CSCI 331 PROJECT 1 – INDIVIDUAL PDF

PROFESSOR: PETER HELLER

SECTION: 9:15-10:30 AM

GROUP: G9-5

PDF OF: KONRAD RAKOWSKI

THIS PDF CONTAINS THE 3 TOP, 3 WORST, AND WORST CORRECTED QUERIES

CONTENTS

TOP #1: A SIMPLE QUERY

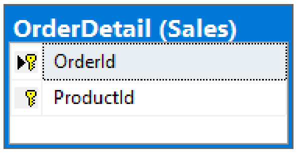
PROBLEM STATEMENT

Using Northwinds2022TSQLV, display orders with a total value less than 15000

REASON IT IS A TOP

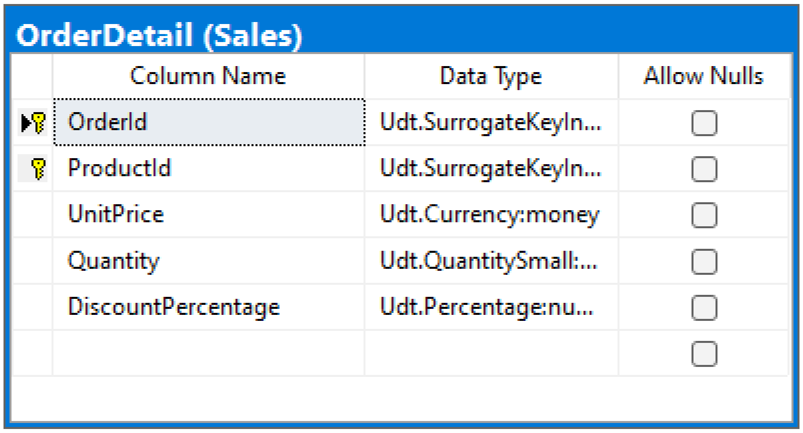
Efficient and relatively easy to read

KEY AND STANDARD VIEW OF TABLES USED



OrderDetail (Sales)

- OrderId
- ProductId



OrderDetail (Sales)

Column Name	Data Type	Allow Nulls
OrderId	Udt.SurrogateKeyIn...	□
ProductId	Udt.SurrogateKeyIn...	□
UnitPrice	Udt.Currency:money	□
Quantity	Udt.QuantitySmall:...	□
DiscountPercentage	Udt.Percentage:nu...	□
		□

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
[Sales].[OrderDetail]	OrderId
Derived	Total

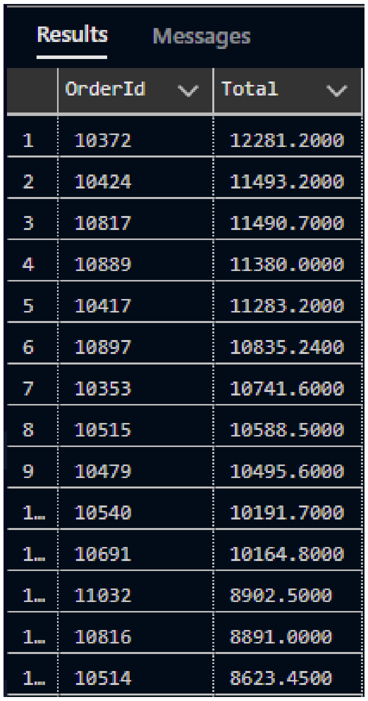
TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Not applicable

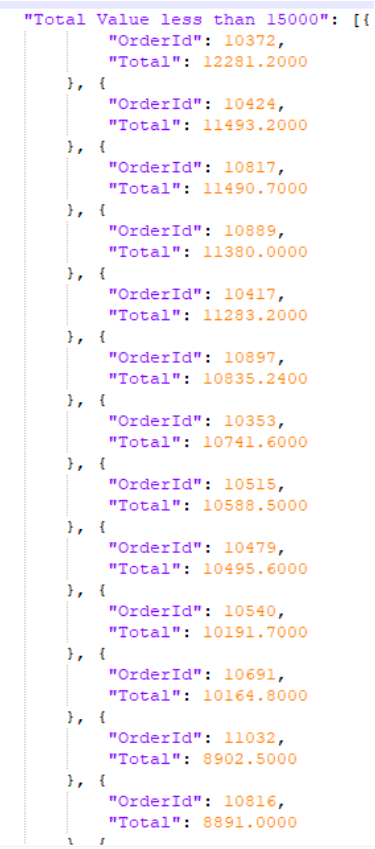
QUERY

```
USE Northwinds2022TSQLV7;
SELECT D.OrderId,
       SUM(D.Quantity * D.UnitPrice) AS Total
FROM [Sales].[OrderDetail] AS D
GROUP BY D.OrderId
HAVING SUM(D.Quantity * D.UnitPrice) < 15000
ORDER BY Total DESC;
GO
```

RELATIONAL AND JSON OUTPUT (827 ROWS AFFECTED)



	OrderId	Total
1	10372	12281.2000
2	10424	11493.2000
3	10817	11490.7000
4	10889	11380.0000
5	10417	11283.2000
6	10897	10835.2400
7	10353	10741.6000
8	10515	10588.5000
9	10479	10495.6000
1…	10540	10191.7000
1…	10691	10164.8000
1…	11032	8902.5000
1…	10816	8891.0000
1…	10514	8623.4500



```
  "Total Value less than 15000": [{
          "OrderId": 10372,
          "Total": 12281.2000
      }, {
          "OrderId": 10424,
          "Total": 11493.2000
      }, {
          "OrderId": 10817,
          "Total": 11490.7000
      }, {
          "OrderId": 10889,
          "Total": 11380.0000
      }, {
          "OrderId": 10417,
          "Total": 11283.2000
      }, {
          "OrderId": 10897,
          "Total": 10835.2400
      }, {
          "OrderId": 10353,
          "Total": 10741.6000
      }, {
          "OrderId": 10515,
          "Total": 10588.5000
      }, {
          "OrderId": 10479,
          "Total": 10495.6000
      }, {
          "OrderId": 10540,
          "Total": 10191.7000
      }, {
          "OrderId": 10691,
          "Total": 10164.8000
      }, {
          "OrderId": 11032,
          "Total": 8902.5000
      }, {
          "OrderId": 10816,
          "Total": 8891.0000
      }, {
```

TOP #2: A MEDIUM QUERY

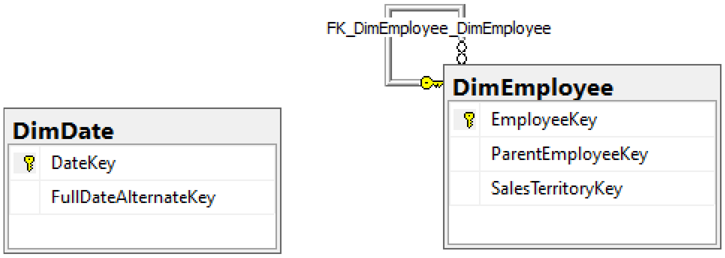
PROBLEM STATEMENT

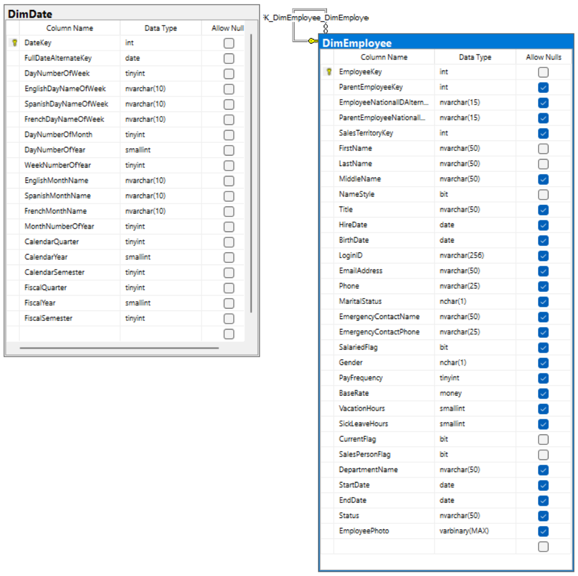
Using AdventureWorksDW2017, display every employee (oldest to youngest) who’s birthday is in July

REASON IT IS A TOP

Interesting concept and relatively easy to read as well

KEY AND STANDARD VIEW OF TABLES USED





DimDate

	Column Name	Data Type	Allow Null
🔑	DateKey	int	☐
	FullDateAlternateKey	date	☐
	DayNumberOfWeek	tinyint	☐
	EnglishDayNameOfWeek	nvarchar(10)	☐
	SpanishDayNameOfWeek	nvarchar(10)	☐
	FrenchDayNameOfWeek	nvarchar(10)	☐
	DayNumberOfMonth	tinyint	☐
	DayNumberOfYear	smallint	☐
	WeekNumberOfYear	tinyint	☐
	EnglishMonthName	nvarchar(10)	☐
	SpanishMonthName	nvarchar(10)	☐
	FrenchMonthName	nvarchar(10)	☐
	MonthNumberOfYear	tinyint	☐
	CalendarQuarter	tinyint	☐
	CalendarYear	smallint	☐
	CalendarSemester	tinyint	☐
	FiscalQuarter	tinyint	☐
	FiscalYear	smallint	☐
	FiscalSemester	tinyint	☐
			☐

DimEmployee

	Column Name	Data Type	Allow Nulls
🔑	EmployeeKey	int	☐
	ParentEmployeeKey	int	☑
	EmployeeNationalIDAltern...	nvarchar(15)	☑
	ParentEmployeeNationalI...	nvarchar(15)	☑
	SalesTerritoryKey	int	☑
	FirstName	nvarchar(50)	☐
	LastName	nvarchar(50)	☐
	MiddleName	nvarchar(50)	☑
	NameStyle	bit	☐
	Title	nvarchar(50)	☑
	HireDate	date	☑
	BirthDate	date	☑
	LoginID	nvarchar(256)	☑
	EmailAddress	nvarchar(50)	☑
	Phone	nvarchar(25)	☑
	MaritalStatus	nchar(1)	☑
	EmergencyContactName	nvarchar(50)	☑
	EmergencyContactPhone	nvarchar(25)	☑
	SalariedFlag	bit	☑
	Gender	nchar(1)	☑
	PayFrequency	tinyint	☑
	BaseRate	money	☑
	VacationHours	smallint	☑
	SickLeaveHours	smallint	☑
	CurrentFlag	bit	☐
	SalesPersonFlag	bit	☐
	DepartmentName	nvarchar(50)	☑
	StartDate	date	☑
	EndDate	date	☑
	Status	nvarchar(50)	☑
	EmployeePhoto	varbinary(MAX)	☑
			☐

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
[dbo].[DimDate]	EnglishMonthName
[dbo].[DimEmployee]	LastName FirstName
Derived	Day Year

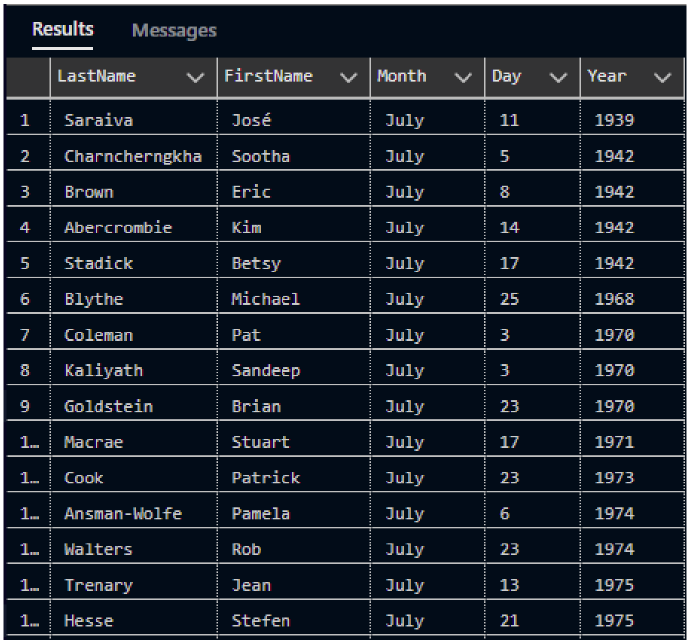
TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Derived	Year Day	ASC ASC

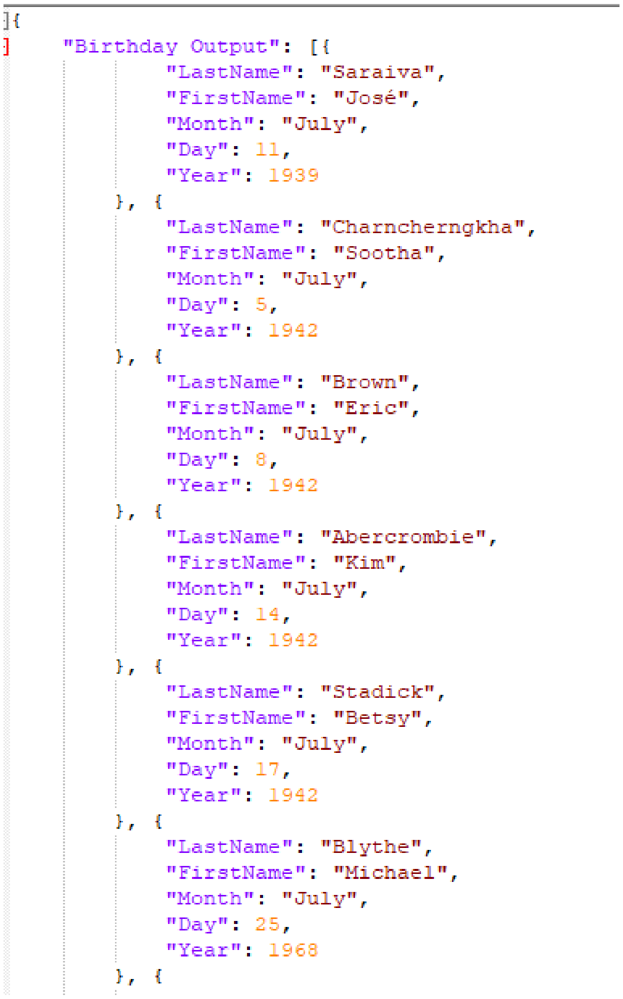
QUERY

```
USE AdventureWorksDW2017;
WITH MonthName
AS (SELECT D.EnglishMonthName AS Month,
           D.MonthNumberOfYear AS MonthNum
    FROM [dbo].[DimDate] AS D
    GROUP BY D.EnglishMonthName,
             D.MonthNumberOfYear)
SELECT DISTINCT
       E.LastName,
       E.FirstName,
       D.Month,
       DAY(E.BirthDate) AS Day,
       YEAR(E.BirthDate) AS Year
FROM [dbo].[DimEmployee] AS E
    JOIN MonthName AS D
        ON MONTH(E.BirthDate) = D.MonthNum
WHERE MONTH(E.BirthDate) = 7
ORDER BY Year,
         Day;
GO
```

RELATIONAL AND JSON OUTPUT (28 ROWS AFFECTED)



	LastName	FirstName	Month	Day	Year
1	Saraiva	José	July	11	1939
2	Charncherngkha	Sootha	July	5	1942
3	Brown	Eric	July	8	1942
4	Abercrombie	Kim	July	14	1942
5	Stadick	Betsy	July	17	1942
6	Blythe	Michael	July	25	1968
7	Coleman	Pat	July	3	1970
8	Kaliyath	Sandeep	July	3	1970
9	Goldstein	Brian	July	23	1970
1…	Macrae	Stuart	July	17	1971
1…	Cook	Patrick	July	23	1973
1…	Ansman-Wolfe	Pamela	July	6	1974
1…	Walters	Rob	July	23	1974
1…	Trenary	Jean	July	13	1975
1…	Hesse	Stefen	July	21	1975



```
{
    "Birthday Output": [{
            "LastName": "Saraiva",
            "FirstName": "José",
            "Month": "July",
            "Day": 11,
            "Year": 1939
        }, {
            "LastName": "Charncherngkha",
            "FirstName": "Sootha",
            "Month": "July",
            "Day": 5,
            "Year": 1942
        }, {
            "LastName": "Brown",
            "FirstName": "Eric",
            "Month": "July",
            "Day": 8,
            "Year": 1942
        }, {
            "LastName": "Abercrombie",
            "FirstName": "Kim",
            "Month": "July",
            "Day": 14,
            "Year": 1942
        }, {
            "LastName": "Stadick",
            "FirstName": "Betsy",
            "Month": "July",
            "Day": 17,
            "Year": 1942
        }, {
            "LastName": "Blythe",
            "FirstName": "Michael",
            "Month": "July",
            "Day": 25,
            "Year": 1968
        }, {
```

TOP #3: A COMPLEX QUERY

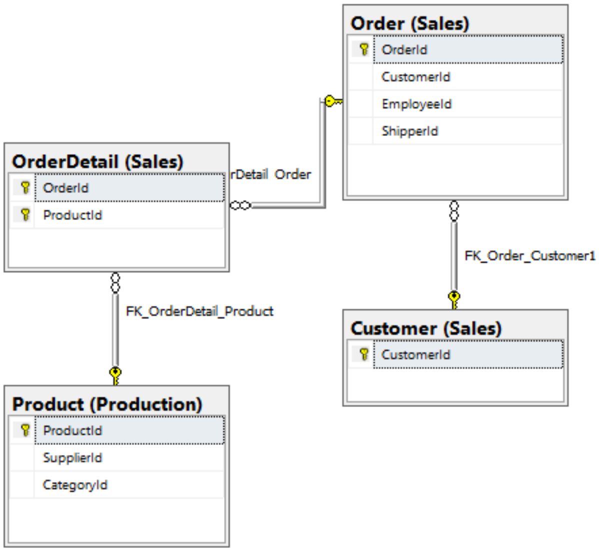
PROBLEM STATEMENT

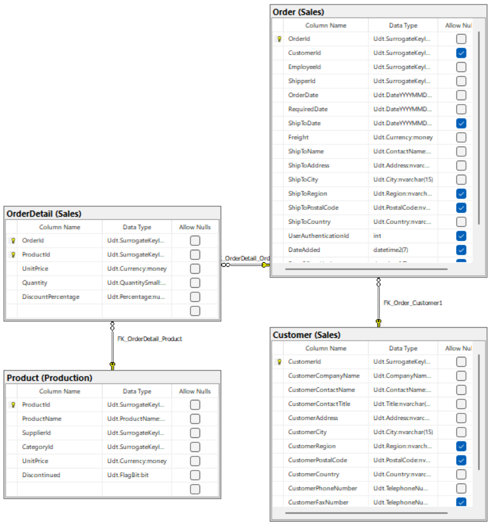
Using Northwinds2022TSQLV7, display the information of customers who purchased discontinued items

REASON IT IS A TOP

Separates the main code from the function cleanly, allowing for better readability

KEY AND STANDARD VIEW OF TABLES USED





Order (Sales)

	Column Name	Data Type	Allow Nul
🔑	OrderId	Udt.SurrogateKeyI...	☐
	CustomerId	Udt.SurrogateKeyI...	☑
	EmployeeId	Udt.SurrogateKeyI...	☐
	ShipperId	Udt.SurrogateKeyI...	☐
	OrderDate	Udt.DateYYYYMMD...	☐
	RequiredDate	Udt.DateYYYYMMD...	☐
	ShipToDate	Udt.DateYYYYMMD...	☑
	Freight	Udt.Currency:money	☐
	ShipToName	Udt.ContactName:...	☐
	ShipToAddress	Udt.Address:nvarc...	☐
	ShipToCity	Udt.City:nvarchar(15)	☐
	ShipToRegion	Udt.Region:nvarch...	☑
	ShipToPostalCode	Udt.PostalCode:nv...	☑
	ShipToCountry	Udt.Country:nvarc...	☐
	UserAuthenticationId	int	☑
	DateAdded	datetime2(7)	☑

OrderDetail (Sales)

	Column Name	Data Type	Allow Nulls
🔑	OrderId	Udt.SurrogateKeyI...	☐
🔑	ProductId	Udt.SurrogateKeyI...	☐
	UnitPrice	Udt.Currency:money	☐
	Quantity	Udt.QuantitySmall:...	☐
	DiscountPercentage	Udt.Percentage:nu...	☐
			☐

Product (Production)

	Column Name	Data Type	Allow Nulls
🔑	ProductId	Udt.SurrogateKeyI...	☐
	ProductName	Udt.ProductName:...	☐
	SupplierId	Udt.SurrogateKeyI...	☐
	CategoryId	Udt.SurrogateKeyI...	☐
	UnitPrice	Udt.Currency:money	☐
	Discontinued	Udt.FlagBit:bit	☐
			☐

Customer (Sales)

	Column Name	Data Type	Allow Nul
🔑	CustomerId	Udt.SurrogateKeyI...	☐
	CustomerCompanyName	Udt.CompanyNam...	☐
	CustomerContactName	Udt.ContactName:...	☐
	CustomerContactTitle	Udt.Title:nvarchar(...	☐
	CustomerAddress	Udt.Address:nvarc...	☐
	CustomerCity	Udt.City:nvarchar(15)	☐
	CustomerRegion	Udt.Region:nvarch...	☑
	CustomerPostalCode	Udt.PostalCode:nv...	☑
	CustomerCountry	Udt.Country:nvarc...	☐
	CustomerPhoneNumber	Udt.TelephoneNu...	☐
	CustomerFaxNumber	Udt.TelephoneNu...	☑

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
[Production].[Product]	ProductId
[Sales].[Order]	OrderId, OrderDate
[Sales].[Customer]	CustomerCompanyName
Derived	Paid

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

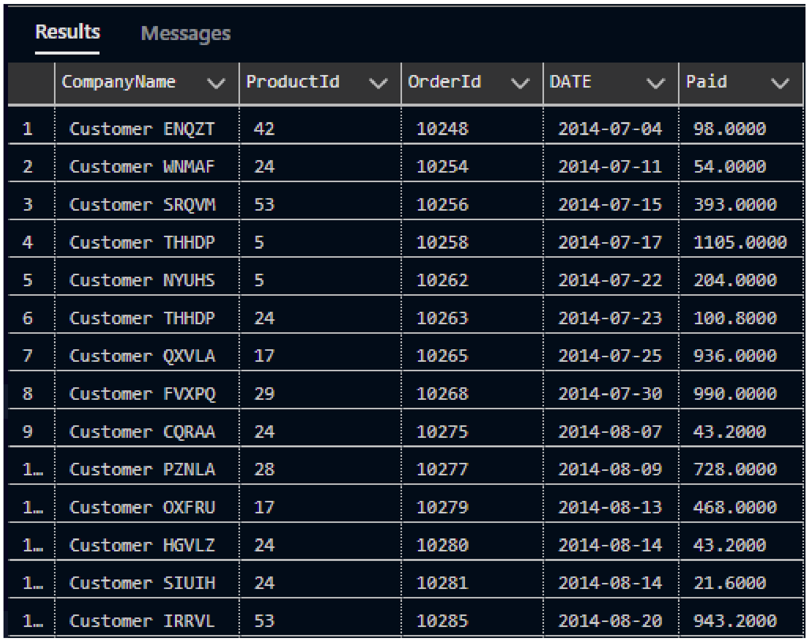
Not applicable

QUERY

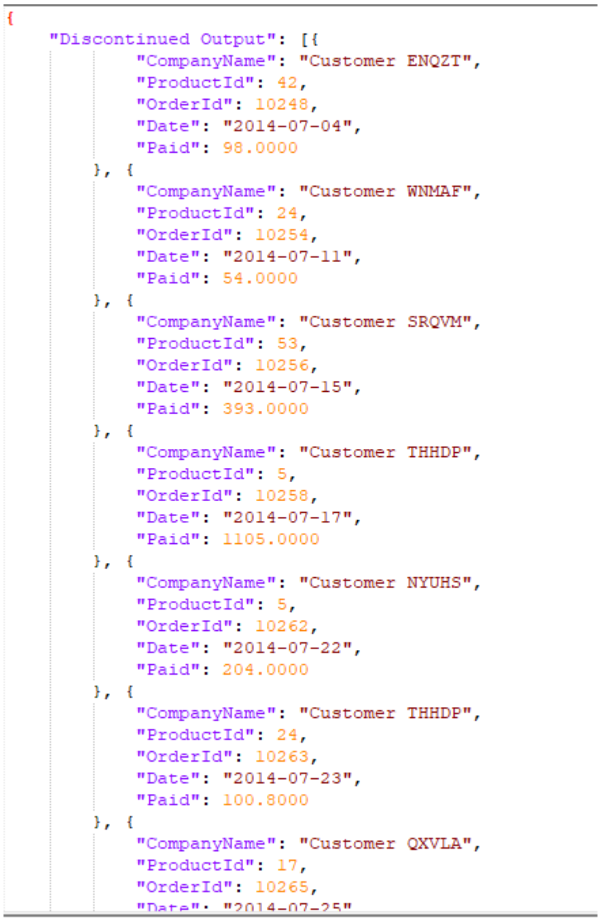
```
USE Northwinds2022TSQLV7;

DROP FUNCTION IF EXISTS getDiscontinued;
GO
CREATE FUNCTION getDiscontinued
(
    @disc AS INT
)
RETURNS TABLE
AS
RETURN SELECT ProductId,
              ProductName,
              SupplierId,
              CategoryId,
              UnitPrice,
              Discontinued
       FROM [Production].[Product]
       WHERE Discontinued = @disc;
GO
WITH Ord
AS (SELECT O.OrderId,
           O.OrderDate AS DATE,
           O.CustomerId
    FROM [Sales].[Order] AS O
    GROUP BY O.OrderId,
             O.OrderDate,
             O.CustomerId),
     Ord2
AS (SELECT C.CustomerCompanyName AS CompanyName,
           C.CustomerId
    FROM [Sales].[Customer] AS C
    GROUP BY C.CustomerCompanyName,
             C.CustomerId)
SELECT C.CompanyName,
       P.ProductId,
       O.OrderId,
       O.DATE,
       SUM(D.UnitPrice * D.Quantity) AS Paid
FROM Ord2 AS C
    JOIN Ord AS O
        ON C.CustomerId = O.CustomerId
    JOIN [Sales].[OrderDetail] AS D
        ON O.OrderId = D.OrderId
    JOIN getDiscontinued(1) AS P
        ON D.ProductId = P.ProductId
GROUP BY C.CompanyName,
         P.ProductId,
         O.OrderId,
         O.DATE;
```

RELATIONAL AND JSON OUTPUT (228 ROWS AFFECTED)



	CompanyName	ProductId	OrderId	DATE	Paid
1	Customer ENQZT	42	10248	2014-07-04	98.0000
2	Customer WNMAF	24	10254	2014-07-11	54.0000
3	Customer SRQVM	53	10256	2014-07-15	393.0000
4	Customer THHDP	5	10258	2014-07-17	1105.0000
5	Customer NYUHS	5	10262	2014-07-22	204.0000
6	Customer THHDP	24	10263	2014-07-23	100.8000
7	Customer QXVLA	17	10265	2014-07-25	936.0000
8	Customer FVXPQ	29	10268	2014-07-30	990.0000
9	Customer CQRAA	24	10275	2014-08-07	43.2000
1…	Customer PZNLA	28	10277	2014-08-09	728.0000
1…	Customer OXFRU	17	10279	2014-08-13	468.0000
1…	Customer HGVLZ	24	10280	2014-08-14	43.2000
1…	Customer SIUIH	24	10281	2014-08-14	21.6000
1…	Customer IRRVL	53	10285	2014-08-20	943.2000



```
{
    "Discontinued Output": [{
            "CompanyName": "Customer ENQZT",
            "ProductId": 42,
            "OrderId": 10248,
            "Date": "2014-07-04",
            "Paid": 98.0000
        }, {
            "CompanyName": "Customer WNMAF",
            "ProductId": 24,
            "OrderId": 10254,
            "Date": "2014-07-11",
            "Paid": 54.0000
        }, {
            "CompanyName": "Customer SRQVM",
            "ProductId": 53,
            "OrderId": 10256,
            "Date": "2014-07-15",
            "Paid": 393.0000
        }, {
            "CompanyName": "Customer THHDP",
            "ProductId": 5,
            "OrderId": 10258,
            "Date": "2014-07-17",
            "Paid": 1105.0000
        }, {
            "CompanyName": "Customer NYUHS",
            "ProductId": 5,
            "OrderId": 10262,
            "Date": "2014-07-22",
            "Paid": 204.0000
        }, {
            "CompanyName": "Customer THHDP",
            "ProductId": 24,
            "OrderId": 10263,
            "Date": "2014-07-23",
            "Paid": 100.8000
        }, {
            "CompanyName": "Customer QXVLA",
            "ProductId": 17,
            "OrderId": 10265,
            "Date": "2014-07-25"
```

WORST #1: A SIMPLE QUERY

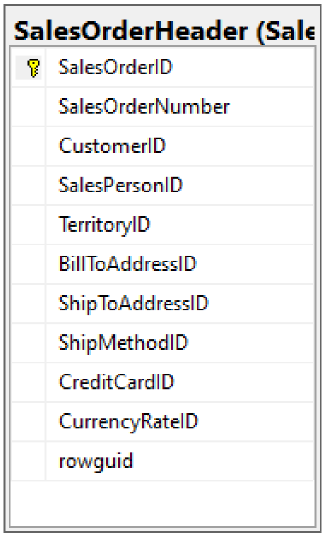
PROBLEM STATEMENT

Using AdventureWorks2017, display the order from the highest orderid

REASON IT IS A WORST

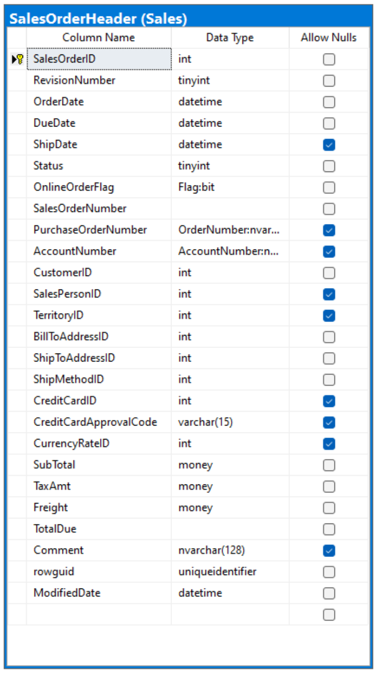
Can be written much easier and have a simpler output

KEY AND STANDARD VIEW OF TABLES USED



SalesOrderHeader (Sale

- SalesOrderID
- SalesOrderNumber
- CustomerID
- SalesPersonID
- TerritoryID
- BillToAddressID
- ShipToAddressID
- ShipMethodID
- CreditCardID
- CurrencyRateID
- rowguid



SalesOrderHeader (Sales)

Column Name	Data Type	Allow Nulls
SalesOrderID	int	□
RevisionNumber	tinyint	□
OrderDate	datetime	□
DueDate	datetime	□
ShipDate	datetime	☑
Status	tinyint	□
OnlineOrderFlag	Flag:bit	□
SalesOrderNumber		□
PurchaseOrderNumber	OrderNumber:nvar...	☑
AccountNumber	AccountNumber:n...	☑
CustomerID	int	□
SalesPersonID	int	☑
TerritoryID	int	☑
BillToAddressID	int	□
ShipToAddressID	int	□
ShipMethodID	int	□
CreditCardID	int	☑
CreditCardApprovalCode	varchar(15)	☑
CurrencyRateID	int	☑
SubTotal	money	□
TaxAmt	money	□
Freight	money	□
TotalDue		□
Comment	nvarchar(128)	☑
rowguid	uniqueidentifier	□
ModifiedDate	datetime	□
		□

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
[Sales].[SalesOrderHeader]	OrderDate, CustomerId
Derived	MAX(SalesOrderId)

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Derived	MAX(SalesOrderId)	DESC

QUERY OF WORST

```
USE AdventureWorks2017;

SELECT MAX(H.SalesOrderID),
       H.OrderDate,
       H.CustomerID
FROM [Sales].[SalesOrderHeader] AS H
GROUP BY H.OrderDate,
         H.CustomerID
ORDER BY MAX(H.SalesOrderID) DESC;
```

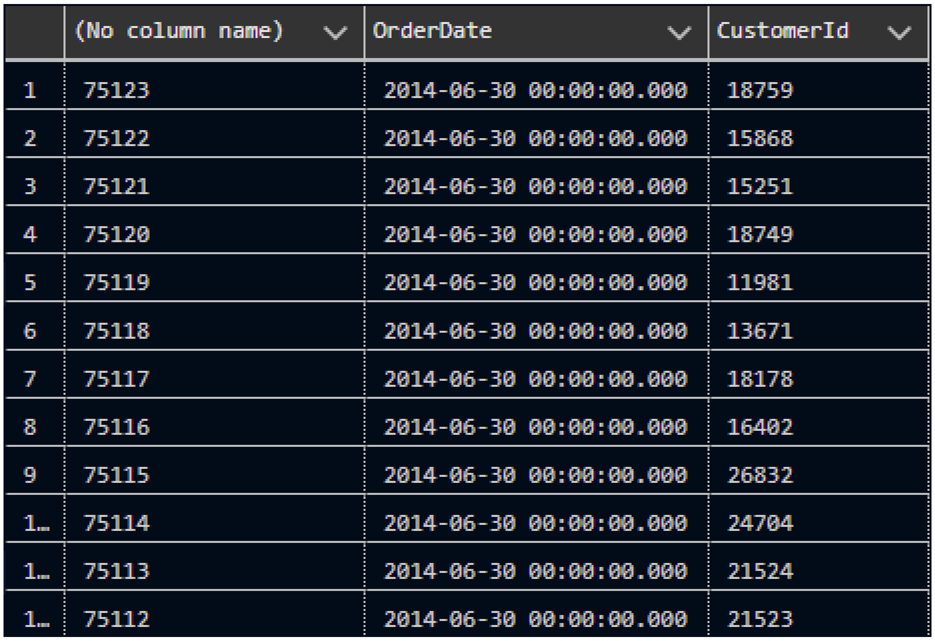
QUERY OF WORST CORRECTED

```
USE AdventureWorks2017;
SELECT H.SalesOrderID,
       H.OrderDate,
       H.CustomerID
FROM [Sales].[SalesOrderHeader] AS H
WHERE H.SalesOrderID =
(
    SELECT MAX(H2.SalesOrderID)FROM [Sales].[SalesOrderHeader] AS H2
);
GO
```

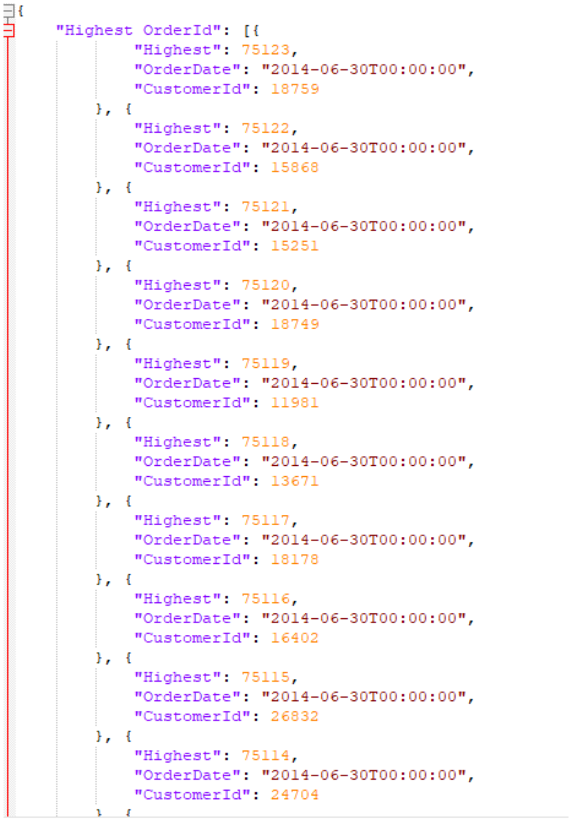
HOW IT WAS CORRECTED:

Used a subquery to allow the maximum ordered to only be outputted

RELATIONAL AND JSON OUTPUT (31406 ROWS AFFECTED)

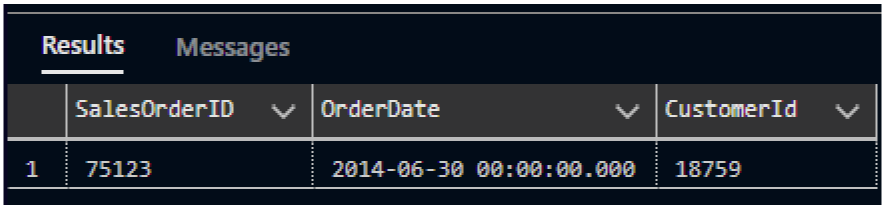


	(No column name)	OrderDate	CustomerId
1	75123	2014-06-30 00:00:00.000	18759
2	75122	2014-06-30 00:00:00.000	15868
3	75121	2014-06-30 00:00:00.000	15251
4	75120	2014-06-30 00:00:00.000	18749
5	75119	2014-06-30 00:00:00.000	11981
6	75118	2014-06-30 00:00:00.000	13671
7	75117	2014-06-30 00:00:00.000	18178
8	75116	2014-06-30 00:00:00.000	16402
9	75115	2014-06-30 00:00:00.000	26832
1…	75114	2014-06-30 00:00:00.000	24704
1…	75113	2014-06-30 00:00:00.000	21524
1…	75112	2014-06-30 00:00:00.000	21523

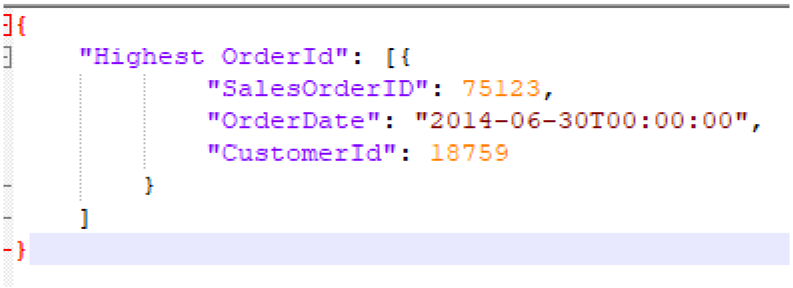


```
{
    "Highest OrderId": [{
            "Highest": 75123,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 18759
        }, {
            "Highest": 75122,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 15868
        }, {
            "Highest": 75121,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 15251
        }, {
            "Highest": 75120,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 18749
        }, {
            "Highest": 75119,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 11981
        }, {
            "Highest": 75118,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 13671
        }, {
            "Highest": 75117,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 18178
        }, {
            "Highest": 75116,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 16402
        }, {
            "Highest": 75115,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 26832
        }, {
            "Highest": 75114,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 24704
        }, {
```

CORRECTED RELATIONAL AND JSON OUTPUT (1 ROW AFFECTED)



	SalesOrderID	OrderDate	CustomerId
1	75123	2014-06-30 00:00:00.000	18759



```
{
    "Highest OrderId": [{
            "SalesOrderID": 75123,
            "OrderDate": "2014-06-30T00:00:00",
            "CustomerId": 18759
        }
    ]
}
```

WORST #2: A MEDIUM QUERY

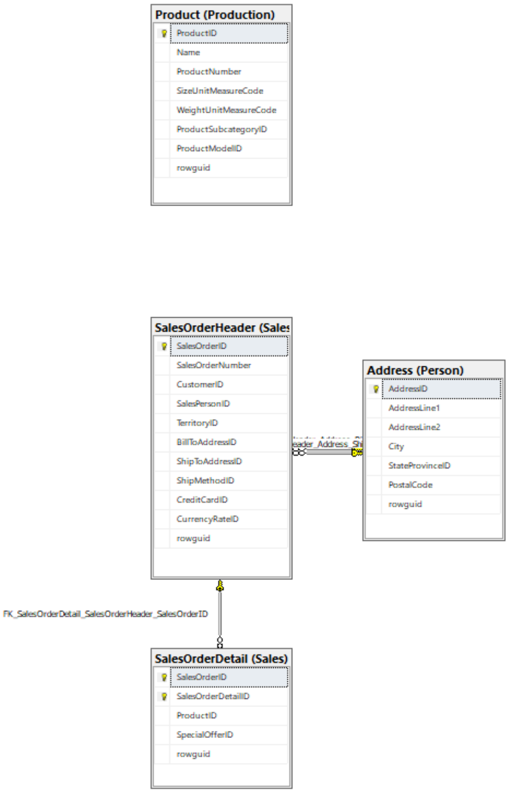
PROBLEM STATEMENT

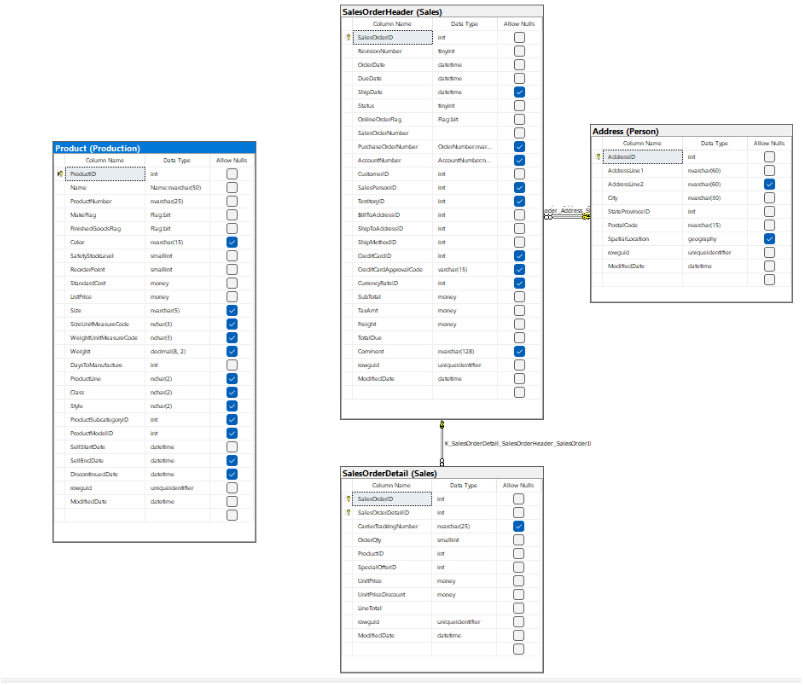
Using AdventureWorks2017, display how many ‘silver’ products have been sold to an address that is in ‘New York’ (city)

REASON IT IS A WORST

Can create an extension CTE to make the main a little cleaner

KEY AND STANDARD VIEW OF TABLES USED





Product (Production)

Column Name	Data Type	Allow Nulls
ProductID	int	☐
Name	Name:nvarchar(50)	☐
ProductNumber	nvarchar(25)	☐
MakeFlag	Flag:bit	☐
FinishedGoodsFlag	Flag:bit	☐
Color	nvarchar(15)	☑
SafetyStockLevel	smallint	☐
ReorderPoint	smallint	☐
StandardCost	money	☐
ListPrice	money	☐
Size	nvarchar(5)	☑
SizeUnitMeasureCode	nchar(3)	☑
WeightUnitMeasureCode	nchar(3)	☑
Weight	decimal(8, 2)	☑
DaysToManufacture	int	☐
ProductLine	nchar(2)	☑
Class	nchar(2)	☑
Style	nchar(2)	☑
ProductSubcategoryID	int	☑
ProductModelID	int	☑
SellStartDate	datetime	☐
SellEndDate	datetime	☑
DiscontinuedDate	datetime	☑
rowguid	uniqueidentifier	☐
ModifiedDate	datetime	☐
		☐

SalesOrderHeader (Sales)

Column Name	Data Type	Allow Nulls
SalesOrderID	int	☐
RevisionNumber	tinyint	☐
OrderDate	datetime	☐
DueDate	datetime	☐
ShipDate	datetime	☑
Status	tinyint	☐
OnlineOrderFlag	Flag:bit	☐
SalesOrderNumber		☐
PurchaseOrderNumber	OrderNumber:nvar...	☑
AccountNumber	AccountNumber:n...	☑
CustomerID	int	☐
SalesPersonID	int	☑
TerritoryID	int	☑
BillToAddressID	int	☐
ShipToAddressID	int	☐
ShipMethodID	int	☐
CreditCardID	int	☑
CreditCardApprovalCode	varchar(15)	☑
CurrencyRateID	int	☑
SubTotal	money	☐
TaxAmt	money	☐
Freight	money	☐
TotalDue		☐
Comment	nvarchar(128)	☑
rowguid	uniqueidentifier	☐
ModifiedDate	datetime	☐
		☐

Address (Person)

Column Name	Data Type	Allow Nulls
AddressID	int	☐
AddressLine1	nvarchar(60)	☐
AddressLine2	nvarchar(60)	☑
City	nvarchar(30)	☐
StateProvinceID	int	☐
PostalCode	nvarchar(15)	☐
SpatialLocation	geography	☑
rowguid	uniqueidentifier	☐
ModifiedDate	datetime	☐
		☐

SalesOrderDetail (Sales)

Column Name	Data Type	Allow Nulls
SalesOrderID	int	☐
SalesOrderDetailID	int	☐
CarrierTrackingNumber	nvarchar(25)	☑
OrderQty	smallint	☐
ProductID	int	☐
SpecialOfferID	int	☐
UnitPrice	money	☐
UnitPriceDiscount	money	☐
LineTotal		☐
rowguid	uniqueidentifier	☐
ModifiedDate	datetime	☐
		☐

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Derived	SilverTotal

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Not applicable

QUERY OF WORST

```
USE AdventureWorks2017;
WITH Product
AS (SELECT P.ProductID,
           P.Color
    FROM [Production].[Product] AS P
    WHERE P.Color = 'silver')
SELECT SUM(D.OrderQty) AS SilverTotal
FROM Product AS P
    JOIN [Sales].[SalesOrderDetail] AS D
        ON P.ProductID = D.ProductID
    JOIN [Sales].[SalesOrderHeader] AS H
        ON D.SalesOrderID = H.SalesOrderID
    JOIN [Person].[Address] AS A
        ON H.ShipToAddressID = A.AddressID
WHERE A.City = 'New York';
GO
```

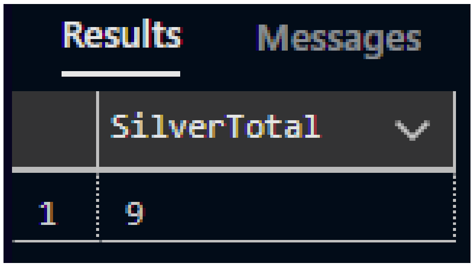
QUERY OF WORST CORRECTED

```sql
USE AdventureWorks2017;
WITH Product
AS (SELECT P.ProductID,
           P.Color
    FROM [Production].[Product] AS P
    WHERE P.Color = 'silver'),
     City
AS (SELECT A.AddressID,
           A.City
    FROM [Person].[Address] AS A
    WHERE A.City = 'New York')
SELECT SUM(D.OrderQty) AS SilverTotal
FROM Product AS P
    JOIN [Sales].[SalesOrderDetail] AS D
        ON P.ProductID = D.ProductID
    JOIN [Sales].[SalesOrderHeader] AS H
        ON D.SalesOrderID = H.SalesOrderID
    JOIN City AS A
        ON H.ShipToAddressID = A.AddressID;
GO
```

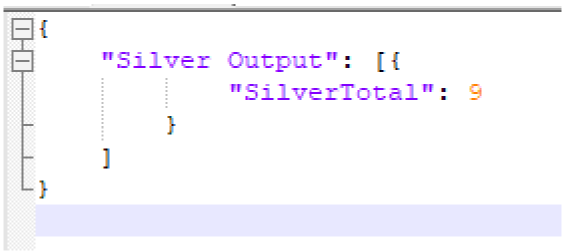
HOW IT WAS CORRECTED:

Created an extension to the CTE to clear up the main code

RELATIONAL AND JSON OUTPUT (1 ROW AFFECTED)



	SilverTotal
1	9



```
{
    "Silver Output": [{
            "SilverTotal": 9
        }
    ]
}
```

WORST #3: A COMPLEX QUERY

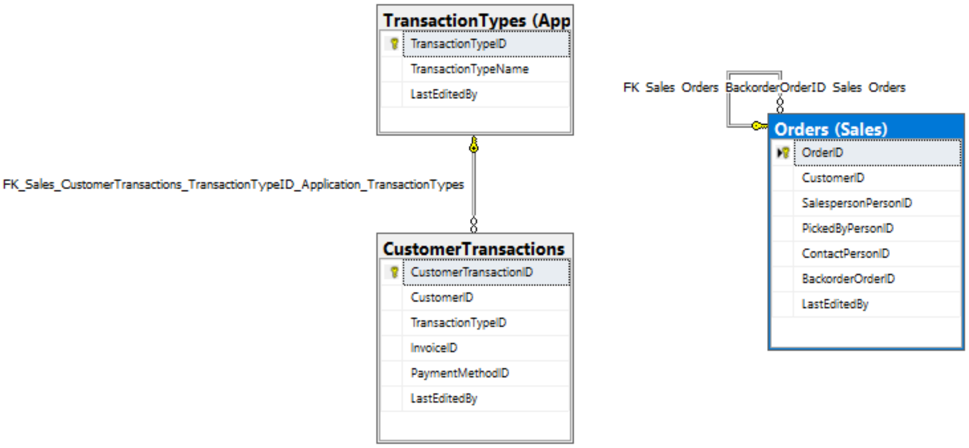
PROBLEM STATEMENT

Using WideWorldImporters, display the transaction id of every customer who ordered where the payment was received at the end of every month in 2015

REASON IT IS A WORST

Too much unnecessary information in the custom function

KEY AND STANDARD VIEW OF TABLES USED

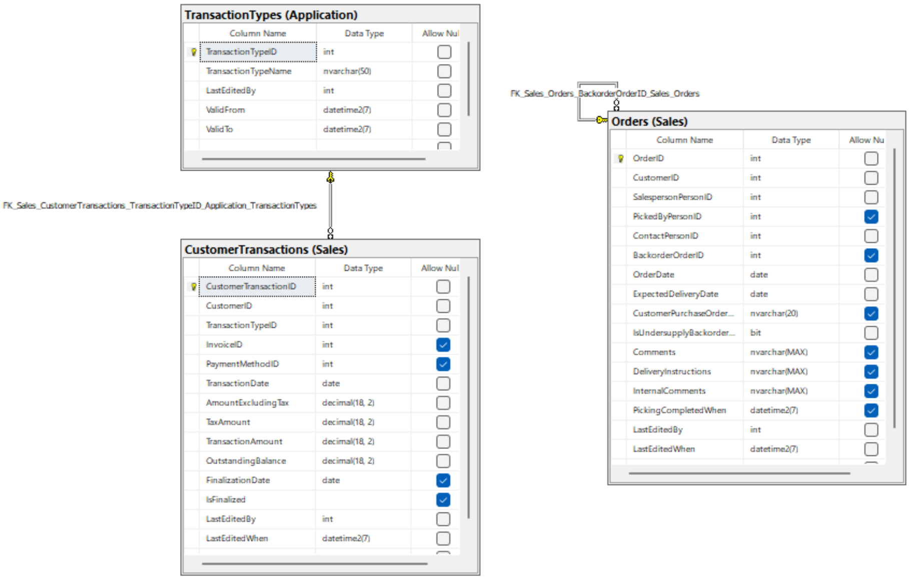


FK_Sales_CustomerTransactions_TransactionTypeID_Application_TransactionTypes
FK Sales Orders BackorderOrderID Sales Orders

TransactionTypes (App
- TransactionTypeID
- TransactionTypeName
- LastEditedBy

CustomerTransactions
- CustomerTransactionID
- CustomerID
- TransactionTypeID
- InvoiceID
- PaymentMethodID
- LastEditedBy

Orders (Sales)
- OrderID
- CustomerID
- SalespersonPersonID
- PickedByPersonID
- ContactPersonID
- BackorderOrderID
- LastEditedBy



FK_Sales_CustomerTransactions_TransactionTypeID_Application_TransactionTypes
FK_Sales_Orders_BackorderOrderID_Sales_Orders

TransactionTypes (Application)

Column Name	Data Type	Allow Nul
TransactionTypeID	int	☐
TransactionTypeName	nvarchar(50)	☐
LastEditedBy	int	☐
ValidFrom	datetime2(7)	☐
ValidTo	datetime2(7)	☐

CustomerTransactions (Sales)

Column Name	Data Type	Allow Nul
CustomerTransactionID	int	☐
CustomerID	int	☐
TransactionTypeID	int	☐
InvoiceID	int	☑
PaymentMethodID	int	☑
TransactionDate	date	☐
AmountExcludingTax	decimal(18, 2)	☐
TaxAmount	decimal(18, 2)	☐
TransactionAmount	decimal(18, 2)	☐
OutstandingBalance	decimal(18, 2)	☐
FinalizationDate	date	☑
IsFinalized		☑
LastEditedBy	int	☐
LastEditedWhen	datetime2(7)	☐

Orders (Sales)

Column Name	Data Type	Allow Nu
OrderID	int	☐
CustomerID	int	☐
SalespersonPersonID	int	☐
PickedByPersonID	int	☑
ContactPersonID	int	☐
BackorderOrderID	int	☑
OrderDate	date	☐
ExpectedDeliveryDate	date	☐
CustomerPurchaseOrder...	nvarchar(20)	☑
IsUndersupplyBackorder...	bit	☐
Comments	nvarchar(MAX)	☑
DeliveryInstructions	nvarchar(MAX)	☑
InternalComments	nvarchar(MAX)	☑
PickingCompletedWhen	datetime2(7)	☑
LastEditedBy	int	☐
LastEditedWhen	datetime2(7)	☐

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
[Sales].[CustomerTransactions]	CustomerTransactionId
[Sales].[Orders]	OrderDate
[Application].[TransactionTypes]	TransactionTypeName

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Not applicable

QUERY OF WORST

```
DROP FUNCTION IF EXISTS getDte;
GO
CREATE FUNCTION getDte
(
    @date AS INT
)
RETURNS TABLE
AS
RETURN SELECT OrderID,
              CustomerID,
              SalespersonPersonID,
              PickedByPersonID,
              ContactPersonID,
              BackorderOrderID,
              OrderDate,
              ExpectedDeliveryDate,
              CustomerPurchaseOrderNumber,
              IsUndersupplyBackordered,
              Comments,
              DeliveryInstructions,
              InternalComments,
              PickingCompletedWhen,
              LastEditedBy,
              LastEditedWhen
       FROM [Sales].[Orders]
       WHERE EOMONTH(OrderDate) = OrderDate
             AND YEAR(OrderDate) = @date;
GO
WITH TransactionName
AS (SELECT TT.TransactionTypeName AS Condition,
           TT.TransactionTypeID AS ID
    FROM [Application].[TransactionTypes] AS TT
    WHERE TT.TransactionTypeID = 3)
SELECT T.CustomerTransactionID,
       O.OrderDate,
       TT.Condition
FROM [Sales].[CustomerTransactions] AS T
    JOIN getDte(2015) AS O
        ON T.CustomerID = O.CustomerID
    JOIN TransactionName AS TT
        ON T.TransactionTypeID = TT.ID
GROUP BY T.CustomerTransactionID,
         O.OrderDate,
         TT.Condition
ORDER BY O.OrderDate;
```

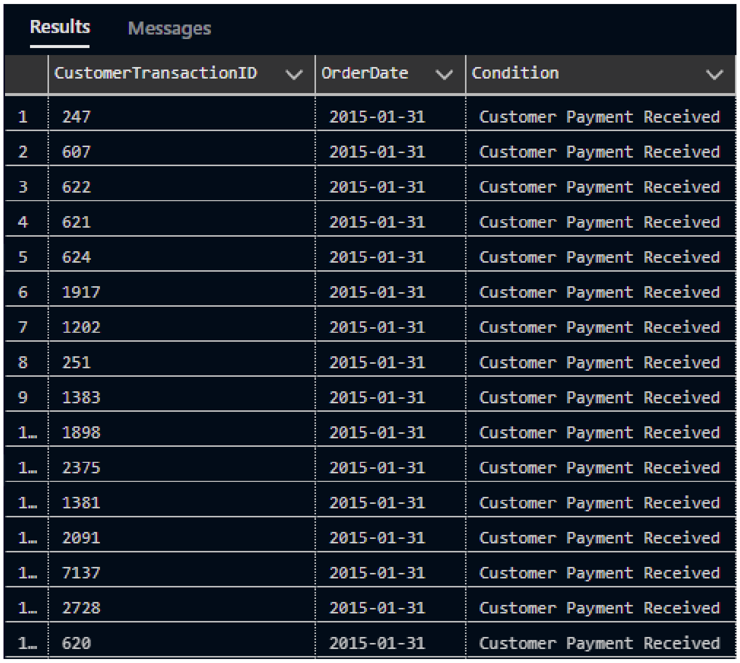
QUERY OF WORST CORRECTED

```
USE WideWorldImporters;
DROP FUNCTION IF EXISTS getDte;
GO
CREATE FUNCTION getDte
(
    @date AS INT
)
RETURNS TABLE
AS
RETURN SELECT CustomerID,
              OrderDate
       FROM [Sales].[Orders]
       WHERE EOMONTH(OrderDate) = OrderDate
             AND YEAR(OrderDate) = @date;
GO
WITH TransactionName
AS (SELECT TT.TransactionTypeName AS Condition,
           TT.TransactionTypeID AS ID
    FROM [Application].[TransactionTypes] AS TT
    WHERE TT.TransactionTypeID = 3)
SELECT T.CustomerTransactionID,
       O.OrderDate,
       TT.Condition
FROM [Sales].[CustomerTransactions] AS T
    JOIN getDte(2015) AS O
        ON T.CustomerID = O.CustomerID
    JOIN TransactionName AS TT
        ON T.TransactionTypeID = TT.ID
GROUP BY T.CustomerTransactionID,
         O.OrderDate,
         TT.Condition
ORDER BY O.OrderDate;
```

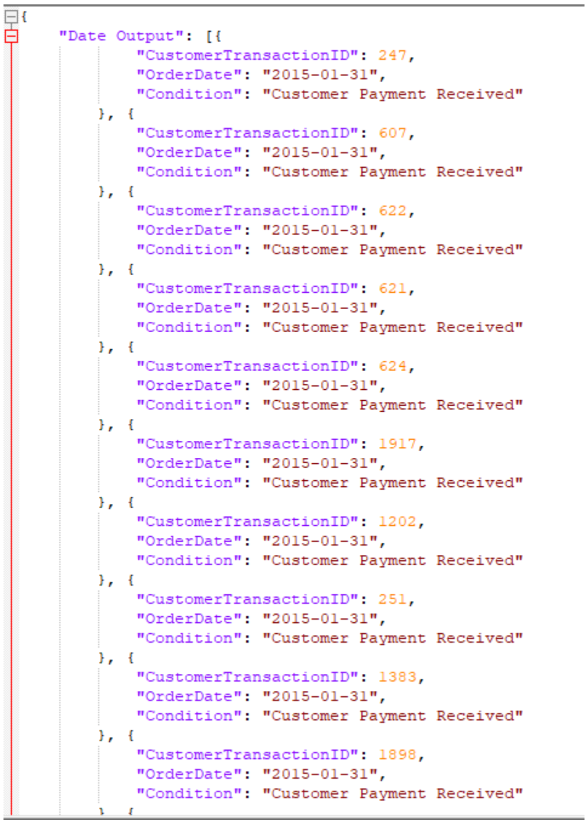
HOW IT WAS CORRECTED:

Shrank number of rows selected in the Custom Function

RELATIONAL AND JSON OUTPUT (23578 ROWS AFFECTED)



	CustomerTransactionID	OrderDate	Condition
1	247	2015-01-31	Customer Payment Received
2	607	2015-01-31	Customer Payment Received
3	622	2015-01-31	Customer Payment Received
4	621	2015-01-31	Customer Payment Received
5	624	2015-01-31	Customer Payment Received
6	1917	2015-01-31	Customer Payment Received
7	1202	2015-01-31	Customer Payment Received
8	251	2015-01-31	Customer Payment Received
9	1383	2015-01-31	Customer Payment Received
1…	1898	2015-01-31	Customer Payment Received
1…	2375	2015-01-31	Customer Payment Received
1…	1381	2015-01-31	Customer Payment Received
1…	2091	2015-01-31	Customer Payment Received
1…	7137	2015-01-31	Customer Payment Received
1…	2728	2015-01-31	Customer Payment Received
1…	620	2015-01-31	Customer Payment Received



```
{
    "Date Output": [{
            "CustomerTransactionID": 247,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 607,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 622,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 621,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 624,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 1917,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 1202,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 251,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 1383,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
            "CustomerTransactionID": 1898,
            "OrderDate": "2015-01-31",
            "Condition": "Customer Payment Received"
        }, {
```