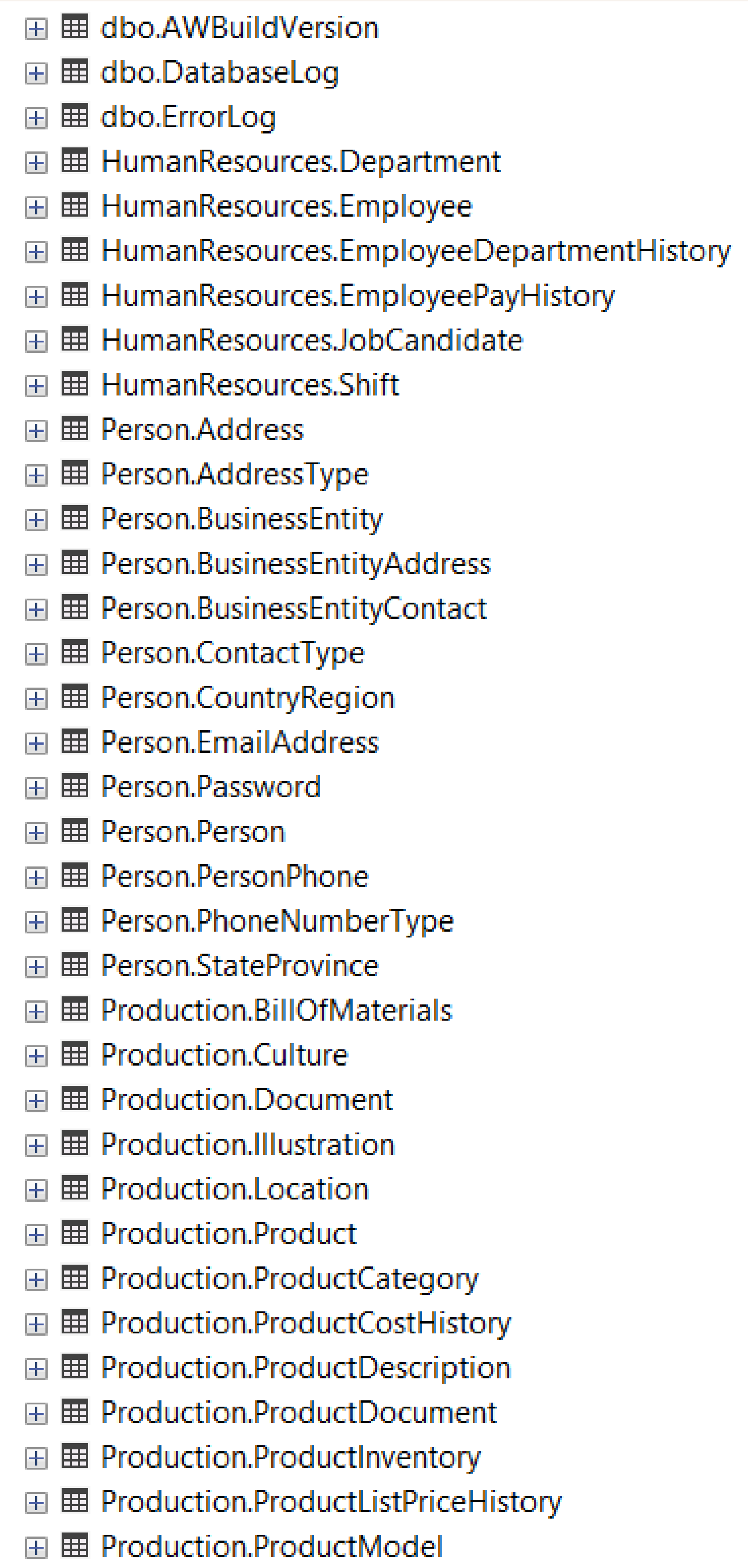
SQL Project

Total no of tables in Database 68

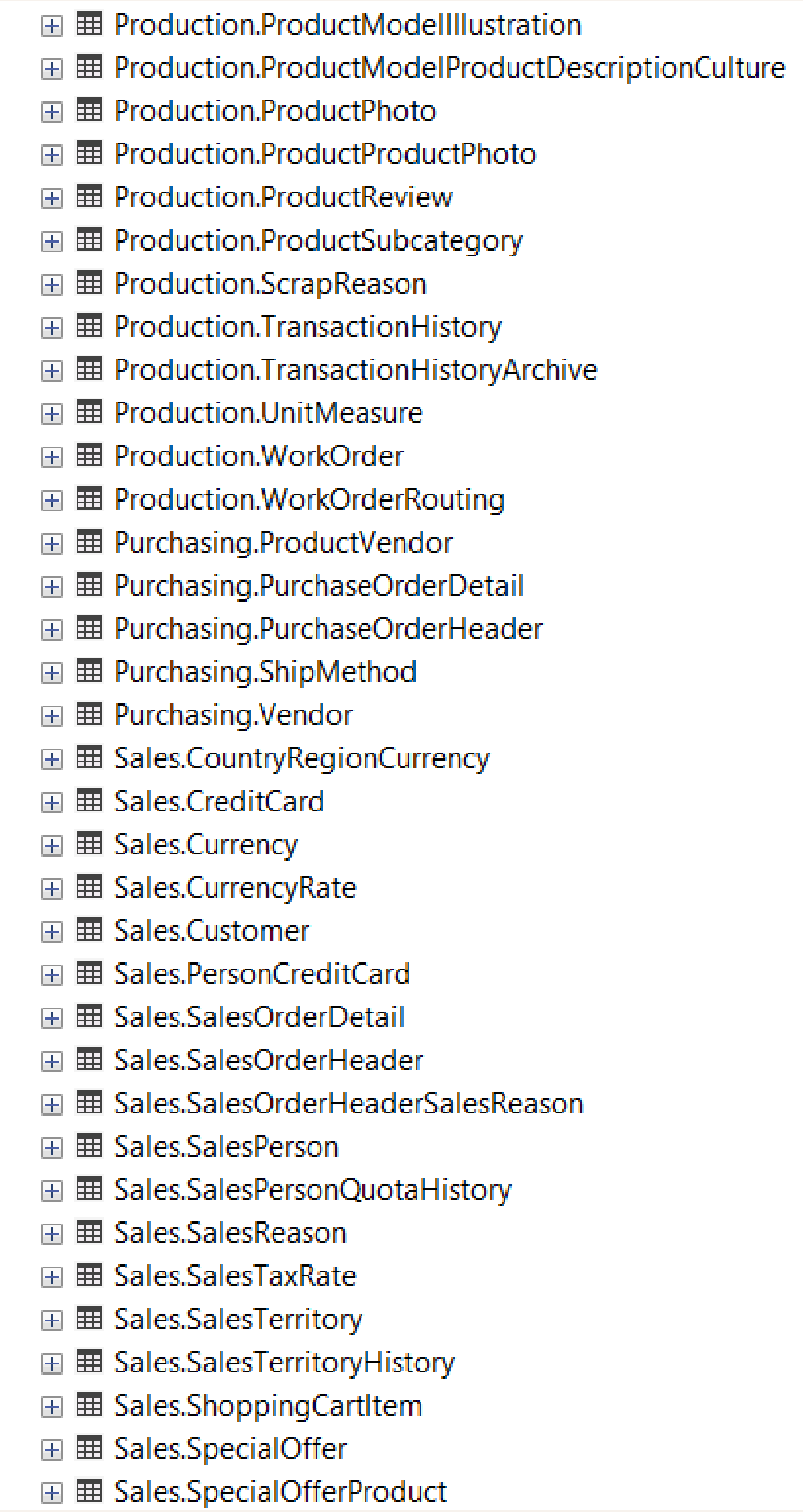
Tables i have used for analysis- 15

Going to solve 15 Questions

Source: Adventure Works Database 2019



dbo.AWBuildVersion
dbo.DatabaseLog
dbo.ErrorLog
HumanResources.Department
HumanResources.Employee
HumanResources.EmployeeDepartmentHistory
HumanResources.EmployeePayHistory
HumanResources.JobCandidate
HumanResources.Shift
Person.Address
Person.AddressType
Person.BusinessEntity
Person.BusinessEntityAddress
Person.BusinessEntityContact
Person.ContactType
Person.CountryRegion
Person.EmailAddress
Person.Password
Person.Person
Person.PersonPhone
Person.PhoneNumberType
Person.StateProvince
Production.BillOfMaterials
Production.Culture
Production.Document
Production.Illustration
Production.Location
Production.Product
Production.ProductCategory
Production.ProductCostHistory
Production.ProductDescription
Production.ProductDocument
Production.ProductInventory
Production.ProductListPriceHistory
Production.ProductModel



Production.ProductModelIllustration
Production.ProductModelProductDescriptionCulture
Production.ProductPhoto
Production.ProductProductPhoto
Production.ProductReview
Production.ProductSubcategory
Production.ScrapReason
Production.TransactionHistory
Production.TransactionHistoryArchive
Production.UnitMeasure
Production.WorkOrder
Production.WorkOrderRouting
Purchasing.ProductVendor
Purchasing.PurchaseOrderDetail
Purchasing.PurchaseOrderHeader
Purchasing.ShipMethod
Purchasing.Vendor
Sales.CountryRegionCurrency
Sales.CreditCard
Sales.Currency
Sales.CurrencyRate
Sales.Customer
Sales.PersonCreditCard
Sales.SalesOrderDetail
Sales.SalesOrderHeader
Sales.SalesOrderHeaderSalesReason
Sales.SalesPerson
Sales.SalesPersonQuotaHistory
Sales.SalesReason
Sales.SalesTaxRate
Sales.SalesTerritory
Sales.SalesTerritoryHistory
Sales.ShoppingCartItem
Sales.SpecialOffer
Sales.SpecialOfferProduct

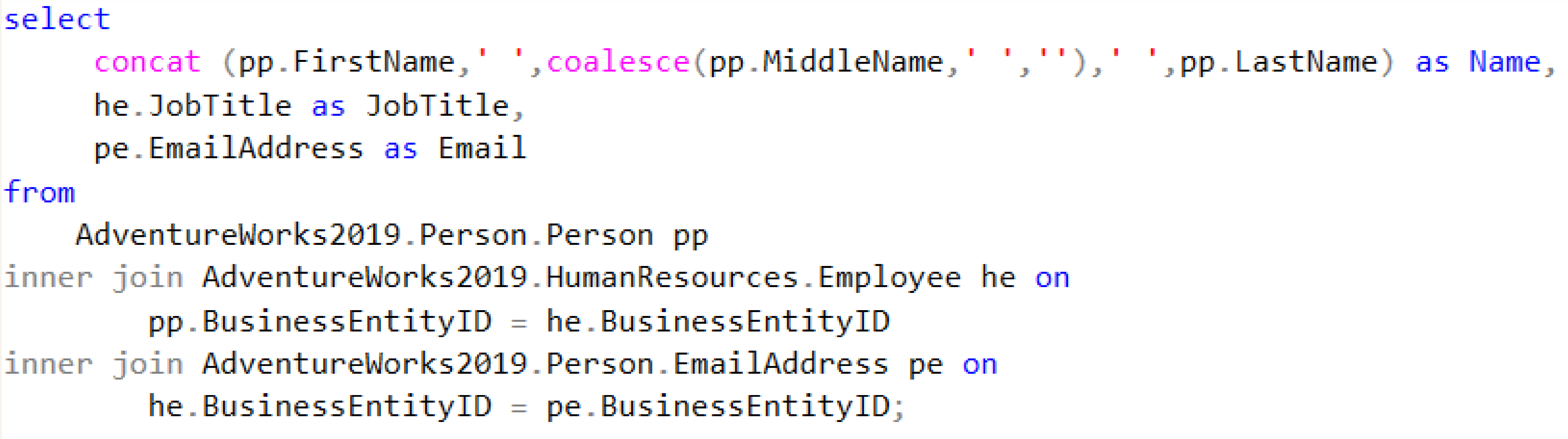
SQL Project

1. Retrieve the names and email addresses of all employees, including their job titles

SQL Project

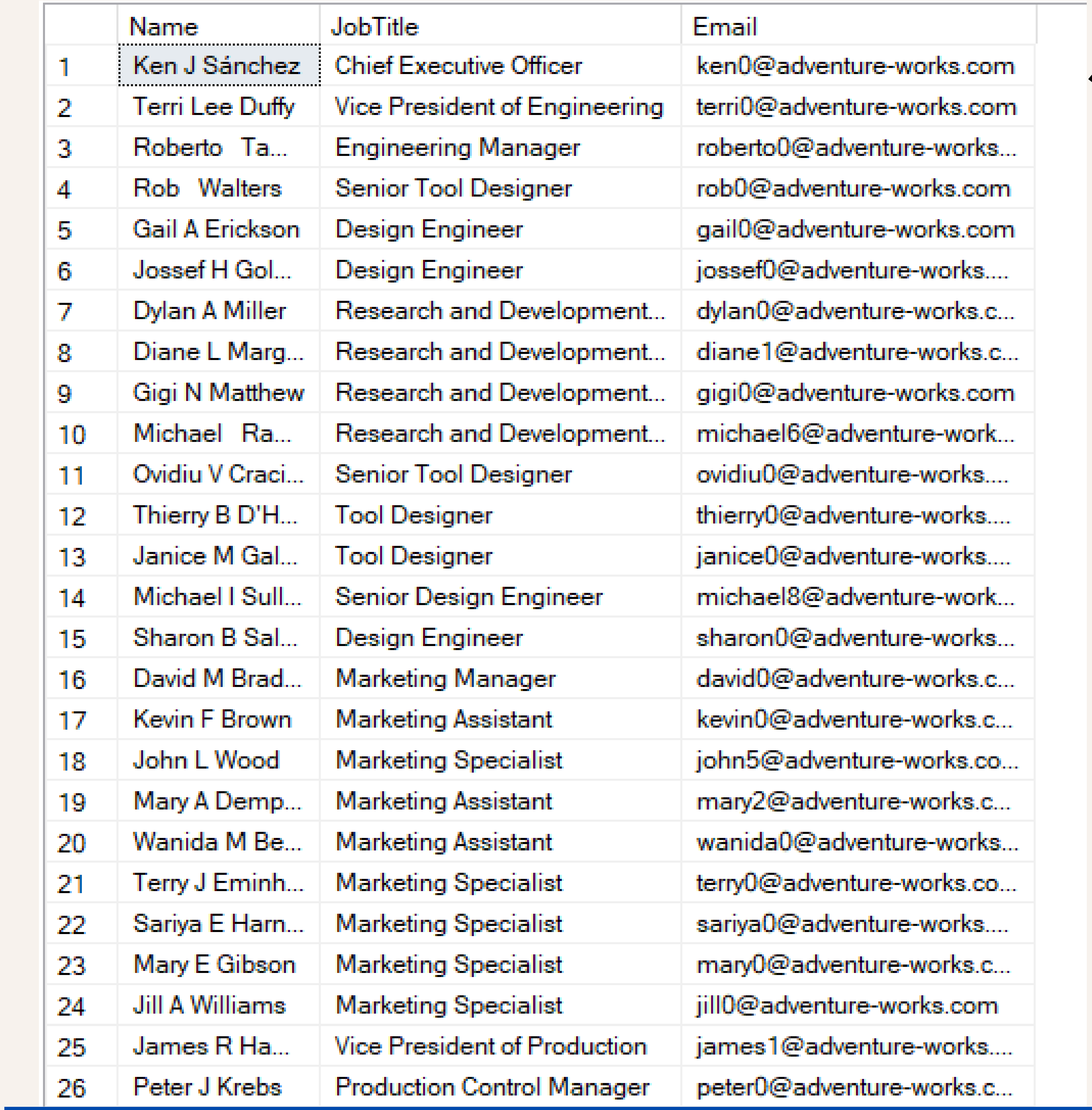
in Microsoft SQL Server

QUERY



```sql
select
     concat (pp.FirstName,' ',coalesce(pp.MiddleName,' ',''),' ',pp.LastName) as Name,
     he.JobTitle as JobTitle,
     pe.EmailAddress as Email
from
    AdventureWorks2019.Person.Person pp
inner join AdventureWorks2019.HumanResources.Employee he on
        pp.BusinessEntityID = he.BusinessEntityID
inner join AdventureWorks2019.Person.EmailAddress pe on
        he.BusinessEntityID = pe.BusinessEntityID;
```

OUTPUT



	Name	JobTitle	Email
1	Ken J Sánchez	Chief Executive Officer	ken0@adventure-works.com
2	Terri Lee Duffy	Vice President of Engineering	terri0@adventure-works.com
3	Roberto Ta...	Engineering Manager	roberto0@adventure-works...
4	Rob Walters	Senior Tool Designer	rob0@adventure-works.com
5	Gail A Erickson	Design Engineer	gail0@adventure-works.com
6	Jossef H Gol...	Design Engineer	jossef0@adventure-works....
7	Dylan A Miller	Research and Development...	dylan0@adventure-works.c...
8	Diane L Marg...	Research and Development...	diane1@adventure-works.c...
9	Gigi N Matthew	Research and Development...	gigi0@adventure-works.com
10	Michael Ra...	Research and Development...	michael6@adventure-work...
11	Ovidiu V Craci...	Senior Tool Designer	ovidiu0@adventure-works....
12	Thierry B D'H...	Tool Designer	thierry0@adventure-works....
13	Janice M Gal...	Tool Designer	janice0@adventure-works....
14	Michael I Sull...	Senior Design Engineer	michael8@adventure-work...
15	Sharon B Sal...	Design Engineer	sharon0@adventure-works...
16	David M Brad...	Marketing Manager	david0@adventure-works.c...
17	Kevin F Brown	Marketing Assistant	kevin0@adventure-works.c...
18	John L Wood	Marketing Specialist	john5@adventure-works.co...
19	Mary A Demp...	Marketing Assistant	mary2@adventure-works.c...
20	Wanida M Be...	Marketing Assistant	wanida0@adventure-works...
21	Terry J Eminh...	Marketing Specialist	terry0@adventure-works.co...
22	Sariya E Harn...	Marketing Specialist	sariya0@adventure-works....
23	Mary E Gibson	Marketing Specialist	mary0@adventure-works.c...
24	Jill A Williams	Marketing Specialist	jill0@adventure-works.com
25	James R Ha...	Vice President of Production	james1@adventure-works....
26	Peter J Krebs	Production Control Manager	peter0@adventure-works.c...

Displaying a sample of the data. The total number of rows in the complete output is 290.

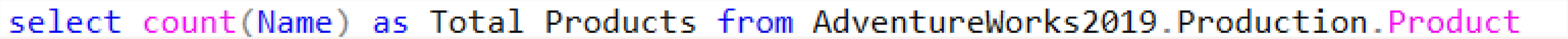
SQL Project

2. Find the total number of products in the database

SQL Project

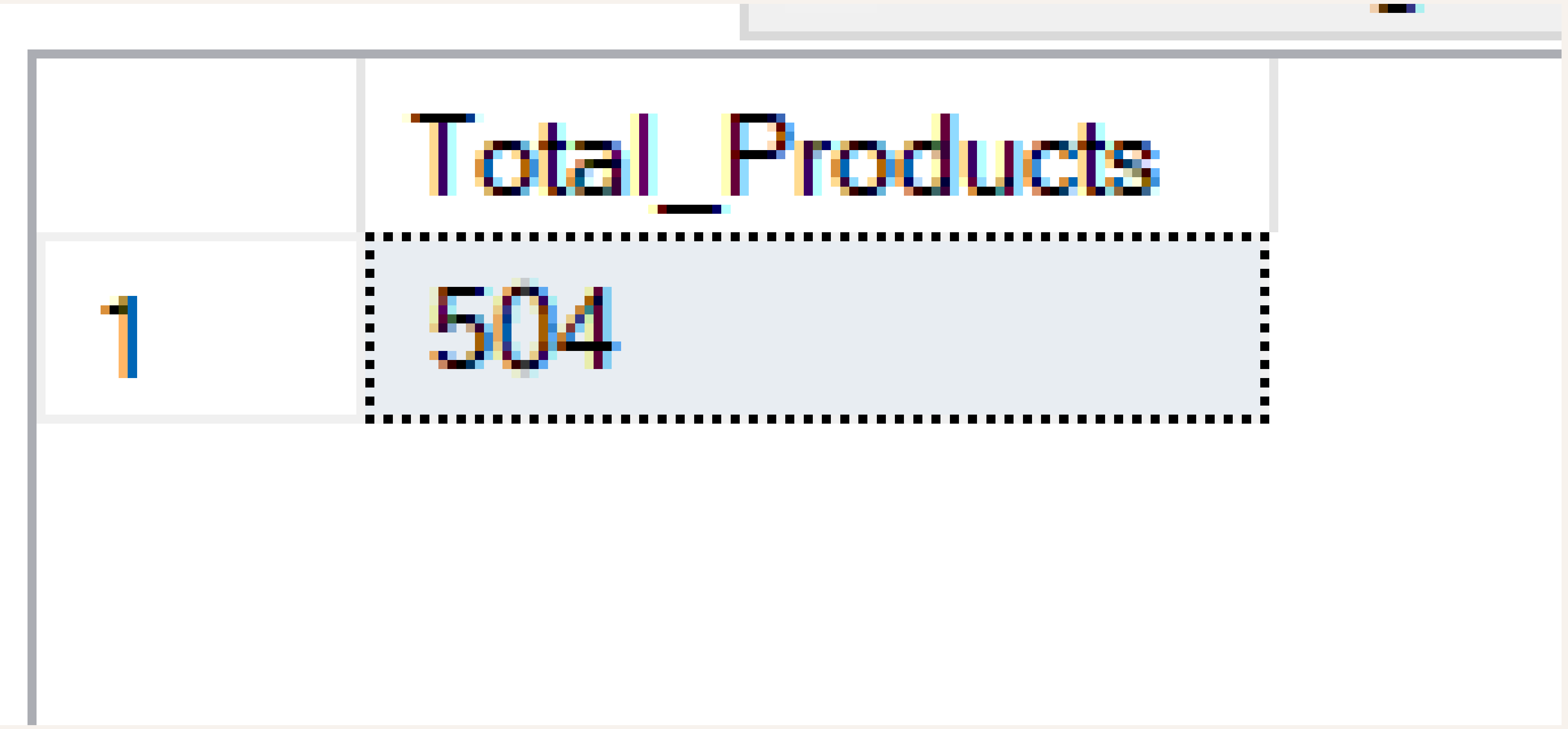
in Microsoft SQL Server

QUERY



```
select count(Name) as Total Products from AdventureWorks2019.Production.Product
```

OUTPUT



	Total_Products
1	504

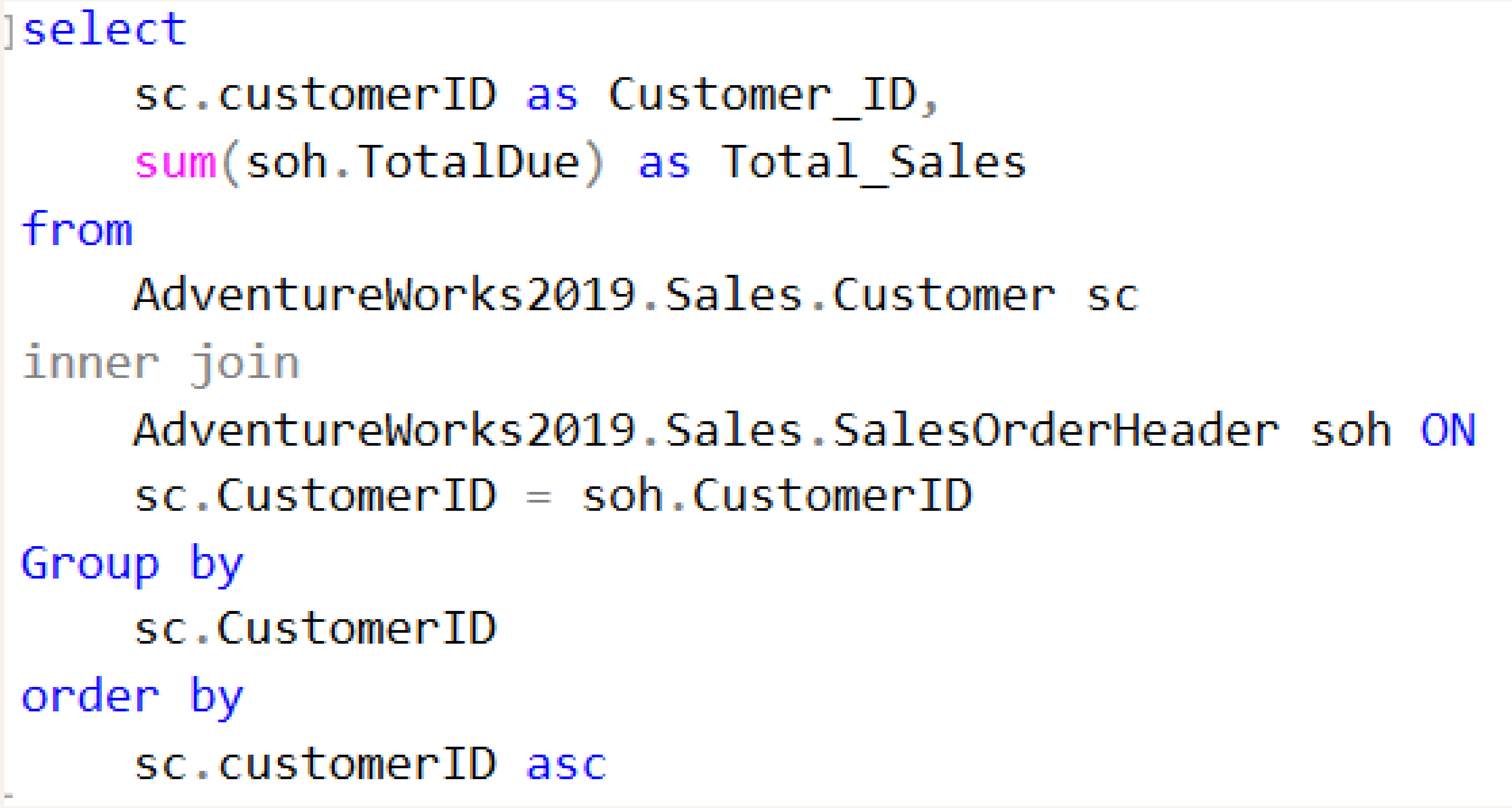
SQL Project

3. List the total sales amount for each customer

SQL Project

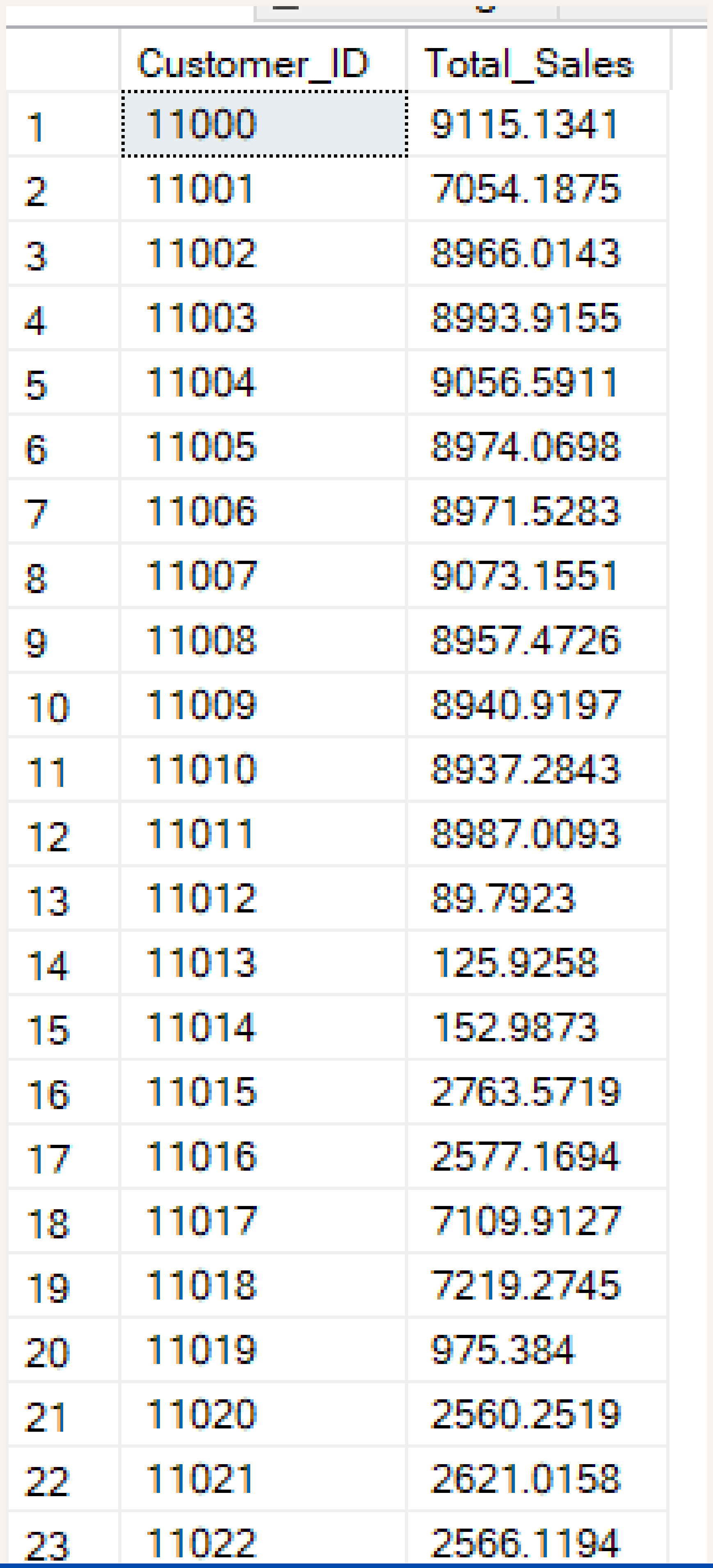
in Microsoft SQL Server

QUERY



```sql
select
    sc.customerID as Customer_ID,
    sum(soh.TotalDue) as Total_Sales
from
    AdventureWorks2019.Sales.Customer sc
inner join
    AdventureWorks2019.Sales.SalesOrderHeader soh ON
    sc.CustomerID = soh.CustomerID
Group by
    sc.CustomerID
order by
    sc.customerID asc
```

OUTPUT



	Customer_ID	Total_Sales
1	11000	9115.1341
2	11001	7054.1875
3	11002	8966.0143
4	11003	8993.9155
5	11004	9056.5911
6	11005	8974.0698
7	11006	8971.5283
8	11007	9073.1551
9	11008	8957.4726
10	11009	8940.9197
11	11010	8937.2843
12	11011	8987.0093
13	11012	89.7923
14	11013	125.9258
15	11014	152.9873
16	11015	2763.5719
17	11016	2577.1694
18	11017	7109.9127
19	11018	7219.2745
20	11019	975.384
21	11020	2560.2519
22	11021	2621.0158
23	11022	2566.1194

Displaying a sample of the data. The total number of rows in the complete output is 19119.

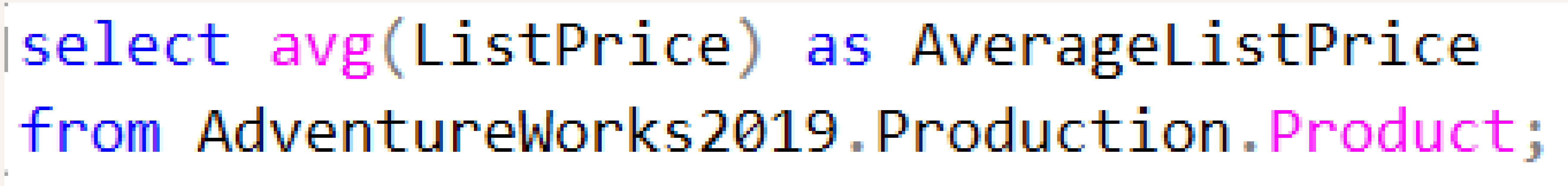
SQL Project

4. Calculate the average list price of all products.

SQL Project

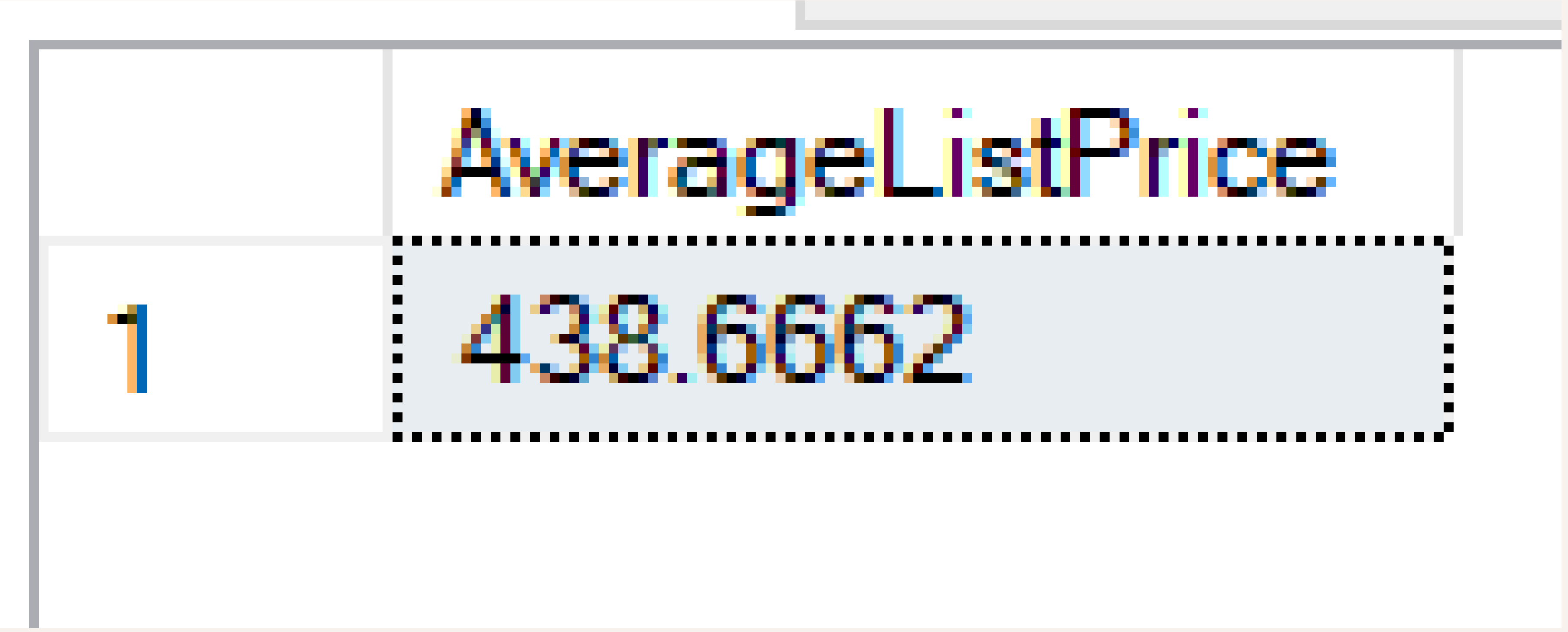
in Microsoft SQL Server

QUERY



```sql
select avg(ListPrice) as AverageListPrice
from AdventureWorks2019.Production.Product;
```

OUTPUT



	AverageListPrice
1	438.6662

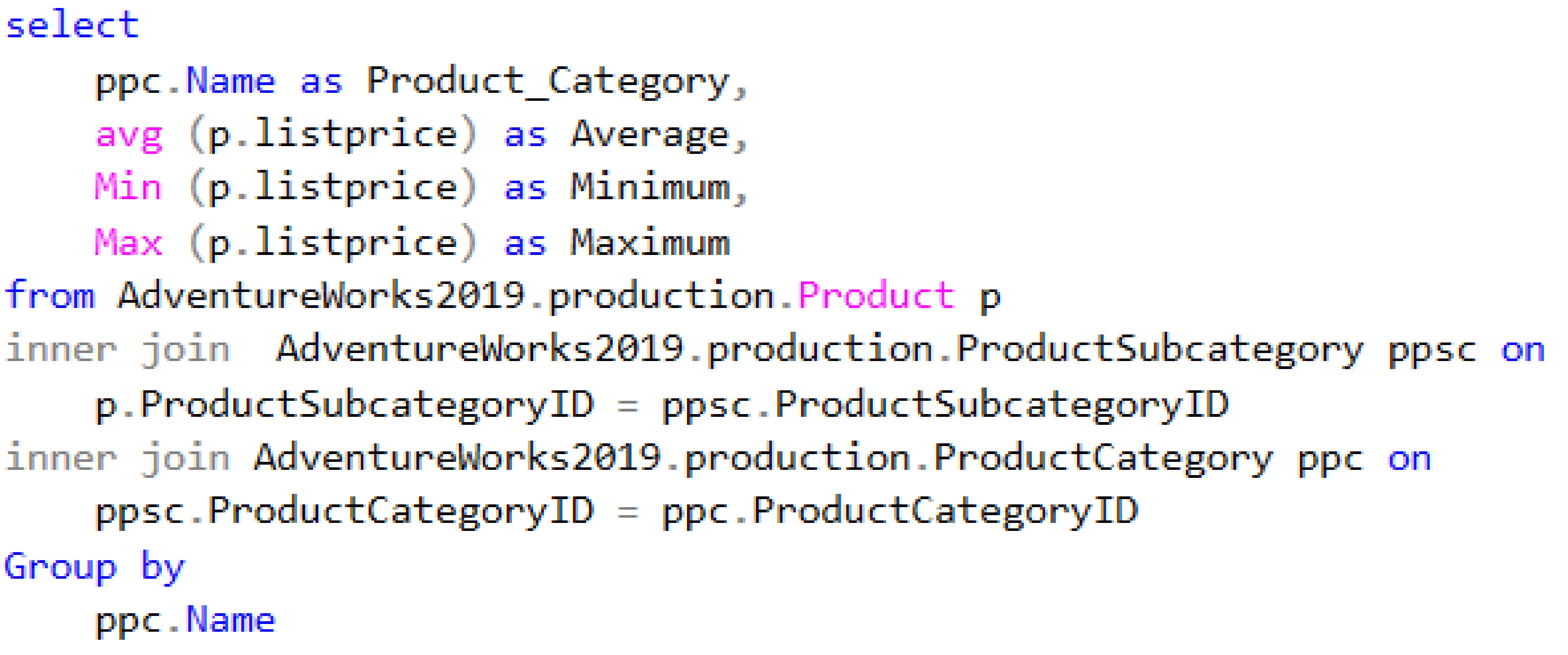
SQL Project

5. For each product category, calculate the average, minimum, and maximum list price of the products.

SQL Project

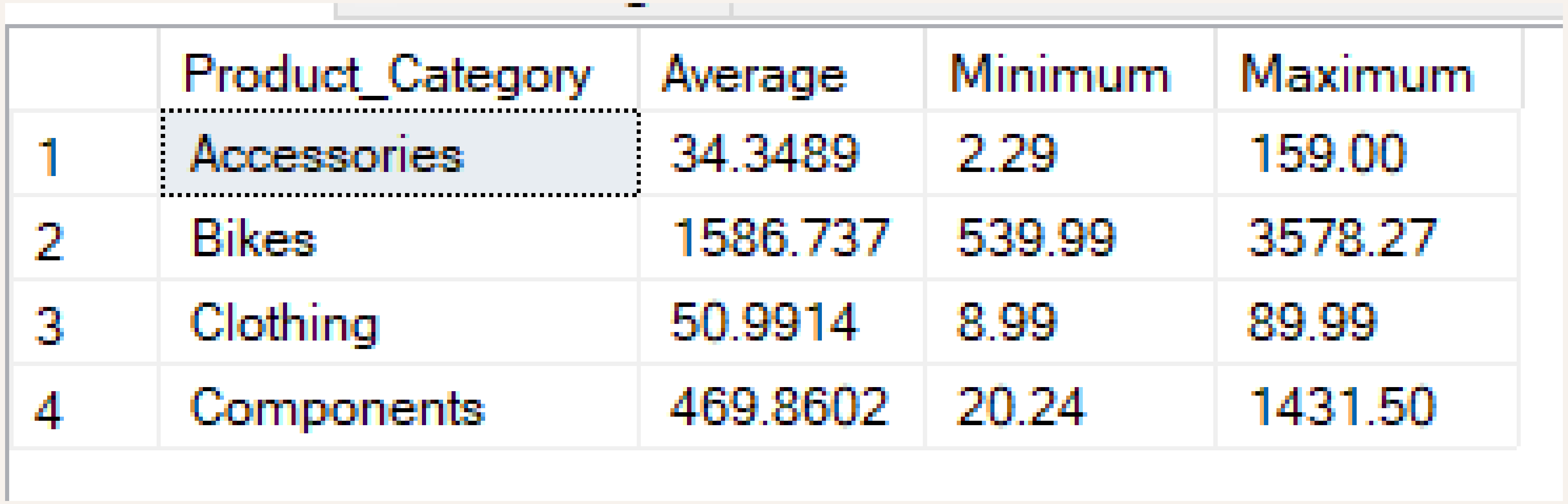
in Microsoft SQL Server

QUERY



```sql
select
    ppc.Name as Product_Category,
    avg (p.listprice) as Average,
    Min (p.listprice) as Minimum,
    Max (p.listprice) as Maximum
from AdventureWorks2019.production.Product p
inner join  AdventureWorks2019.production.ProductSubcategory ppsc on
    p.ProductSubcategoryID = ppsc.ProductSubcategoryID
inner join AdventureWorks2019.production.ProductCategory ppc on
    ppsc.ProductCategoryID = ppc.ProductCategoryID
Group by
    ppc.Name
```

OUTPUT



	Product_Category	Average	Minimum	Maximum
1	Accessories	34.3489	2.29	159.00
2	Bikes	1586.737	539.99	3578.27
3	Clothing	50.9914	8.99	89.99
4	Components	469.8602	20.24	1431.50

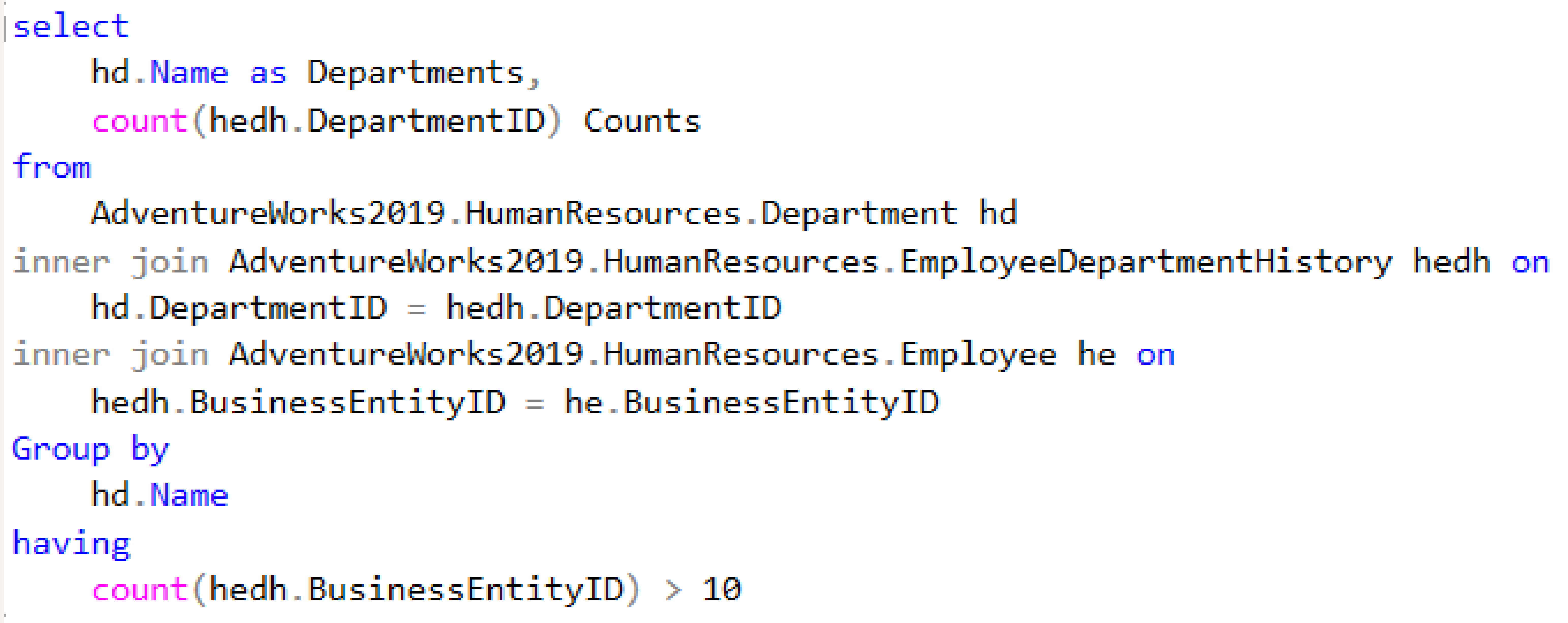
SQL Project

6. Find the departments with more than 10 employees

SQL Project

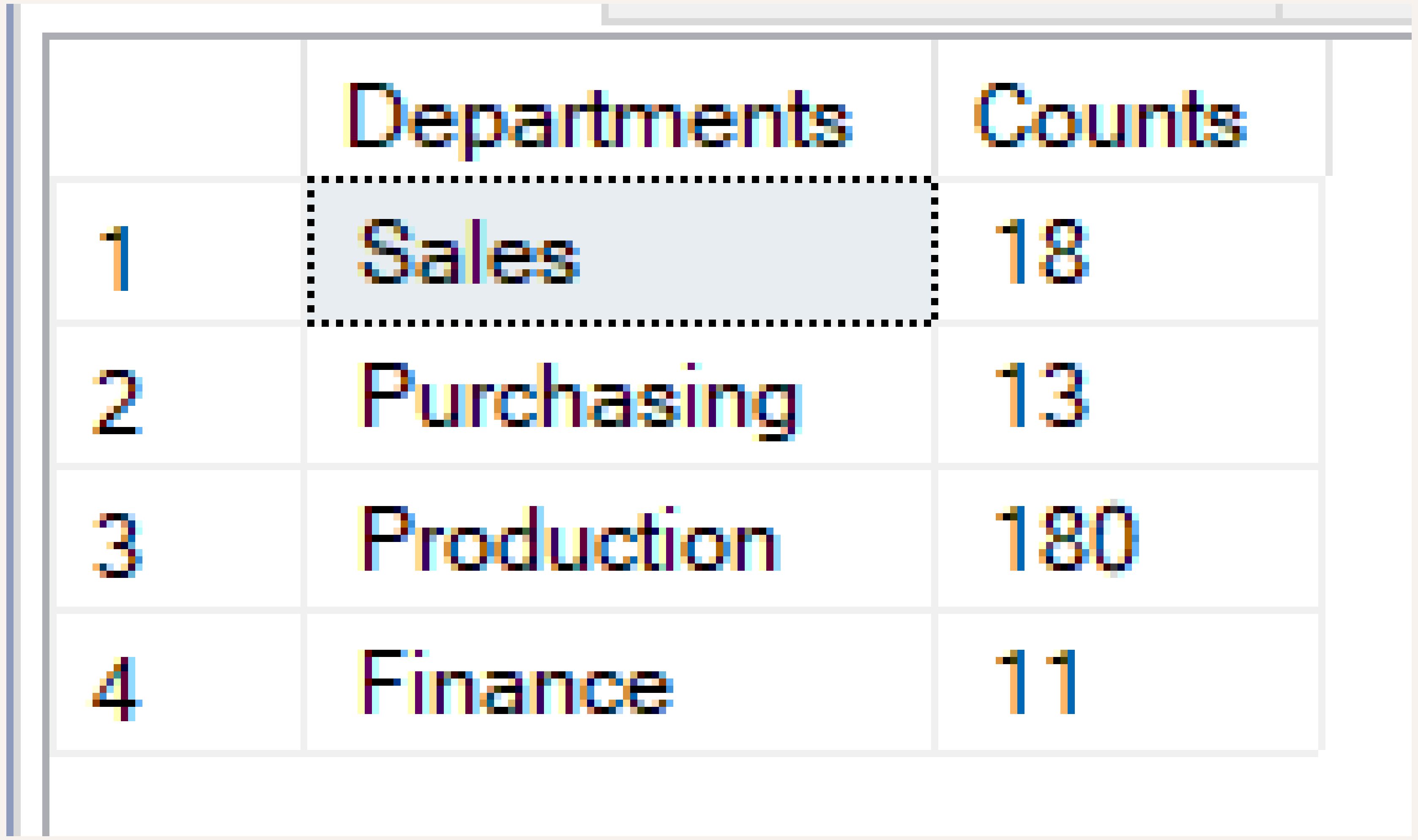
in Microsoft SQL Server

QUERY



```sql
select
    hd.Name as Departments,
    count(hedh.DepartmentID) Counts
from
    AdventureWorks2019.HumanResources.Department hd
inner join AdventureWorks2019.HumanResources.EmployeeDepartmentHistory hedh on
    hd.DepartmentID = hedh.DepartmentID
inner join AdventureWorks2019.HumanResources.Employee he on
    hedh.BusinessEntityID = he.BusinessEntityID
Group by
    hd.Name
having
    count(hedh.BusinessEntityID) > 10
```

OUTPUT



	Departments	Counts
1	Sales	18
2	Purchasing	13
3	Production	180
4	Finance	11

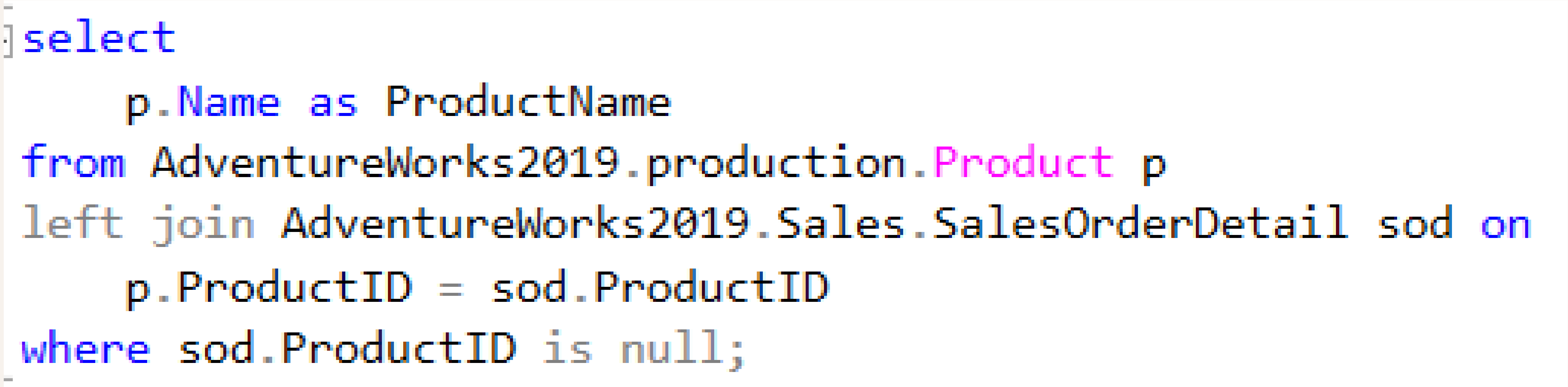
SQL Project

7. List the products that have never been sold

SQL Project

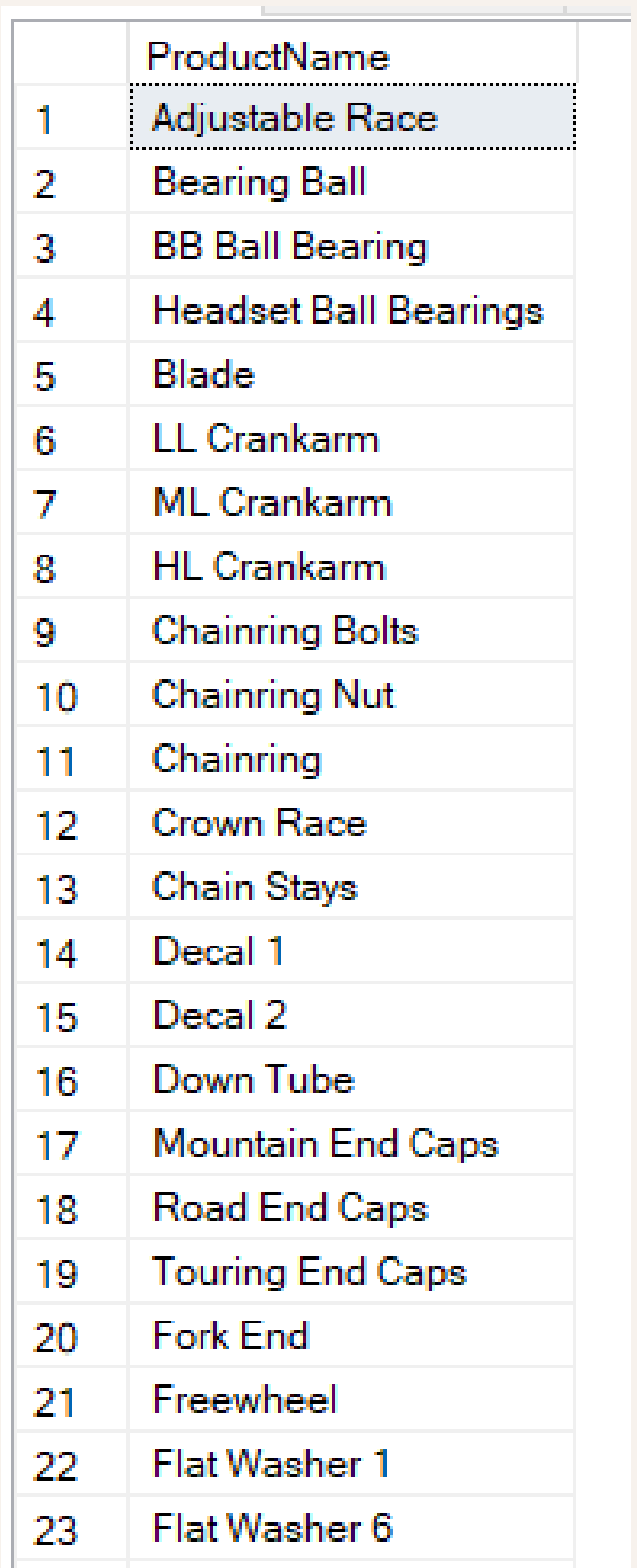
in Microsoft SQL Server

QUERY



```sql
select
    p.Name as ProductName
from AdventureWorks2019.production.Product p
left join AdventureWorks2019.Sales.SalesOrderDetail sod on
    p.ProductID = sod.ProductID
where sod.ProductID is null;
```

OUTPUT



	ProductName
1	Adjustable Race
2	Bearing Ball
3	BB Ball Bearing
4	Headset Ball Bearings
5	Blade
6	LL Crankarm
7	ML Crankarm
8	HL Crankarm
9	Chainring Bolts
10	Chainring Nut
11	Chainring
12	Crown Race
13	Chain Stays
14	Decal 1
15	Decal 2
16	Down Tube
17	Mountain End Caps
18	Road End Caps
19	Touring End Caps
20	Fork End
21	Freewheel
22	Flat Washer 1
23	Flat Washer 6

Displaying a sample of the data. The total number of rows in the complete output is 228.

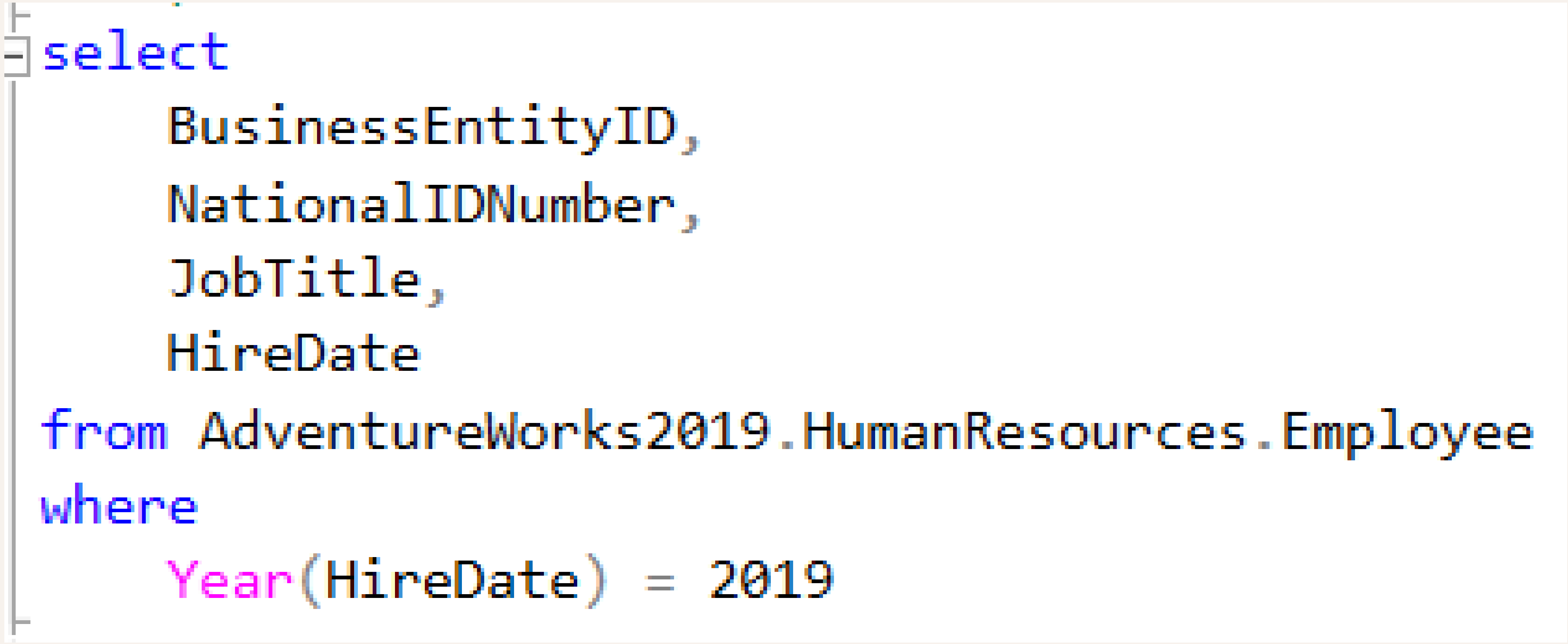
SQL Project

8. Find the employees who were hired in the year 2019

SQL Project

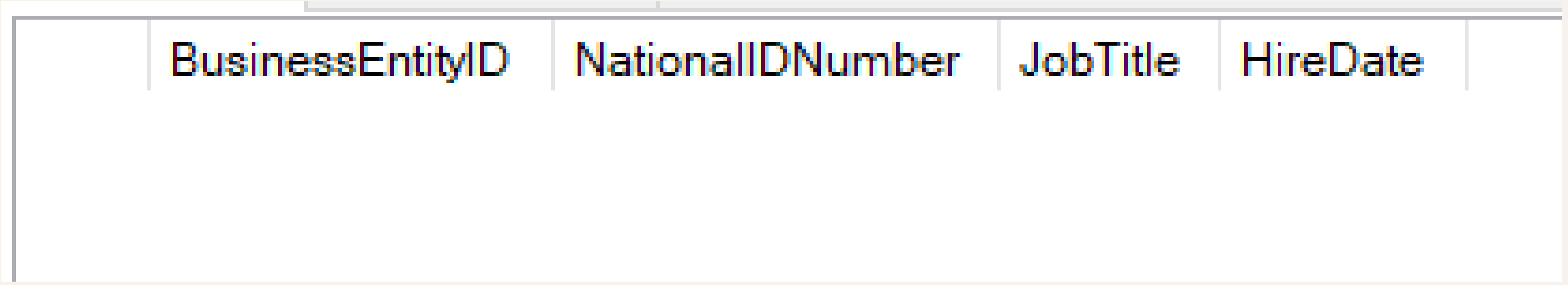
in Microsoft SQL Server

QUERY



```sql
select
    BusinessEntityID,
    NationalIDNumber,
    JobTitle,
    HireDate
from AdventureWorks2019.HumanResources.Employee
where
    Year(HireDate) = 2019
```

OUTPUT



	BusinessEntityID	NationalIDNumber	JobTitle	HireDate

Nobody was hired in 2019.

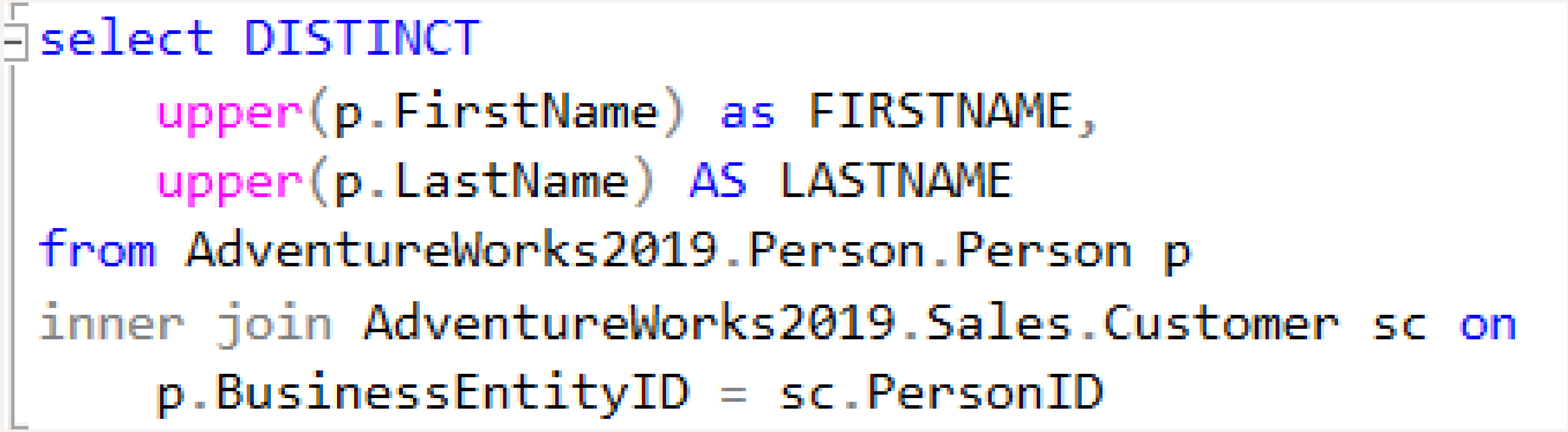
SQL Project

9. List the first names and last names of customers in uppercase

SQL Project

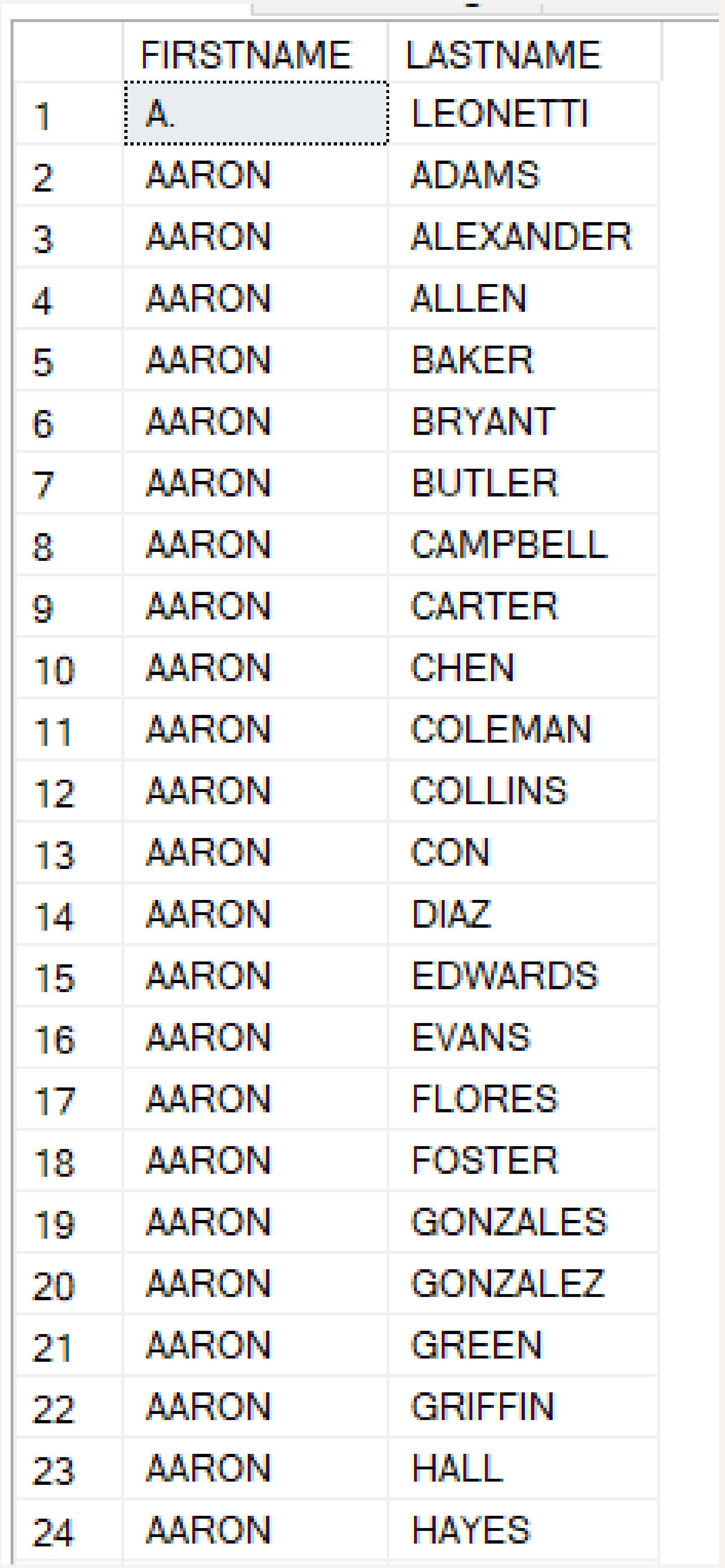
in Microsoft SQL Server

QUERY



```
select DISTINCT
    upper(p.FirstName) as FIRSTNAME,
    upper(p.LastName) AS LASTNAME
from AdventureWorks2019.Person.Person p
inner join AdventureWorks2019.Sales.Customer sc on
    p.BusinessEntityID = sc.PersonID
```

OUTPUT



	FIRSTNAME	LASTNAME
1	A.	LEONETTI
2	AARON	ADAMS
3	AARON	ALEXANDER
4	AARON	ALLEN
5	AARON	BAKER
6	AARON	BRYANT
7	AARON	BUTLER
8	AARON	CAMPBELL
9	AARON	CARTER
10	AARON	CHEN
11	AARON	COLEMAN
12	AARON	COLLINS
13	AARON	CON
14	AARON	DIAZ
15	AARON	EDWARDS
16	AARON	EVANS
17	AARON	FLORES
18	AARON	FOSTER
19	AARON	GONZALES
20	AARON	GONZALEZ
21	AARON	GREEN
22	AARON	GRIFFIN
23	AARON	HALL
24	AARON	HAYES

Displaying a sample of the data. The total number of rows in the complete output is 19018.

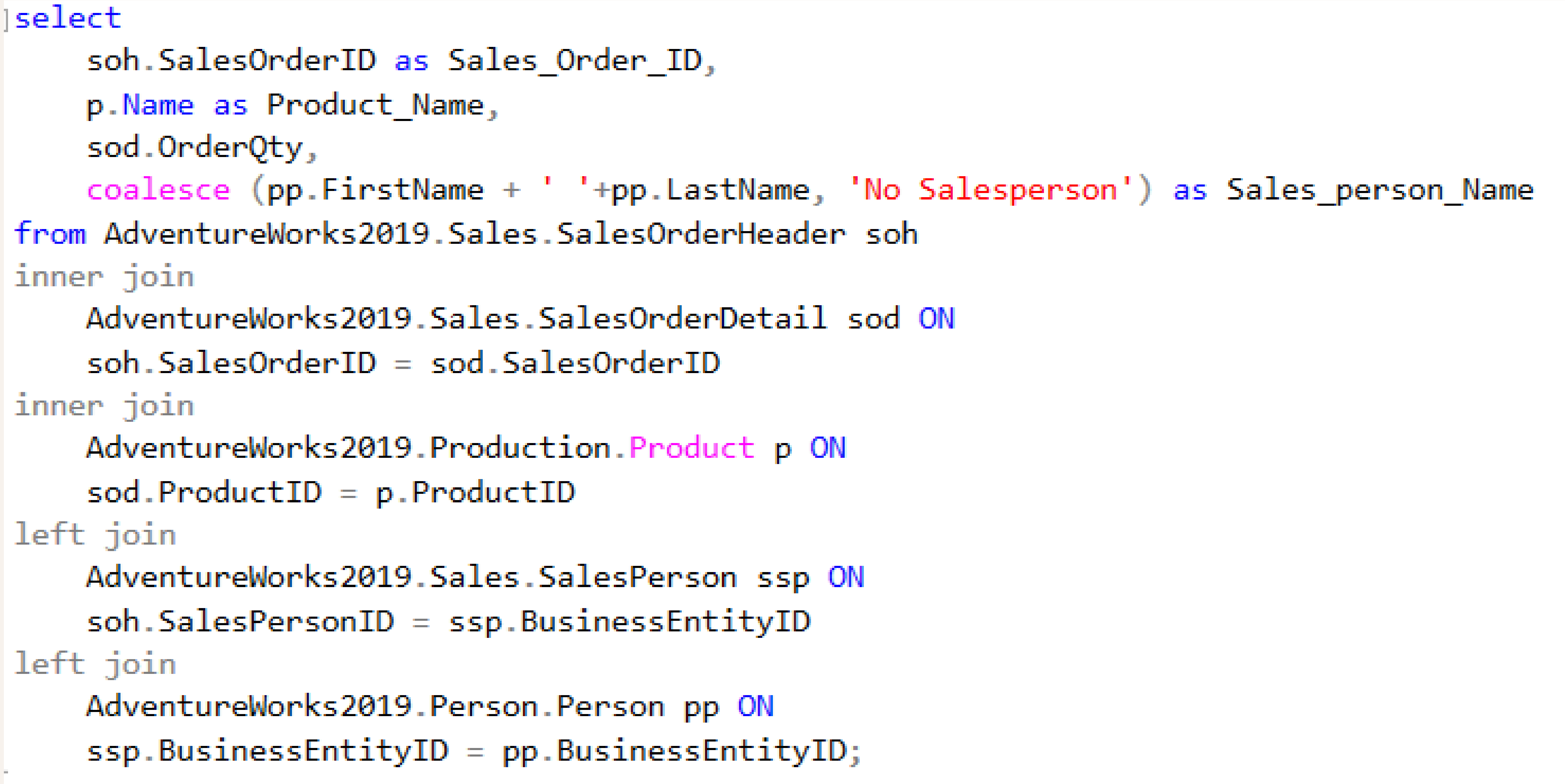
SQL Project

10. Retrieve the sales order ID, product name, order quantity, and the salesperson's name for all sales orders.

SQL Project

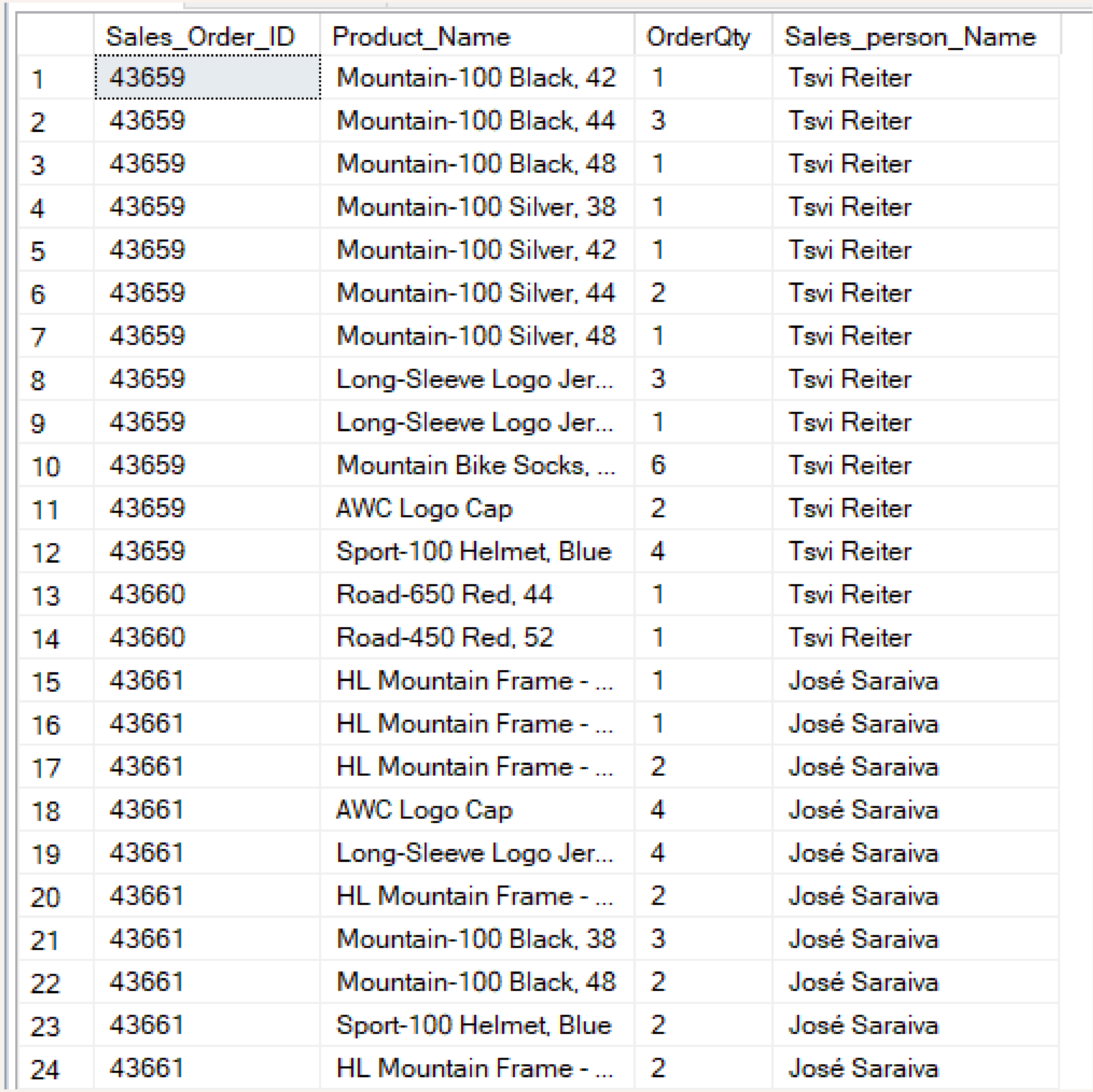
in Microsoft SQL Server

QUERY



```
select
    soh.SalesOrderID as Sales_Order_ID,
    p.Name as Product_Name,
    sod.OrderQty,
    coalesce (pp.FirstName + ' '+pp.LastName, 'No Salesperson') as Sales_person_Name
from AdventureWorks2019.Sales.SalesOrderHeader soh
inner join
    AdventureWorks2019.Sales.SalesOrderDetail sod ON
    soh.SalesOrderID = sod.SalesOrderID
inner join
    AdventureWorks2019.Production.Product p ON
    sod.ProductID = p.ProductID
left join
    AdventureWorks2019.Sales.SalesPerson ssp ON
    soh.SalesPersonID = ssp.BusinessEntityID
left join
    AdventureWorks2019.Person.Person pp ON
    ssp.BusinessEntityID = pp.BusinessEntityID;
```

OUTPUT



	Sales_Order_ID	Product_Name	OrderQty	Sales_person_Name
1	43659	Mountain-100 Black, 42	1	Tsvi Reiter
2	43659	Mountain-100 Black, 44	3	Tsvi Reiter
3	43659	Mountain-100 Black, 48	1	Tsvi Reiter
4	43659	Mountain-100 Silver, 38	1	Tsvi Reiter
5	43659	Mountain-100 Silver, 42	1	Tsvi Reiter
6	43659	Mountain-100 Silver, 44	2	Tsvi Reiter
7	43659	Mountain-100 Silver, 48	1	Tsvi Reiter
8	43659	Long-Sleeve Logo Jer...	3	Tsvi Reiter
9	43659	Long-Sleeve Logo Jer...	1	Tsvi Reiter
10	43659	Mountain Bike Socks, ...	6	Tsvi Reiter
11	43659	AWC Logo Cap	2	Tsvi Reiter
12	43659	Sport-100 Helmet, Blue	4	Tsvi Reiter
13	43660	Road-650 Red, 44	1	Tsvi Reiter
14	43660	Road-450 Red, 52	1	Tsvi Reiter
15	43661	HL Mountain Frame - ...	1	José Saraiva
16	43661	HL Mountain Frame - ...	1	José Saraiva
17	43661	HL Mountain Frame - ...	2	José Saraiva
18	43661	AWC Logo Cap	4	José Saraiva
19	43661	Long-Sleeve Logo Jer...	4	José Saraiva
20	43661	HL Mountain Frame - ...	2	José Saraiva
21	43661	Mountain-100 Black, 38	3	José Saraiva
22	43661	Mountain-100 Black, 48	2	José Saraiva
23	43661	Sport-100 Helmet, Blue	2	José Saraiva
24	43661	HL Mountain Frame - ...	2	José Saraiva

Displaying a sample of the data. The total number of rows in the complete output is 121317.

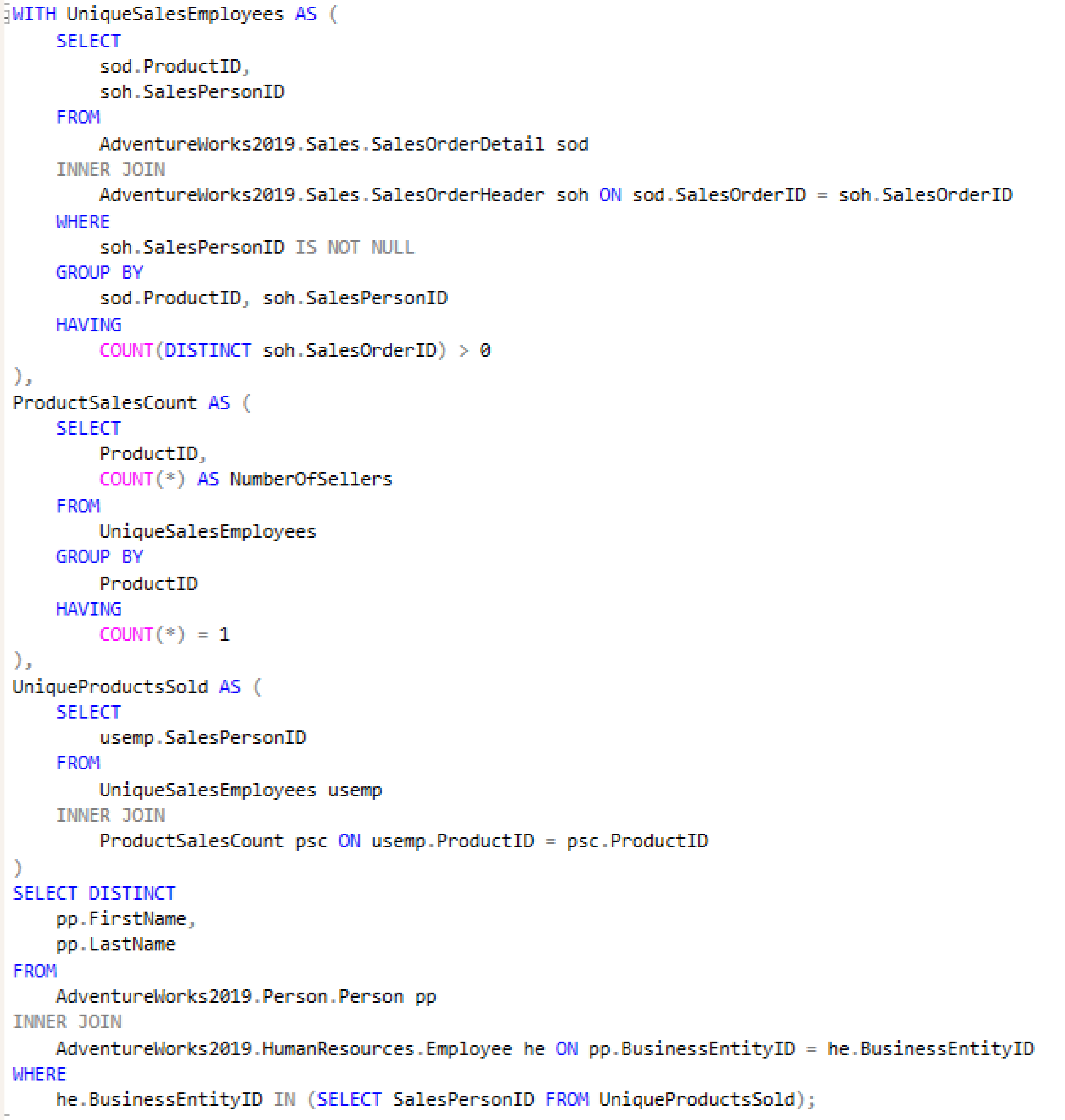
SQL Project

11. Find the names of employees who have sold products that were never sold by any other employee.

SQL Project

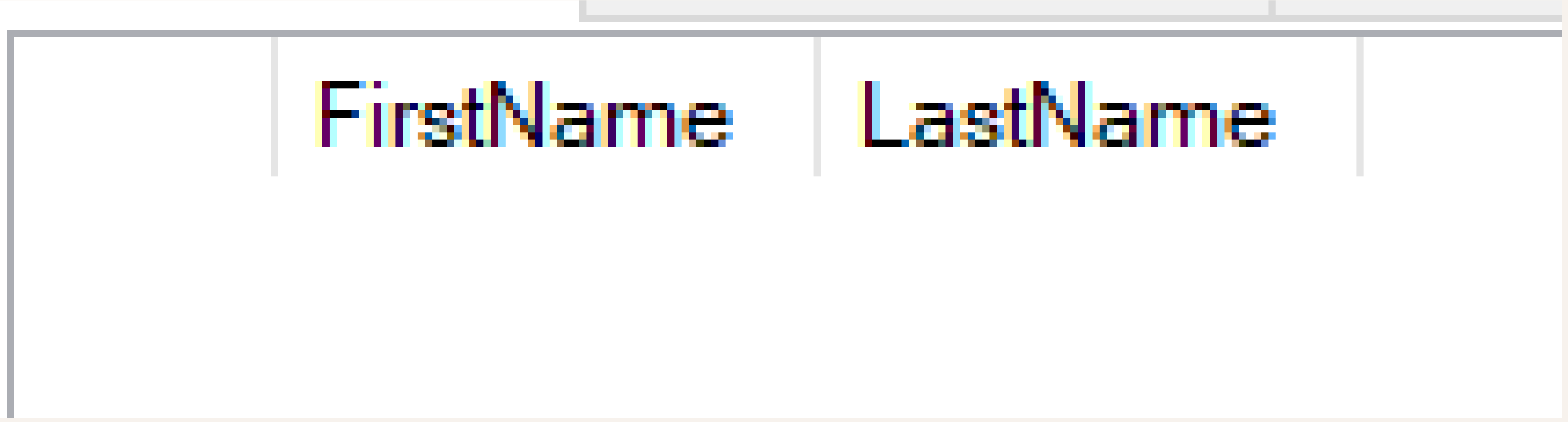
in Microsoft SQL Server

QUERY



```sql
WITH UniqueSalesEmployees AS (
    SELECT
        sod.ProductID,
        soh.SalesPersonID
    FROM
        AdventureWorks2019.Sales.SalesOrderDetail sod
    INNER JOIN
        AdventureWorks2019.Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
    WHERE
        soh.SalesPersonID IS NOT NULL
    GROUP BY
        sod.ProductID, soh.SalesPersonID
    HAVING
        COUNT(DISTINCT soh.SalesOrderID) > 0
),
ProductSalesCount AS (
    SELECT
        ProductID,
        COUNT(*) AS NumberOfSellers
    FROM
        UniqueSalesEmployees
    GROUP BY
        ProductID
    HAVING
        COUNT(*) = 1
),
UniqueProductsSold AS (
    SELECT
        usemp.SalesPersonID
    FROM
        UniqueSalesEmployees usemp
    INNER JOIN
        ProductSalesCount psc ON usemp.ProductID = psc.ProductID
)
SELECT DISTINCT
    pp.FirstName,
    pp.LastName
FROM
    AdventureWorks2019.Person.Person pp
INNER JOIN
    AdventureWorks2019.HumanResources.Employee he ON pp.BusinessEntityID = he.BusinessEntityID
WHERE
    he.BusinessEntityID IN (SELECT SalesPersonID FROM UniqueProductsSold);
```

OUTPUT



FirstName	LastName

Result: No such employees exist in the data.

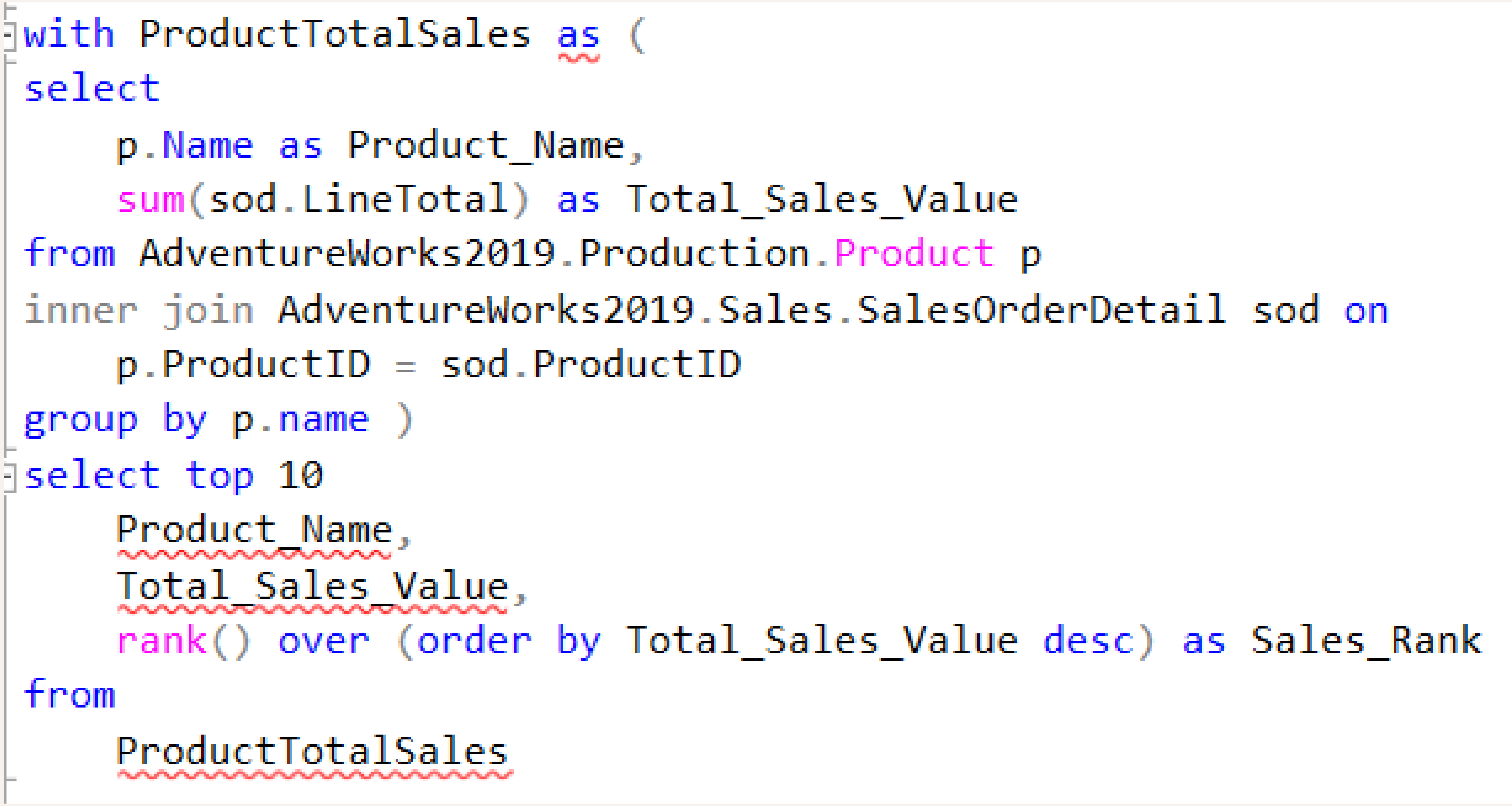
SQL Project

12. Rank the products based on their total sales amount and display the top 10 products.

SQL Project

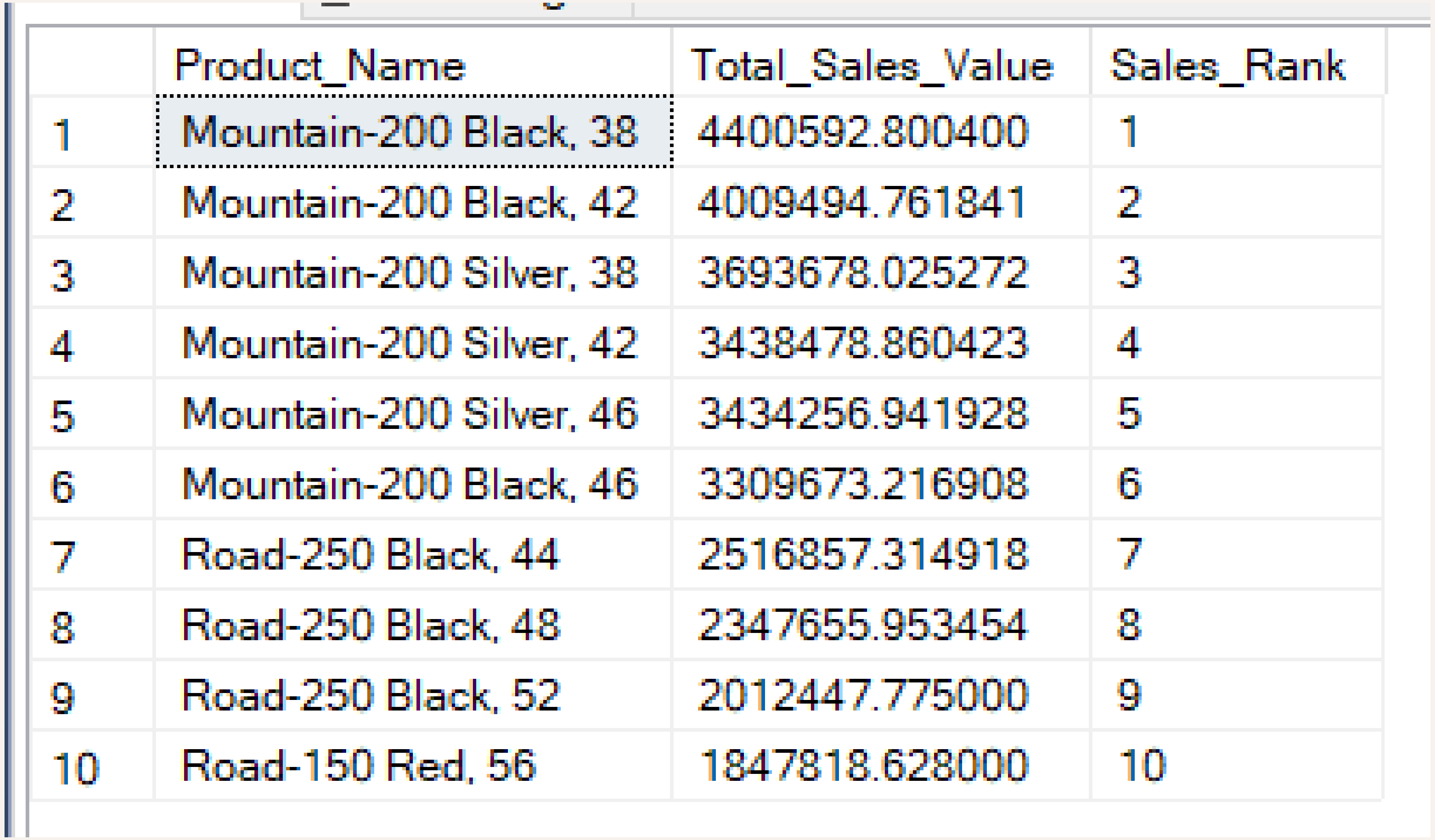
in Microsoft SQL Server

QUERY



```sql
with ProductTotalSales as (
select
    p.Name as Product_Name,
    sum(sod.LineTotal) as Total_Sales_Value
from AdventureWorks2019.Production.Product p
inner join AdventureWorks2019.Sales.SalesOrderDetail sod on
    p.ProductID = sod.ProductID
group by p.name )
select top 10
    Product_Name,
    Total_Sales_Value,
    rank() over (order by Total_Sales_Value desc) as Sales_Rank
from
    ProductTotalSales
```

OUTPUT



	Product_Name	Total_Sales_Value	Sales_Rank
1	Mountain-200 Black, 38	4400592.800400	1
2	Mountain-200 Black, 42	4009494.761841	2
3	Mountain-200 Silver, 38	3693678.025272	3
4	Mountain-200 Silver, 42	3438478.860423	4
5	Mountain-200 Silver, 46	3434256.941928	5
6	Mountain-200 Black, 46	3309673.216908	6
7	Road-250 Black, 44	2516857.314918	7
8	Road-250 Black, 48	2347655.953454	8
9	Road-250 Black, 52	2012447.775000	9
10	Road-150 Red, 56	1847818.628000	10

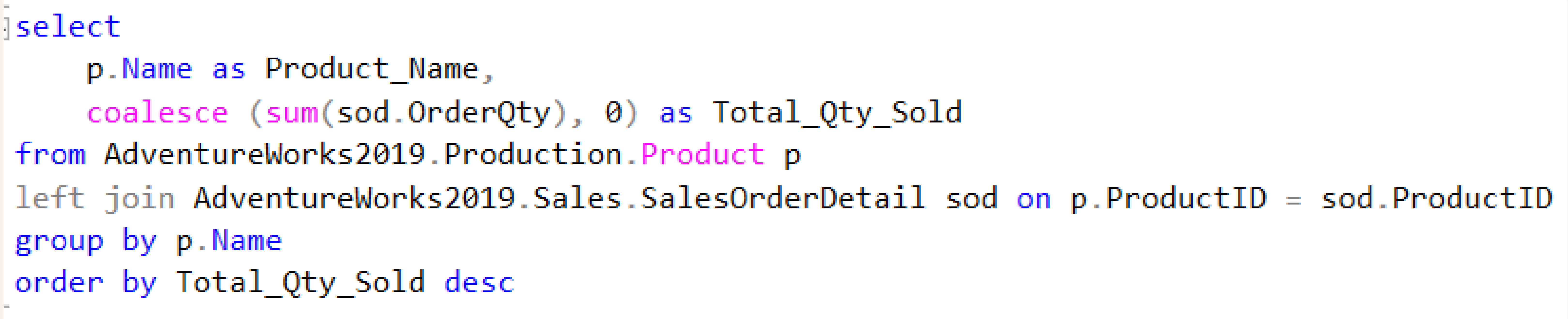
SQL Project

13. Retrieve a list of all products and the total quantity sold, including products that have never been sold.

SQL Project

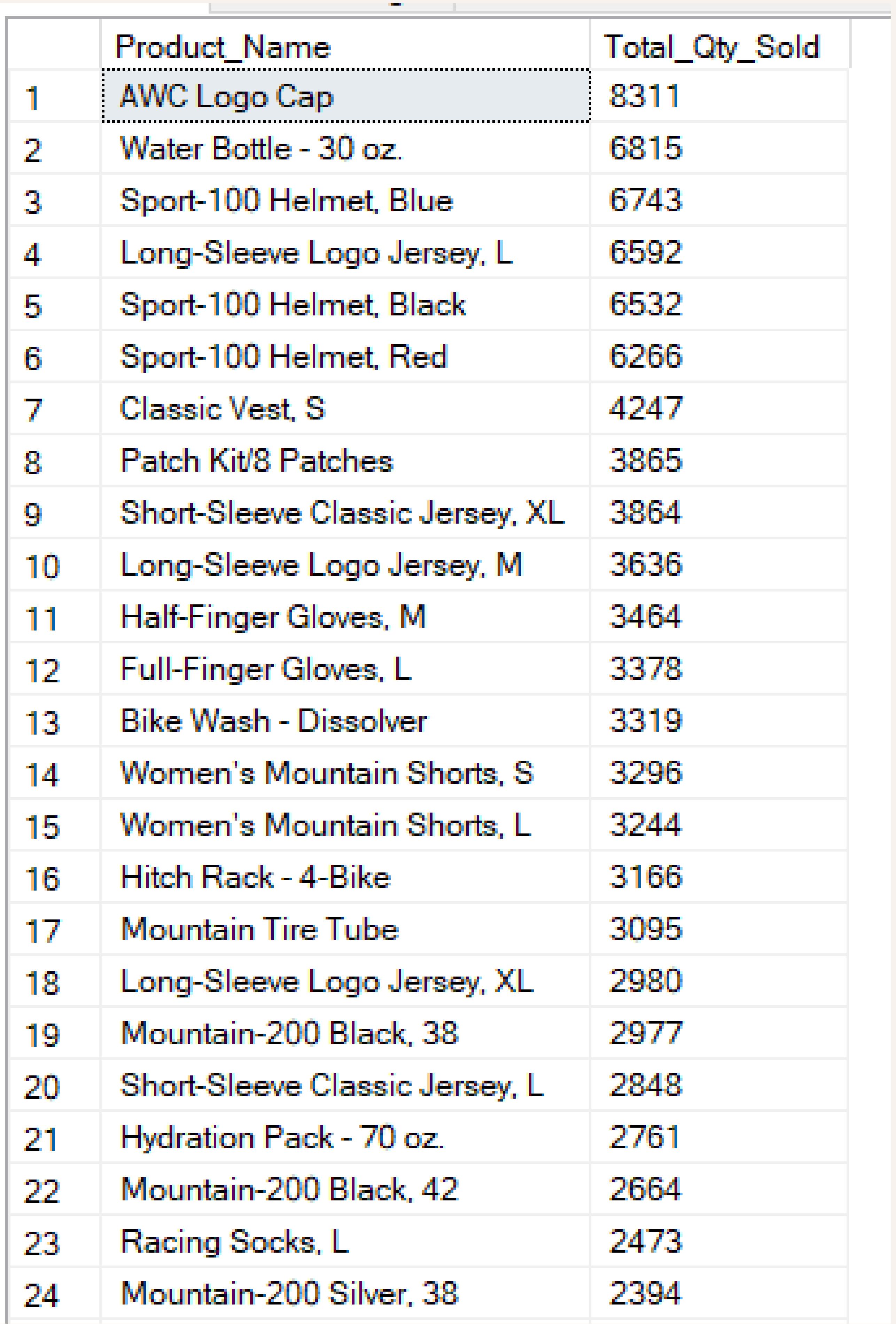
in Microsoft SQL Server

QUERY



```
select
    p.Name as Product_Name,
    coalesce (sum(sod.OrderQty), 0) as Total_Qty_Sold
from AdventureWorks2019.Production.Product p
left join AdventureWorks2019.Sales.SalesOrderDetail sod on p.ProductID = sod.ProductID
group by p.Name
order by Total_Qty_Sold desc
```

OUTPUT



	Product_Name	Total_Qty_Sold
1	AWC Logo Cap	8311
2	Water Bottle - 30 oz.	6815
3	Sport-100 Helmet, Blue	6743
4	Long-Sleeve Logo Jersey, L	6592
5	Sport-100 Helmet, Black	6532
6	Sport-100 Helmet, Red	6266
7	Classic Vest, S	4247
8	Patch Kit/8 Patches	3865
9	Short-Sleeve Classic Jersey, XL	3864
10	Long-Sleeve Logo Jersey, M	3636
11	Half-Finger Gloves, M	3464
12	Full-Finger Gloves, L	3378
13	Bike Wash - Dissolver	3319
14	Women's Mountain Shorts, S	3296
15	Women's Mountain Shorts, L	3244
16	Hitch Rack - 4-Bike	3166
17	Mountain Tire Tube	3095
18	Long-Sleeve Logo Jersey, XL	2980
19	Mountain-200 Black, 38	2977
20	Short-Sleeve Classic Jersey, L	2848
21	Hydration Pack - 70 oz.	2761
22	Mountain-200 Black, 42	2664
23	Racing Socks, L	2473
24	Mountain-200 Silver, 38	2394

Displaying a sample of the data. The total number of rows in the complete output is 504.

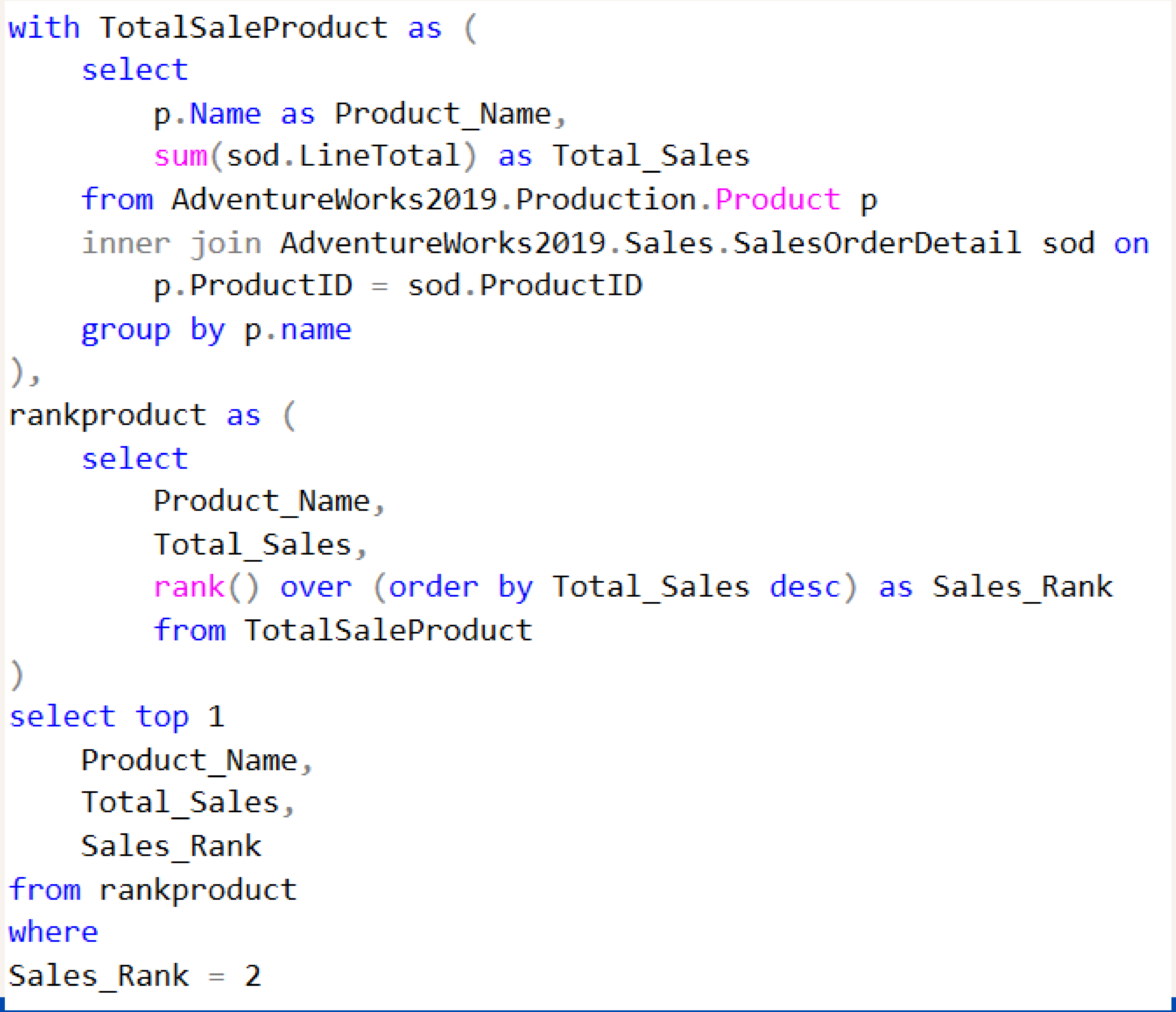
SQL Project

14. Find the second highest selling product in terms of total sales amount.

SQL Project

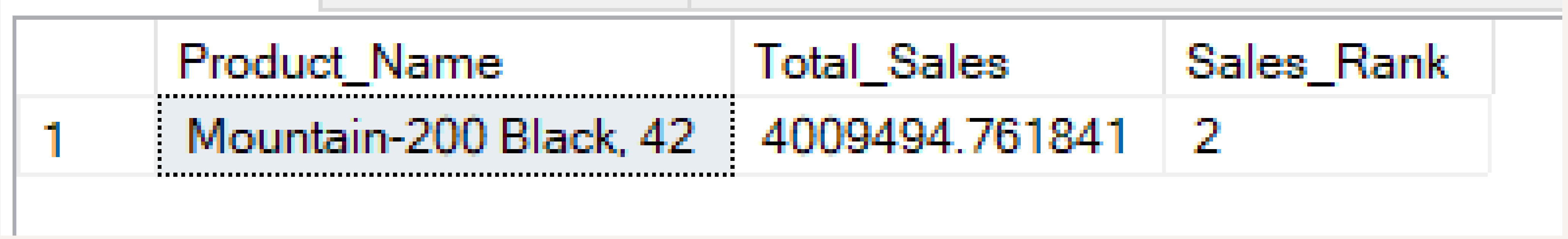
in Microsoft SQL Server

QUERY



```sql
with TotalSaleProduct as (
    select
        p.Name as Product_Name,
        sum(sod.LineTotal) as Total_Sales
    from AdventureWorks2019.Production.Product p
    inner join AdventureWorks2019.Sales.SalesOrderDetail sod on
        p.ProductID = sod.ProductID
    group by p.name
),
rankproduct as (
    select
        Product_Name,
        Total_Sales,
        rank() over (order by Total_Sales desc) as Sales_Rank
        from TotalSaleProduct
)
select top 1
    Product_Name,
    Total_Sales,
    Sales_Rank
from rankproduct
where
Sales_Rank = 2
```

OUTPUT



	Product_Name	Total_Sales	Sales_Rank
1	Mountain-200 Black, 42	4009494.761841	2

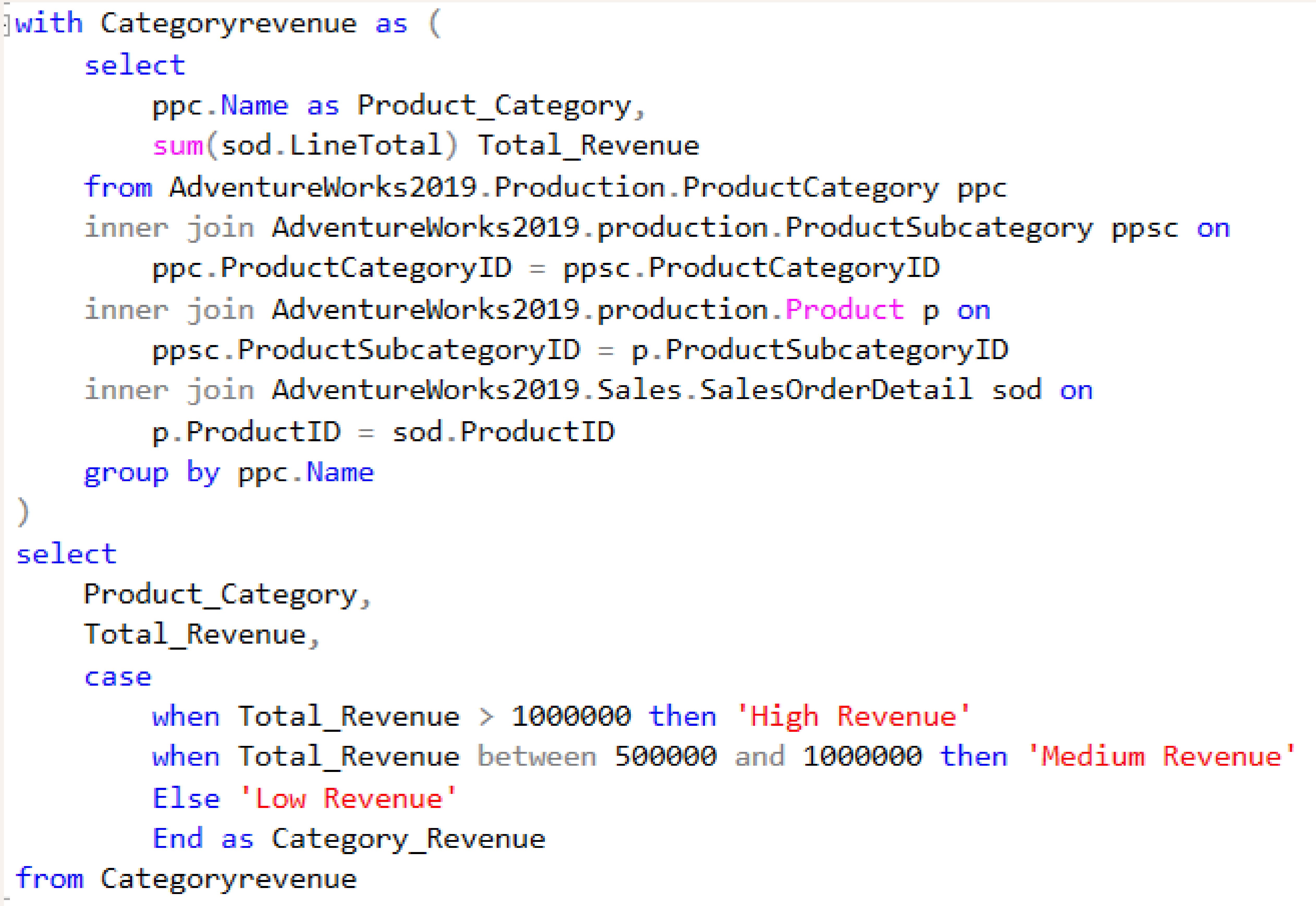
SQL Project

15. Calculate the total revenue generated by each product category and classify them as 'High Revenue' if the total revenue is greater than $1,000,000, 'Medium Revenue' if between $500,000 and $1,000,000, and 'Low Revenue' otherwise.

SQL Project

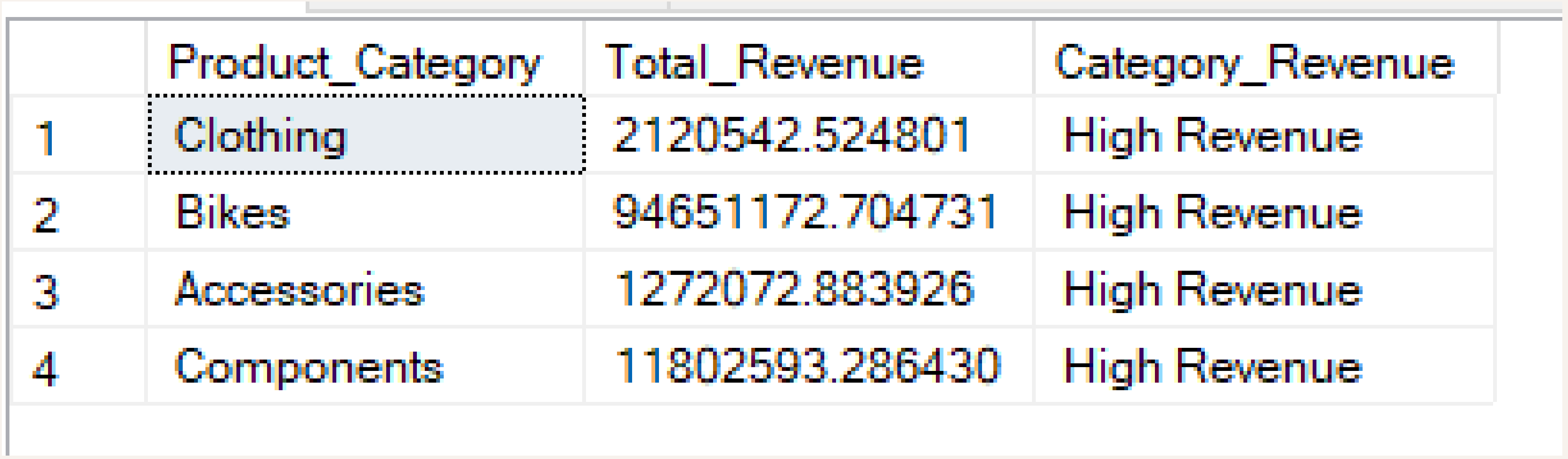
in Microsoft SQL Server

QUERY



```sql
with Categoryrevenue as (
    select
        ppc.Name as Product_Category,
        sum(sod.LineTotal) Total_Revenue
    from AdventureWorks2019.Production.ProductCategory ppc
    inner join AdventureWorks2019.production.ProductSubcategory ppsc on
        ppc.ProductCategoryID = ppsc.ProductCategoryID
    inner join AdventureWorks2019.production.Product p on
        ppsc.ProductSubcategoryID = p.ProductSubcategoryID
    inner join AdventureWorks2019.Sales.SalesOrderDetail sod on
        p.ProductID = sod.ProductID
    group by ppc.Name
)
select
    Product_Category,
    Total_Revenue,
    case
        when Total_Revenue > 1000000 then 'High Revenue'
        when Total_Revenue between 500000 and 1000000 then 'Medium Revenue'
        Else 'Low Revenue'
        End as Category_Revenue
from Categoryrevenue
```

OUTPUT



	Product_Category	Total_Revenue	Category_Revenue
1	Clothing	2120542.524801	High Revenue
2	Bikes	94651172.704731	High Revenue
3	Accessories	1272072.883926	High Revenue
4	Components	11802593.286430	High Revenue

Thank You