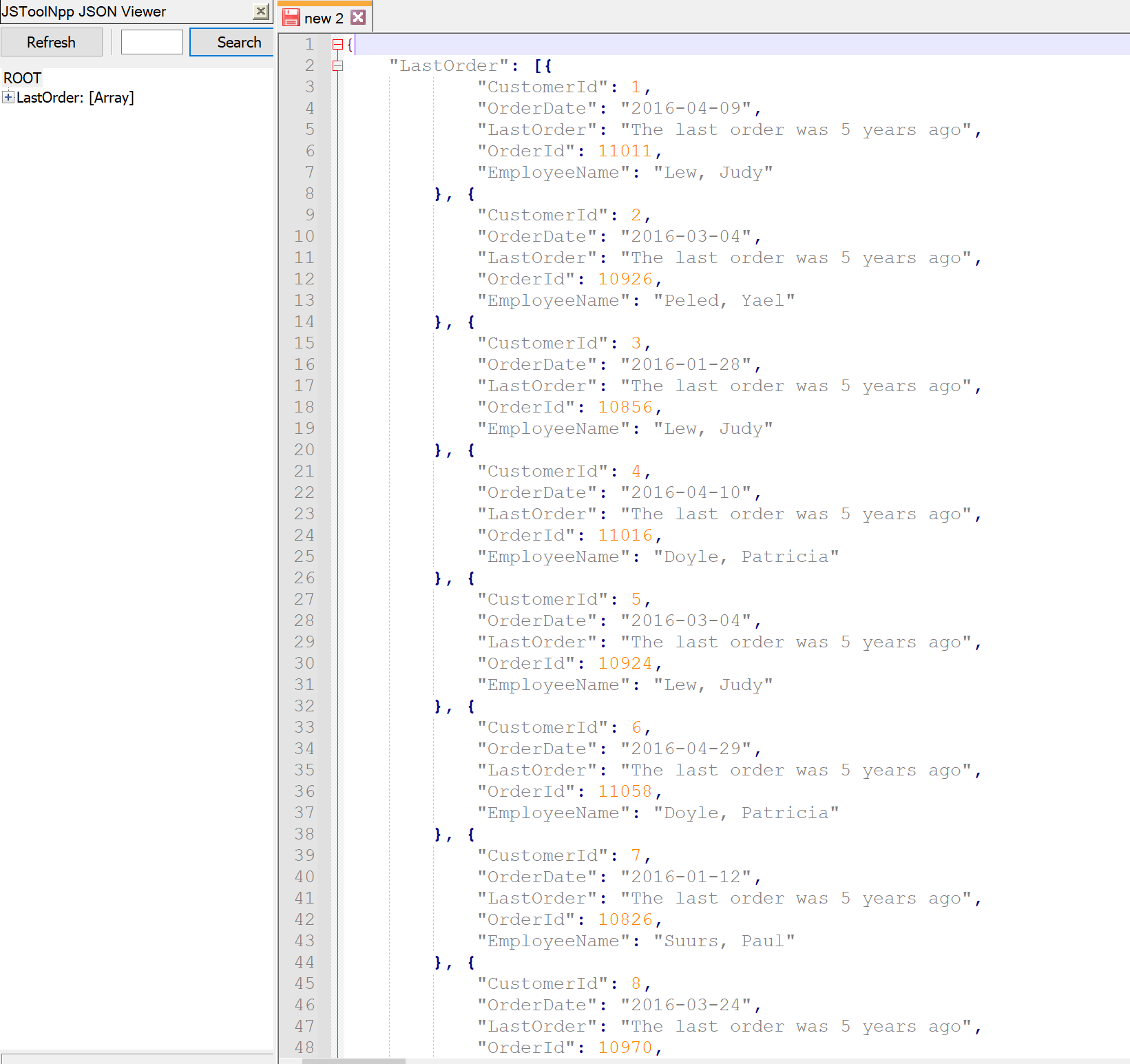
INNER JOIN Sales.[Order] AS o ON c.OrderDate = o.OrderDate

AND c.CustomerId = o.CustomerId

INNER JOIN HumanResources.Employee AS e ON o.EmployeeId = e.EmployeeId

ORDER BY c.CustomerId

FOR JSON PATH, ROOT('LastOrder'), INCLUDE\_NULL\_VALUES;



# Proposition 19: What quart was each order place, and how many items were in the order, what was its cost, and who supplied the items using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

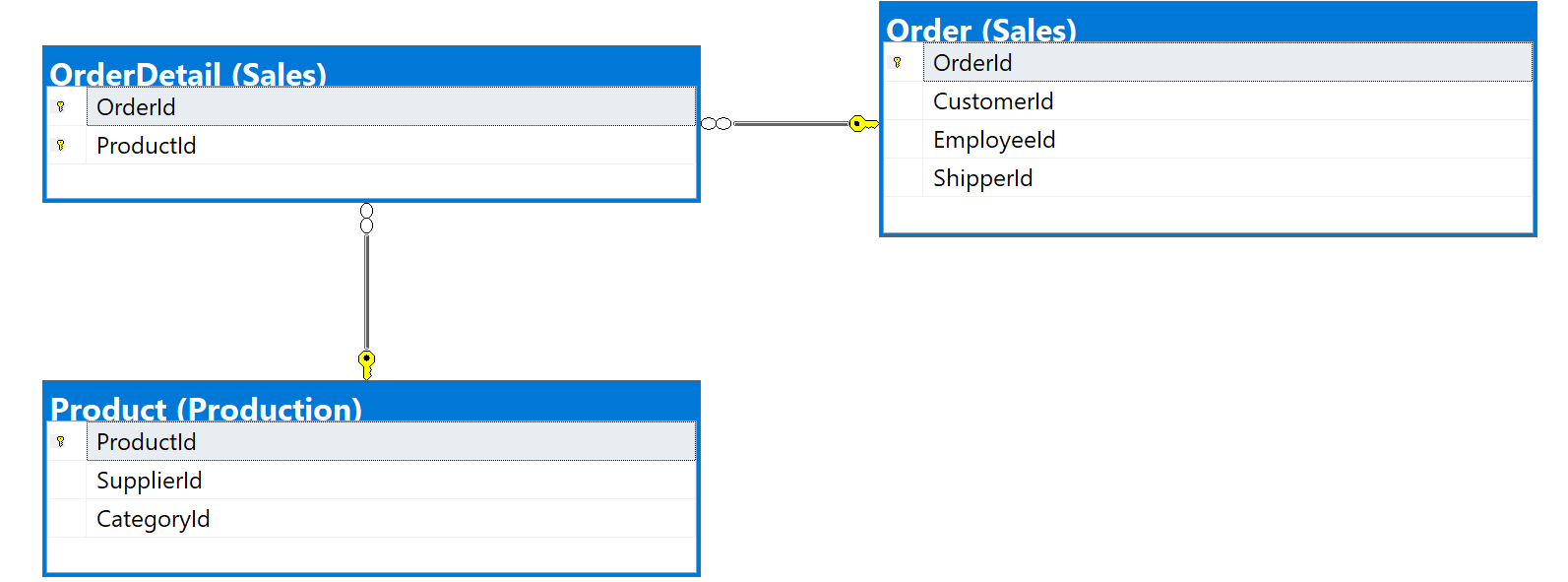
Use a function to find what quarter the order was made, and count the number of products in each order, and aggregate the supplier ids.

## Database

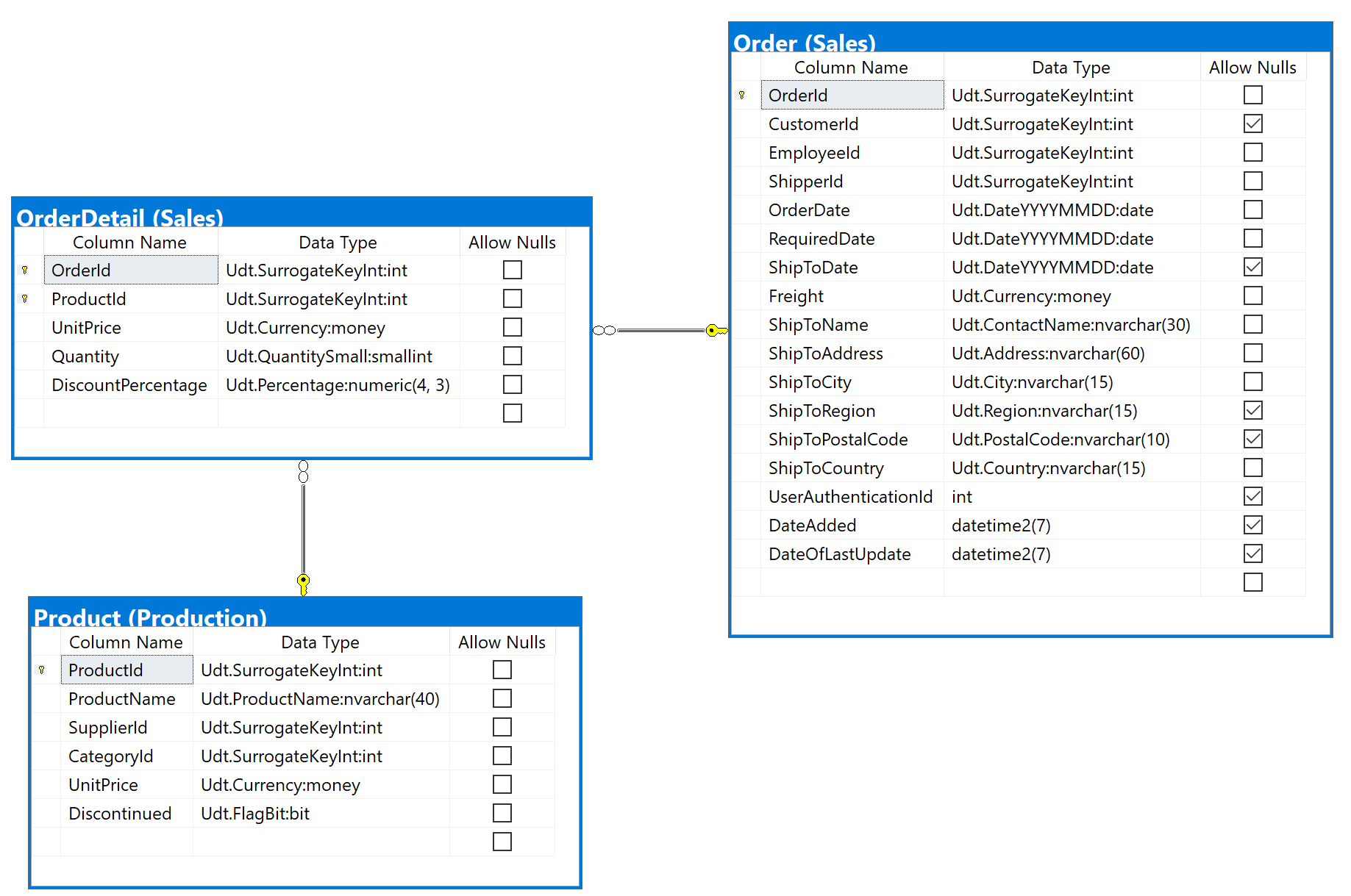
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Derived Column | Qrt  NumOfItems  CostWithoutDiscount  Suppliers |
| Order | OrderId |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Orders | OrderId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.YearQuarter;

GO

CREATE FUNCTION Sales.YearQuarter (@date DATE)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @year INT = YEAR(@date);

DECLARE @month INT = MONTH(@date);

DECLARE @qrt VARCHAR(30) = CASE

WHEN @month BETWEEN 1

AND 3

THEN CONCAT (

'QI-'

,@year

)

WHEN @month BETWEEN 4

AND 6

THEN CONCAT (

'QII-'

,@year

)

WHEN @month BETWEEN 7

AND 9

THEN CONCAT (

'QIII-'

,@year

)

WHEN @month BETWEEN 10

AND 12

THEN CONCAT (

'QIV-'

,@year

)

ELSE 'You got some problems'

END;

RETURN @qrt;

END;

GO

SELECT o.OrderId

,Sales.YearQuarter(o.OrderDate) AS Qtr

,COUNT(od.ProductId) AS NumOfItems

,SUM(od.UnitPrice \* od.Quantity) AS CostWithoutDiscount

,STRING\_AGG(p.SupplierId, ',') AS Suppliers

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

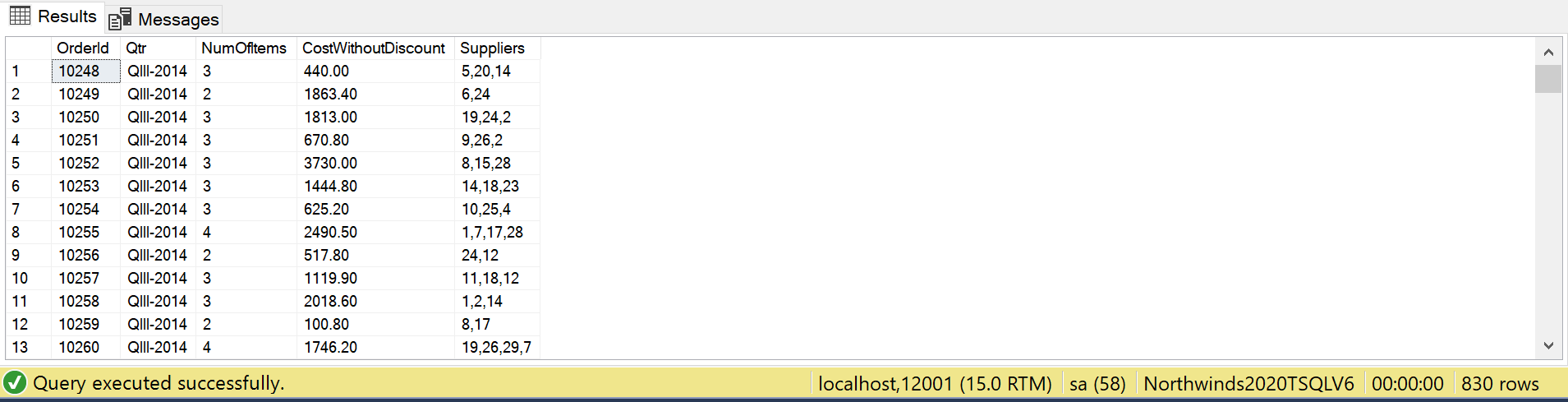
INNER JOIN Production.Product AS p ON od.ProductId = p.ProductId

GROUP BY O.OrderId

,Sales.YearQuarter(o.OrderDate)

ORDER BY o.OrderId;

## Relational Output with total number of rows returned (830)



## Sample JSON Output with total number of rows returned (830)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.YearQuarter;

GO

CREATE FUNCTION Sales.YearQuarter (@date DATE)

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @year INT = YEAR(@date);

DECLARE @month INT = MONTH(@date);

DECLARE @qrt VARCHAR(30) = CASE

WHEN @month BETWEEN 1

AND 3

THEN CONCAT (

'QI-'

,@year

)

WHEN @month BETWEEN 4

AND 6

THEN CONCAT (

'QII-'

,@year

)

WHEN @month BETWEEN 7

AND 9

THEN CONCAT (

'QIII-'

,@year

)

WHEN @month BETWEEN 10

AND 12

THEN CONCAT (

'QIV-'

,@year

)

ELSE 'You got some problems'

END;

RETURN @qrt;

END;

GO

SELECT o.OrderId

,Sales.YearQuarter(o.OrderDate) AS Qtr

,COUNT(od.ProductId) AS NumOfItems

,SUM(od.UnitPrice \* od.Quantity) AS CostWithoutDiscount

,STRING\_AGG(p.SupplierId, ',') AS Suppliers

FROM Sales.[Order] AS o

INNER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

INNER JOIN Production.Product AS p ON od.ProductId = p.ProductId

GROUP BY O.OrderId

,Sales.YearQuarter(o.OrderDate)

ORDER BY o.OrderId

FOR JSON PATH, ROOT('Qrt'), INCLUDE\_NULL\_VALUES;



# Proposition 20: What are the individual orders each customer made and how many items did they get with each order, and are they located where the employee who handled the order is using Northwinds2020TSQLV6?

## Detailed explanation of the problem that will help the developer to write the query to resolve the issue.

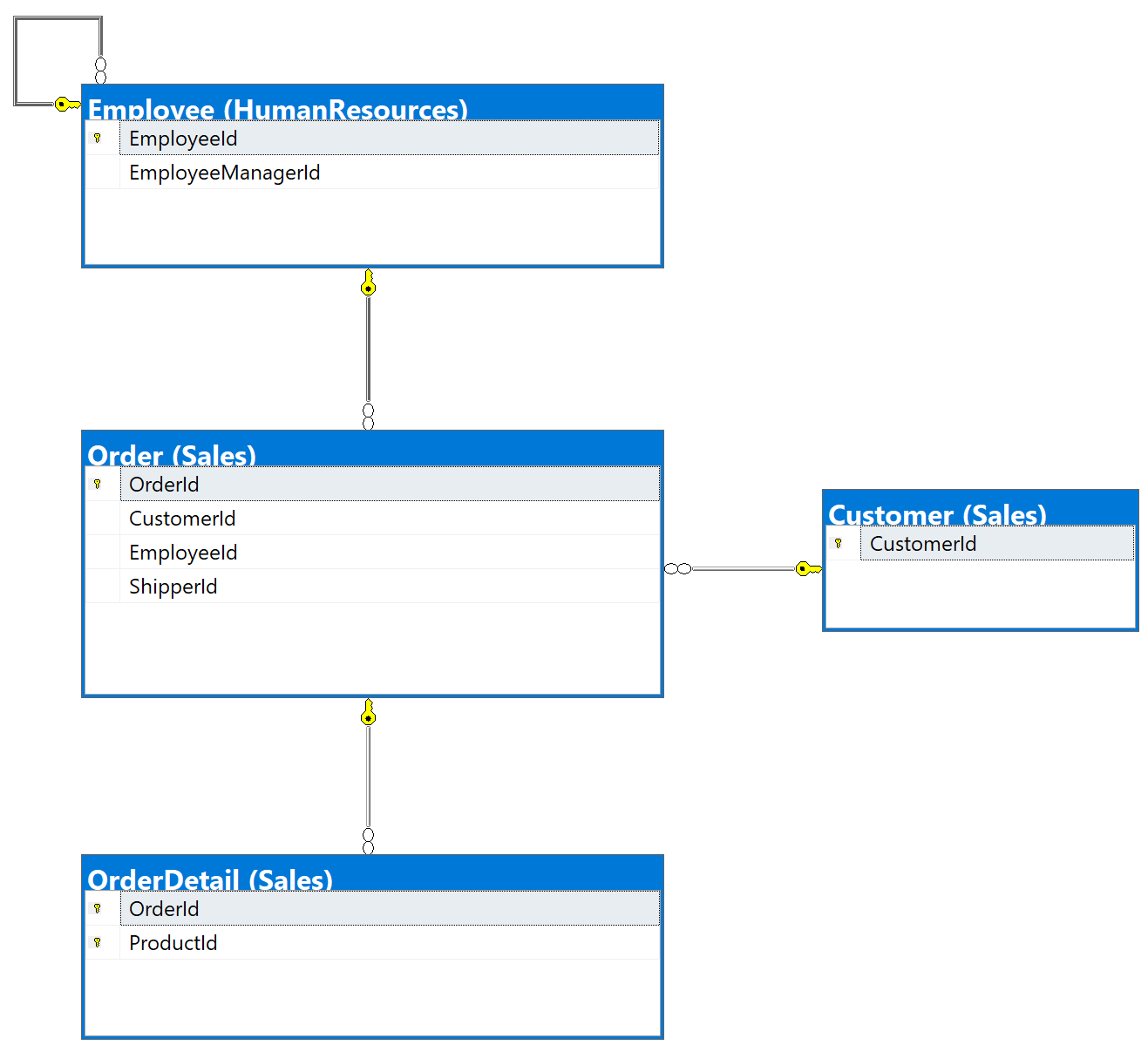
Find the CustomerId, EmployeeId, City, OrderId, and count the number of products. Use a function to determine if the customer and employee are in the same city.

## Database

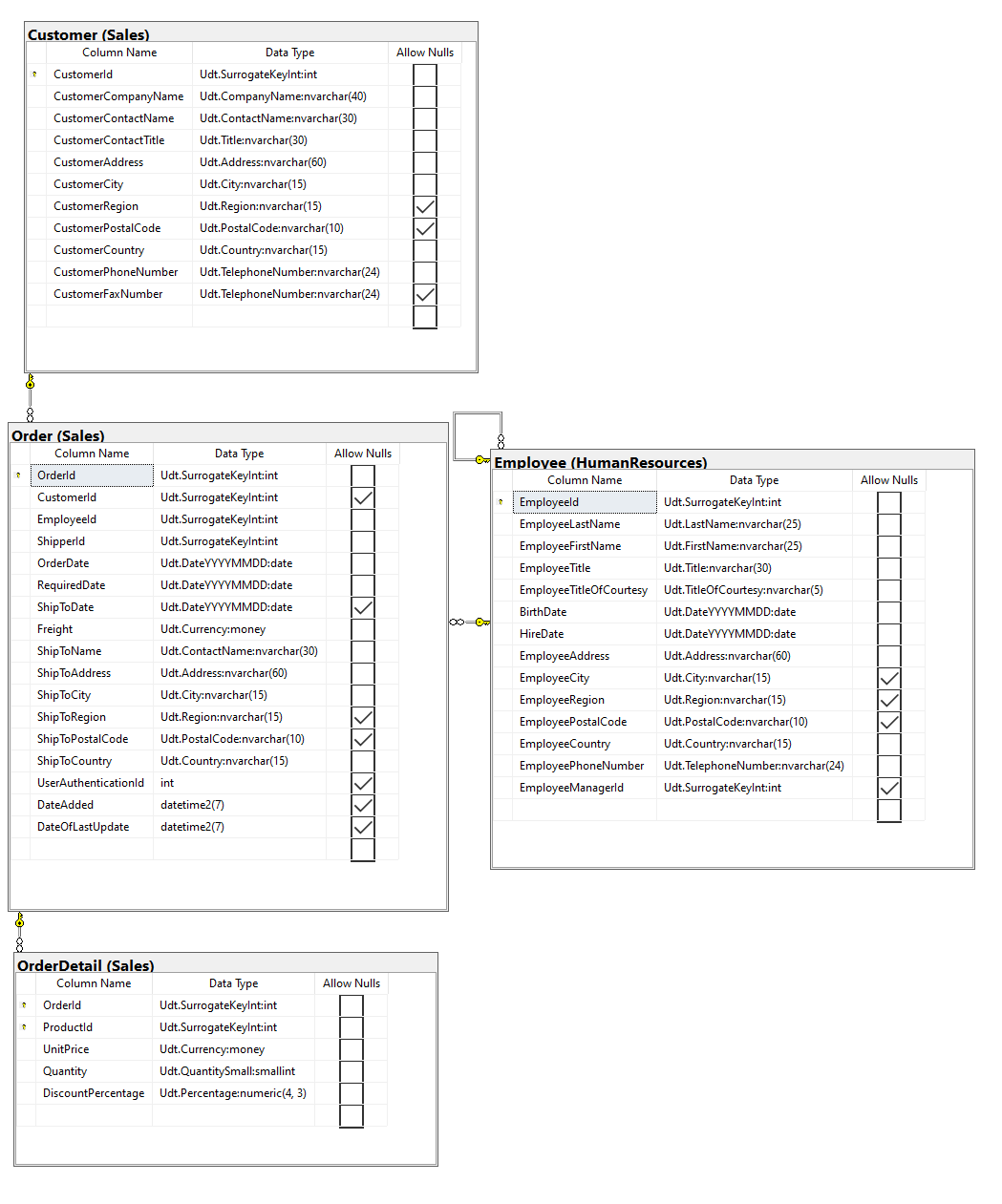
Northwinds2020TSQLV6

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Project following columns from their respective tables in the select clause

|  |  |
| --- | --- |
| Table Name | Column Name |
| Customer | CustomerId  CustomerCity |
| Order | OrderId |
| Employee | EmployeeId |
| Derived Column | Location  NumOfItems |

## Order by

|  |  |  |
| --- | --- | --- |
| Table Name | Column Name | Sort Order |
| Customer | CustomerId | DESC |

## Problem solving Query

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.EmpCustCity;

GO

CREATE FUNCTION Sales.EmpCustCity (

@custCity NVARCHAR(15)

,@empCity NVARCHAR(15)

)

RETURNS NVARCHAR(80)

AS

BEGIN

IF (

@custCity IS NULL

OR @empCity IS NULL

)

RETURN 'Unknown';

IF (@custCity = @empCity)

RETURN 'Employee and Customer are in the same city';

RETURN 'Employee and Customer are not in the same city';

END;

GO

SELECT c.CustomerId

,e.EmployeeId

,c.CustomerCity

,Sales.EmpCustCity(c.CustomerCity, e.EmployeeCity) AS [Location]

,o.OrderId

,COUNT(od.ProductId) AS NumofItems

FROM Sales.Customer AS c

LEFT OUTER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

LEFT OUTER JOIN HumanResources.Employee AS e ON o.EmployeeId = e.EmployeeId

LEFT OUTER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

GROUP BY c.CustomerId

,e.EmployeeId

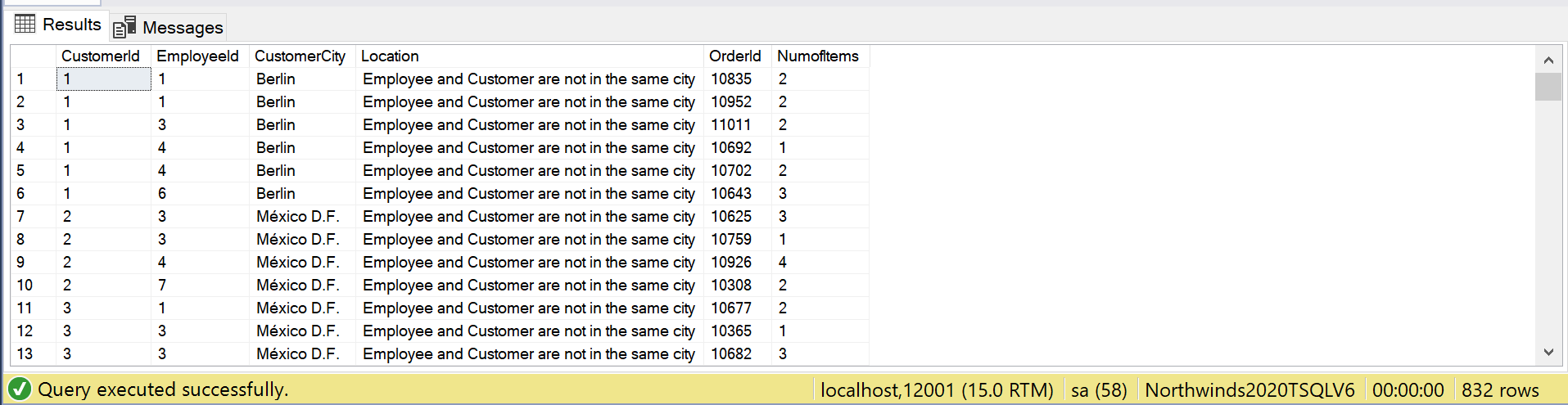
,c.CustomerCity

,Sales.EmpCustCity(c.CustomerCity, e.EmployeeCity)

,o.OrderId

ORDER BY c.CustomerId;

## Relational Output with total number of rows returned (832)



## Sample JSON Output with total number of rows returned (832)

USE Northwinds2020TSQLV6;

GO

DROP FUNCTION

IF EXISTS Sales.EmpCustCity;

GO

CREATE FUNCTION Sales.EmpCustCity (

@custCity NVARCHAR(15)

,@empCity NVARCHAR(15)

)

RETURNS NVARCHAR(80)

AS

BEGIN

IF (

@custCity IS NULL

OR @empCity IS NULL

)

RETURN 'Unknown';

IF (@custCity = @empCity)

RETURN 'Employee and Customer are in the same city';

RETURN 'Employee and Customer are not in the same city';

END;

GO

SELECT c.CustomerId

,e.EmployeeId

,c.CustomerCity

,Sales.EmpCustCity(c.CustomerCity, e.EmployeeCity) AS [Location]

,o.OrderId

,COUNT(od.ProductId) AS NumofItems

FROM Sales.Customer AS c

LEFT OUTER JOIN Sales.[Order] AS o ON c.CustomerId = o.CustomerId

LEFT OUTER JOIN HumanResources.Employee AS e ON o.EmployeeId = e.EmployeeId

LEFT OUTER JOIN Sales.OrderDetail AS od ON o.OrderId = od.OrderId

GROUP BY c.CustomerId

,e.EmployeeId

,c.CustomerCity

,Sales.EmpCustCity(c.CustomerCity, e.EmployeeCity)

,o.OrderId

ORDER BY c.CustomerId

FOR JSON PATH, ROOT('CustEmpCity'), INCLUDE\_NULL\_VALUES;

