

CSCI 331 PROJECT 1 – INDIVIDUAL PDF

PROFESSOR: PETER HELLER

SECTION: 9:15-10:30 AM

GROUP: G9-5

PDF OF: TIMOTHY DAKIS

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

## CONTENTS

Top #1: A Complex Query .....	4
Problem Statement.....	4
Reason it is a top.....	4
Key and Standard View of Tables Used .....	4
Table showing columns projected in the end .....	5
Table showing how projection sorted (if applicable) .....	5
Query .....	6
Relational and Json Output (27 Rows Affected) .....	8
Top #2: A Complex Query .....	9
Problem Statement.....	9
Reason it is a top.....	9
Key and Standard View of Tables Used .....	9
Table showing columns projected in the end .....	10
Table showing how projection sorted (if applicable) .....	10
Query .....	11
Relational and Json Output (673 Rows Affected) .....	12
Top #3: A Complex Query .....	13
Problem Statement.....	13
Reason it is a top.....	13
Key and Standard View of Tables Used .....	13
Table showing columns projected in the end .....	14
Table showing how projection sorted (if applicable) .....	14
Query .....	15
Relational and Json Output (22 Rows Affected) .....	16
Worst #1: A Simple Query .....	17
Problem Statement.....	17
Reason it is a Worst .....	17
Key and Standard View of Tables Used .....	17
Table showing columns projected in the end .....	18
Table showing how projection sorted (if applicable) .....	18
Query Of Worst.....	18
Query of Worst Corrected .....	18

How it was Corrected:.....	18
Relational and Json Output (8 Rows Affected) .....	19
Worst #2: A Medium Query.....	20
Problem Statement.....	20
Reason it is a Worst .....	20
Key and Standard View of Tables Used .....	20
Table showing columns projected in the end .....	21
Table showing how projection sorted (if applicable) .....	21
Query Of Worst.....	21
Query of Worst Corrected .....	22
How it was Corrected:.....	22
Relational and Json Output (29 Rows Affected) .....	22
Worst #3: A Medium Query.....	23
Problem Statement.....	23
Reason it is a Worst .....	23
Key and Standard View of Tables Used .....	23
Table showing columns projected in the end .....	24
Table showing how projection sorted (if applicable) .....	24
Query Of Worst.....	25
Query of Worst Corrected .....	26
How it was Corrected:.....	26
Relational and Json Output (3 Rows Affected) .....	27

## TOP #1: A COMPLEX QUERY

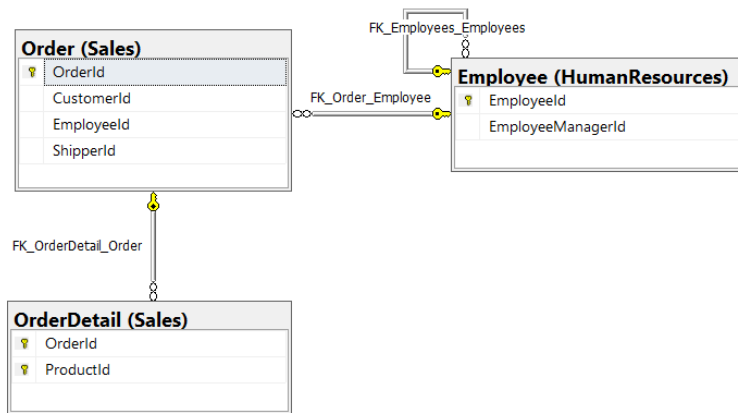
### PROBLEM STATEMENT

Find which quarter of each year every employee handled most of their orders, how much those orders were worth after discount, and list employees by name using Northwinds2022TSQLV7

### REASON IT IS A TOP

All things considered it is readable for what it is doing, has a decent logical flow, and it makes use of a lot of concepts learned by this point in this course. So, I feel like it is a good representation of the fruits of that learning

### KEY AND STANDARD VIEW OF TABLES USED



Order (Sales)			
Column Name	Data Type	Allow Nulls	
OrderId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
CustomerId	Udt.SurrogateKeyIntint	<input checked="" type="checkbox"/>	
EmployeeId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
ShipperId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
OrderDate	Udt.DateYYYYMMDD:d...	<input type="checkbox"/>	
RequiredDate	Udt.DateYYYYMMDD:d...	<input type="checkbox"/>	
ShipToDate	Udt.DateYYYYMMDD:d...	<input checked="" type="checkbox"/>	
Freight	Udt.Currency:money	<input type="checkbox"/>	
ShipToName	Udt.ContactName:nvarc...	<input type="checkbox"/>	
ShipToAddress	Udt.Address:nvarchar(6...	<input type="checkbox"/>	
ShipToCity	Udt.City:nvarchar(15)	<input type="checkbox"/>	
ShipToRegion	Udt.Region:nvarchar(15)	<input checked="" type="checkbox"/>	
ShipToPostalCode	Udt.PostalCode:nvarch...	<input checked="" type="checkbox"/>	
ShipToCountry	Udt.Country:nvarchar(15)	<input type="checkbox"/>	
UserAuthenticationId	int	<input checked="" type="checkbox"/>	
DateAdded	datetime2(7)	<input checked="" type="checkbox"/>	
DateOfLastUpdate	datetime2(7)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Employee (HumanResources)			
Column Name	Data Type	Allow Nulls	
EmployeeId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
EmployeeLastName	Udt.LastName:nvarchar...	<input type="checkbox"/>	
EmployeeFirstName	Udt.FirstName:nvarchar...	<input type="checkbox"/>	
EmployeeTitle	Udt.Title:nvarchar(30)	<input type="checkbox"/>	
EmployeeTitleOfCourtesy	Udt.TitleOfCourtesy:nva...	<input type="checkbox"/>	
BirthDate	Udt.DateYYYYMMDD:d...	<input type="checkbox"/>	
HireDate	Udt.DateYYYYMMDD:d...	<input type="checkbox"/>	
EmployeeAddress	Udt.Address:nvarchar(6...	<input type="checkbox"/>	
EmployeeCity	Udt.City:nvarchar(15)	<input checked="" type="checkbox"/>	
EmployeeRegion	Udt.Region:nvarchar(15)	<input checked="" type="checkbox"/>	
EmployeePostalCode	Udt.PostalCode:nvarch...	<input checked="" type="checkbox"/>	
EmployeeCountry	Udt.Country:nvarchar(15)	<input type="checkbox"/>	
EmployeePhoneNumber	Udt.TelephoneNumber:...	<input type="checkbox"/>	
EmployeeManagerId	Udt.SurrogateKeyIntint	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

OrderDetail (Sales)			
Column Name	Data Type	Allow Nulls	
OrderId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
ProductId	Udt.SurrogateKeyIntint	<input type="checkbox"/>	
UnitPrice	Udt.Currency:money	<input type="checkbox"/>	
Quantity	Udt.QuantitySmall:small...	<input type="checkbox"/>	
DiscountPercentage	Udt.Percentage:numeri...	<input type="checkbox"/>	
		<input type="checkbox"/>	

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Order	EmployeeId
Derived	EmployeeFullName OrderYear QuarterlyRevenue Quarter

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Order	EmployeeId	ASC

## QUERY

```

USE Northwinds2022TSQLV7;

DROP FUNCTION IF EXISTS Sales.udf_FindYearQuarter;
GO
CREATE FUNCTION Sales.udf_FindYearQuarter
(
    @date DATE
)
RETURNS NVARCHAR(20)
AS
BEGIN
    DECLARE @Result NVARCHAR(20);

    SELECT @Result = CASE
        WHEN MONTH(@date)
            BETWEEN 1 AND 3 THEN
            'Quarter I'
        WHEN MONTH(@date)
            BETWEEN 4 AND 6 THEN
            'Quarter II'
        WHEN MONTH(@date)
            BETWEEN 7 AND 9 THEN
            'Quarter III'
        WHEN MONTH(@date)
            BETWEEN 10 AND 12 THEN
            'Quarter IV'
        ELSE
            'ERROR - CANNOT CALCULATE QUARTER'
    END;

    RETURN @Result;
END;
GO
DROP FUNCTION IF EXISTS Sales.udf_RevenueAfterDiscount;
GO
CREATE FUNCTION Sales.udf_RevenueAfterDiscount
(
    @quantity AS INT,
    @unitprice AS MONEY,
    @discount AS NUMERIC(4, 3)
)
RETURNS NUMERIC(20, 2)
AS
BEGIN
    DECLARE @Result NUMERIC(20, 2) = @quantity * @unitprice * (1.0 - @discount);
    RETURN @Result;
END;
GO

WITH OrderByYearAndQuarter
AS (SELECT O.OrderId,
    O.EmployeeId,
    YEAR(O.OrderDate) AS OrderYear,
    Sales.udf_FindYearQuarter(O.OrderDate) AS [Quarter]
    FROM Sales.[Order] AS O),
    EmployeeRevenuePerQuarter
AS (SELECT O.EmployeeId,

```

```

        O.OrderYear,
        O.[Quarter],
        CONCAT(E.EmployeeFirstName + ' ', E.EmployeeLastName) AS EmployeeFullName,
        SUM(Sales.udf_RevenueAfterDiscount(OD.Quantity, OD.UnitPrice,
OD.DiscountPercentage)) AS QuarterlyRevenue
FROM HumanResources.Employee AS E
    INNER JOIN OrderByYearAndQuarter AS O
        ON O.EmployeeId = E.EmployeeId
    INNER JOIN Sales.OrderDetail AS OD
        ON O.OrderId = OD.OrderId
GROUP BY O.EmployeeId,
        O.OrderYear,
        O.[Quarter],
        CONCAT(E.EmployeeFirstName + ' ', E.EmployeeLastName))
SELECT Q1.EmployeeId,
        Q1.EmployeeFullName,
        Q1.OrderYear,
        Q1.[Quarter],
        FORMAT(Q1.QuarterlyRevenue, 'C') AS QuarterlyRevenue
FROM EmployeeRevenuePerQuarter AS Q1
WHERE Q1.[Quarter] =
(
    SELECT TOP (1)
        Q2.[Quarter]
    FROM EmployeeRevenuePerQuarter AS Q2
    WHERE Q2.EmployeeId = Q1.EmployeeId
        AND Q2.OrderYear = Q1.OrderYear
    ORDER BY Q2.QuarterlyRevenue DESC
)
ORDER BY EmployeeId;

```

## RELATIONAL AND JSON OUTPUT (27 ROWS AFFECTED)

Results Messages					
	EmployeeId	EmployeeFullName	OrderYear	Quarter	QuarterlyRevenue
1	1	Sara Davis	2014	Quarter IV	\$21,942.74
2	1	Sara Davis	2015	Quarter III	\$32,077.22
3	1	Sara Davis	2016	Quarter I	\$44,090.32
4	2	Don Funk	2014	Quarter IV	\$15,816.26
5	2	Don Funk	2015	Quarter II	\$24,374.17
6	2	Don Funk	2016	Quarter I	\$41,416.30
7	3	Judy Lew	2014	Quarter IV	\$10,046.66
8	3	Judy Lew	2015	Quarter IV	\$34,861.70
9	3	Judy Lew	2016	Quarter I	\$63,605.39
10	4	Yael Peled	2014	Quarter IV	\$31,129.59
11	4	Yael Peled	2015	Quarter I	\$41,088.55
12	4	Yael Peled	2016	Quarter I	\$38,187.48
13	5	Sven Mortensen	2014	Quarter IV	\$15,325.10
14	5	Sven Mortensen	2015	Quarter III	\$12,085.81
15	5	Sven Mortensen	2016	Quarter I	\$19,481.90
16	6	Paul Suurs	2014	Quarter III	\$9,082.75
17	6	Paul Suurs	2015	Quarter IV	\$19,939.27

Query executed successfully. localhost:13001 (15.0 RTM) sa (66) Northwinds2022TSQV7 00:00:00 27 rows

JSToolNpp JSON Viewer	
Refresh	
ROOT	
Best Quarter Per Employee: [A	
[0]: [Object]	
[1]: [Object]	
[2]: [Object]	
[3]: [Object]	
[4]: [Object]	
[5]: [Object]	
[6]: [Object]	
[7]: [Object]	
[8]: [Object]	
[9]: [Object]	
[10]: [Object]	
[11]: [Object]	
[12]: [Object]	
[13]: [Object]	
[14]: [Object]	
[15]: [Object]	
[16]: [Object]	
[17]: [Object]	
[18]: [Object]	
[19]: [Object]	
[20]: [Object]	
[21]: [Object]	
[22]: [Object]	
[23]: [Object]	
[24]: [Object]	
[25]: [Object]	
[26]: [Object]	

```

1
2   "Best Quarter Per Employee": [{
3       "EmployeeId": 1,
4       "EmployeeFullName": "Sara Davis",
5       "OrderYear": 2014,
6       "Quarter": "Quarter IV",
7       "QuarterlyRevenue": "$21,942.74"
8   }, {
9       "EmployeeId": 1,
10      "EmployeeFullName": "Sara Davis",
11      "OrderYear": 2015,
12      "Quarter": "Quarter III",
13      "QuarterlyRevenue": "$32,077.22"
14   }, {
15      "EmployeeId": 1,
16      "EmployeeFullName": "Sara Davis",
17      "OrderYear": 2016,
18      "Quarter": "Quarter I",
19      "QuarterlyRevenue": "$44,090.32"
20   }, {
21      "EmployeeId": 2,
22      "EmployeeFullName": "Don Funk",
23      "OrderYear": 2014,
24      "Quarter": "Quarter IV",
25      "QuarterlyRevenue": "$15,816.26"
26   }, {
27      "EmployeeId": 2,
28      "EmployeeFullName": "Don Funk",
29      "OrderYear": 2015,
30      "Quarter": "Quarter II",
31      "QuarterlyRevenue": "$24,374.17"

```



## TOP #2: A COMPLEX QUERY

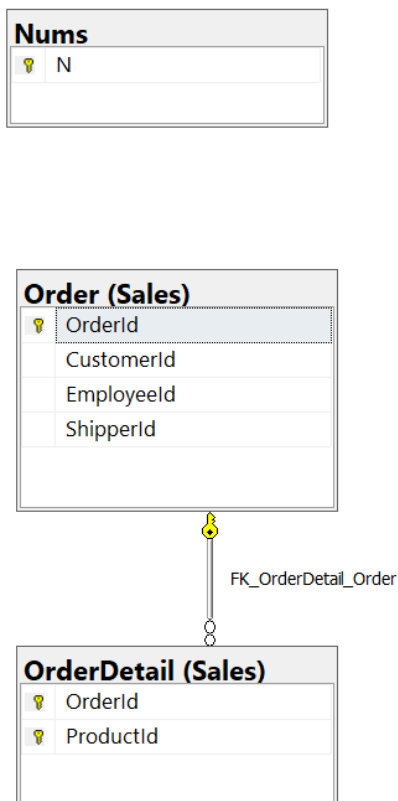
### PROBLEM STATEMENT

List all dates between the earliest and most recent order, and if an order has been placed or not, and how much all orders cost on each date after discount using Northwinds2022TSQLV4

### REASON IT IS A TOP

This is a top because I think it's pretty interesting and has a pretty nice logical flow and read to it. It also makes use of an auxiliary numbers table to assist in querying which is pretty interesting to utilize

### KEY AND STANDARD VIEW OF TABLES USED



	Column Name	Data Type	Allow Nulls
	N	Udt.SurrogateKey...	<input type="checkbox"/>
			<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
	Orderid	Udt.SurrogateKey...	<input type="checkbox"/>
	Customerid	Udt.SurrogateKey...	<input checked="" type="checkbox"/>
	Employeeid	Udt.SurrogateKey...	<input type="checkbox"/>
	Shipperid	Udt.SurrogateKey...	<input type="checkbox"/>
	OrderDate	Udt.DateYYYYMM...	<input type="checkbox"/>
	RequiredDate	Udt.DateYYYYMM...	<input type="checkbox"/>
	ShipToDate	Udt.DateYYYYMM...	<input checked="" type="checkbox"/>
	Freight	Udt.Currency:mon...	<input type="checkbox"/>
	ShipToName	Udt.ContactName...	<input type="checkbox"/>
	ShipToAddress	Udt.Address:nvar...	<input type="checkbox"/>
	ShipToCity	Udt.City:nvarchar(...	<input type="checkbox"/>
	ShipToRegion	Udt.Region:nvarc...	<input checked="" type="checkbox"/>
	ShipToPostalCode	Udt.PostalCode:n...	<input checked="" type="checkbox"/>
	ShipToCountry	Udt.Country:nvarc...	<input type="checkbox"/>
	UserAuthenticationid	int	<input checked="" type="checkbox"/>
	DateAdded	datetime2(7)	<input checked="" type="checkbox"/>
	DateOfLastUpdate	datetime2(7)	<input checked="" type="checkbox"/>

FK\_OrderDetail\_Order

	Column Name	Data Type	Allow Nulls
	Orderid	Udt.SurrogateKey...	<input type="checkbox"/>
	Productid	Udt.SurrogateKey...	<input type="checkbox"/>
	UnitPrice	Udt.Currency:mon...	<input type="checkbox"/>
	Quantity	Udt.Quantity:Small...	<input type="checkbox"/>
	DiscountPercentage	Udt.Percentage:n...	<input type="checkbox"/>

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Derived	Date  ThereHasBeenAnOrder  RevenueAfterDiscount

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Derived	Date	ASC

## QUERY

```

USE Northwinds2022TSQLV7;

DROP FUNCTION IF EXISTS Sales.udf_DateFromAnchor;
GO
CREATE FUNCTION Sales.udf_DateFromAnchor
(
    @AnchorDate DATE,
    @DaysSinceAnchor INT
)
RETURNS DATE
AS
BEGIN
    DECLARE @Result DATE;

    SELECT @Result = CAST(DATEADD(DAY, @DaysSinceAnchor, @AnchorDate) AS DATE);
    RETURN @Result;
END;
GO
DROP FUNCTION IF EXISTS Sales.udf_RevenueAfterDiscount;
GO
CREATE FUNCTION Sales.udf_RevenueAfterDiscount
(
    @quantity AS INT,
    @unitprice AS MONEY,
    @discount AS NUMERIC(4, 3)
)
RETURNS NUMERIC(20, 2)
AS
BEGIN
    DECLARE @Result NUMERIC(20, 2) = @quantity * @unitprice * (1.0 - @discount);
    RETURN @Result;
END;
GO

WITH AllDatesFromAnchor
AS (SELECT Sales.udf_DateFromAnchor(
    (
        SELECT MIN(O.OrderDate) FROM Sales.[Order] AS O
    ),
    N - 1
    ) AS [Date]
FROM dbo.Nums
WHERE Sales.udf_DateFromAnchor(
    (
        SELECT MIN(O.OrderDate) FROM Sales.[Order] AS O
    ),
    N - 1
    ) <=
    (
        SELECT MAX(O.OrderDate) FROM Sales.[Order] AS O
    ))
SELECT D.[Date],
CASE
    WHEN MAX(O.OrderId) IS NULL THEN
        'No'
    ELSE

```

```

        'Yes'
    END AS ThereHasBeenAnOrder,
    CONCAT('$', COALESCE(SUM(Sales.udf_RevenueAfterDiscount(OD.Quantity, OD.UnitPrice,
    OD.DiscountPercentage)), 0)) AS RevenueAfterDiscount
FROM AllDatesFromAnchor AS D
    LEFT OUTER JOIN(Sales.[Order] AS O
    INNER JOIN Sales.OrderDetail AS OD
        ON OD.OrderId = O.OrderId)
        ON D.[Date] = O.OrderDate
GROUP BY D.[Date]
ORDER BY D.[Date];

```

## RELATIONAL AND JSON OUTPUT (673 ROWS AFFECTED)

	Date	ThereHasBeenAnOrder	RevenueAfterDiscount
1	2014-07-04	Yes	\$440.00
2	2014-07-05	Yes	\$1863.40
3	2014-07-06	No	\$0.00
4	2014-07-07	No	\$0.00
5	2014-07-08	Yes	\$2206.66
6	2014-07-09	Yes	\$3597.90
7	2014-07-10	Yes	\$1444.80
8	2014-07-11	Yes	\$556.62
9	2014-07-12	Yes	\$2490.50
10	2014-07-13	No	\$0.00
11	2014-07-14	No	\$0.00
12	2014-07-15	Yes	\$517.80
13	2014-07-16	Yes	\$1119.90
14	2014-07-17	Yes	\$1614.88
15	2014-07-18	Yes	\$100.80
16	2014-07-19	Yes	\$1952.65
17	2014-07-20	No	\$0.00

Query executed successfully. | localhost:13001 (15.0 RTM) | sa (66) | Northwinds2022TSQLV7 | 00:00:00 | 673 rows

JSToolNpp JSON Viewer	
Refresh	new 2
<div> <div>[641]: [Object]</div> <div>[642]: [Object]</div> <div>[643]: [Object]</div> <div>[644]: [Object]</div> <div>[645]: [Object]</div> <div>[646]: [Object]</div> <div>[647]: [Object]</div> <div>[648]: [Object]</div> <div>[649]: [Object]</div> <div>[650]: [Object]</div> <div>[651]: [Object]</div> <div>[652]: [Object]</div> <div>[653]: [Object]</div> <div>[654]: [Object]</div> <div>[655]: [Object]</div> <div>[656]: [Object]</div> <div>[657]: [Object]</div> <div>[658]: [Object]</div> <div>[659]: [Object]</div> <div>[660]: [Object]</div> <div>[661]: [Object]</div> <div>[662]: [Object]</div> <div>[663]: [Object]</div> <div>[664]: [Object]</div> <div>[665]: [Object]</div> <div>[666]: [Object]</div> <div>[667]: [Object]</div> <div>[668]: [Object]</div> <div>[669]: [Object]</div> <div>[670]: [Object]</div> <div>[671]: [Object]</div> <div>[672]: [Object]</div> </div>	<pre> 1 2  "Dates": [{ 3      "Date": "2014-07-04", 4      "ThereHasBeenAnOrder": "Yes", 5      "RevenueAfterDiscount": "\$440.00" 6  }, { 7      "Date": "2014-07-05", 8      "ThereHasBeenAnOrder": "Yes", 9      "RevenueAfterDiscount": "\$1863.40" 10 }, { 11     "Date": "2014-07-06", 12     "ThereHasBeenAnOrder": "No", 13     "RevenueAfterDiscount": "\$0.00" 14 }, { 15     "Date": "2014-07-07", 16     "ThereHasBeenAnOrder": "No", 17     "RevenueAfterDiscount": "\$0.00" 18 }, { 19     "Date": "2014-07-08", 20     "ThereHasBeenAnOrder": "Yes", 21     "RevenueAfterDiscount": "\$2206.66" 22 }, { 23     "Date": "2014-07-09", 24     "ThereHasBeenAnOrder": "Yes", 25     "RevenueAfterDiscount": "\$3597.90" 26 }, { 27     "Date": "2014-07-10", 28     "ThereHasBeenAnOrder": "Yes", 29     "RevenueAfterDiscount": "\$1444.80" 30 }, { 31     "Date": "2014-07-11", </pre>

## TOP #3: A COMPLEX QUERY

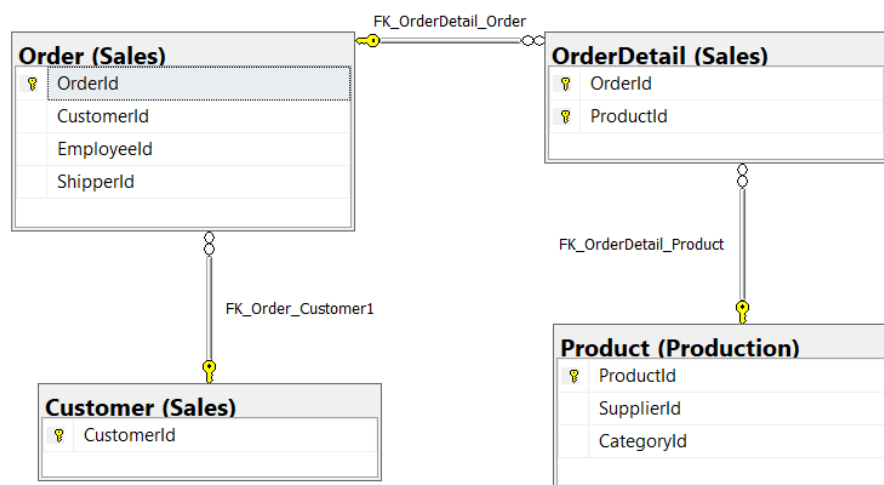
### PROBLEM STATEMENT

What type of product is ordered most often per customer country using Northwinds2022TSQLV7

### REASON IT IS A TOP

While a bit messy I think this query is pretty interesting and has a somewhat decent logical flow to it, utilizing CTE and functions and a subquery to break down the top per country

### KEY AND STANDARD VIEW OF TABLES USED



Column Name	Data Type	Allow Nulls
OrderId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
CustomerId	Udt.SurrogateKey.Intint	<input checked="" type="checkbox"/>
EmployeeId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
ShipperId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
OrderDate	Udt.Date.YYYYMMDD.date	<input type="checkbox"/>
RequiredDate	Udt.Date.YYYYMMDD.date	<input type="checkbox"/>
ShipToDate	Udt.Date.YYYYMMDD.date	<input checked="" type="checkbox"/>
Freight	Udt.Currency.money	<input type="checkbox"/>
ShipToName	Udt.ContactName.nvarchar(100)	<input type="checkbox"/>
ShipToAddress	Udt.Address.nvarchar(80)	<input type="checkbox"/>
ShipToCity	Udt.City.nvarchar(15)	<input type="checkbox"/>
ShipToRegion	Udt.Region.nvarchar(15)	<input checked="" type="checkbox"/>
ShipToPostalCode	Udt.PostalCode.nvarchar(10)	<input checked="" type="checkbox"/>
ShipToCountry	Udt.Country.nvarchar(15)	<input checked="" type="checkbox"/>
UserAuthenticationId	int	<input checked="" type="checkbox"/>
DateAdded	datetime2(7)	<input checked="" type="checkbox"/>
DateOfLastUpdate	datetime2(7)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
OrderId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
ProductId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
UnitPrice	Udt.Currency.money	<input type="checkbox"/>
Quantity	Udt.Quantity.Smallint	<input type="checkbox"/>
DiscountPercentage	Udt.Percentage.numeric(4, 3)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
CustomerId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
CustomerCompanyName	Udt.CompanyName.nvarchar(100)	<input type="checkbox"/>
CustomerContactName	Udt.ContactName.nvarchar(100)	<input type="checkbox"/>
CustomerContactTitle	Udt.Title.nvarchar(30)	<input type="checkbox"/>
CustomerAddress	Udt.Address.nvarchar(80)	<input type="checkbox"/>
CustomerCity	Udt.City.nvarchar(15)	<input type="checkbox"/>
CustomerRegion	Udt.Region.nvarchar(15)	<input checked="" type="checkbox"/>
CustomerPostalCode	Udt.PostalCode.nvarchar(10)	<input checked="" type="checkbox"/>
CustomerCountry	Udt.Country.nvarchar(15)	<input type="checkbox"/>
CustomerPhoneNumber	Udt.TelephoneNumber.nvarchar(20)	<input type="checkbox"/>
CustomerFaxNumber	Udt.TelephoneNumber.nvarchar(20)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
ProductId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
ProductName	Udt.ProductName.nvarchar(100)	<input type="checkbox"/>
SupplierId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
CategoryId	Udt.SurrogateKey.Intint	<input type="checkbox"/>
UnitPrice	Udt.Currency.money	<input type="checkbox"/>
Discontinued	Udt.Flag.Bitbit	<input type="checkbox"/>

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Customer	CustomerCountry
Derived	ProductType TimesOrdered

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Derived	COUNT(ProductId)	DESC

## QUERY

```
USE Northwinds2022TSQLV7;

DROP FUNCTION IF EXISTS Sales.udf_GetProductCategory;
GO
CREATE FUNCTION Sales.udf_GetProductCategory
(
    @categoryid INT
)
RETURNS NVARCHAR(20)
AS
BEGIN
    DECLARE @Result NVARCHAR(20);

    SELECT @Result =
    (
        SELECT TOP (1)
            CategoryName
        FROM Production.Category AS P
        WHERE @categoryid = P.CategoryId
        ORDER BY P.CategoryId
    );
    RETURN @Result;
END;
GO

WITH ProductAndCategory
AS (SELECT P.ProductId,
        Sales.udf_GetProductCategory(P.CategoryId) AS ProductType
    FROM Production.Product AS P)
SELECT C.CustomerCountry,
        P.ProductType,
        COUNT(OD.ProductId) TimesOrdered
FROM Sales.Customer AS C
    INNER JOIN Sales.[Order] AS O
        ON O.CustomerId = C.CustomerId
    INNER JOIN Sales.OrderDetail AS OD
        ON OD.OrderId = O.OrderId
    INNER JOIN ProductAndCategory AS P
        ON P.ProductId = OD.ProductId
GROUP BY C.CustomerCountry,
        P.ProductType
HAVING P.ProductType IN
(
    SELECT TOP (1) WITH TIES
        Sales.udf_GetProductCategory(P2.CategoryId) AS ProductType
    FROM Sales.[Order] AS O2
        INNER JOIN Sales.Customer AS C2
            ON C2.CustomerId = O2.CustomerId
        INNER JOIN Sales.OrderDetail AS OD2
            ON OD2.OrderId = O2.OrderId
        INNER JOIN Production.Product AS P2
            ON P2.ProductId = OD2.ProductId
    WHERE C2.CustomerCountry = C.CustomerCountry
    GROUP BY Sales.udf_GetProductCategory(P2.CategoryId)
    ORDER BY COUNT(OD2.ProductId) DESC
)
```

ORDER BY COUNT(OD.ProductId) DESC;

## RELATIONAL AND JSON OUTPUT (22 ROWS AFFECTED)

Results		Messages	
	CustomerCountry	ProductType	TimesOrdered
1	Germany	Dairy Products	65
2	USA	Confections	62
3	Brazil	Beverages	40
4	France	Beverages	35
5	UK	Dairy Products	29
6	Austria	Dairy Products	27
7	Sweden	Beverages	25
8	Venezuela	Seafood	23
9	Venezuela	Dairy Products	23
10	Mexico	Beverages	18
11	Canada	Confections	15
12	Ireland	Dairy Products	13
13	Switzerland	Seafood	11
14	Finland	Dairy Products	11
15	Belgium	Beverages	11
16	Italy	Confections	10
17	Spain	Confections	10

Query executed successfully. | localhost,13001 (15.0 RTM) | sa (66) | Northwinds2022TSQLV7 | 00:00:00 | 22 rows

JSToolNpp JSON Viewer		new 1	
Refresh			
ROOT			
Product Per Country: [Array]			
[0]: [Object]		1	
[1]: [Object]		2	
[2]: [Object]		3	
[3]: [Object]		4	
[4]: [Object]		5	
[5]: [Object]		6	
[6]: [Object]		7	
[7]: [Object]		8	
[8]: [Object]		9	
[9]: [Object]		10	
[10]: [Object]		11	
[11]: [Object]		12	
[12]: [Object]		13	
[13]: [Object]		14	
[14]: [Object]		15	
[15]: [Object]		16	
[16]: [Object]		17	
[17]: [Object]		18	
[18]: [Object]		19	
[19]: [Object]		20	
[20]: [Object]		21	
[21]: [Object]		22	
		23	
		24	
		25	
		26	
		27	
		28	
		29	
		30	
		31	

```
"Product Per Country": [{
  "CustomerCountry": "Germany",
  "ProductType": "Dairy Products",
  "TimesOrdered": 65
}, {
  "CustomerCountry": "USA",
  "ProductType": "Confections",
  "TimesOrdered": 62
}, {
  "CustomerCountry": "Brazil",
  "ProductType": "Beverages",
  "TimesOrdered": 40
}, {
  "CustomerCountry": "France",
  "ProductType": "Beverages",
  "TimesOrdered": 35
}, {
  "CustomerCountry": "UK",
  "ProductType": "Dairy Products",
  "TimesOrdered": 29
}, {
  "CustomerCountry": "Austria",
  "ProductType": "Dairy Products",
  "TimesOrdered": 27
}, {
  "CustomerCountry": "Sweden",
  "ProductType": "Beverages",
  "TimesOrdered": 25
}, {
  "CustomerCountry": "Venezuela",
```



## WORST #1: A SIMPLE QUERY

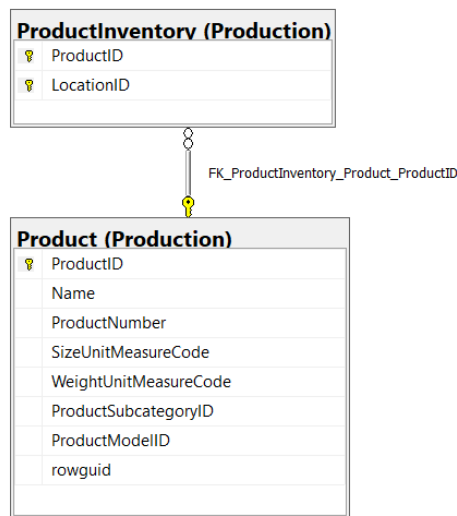
### PROBLEM STATEMENT

What products need to be reordered (replenished) using AdventureWorks2017?

### REASON IT IS A WORST

While required, it just does not make sense to group by ReorderPoint even though it does not negatively affect outcome as ReorderPoint for each ProductId is the same

### KEY AND STANDARD VIEW OF TABLES USED



ProductInventory (Production)			
Column Name	Data Type	Allow Nulls	
ProductID	int	<input type="checkbox"/>	
LocationID	smallint	<input type="checkbox"/>	
Shelf	nvarchar(10)	<input type="checkbox"/>	
Bin	tinyint	<input type="checkbox"/>	
Quantity	smallint	<input type="checkbox"/>	
rowguid	uniqueidentifier	<input type="checkbox"/>	
ModifiedDate	datetime	<input type="checkbox"/>	

Product (Production)			
Column Name	Data Type	Allow Nulls	
ProductID	int	<input type="checkbox"/>	
Name	nvarchar(50)	<input type="checkbox"/>	
ProductNumber	nvarchar(25)	<input type="checkbox"/>	
MakeFlag	Flag:bit	<input type="checkbox"/>	
FinishedGoodsFlag	Flag:bit	<input type="checkbox"/>	
Color	nvarchar(15)	<input checked="" type="checkbox"/>	
SafetyStockLevel	smallint	<input type="checkbox"/>	
ReorderPoint	smallint	<input type="checkbox"/>	
StandardCost	money	<input type="checkbox"/>	
ListPrice	money	<input type="checkbox"/>	
Size	nvarchar(5)	<input checked="" type="checkbox"/>	
SizeUnitMeasureCode	nchar(3)	<input checked="" type="checkbox"/>	
WeightUnitMeasureCode	nchar(3)	<input checked="" type="checkbox"/>	
Weight	decimal(8, 2)	<input checked="" type="checkbox"/>	
DaysToManufacture	int	<input type="checkbox"/>	
ProductLine	nchar(2)	<input checked="" type="checkbox"/>	
Class	nchar(2)	<input checked="" type="checkbox"/>	
Style	nchar(2)	<input checked="" type="checkbox"/>	
ProductSubcategoryID	int	<input checked="" type="checkbox"/>	
ProductModelID	int	<input checked="" type="checkbox"/>	
SellStartDate	datetime	<input type="checkbox"/>	
SellEndDate	datetime	<input checked="" type="checkbox"/>	
DiscontinuedDate	datetime	<input checked="" type="checkbox"/>	
rowguid	uniqueidentifier	<input type="checkbox"/>	
ModifiedDate	datetime	<input type="checkbox"/>	

TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Product	ProductId ReorderPoint
Derived	CurrentStock

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

NOT APPLICABLE

QUERY OF WORST

```
USE AdventureWorks2017;

SELECT P.ProductID,
       P.ReorderPoint,
       SUM([PI].Quantity) AS CurrentStock
FROM Production.Product AS P
     INNER JOIN Production.ProductInventory AS [PI]
       ON P.ProductID = [PI].ProductID
GROUP BY P.ProductID,
         P.ReorderPoint
HAVING SUM([PI].Quantity) <= P.ReorderPoint;
```

QUERY OF WORST CORRECTED

```
USE AdventureWorks2017;

WITH ProductInventoryStock
AS (SELECT P.ProductID,
          SUM(P.Quantity) AS CurrentStock
     FROM Production.ProductInventory AS P
     GROUP BY P.ProductID)
SELECT P.ProductID,
       P.ReorderPoint,
       S.CurrentStock
FROM Production.Product AS P
     INNER JOIN ProductInventoryStock AS S
       ON S.ProductID = P.ProductID
       AND S.CurrentStock <= P.ReorderPoint;
```

---

HOW IT WAS CORRECTED:

Using a CTE makes the logical flow better and makes it mildly more readable. Also does not need to group by things it does not make sense to group by

## RELATIONAL AND JSON OUTPUT (8 ROWS AFFECTED)

	ProductID	ReorderPoint	CurrentStock
1	386	750	725
2	462	750	701
3	853	3	0
4	859	3	0
5	876	3	0
6	882	3	0
7	910	375	355
8	956	75	75

Query executed successfully.

localhost:13001 (15.0 RTM) | sa (66) | AdventureWorks2017 | 00:00:00 | 8 rows

ROOT

Products to Restock: [Array]

- [0]: [Object]
- [1]: [Object]
- [2]: [Object]
- [3]: [Object]
- [4]: [Object]
- [5]: [Object]
- [6]: [Object]
- [7]: [Object]

```

"Products to Restock": [{
  "ProductID": 386,
  "ReorderPoint": 750,
  "CurrentStock": 725
}, {
  "ProductID": 462,
  "ReorderPoint": 750,
  "CurrentStock": 701
}, {
  "ProductID": 853,
  "ReorderPoint": 3,
  "CurrentStock": 0
}, {
  "ProductID": 859,
  "ReorderPoint": 3,
  "CurrentStock": 0
}, {
  "ProductID": 876,
  "ReorderPoint": 3,
  "CurrentStock": 0
}, {
  "ProductID": 882,
  "ReorderPoint": 3,
  "CurrentStock": 0
}, {
  "ProductID": 910,
  "ReorderPoint": 375,
  "CurrentStock": 355
}, {
  "ProductID": 956,
  "ReorderPoint": 75,
  "CurrentStock": 75
}]

```

## WORST #2: A MEDIUM QUERY

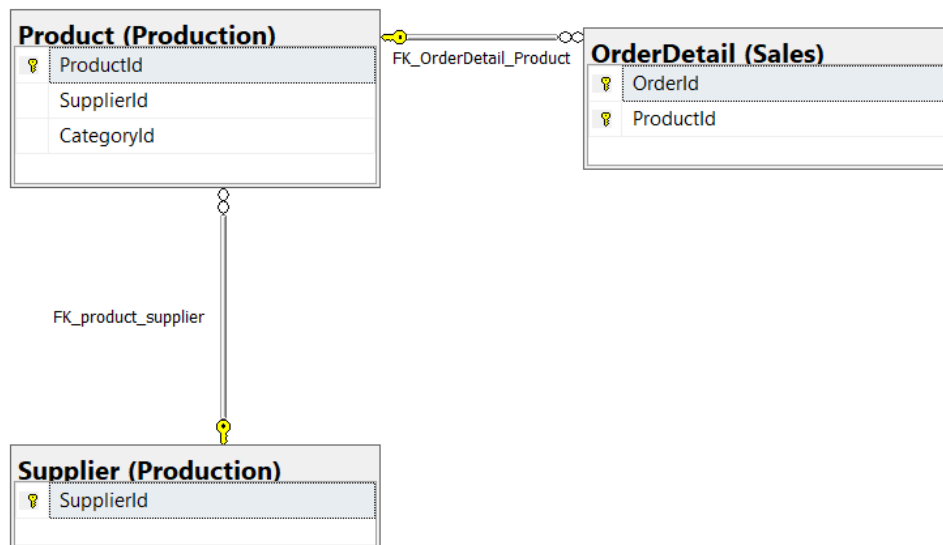
### PROBLEM STATEMENT

How much has every supplier sold using Northwinds2022TSQLV7?

### REASON IT IS A WORST

Use of CTE largely unnecessary, just reduces readability as opposed to improving it as is the goal with CTEs

### KEY AND STANDARD VIEW OF TABLES USED



Product (Production)			
Column Name	Data Type	Allow Nulls	
ProductId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
ProductName	Udt.ProductName: nvarchar(100)	<input type="checkbox"/>	
SupplierId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
CategoryId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
UnitPrice	Udt.Currency: money	<input type="checkbox"/>	
Discontinued	Udt.FlagBit: bit	<input type="checkbox"/>	

OrderDetail (Sales)			
Column Name	Data Type	Allow Nulls	
OrderId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
ProductId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
UnitPrice	Udt.Currency: money	<input type="checkbox"/>	
Quantity	Udt.QuantitySmall: smallint	<input type="checkbox"/>	
DiscountPercentage	Udt.Percentage: numeric(5,2)	<input type="checkbox"/>	

Supplier (Production)			
Column Name	Data Type	Allow Nulls	
SupplierId	Udt.SurrogateKeyInt: int	<input type="checkbox"/>	
SupplierCompanyName	Udt.CompanyName: nvarchar(100)	<input type="checkbox"/>	
SupplierContactName	Udt.ContactName: nvarchar(100)	<input type="checkbox"/>	
SupplierContactTitle	Udt.ContactTitle: nvarchar(100)	<input type="checkbox"/>	
SupplierAddress	Udt.Address: nvarchar(60)	<input type="checkbox"/>	
SupplierCity	Udt.City: nvarchar(15)	<input type="checkbox"/>	
SupplierRegion	Udt.Region: nvarchar(15)	<input checked="" type="checkbox"/>	
SupplierPostalCode	Udt.PostalCode: nvarchar(10)	<input checked="" type="checkbox"/>	
SupplierCountry	Udt.Country: nvarchar(15)	<input type="checkbox"/>	
SupplierPhoneNumber	Udt.TelephoneNumber: nvarchar(20)	<input type="checkbox"/>	
SupplierFaxNumber	Udt.TelephoneNumber: nvarchar(20)	<input checked="" type="checkbox"/>	

Relationships:

- FK\_OrderDetail\_Product**: Connects ProductId in OrderDetail to ProductId in Product.
- FK\_product\_supplier**: Connects SupplierId in Product to SupplierId in Supplier.

#### TABLE SHOWING COLUMNS PROJECTED IN THE END

Table Name	Column Name
Supplier	SupplierId
Derived	TotalProductSold

#### TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)

Table Name	Column Name	Sort Order
Supplier	SupplierId	ASC

#### QUERY OF WORST

```
USE Northwinds2022TSQLV7;
```

```
WITH SupplierAndProduct
AS (SELECT S.SupplierId,
          P.ProductId
     FROM Production.Supplier AS S
          INNER JOIN Production.Product AS P
              ON P.SupplierId = S.SupplierId)
SELECT SP.SupplierId,
       SUM(OD.Quantity) AS TotalProductsSold
FROM SupplierAndProduct AS SP
     INNER JOIN Sales.OrderDetail AS OD
         ON OD.ProductId = SP.ProductId
GROUP BY SP.SupplierId
ORDER BY SP.SupplierId;
```

## QUERY OF WORST CORRECTED

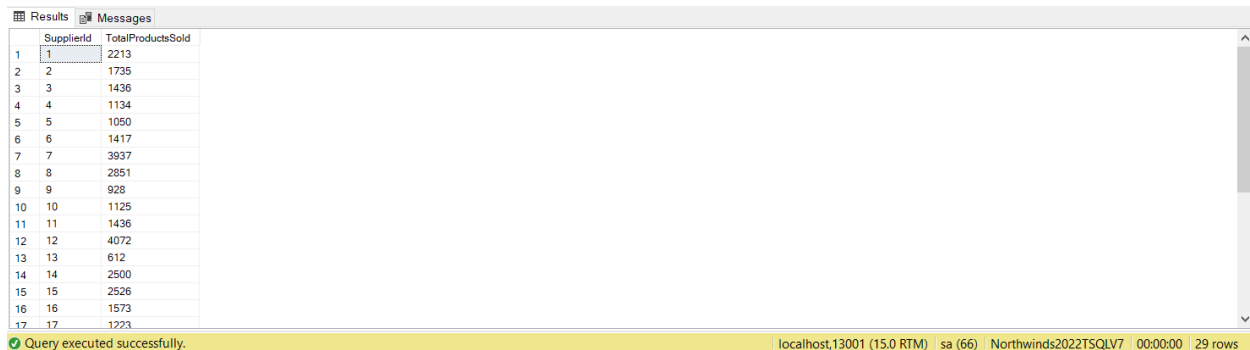
```
USE Northwinds2022TSQLV7;
```

```
SELECT P.SupplierId,  
       SUM(OD.Quantity) AS TotalProductSold  
FROM Production.Supplier AS S  
      INNER JOIN Production.Product AS P  
              ON S.SupplierId = P.SupplierId  
      INNER JOIN Sales.OrderDetail AS OD  
              ON OD.ProductId = P.ProductId  
GROUP BY P.SupplierId  
ORDER BY P.SupplierId;
```

## HOW IT WAS CORRECTED:

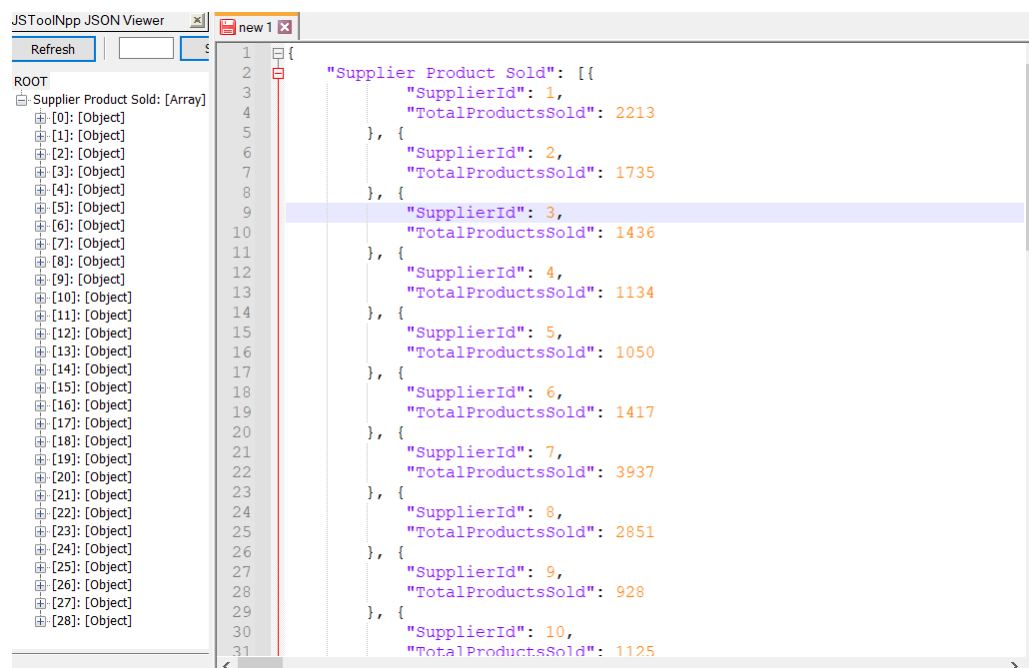
Removal of CTE with just a direct 3 table join makes the query more readable

## RELATIONAL AND JSON OUTPUT (29 ROWS AFFECTED)



	SupplierId	TotalProductsSold
1	1	2213
2	2	1735
3	3	1436
4	4	1134
5	5	1050
6	6	1417
7	7	3937
8	8	2851
9	9	928
10	10	1125
11	11	1436
12	12	4072
13	13	612
14	14	2500
15	15	2526
16	16	1573
17	17	1223

Query executed successfully. localhost:13001 (15.0 RTM) sa (66) Northwinds2022TSQLV7 00:00:00 29 rows



```
1 {  
2   "Supplier Product Sold": [  
3     {  
4       "SupplierId": 1,  
5       "TotalProductsSold": 2213  
6     }, {  
7       "SupplierId": 2,  
8       "TotalProductsSold": 1735  
9     }, {  
10      "SupplierId": 3,  
11      "TotalProductsSold": 1436  
12    }, {  
13      "SupplierId": 4,  
14      "TotalProductsSold": 1134  
15    }, {  
16      "SupplierId": 5,  
17      "TotalProductsSold": 1050  
18    }, {  
19      "SupplierId": 6,  
20      "TotalProductsSold": 1417  
21    }, {  
22      "SupplierId": 7,  
23      "TotalProductsSold": 3937  
24    }, {  
25      "SupplierId": 8,  
26      "TotalProductsSold": 2851  
27    }, {  
28      "SupplierId": 9,  
29      "TotalProductsSold": 928  
30    }, {  
31      "SupplierId": 10,  
32      "TotalProductsSold": 1125  
33    }  
34  ]  
35 }
```

### WORST #3: A MEDIUM QUERY

#### PROBLEM STATEMENT

How much revenue was generated by each gender (assuming only male/female gender) in 2015 using Northwinds2022TSQLV7

#### REASON IT IS A WORST

The many derived tables hinder readability

#### KEY AND STANDARD VIEW OF TABLES USED

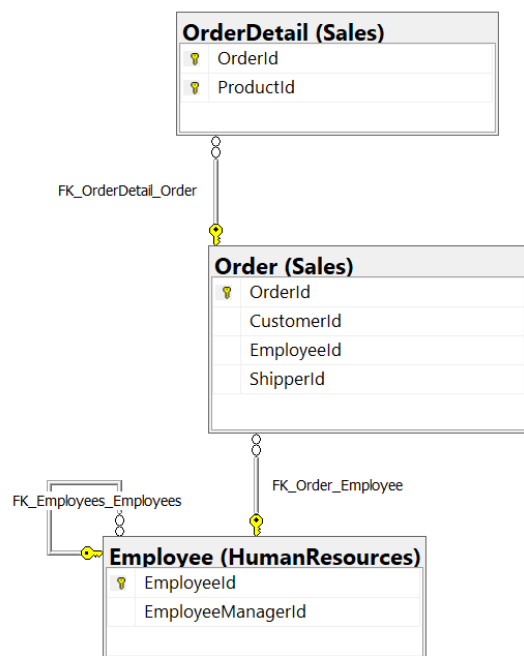




TABLE SHOWING COLUMNS PROJECTED IN THE END	
Table Name	Column Name
Derived	Gender
	RevenueAfterDiscount

TABLE SHOWING HOW PROJECTION SORTED (IF APPLICABLE)		
Table Name	Column Name	Sort Order
Derived	SUM(EmployeeRevenue)	DESC



## QUERY OF WORST

```

USE Northwinds2022TSQLV7;

SELECT E.Gender,
       FORMAT(SUM([O+OD].EmployeeRevenue), 'C') AS RevenueAfterDiscount
FROM
(
    SELECT E.EmployeeId,
           CASE
               WHEN E.EmployeeTitleOfCourtesy = N'Mr.' THEN
                   'Male'
               WHEN E.EmployeeTitleOfCourtesy IN ( N'Mrs.', N'Ms.' ) THEN
                   'Female'
               ELSE
                   'Unspecified'
           END AS Gender
    FROM HumanResources.Employee AS E
) AS E
INNER JOIN
(
    SELECT O.EmployeeId,
           SUM(OD.Quantity * OD.UnitPrice * (1.0 - OD.DiscountPercentage)) AS
EmployeeRevenue
    FROM Sales.[Order] AS O
        INNER JOIN Sales.OrderDetail AS OD
            ON OD.OrderId = O.OrderId
    WHERE YEAR(O.OrderDate) = 2015
    GROUP BY O.EmployeeId
) AS [O+OD]
    ON [O+OD].EmployeeId = E.EmployeeId
GROUP BY E.Gender
ORDER BY SUM([O+OD].EmployeeRevenue) DESC;

```

#### QUERY OF WORST CORRECTED

```
USE Northwinds2022TSQLV7;

WITH EmployeeGender
AS (SELECT E.EmployeeId,
CASE
    WHEN E.EmployeeTitleOfCourtesy = N'Mr.' THEN
        'Male'
    WHEN E.EmployeeTitleOfCourtesy IN ( N'Mrs.', N'Ms.' ) THEN
        'Female'
    ELSE
        'Unspecified'
END AS Gender
FROM HumanResources.Employee AS E),
EmployeeGeneratedRevenue2015
AS (SELECT O.EmployeeId,
SUM(OD.Quantity * OD.UnitPrice * (1.0 - OD.DiscountPercentage)) AS
EmployeeRevenue
FROM Sales.[Order] AS O
INNER JOIN Sales.OrderDetail AS OD
ON OD.OrderId = O.OrderId
WHERE YEAR(O.OrderDate) = 2015
GROUP BY O.EmployeeId)
SELECT G.Gender,
FORMAT(SUM(R.EmployeeRevenue), 'C') AS RevenueAfterDiscount
FROM EmployeeGender AS G
INNER JOIN EmployeeGeneratedRevenue2015 AS R
ON R.EmployeeId = G.EmployeeId
GROUP BY G.Gender
ORDER BY SUM(R.EmployeeRevenue) DESC;
```

---

#### HOW IT WAS CORRECTED:

Used CTEs to improve readability and give it a more logical flow

## RELATIONAL AND JSON OUTPUT (3 ROWS AFFECTED)

	Gender	RevenueAfterDiscount
1	Female	\$412,327.03
2	Male	\$134,314.03
3	Unspecified	\$70,444.14

Query executed successfully.

localhost,13001 (15.0 RTM) sa (66) Northwinds2022TSQLV7 00:00:00 3 rows

JSToolNpp JSON Viewer

Refresh

ROOT

Revenue Generated By Gender

[0]: [Object]

[1]: [Object]

[2]: [Object]

```
1  {
2    "Revenue Generated By Gender": [{
3      "Gender": "Female",
4      "RevenueAfterDiscount": "$412,327.03"
5    }, {
6      "Gender": "Male",
7      "RevenueAfterDiscount": "$134,314.03"
8    }, {
9      "Gender": "Unspecified",
10     "RevenueAfterDiscount": "$70,444.14"
11   }
12 ]
13 }
14
```