ASSIGNMENT NO. - 3

Student Name: Basavraj Nagappa Jaliminche

Student Id: 8800149

PROGRAM: Reporting Systems and Database Development (1517)

Course Name: Relational Databases PROG 8590

ASSIGNMENT NO. - 3

Q.1 We need to know the number of products we have in the PurchaseOrderDetail table. (count the number of un-repeated productid)

Query:

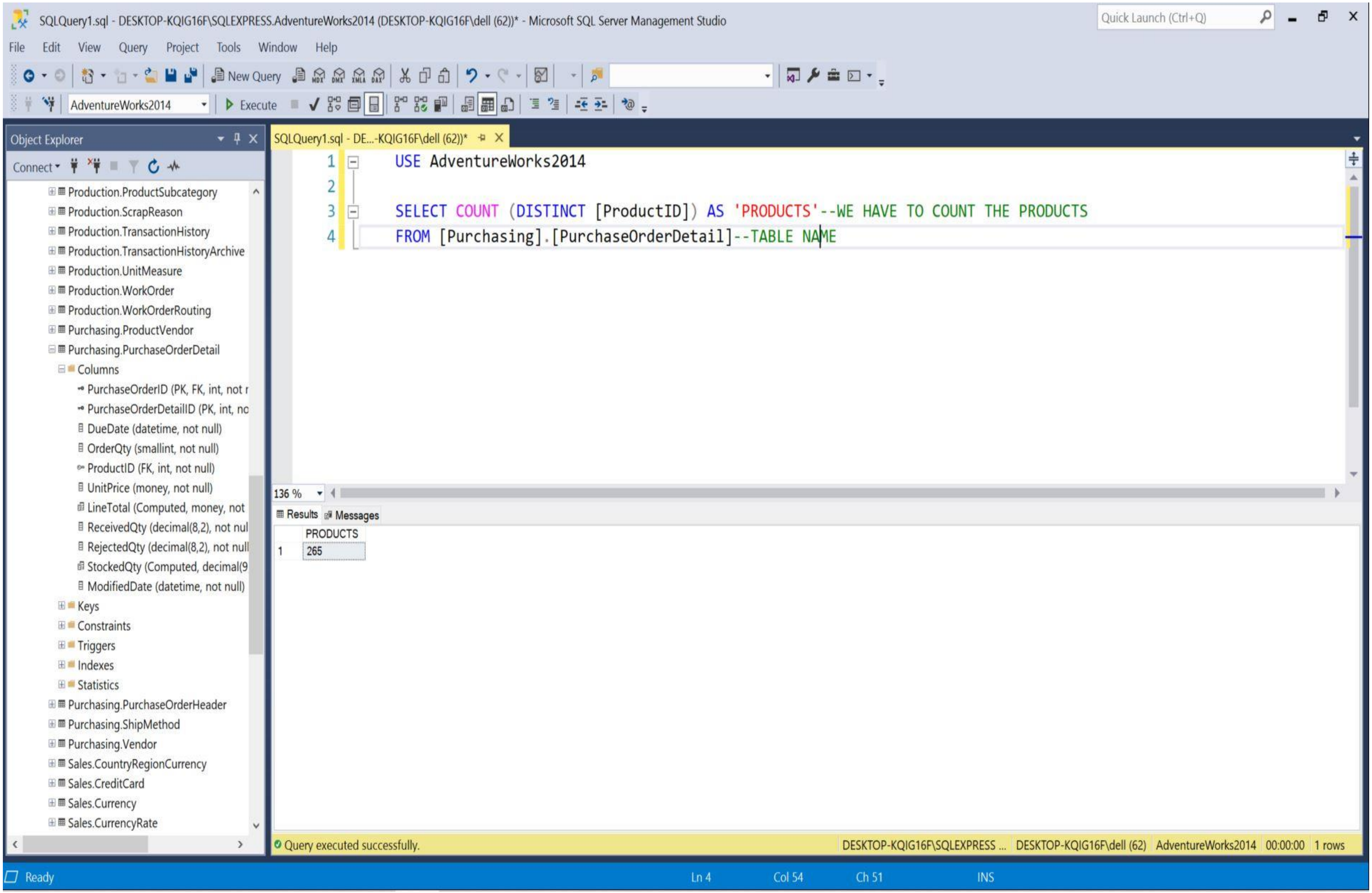
```
USE AdventureWorks2014

SELECT COUNT (DISTINCT [ProductID]) AS 'PRODUCTS'--WE HAVE TO COUNT THE PRODUCTS
FROM [Purchasing].[PurchaseOrderDetail]--TABLE NAME
```

Description:

- We have to select the products from the table `[Purchasing].[PurchaseOrderDetail]`
- We have to count the number of products so we have used the count function and the products does not duplicate so that we have to use DISTICT keyword.

Screenshot of the query is provided on the further page ..


SQLQuery1.sql - DESKTOP-KQIG16F\SQLEXPRESS.AdventureWorks2014 (DESKTOP-KQIG16F\dell (62))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Query
Project
Tools
Window
Help
New Query
AdventureWorks2014
Execute
Object Explorer
Connect
Production.ProductSubcategory
Production.ScrapReason
Production.TransactionHistory
Production.TransactionHistoryArchive
Production.UnitMeasure
Production.WorkOrder
Production.WorkOrderRouting
Purchasing.ProductVendor
Purchasing.PurchaseOrderDetail
Columns
PurchaseOrderID (PK, FK, int, not r
PurchaseOrderDetailID (PK, int, no
DueDate (datetime, not null)
OrderQty (smallint, not null)
ProductID (FK, int, not null)
UnitPrice (money, not null)
LineTotal (Computed, money, not
ReceivedQty (decimal(8,2), not nul
RejectedQty (decimal(8,2), not null
StockedQty (Computed, decimal(9
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
Indexes
Statistics
Purchasing.PurchaseOrderHeader
Purchasing.ShipMethod
Purchasing.Vendor
Sales.CountryRegionCurrency
Sales.CreditCard
Sales.Currency
Sales.CurrencyRate
SQLQuery1.sql - DE...-KQIG16F\dell (62))*
136 %
Results
Messages
Ready
Ln 4
Col 54
Ch 51
INS

```sql
USE AdventureWorks2014

SELECT COUNT (DISTINCT [ProductID]) AS 'PRODUCTS'--WE HAVE TO COUNT THE PRODUCTS
FROM [Purchasing].[PurchaseOrderDetail]--TABLE NAME
```

	PRODUCTS
1	265

```
Query executed successfully.    DESKTOP-KQIG16F\SQLEXPRESS ...   DESKTOP-KQIG16F\dell (62)   AdventureWorks2014   00:00:00   1 rows
```

Query no 1

Q.2 Write a query to show the productID of the most profitable product(ignore production costs) after price and order quantity are considered (maximum amount of money gained for each product id) • Use SUM and group by to get the best result. • HINT: Should be 3358797.75

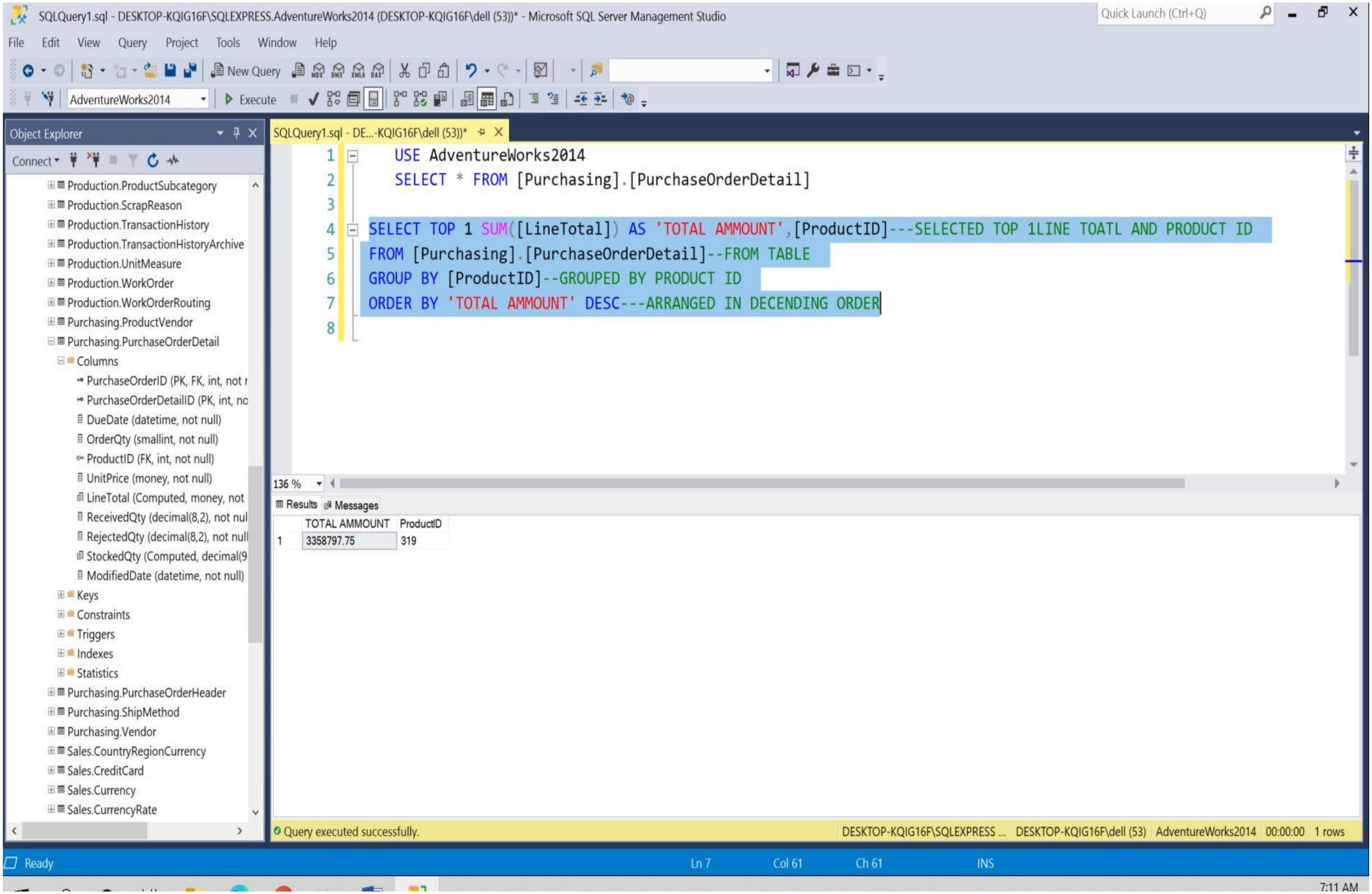
Query:

```
            SELECT TOP 1 SUM([LineTotal]) AS 'TOTAL AMMOUNT',[ProductID]---SELECTED TOP 1LINE TOATL AND 
PRODUCT ID
             FROM [Purchasing].[PurchaseOrderDetail]--FROM TABLE
              GROUP BY [ProductID]--GROUPED BY PRODUCT ID
            ORDER BY 'TOTAL AMMOUNT' DESC---ARRANGED IN DECENDING ORDER
```

Description:

- We have selected table [Purchasing].[PurchaseOrderDetail]
- We have made sum of line total and then made group and we have arranged the result in the descending manner.
- After that we have selected the top most of the result along with the product id.

Screenshot of the query is provided on the further page ..


SQLQuery1.sql - DESKTOP-KQIG16F\SQLEXPRESS.AdventureWorks2014 (DESKTOP-KQIG16F\dell (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Query
Project
Tools
Window
Help
New Query
AdventureWorks2014
Execute
Object Explorer
Connect
Production.ProductSubcategory
Production.ScrapReason
Production.TransactionHistory
Production.TransactionHistoryArchive
Production.UnitMeasure
Production.WorkOrder
Production.WorkOrderRouting
Purchasing.ProductVendor
Purchasing.PurchaseOrderDetail
Columns
PurchaseOrderID (PK, FK, int, not r
PurchaseOrderDetailID (PK, int, no
DueDate (datetime, not null)
OrderQty (smallint, not null)
ProductID (FK, int, not null)
UnitPrice (money, not null)
LineTotal (Computed, money, not
ReceivedQty (decimal(8,2), not nul
RejectedQty (decimal(8,2), not null
StockedQty (Computed, decimal(9
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
Indexes
Statistics
Purchasing.PurchaseOrderHeader
Purchasing.ShipMethod
Purchasing.Vendor
Sales.CountryRegionCurrency
Sales.CreditCard
Sales.Currency
Sales.CurrencyRate
SQLQuery1.sql - DE...-KQIG16F\dell (53))*
136 %
Results
Messages
DESKTOP-KQIG16F\SQLEXPRESS ...
DESKTOP-KQIG16F\dell (53)
AdventureWorks2014
00:00:00
1 rows
Ready
Ln 7
Col 61
Ch 61
INS
7:11 AM

```sql
USE AdventureWorks2014
SELECT * FROM [Purchasing].[PurchaseOrderDetail]

SELECT TOP 1 SUM([LineTotal]) AS 'TOTAL AMMOUNT',[ProductID]---SELECTED TOP 1LINE TOATL AND PRODUCT ID
FROM [Purchasing].[PurchaseOrderDetail]--FROM TABLE
GROUP BY [ProductID]--GROUPED BY PRODUCT ID
ORDER BY 'TOTAL AMMOUNT' DESC---ARRANGED IN DECENDING ORDER
```

```
   TOTAL AMMOUNT  ProductID
1  3358797.75     319
```

Query executed successfully.

Query 2

Q.3 Write a query to show the names of the top 5 most profitable products, as in question 2. Remember to take both price and quantity sold into account. • You must join two tables.

Query:

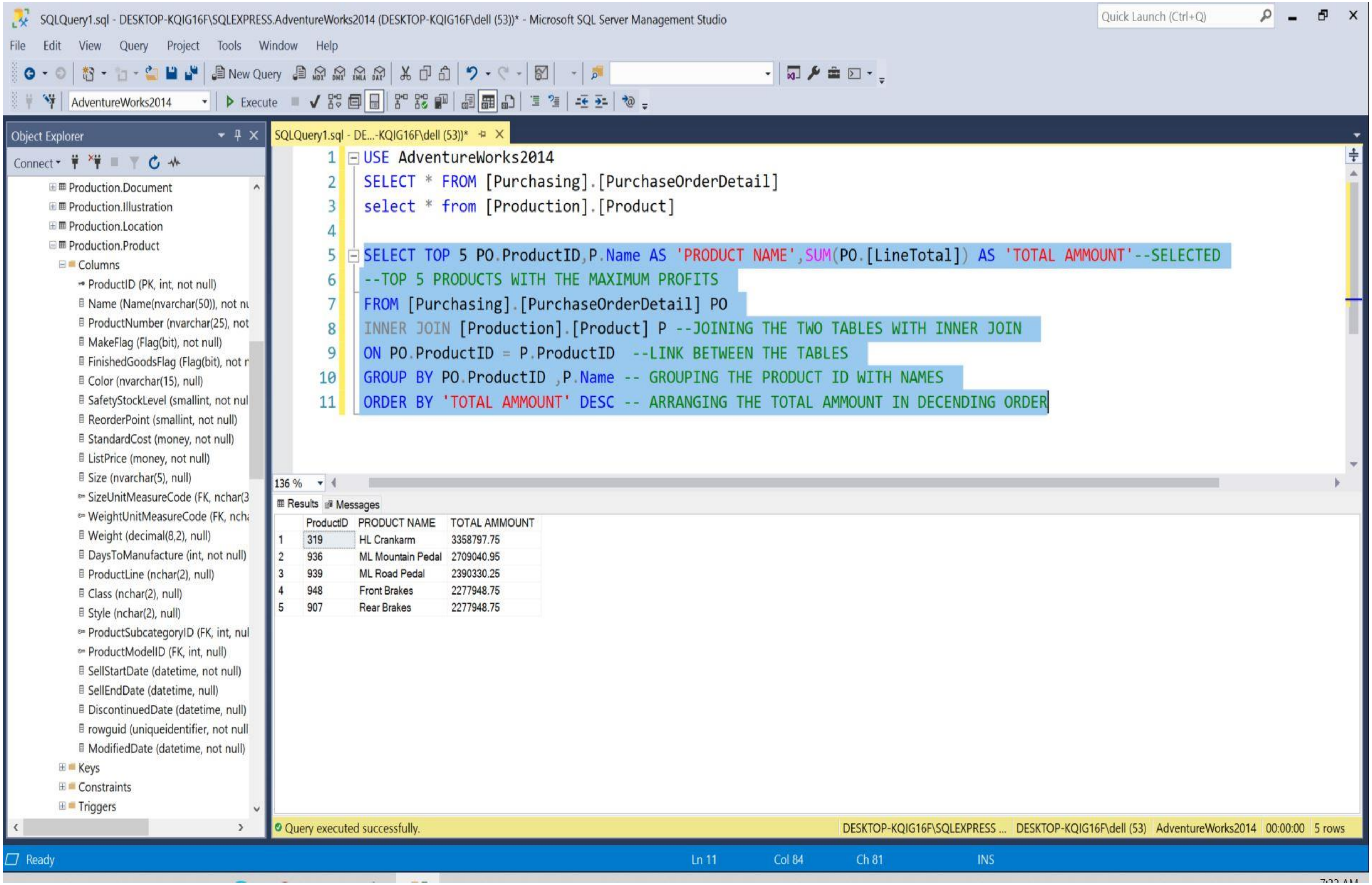
```
SELECT TOP 5 PO.ProductID,P.Name AS 'PRODUCT NAME',SUM(PO.[LineTotal]) AS 'TOTAL AMMOUNT'--SELECTED
--TOP 5 PRODUCTS WITH THE MAXIMUM PROFITS
FROM [Purchasing].[PurchaseOrderDetail] PO
INNER JOIN [Production].[Product] P --JOINING THE TWO TABLES WITH INNER JOIN
ON PO.ProductID = P.ProductID  --LINK BETWEEN THE TABLES
GROUP BY PO.ProductID ,P.Name -- GROUPING THE PRODUCT ID WITH NAMES

ORDER BY 'TOTAL AMMOUNT' DESC -- ARRANGING THE TOTAL AMMOUNT IN DECENDING ORDER
```

Description:

- We have selected the two tables `[Purchasing].[PurchaseOrderDetail] and Production].[Product]`
- Then we have joined the two tables with the link `ProductID.`
- After joining the two tables we have groped the results with tha product id and product name.
- After that we have arranged the sum of line total in the descending order.
- For showing the results we hae selected the top 5 products with the product name and total amount.

Screenshot of the query is provided on the further page ..



SQLQuery1.sql - DESKTOP-KQIG16F\SQLEXPRESS.AdventureWorks2014 (DESKTOP-KQIG16F\dell (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Query
Project
Tools
Window
Help
New Query
AdventureWorks2014
Execute
Object Explorer
Connect
Production.Document
Production.Illustration
Production.Location
Production.Product
Columns
ProductID (PK, int, not null)
Name (Name(nvarchar(50)), not nu
ProductNumber (nvarchar(25), not
MakeFlag (Flag(bit), not null)
FinishedGoodsFlag (Flag(bit), not r
Color (nvarchar(15), null)
SafetyStockLevel (smallint, not nul
ReorderPoint (smallint, not null)
StandardCost (money, not null)
ListPrice (money, not null)
Size (nvarchar(5), null)
SizeUnitMeasureCode (FK, nchar(3
WeightUnitMeasureCode (FK, nchi
Weight (decimal(8,2), null)
DaysToManufacture (int, not null)
ProductLine (nchar(2), null)
Class (nchar(2), null)
Style (nchar(2), null)
ProductSubcategoryID (FK, int, nul
ProductModelID (FK, int, null)
SellStartDate (datetime, not null)
SellEndDate (datetime, null)
DiscontinuedDate (datetime, null)
rowguid (uniqueidentifier, not null
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
SQLQuery1.sql - DE...-KQIG16F\dell (53))*
136 %
Results
Messages
DESKTOP-KQIG16F\SQLEXPRESS ...
DESKTOP-KQIG16F\dell (53)
AdventureWorks2014
00:00:00
5 rows
Ready
Ln 11
Col 84
Ch 81
INS

```
USE AdventureWorks2014
SELECT * FROM [Purchasing].[PurchaseOrderDetail]
select * from [Production].[Product]

SELECT TOP 5 PO.ProductID,P.Name AS 'PRODUCT NAME',SUM(PO.[LineTotal]) AS 'TOTAL AMMOUNT'--SELECTED
--TOP 5 PRODUCTS WITH THE MAXIMUM PROFITS
FROM [Purchasing].[PurchaseOrderDetail] PO
INNER JOIN [Production].[Product] P --JOINING THE TWO TABLES WITH INNER JOIN
ON PO.ProductID = P.ProductID  --LINK BETWEEN THE TABLES
GROUP BY PO.ProductID ,P.Name -- GROUPING THE PRODUCT ID WITH NAMES
ORDER BY 'TOTAL AMMOUNT' DESC -- ARRANGING THE TOTAL AMMOUNT IN DECENDING ORDER
```

	ProductID	PRODUCT NAME	TOTAL AMMOUNT
1	319	HL Crankarm	3358797.75
2	936	ML Mountain Pedal	2709040.95
3	939	ML Road Pedal	2390330.25
4	948	Front Brakes	2277948.75
5	907	Rear Brakes	2277948.75

Query executed successfully.

Query 3

Q.4 Write a query to show all product ID with the stock quantity less than average stock quantity. • You have to use sub query.

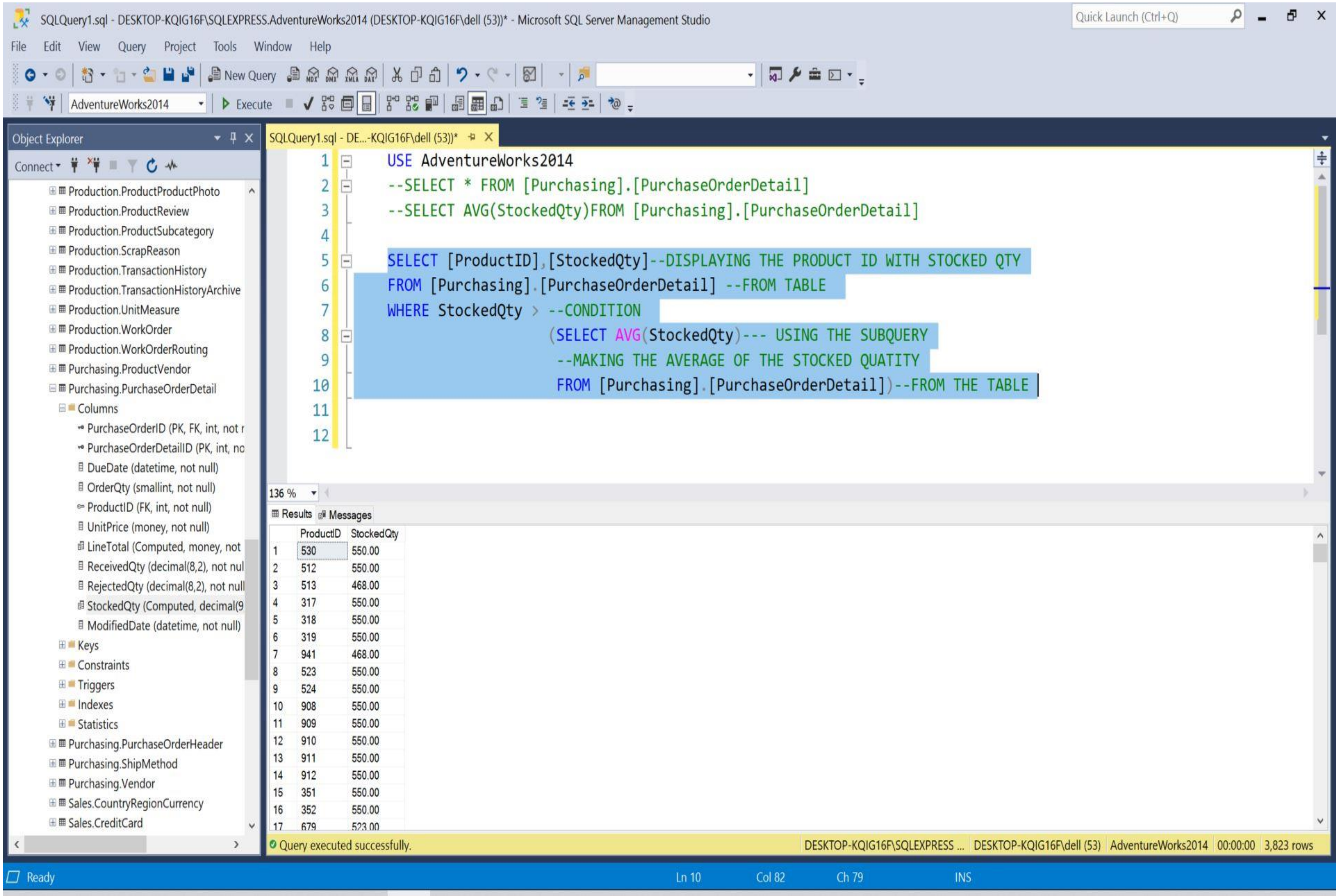
Query:

```
    SELECT [ProductID],[StockedQty]--DISPLAYING THE PRODUCT ID WITH STOCKED QTY
    FROM [Purchasing].[PurchaseOrderDetail] --FROM TABLE
    WHERE StockedQty > --CONDITION
                         (SELECT AVG(StockedQty)--- USING THE SUBQUERY
                              --MAKING THE AVERAGE OF THE STOCKED QUATITY
                          FROM [Purchasing].[PurchaseOrderDetail])--FROM THE TABLE
```

Description:

- We have select the table [Purchasing].[PurchaseOrderDetail].
- Then calculated average of the stocked qty.
- Then using subquery, we have given condition greater than the average stocked qty.
- Then we have displayed the results of product id with the stocked qty.

Screenshot of the query is provided on the further page ..


SQLQuery1.sql - DESKTOP-KQIG16F\SQLEXPRESS.AdventureWorks2014 (DESKTOP-KQIG16F\dell (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Query
Project
Tools
Window
Help
New Query
AdventureWorks2014
Execute
Object Explorer
Connect
Production.ProductProductPhoto
Production.ProductReview
Production.ProductSubcategory
Production.ScrapReason
Production.TransactionHistory
Production.TransactionHistoryArchive
Production.UnitMeasure
Production.WorkOrder
Production.WorkOrderRouting
Purchasing.ProductVendor
Purchasing.PurchaseOrderDetail
Columns
PurchaseOrderID (PK, FK, int, not r
PurchaseOrderDetailID (PK, int, no
DueDate (datetime, not null)
OrderQty (smallint, not null)
ProductID (FK, int, not null)
UnitPrice (money, not null)
LineTotal (Computed, money, not
ReceivedQty (decimal(8,2), not nul
RejectedQty (decimal(8,2), not null
StockedQty (Computed, decimal(9
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
Indexes
Statistics
Purchasing.PurchaseOrderHeader
Purchasing.ShipMethod
Purchasing.Vendor
Sales.CountryRegionCurrency
Sales.CreditCard
SQLQuery1.sql - DE...-KQIG16F\dell (53))*
136 %
Results
Messages

```
USE AdventureWorks2014
--SELECT * FROM [Purchasing].[PurchaseOrderDetail]
--SELECT AVG(StockedQty)FROM [Purchasing].[PurchaseOrderDetail]

SELECT [ProductID],[StockedQty]--DISPLAYING THE PRODUCT ID WITH STOCKED QTY
FROM [Purchasing].[PurchaseOrderDetail] --FROM TABLE
WHERE StockedQty > --CONDITION
                   (SELECT AVG(StockedQty)--- USING THE SUBQUERY
                    --MAKING THE AVERAGE OF THE STOCKED QUATITY
                    FROM [Purchasing].[PurchaseOrderDetail])--FROM THE TABLE
```

	ProductID	StockedQty
1	530	550.00
2	512	550.00
3	513	468.00
4	317	550.00
5	318	550.00
6	319	550.00
7	941	468.00
8	523	550.00
9	524	550.00
10	908	550.00
11	909	550.00
12	910	550.00
13	911	550.00
14	912	550.00
15	351	550.00
16	352	550.00
17	679	523.00

```
Query executed successfully.    DESKTOP-KQIG16F\SQLEXPRESS ...    DESKTOP-KQIG16F\dell (53)    AdventureWorks2014    00:00:00    3,823 rows
Ready    Ln 10    Col 82    Ch 79    INS
```

Query 4

Q.5 We need to know the product id and the modified date of the products with special offer “Half-Price Pedal Sale”.

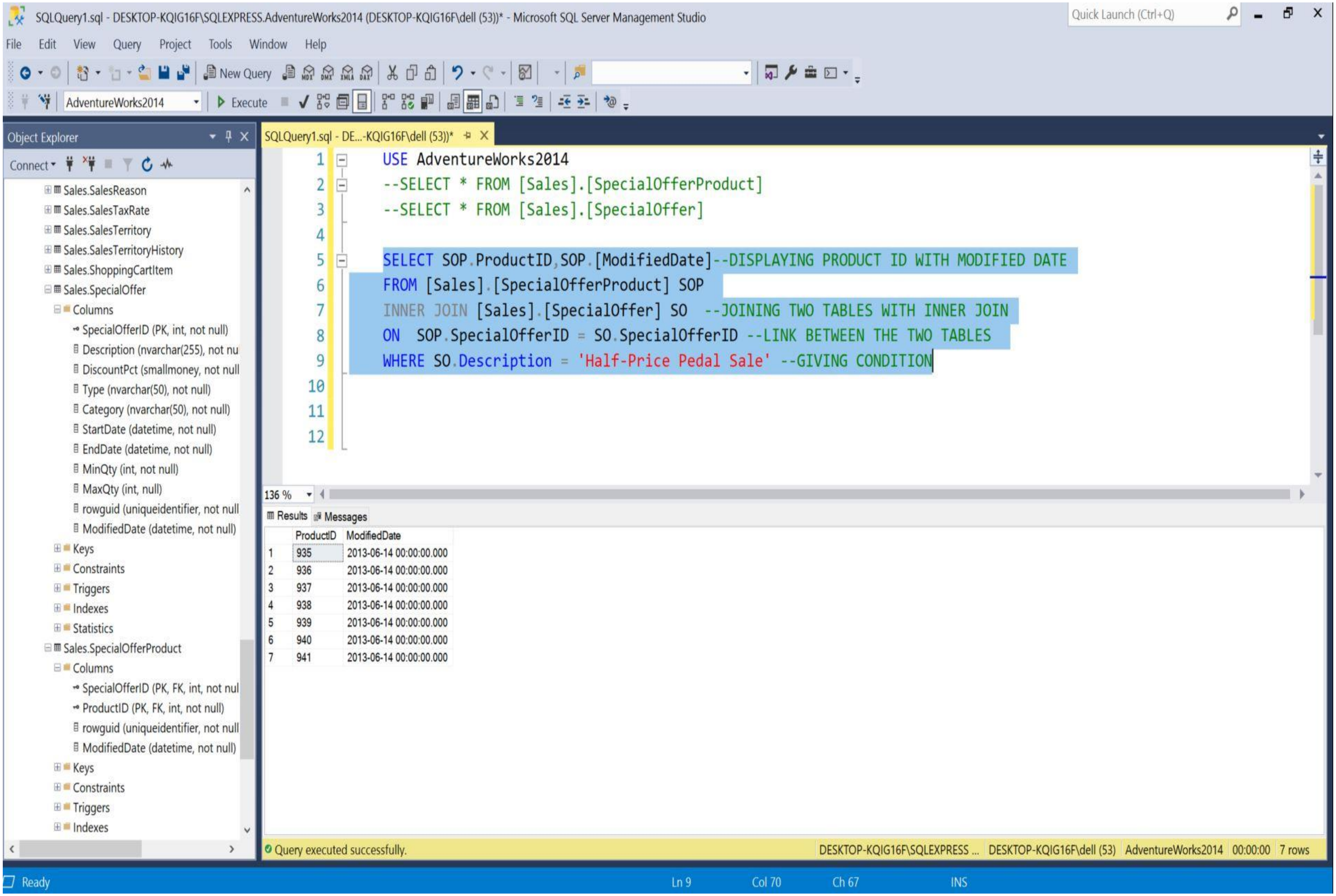
Query:

```
SELECT SOP. ProductID,SOP.[ModifiedDate]--DISPLAYING PRODUCT ID WITH MODIFIED DATE
FROM [Sales].[SpecialOfferProduct] SOP
INNER JOIN [Sales].[SpecialOffer] SO  --JOINING TWO TABLES WITH INNER JOIN
ON  SOP.SpecialOfferID = SO.SpecialOfferID --LINK BETWEEN THE TWO TABLES
WHERE SO.Description = 'Half-Price Pedal Sale' --GIVING CONDITION
```

Description:

- We have selected two tables `[SpecialOfferProduct]` and `[Sales].[SpecialOffer]`.
- Then we have joined them with inner join .
- Link between the two tables is `SpecialOfferID`.
- Then we have put condition as `Description = 'Half-Price Pedal Sale'`
- Then displayed product id with modified date.

Screenshot of the query is provided on the further page ..


SQLQuery1.sql - DESKTOP-KQIG16F\SQLEXPRESS.AdventureWorks2014 (DESKTOP-KQIG16F\dell (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Query
Project
Tools
Window
Help
New Query
AdventureWorks2014
Execute
Object Explorer
Connect
Sales.SalesReason
Sales.SalesTaxRate
Sales.SalesTerritory
Sales.SalesTerritoryHistory
Sales.ShoppingCartItem
Sales.SpecialOffer
Columns
SpecialOfferID (PK, int, not null)
Description (nvarchar(255), not nu
DiscountPct (smallmoney, not null
Type (nvarchar(50), not null)
Category (nvarchar(50), not null)
StartDate (datetime, not null)
EndDate (datetime, not null)
MinQty (int, not null)
MaxQty (int, null)
rowguid (uniqueidentifier, not null
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
Indexes
Statistics
Sales.SpecialOfferProduct
Columns
SpecialOfferID (PK, FK, int, not nul
ProductID (PK, FK, int, not null)
rowguid (uniqueidentifier, not null
ModifiedDate (datetime, not null)
Keys
Constraints
Triggers
Indexes
SQLQuery1.sql - DE...-KQIG16F\dell (53))*
136 %
Results
Messages
DESKTOP-KQIG16F\SQLEXPRESS ...
DESKTOP-KQIG16F\dell (53)
AdventureWorks2014
00:00:00
7 rows
Ready
Ln 9
Col 70
Ch 67
INS

```
USE AdventureWorks2014
--SELECT * FROM [Sales].[SpecialOfferProduct]
--SELECT * FROM [Sales].[SpecialOffer]

SELECT SOP.ProductID,SOP.[ModifiedDate]--DISPLAYING PRODUCT ID WITH MODIFIED DATE
FROM [Sales].[SpecialOfferProduct] SOP
INNER JOIN [Sales].[SpecialOffer] SO  --JOINING TWO TABLES WITH INNER JOIN
ON  SOP.SpecialOfferID = SO.SpecialOfferID --LINK BETWEEN THE TWO TABLES
WHERE SO.Description = 'Half-Price Pedal Sale' --GIVING CONDITION
```

	ProductID	ModifiedDate
1	935	2013-06-14 00:00:00.000
2	936	2013-06-14 00:00:00.000
3	937	2013-06-14 00:00:00.000
4	938	2013-06-14 00:00:00.000
5	939	2013-06-14 00:00:00.000
6	940	2013-06-14 00:00:00.000
7	941	2013-06-14 00:00:00.000

Query executed successfully.

Query 5