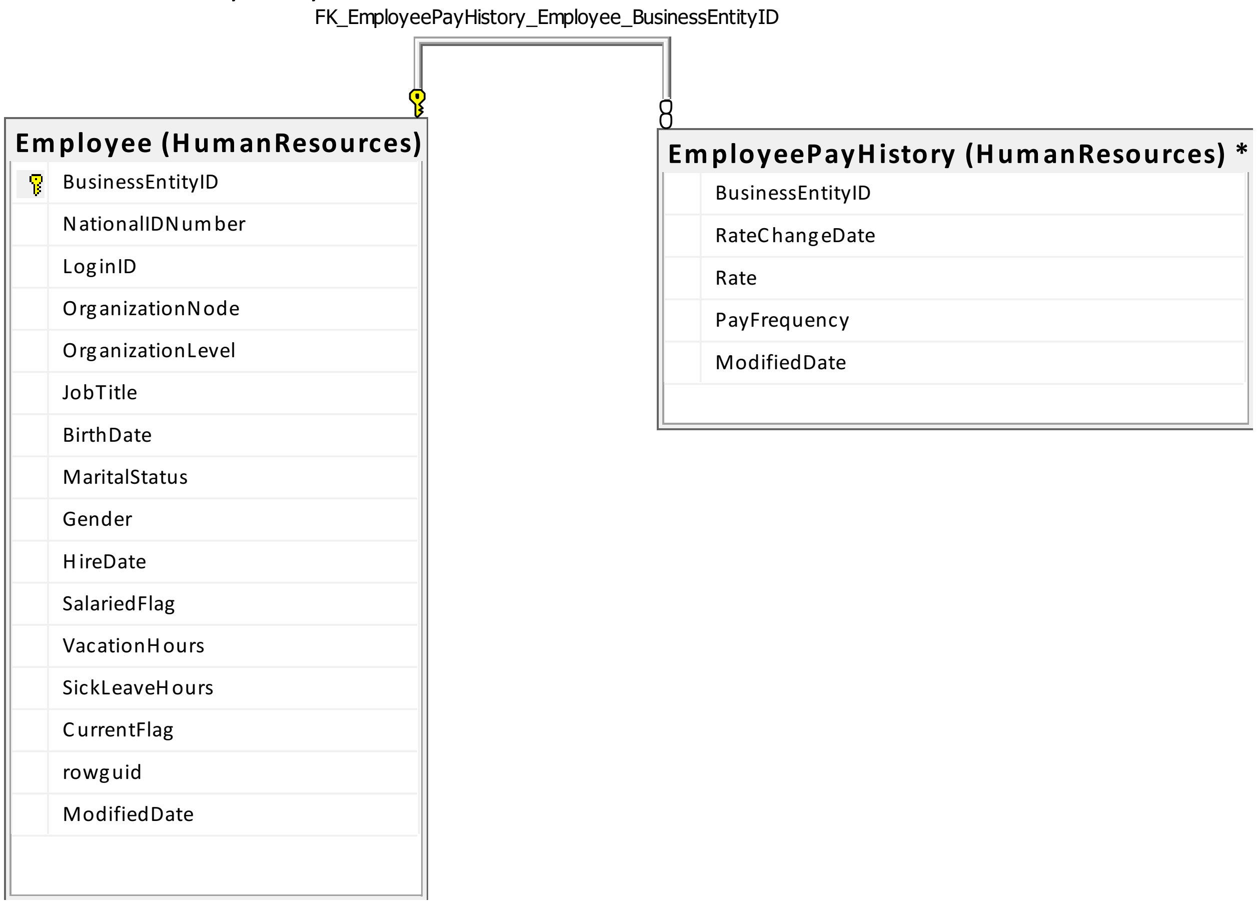
Darshan Patel
Project 1

CS331
Professor Heller

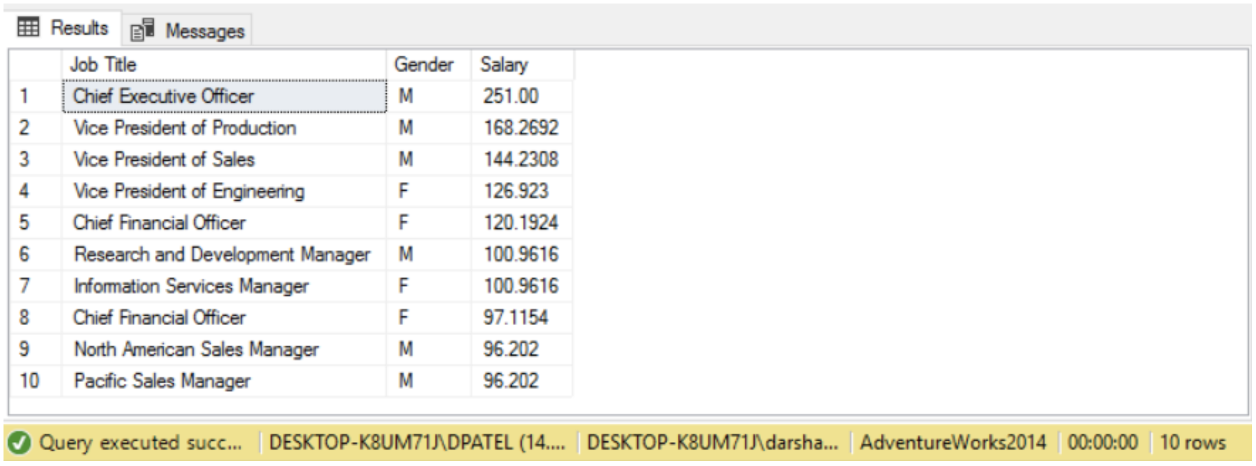
Question 1 (simple): Using AdventureWorks2014, list the top 10 paid employees' job title, gender and salary in descending order of their salary. This query will be used to examine the top paid employees and see if there is a wage gap among different genders.

```
USE AdventureWorks2014;
SELECT TOP 10 E.JobTitle as [Job Title], E.Gender, (P.Rate * P.PayFrequency) as Salary
FROM HumanResources.Employee as E
        INNER JOIN HumanResources.EmployeePayHistory as P
                ON E.BusinessEntityID = P.BusinessEntityID
ORDER BY Salary DESC;
```

Database Diagram: The Employee column and EmployeePayHistory column are joined using the BusinessEntityID key.



Output Table:



	Job Title	Gender	Salary
1	Chief Executive Officer	M	251.00
2	Vice President of Production	M	168.2692
3	Vice President of Sales	M	144.2308
4	Vice President of Engineering	F	126.923
5	Chief Financial Officer	F	120.1924
6	Research and Development Manager	M	100.9616
7	Information Services Manager	F	100.9616
8	Chief Financial Officer	F	97.1154
9	North American Sales Manager	M	96.202
10	Pacific Sales Manager	M	96.202

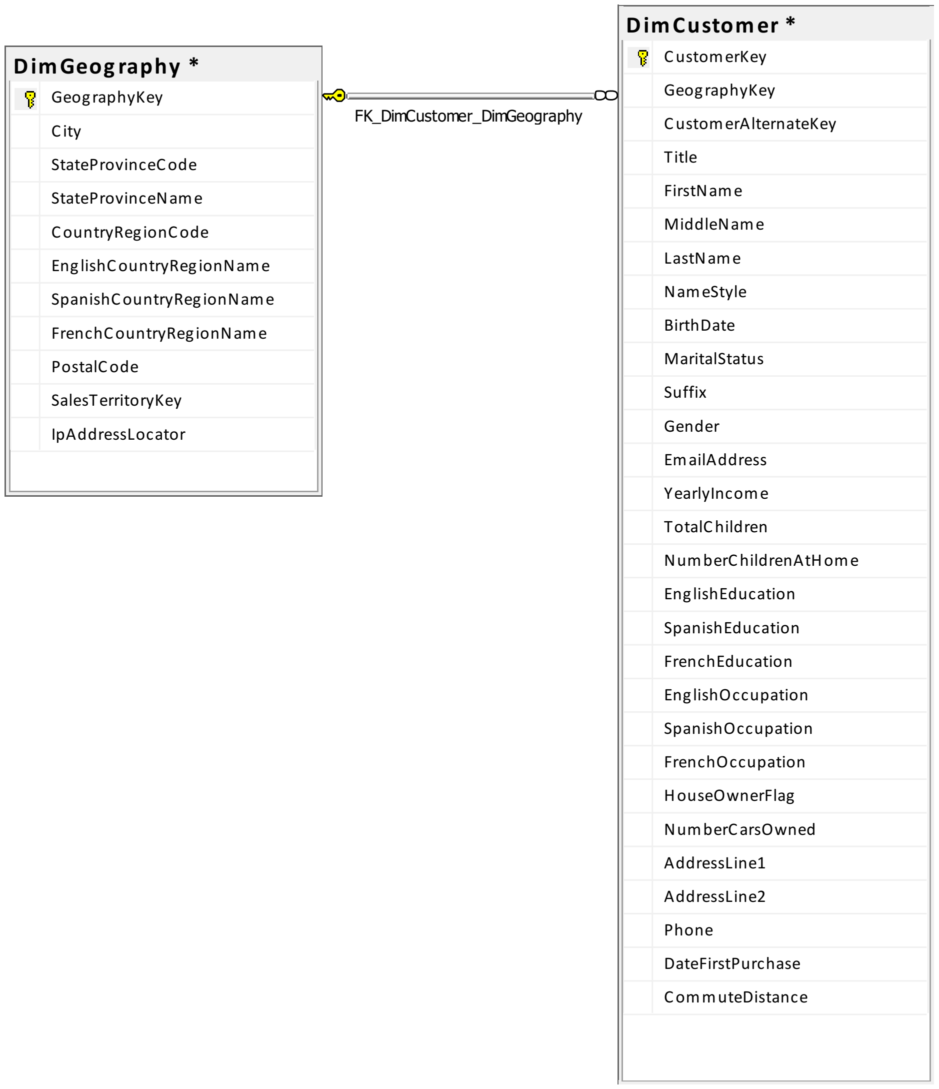
```
Query executed succ... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorks2014 | 00:00:00 | 10 rows
```

Conclusion: Of the top 10 employees, 6 are male and 4 are female. There is a huge wage gap between the first male and female employees, a bit greater than $100,000.

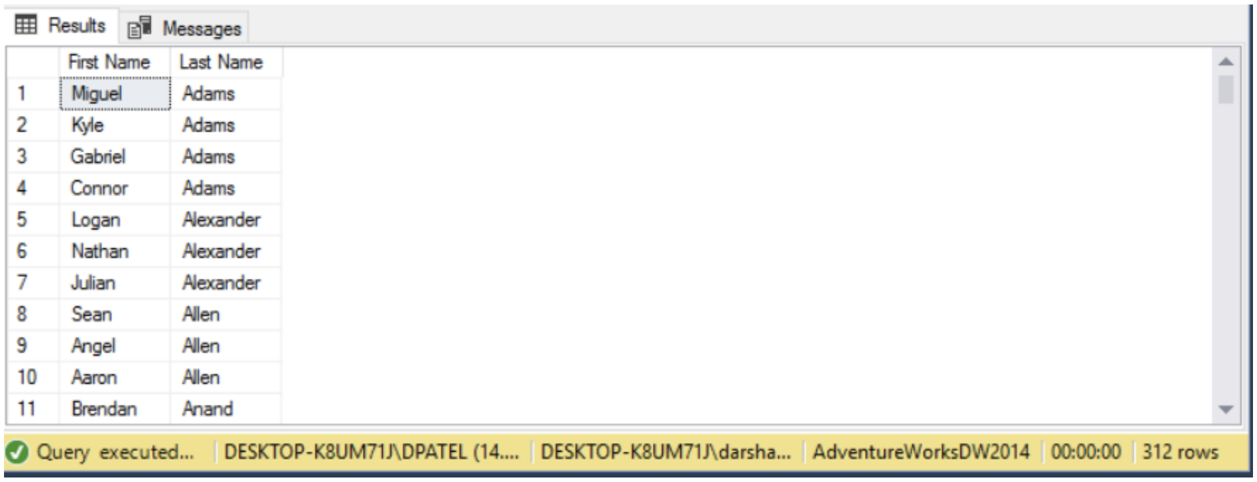
Question 2 (simple): Using AdventureWorksDW2014, list all the single male customers' full name from British Columbia in order of their last name. This query will be used to determine the number of single male customers in British Columbia.

```
USE AdventureWorksDW2014;
SELECT C.FirstName as [First Name], C.LastName as [Last Name]
FROM dbo.DimCustomer as C
            INNER JOIN dbo.DimGeography as G
                      ON C.GeographyKey = G.GeographyKey
WHERE C.gender = N'M' and C.MaritalStatus = N'S'
             and G.StateProvinceName = N'British Columbia'
ORDER BY C.LastName;
```

Database Diagram: The Customer column and Geography column are joined using the Geography key.



Output Table:


Results | Messages

	First Name	Last Name
1	Miguel	Adams
2	Kyle	Adams
3	Gabriel	Adams
4	Connor	Adams
5	Logan	Alexander
6	Nathan	Alexander
7	Julian	Alexander
8	Sean	Allen
9	Angel	Allen
10	Aaron	Allen
11	Brendan	Anand

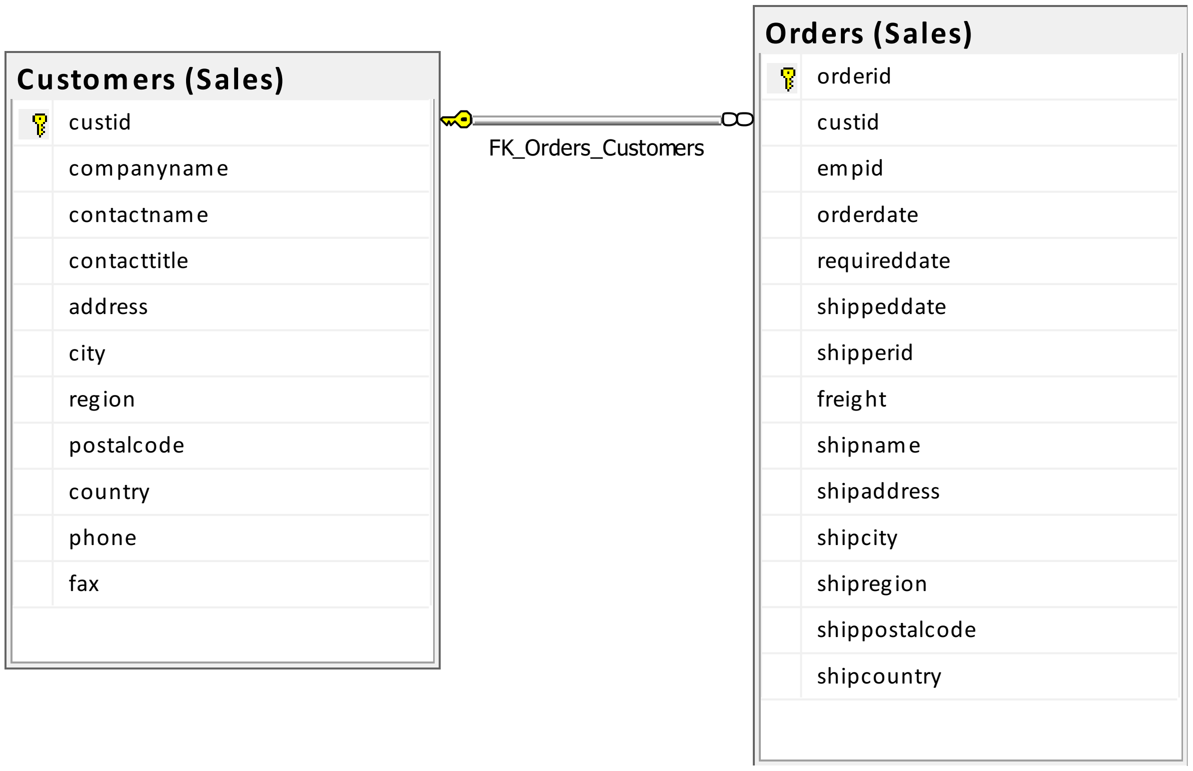
```
Query executed... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorksDW2014 | 00:00:00 | 312 rows
```

Conclusion: There are a total of 312 single male customers from British Columbia. More information could be displayed here.

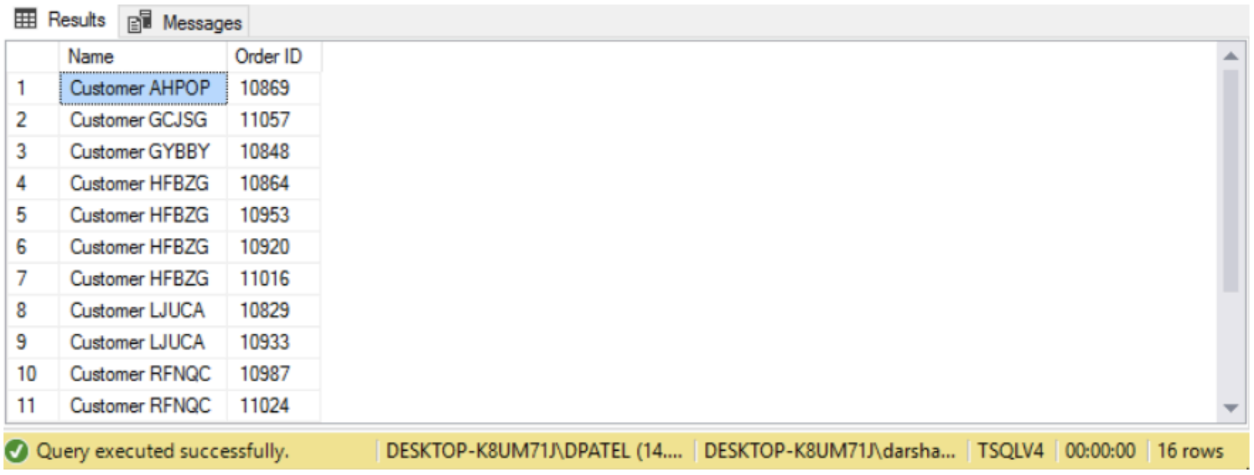
Question 3 (simple): Using TSQLV4, list the names and order id of customers who ordered from the UK in 2016. This query will be used to count how many orders were sent to the UK in 2016.

```
USE TSQLV4;
SELECT C.companyname as [Name], O.orderid as [Order ID]
FROM Sales.Orders as O
        INNER JOIN Sales.Customers as C
                ON O.custid = C.custid
WHERE O.shipcountry = N'UK' AND O.orderdate >= '20160101' AND O.orderdate < '20170101'
GROUP BY C.companyname, O.orderid
ORDER BY C.companyname;
```

Database Diagram: The Orders column and Customers column are joined using the custid key.



Output Table:



Results | Messages

	Name	Order ID
1	Customer AHPOP	10869
2	Customer GCJSG	11057
3	Customer GYBBY	10848
4	Customer HFBZG	10864
5	Customer HFBZG	10953
6	Customer HFBZG	10920
7	Customer HFBZG	11016
8	Customer LJUCA	10829
9	Customer LJUCA	10933
10	Customer RFNQC	10987
11	Customer RFNQC	11024

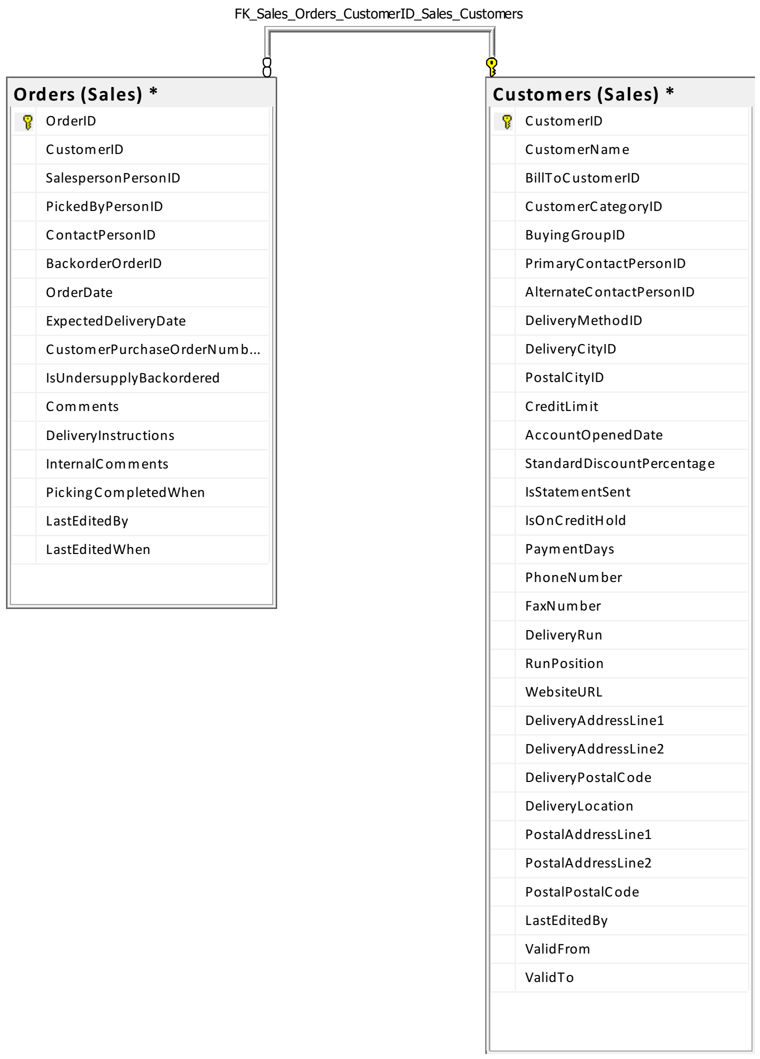
```
Query executed successfully. | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | TSQLV4 | 00:00:00 | 16 rows
```

Conclusion: A total of 16 orders were made by UK customers in 2016. Many customers ordered as much as three times.

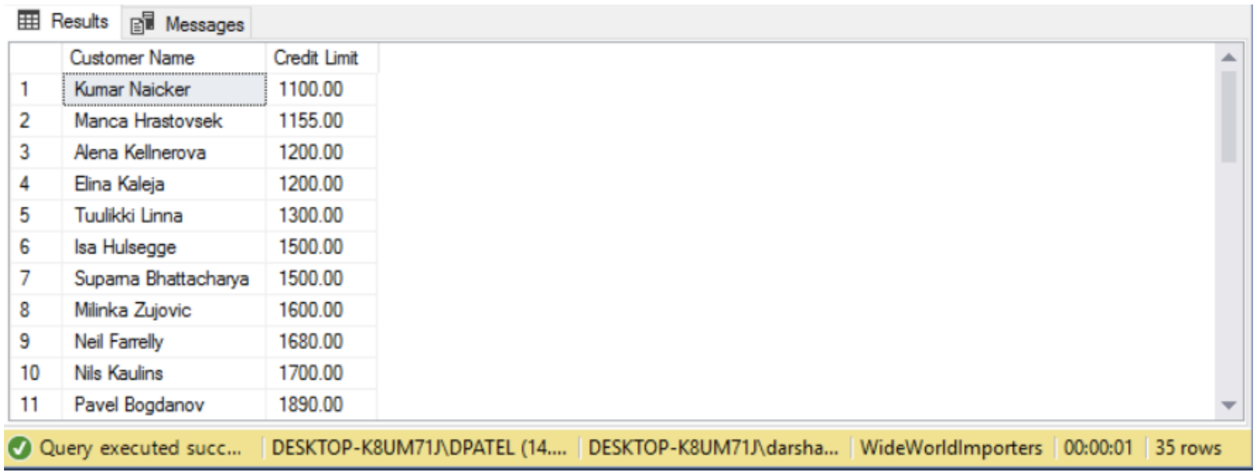
Question 4 (simple): Using WideWorldImporters, list distinct customer names who have a credit limit, their expected delivery date in the second half of 2016 and their credit limit. Order by credit limit. This query can be used to determine the customers who have a credit limit and had a delivered order in January or February 2016.

```
USE WideWorldImporters;
SELECT DISTINCT C.CustomerName as [Customer Name], C.CreditLimit as [Credit Limit]
FROM Sales.Customers as C
        INNER JOIN Sales.Orders as O
                ON C.CustomerId = O.CustomerId
WHERE C.CreditLimit is not NULL and O.ExpectedDeliveryDate >= '20160601'
         AND O.ExpectedDeliveryDate < '20170101'
ORDER BY C.CreditLimit;
```

Database Diagram: The Orders column and Customers column are joined using the CustomerID key.



Output Table:



Results | Messages

	Customer Name	Credit Limit
1	Kumar Naicker	1100.00
2	Manca Hrastovsek	1155.00
3	Alena Kellnerova	1200.00
4	Elina Kaleja	1200.00
5	Tuulikki Linna	1300.00
6	Isa Hulsegge	1500.00
7	Suparna Bhattacharya	1500.00
8	Milinka Zujovic	1600.00
9	Neil Farrelly	1680.00
10	Nils Kaulins	1700.00
11	Pavel Bogdanov	1890.00

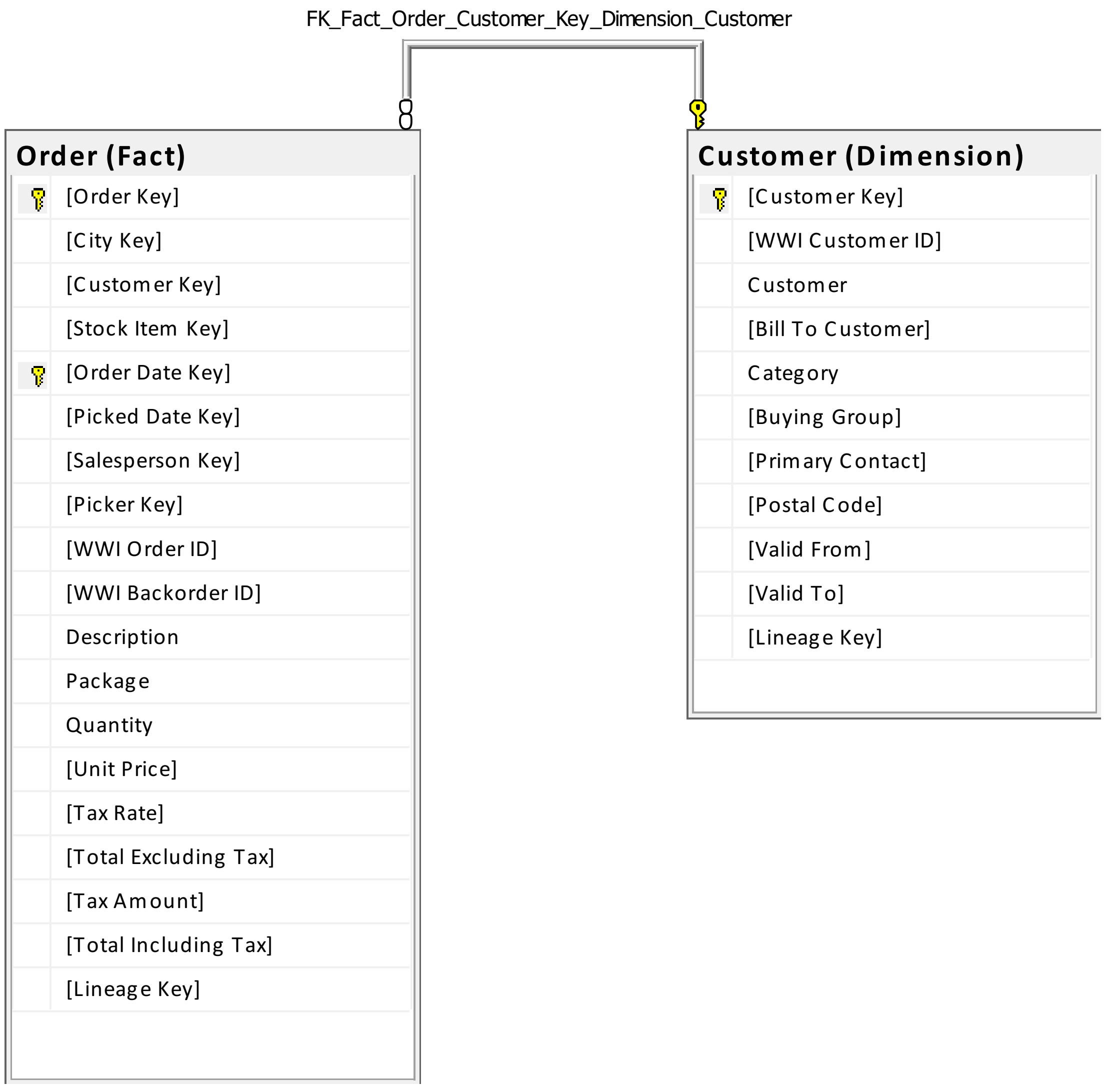
```
Query executed succ... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImporters | 00:00:01 | 35 rows
```

Conclusion: There are 35 customers who ordered in the second half of 2016 and have a credit limit. The lowest credit limit is $1100.00.

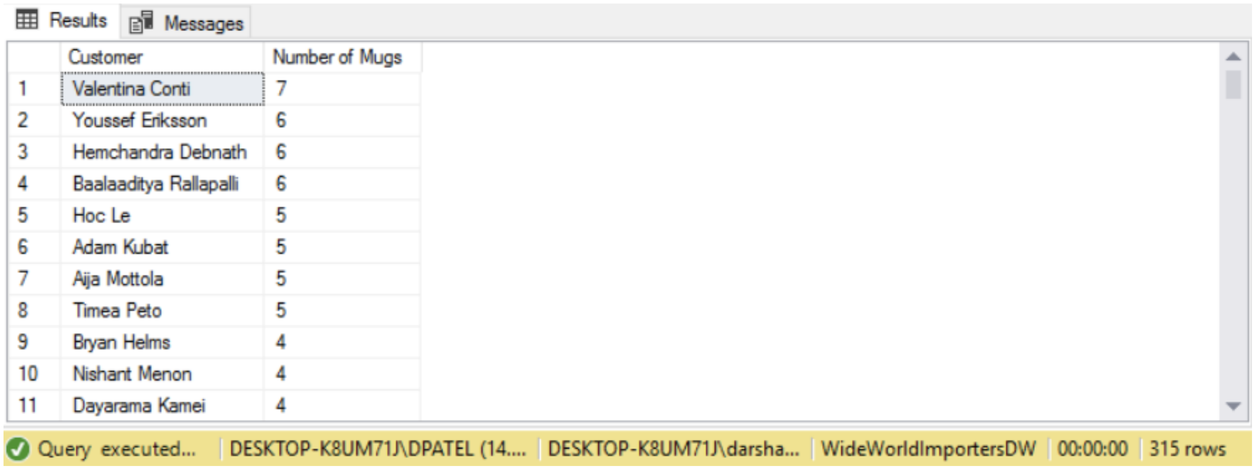
Question 5 (simple): Using WideWorldImportersDW, list the name of all the customers who have brought a 'DBA joke mug - mind if I join you? (Black)' mug and the quantity they brought in descending order. This query can be used to figure out how popular the mugs were.

```
USE WideWorldImportersDW;
SELECT C.[Primary Contact] as [Customer], COUNT(C.[Primary Contact]) as [Number of Mugs]
FROM Fact.[Order] as O
        INNER JOIN Dimension.Customer as C
                ON C.[Customer Key] = O.[Customer Key]
WHERE C.[Customer Key] != 0
         and O.[Description]  LIKE N'%DBA joke mug - mind if I join you? (Black)%'
GROUP BY C.[Primary Contact]
ORDER BY COUNT(C.[Primary Contact]) DESC;
```

Database Diagram: The Order column and Customer column are joined the Customer key.



Output Table:



Results | Messages

	Customer	Number of Mugs
1	Valentina Conti	7
2	Youssef Eriksson	6
3	Hemchandra Debnath	6
4	Baalaaditya Rallapalli	6
5	Hoc Le	5
6	Adam Kubat	5
7	Aija Mottola	5
8	Timea Peto	5
9	Bryan Helms	4
10	Nishant Menon	4
11	Dayarama Kamei	4

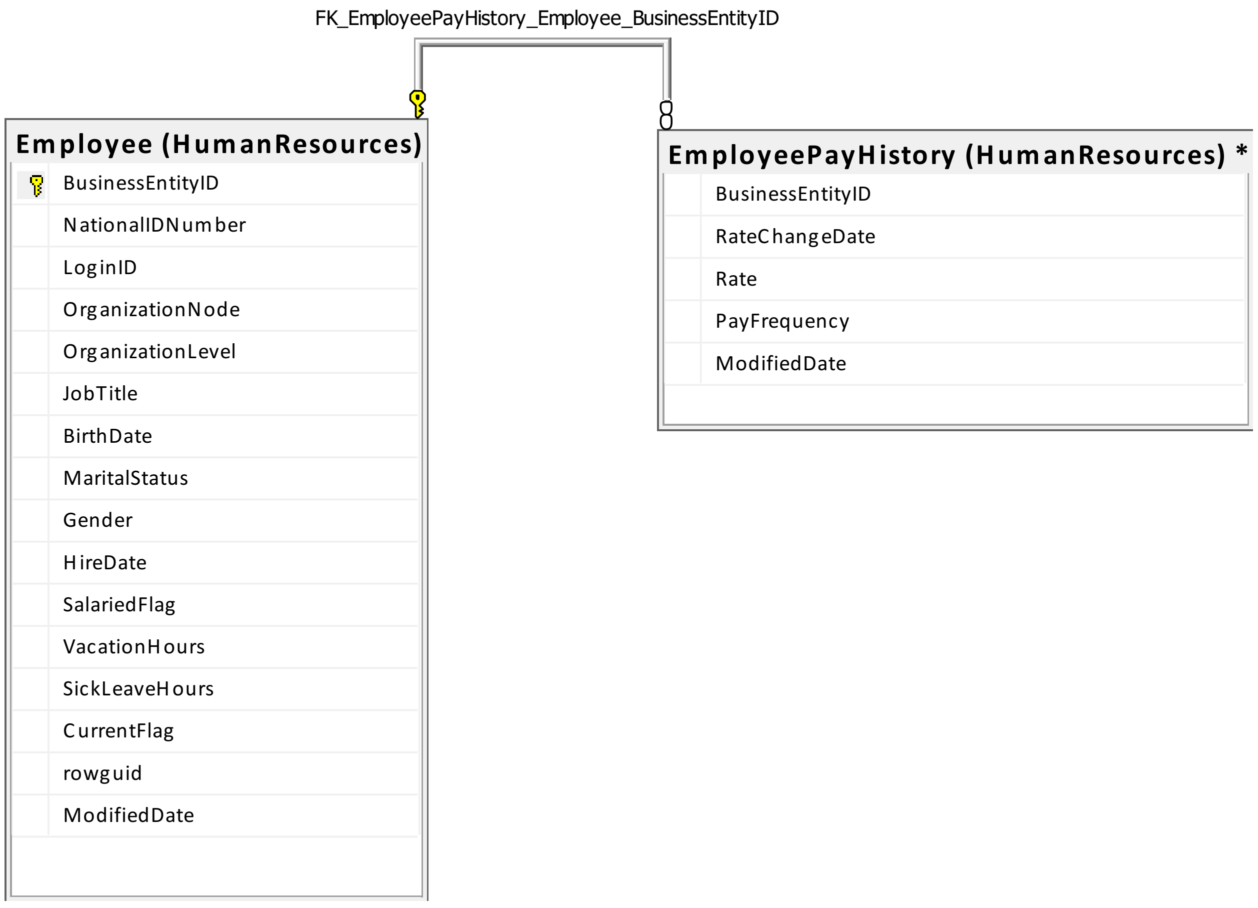
```
Query executed... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImportersDW | 00:00:00 | 315 rows
```

Conclusion: There are a total of 315 customers who brought at least one of the 'DBA joke mug - mind if I join you? (Black)' mug. Customer Valentina Conti was the one who brought the greatest quantity, 7. This query could be improved by simply listing how many customers brought x numbers of 'DBA joke mug - mind if I join you? (Black)' mugs.

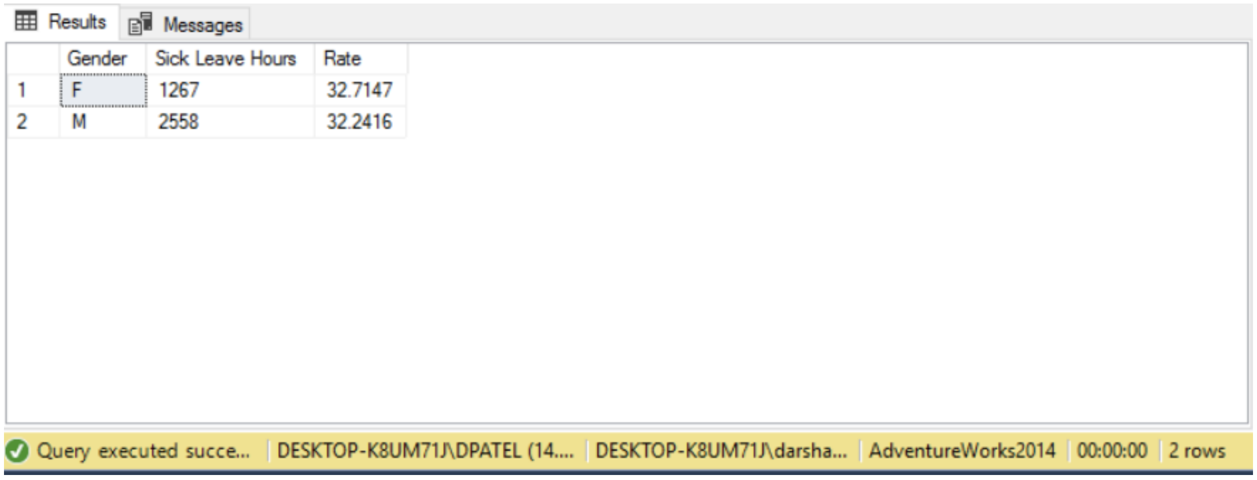
Question 6 (medium): Using AdventureWorks2014, sum up the number of sick leave hours by gender and the average rate if their rate was greater than $20. Group by their gender. This query can be used to determine if there's a rate sick hour difference amongst gender and its impact on rate.

```
USE AdventureWorks2014;
SELECT E.Gender, SUM(E.SickLeaveHours) as [Sick Leave Hours], AVG(P.Rate) as Rate
FROM HumanResources.Employee as E
        INNER JOIN HumanResources.EmployeePayHistory as P
                ON E.BusinessEntityID = P.BusinessEntityID
WHERE P.Rate > 20
GROUP BY E.Gender;
```

Database Diagram: The Employee column and EmployeePayHistory column are joined using the BusinessEntityID key.



Output Table:



Results | Messages

	Gender	Sick Leave Hours	Rate
1	F	1267	32.7147
2	M	2558	32.2416

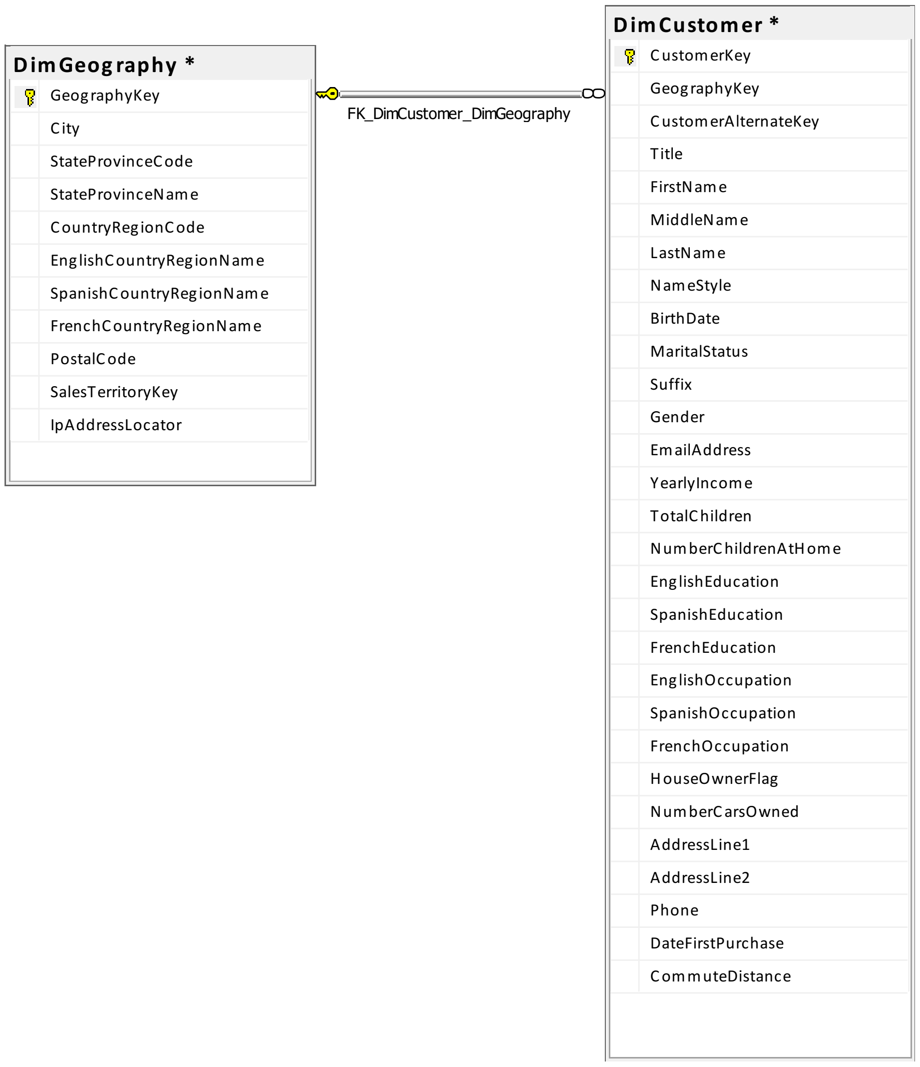
```
Query executed succe... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorks2014 | 00:00:00 | 2 rows
```

Conclusion: Male employees had a greater number of sick leave hours than female employees. However, their average rates are nearly the same.

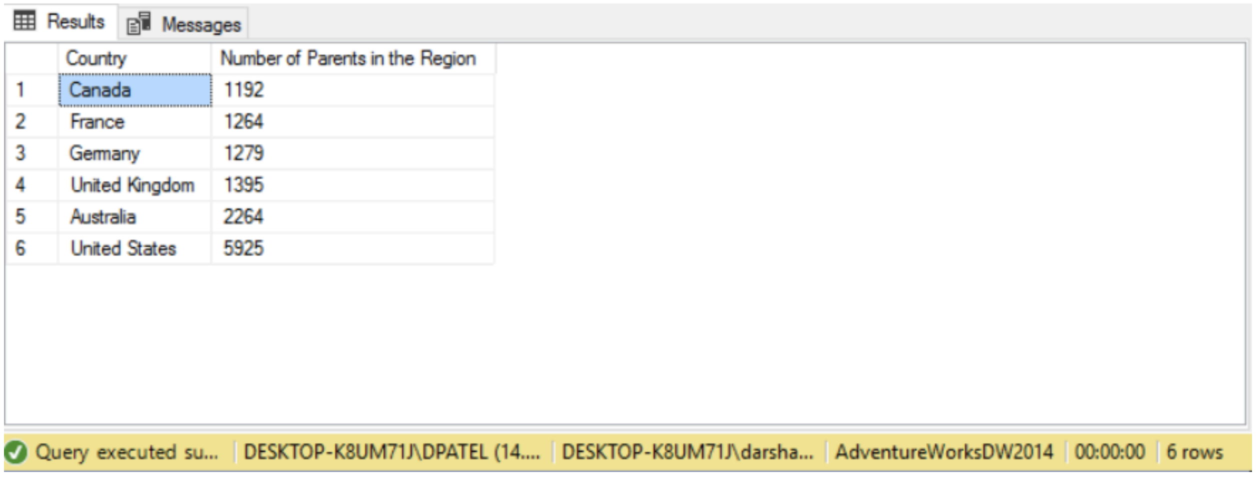
Question 7 (medium): Using AdventureWorksDW2014, list the number of parents in each of the countries in ascending order of number of parents. This query can be used for census purposes.

```
USE AdventureWorksDW2014;
SELECT G.EnglishCountryRegionName as [Country],
            COUNT(G.EnglishCountryRegionName) as [Number of Parents in the Region]
FROM dbo.DimCustomer as C
           INNER JOIN dbo.DimGeography as G
                    ON C.GeographyKey = G.GeographyKey
WHERE C.TotalChildren > 0
GROUP BY G.EnglishCountryRegionName
ORDER BY COUNT(G.EnglishCountryRegionName);
```

Database Diagram: The Geography column and Customer column are joined using the Geography key.



Output Table:



Results | Messages

	Country	Number of Parents in the Region
1	Canada	1192
2	France	1264
3	Germany	1279
4	United Kingdom	1395
5	Australia	2264
6	United States	5925

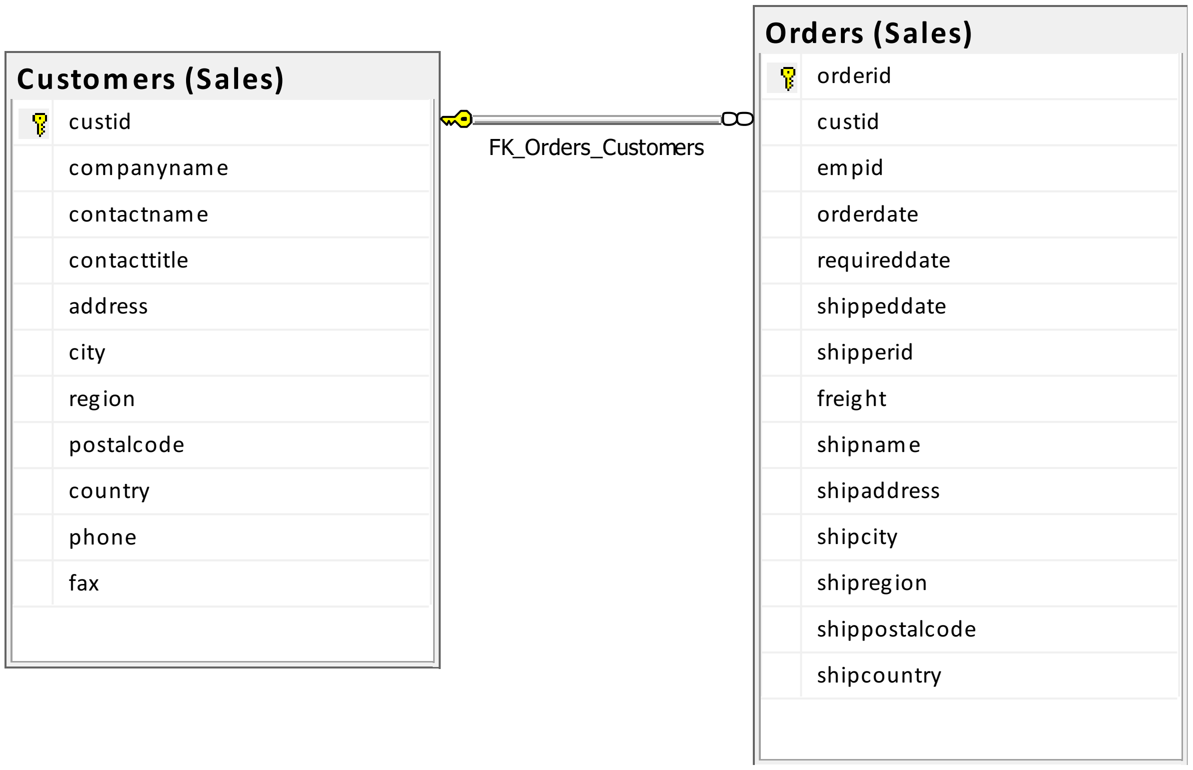
```
Query executed su... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorksDW2014 | 00:00:00 | 6 rows
```

Conclusion: Out of 6 countries, the least number of parents lived in Canada, and the most number of parents lived in the United States, with a difference of nearly 5000 parents between them.

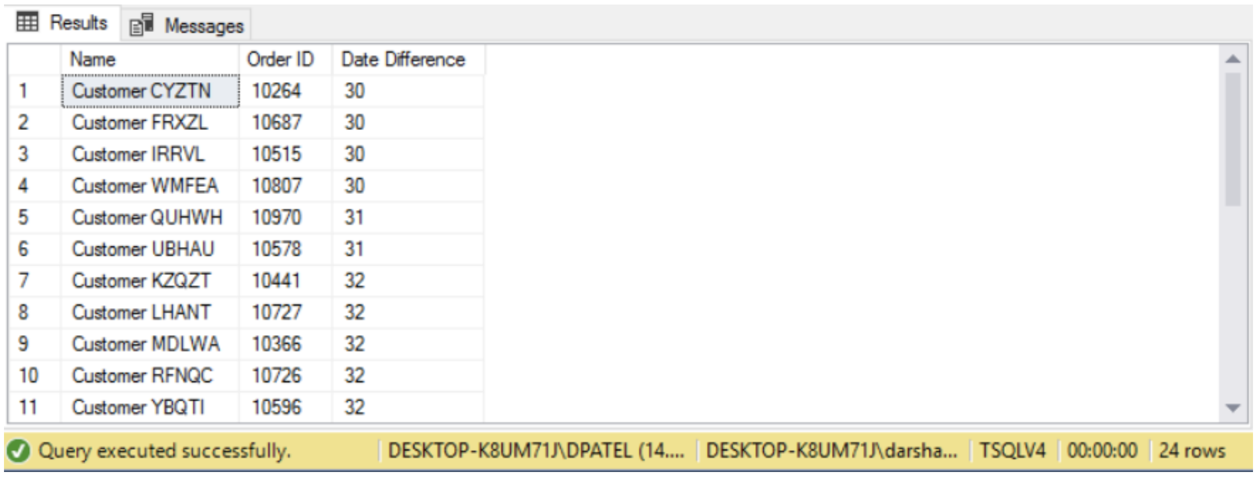
Question 8 (medium): Using TSQLV4, list the name of customers who had a difference of at least 30 days between order date and shipped date along with their order id and date difference in ascending order. This query can be used to evaluate shipping efficiencies.

```
USE TSQLV4;
SELECT C.companyname as [Name], O.orderid as [Order ID],
            DATEDIFF(day, O.orderdate, O.shippeddate) as [Date Difference]
FROM Sales.Orders as O
          INNER JOIN Sales.Customers as C
                    ON O.custid = C.custid
WHERE DATEDIFF(day, O.orderdate, O.shippeddate) >= 30
ORDER BY DATEDIFF(day, O.orderdate, O.shippeddate), C.companyname;
```

Database Diagram: The Customers column and Sales column are joined using the custid key.



Output Table:



Results | Messages

	Name	Order ID	Date Difference
1	Customer CYZTN	10264	30
2	Customer FRXZL	10687	30
3	Customer IRRVL	10515	30
4	Customer WMFEA	10807	30
5	Customer QUHWH	10970	31
6	Customer UBHAU	10578	31
7	Customer KZQZT	10441	32
8	Customer LHANT	10727	32
9	Customer MDLWA	10366	32
10	Customer RFNQC	10726	32
11	Customer YBQTI	10596	32

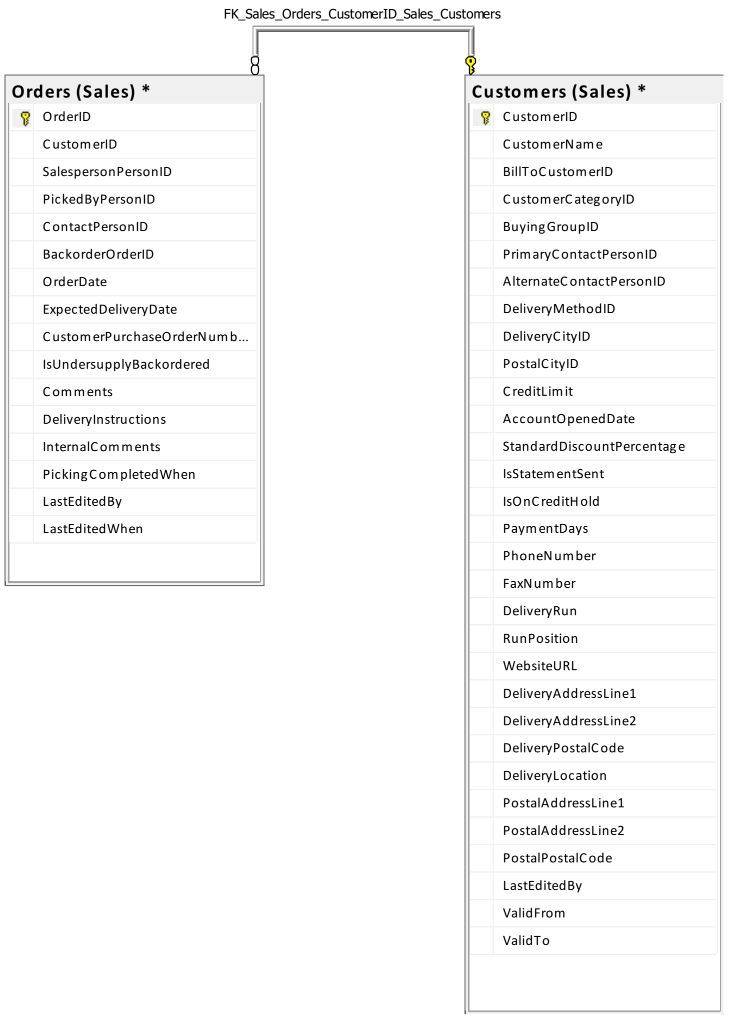
```
Query executed successfully. | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | TSQLV4 | 00:00:00 | 24 rows
```

Conclusion: A total of 24 orders were shipped with a delay of at least 30 days. This query could be improved by ordering the date difference in descending order.

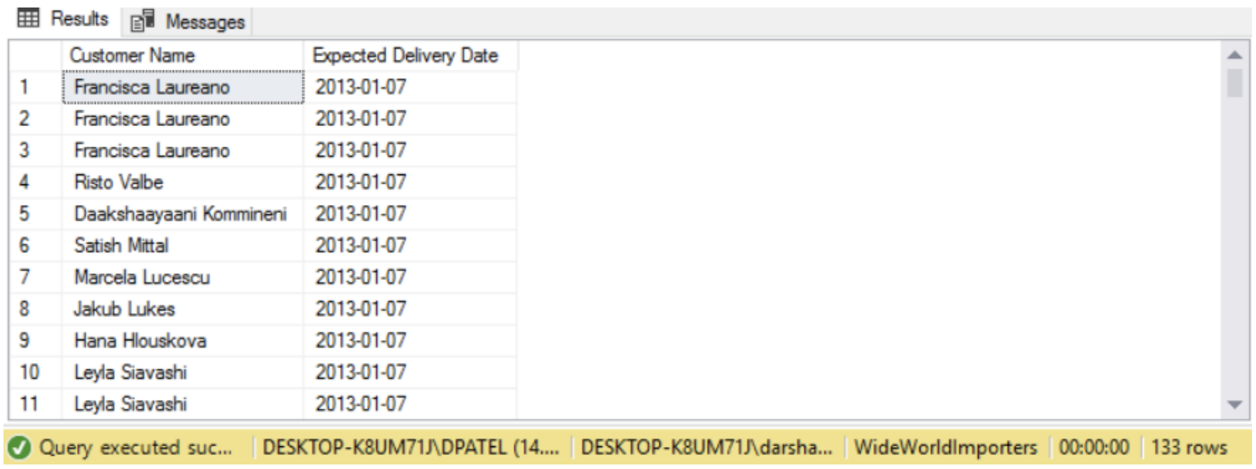
Question 9 (medium): Using WideWorldImporters, find individual customers (not toy stores) who ordered in 2013 and waited more than a day for their order to be delivered. This query can be used to examine delivery efficiency.

```
USE WideWorldImporters;
SELECT C.CustomerName as [Customer Name], O.ExpectedDeliveryDate as [Expected Delivery Date]
FROM Sales.Customers as C
        INNER JOIN Sales.Orders as O
                ON C.CustomerId = O.CustomerId
WHERE DATEDIFF(day, O.orderdate, O.ExpectedDeliveryDate) > 1
                and O.orderdate >= '20130101' and O.orderdate < '20130201'
                and C.CustomerName not LIKE '%toys%'
ORDER BY O.ExpectedDeliveryDate;
```

Database Diagram: The Orders column and Customers column are joined using the CustomerId key.



Output Table:



	Customer Name	Expected Delivery Date
1	Francisca Laureano	2013-01-07
2	Francisca Laureano	2013-01-07
3	Francisca Laureano	2013-01-07
4	Risto Valbe	2013-01-07
5	Daakshaayaani Kommineni	2013-01-07
6	Satish Mittal	2013-01-07
7	Marcela Lucescu	2013-01-07
8	Jakub Lukes	2013-01-07
9	Hana Hlouskova	2013-01-07
10	Leyla Siavashi	2013-01-07
11	Leyla Siavashi	2013-01-07

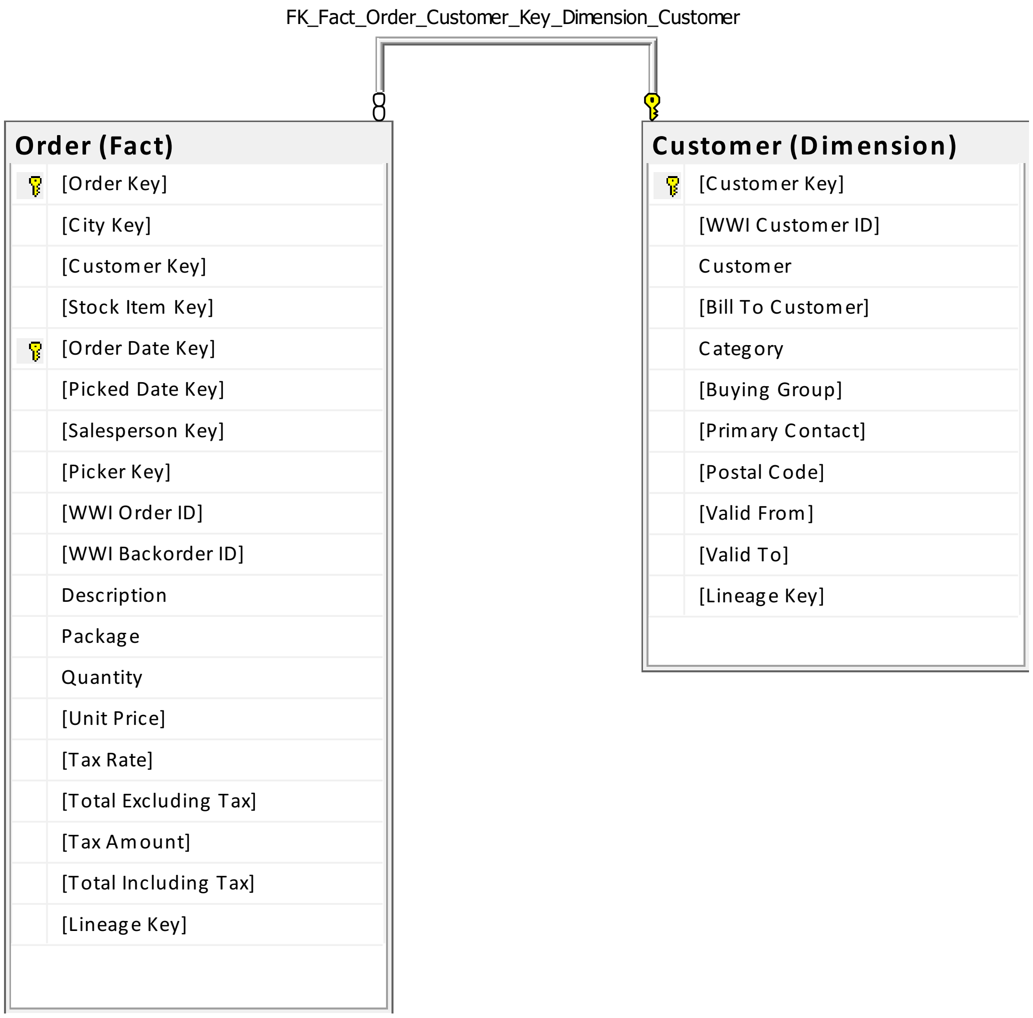
```
Query executed suc... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImporters | 00:00:00 | 133 rows
```

Conclusion: There are 133 non-distinct customers who ordered in 2013 and waited at least a day for delivery. This query could be improved by having a distinct clause to name a customer once for a single day.

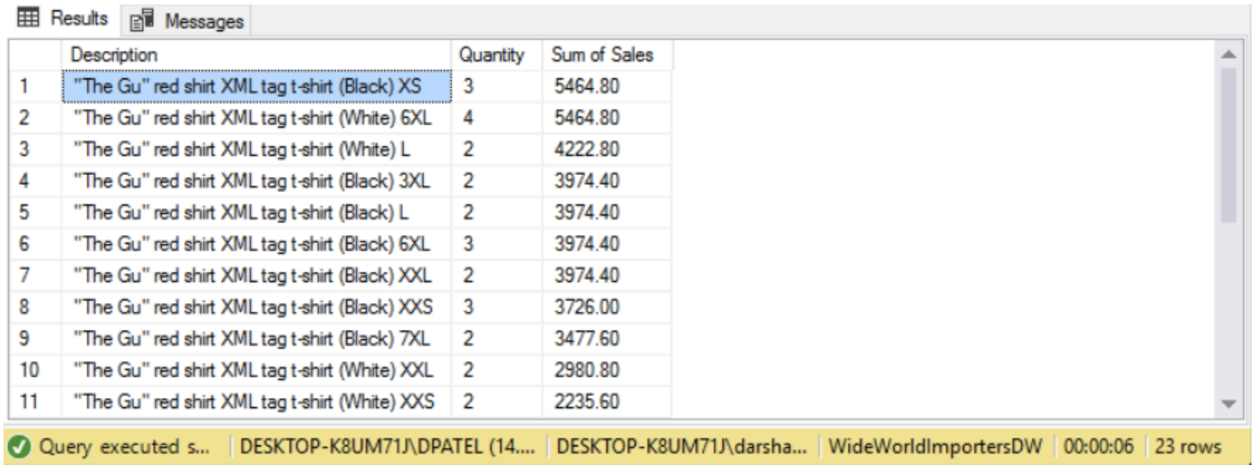
Question 10 (medium): Using WideWorldImportersDW, list the different "The Gu" red shirts bought at the Belgreen, AL location of Tailspin Toys, the quantity brought and the sum of sales accumulated in ascending order. This query can be used to determine which size is most profitable.

```
USE WideWorldImportersDW;
SELECT O.[Description], COUNT(O.[Description]) as [Quantity],
            SUM(O.[Total Including Tax]) as [Sum of Sales]
FROM Fact.[Order] as O
          INNER JOIN Dimension.Customer as C
                    ON C.[Customer Key] = O.[Customer Key]
WHERE C.Customer LIKE '%Belgreen%' and O.[Description]  LIKE N'"The Gu%'
GROUP BY O.[Description]
ORDER BY SUM(O.[Total Including Tax]) DESC;
```

Database Diagram: The Order column and Customer column are joined using the Customer key.



Output Table:



Results | Messages

	Description	Quantity	Sum of Sales
1	"The Gu" red shirt XML tag t-shirt (Black) XS	3	5464.80
2	"The Gu" red shirt XML tag t-shirt (White) 6XL	4	5464.80
3	"The Gu" red shirt XML tag t-shirt (White) L	2	4222.80
4	"The Gu" red shirt XML tag t-shirt (Black) 3XL	2	3974.40
5	"The Gu" red shirt XML tag t-shirt (Black) L	2	3974.40
6	"The Gu" red shirt XML tag t-shirt (Black) 6XL	3	3974.40
7	"The Gu" red shirt XML tag t-shirt (Black) XXL	2	3974.40
8	"The Gu" red shirt XML tag t-shirt (Black) XXS	3	3726.00
9	"The Gu" red shirt XML tag t-shirt (Black) 7XL	2	3477.60
10	"The Gu" red shirt XML tag t-shirt (White) XXL	2	2980.80
11	"The Gu" red shirt XML tag t-shirt (White) XXS	2	2235.60

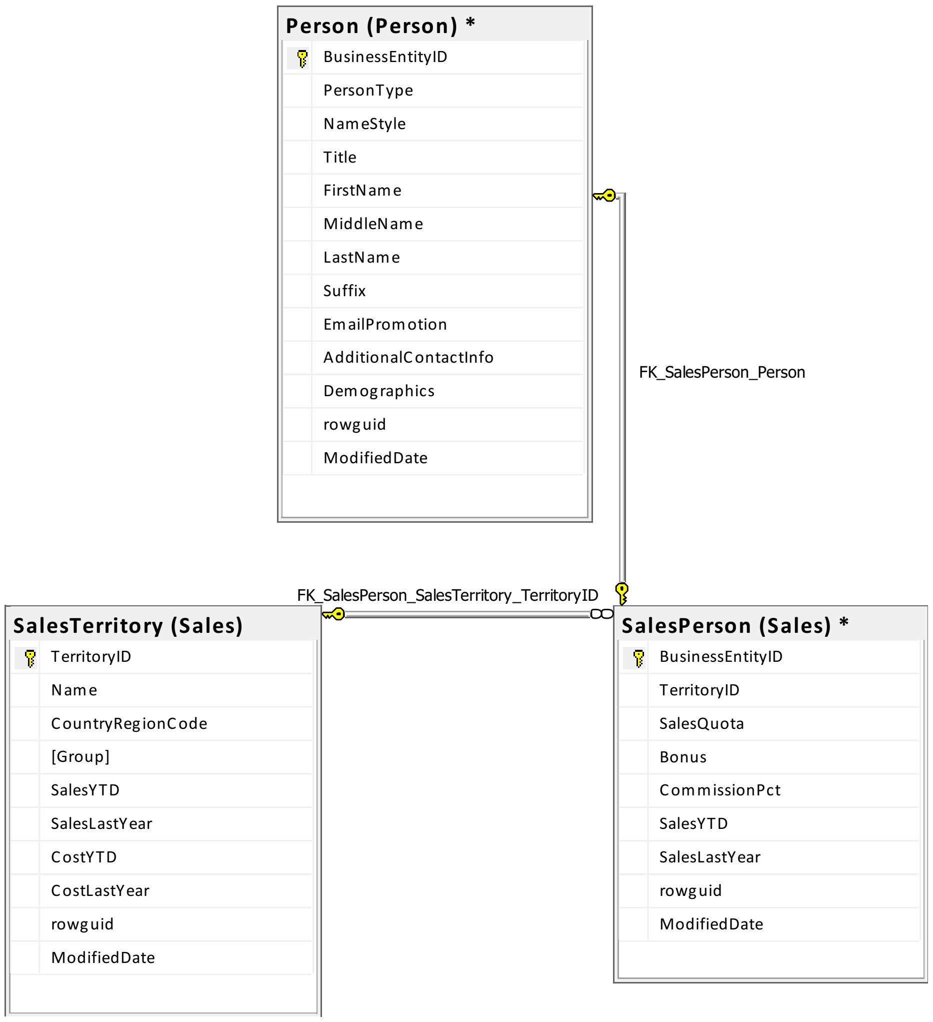
```
Query executed s... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImportersDW | 00:00:06 | 23 rows
```

Conclusion: There are a total of 23 different "The Gu" t-shirts sold in varying sizes and color. The maximum sales were made by 3 XS black shirts and 4 6XL white shirts.

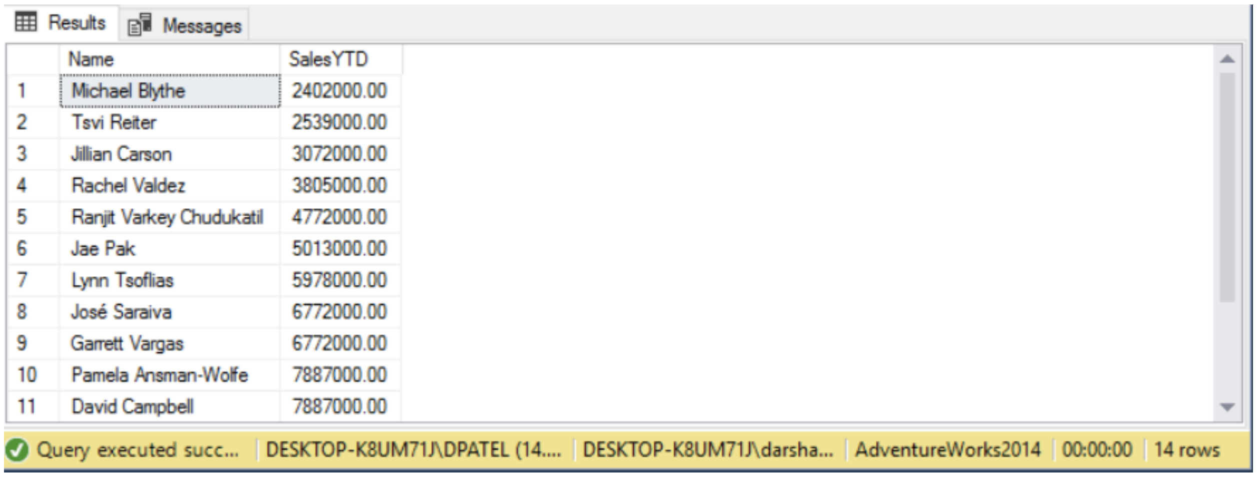
Question 11 (medium): Using AdventureWorks2014, list the SalesYTD rounded to the thousandths for each of the salesperson, ordered by SalesYTD and last name. This query can be used to determine who is making the most sale.

```
USE AdventureWorks2014;
SELECT CONCAT(P.FirstName, ' ',P.LastName) as [Name], ROUND(T.SalesYTD,-3) as [SalesYTD]
FROM Sales.SalesPerson as S
        INNER JOIN Person.Person as P
                ON S.BusinessEntityID = P.BusinessEntityID
        INNER JOIN Sales.SalesTerritory as T
                ON S.TerritoryID = T.TerritoryID
GROUP BY P.LastName, P.FirstName, T.SalesYTD
ORDER BY T.SalesYTD, P.LastName;
```

Database Diagram: The SalesPerson column and Person column are joined by a BusinessEntityID key while the SalesPerson column and SalesTerritory column are joined by a TerritoryID key.



Output Table:



Results | Messages

	Name	SalesYTD
1	Michael Blythe	2402000.00
2	Tsvi Reiter	2539000.00
3	Jillian Carson	3072000.00
4	Rachel Valdez	3805000.00
5	Ranjit Varkey Chudukatil	4772000.00
6	Jae Pak	5013000.00
7	Lynn Tsoflias	5978000.00
8	José Saraiva	6772000.00
9	Garrett Vargas	6772000.00
10	Pamela Ansman-Wolfe	7887000.00
11	David Campbell	7887000.00

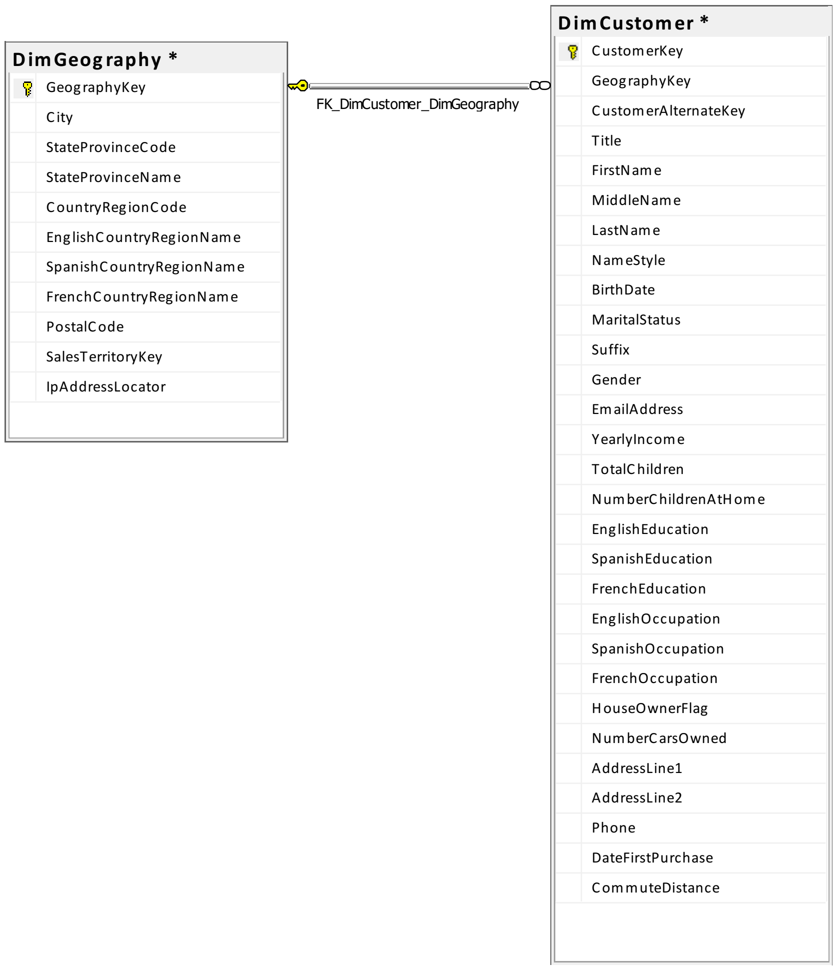
Query executed succ... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorks2014 | 00:00:00 | 14 rows

Conclusion: There are a total of 14 salespersons who have made at least $2,402,000 in the current year up to the current date. If this was descending, we can determine who is making the most sale.

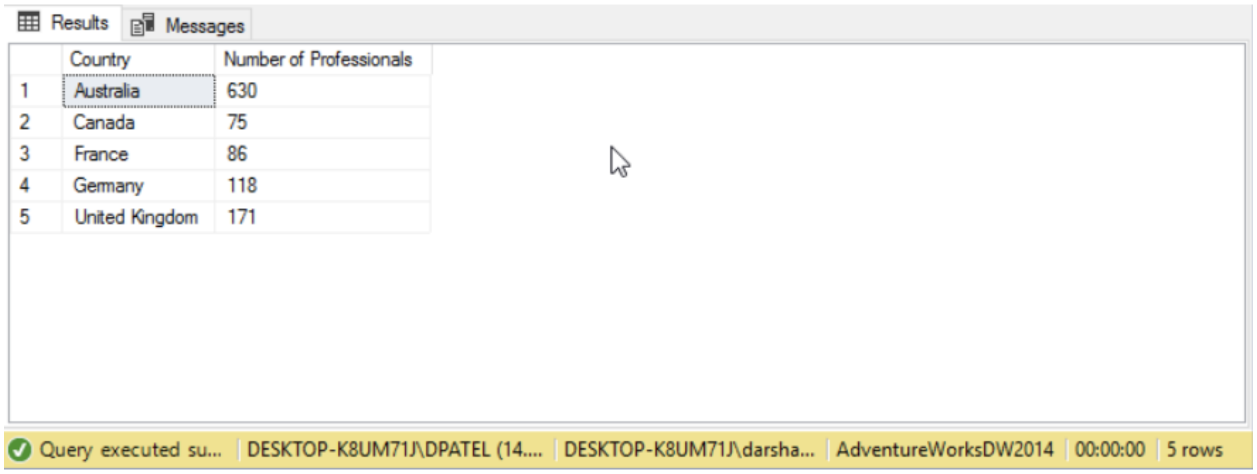
Question 12 (medium): Using AdventureWorksDW2014, list the number of non-American professionals who own 3 or more cars by country in ascending order. This query can be used for car data census.

```
USE AdventureWorksDW2014;
SELECT G.EnglishCountryRegionName as [Country],
    COUNT(C.EnglishOccupation) as [Number of Professionals]
FROM dbo.DimCustomer as C
        INNER JOIN dbo.DimGeography as G
                ON C.GeographyKey = G.GeographyKey
WHERE G.EnglishCountryRegionName != N'United States'
                and C.EnglishOccupation = N'Professional'
                and C.NumberCarsOwned >= 3
GROUP BY G.EnglishCountryRegionName, C.EnglishOccupation
ORDER BY G.EnglishCountryRegionName;
```

Database Diagram: The Geography column and Customer column are joined by the Geography key.



Output Table:



Results | Messages

	Country	Number of Professionals
1	Australia	630
2	Canada	75
3	France	86
4	Germany	118
5	United Kingdom	171

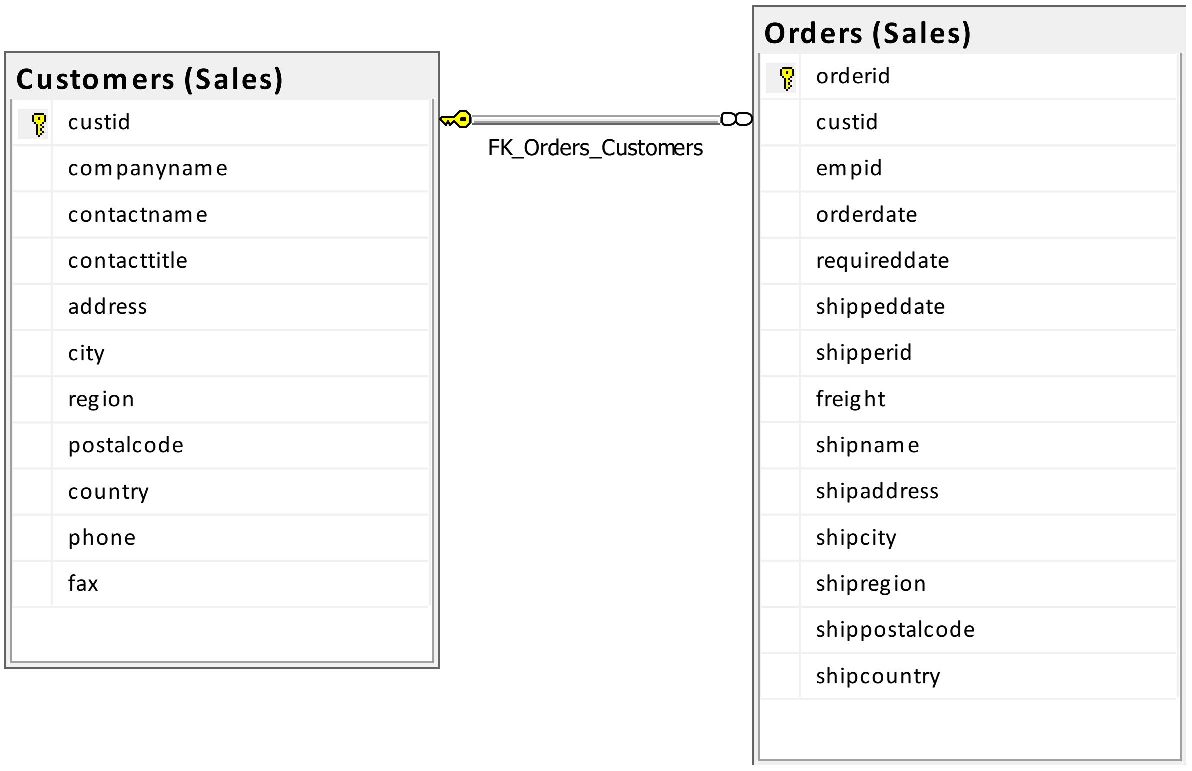
Query executed su... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorksDW2014 | 00:00:00 | 5 rows

Conclusion: Of the 5 countries, the country that had the most professionals with 3 or more cars was Australia, with 630 professionals. The country that had the least professionals with 3 or more cars was the United Kingdom, with 171 professionals.

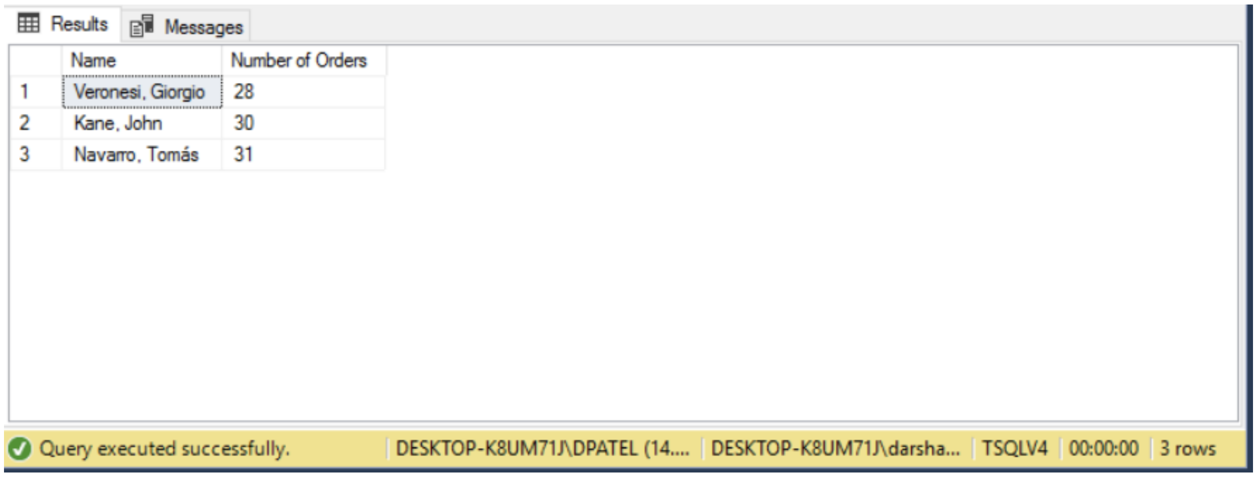
Question 13 (medium): Using TSQLV4, list the customers who placed at least 20 orders and the number of orders. This query can be used to indicate who is making the most orders.

```
USE TSQLV4;
SELECT C.contactname AS [Name], COUNT(O.orderid) AS [Number of Orders]
FROM Sales.Customers AS C
	INNER JOIN Sales.Orders AS O
		ON C.custid = O.custid
GROUP BY C.contactname
HAVING COUNT(O.orderid) >= 20
ORDER BY COUNT(O.orderid), C.contactname;
```

Database Diagram: The Customers column and Orders column are joined using the custid key.



Output Table:


Results | Messages

	Name	Number of Orders
1	Veronesi, Giorgio	28
2	Kane, John	30
3	Navarro, Tomás	31

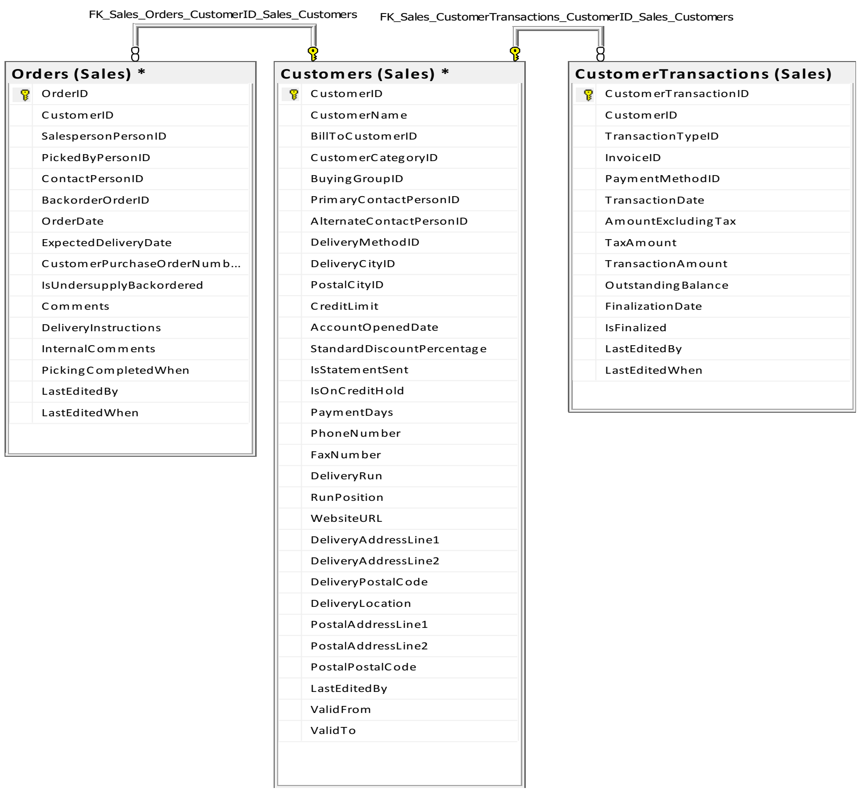
Query executed successfully. | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | TSQLV4 | 00:00:00 | 3 rows

Conclusion: 3 customers were found that ordered at least 20 times, from 28 to 31 orders.

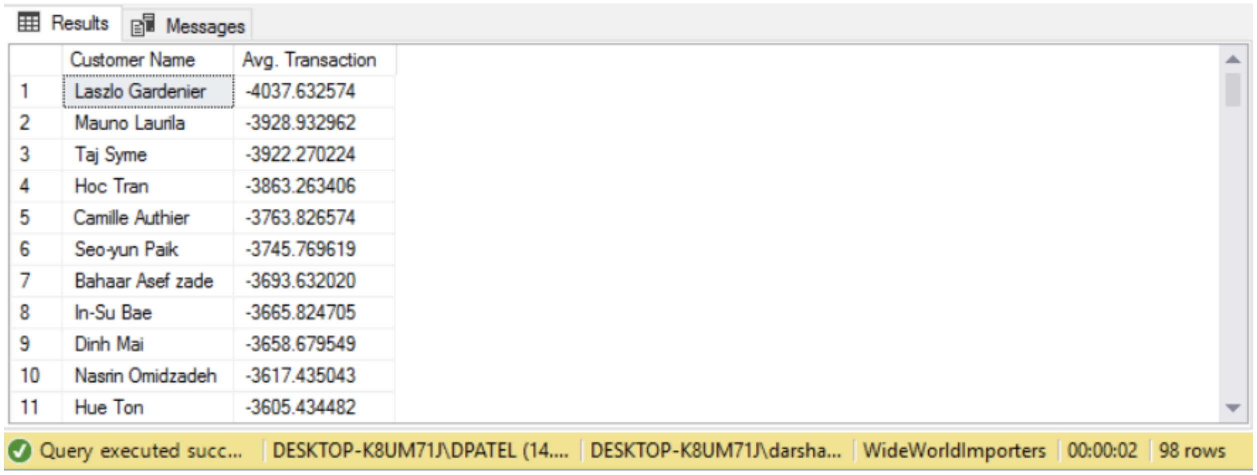
Question 14 (medium): Using WideWorldImporters, list the non toy store customers who have ordered in the first two months of 2014 and had an average transaction less than -3000 in the time frame in ascending order. This query can determine who is making the most purchase in the first two months of 2014.

```
USE WideWorldImporters;
SELECT C.CustomerName as [Customer Name], AVG(T.TransactionAmount) as [Avg.
Transaction]
FROM Sales.Customers as C
        INNER JOIN Sales.Orders as O
                ON C.CustomerId = O.CustomerId
        INNER JOIN Sales.CustomerTransactions as T
                ON C.CustomerId = T.CustomerId
WHERE T.TaxAmount = 0 and O.OrderDate >= '20140101' and O.Orderdate < '20140301'
         and C.CustomerName NOT LIKE N'Wingtip%' and C.CustomerName NOT LIKE
N'Tailspin%'
GROUP BY C.CustomerName
HAVING AVG(T.TransactionAmount) < -3000
ORDER BY AVG(T.TransactionAmount);
```

Database Diagram: The Orders column and Customers column are joined using the CustomerId key while the Customers column and CustomerTransaction column are also joined using the CustomerId key.



Output Table:



Results | Messages

	Customer Name	Avg. Transaction
1	Laszlo Gardenier	-4037.632574
2	Mauno Laurila	-3928.932962
3	Taj Syme	-3922.270224
4	Hoc Tran	-3863.263406
5	Camille Authier	-3763.826574
6	Seo-yun Paik	-3745.769619
7	Bahaar Asef zade	-3693.632020
8	In-Su Bae	-3665.824705
9	Dinh Mai	-3658.679549
10	Nasrin Omidzadeh	-3617.435043
11	Hue Ton	-3605.434482

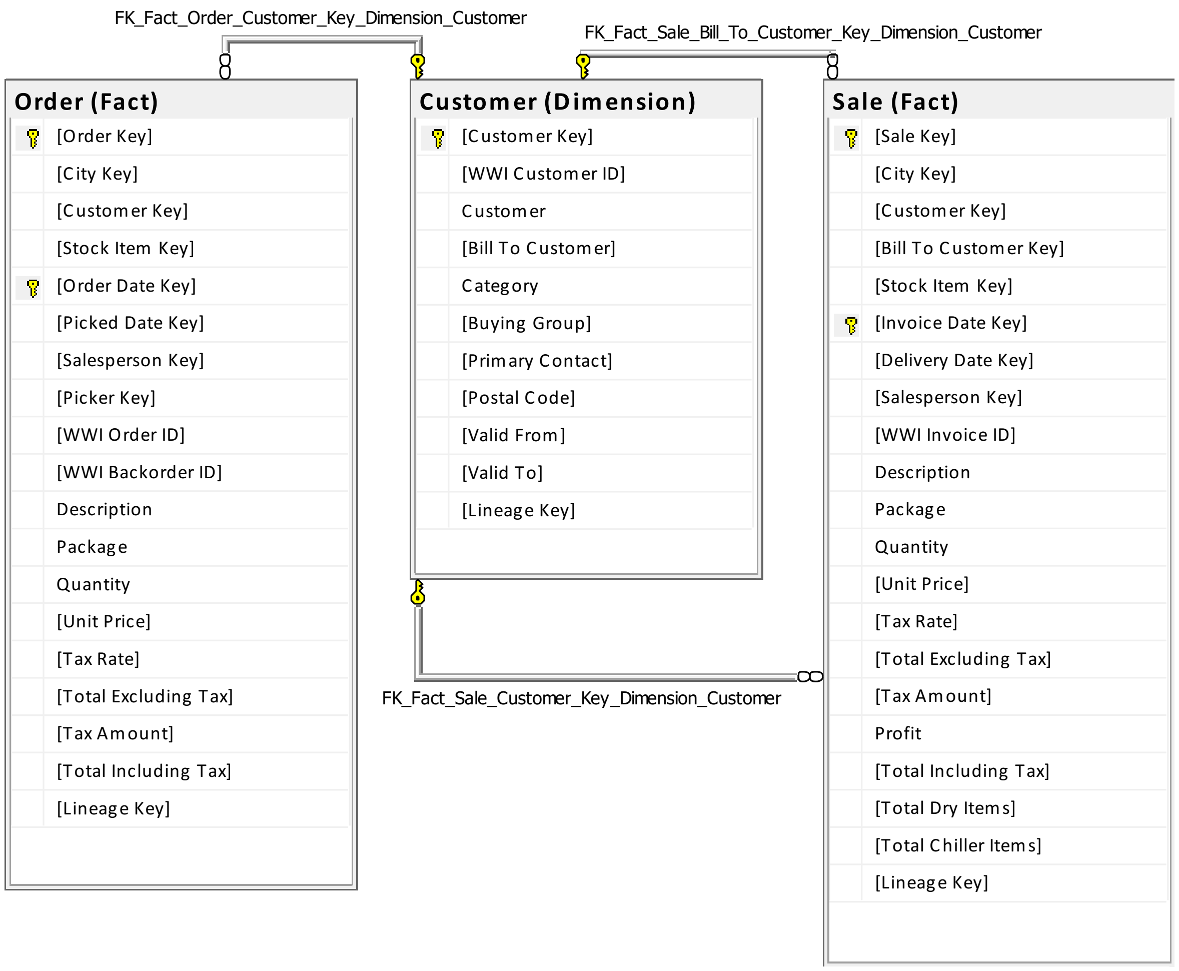
```
Query executed succ... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImporters | 00:00:02 | 98 rows
```

Conclusion: A total of 98 customers had an average transaction of less than $-3000.00 in the first two months of 2014. This means that they spend up to $4037.63 on the average amount. This query can be improved by getting rid of the negative sign ease confusion.

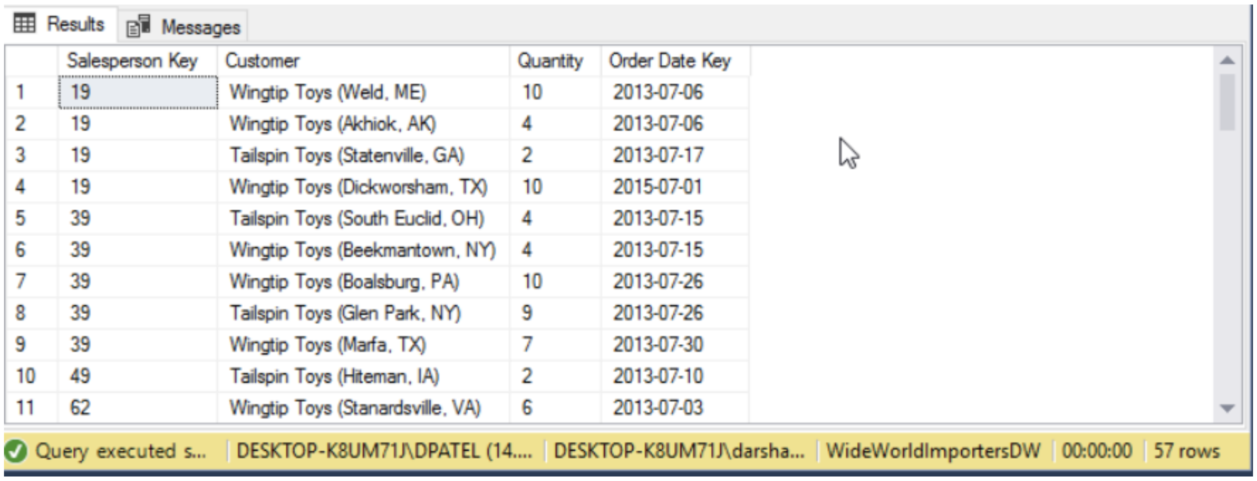
Question 15 (medium): Using WideWorldImportersDW, list the customers who brought a medium blue jacket in July and the quantity as well as the salesperson associated. This query can be used to determine which salesperson is making an effort to sell a jacket in the summer.

```
USE WideWorldImportersDW;
SELECT S.[SalesPerson Key] as [Salesperson Key], C.[Customer],  O.Quantity, O.[Order Date Key]
FROM Fact.[Order] as O
        INNER JOIN Dimension.Customer as C
                ON C.[Customer Key] = O.[Customer Key]
        INNER JOIN Fact.Sale as S
                ON S.[SalesPerson Key] = O.[SalesPerson Key]
WHERE C.[Customer Key] != 0 and O.[Description] LIKE N'%jacket (Blue) M%'
   and MONTH(O.[Order Date Key]) = 7
GROUP BY S.[SalesPerson Key], C.[Customer], O.Quantity, O.[Order Date Key]
ORDER BY S.[SalesPerson Key], O.[Order Date Key];
```

Database Diagram: The Order column and Customer column are joined using the Customer key while the Customer column and Sale column are joined using the SalesPerson key.



Output Table:



	Salesperson Key	Customer	Quantity	Order Date Key
1	19	Wingtip Toys (Weld, ME)	10	2013-07-06
2	19	Wingtip Toys (Akhiok, AK)	4	2013-07-06
3	19	Tailspin Toys (Statenville, GA)	2	2013-07-17
4	19	Wingtip Toys (Dickworsham, TX)	10	2015-07-01
5	39	Tailspin Toys (South Euclid, OH)	4	2013-07-15
6	39	Wingtip Toys (Beekmantown, NY)	4	2013-07-15
7	39	Wingtip Toys (Boalsburg, PA)	10	2013-07-26
8	39	Tailspin Toys (Glen Park, NY)	9	2013-07-26
9	39	Wingtip Toys (Marfa, TX)	7	2013-07-30
10	49	Tailspin Toys (Hiteman, IA)	2	2013-07-10
11	62	Wingtip Toys (Stanardsville, VA)	6	2013-07-03

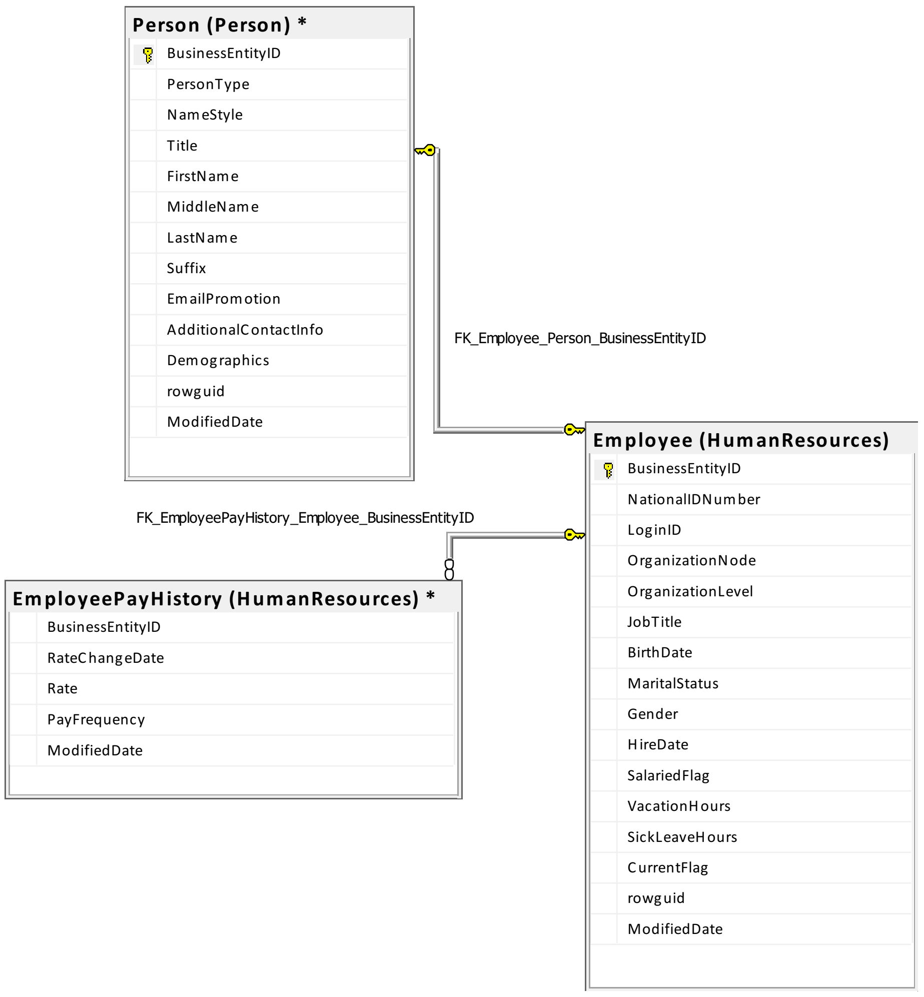
```
Query executed s... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImportersDW | 00:00:00 | 57 rows
```

Conclusion: A total of 57 customers were found who ordered a medium blue jacket in July, from several different salespersons. A better query would be to sort by order date and then quantity to analyze changes in purchases rather than analyzing the efficiency of salesperson, for which there appears to be none.

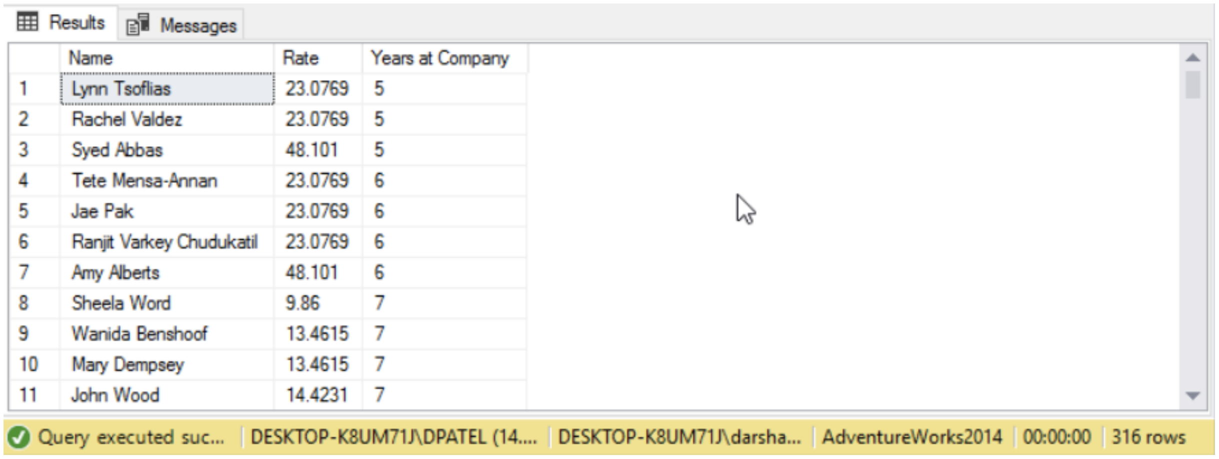
Question 16 (complex): Construct a function that will take in a date and return the number of years in between then and the current date. Using this function and AdventureWorks2014, list the name, rate and number of years worked by each Human Resources employee up to the present day in order of number of years. This query can be used to determine which employees were here for the longest time and if there's a wage difference amongst people who been in the company longer than those who been in the company for a shorter time.

```
USE AdventureWorks2014;

IF OBJECT_ID (N'dbo.YearsAtWork', N'FN') IS NOT NULL
        DROP FUNCTION YearsAtWork
CREATE FUNCTION dbo.YearsAtWork(@originaldate date)
RETURNS INT
AS
BEGIN
        DECLARE @years int;
        SELECT @years = DATEDIFF(year, @originaldate, GETDATE())
        FROM HumanResources.Employee
        WHERE HireDate = @originaldate
        RETURN @years;
END;

SELECT CONCAT(P.FirstName, ' ', P.LastName) as [Name], H.Rate,
    dbo.YearsAtWork(E.HireDate) as [Years at Company]
FROM HumanResources.Employee as E
INNER JOIN Person.Person as P
        ON E.BusinessEntityID = P.BusinessEntityID
INNER JOIN HumanResources.EmployeePayHistory as H
        ON E.BusinessEntityID = H.BusinessEntityID
ORDER BY dbo.YearsAtWork(E.HireDate), H.Rate, P.LastName;
```

Database Diagram: The Employee column and Person column are joined by the BusinessEntityID key as well as the Employee column and EmployeePayHistory column.



Output Table:



Results | Messages

	Name	Rate	Years at Company
1	Lynn Tsoflias	23.0769	5
2	Rachel Valdez	23.0769	5
3	Syed Abbas	48.101	5
4	Tete Mensa-Annan	23.0769	6
5	Jae Pak	23.0769	6
6	Ranjit Varkey Chudukatil	23.0769	6
7	Amy Alberts	48.101	6
8	Sheela Word	9.86	7
9	Wanida Benshoof	13.4615	7
10	Mary Dempsey	13.4615	7
11	John Wood	14.4231	7

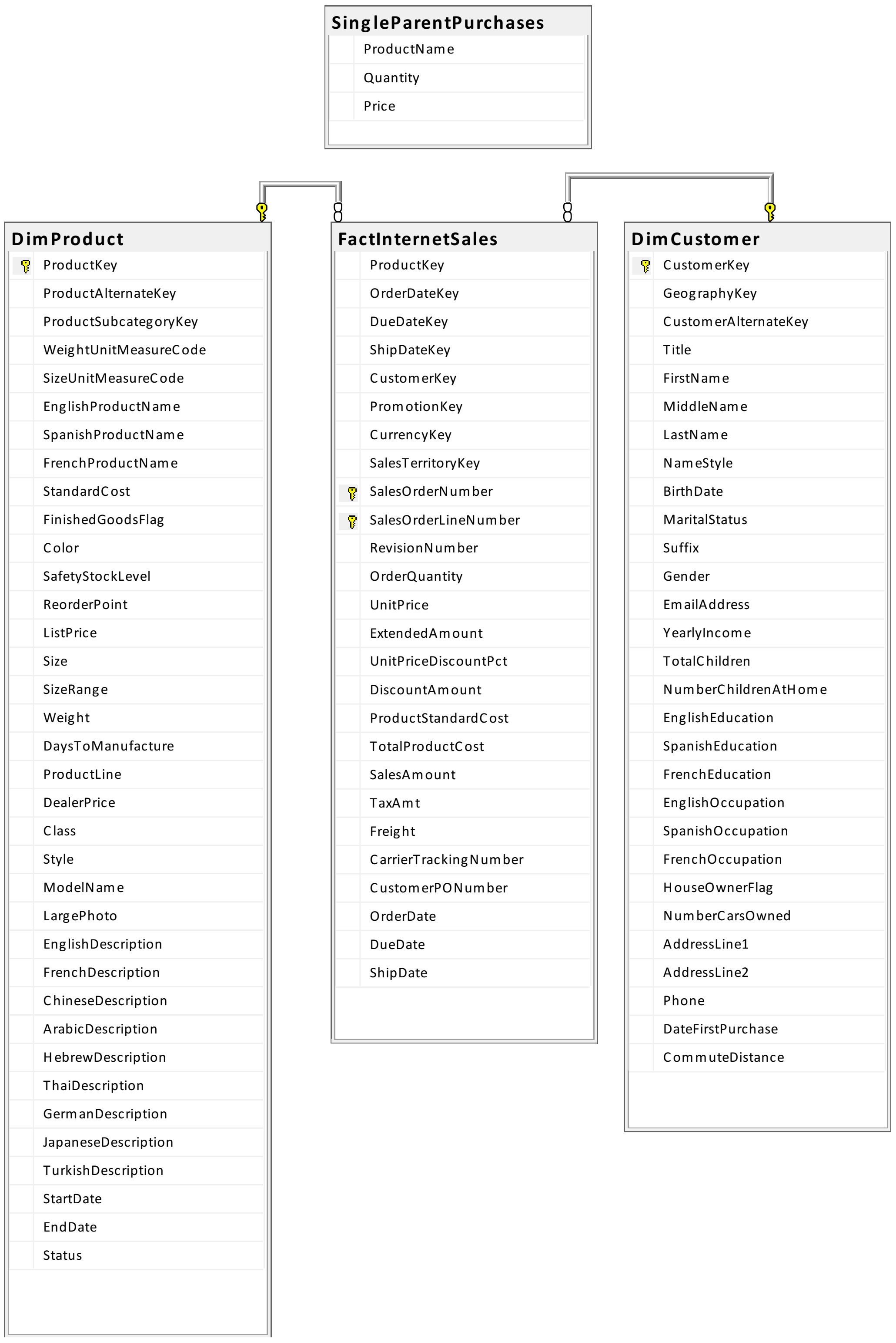
Query executed suc... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorks2014 | 00:00:00 | 316 rows

Conclusion: The minimum number of years worked by an HR employee is 5 years with a rate of $23.07. There is no discrepancy in years at the company and rate. Job title could be a better indicator of rate.

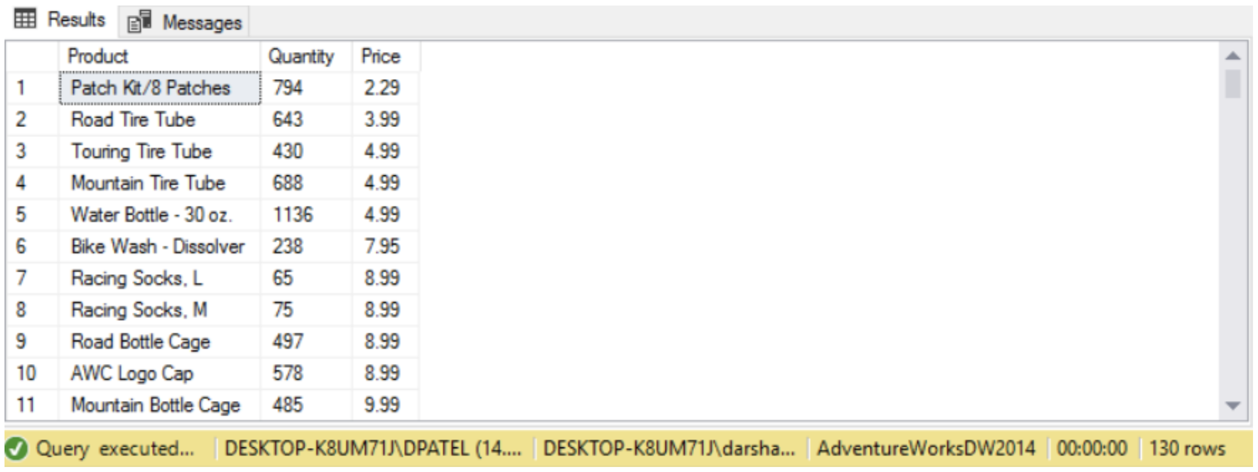
Question 17 (complex): Construct a function to create and populate a table with products purchased, the quantity, and average price a single parent has brought. Using the function and AdventureWorksDW2014, display the products, price and quantity where the latter two are in ascending order. This query can be used to determine the popularity of products single parents buy.

```
USE AdventureWorksDW2014;

DROP TABLE IF EXISTS dbo.SingleParentPurchases;
CREATE TABLE dbo.SingleParentPurchases(
        ProductName nvarchar(50) not null,
        Quantity int not null,
        Price float not null
        CONSTRAINT productname_pk PRIMARY KEY (ProductName)
);

INSERT INTO dbo.SingleParentPurchases(ProductName,Quantity,Price)
SELECT  P.EnglishProductName as [Product Name], COUNT(P.EnglishProductName) as Quantity,
    AVG(I.SalesAmount) as Price
FROM dbo.FactInternetSales as I
        INNER JOIN dbo.DimCustomer as C
                ON I.CustomerKey = C.CustomerKey
        INNER JOIN dbo.DimProduct as P
                ON I.ProductKey = P.ProductKey
WHERE C.MaritalStatus = N'S' and C.TotalChildren > 0
GROUP BY P.EnglishProductName;

SELECT ProductName as [Product], Quantity, Price
FROM dbo.SingleParentPurchases
ORDER BY Price, Quantity;
```

Database Diagram: The Product column and InternetSales column are joined by the Product key while the InternetSales column and Customer column are joined by the Customer key.



Output Table:



Results | Messages

	Product	Quantity	Price
1	Patch Kit/8 Patches	794	2.29
2	Road Tire Tube	643	3.99
3	Touring Tire Tube	430	4.99
4	Mountain Tire Tube	688	4.99
5	Water Bottle - 30 oz.	1136	4.99
6	Bike Wash - Dissolver	238	7.95
7	Racing Socks, L	65	8.99
8	Racing Socks, M	75	8.99
9	Road Bottle Cage	497	8.99
10	AWC Logo Cap	578	8.99
11	Mountain Bottle Cage	485	9.99

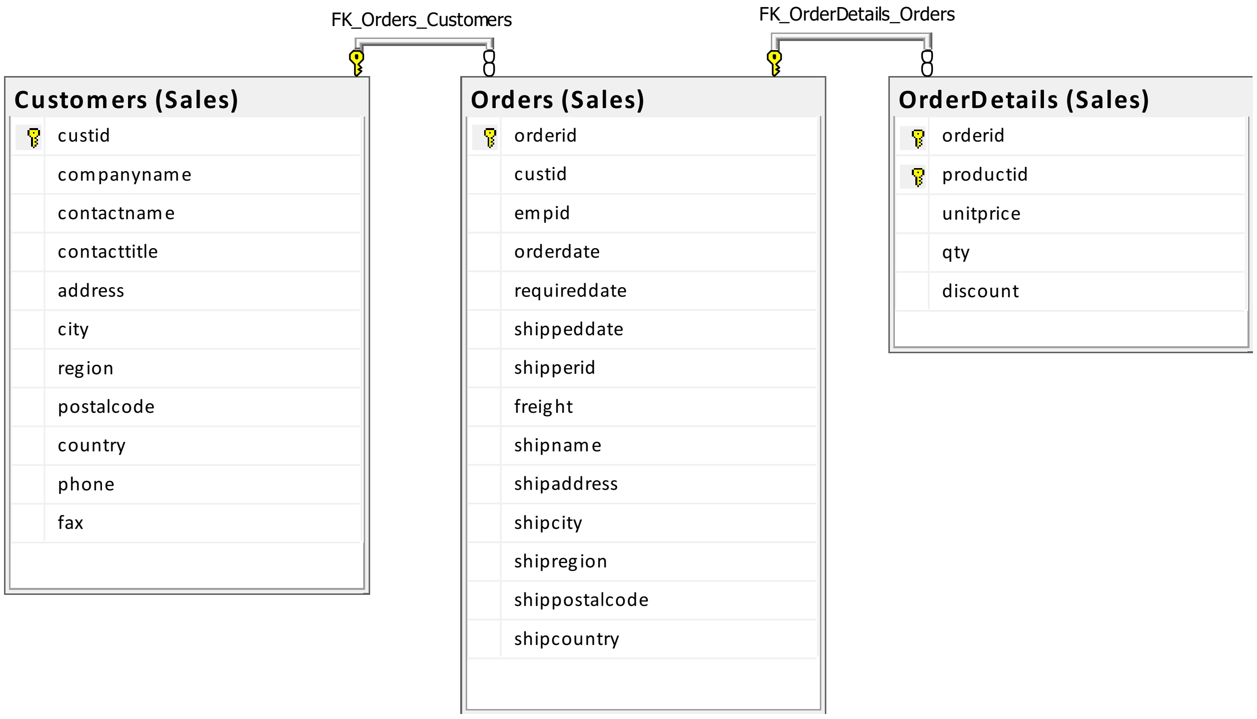
```
Query executed... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | AdventureWorksDW2014 | 00:00:00 | 130 rows
```

Conclusion: A total of 130 different products were found that single parents have purchased. An immense number of water bottles were purchased compared to other similarly priced goods.

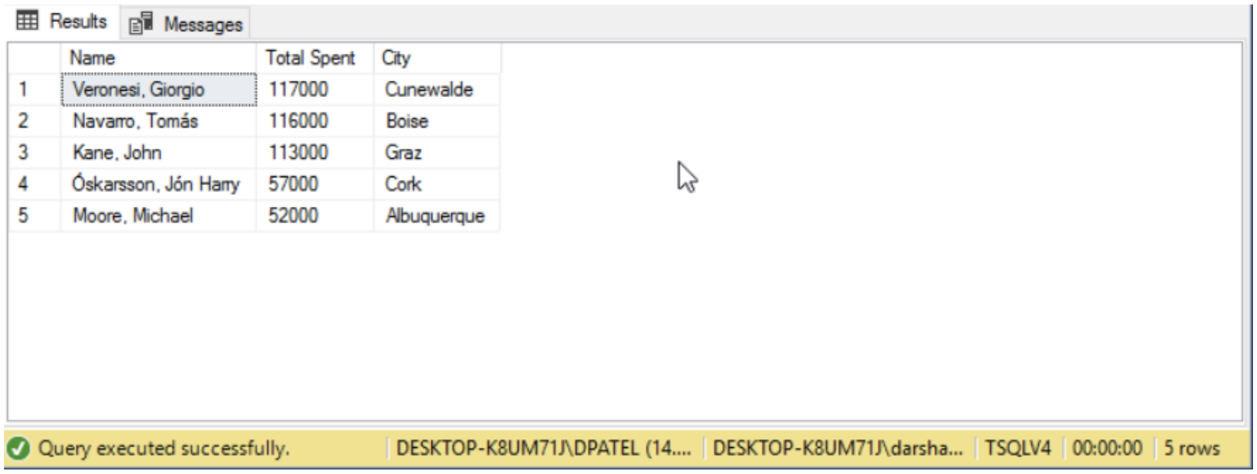
Question 18 (complex): Write a function that computes the total amount of money a customer spent. Using the function and TSQLV4, list the top 5 buyers rounded to the thousandths, the total amount they spent in descending order and their shipping city. This query can be used to determine who is spending the most money at a store and how much they're spending.

```
USE TSQLV4;

IF OBJECT_ID (N'dbo.TotalMoneySpent', N'FN') IS NOT NULL
        DROP FUNCTION TotalMoneySpent
GO
CREATE FUNCTION dbo.TotalMoneySpent(@name VARCHAR(50))
RETURNS FLOAT
AS
BEGIN
        DECLARE @total float
        SELECT @total = SUM(OD.unitprice * OD.qty)
        FROM Sales.Orders as O
        INNER JOIN Sales.OrderDetails as OD
                ON O.orderid = OD.orderid
        INNER JOIN Sales.Customers as C
                ON O.custid = C.custid
        WHERE C.contactname = @name
        RETURN @total;
END;

SELECT TOP 5 C.contactname as [Name],
    ROUND(dbo.TotalMoneySpent(C.contactname),-3) as [Total Spent], O.shipcity as City
FROM Sales.Orders as O
        INNER JOIN Sales.OrderDetails as OD
                ON O.orderid = OD.orderid
        INNER JOIN Sales.Customers as C
                ON O.custid = C.custid
GROUP BY C.contactname, O.shipcity
ORDER BY dbo.TotalMoneySpent(C.contactname) DESC;
```

Database Diagram: The Customers column and Orders column are joined by the custid key while the Orders column and OrderDetails column are joined by the orderid key.



Output Table:



	Name	Total Spent	City
1	Veronesi, Giorgio	117000	Cunewalde
2	Navarro, Tomás	116000	Boise
3	Kane, John	113000	Graz
4	Óskarsson, Jón Harry	57000	Cork
5	Moore, Michael	52000	Albuquerque

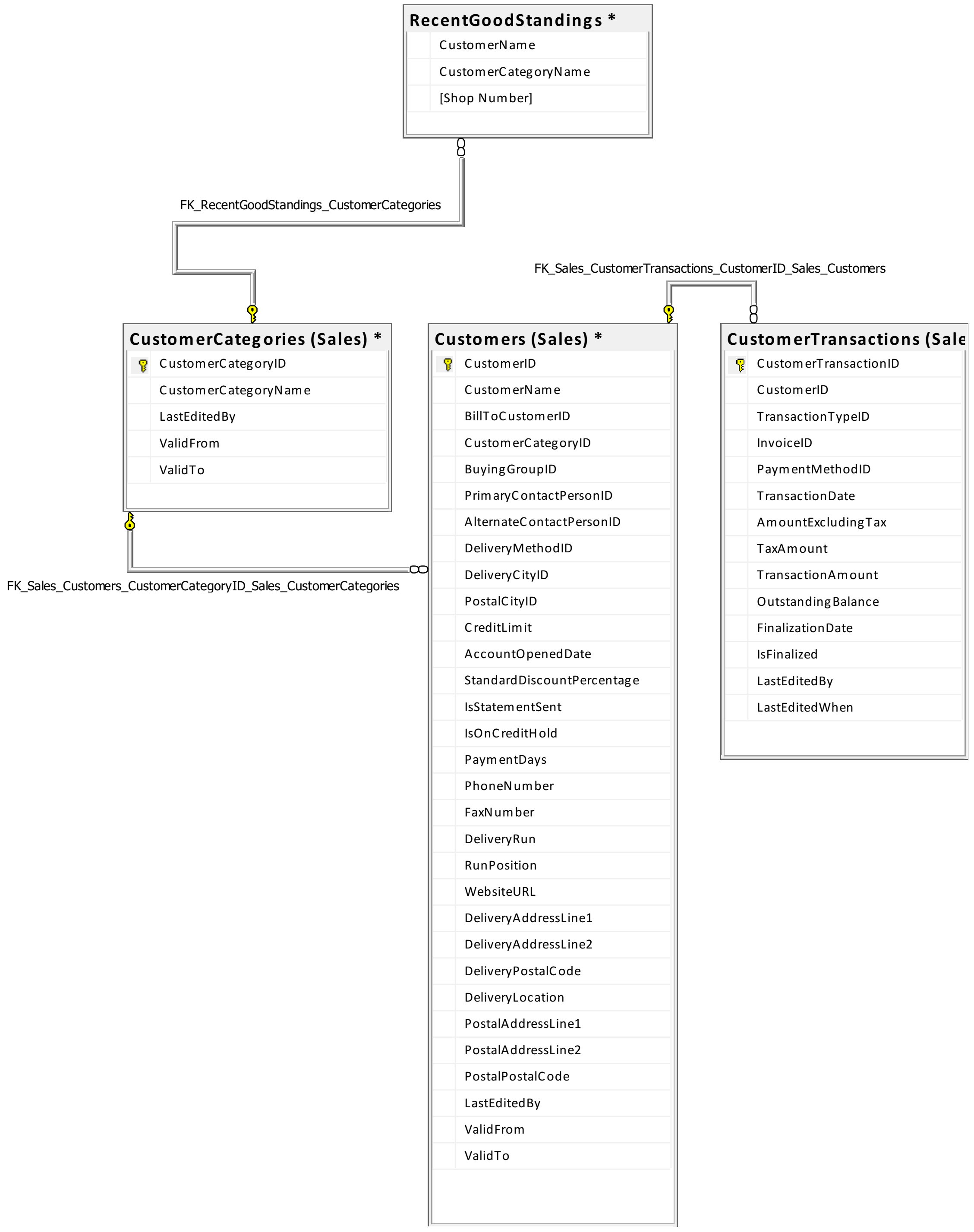
```
Query executed successfully. | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | TSQLV4 | 00:00:00 | 5 rows
```

Conclusion: The top 5 buyers spent between $52,000 and $117,000. Three of them purchases more than $100,000 amount of goods. All five customers shipped to different cities.

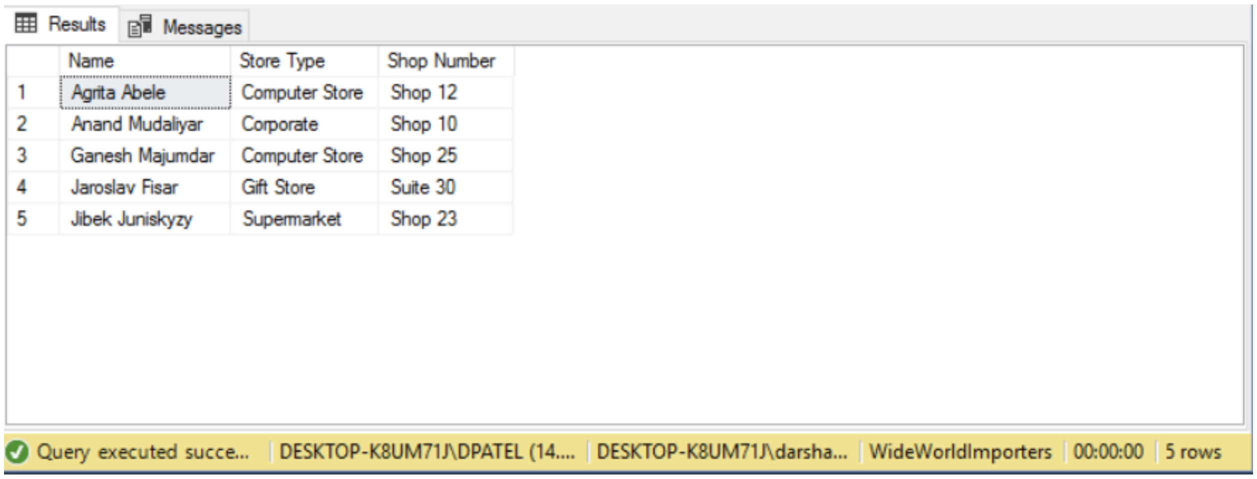
Question 19 (complex): Write a function that will create and populate a table of customers who do not go over their credit limit, their category name and shop number. Use the function and WideWorldImporters to create the table and then display it. This query can be used to indicator who is in good standings with their credit limit.

```
USE WideWorldImporters;

DROP TABLE IF EXISTS dbo.RecentGoodStandings;
CREATE TABLE dbo.RecentGoodStandings(
        CustomerName nvarchar(100) not null,
        CustomerCategoryName nvarchar(50) not null,
        ShopNumber nvarchar(50) not null
        CONSTRAINT customername_pk PRIMARY KEY (CustomerName)
);

INSERT INTO
dbo.RecentGoodStandings(CustomerName,CustomerCategoryName,ShopNumber)
SELECT DISTINCT C.CustomerName as CustomerName,
                                CC.CustomerCategoryName as CustomerCategoryName,
                                C.DeliveryAddressLine1 as ShopNumber
FROM Sales.CustomerTransactions as CT
        INNER JOIN Sales.Customers as C
                ON C.CustomerID = CT.CustomerID
        INNER JOIN Sales.CustomerCategories as CC
                ON C.CustomerCategoryID = CC.CustomerCategoryID
WHERE C.CreditLimit is not NULL and CT.TransactionAmount >=0
                and CT.TransactionAmount < C.CreditLimit
                and DATEDIFF(year, C.AccountOpenedDate,GETDATE()) <= 2
ORDER BY C.CustomerName;

SELECT CustomerName as [Name], CustomerCategoryName as [Store Type],
         ShopNumber as [Shop Number]
FROM dbo.RecentGoodStandings;
```

Database Diagram: The Customers column and CustomerCategories column are joined by the CustomerCategoryID key while the Customers column and CustomerTransactions column are joined by the CustomerID key. The newly created RecentGoodStandings column is joined to the CustomerCategories column using the CustomerCategoryID.



Output Table:



Results | Messages

	Name	Store Type	Shop Number
1	Agrita Abele	Computer Store	Shop 12
2	Anand Mudaliyar	Corporate	Shop 10
3	Ganesh Majumdar	Computer Store	Shop 25
4	Jaroslav Fisar	Gift Store	Suite 30
5	Jibek Juniskyzy	Supermarket	Shop 23

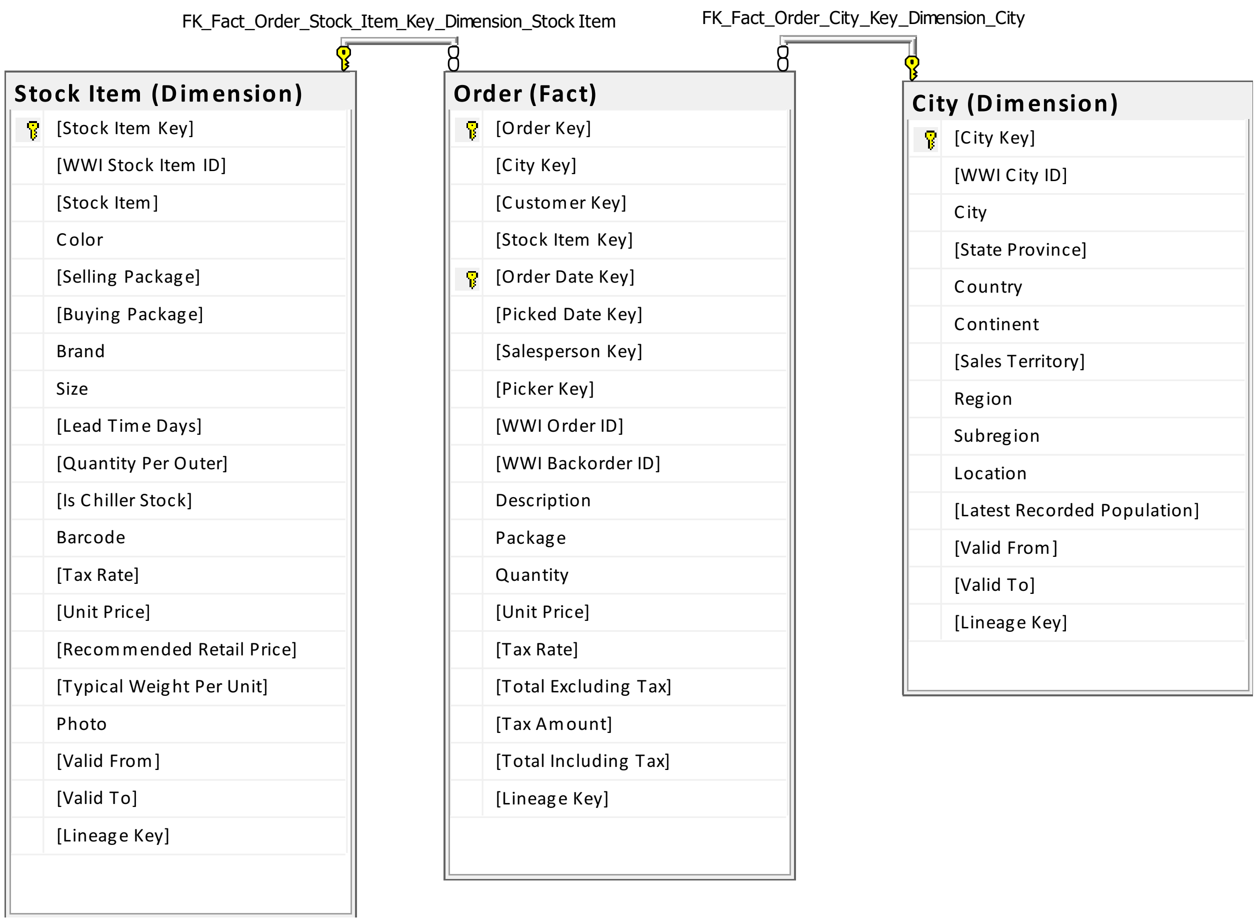
```
Query executed succe... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImporters | 00:00:00 | 5 rows
```

Conclusion: Out of all customers, there are only five customers who did not go over their credit limit. Two of them are computer stores.

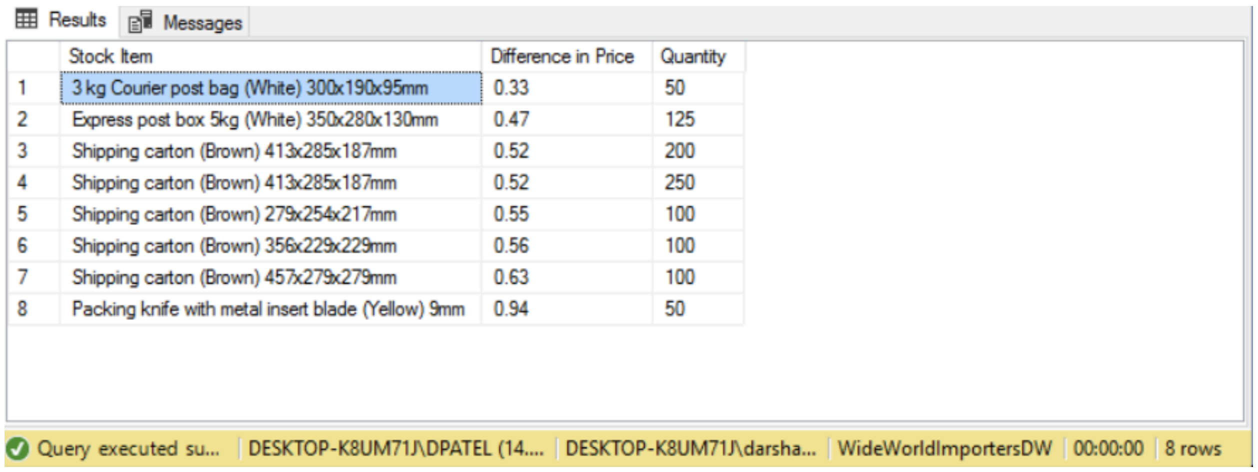
Question 20 (complex): Construct a function that will compute the difference in retail price and unit price for a certain product. Using the function and WideWorldImportersDW, list the products that differ in less than $1 from its retail and unit price and its quantity in the city Astor Park. Order by the difference. This query can be used to determine which goods are not expected to be elevated in price.

```
USE WideWorldImportersDW;

IF OBJECT_ID (N'dbo.DifferenceInPrice', N'FN') IS NOT NULL
        DROP FUNCTION DifferenceInPrice
GO
CREATE FUNCTION dbo.DifferenceInPrice(@product varchar(50))
RETURNS float
AS
BEGIN
        DECLARE @difference float
        SELECT @difference = [Recommended Retail Price] - [Unit Price]
        FROM Dimension.[Stock Item]
        WHERE [Stock Item] = @product
        RETURN @difference;
END;

SELECT SI.[Stock Item], dbo.DifferenceInPrice([Stock Item]) as [Difference in Price], O.Quantity
FROM Dimension.[Stock Item] as SI
        INNER JOIN Fact.[Order] as O
                ON SI.[Stock Item Key] = O.[Stock Item Key]
        INNER JOIN Dimension.City as C
                ON O.[City Key] = C.[City Key]
WHERE dbo.DifferenceInPrice([Stock Item]) < 1.00 and C.city = N'Astor Park'
ORDER BY dbo.DifferenceInPrice([Stock Item]), O.Quantity;
```

Database Diagram: The Stock Item column and Order column are joined by the Stock Item key while the Order column and City column are joined by the City key.



Output Table:



	Stock Item	Difference in Price	Quantity
1	3 kg Courier post bag (White) 300x190x95mm	0.33	50
2	Express post box 5kg (White) 350x280x130mm	0.47	125
3	Shipping carton (Brown) 413x285x187mm	0.52	200
4	Shipping carton (Brown) 413x285x187mm	0.52	250
5	Shipping carton (Brown) 279x254x217mm	0.55	100
6	Shipping carton (Brown) 356x229x229mm	0.56	100
7	Shipping carton (Brown) 457x279x279mm	0.63	100
8	Packing knife with metal insert blade (Yellow) 9mm	0.94	50

Query executed su... | DESKTOP-K8UM71J\DPATEL (14.... | DESKTOP-K8UM71J\darsha... | WideWorldImportersDW | 00:00:00 | 8 rows

Conclusion: Only 8 goods were found that differed in only $1 between the retail and unit price at Astor Park, with 5 of them being shipping cartons.